

**A CHATBOT FOR TEACHING SOFTWARE  
TESTING CTFL SYLLABUS**

**TEH WING FUNG**

**UNIVERSITI TUNKU ABDUL RAHMAN**

**A CHATBOT FOR TEACHING SOFTWARE TESTING CTFL  
SYLLABUS**

**TEH WING FUNG**

**A project report submitted in partial fulfilment of the  
requirements for the award of Bachelor of Software  
Engineering (Honours)**

**Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman**

**September 2025**

## DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Name : Teh Wing Fung \_\_\_\_\_

ID No. : 2104658 \_\_\_\_\_

Date : 2/7/2025 \_\_\_\_\_

## **COPYRIGHT STATEMENT**

© 2025, TEH WING FUNG. All right reserved.

This final year project report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Software Engineering (Honours) at Universiti Tunku Abdul Rahman (UTAR). This final year project report represents the work of the author, except where due acknowledgement has been made in the text. No part of this final year project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

## ABSTRACT

Software testing is a vital process in software development, ensuring product quality by minimizing defects and preventing errors that affect usability. The ISTQB Certified Tester Foundation Level (CTFL) syllabus introduces fundamental concepts of software testing, serving as an important resource for students and lecturers. This project develops a web application with an integrated chatbot specifically designed to teach the CTFL syllabus. Developed using robust Laravel backend and React front end, integrating a large-language-model-driven chatbot to provide concise explanations, practical examples, and automatically generated quiz questions to reinforce user understanding. The development followed V-Model methodology, conducting different kind of testing including unit, integration, system, and user acceptance to ensure the web application to implement all the stated requirements from users, and to reinforce its core educational objectives through systematic validation. In the results of conducted testings, the web application is proven to have outstanding performance without defects affecting the usage. By combining natural language interaction with structured syllabus content, the system offers an engaging and accessible learning tool that enhances comprehension, supports teaching, and better prepares learners for CTFL certification and implement professional practice in the workfield.

**Keywords:** Software testing, CTFL, chatbot, web application, large language model, learning tool

**Subject Area:** QA76.75-76.765 Computer software

## TABLE OF CONTENTS

<b>DECLARATION</b>	<b>i</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>TABLE OF CONTENTS</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF SYMBOLS / ABBREVIATIONS</b>	<b>xiv</b>
<b>LIST OF APPENDICES</b>	<b>xv</b>

## CHAPTER

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 General Introduction	1
	1.2 Importance of the Study	1
	1.3 Problem Statement	2
	1.3.1 Lack in practical applications	3
	1.3.2 Understanding concepts in the syllabus	3
	1.3.3 Unfamiliar with terms used in the syllabus	3
	1.4 Aim and Objectives	4
	1.5 Scope and Limitation of the Study	4
	1.5.1 Target users	4
	1.5.2 Modules	5
	1.6 Project Solution	6
	1.7 Project Approach	8
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>10</b>
	2.1 Introduction	10
	2.2 Benefits and disadvantages of students using AI Chatbots for education	10
	2.2.1 Benefits from students' perspective	10
	2.2.2 Disadvantage of students using chatbots for education purpose	11

2.3	Existing usage of chatbots in education sector	12
2.3.1	AI-based Chatbot for Practicing Responsive Teaching in Mathematics	12
2.3.2	Chatbot on Students' Learning in Ghanaian Higher Education	13
2.4	Comparison of AI models on the Internet	13
2.4.1	Gemini	13
2.4.2	ChatGPT	15
2.4.3	DeepSeek	16
2.4.4	Grok	17
2.4.5	Comparison between each AI Models	18
2.5	Development Methodologies	19
2.5.1	Agile	20
2.5.2	V-Model	22
2.5.3	Lean development	25
2.5.4	Comparison of Development Methodologies	27
2.6	Web Application Framework	28
2.6.1	Front-end Web Application Frameworks	28
2.6.2	Back-end Web Application Frameworks	31
2.7	Large Language Models (LLMs)	34
2.7.1	Gemma 2 2B	34
2.7.2	Phi-3.5-mini	35
2.7.3	LLaMA 3.2 3B	36
2.7.4	Comparison Table between LLMs	37
2.8	Conclusion on Literature Review	38
<b>3</b>	<b>METHODOLOGY AND WORK PLAN</b>	<b>41</b>
3.1	Introduction	41
3.2	Methodology	41
3.2.1	Gathering of requirements	41
3.2.2	Analysis on requirements	43
3.2.3	Planning for the project	43
3.2.4	Development phase	44
3.2.5	Testing on the web application	44

3.3	Development tools	45
3.3.1	Visual Studio Code	45
3.3.2	React	45
3.3.3	Laravel	46
3.3.4	MySQL	46
3.3.5	Draw.io	46
3.3.6	Axure RP	46
3.3.7	Google Colab	46
3.3.8	SQLite	47
3.4	Work Plan	47
3.4.1	Work Breakdown Structure (WBS)	47
3.5	Gantt Chart	49
3.5.1	Gantt Chart for FYP 1	50
3.5.2	Gantt Chart for FYP 2	50
3.6	Test plan	51
3.6.1	Objective of test plan	51
3.6.2	Scope of Test	51
3.6.3	Items need to be tested	51
3.6.4	Chatbot web application features to be tested	51
3.6.5	Chatbot web application features not to be tested	52
3.6.6	Test Basis	52
3.6.7	Conditions of Test	52
3.6.8	Criteria of Entry	53
3.6.9	Criteria of Exit	53
<b>4</b>	<b>PROJECT INITIAL SPECIFICATION</b>	<b>54</b>
4.1	Introduction	54
4.2	Requirements specification	54
4.2.1	Collecting Requirements	54
4.2.2	Functional Requirements	57
4.2.3	Non-functional Requirements	58
4.3	Use Case Diagram	59



4.3.1	Use Case Diagram for A chatbot for teaching software testing CTFL syllabus	60
4.4	Use case description	60
4.4.1	Use case descriptions for students/ self-learners / software testers who want to take the CTFL exam	61
4.4.2	Use case descriptions for lecturers/ trainers from the training academy	70
4.4.3	Use case descriptions for Admin	73
4.5	Flowchart of A chatbot for teaching software testing CTFL syllabus web application	81
<b>5</b>	<b>SYSTEM DESIGN</b>	<b>82</b>
5.1	Introduction	82
5.2	System Architecture Design	82
5.3	Entity Relationship Diagram (ERD)	83
5.3.1	Data Dictionary	83
5.4	System User Interface (UI) Design	86
5.4.1	Login UI	87
5.4.2	Registration UI	88
5.4.3	Forget Password UI	89
5.4.4	Chat Page UI	90
5.4.5	Sidebar UI	91
5.4.6	Profile Dropdown Menu UI	92
5.4.7	Starred Message UI	92
5.4.8	Upload File UI	93
5.4.9	Profile Page UI	94
5.4.10	Dashboard Page UI	95
5.4.11	Add User UI	96
<b>6</b>	<b>IMPLEMENTATION</b>	<b>97</b>
6.1	Introduction	97
6.2	Implementation of front-end	97
6.2.1	Account Registration Module	97
6.2.2	Chatbot Module	101
6.2.3	Star Message module	105

	6.2.4 Upload file module	108
6.3	Implementation of back-end	110
	6.3.1 Registration process	110
	6.3.2 Login process	111
	6.3.3 Reset password process	113
	6.3.4 Chat page	114
	6.3.5 Star message	118
	6.3.6 Upload file	120
	6.3.7 Chatbot	121
<b>7</b>	<b>TESTING</b>	<b>124</b>
	7.1 Introduction	124
	7.2 Unit Testing	125
	7.3 Integration Testing	181
	7.4 System Testing	184
	7.5 User Acceptance Testing (UAT)	187
	7.6 Analysis on the test results	187
<b>8</b>	<b>CONCLUSION AND FUTURE WORKS</b>	<b>188</b>
	8.1 Introduction	188
	8.2 Limitations	188
	8.3 Future works	188
	8.4 Conclusion	189
	<b>REFERENCES</b>	<b>190</b>
	<b>APPENDICES</b>	<b>193</b>

## LIST OF TABLES

Table 2.1:	Comparison table between each AI Models	18
Table 2.2:	Comparison Table of Each Software Development Methodologies.	27
Table 2.3:	Comparison Table of Each Front-end Web Application Frameworks.	30
Table 2.4:	Comparison Table of Each Back-end Web Application Frameworks.	33
Table 2.5:	Comparison Table of Each Large Language Models.	37
Table 4.1:	Functional Requirements for Students/ Self-learners/ Software Testers	57
Table 4.2:	Functional Requirements for Lecturers/ Trainers	58
Table 4.3:	Functional Requirements for Admins	58
Table 4.4:	Non functional requirements	58
Table 4.5:	Sign up use case description	61
Table 4.6:	Login use case description	62
Table 4.7:	Edit profile use case description	63
Table 4.8:	Edit username/ email use case description	64
Table 4.9:	Update password use case description	65
Table 4.10:	Delete own account use case description	66
Table 4.11:	Input questions use case description	67
Table 4.12:	Ask CTFL syllabus questions use case description	68
Table 4.13:	Generate questions/ quizzes use case description	69
Table 4.14:	Sign up (lecturers) use case description	71
Table 4.15:	Upload files use case description	72
Table 4.16:	Sign up (admin) use case description	73
Table 4.17:	Dashboard Management use case description	74

Table 4.18:	Deactivate/ activate account use case description	75
Table 4.19:	Delete account use case description	76
Table 4.20:	View uploaded files use case description	77
Table 4.21:	Download uploaded files use case description	78
Table 4.22:	Delete uploaded files use case description	79
Table 5.1:	Data dictionary for users table	83
Table 5.2:	Data dictionary for chats table	84
Table 5.3:	Data dictionary for messages table	84
Table 5.4:	Data dictionary for starred_messages table	85
Table 5.5:	Data dictionary for uploaded_files table	86
Table 7.1:	Account Registration Module Test Cases	125
Table 7.2:	Chatbot Module Test Cases	152
Table 7.3:	Quiz Module Test Cases	158
Table 7.4:	Star Message Module Test Cases	165
Table 7.5:	Upload file Module Test Cases	170
Table 7.6:	Integration testing test cases	182
Table 7.7:	System testing test cases	185

## LIST OF FIGURES

Figure 1.1:	Overall review of using the chatbot	8
Figure 1.2:	RAG procedure (Adopted from <i>Retrieval-Augmented Generation (RAG)</i>   Deepgram, n.d.)	8
Figure 2.1:	Gemini summarizing software testing	14
Figure 2.2:	Sample conversation of Gemini	15
Figure 2.3:	Sample conversation of ChatGPT	16
Figure 2.4:	Sample conversation of DeepSeek	17
Figure 2.5:	Sample conversation of Grok	18
Figure 2.6:	Agile software development processes (adopted from GeeksForGeeks, 2024b)	22
Figure 2.7:	V-Model methodology (Adopted from Oppermann, 2023d)	23
Figure 3.1:	FYP 1 Gantt Chart	50
Figure 3.2:	FYP 2 Gantt Chart	50
Figure 4.1:	Result of survey questions (1)	54
Figure 4.2:	Result of survey questions (2)	55
Figure 4.3:	Result of survey questions (3)	55
Figure 4.4:	Result of survey questions (4)	56
Figure 4.5:	Result of survey questions (5)	56
Figure 4.6:	Use case diagram for the project.	60
Figure 4.7:	Flowchart	81
Figure 5.1:	System Architecture Diagram	82
Figure 5.2:	ERD	83
Figure 5.3:	Login page UI	87
Figure 5.4:	Registration page UI	88

Figure 5.5:	Registring for Lecturers/Trainers role	88
Figure 5.6:	Registring for Admin role	88
Figure 5.7:	Forget Password page UI	89
Figure 5.8:	Chat page UI	90
Figure 5.9:	Example chat in chat page	90
Figure 5.10:	Sidebar UI	91
Figure 5.11:	Sidebar actions	91
Figure 5.12:	Profile icon dropdown	92
Figure 5.13:	Starred Message page UI	92
Figure 5.14:	Upload file page UI	93
Figure 5.15:	Upper part of profile page UI	94
Figure 5.16:	Lower part of profile page UI	94
Figure 5.17:	Upper part of Dashboard page UI	95
Figure 5.18:	Lower part of Dashboard page UI	95
Figure 5.19:	Add user page UI	96
Figure 6.1:	Code snippet in Login.jsx (1)	97
Figure 6.2:	Code snippet in Login.jsx (2)	98
Figure 6.3:	Code snippet in Register.jsx (1)	99
Figure 6.4:	Code snippet in Register.jsx (2)	100
Figure 6.5:	Code snippet in ForgetPassword.jsx	101
Figure 6.6:	Code snippet in Chatbot.jsx (1)	101
Figure 6.7:	Code snippet in Chatbot.jsx (2)	102
Figure 6.8:	Code snippet in Chatbot.jsx (3)	102
Figure 6.9:	Code snippet in Chatbot.jsx (4)	103
Figure 6.10:	Star message code snippet from Chatbot.jsx (1)	105

Figure 6.11: Star message code snippet from Chatbot.jsx (2)	105
Figure 6.12: Code snippet in Starred.jsx	107
Figure 6.13: Code snippet in Upload.jsx (1)	108
Figure 6.14: Code snippet in Upload.jsx (2)	108
Figure 6.15: Code snippet in RegisteredUserController.php (1)	110
Figure 6.16: Code snippet in RegisteredUserController.php (2)	110
Figure 6.17: Code snippet for AuthenticatedSessionController.php (1)	111
Figure 6.18: Code snippet for AuthenticatedSessionController.php (2)	112
Figure 6.19: Code snippet in ForgotPasswordController.php	113
Figure 6.20: Code snippet in ChatController.php (1)	114
Figure 6.21: Code snippet in ChatController.php (2)	115
Figure 6.22: Code snippet in ChatController.php (3)	116
Figure 6.23: Code snippet in ChatController.php (4)	116
Figure 6.24: Code snippet in MessageController.php (1)	118
Figure 6.25: Code snippet in MessageController.php (2)	119
Figure 6.26: Code snippet in LecturerController.php	120
Figure 6.27: Code snippet in LLMController.php (1)	121
Figure 6.28: Code snippet in LLMController.php (2)	122
Figure 6.29: Code snippet in LLMController.php (3)	122

**LIST OF SYMBOLS / ABBREVIATIONS**

CTFL	Certified Tester Foundation Level
ISTQB	International Software Testing Quality Board
LLM	Large Language Model
RAG	Retrieval-Augmented Generation



## LIST OF APPENDICES

Appendix A: Survey Questions	193
Appendix B: Consent forms for UAT	197
Appendix C: UAT forms and results	205

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 General Introduction**

This report will be introducing the project of a chatbot for teaching software testing CTFL syllabus. International Software Testing Quality Board (ISTQB®) Certified Tester Foundation Level (CTFL) certification is the base of testing knowledge that can be implemented in actual testing scenarios in the world. CTFL syllabus is able to deliver an in-depth overview of the concepts and principles used in the testing field around the world, making the syllabus useful for every approach and process for software delivery. Some of the examples includes Waterfall, Agile, DevOps, and Continuous Delivery (Admin, 2025).

#### **1.2 Importance of the Study**

Software testing is an important factor when comes to developing a software. One of the testing principles is testing shows the presence of bugs, although the development of the software run smoothly, there could still be bugs in the software or the system. Hence testing is essential because this action can detect which part of the software contains bug or defect.

If testing is being ignored and not practiced, the quality of the software will drop drastically. This is due to the fact that bugs will occur which causes defect, this will sabotage the usage of every user and directly reducing satisfaction while using the software or system.

This leads to software engineers learning software testing becoming essential, which help to develop suitable test cases, run testing to check the quality of the software or system being developed. Not just limited to software engineers, those who are interested can learn as well to broaden knowledge. To understand more about software testing, learning the CTFL syllabus is recommended as the syllabus is the base of fundamental testing knowledge, CTFL certification is also needed if one is interested in pursuing other ISTQB® certifications.

Pursuing the CTFL certification is not an easy task, one should study the materials and also understand the concepts of software testing which can help to understand more about software testing and aid in the exams for the certification. After getting the certification and become a tester, one also need to implement the knowledge into work field. No doubt that real-world scenarios are totally different compared to those learned in textbooks, this could be a stepping stone for future testers as one do not have practical experience to help.

On top of that, there are less CTFL syllabus materials in the Internet which directly hinders students' learning for software testing. One could potentially face the issue of not fully understanding the syllabus. In addition, self-learners could also have the issue of not able to have the motivation to keep learning, giving up halfway during the studies because of the materials being too boring and do not have any fun interactions between lecturers and students. All these reasons could cause people to give up on learning software testing, to tackle these problems, this project propose on building a chatbot that teaches software testing CTFL syllabus.

How can a chatbot aid in teaching software testing? In this modern era, people have been utilizing chatbots such as ChatGPT, DeepSeek, Grok and many more AI chatbots to help in daily life. No doubt that chatbots can provide solutions to our problems, helping us in solving problems and broaden our knowledge. The chatbot can also help us to study, track our progress by giving some quizzes to us, using ChatGPT as an example, the response is human-alike which can give users an interactive and fun interactions, providing more engagement between learners and the chatbot. This also keep students to learn software testing CTFL syllabus without feeling burned out.

### **1.3 Problem Statement**

Some problems were discovered during the study of the project, the problems will be discussed in this section. On top of that, a survey is conducted to collect the problems of learning software testing CTFL syllabus, some of the main

problems will be discussed here. The result of the conducted survey will be attached in the appendix.

### **1.3.1 Lack in practical applications**

According to Szatmári, Gergely, and Beszédes (2023), those who are interested in learning the software testing CTFL syllabus can access to the Internet to obtain the learning materials although the availability of the materials are limited. However, after one learned the concepts, one does not have a chance to apply knowledge in real-world situations. Students, self learners or software tester who want to take the exam will be limited with the understanding of concepts only.

With the help of chatbot, chatbot can provide some quizzes that imitate some practical situations. This can test out students' understanding and allowing to utilize the knowledge studentss learned in a practical situation that is similar to real-world scenarios.

### **1.3.2 Understanding concepts in the syllabus**

In the results of the conducted survey, several respondents indicated that the main difficulty in learning the CTFL syllabus lies in understanding the concepts. This challenge often arises when studying or attempting exam questions, as the syllabus introduces new information and unfamiliar terminology that can be difficult to grasp without additional support.

To tackle this problem, the chatbot in this project can generate clear definitions of the topic, provide examples that are easy to understand. All these could help students or self learners to quickly understand the topic, answering questions and can provide correct answers in exams.

### **1.3.3 Unfamiliar with terms used in the syllabus**

As collected from the participants who participated in the survey, another issue while learning the software testing CTFL syllabus is understanding the terms used in the syllabus. For instance, there are some terms in the syllabus like

absence-of-defects fallacy, CLI testing, exit criteria and many more. There could be confusions when individuals start to learn the syllabus.

With the chatbot's help, accurate response can be generated to provide clear explanations and definitions on the confusing terms from users. This helps learners to understand the terms better and enhance understanding.

#### **1.4 Aim and Objectives**

This project carries certain objectives that need to be achieved, the objectives are as below:

1. To integrate a chatbot into the web application that may assist lecturers in teaching software testing CTFL syllabus.
2. To integrate a chatbot into the web application that can assist students in studying software testing CTFL syllabus.
3. To evaluate the effectiveness of the chatbot in teaching software testing using UAT.

#### **1.5 Scope and Limitation of the Study**

For this chatbot, certain type of users is mainly targeted. In this section, targeted users will be discussed. Furthermore, features to be included in the chatbot will be explained in this section as well.

The chatbot for teaching software testing CTFL syllabus will be only covering the foundation level's syllabus, for higher level certification syllabus will not be included into the chatbot.

##### **1.5.1 Target users**

###### **1.5.1.1 Students/ Software Testers who wish to take the CTFL exam**

Those students who are pursuing in software engineering course or other computer related courses will need to learn about software testing. As mentioned before, testing the software before deployment is important because testing can detect bugs that causes failure in the software. Quality of the software will be sabotaged if no testing is done before the deployment, cost of fixing the bugs after deploying the software will be high as well.

Students will face problem during studies, when lecturers are not free to provide feedback on the problem, students can make use of the chatbot to help. The chatbot can assist students by providing answers to the questions asked by the students regarding the syllabus of CTFL.

#### **1.5.1.2 Self-learners**

Not just software engineering students that will be studying software testing, there will also be interested people who will be learning the CTFL syllabus. As self-learners are not studying software testing in universities or colleges, one need to find the materials to study the syllabus Without any feedback from lecturers, one could not fully understand some topics in the syllabus as one are not from software course backgrounds.

The chatbot can assist in providing feedback for the problems, provide progress tracking and quizzes as well to test the understanding.

#### **1.5.1.3 Lecturers/ Trainer from the training academy**

Lecturers can make use of the chatbot to generate one quiz for the students before the class. In addition, chatbot can also assist to refresh memory when lecturers or trainers forgot about certain topics.

### **1.5.2 Modules**

#### **1.5.2.1 Account registration module**

This scope introduces the module that allows users to register an account for using this chatbot. Similar to ChatGPT, users are required to login using an existing Gmail account. This module can help to save every user's progress as one can check back the previous chat histories one have, users can also delete those unnecessary chats.

#### **1.5.2.2 Quiz module**

This module allows the chatbot to generate quiz based on different needs. Students and self-learners can use the chatbot to generate a quiz to test and track studies; by doing the quizzes the users can see which part excel in, which part

is needed to emphasize more during revisions, this helps to boost understanding in software testing.

#### **1.5.2.3 Star message module**

In some scenarios, students or self-learners will need to refer back to some specific response from the chatbot to do revision or recall back memories. To prevent from scrolling through vast chat histories, this module is introduced to star a specific message, students or self-learners can visit a page to view all the starred messages or go to the starred message. This module helps to save time from extensive searching.

#### **1.5.2.4 Chatbot module**

This project's web application will include a chatbot that will handle the prompts from the intended users. The chatbot will be using a Large Language Model (LLM) that will be fine-tuned to generate more accurate response that is based on software testing CTFL syllabus.

Cases where users asked questions that are not related to software testing CTFL syllabus, the chatbot will also handle the questions correctly by generating a message showing that chatbot are not able to provide a solid response for that question.

#### **1.5.2.5 Upload file module**

To assure that the chatbot can provide examples with the latest version of the CTFL syllabus, this module is introduced to allow lecturer role users to upload the syllabus file to the web application of this project.

Lecturer role users can choose to upload the latest version of the CTFL syllabus materials to the web application, then admin users can extract the file content and save into a database and act for RAG knowledge base.

### **1.6 Project Solution**

In order to tackle the problems mentioned above, a chatbot website will be developed. Most of the AI chatbots on the Internet right now is accessed through

websites. For instance, users need to visit the website to use ChatGPT. Hence, our users will be easier to access to the chatbot being developed as long one has internet connection and access to a browser. The interface will be user friendly and similar with other existing chatbots so that users can get a grasp of chatbot easily.

The chatbot will be able to provide concise answers based on users' inquiries, the inquiries are limited to questions related to software testing CTFL syllabus only as the chatbot's sole purpose is to teach software testing CTFL syllabus. With the help of chatbot, the users will not be afraid of the issue of not able to find the limited study materials on the Internet.

On top of that, the chatbot can also generate some quizzes related to software testing to test users' understanding, helping in tracking progress. By doing some practical quizzes related to software testing, users can apply knowledge on simulated practical scenarios, boosting understanding of the concepts.

To develop the chatbot web application, React, Laravel and MySQL will be utilized together. React and Laravel is mainly for the development of the website interface and accessibility of the chatbot. While MySQL will be the database used for the chatbot to store the chat history of each user and also the registered accounts' information.

The below image shows the overall review of the usage of A chatbot for teaching software testing CTFL syllabus web application.

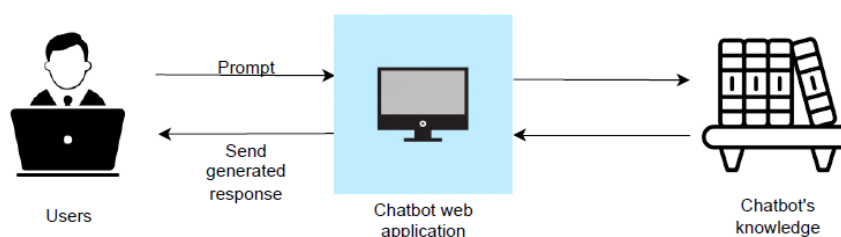




Figure 1.1: Overall review of using the chatbot

Moreover, the lecturers or trainers that uses this web application can also add or update the content of the chatbot. After the addition and updating of the knowledge base, the chatbot will first retrieve the knowledge from the new knowledge base first instead of pre-trained knowledge. This procedure for the chatbot is known as Retrieval-Augmented Generation (RAG). Below shows the new process of the chatbot utilizing RAG procedure.

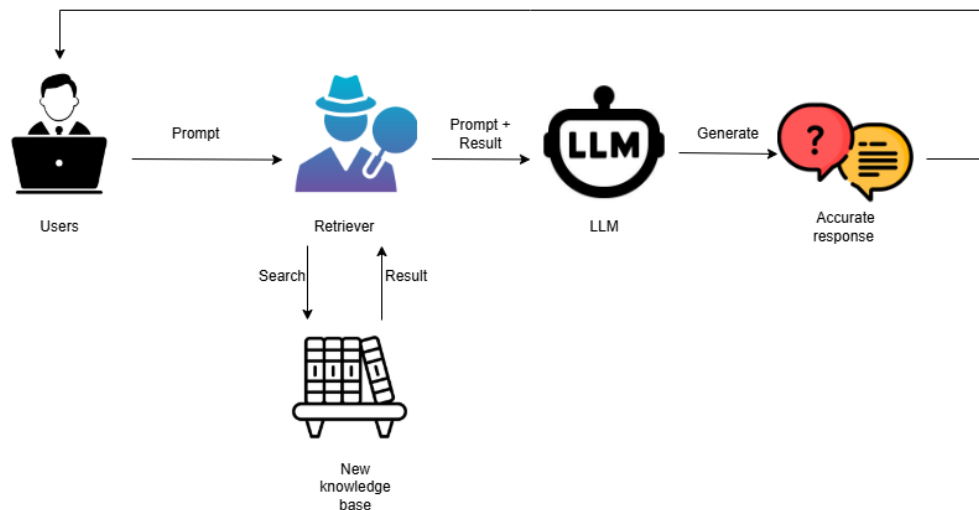


Figure 1.2: RAG procedure (Adopted from *Retrieval-Augmented Generation (RAG)* | Deepgram, n.d.)

## 1.7 Project Approach

V-Model methodology are applied to this project. The reason behind choosing V-Model as the methodology for the project is that this methodology is a testing focused methodology, for each phases in verification phase, there will be a testing phase in validation phases.

Testing helps to ensure the quality of the product, by implementing testing suitable to each development activity in the project helps to find and fix the defect early, preventing defects destroying the quality of the project. By doing so cost can be saved of the project since fixing bug after deployment will be very high compared to fixing the bug before deployment.

Moreover, the chatbot is designed to teach software testing CTFL syllabus, following a testing focused methodology enhances the core learning value. This makes the chatbot more reliable as went through many testing to ensure the chatbot's quality.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

In this modern era, chatbots are becoming a part of our life. AI was like a dream in the old days, where one does not have sufficient power and resources to fully build AI to aid people's daily life. With the advancement of technologies, one is given the environment to build a chatbot easily compared to the past.

AI Chatbots can be built to be utilized in different areas, one of the examples of chatbots is ChatGPT. Launched in the year 2022 by OpenAI, ChatGPT single handedly changed many lives in the world. Powerful and useful at the same time, ChatGPT can solve lots of questions asked by the users, no matter daily life issues, students' uncertainties in homework or assignments, help to generate a draft of email and many more.

This project focus on developing a chatbot that can be used in educations, which can provide benefits to the education sector. Below are some of the benefits if education organizations have a chatbot to assist in teaching students.

#### **2.2 Benefits and disadvantages of students using AI Chatbots for education**

##### **2.2.1 Benefits from students' perspective**

###### **2.2.1.1 Personal assistance for students**

Lecturers, teachers are not always free all the time to provide assistance to students which need help in homework or assignments. Students could be perplexed, uncertain about the homework, this put a halt to the progress in finishing homework. With the help of chatbots, students can ask the chatbot about the questions or problems faced and chatbot can provide solutions swiftly (Labadze et al., 2023).

###### **2.2.1.2 Enhancing students' skills**

Students can make use of chatbots to help strengthen personal skills such as writing, vocabulary, thinking. Chatbots are capable of providing students a vocabulary's synonyms, allowing students to learn more words that share similar meanings, which directly enhances the writing skills as students have learned many vocabularies. Whenever students face some issue, by looking at the response generated by the chatbot, students can learn new ways of thinking as the response could be something that the student never thought of. This encourages students to think from more perspective, aid in problem-solving skill (Labadze et al., 2023).

### **2.2.1.3 New method to learn new things**

Now there are a lot of chatbots in the Internet, students can learn many new knowledges by using all these chatbots. Chatbots can provide step by step learning methods, point out the mistakes done by students, provide examples on the study material, these all can ease students in learning new knowledges. For instance, an accounting student can make use of suitable chatbots to teach how to code in any coding language. Chatbots can provide explanations, examples for the student to learn, when the student encounter mistakes or problems, chatbots can also point out the mistakes and provide solutions to tackle the issue, helping to understand coding easily. Most importantly, students do not have to wait for lecturer's reply, chatbots can be accessed easily anytime.

With all these benefits, does that mean this project can conclude that chatbots are a great invention which does not bring any negative sides to humans? The answer is no, with the rising of chatbots, some unforeseen issues are also rising along. What could be the disadvantages of students using chatbots in educations?

## **2.2.2 Disadvantage of students using chatbots for education purpose**

### **2.2.2.1 Rely too much on chatbots**

Chatbots sure can aid students in studies, but if chatbots is being used in the wrong way, chatbots are there to destroy a student. Whenever students become lazy, students will just use chatbots to help to finish assignments, the work is done by chatbot solely and that student did not learn anything. If this scenario

keeps happening, the future students will not develop any skills or gain knowledge to be utilized when stepping into working environment. Whatever is done in the school life was all by chatbots, students learned zero knowledge which will harm the students (Labadze et al., 2023).

#### **2.2.2.2 Misleading information**

According to Labadze et al.(2023), in some circumstances, the response generated by chatbots are not from trusted sources or cited from reliable authors. This causes the response generated is not 100% factual, students who does not know might take the response from the chatbot as real incidents and learned wrong knowledges. Students will mislead by the wrong information and does not even realize the mistake.

### **2.3 Existing usage of chatbots in education sector**

There are cases where chatbots are implemented in order to ease the teaching in education sector, benefits were brought by the help of chatbot. This section will be studying some cases of chatbots in education.

#### **2.3.1 AI-based Chatbot for Practicing Responsive Teaching in Mathematics**

As stated by Lee and Yeo (2022), a chatbot is developed to train preservice teachers (PST). Teachers are busy with tons of work on hands, and hardly have enough time to attend any form of training. To tackle this issue, the chatbot is developed to practice the responsive teaching skills especially in the questioning strategies for mathematics.

Designed using design-based research along with two iterations of chatbot development, implementation, and evaluation, the chatbot will act as a virtual student, throwing questions to PSTs, simulating scenarios in classrooms or lectures. This aids PSTs to learn how to handle the questions from students and will help PSTs to develop better ways to answer students' questions.

From this study not just students will be benefiting from the help of chatbots, even PSTs can be benefited. With the help of chatbots everyone can

get to learn more knowledge which are intended to learn, anytime and anywhere as long as have an Internet connection and access to the chatbots.

### **2.3.2 Chatbot on Students' Learning in Ghanaian Higher Education**

According to Essel et al. (2022), chatbot are assessed to observe the effectiveness of assisting students in studying. The chatbot also known as KNUSTbot, developed without using any code and was integrated into WhatsApp.

Used by students from multimedia programming course, the usage during nighttime is significantly higher than daytime. This shows that chatbots provide more flexibility in terms of learning, after school lecturers might not have time to reply to students' question immediately, as such students utilize chatbots to help in studies. Chatbots could come in handy when the school has high student-teacher ratios.

Observation shows that students that used chatbot scored higher in tests, which is a good sign where chatbot is proven that the chatbot does help in studies and enhances students' understanding. Despite having some issues in the chatbot, the chatbot overall provided benefits to students.

## **2.4 Comparison of AI models on the Internet**

As of 2025, there are already many AI models released, this section will be the discussion and comparison of some famous AI models at the moment. The AI models being discussed includes Gemini, ChatGPT, DeepSeek and lastly Grok. Each of these AI models excels in different areas and serve different purposes.

### **2.4.1 Gemini**

Developed by Google DeepMind, Gemini is a multimodal model that possesses the ability to process images, code, text which significantly assist in researching and productivity. Gemini is a highly flexible AI model as Gemini is a multimodal model; Gemini can process many things concurrently (Souza & Li, 2025).

Nowadays, when using Google to search for an information, there is a new feature of Gemini summarizing the content that was intend to search for, by helping to do the searching. By searching through multiple websites and gathering the information needed, Gemini will generate a summary of the content at the top of the search result page in Google.

This feature is friendly as people does not need to surf too many websites to find the information needed, Gemini will help users to search and summarize a general overview of the content, which saves users a lot of time.

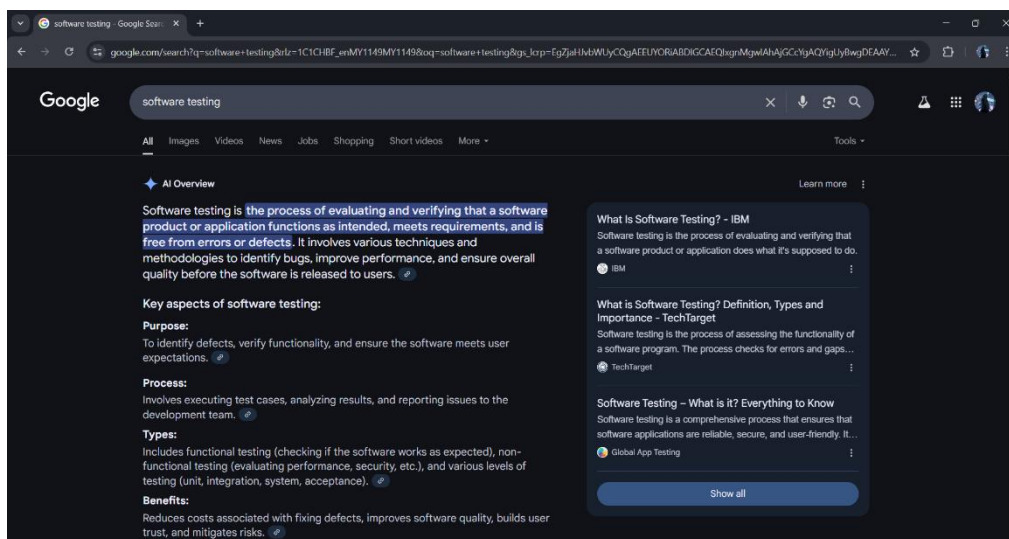


Figure 2.1: Gemini summarizing software testing

One slight disadvantage of this feature is that Gemini could cite those websites without checking the correctness of the information, this causes the user to observe a wrong summary generated by Gemini, causing misleading and misinterpretation. This is a serious issue as the chatbot may cause people to misunderstand the concept of certain topics, as one got the result from the generated content by Gemini.

On top of that, according to Souza and Li (2025), a downside of a multimodal model is that the cost of computational will rise. The complexity of the computation could affect the performance of those devices that does not come with high processing ability. In addition, due to the integration of Gemini in Google, the issue related to personal privacies need to be considered as well.

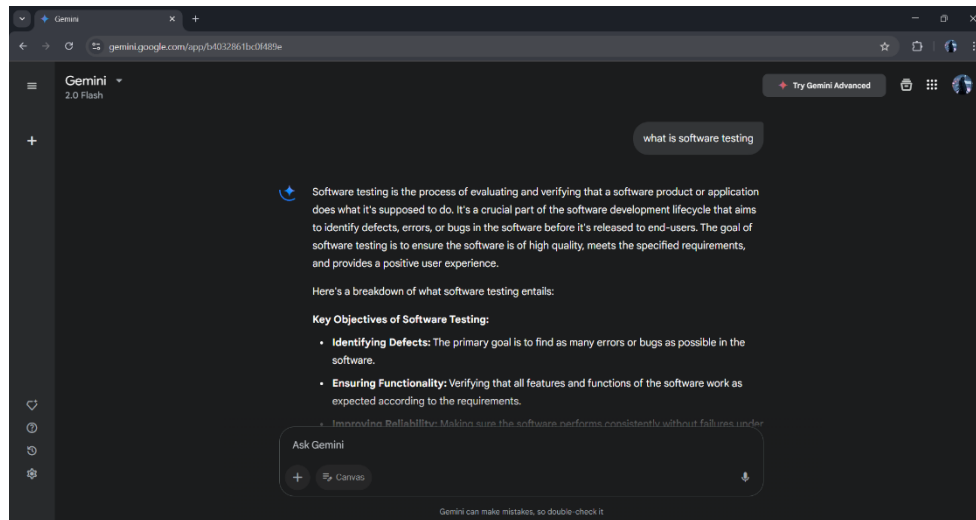


Figure 2.2: Sample conversation of Gemini

### 2.4.2 ChatGPT

Being a well-known and widely used AI model since early 2023, ChatGPT is a work from OpenAI. ChatGPT is a general-purpose AI that can help everyone in the whole world, including teaching, automation, customer support and many more. ChatGPT can be used in any field, and can provide outstanding performance (Souza & Li, 2025).

ChatGPT is a very powerful AI model as in the early days there are not many competitors from other AI models, the performance is impeccable, able to solve almost every problem asked by user. Students can utilize ChatGPT to aid in studies, using ChatGPT as a tool to assist in assignments, company employees can also use ChatGPT to generate report.

As stated by Souza and Li (2025), one obvious disadvantage regarding ChatGPT underlies in the response provided, the generated response in ChatGPT upon inquiring could be biased or one-sided. As a result, the response generated would not be hundred percent accurate which could greatly affect the work or studies of people around the world. Wrongly generated response would cause spreading wrong information to intended users, harming users as the wrong knowledge is learned.



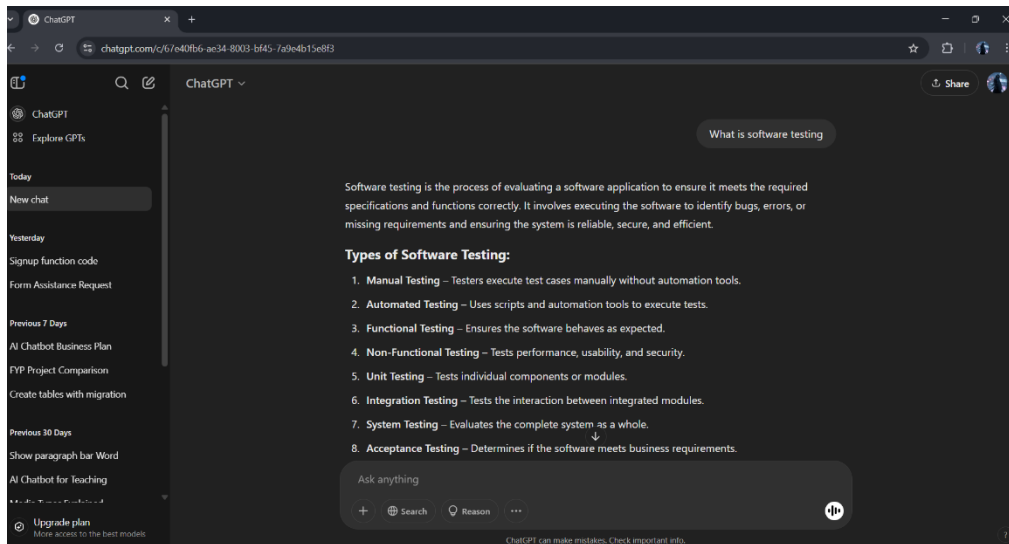


Figure 2.3: Sample conversation of ChatGPT

### 2.4.3 DeepSeek

According to Souza and Li (2025), DeepSeek is one of the AI models that shook the world, claiming that less resources were used to train compared to ChatGPT. Developed by DeepSeek AI, DeepSeek is released in year 2023, but made a huge appearance in January 2025. After claiming that it utilized less resources to train, DeepSeek even caused NVIDIA's stock price to decrease. Specialized in retrieving information, DeepSeek excels in providing verified responses rather than biased responses, which is a greater tool when comes to researching and studying compared to using ChatGPT.

Deepseek may provide valuable and accurate information for users, however the downside is that Deepseek lacks in the ability to generate innovative and relaxed interactions as DeepSeek has a limited content generation capacity. This may cause dissatisfaction in using Deepseek as Deepseek is not as interactive compared to the other AI models available on the Internet.

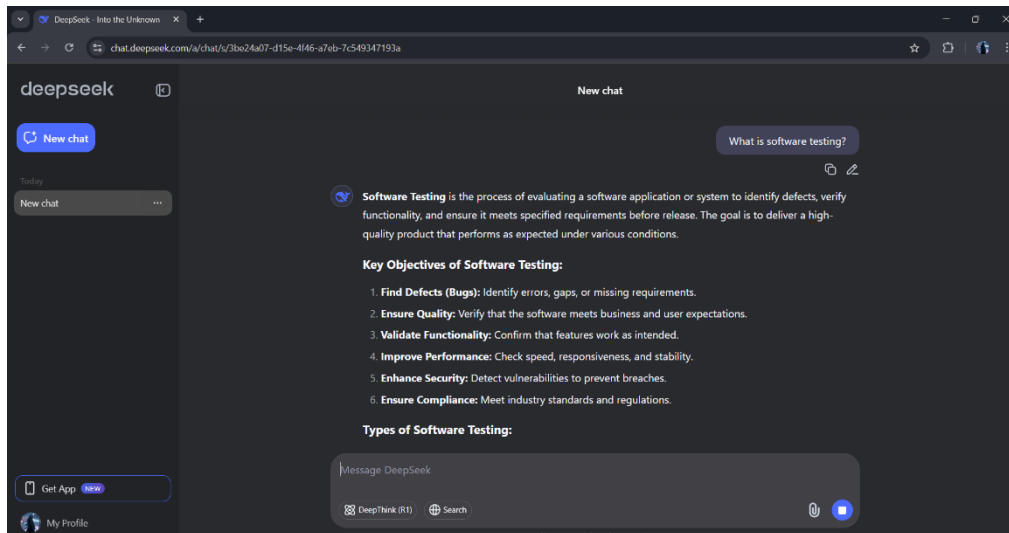


Figure 2.4: Sample conversation of DeepSeek

#### 2.4.4 Grok

Founded by xAI, Grok is seamlessly integrated with X, formerly known as Twitter. Grok becomes the most powerful AI models according to the sharing of Elon Musk during launching of Grok, the online stream compared each AI models above to test the AI's performance, Grok scored higher compared to the other AIs. With the strong integration with X, the generated response is based on real-time discussions, trends can also be analyzed. On top of that, Grok comes with the "Think Mode", which the AI can solve the problem faced better if Grok is provided time to think of the most optimal solution.

Similar to ChatGPT, as Grok's integration with X, the response from Grok could be biased as well because Grok searches in X platform for some information. Other than that, Grok has less integration ability compared to ChatGPT which can be integrated into more services (Souza & Li, 2025).

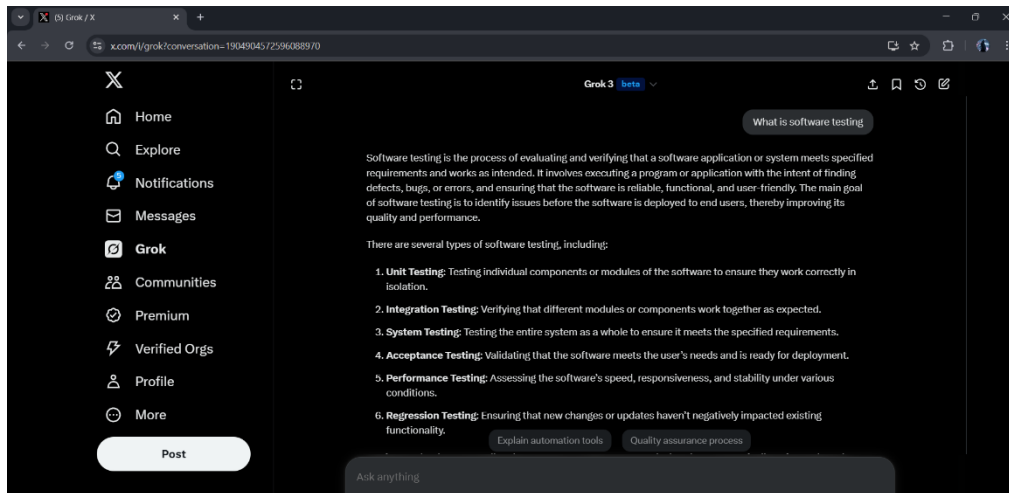


Figure 2.5: Sample conversation of Grok

### 2.4.5 Comparison between each AI Models

Table 2.1: Comparison table between each AI Models

AI Model	Developer	Strengths	Weaknesses
Gemini	Google DeepMind	<ul style="list-style-type: none"> <li>- Multimodal (processes text, image, code)</li> <li>- Summarizes search content from multiple websites</li> <li>- Time-saving and efficient for users</li> </ul>	<ul style="list-style-type: none"> <li>- May generate misleading or inaccurate summaries</li> <li>- High computational cost</li> <li>- Potential privacy concerns due to integration with Google</li> </ul>
ChatGPT	OpenAI	<ul style="list-style-type: none"> <li>- General-purpose AI, widely used</li> <li>- Versatile across fields (education, support, automation)</li> <li>- Strong early</li> </ul>	<ul style="list-style-type: none"> <li>- Responses may be biased or one-sided</li> <li>- Can spread misinformation if relied on blindly</li> </ul>

		performance with fewer competitors	
DeepSeek	DeepSeek AI	<ul style="list-style-type: none"> <li>- Efficient training with fewer resources than ChatGPT</li> <li>- Provides verified, unbiased responses</li> <li>- Strong tool for research and study</li> </ul>	<ul style="list-style-type: none"> <li>- Limited content generation capabilities</li> <li>- Less interactive and engaging than other AIs</li> </ul>
Grok	xAI (Elon Musk)	<ul style="list-style-type: none"> <li>- Real-time trend analysis from X (Twitter)</li> <li>- “Think Mode” for better problem-solving</li> <li>- Scored highest in performance comparisons</li> </ul>	<ul style="list-style-type: none"> <li>- Responses may be biased due to X-based data</li> <li>- Less integration capability than ChatGPT</li> </ul>

## 2.5 Development Methodologies

There are multiple types of software development methodologies, each methodology act as the skeleton for structuring, planning, managing the development process of a software. Methodology helps to maintain the specifications, cost, risk of the software in the deployment stage, aiding companies to produce well-developed software. Every methodology serve the same purpose but with different structures, each differs in steps, activities, communication ways, outputs. Companies shall take deep consideration and discuss and select the best methodology for developing the software (*Software Development Methodologies: Types and Comparison*, n.d.).

In this section, different types of methodologies will be discussed, advantages and disadvantages to select the best methodology for developing this chatbot project.

### **2.5.1 Agile**

One of the software development methodologies which integrates a certain degree of pragmatism to the final product's deployment and foresee the requirement for flexibility. By actively delivering small and feasible addition to the project, the development team can adapt to the changes of requirements from client side (*Software Development Methodologies: Types and Comparison*, n.d.-b).

Agile methodology can bring one advantage to the software company which is flexibility and adaptability. Agile is versatile and resilient, this leads to making changes in terms of primary considerations, user requirements are not difficult, the developer team can change the method of developing while focusing on delivering the deliverables. Moreover, agile are proved to improve a software's reliability and quality. Agile focuses on assuring the quality of the software and will not ignore testing. With testing being done, the delivered software will have ensured quality, risk of bugs occurring which led to defect will be reduced, user's satisfaction will be satisfied. On top of that, agile also can amplify satisfaction of clients. As agile emphasizes on delivering high quality deliverables, clients are always engaged in the process of developing the software, the software delivered will satisfy the clients.

On the other hand, what could possibly be the disadvantages of Agile? One of the biggest disadvantages is scope control is heavily restricted, Agile focuses on being malleable which led to a lot of changes in project scope. If no proper scope control is done, scope creep will occur and bring harm to the development of the project. Other than that, lacking in staleness is also a disadvantage for Agile. This is because Agile often depends on clients' comment on the delivered deliverable, making the project team to predict the cost, project schedule and result. Agile methodology can also lead to the

developer team to be very exhausted, as the developer has many target dates to meet and not enough time. Team members could feel ill due to lack of rest, causing the project to be delayed more.

There are six processes in the Agile software development methodology, which the first process is gathering the user's requirements. In this process, meeting, online/physical survey or any other requirement gathering methods will be conducted to obtain the requirements of users towards the software.

The second process is planning. The gathered requirements will be discussed by the development team, the team lead will prepare a plan for the team members to act as a guide to develop the software. The plan should include the characteristics of the software which needed to be provided during the development stage. Right after the planning process, development process will be carried out. Developers can develop the software according to the plan prepared by the team lead, assuring required features is included in the software.

Testing process is carried out after the developers has developed the software. Unit testing, integration testing, system testing, user acceptance testing and many other types of tests should be carried out during the testing process. However, testers shall find all the bugs before the user acceptance test for the developers to fix the bugs. By constantly testing the software, a high-quality software can be produced where the software does not sabotage the usage of users.

After the testing process, the software will go through deployment process where the software will be officially published, users can start to make use of the software. Last process of the agile software development methodology is maintenance. Any bugs occurred and faced by users will be fixed by the developer team in order to maintain the quality of the software.



Figure 2.6: Agile software development processes (adopted from GeeksForGeeks, 2024b)

Nowadays, Agile software development is being used a lot by many companies as Agile is suitable for many scenarios including software development process became very fast paced and the user requirements kept on changing. Agile do no prefer to finish developing all the software once and for all, instead Agile prefers in a progressive way to develop software step-by-step from the beginning of the project (GeeksforGeeks, 2024b).

### 2.5.2 V-Model

The name of V-Model comes from the shape, the left of the V contains the verification phase whereas the right side represents the validation phase. In V model methodology, the software development life cycle is divided into certain phases, each phases are correlated with testing phases. V-Model can be used in the situations where the user requirements and knowledge of the usability of the software is clearly defined since the start, scope of the project is transparent, and development team is conscious of the requirements of the software. With these conditions, V-Model methodology can provide a great help to deliver a software with high quality.

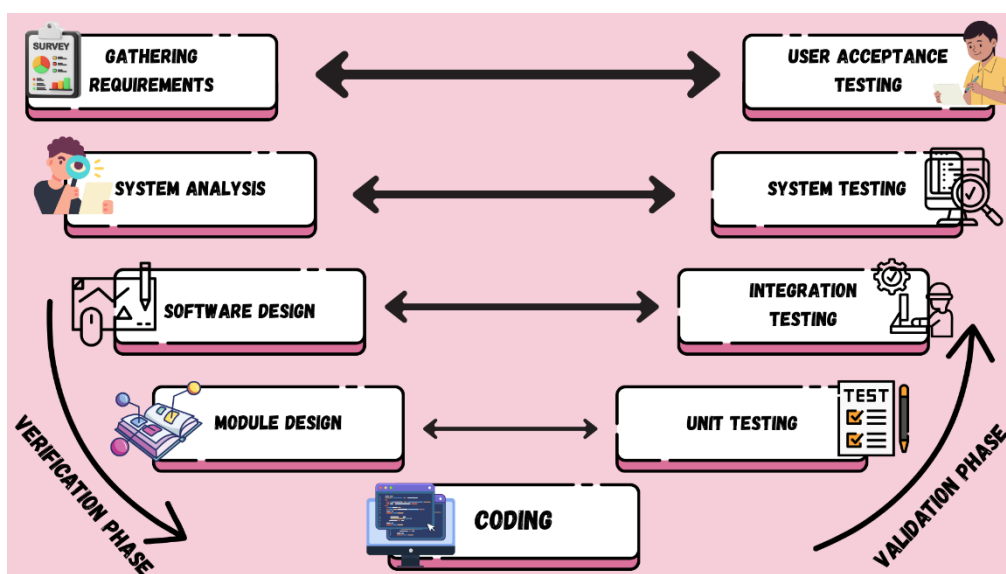


Figure 2.7: V-Model methodology (Adopted from Oppermann, 2023d)

The obvious advantage of utilizing V-Model methodology is that V-Model helps to minimize the probability of risk occurring. How so? V-Model lay out the complete guideline for the process for the development of the software, risk management and risk mitigation can be done better and easier. Moreover, efficiency of developing the software can be escalated. V-Model emphasizes on cooperation of distinct team and stakeholders, the process of developing and testing can be improved. In addition, V-Model enhances the communication of the developer team and clients as V-Model prefers communications is done between the stakeholders of the project, to make sure that developers have understood the requirements and goal of the project.

Being very inflexible is a disadvantage for V-Model methodology, there is no sufficient space for substitute or divergence for the project. Any new requirements or feedback from clients could be troublesome as changing the requirements of the project is difficult. Another disadvantage of V-Model is time-consuming, in every stage of V-Model, preparations and documentation must be done, which could be the possible reason of delay in developing the project and causes project schedule to be extended. If project schedule got extended, the cost of the project will also increase. Other than that, V-Model need rigorous resources which time, cost, employees are included. V-Model is not recommended for those small companies or low number of members in



developer team, as developers could face issue in allocating sufficient resources for the project (Oppermann, 2023c).

V-Model consists of two phases which are verification and validation phases. Firstly, in the verification phases, no coding will be done, instead is the step of assessment of the phase of developing product to check if the requirements are satisfied and checking if the process is being followed. First phase of the verification phases is gathering of requirements, what is the expected outcome that user required. Meeting will be conducted in order to obtain requirements which cannot be ignored as this process is vital for the development of the software. When the clients do not know what is really wanted, developer team can prepare acceptance test design, which can be also used during acceptance testing. Following that, analysis of system will be conducted. By understanding the requirements gathered, the developer team shall make a blueprint of the software which can help in designing test cases as well.

The next process is designing the software, in this stage developer team shall have clear understanding of the internal function and external system. Testers can also start to prepare integration testing in this stage. Lastly, the design of module is carried out. The design or other modules should have enough affinity, unit tests can be designed as well since unit tests help to recognize and eliminate the errors in prior. Coding phase will be conducted after this, developer team will start writing the code of the software, the written code should always be following the principles and checked during code reviews before the compiled code is confirmed as the final.

After verification phases, validation phases will be carried out. Validation phases is responsible to find bugs in the software based on the deliverables, checking the quality of the software from client's perspective. Several testing will be done to ensure that the software has good quality and is able to satisfy the requirements of the clients. As mentioned above, unit testing is performed during the phase of designing module, unit test plans are prepared and have the purpose of removing the bugs in the code or level in unit, gain

confidence, reduce risk in the pieces of software under test prior to integration of system. Next, integration testing will be conducted after unit tests are concluded, done in the software design phase, integrated modules and the software will be tested, to inspect the transmission between the modules. Integration testing also reduces risk in the relationships. Interface between pairs and groups in the system under test as the pieces come together. Following up, system testing is done to reduce risk in the overall, particular behaviours, responses, functions in the system under test as a whole. Lastly in the validation phase, user acceptance testing is conducted. Demonstrations are done to show the clients that the software can satisfy potential users and is ready to be deployed, in this phase finding bugs in the software is excluded.

V-Model is preferred to use by large size firms as V-Model is effortless to control because the model is inflexible, each phases have own output and evaluation. Moreover, the quality of the deployed software is guaranteed as there are many tests being performed, bugs that leads to defects can be eliminated at early time (GeeksforGeeks, 2024a).

### **2.5.3 Lean development**

One of the vital segments of Agile software development methodology, Lean software development methodology focuses on reducing scrap and improving the effectiveness of the process of developing software. There are seven main principles of Lean software development methodology, all principles will be discussed below.

The first principle of Lean software development methodology is abolishing wastes, the team lead shall do code reviews and identify the excess coding or attributes, civic processes, then develop a strategy to remove these unnecessary items as Lean does not provide quality to the software developed. Second principle is integrating quality; by using some methods like pair programming or test-driven development, the quality of the developed software can be assured. Other than that, the third principle is magnify training. As mentioned before, code review is done to identify and locate which part is unnecessary, in this process every member in the developer team can learn,

where the other development team member did wrong and how to avoid repeating the same mistake.

The fourth principle of Lean software development methodology is postponing obligation until last moment, the functions and attribute should be included at the last moment available to prevent the risk to reconstruct the software, study and test more before any worst-case scenarios happen. Next, the fifth principle is swiftness in delivering. The developer team can obtain the comments from users as soon the software is delivered; with the comments the team can determine which part of the software have rooms for improvement and come out with better functionality and quality. One of the most important principles in Lean is the sixth principle, respect every individual. As the base of providing a fruitful and communal room during the development of the software, any disagreement is fine as long as both parties hopes for the best for the project. Lastly, the last principle is maximizing the entire project. The development process is analysed by the whole developer team, allowing the value of Lean flows efficiently throughout the project (Lutkevich & Silverthorne, 2021).

For the first process of Lean methodology, is examining the requirements. Similar to many other software development methodologies, by obtaining the requirements from users, developers are able to produce the functions that satisfy the requirements. Next, plan the value chain, developer team shall conduct meetings to examine the process that does not provide quality throughout the development process and remove entirely, this helps to ensure that each steps provide absolute quality for the development. On top of that, design stream will be the next step. This step aids in providing a seamless workflow that avoids the risk of delaying the delivery of software.

After that, initiate demand-driven system is the fourth step. The developer team shall design the functions recommended users, and can obtain the recommendations after each iteration. The upcoming step is consistent in refining, code reviews should be conducted consistently in order to let the members of developer team to understand which mistake the developer team should avoid and how improvements can be made. The sixth step is providing

quality, Lean methodology promote developers to utilize Test-Driven Development (TDD) to make sure that quality is merged in the whole development process. Lastly, permit the teams, this step is crucial as the whole developer team can access tools needed to develop the software.

Aiming in delivering quality assured software which functions well and efficient, Lean possesses the ability to magnify the process of developing the software (GeeksforGeeks, 2024c).

#### 2.5.4 Comparison of Development Methodologies

Table 2.2: Comparison Table of Each Software Development Methodologies.

Methodology Aspects	Agile	V-Model	Lean
Description	Integrates a certain degree of pragmatism to the final product's deployment and foresee the requirement for flexibility.	Named by the model shape V, left of the V contains the verification phase whereas the right side represents the validation phase. SDLC is divided into certain phases, each of the phases are correlated with testing phases.	One of the important steps in Agile. Focuses on reducing scrap and improving the effectiveness of the process of developing software.
Simplicity	Moderate	Low	High

Specification of requirement	Flexible	Fixed	Moderate
Flexibility	Highly flexible	Low	Moderate
Requirement of documentation	Low	High	Low
Involvement of user	High involvement	Low involvement	Moderate involvement
Time frame	Moderate	Long	Short
Cost of development	Vary based on expertise	High	Moderate

## 2.6 Web Application Framework

The chatbot for teaching software testing CTFL syllabus will be web-based application, using certain web application frameworks aids in the development of the project as the framework already contained authentication of users, handling HTTP requests, structure and components, libraries and many more.

### 2.6.1 Front-end Web Application Frameworks

#### 2.6.1.1 React

Developed by Meta (formerly known as Facebook), React is an open-sourced front-end JavaScript library that allows developers to develop responsive user interfaces, especially in rendering the web application's elements.

One of the advantages of utilizing React to develop the front-end for web applications is that this project can create responsive user interfaces by using Virtual Document Object Model (DOM). When any state of the object changes, React will update the state but not the whole DOM. Other than that, React is an active community that keep on produces new libraries and support which aid developers to search for a solution for the problems rapidly.

React has some difficult concepts like JSX, component lifecycle, these could cause perplexity to new learners and beginners as those are not some concepts that can be understood swiftly. Moreover, React is moving in a fast pace, when beginners wish to learn React by studying documentations, one might end up studying the old documents which were outdated, this highly halt the progress in learning React.

#### **2.6.1.2 Angular**

Another open-sourced front-end framework, Angular is a TypeScript-based web application framework that is produced by Google. Angular is mainly used for developing a dynamic, complicated web applications. Angular follows component-based architecture like React and provides gadgets for building routes, forms, communication of HTTP.

Angular provides two-way data binding, this helps in maintaining the changes in the UI are also returned in the project's model and vice versa. Benefit of this is that the speed of development is increased, and chances of code overlapping is lowered. In addition, as a TypeScript-based web application framework, Angular can help with debugging, autocompletion which ensures elegant development process.

As being a Type-Script-based framework, there are also disadvantages. Angular's features could possibly be troublesome for beginners, all those features require a huge amount of time to learn and eventually master the framework, which affects the learning progress. Next, there are less flexibility while utilizing Angular as the framework follows a stern design, this puts a

limitation to developers as this could cause some problems while integrating with other third-party tools.

### 2.6.1.3 Vue.js

Vue.js is also a open-sourced JavaScript web application framework that can develop dynamic web applications by focusing on view layer, this helps developers to integrate external libraries into every web applications. Vue.js is designed and released by Evan You in the year 2014.

Vue.js is known for simplicity, this is a benefit for new learners, as Vue.js aids beginners that wanted to start developing web applications. Vue.js follows a single-file component structure, which means that Vue.js combines the HTML, JS, CSS together to trim an efficient project development and debugging of project. Moreover, Vue.js can perform better thanks to the light size of 20KB. This speeds up the time of loading the web application and lead to faster web application development.

The main disadvantage of using Vue.js is that Vue.js has small environment, not like React. There is not much documentation for beginners to refer to and Vue.js has less plugins which could be a trouble when developing some modern designs. Secondly, if developers want to develop a large project, Vue.js is not a recommended framework as the community of Vue.js is not as big compared to React or Angular, finding solutions or support could become a serious issue while developing the web application.

### 2.6.1.4 Comparison Table

Table 2.3: Comparison Table of Each Front-end Web Application Frameworks.

Framework	Developer	Advantages	Disadvantages
React	Meta (Facebook)	- Responsive UIs via Virtual DOM - Active community with	- Complex concepts like JSX and lifecycle - Fast-paced

		support and libraries	updates may result in outdated learning materials
Angular	Google	<ul style="list-style-type: none"> <li>- Two-way data binding increases speed and reduces overlap</li> <li>- TypeScript aids debugging and autocompletion</li> </ul>	<ul style="list-style-type: none"> <li>- Steep learning curve for beginners</li> <li>- Less flexibility due to strict design patterns</li> </ul>
Vue.js	Evan You (2014)	<ul style="list-style-type: none"> <li>- Simple for beginners</li> <li>- Single-file component structure aids development and debugging</li> <li>- Lightweight (20KB) enables faster load times</li> </ul>	<ul style="list-style-type: none"> <li>- Smaller community and fewer plugins</li> <li>- Less suited for large projects due to limited support and documentation</li> </ul>

## 2.6.2 Back-end Web Application Frameworks

### 2.6.2.1 Laravel

An open source, free to use PHP web application framework, Laravel is meant to make the development of web application to be done efficiently, faster and simpler development. Laravel utilizes the Model-View-Controller (MVC) architectural design. Model, View, Controller components streamline the flow of data and also the logic, because controller is responsible for requesting the data from the model component and also helps to render the views.

The advantage of utilizing Laravel to build a web application is Laravel has a powerful built-in security features like cross-site request forgery (CSRF), the security measure in Laravel makes sure the data of user is being protected at all times. In addition, Laravel is a friendly framework for web developers as



Laravel included complete documentation in the website that can allow new learners to understand how to use Laravel swiftly.

Looking onto the downside of using Laravel, Laravel has constant updates on the versions. Problems might occur while upgrading the version of Laravel as the Laravel development team might make root changes in the new update which causes the web application to fail for a while. Other than that, Laravel could be easy to learn but hard to master, this is because there are many new concepts like MVC, commands, blade templating and many more, new learners might have a hard time to fully learn the full usage of Laravel.

#### **2.6.2.2 Django**

A powerful Python web application framework, Django aims in motivating the speed of web application development and practical designs. Django is developed by the seniors of the web development field, an open-source web application framework that can ease web developers on building dynamic web applications where the problems faced during development is taken care by Django.

One of the advantages of using Django is Django is highly adaptable, Django is utilized by many sites that will have many visitors every day, as Django has the power to scale with high speed. On top of that, Django uses Python language which is friendly for any beginners to start coding with, Django is the perfect tool for anyone who wants to start developing web application.

Although Django can provide some advantages for web developers, there are also downsides of Django. Firstly, Django could cause perplexity for learners. Django might not be suitable for easy web projects as new learners is not familiar with Django's concept yet, this is because Django is high-level framework which need developers to be familiar with the structure. Next, Django has a inflexible structure for developing the web project, also known as "The Django Way". This could potentially stress the web developers as developers have less control over the coding.

### 2.6.2.3 Express.js

One malleable back-end web application framework for Node.js, Express.js contains features that is required for building a web application or mobile application. Express.js is also famous for simplicity and flexibility, making this framework a good choice for web developers.

Express.js brings the benefits of allowing developers to rapidly shift from constructing plan for web application to releasing. Express.js has a simple architecture which can help in fast development cycles. Moreover, Express.js excels in asynchronous operations to maintain the high number of concurrent requests, a perfect solution for small projects but with huge amount of inbound traffics.

The downside of using Express.js for web application development is that there are enough features and structure for development. There are only needed items like middleware, routing, many other required features are not included in Express.js which causes a discomfort while developing web applications. Moreover, the structure of Express.js is simple which the risk of code inconsistencies could occur in a bigger web application development project.

### 2.6.2.4 Comparison table between Back-end Web Application Frameworks

Table 2.4: Comparison Table of Each Back-end Web Application Frameworks.

Frameworks	Advantages	Disadvantages
Laravel	<ul style="list-style-type: none"> <li>- Powerful built-in security features (e.g., CSRF protection)</li> <li>- Complete documentation for easier learning</li> </ul>	<ul style="list-style-type: none"> <li>- Frequent updates may cause compatibility issues</li> <li>- Easy to learn but hard to master due to many new concepts (e.g., Blade, MVC, commands)</li> </ul>

	<ul style="list-style-type: none"> <li>- MVC architecture streamlines logic and data flow</li> </ul>	
Django	<ul style="list-style-type: none"> <li>- Highly adaptable and scalable for high-traffic sites</li> <li>- Uses Python, beginner-friendly language</li> <li>- Handles common development problems efficiently</li> </ul>	<ul style="list-style-type: none"> <li>- Can be confusing for beginners</li> <li>- Not suitable for simple projects</li> <li>- Has an inflexible structure ("The Django Way") limiting developer freedom</li> </ul>
Express.js	<ul style="list-style-type: none"> <li>- Simple and flexible architecture</li> <li>- Enables rapid development and deployment</li> <li>- Excellent support for asynchronous operations, good for high concurrent traffic</li> </ul>	<ul style="list-style-type: none"> <li>- Lacks built-in features and structure</li> <li>- Not suitable for large-scale applications due to potential code inconsistencies</li> </ul>

## 2.7 Large Language Models (LLMs)

Large language models (LLMs) is a type of foundation models that are trained with huge amount of data, this creates the ability of LLMs to understand the content of the task, then produces response in natural language. LLMs are useful for doing lots of works by fine-tuning the model accordingly.

In this section, three similar LLMs will be compared and studied, in order to select the best model for the chatbot for teaching software testing CTFL syllabus.

### 2.7.1 Gemma 2 2B

Gemma 2 2B is one of the model developed by Google DeepMind, this model possesses 2 billion parameters as stated in the name, 2B. Although having a small parameter size, Gemma 2 2B does not loses in terms of performance. In

the Chatbot Arena, Gemma 2 2B outmatches other models like GPT-3.5, Mixtral 8x7B. This indicates that Gemma 2 2B has better validity even having a small parameters compared to other models.

As stated above, Gemma 2 2B's performance and efficiency is the advantage. Even though only consists of small parameters of 2 billion only, quality is not lost and perform better than other models. Marking a new milestone for performance of AIs. Next, due to small parameters, Gemma 2 2B can be run on tons of devices and cloud servers. Gemma 2 2B is basically accessible by anyone, helps to save costs due to the small size, people does not need to upgrade PC just to run the model.

On the other hand, Gemma 2 2B has a smaller support in language compared to other models. Like LLaMA 3.2 3B, Gemma 2 2B loses in language support as Gemma 2 2B does not have advanced and wide language support. Also, due to the small size, Gemma 2 2B has a small context window. When it comes to long paragraphs, chats with users, this model could have issues in performance (Kamath, 2024).

### **2.7.2 Phi-3.5-mini**

Developed by Microsoft, Phi-3.5-mini is meant to be cost-saving, and possess outstanding ability to perform reasoning, do coding, language and mathematics. With the realese of Phi-3.5-mini, the choice of choosing Phi-models from Microsoft also become larger, people have more choice to select the suitable models for developing own projects.

Phi-3.5-mini is a efficient model, which is a advantage. Phi-3.5-mini is designed for users with limited resources such as limited memories. With the help of Phi-3.5-mini, intended users can deploy the model without having to concern about the performance. On top of that, Phi-3.5-mini consist of context length support of 128K, this outperforms the other models like Gemma-2 family. With this context length support, Phi-3.5-mini can perform tasks like summarization, retrieve useful information excellently (AdinaTru, 2024).

Although possessing such advantages, Phi-3.5-mini is found to be less better compared to other models in some specific tasks. For instance, Phi-3.5-mini is less efficient in prompt editing assignments compared to LLaMA 3.2 3B or Gemma 2 2B. Moreover, Phi-3.5-mini is less secured. Phi-3.5-mini does not have the enhanced safety features like the other LLMs, making this model less secured and privacy of data is not secured at all times (Bonnet, 2024).

### **2.7.3 LLaMA 3.2 3B**

According to Bonnet (2024), along with LLaMA 3.2 1B, LLaMA 3.2 3B is one of the lightweight LLMs in the Llama 3.2 family. Developed by Meta (formerly Facebook), these LLMs are designed for efficiency and strong executions even the LLMs are being restrained. The lightweight LLMs in the Llama 3.2 family contains capabilities such as mutlilanguage supported, ways of training, and privacy for the device.

LLaMA 3.2 with 3B parameter can generate different responses in different languages, the supported language includes English, French, Italian, German, Spanish, Hindi, Portuguese, and Thai. Using these languages can also work with LLaMA 3.2 LLMs, making language barrier disappear as a problem for using the LLMs.

Moreover, LLaMA 3.2 can run on any local devices. This feature is special compared to other types of LLMs as LLaMA 3.2 can process the queries from users, and generate suitable responses back to the user at instant. There could be some data that are sensitive, but LLaMA 3.2 can store the data directly on the local device, while reducing the usage of cloud. This is a great feature as the privacy of the user's data are being secured.

Lastly, LLaMA 3.2 can achieve more advantages by using more enhanced training techniques, the techniques includes refining knowledge, pruning. Refining knowledge technique aids the small LLMs to learn from bigger models, by learning from bigger models, the small LLMs like LLaMA 3.2 with 3B parameter can generate more accurate responses based on user's

queries. Next, pruning ensures that the performance of the models get to be remained while the size of the model is being reduced.

One advantage of using LLaMA 3.2 3B is also mentioned above, which is multilanguage ability, supporting eight different languages, even if the user does not know English well, LLaMA 3.2 3B users can also communicate with LLaMA using the other supported languages. The support for multilingual is also enhanced for mobile devices and edge, the performance of LLaMA will still remain rapid and privacy secured. In addition, LLaMA performs extremely well in terms of following instructions and summarization. Compared to other LLMs with similar parameters like Phi-3.5-mini and Gemma, LLaMA 3.2's performance excels those LLMs.

Looking on the downside of LLaMA 3.2, as stated by Kahani (2025), LLaMA 3.2 struggled with precision in calculating. In one of his calculation task appointed to LLaMA 3.2, there are error in handling the calculation with different segments, indicating that LLaMA 3.2 is limited with calculation ability when comes to complexed questions. Secondly, LLaMA 3.2 could possibly generate unfinished responses. LLaMA 3.2 did not provide a complete response while handling a task related to comparing, the reason behind that is this model faces a tool error. LLaMA 3.2 might have the issue of delivering incomplete results back to users if faces unforeseen problems while processing the input from users.

#### 2.7.4 Comparison Table between LLMs

Below is the table of comparing each LLM's advantages and disadvantages that were discussed above.

Table 2.5: Comparison Table of Each Large Language Models.

Model	Developed by	Advantages	Disadvantages
Gemma 2 2B	Google DeepMind	- High performance despite small size - Outperforms models like GPT-	- Limited language support - Small context

		3.5, Mixtral 8x7B - Efficient and cost-saving - Can run on many devices and cloud servers	window, struggles with long input
Phi-3.5-mini	Microsoft	- Efficient for users with limited resources - Context length support of 128K - Performs well in reasoning, coding, and summarization	- Less effective in prompt editing tasks - Weaker security and privacy features
LLaMA 3.2 3B	Meta (formerly Facebook)	-Multilingual support (8 languages) - Local device processing and strong privacy - Uses enhanced training techniques (refining, pruning) - Excellent in instruction following and summarization	- Weak in complex calculations - May generate incomplete responses due to tool errors

## 2.8 Conclusion on Literature Review

From studying the existing chatbots for education purpose, chatbots can bring more benefits to students for studying. Education organizations should learn and understand how chatbots can aid students along with the concerns to use chatbots for studies so this can allow students to use chatbots in the correct way to boost the grades, build up sufficient skills.

This project aims to develop a chatbot to become a strong tool for studying software testing just like how DeepSeek excels from the AI models in terms of aiding in research and study, these can help intended and targeted users to learn software testing CTFL syllabus more effectively and efficiently. Also, this project aims that the response generated by the chatbot is similar to ChatGPT's. This make the interaction between users and the chatbot to become more as ChatGPT is interesting to chat with, increasing the willingness of studying for users as one will not feel bored by looking at the dull interactions and responses from the chatbot.

Moreover, while the chatbots above typically save entire chat histories, chatbot lack the ability to save specific messages. As a result, when users wish to revisit a particular response, users must either scroll through long conversations or ask the chatbot again. This is very inconvenient when the chat history is vast, making the searching process time-consuming to locate important information. This project will propose a new function to allow users to save a specific response provided by the chatbot, saving users' time to do exhaustive searching just to refer back to the response.

The methodology is chosen to be applied on this project is V-Model.V-Model methodology focuses on early testing, each of the phases in the development phase are linked to testing phase respectively. By doing early testing, defects can be identified early and fix early so the usage of chatbot will not be affected. In addition, the chatbot is build for teaching software testing CTFL syllabus, V-Model emphasize on testing as well. This strengthen the core of developing this project, which testing is crucial for developing a project.

After deep considerations, the front-end web application framework this project has chosen to use for developing the chatbot for teaching software testing CTFL syllabus is React while the back-end web application framework chosen is Laravel. React has a robust libraries and reusable UI components, which can aid in speeding up the speed of my project's development. Moreover, React has a large community compared to the other frameworks. Solution,



resources can be searched easily when is needed developing this project. The main reason for choosing Laravel as the back-end web application framework is that the project involves sign up and login features, Laravel can provide a strong security feature for the project, such as protection for cross-site request forgery (CSRF), which helps to ensure the user data is being safeguarded. With this reason, Laravel excels compared to the other web application frameworks, hence Laravel is the suitable framework for developing the project.

LLaMA 3.2 3B is the best LLM for the chatbot, due to robustness and performance in following instructions. By fine-tuning the model with the software testing CTFL syllabus, the chatbot will be having human-like responses and can generate accurate responses based on users' questions.

## CHAPTER 3

### METHODOLOGY AND WORK PLAN

#### 3.1 Introduction

In this chapter, methodology used for this project will be described and also provide a clear view of the work plan. By outlining the process of developing the project following the methodology chosen, milestones and deliverable for the project can be seen clearly. In addition, tools used for developing this project will also be introduced as well.

#### 3.2 Methodology

The methodology chosen for this project is V-Model methodology. V-Model consists of two phases which are verification and validation phase, these two phases consist of different development phases along with matching testing phases respectively. For verification phases, the development phases include gathering of requirements, analysis of system, design of software, design of module.

In between verification and validation phases, there will be a coding phase. After that will be validation phases which are unit testing, integration testing, system testing, acceptance testing. Each of these validation phases will be linked to phases in the verification phase.

##### 3.2.1 Gathering of requirements

In this phase, literature review is conducted to understand the role of the chatbot in education sector, how chatbot would impact in student's studies. Throughout the literature review, requirements for the chatbot can be collected. Moreover, survey is needed and is conducted to collect requirements from potential users.

##### 3.2.1.1 Literature Review

Literature review is done on certain topics that is related to this project, not limited to benefits and downside of using chatbot for education purposes,

existing similar applications, comparison of AI models on the Internet, software development methodologies, web application development frameworks.

Identifying the advantages and disadvantages of utilizing chatbot for education purposes as this displays the role of chatbot in education, does chatbots really help students in studying? Or chatbot will sabotage the learning of students? These questions are answered after learning the advantages and disadvantages of using chatbot for learning.

In addition, by studying the existing chatbot applications that are used for education purpose helps in identifying how the chatbot should perform in order to provide help to those students which have issues in following up the software testing course or aid new learners.

By comparing each available AI models on the Internet, an understanding on what does each AI models excel in can be established, assisting in recognize which feature from each AI models is suitable for this project.

Studying each methodologies allows to select the suitable methodology, which is crucial for the project as methodology helps to keep track of the progress and checks my deliverables and whether milestones are achieved.

To select the suitable web application development framework for the project, conducting literature reviews on the frameworks are crucial as well. By doing so this project can select the suitable front-end and back-end web application development methodology for the chatbot, which are React and Laravel.

Studies on the Large Language Models (LLMs) were conducted as well to understand the performance, advantages which can helps to decide the most suitable LLM that is used for the chatbot. The selected model for the chatbot is LLaMA 3.2 3B that are developed by Meta. Fine-tuning process is done as well

to train the model to generate accurate response based on software testing CTFL syllabus.

### **3.2.2 Analysis on requirements**

A survey is conducted to collect feedbacks from students, the collected feedback and response is vital. The feedback and responses serve as a foundation for establishing the system requirements, ensuring that the chatbot addresses real user needs rather than assumptions. This process helps prioritize features to be included in the web application, such as quiz generation, clear explanations, and message-saving capabilities. Following the completion of collecting the feedback, the next step is analyzing the requirements to differentiate into functional and non-functional requirements for the project.

Functional requirements are vital because the web application need to include the features in functional requirements, hence the web application can achieve the vital functions to tackle the problems of this project. The functional requirements of this project can be categorized into two parts, one part is for students/software testers who wish to take the CTFL exam and self-learners, while the other part of the functional requirements targets lecturers/ trainer from the training academy. After sorting and listing out the functional requirements for each targeted users, listing out non-functional requirements need to be done as well.

Non-functional requirements controls the performance of the web application, this can provide targeted users a better experience while utilizing the chatbot to learn software testing CTFL syllabus.

### **3.2.3 Planning for the project**

Creating a timetable or timeline for the project is crucial, this allow the project to develop smoothly as each deliverables can be delivered before the deadline. Timeline can also help to control and keep track of the development process, such as checking on when will be the deadline for each task.

To develop a timeline for the project, Work Breakdown Structure (WBS) and Gantt Chart will be constructed. WBS aids in indicating what should be done in each phase of the project, what are the elements that need to be finished. WBS is a great tool for the project to identify what should be finished for each phase. Based on the constructed WBS, building a Gantt Chart of the project is needed.

For this project, two Gantt Charts will be constructed which is for FYP 1 and FYP 2 respectively. These Gantt Charts helps to give a clear view of what should be done for the WBS and how long each task takes to finish. Like setting a goal, Gantt Chart lays out the deliverables and milestones that needed to be achieved for the project. Milestones for the Gantt Chart indicate the completion of the tasks in each phases.

#### **3.2.4 Development phase**

Following the constructed WBS and Gantt Chart, a smooth development process should be carried out. The development of the web application will be starting with developing the modules mentioned above, training on the LLM need to be conducted to refine the chatbot's response generation. Allowing the chatbot to generate better and accurate response which is based on the software testing CTFL syllabus.

With the completion of developing modules and training on LLM, the development of the web application should start immediately, which is also matching the coding part of the methodology. The front-end and back-end of the web application needs to be developed, also setting up database for the web application. Next, integration of the chatbot into the web application needs to be carried out, allowing the web application to have the chatbot to handle questions from the users.

#### **3.2.5 Testing on the web application**

V-Model methodology emphasizes on testing, each phases in verification phase comes with a testing phase respectively. After designing the modules of the

chatbot, unit testing needs to be conducted to test each modules on the functionality, search for defects and fix defects swiftly.

Integration testing is also a must after finish developing the web application and integrated the chatbot model into the web application. This makes sure that the chatbot still works properly while modules are combined with other things like the web application. Then, to test the whole web application, system testing is conducted. System testing helps in testing the whole web application and checks the web application fulfils the requirements stated. In the above testing phases, defects need to be found and fixed as a preparation for the last testing which is User Acceptance Testing (UAT).

By inviting certain participants to conduct the UAT, feedback on the web application can be received, this aids in identifying which part of the web application can be improved in the future and parts that did not meet the requirements.

### **3.3 Development tools**

This section will be laying out the development tools that are used throughout the whole development process for the chatbot.

#### **3.3.1 Visual Studio Code**

Open-source and famous software developed by Microsoft, and is the first choice code editor for majority of the developers in the world. By utilizing Visual Studio Code (VS Code), codes can be written smoothly and test the web application as VS code provides a number of features and extensions. VS Code is a crucial tool for this project as this tool allows this project to successfully build the web application.

#### **3.3.2 React**

A JavaScript library that can provide libraries, reusable codes for building dynamic user interfaces for the chatbot. React have a large community on the Internet, available to provide solutions if any developers faces any issue on

developing projects. For the front-end design of the chatbot, React is utilized to build the user interface.

### **3.3.3 Laravel**

Laravel is a famous open source web application development framework, With the MVC framework, Laravel can aid in streamlining data flow for the chatbot with the help of Model, View, Controller components. Laravel is responsible for the back-end design of the chatbot.

### **3.3.4 MySQL**

An relational database management system (RDBMS), a famous database management system utilized by many platforms such as Netflix, Facebook, Airbnb and many others. Trusted by these companies as MySQL provides outstanding solidity, versatility, hence MySQL is selected to be the database management system for the chatbot for teaching software testing CTFL syllabus.

### **3.3.5 Draw.io**

One of the free online diagram designing software, draw.io is famous for the ability to provide developers tools to create flowcharts, UML, ERD and many other kind of diagrams. Not just being used for designing software development diagrams, and is also used to design many other kind of diagrams as users wish. Due to the ease of use, draw.io is selected to develop the diagrams for this project such as use case diagram, flowcharts.

### **3.3.6 Axure RP**

Axure RP is a strong tool for developers to design an interface for a software or system. Due to the software's robustness, dynamic UI can be created by using Axure RP, allowing developers can demonstrate the prototype of the software or system to intended users. With this reason, Axure RP is the main tool for developing the low fidelity prototype for the chatbot web application.

### **3.3.7 Google Colab**

Google Colab is a tool that includes notebooks for users to merge and run codes within one document. After selecting the suitable LLM for the chatbot, Google

Colab is selected and used to fine-tune the selected LLM, this can make the generated response become more accurate to the software testing CTFL syllabus.

### **3.3.8 SQLite**

SQLite is a lightweight relational database that stores data in a single file, making simple and efficient to implement for retrieval tasks. In this project, SQLite is used as the vector database for the Retrieval-Augmented Generation (RAG) process, where embeddings of the CTFL syllabus are stored and managed. When a user interacts with the chatbot, SQLite enables efficient retrieval of the most relevant syllabus content, which is then used to guide the language model's response. This approach ensures that the chatbot provides accurate, syllabus-aligned answers.

## **3.4 Work Plan**

### **3.4.1 Work Breakdown Structure (WBS)**

The task to be conducted of the project is broken down into the below WBS:

- 0.0 A chatbot for teaching software testing CTFL syllabus
- 1.0 Project Planning
  - 1.1 Research on the project
  - 1.2 Conduct meeting with supervisor
  - 1.3 Point out problems of the project
  - 1.4 Plan out objectives of the project
  - 1.5 Identify scope of the project
  - 1.6 Proposing solution for the project
  - 1.7 Identifying project approach
- 2.0 Requirement Analysis
  - 2.1 Literature Review
    - 2.1.1 Understand benefits and disadvantages of chatbot in education
    - 2.1.2 Reviewing existing chatbots for education
    - 2.1.3 Comparison between AI models on the Internet
    - 2.1.4 Literature review on software development methodology



- 2.1.5 Literature review on web application framework
  - 2.1.5.1 Reviewing front-end web application framework
  - 2.1.5.2 Reviewing back-end web application framework
- 2.1.6 Literature review on Large Language Models
- 2.2 Defining Requirements
  - 2.2.1 Determine functional requirements for the chatbot
  - 2.2.2 Determine non-functional requirements for the chatbot
  - 2.2.3 Documenting requirement specification documents
    - 2.2.3.1 Develop use case diagrams
    - 2.2.3.2 Construct use case descriptions
- 2.3 Project Scheduling
  - 2.3.1 Construct Work Breakdown Structure (WBS)
    - 2.3.1.1 Identify milestones of the project
  - 2.3.2 Construct Gantt Chart according to WBS
    - 2.3.2.1 Measure duration for each tasks
    - 2.3.2.2 Construct and check Gantt Chart
  - 2.3.3 Developing prototype of chatbot
    - 2.3.3.1 Developing low fidelity prototype
- 3.0 System Design
  - 3.1 Design of the chatbot
    - 3.1.1 Design chatbot modules
      - 3.1.1.1 Account Registration Module
      - 3.1.1.2 FAQ Module
      - 3.1.1.3 Quiz Module
    - 3.1.2 Training Model
    - 3.1.3 Develop chatbot web application
      - 3.1.3.1 Develop front-end of the chatbot
      - 3.1.3.2 Develop back-end of the chatbot
      - 3.1.3.3 Set up database for the chatbot
- 4.0 Testing
  - 4.1 Unit Testing

- 4.1.1 Module testing
    - 4.1.2 Identify and fix defects
  - 4.2 Integration Testing
    - 4.2.1 Module interaction testing
    - 4.2.2 Identify and fix integration problems
  - 4.3 System Testing
    - 4.3.1 Overall testing
    - 4.3.2 Check performance of the chatbot
    - 4.3.3 Fix remaining defects
  - 4.4 User Acceptance Testing (UAT)
    - 4.4.1 Conduct UAT with participants
    - 4.4.2 Collect feedback from participants
- 5.0 Deployment
  - 5.1 Deploy of A chatbot for teaching software testing CTFL syllabus

### **3.5 Gantt Chart**

Below are the Gantt Charts for the development of A chatbot for teaching software testing CTFL syllabus. The Gantt Charts are separated into two charts, which are for FYP 1 and FYP 2 respectively. Duration of each tasks and milestones are shown in the picture below.

### 3.5.1 Gantt Chart for FYP 1



Figure 3.1: FYP 1 Gantt Chart

### 3.5.2 Gantt Chart for FYP 2

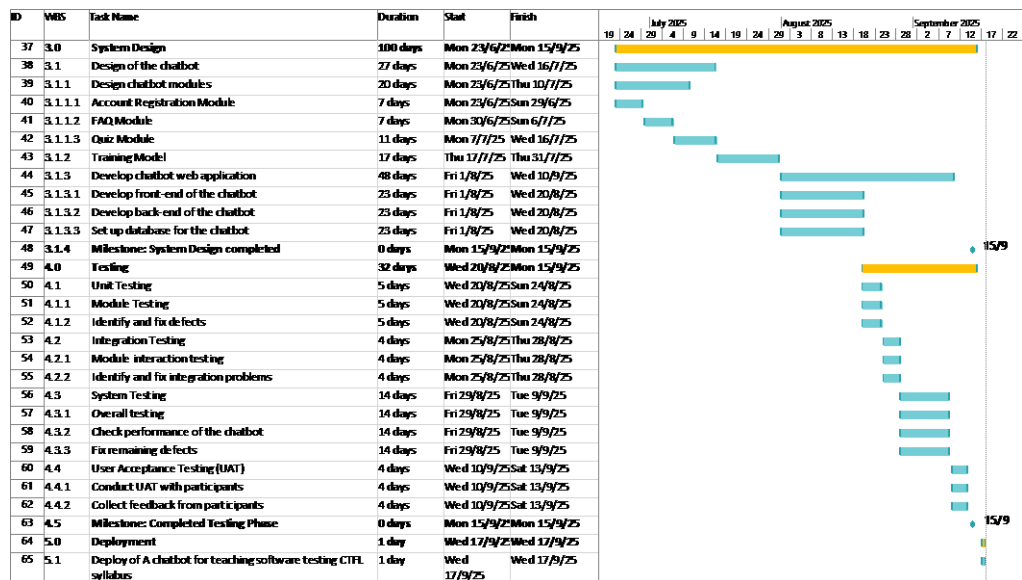


Figure 3.2: FYP 2 Gantt Chart

### **3.6 Test plan**

#### **3.6.1 Objective of test plan**

Test plan for this project is developed to check the functions of A chatbot for teaching software testing CTFL syllabus, defects of this project need to be investigated and fix, while verifying that the chatbot web application can meet all the requirements stated.

#### **3.6.2 Scope of Test**

The test scope of A chatbot for teaching software testing CTFL syllabus are mentioned in this section.

1. Test all the modules such as Account Registration Module, Quiz Module, Starred Message Module and the chatbot.
2. The response generation logic of the chatbot.
3. Chatbot's response generation with RAG implemented.

#### **3.6.3 Items need to be tested**

The items need to be tested will be discussed in the next chapter under functional requirements part.

#### **3.6.4 Chatbot web application features to be tested**

In this section, features of A chatbot for teaching software testing CTFL syllabus that need to be tested is discussed.

1. Sign up function that validates the email format and requires matching passwords in password and confirm password text fields.
2. Login function that checks the correctness of the password of the user.
3. Store chat history function that stores every available chat of each logged in users.
4. Create new chat function that allows users to create a new chat with the chatbot.
5. Delete chat history function that allows users to delete unused or previous chats.

6. Response generation function that checks the chatbot can provide response based on the question from users and software testing CTFL syllabus.
7. Manage user function that validates if admins can perform actions such as add, update, delete users.
8. Star message function that checks messages from the chatbot that can be starred, accessed again.

### **3.6.5 Chatbot web application features not to be tested**

Some of the features of A chatbot for teaching software testing CTFL syllabus are not being tested, those features are described in this section.

1. Changes in user interface (UI), such as changing colour of background or icons in the web application.

### **3.6.6 Test Basis**

The test are designed based on the functional requirements and non-functional requirements stated in this project, this helps validating that this project has fixed bugs and defects and satisfies the stated requirements.

### **3.6.7 Conditions of Test**

The conditions of testing A chatbot for teaching software testing CTFL syllabus are discussed in this section.

1. To check users can sign up for an account.
2. To check users can login into the account that were signed up.
3. To verify chatbot can generate accurate response based on software testing CTFL syllabus.
4. To verify chatbot can handle questions that are not related to software testing CTFL syllabus.
5. To verify admin can manage every registered users through the dashboard.

### **3.6.8 Criteria of Entry**

In this section, the condition of entering testing phase is stated.

1. Development of the chatbot web application is done, including all the features of the chatbot.
2. Developed test cases and are reviewed, approved.

### **3.6.9 Criteria of Exit**

In this section, the conditions need to be fulfilled in order to exit the testing phase will be described.

1. Detected bugs and defects are fixed.
2. Test cases need to have at least 95% of success rate.

## CHAPTER 4

### PROJECT INITIAL SPECIFICATION

#### 4.1 Introduction

This chapter will be discussing the requirements for the chatbot for teaching software testing CTFL syllabus. The chapter will cover the functional and non-functional requirements for the project, along with use case descriptions and diagrams.

#### 4.2 Requirements specification

##### 4.2.1 Collecting Requirements

To fully understand what potential users need from the chatbot in the web application, a survey is conducted to collect requirements and feedback for this project. The survey has responses from a total of 30 respondents, in this section the results will be shown and discussed.

##### 4.2.1.1 Survey Results

When you are studying the software testing syllabus, which of the following you find it hard?  
30 responses

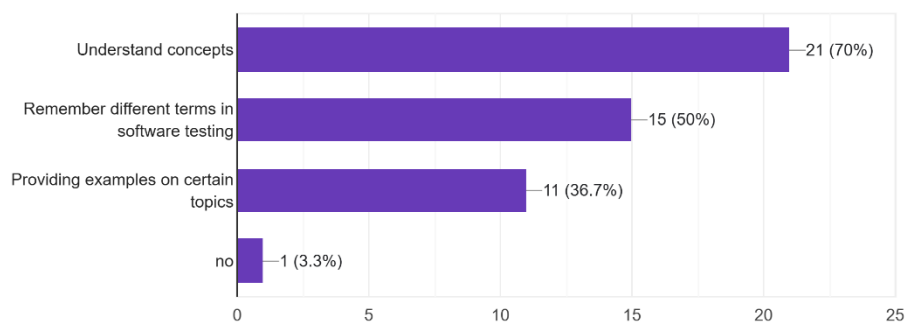


Figure 4.1: Result of survey questions (1)

From the figure above, majority of respondents stated that the major issue of learning the software testing syllabus is to understand the concepts. Up to 21 respondents or 70% of the respondents chosen this selection. The second problem faced by the respondents is remembering the terms in the syllabus, with a total of 15 respondents or 50% of the respondents answered.

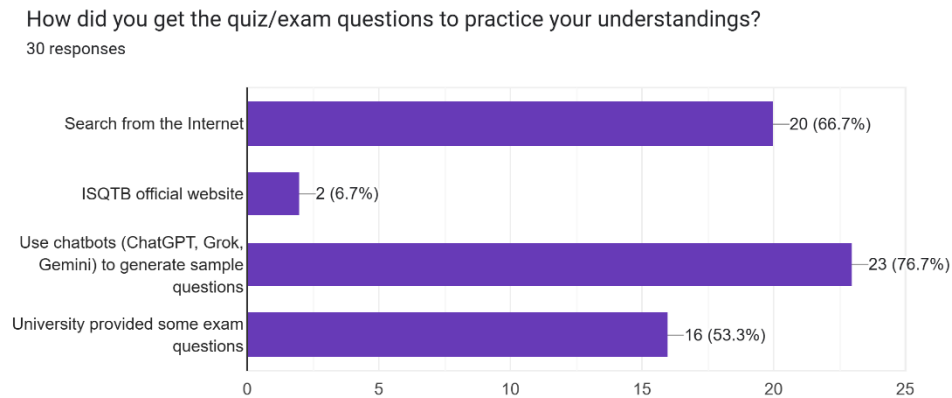


Figure 4.2: Result of survey questions (2)

In the image above, 76.7% of respondents (23 respondents) stated that chatbots like ChatGPT are utilized to help generate some questions to help to test understanding of the syllabus. While the second most respond is respondents search for exam questions from the Internet, with a total of 66.7% of respondents (20 respondents).

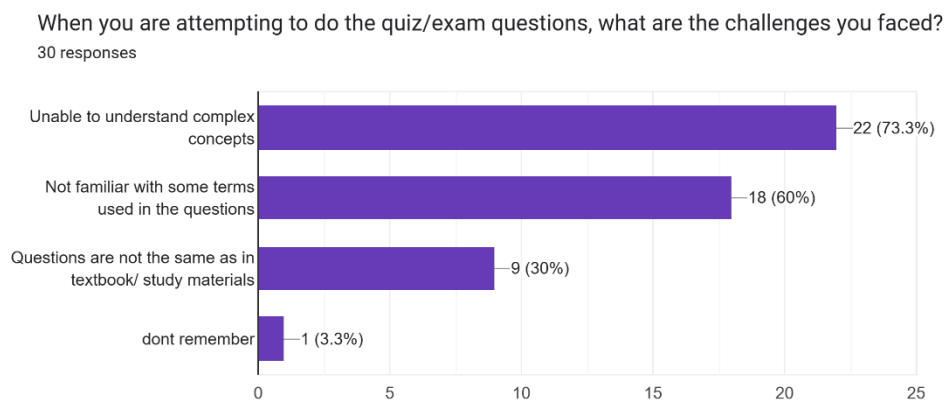


Figure 4.3: Result of survey questions (3)

73.3% of the respondents could not understand complex concepts while attempting the syllabus exam or test, while 60% of the respondents have the issue of unfamiliar with terms used in the question.



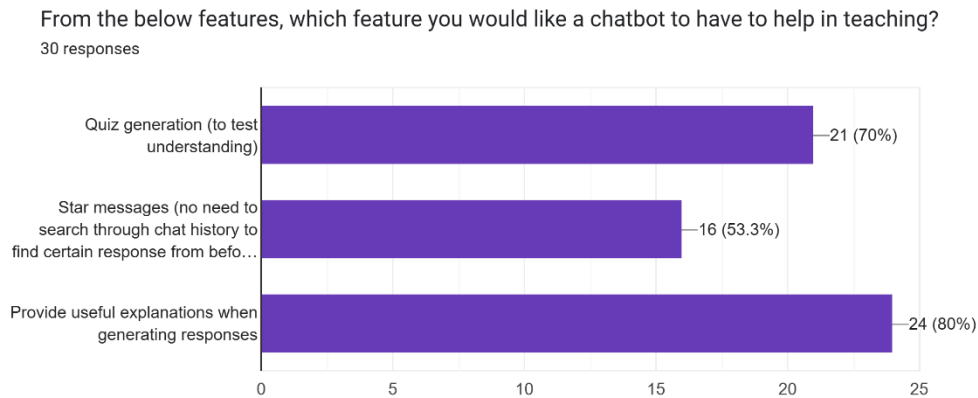


Figure 4.4: Result of survey questions (4)

The most wanted feature respondents wanted from the chatbot in this survey is to provide useful explanations, this feature has up to 80% of respondents choosing. The second feature to have for the chatbot is being able to generate quiz to test understanding, which this feature has 70% of respondents choosed.

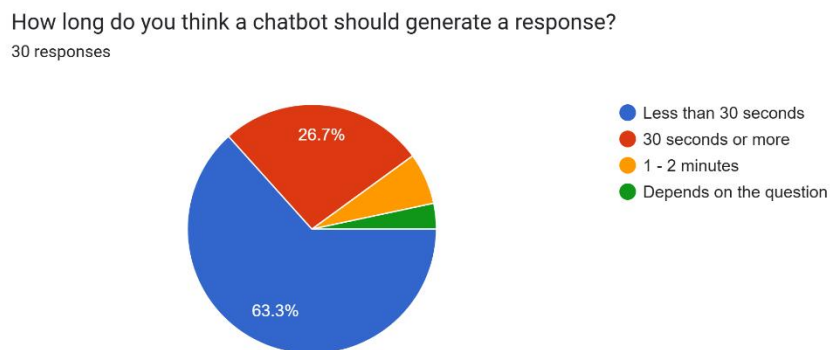


Figure 4.5: Result of survey questions (5)

As for how long the chatbot should take to generate a response, 63.3% of the respondents thinks should be less than 30 seconds. While 26.7% of respondents consider 30 seconds or more is fine.

#### 4.2.1.2 Conclusion on the survey

From the conducted survey, the main problems that respondents faced while learning the software testing CTFL syllabus is that understanding concepts and not familiar with terms used in the syllabus. These problems are stated in the problem statement in the previous chapters. This indicates that the chatbot should answer the prompt from the users and need to provide examples that can clearly state the definition of the concept or terms, aiding individuals to get a better understanding about the syllabus's topics.

The chatbot should also have the ability to generate questions or quiz to let users to attempt, allowing users to test the understanding and check where is the weak part. As for the response generation time, due to limitations of the LLM running locally, the chatbot will take some time to generate a response.

#### **4.2.2 Functional Requirements**

In this section, all the functional requirements for different type of users will be introduced.

##### **4.2.2.1 Functional Requirements for students/ self-learners / software testers who want to take the CTFL exam**

Table 4.1: Functional Requirements for Students/ Self-learners/ Software Testers

FR	Description
FR01	The web application shall allow users to sign up and login before using the chatbot.
FR02	The web application shall provide an interface where users can type in questions to ask questions.
FR03	The chatbot in the web application shall generate accurate responses about software testing CTFL syllabus based on users' questions.
FR04	The chatbot in the web application shall generate quizzes about software testing CTFL syllabus.

##### **4.2.2.2 Functional Requirements for lecturers/ trainers from the training academy**

Table 4.2: Functional Requirements for Lecturers/ Trainers

FR	Description
FR05	The web application shall allow users to sign up and login as lecturers/trainers before using the chatbot.
FR06	The web application shall provide an interface where users can type in questions to ask questions.
FR07	The chatbot in the web application shall generate accurate responses about software testing CTFL syllabus based on users' questions.
FR08	The chatbot in the web application shall generate quizzes about software testing CTFL syllabus.
FR09	The web application shall allow this type of users able to upload files related to software testing CTFL syllabus.

#### 4.2.2.3 Functional Requirements for Admins

Table 4.3: Functional Requirements for Admins

FR	Description
FR10	The web application shall allow users to sign up and login as admin.
FR11	The web application shall provide an interface where users can type in questions to ask questions.
FR12	The chatbot in the web application shall generate accurate responses about software testing CTFL syllabus based on users' questions.
FR13	The web application shall allow this type of users to manage the registered users of the web application.

#### 4.2.3 Non-functional Requirements

In this section, the non-functional requirements of the chatbot for teaching software testing CTFL syllabus are mentioned.

Table 4.4: Non functional requirements

NFR	Description
-----	-------------

NFR01	The web application shall authenticate the users before using.
NFR02	The chatbot in the web application shall generate responses successfully.

### 4.3 Use Case Diagram

Below shows use case diagrams of A chatbot for teaching software testing CTFL syllabus.

#### 4.3.1 Use Case Diagram for A chatbot for teaching software testing CTFL syllabus

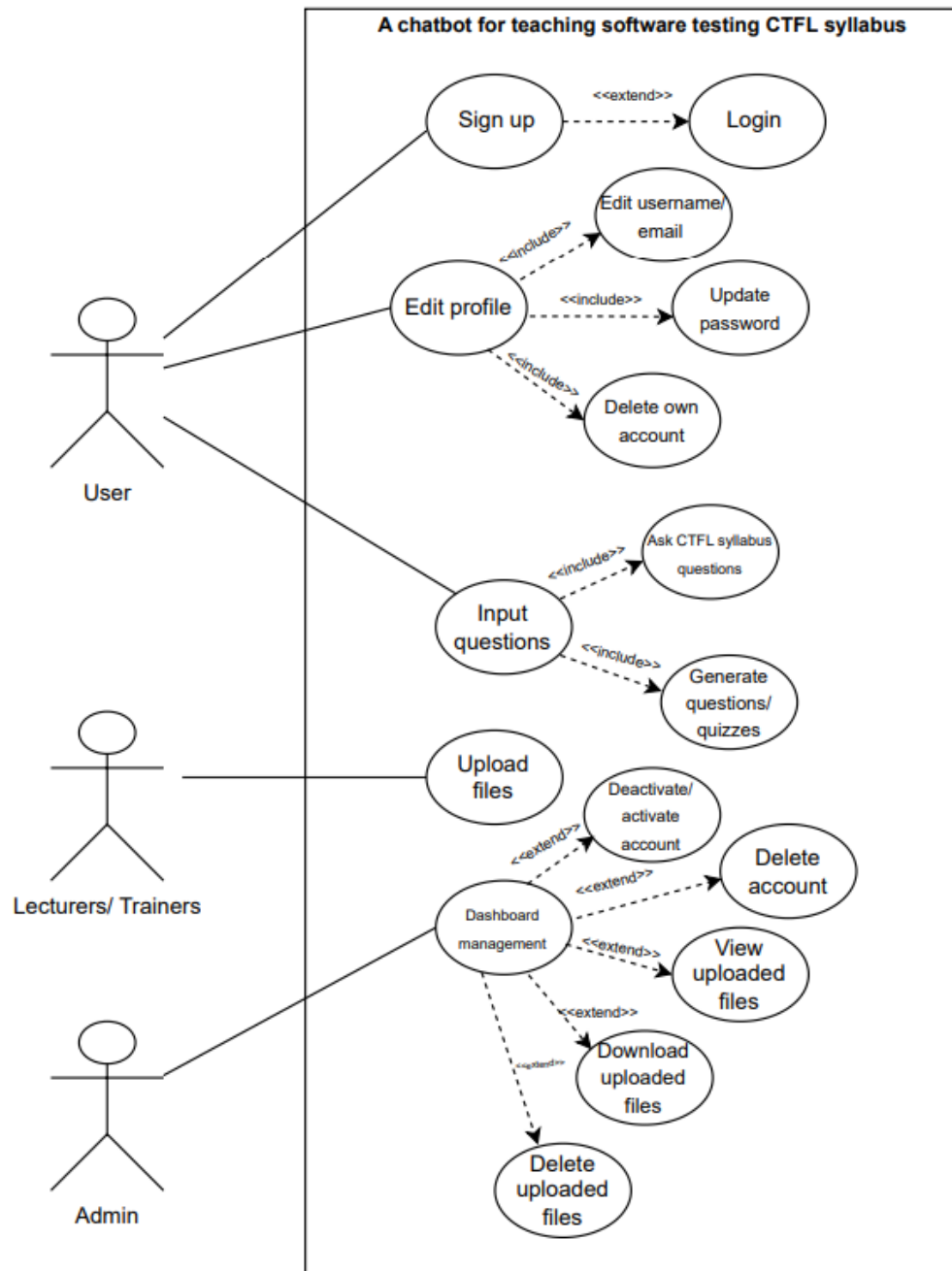


Figure 4.6: Use case diagram for the project.

#### 4.4 Use case description

This section will be showing every use case descriptions for each of the different types of users.

#### 4.4.1 Use case descriptions for students/ self-learners / software testers who want to take the CTFL exam

In this section, the users in the use case descriptions will be referring as students/ self-learners / software testers who want to take the CTFL exam.

##### 4.4.1.1 Sign up

Table 4.5: Sign up use case description

Use Case Name: Sign up		ID: UC01	Importance High	Level:
Primary Actor: Users		Use Case Type: Detail, Essential		
Stakeholders and Interests: Interested chatbot users that need to sign up for an account.				
Brief Description: This use case explains how users sign up for the chatbot web application.				
Trigger: Users want to use the chatbot.				
Relationships:  Association : User  Include : N/A  Extend : Login  Generalization: N/A				
Normal Flow of Events:  1. The web application will show a login page.  2. The users need to sign up for an account for login.  3. The users need to select students/learners for the sign up option.  4. The users need to fill in username, email, password, confirm password.  5. The web application will verify the email, password and confirm password.  6. The users done signing up for an account.				

<p>Sub-flows:</p> <p>1.1 If users does not fill in a proper email address, the web application will output error and ask for correct email address.</p> <p>1.2 If users does not fill in matching password for password and confirm password row, the web application will output error and ask for matching passwords.</p> <p>1.3 If users fill in proper email address, matching passwords, the web application will prompt successful sign up message and users can login afterwards.</p>
<p>Alternate/Exceptional Flows:</p>

#### 4.4.1.2 Login

Table 4.6: Login use case description

Use Case Name: Login	ID: UC02	Importance Level: High
Primary Actor: Users	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Users wishes to login into the chatbot web application.		
Brief Description: This use case describe how users login into own account for the web application.		
Trigger: Users wish to login into the web application.		
Relationships:  Association : Users  Include : N/A  Extend : N/A  Generalization: N/A		

Normal Flow of Events:
<ol style="list-style-type: none"> <li>1. The web application shows a page for login.</li> <li>2. The users input own email address and respective password.</li> <li>3. The web application database verifies the email address and password.</li> <li>4. The users login successfully to the web application.</li> </ol>
Sub-flows:
<ol style="list-style-type: none"> <li>1.5 If the users fill in wrong password, the web application will output error and request users to fill in password again.</li> <li>1.6 If the users fill in incorrect email address, the web application will output error and request users to fill in correct email address.</li> <li>1.7 If the users fill in correct email address and password, <b>flow no.4</b> continues.</li> </ol>
Alternate/Exceptional Flows:

#### 4.4.1.3 Edit profile

Table 4.7: Edit profile use case description

Use Case Name: Edit profile		ID: UC03	Importance Moderate	Level:
Primary Actor: Users		Use Case Type: Detailed		
Stakeholders and Interests: Users wishes to update personal information.				
Brief Description: This use case describe how users edit personal information for the web application.				
Trigger: Users wish to edit account’s personal information.				
Relationships:				
Association : Users				
Include : Edit username/email, Update password, Delete own account				
Extend : N/A				



Generalization: N/A
Normal Flow of Events: <ol style="list-style-type: none"> <li>1. The users direct to the profile page in the web application.</li> <li>2. The users browse through the page and check which information needs to update, or want to delete the account.</li> </ol>
Sub-flows: -
Alternate/Exceptional Flows:

#### 4.4.1.4 Edit username/ email

Table 4.8: Edit username/ email use case description

Use Case Name: Edit username/email		ID: UC04	Importance Moderate	Level:
Primary Actor: Users		Use Case Type: Detailed		
Stakeholders and Interests: Users wishes to update username or email.				
Brief Description: This use case describe how users edit email or username for the web application.				
Trigger: Users wish to edit account’s username or email address.				
Relationships:				
Association		: Users		
Include		: N/A		
Extend		: N/A		

Generalization: N/A
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> <li>1. The users direct to the profile page in the web application.</li> <li>2. The users browse to the profile information section.</li> <li>3. The users can fill in new username or email address.</li> <li>4. The users can click save button to save the updated field.</li> </ol>
<p>Sub-flows:</p> <p>3.1 If the users fill in incorrect email address, the web application will output error and request users to fill in correct email address.</p>
Alternate/Exceptional Flows:

#### 4.4.1.5 Update password

Table 4.9: Update password use case description

Use Case Name: Update password		ID: UC05	Importance Moderate	Level:
Primary Actor: Users		Use Case Type: Detailed		
Stakeholders and Interests: Users wishes to update password.				
Brief Description: This use case describe how users update password for the web application.				
Trigger: Users wish to update account’s password.				
Relationships:  Association : Users  Include : N/A				

Extend : N/A Generalization: N/A
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> <li>1. The users direct to the profile page in the web application.</li> <li>2. The users browse to the update password section.</li> <li>3. The users can fill in current password, new password and confirm password.</li> <li>4. The users can click save button to save the updated password.</li> </ol>
<p>Sub-flows:</p> <p>3.1 If the users fill in incorrect current password, the web application will output error and request users to fill in correct current password.</p> <p>3.2 If users does not fill in matching password for new password and confirm password row, the web application will output error and ask for matching passwords</p>
Alternate/Exceptional Flows:

#### 4.4.1.6 Delete own account

Table 4.10: Delete own account use case description

Use Case Name: Delete own account	ID: UC06	Importance Moderate	Level:
Primary Actor: Users	Use Case Type: Detailed		
Stakeholders and Interests: Users wishes to delete own account.			
Brief Description: This use case describe how users own account for the web application.			
Trigger: Users wish to remove the account.			

<p>Relationships:</p> <p>Association : Users</p> <p>Include : N/A</p> <p>Extend : N/A</p> <p>Generalization: N/A</p>
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> <li>1. The users direct to the profile page in the web application.</li> <li>2. The users browse to the delete account section.</li> <li>3. The users choose to delete the account.</li> <li>4. The users click delete button and the account will be deleted.</li> </ol>
<p>Sub-flows: -</p>
<p>Alternate/Exceptional Flows:</p> <p>4a. Users click on the cancel button, the delete account process will be abandoned.</p>

#### 4.4.1.7 Input questions

Table 4.11: Input questions use case description

Use Case Name: Input Questions		ID: UC07	Importance Level: High
Primary Actor: Users	Use Case Type: Detailed, Essential		
Stakeholders and Interests: Users wish to ask questions to the chatbot.			
Brief Description: This use case shows how users ask questions to the chatbot.			
Trigger: Users want to ask chatbot questions.			

Relationships:	
Association	: Users
Include	: Ask CTFL syllabus questions, Generate questions, quizzes
Extend	: N/A
Generalization: N/A	
Normal Flow of Events:	
<ol style="list-style-type: none"> <li>1. The users key in their question in the text field provided in the web application.</li> <li>2. The chatbot receives the question and start processing.</li> </ol>	
Categories of Prompt Can Be Used For The Chatbot:	
<ol style="list-style-type: none"> <li>1. Definition-based questions, example: “Why is Testing necessary?”</li> <li>2. Generating question/quiz, example: “Can you generate a quiz on Testing Principle?”</li> <li>3. Multiple choice quiz questions, example: “Can you generate a multiple choice type of quiz?”</li> </ol>	
Sub-flows: -	
Alternate/Exceptional Flows:	

#### 4.4.1.8 Ask CTFL syllabus questions

Table 4.12: Ask CTFL syllabus questions use case description

Use Case Name: Ask CTFL syllabus questions	ID: UC08	Importance Level: High
Primary Actor: Users	Use Case Type: Detailed, Essential	

Stakeholders and Interests: Users ask questions about the CTFL syllabus.
Brief Description: This use case shows how users ask questions related to CTFL syllabus.
Trigger: Users ask a prompt to the chatbot.
<p>Relationships:</p> <p>Association : Users</p> <p>Include : N/A</p> <p>Extend : N/A</p> <p>Generalization: N/A</p>
<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> <li>1. Users ask questions about CTFL syllabus.</li> <li>2. The chatbot checks with its trained materials database.</li> <li>3. The chatbot generates response based on the questions from the users.</li> </ol>
<p>Sub-flows:</p> <ol style="list-style-type: none"> <li>2.1 The input is related to software testing CTFL syllabus or trained materials, the chatbot will generate accurate response based on the input.</li> <li>2.2 The input is not related to software testing CTFL syllabus or trained materials, the chatbot will generate a message showing it could not answer the question from the user.</li> </ol>
Alternate/Exceptional Flows:

#### 4.4.1.9 Generate questions/ quizzes

Table 4.13: Generate questions/ quizzes use case description

Use Case Name: Generate questions/ quizzes	ID: UC09	Importance Level: High
Primary Actor: Users	Use Case Type: Detailed, Essential	

Stakeholders and Interests: Users ask the chatbot to generate questions to test.
Brief Description: This use case shows how users ask the chatbot to generate questions.
Trigger: Users ask chatbot provide questions.
Relationships: Association : Users Include : N/A Extend : N/A Generalization: N/A
Normal Flow of Events: 1. Users ask chatbot to generate questions about CTFL syllabus. 2. The chatbot checks with its trained materials database. 3. The chatbot generates questions for the users.
Sub-flows: -
Alternate/Exceptional Flows:

#### 4.4.2 Use case descriptions for lecturers/ trainers from the training academy

In this section, only the use case descriptions that are not included in the previous section will be explained, use cases like login, input questions are the same processes, hence these use cases will not be explained again.

##### 4.4.2.1 Sign up

Table 4.14: Sign up (lecturers) use case description

Use Case Name: Sign up		ID: UC10	Importance High	Level:
Primary Actor: Lecturers		Use Case Type: Detailed, Essential		
Stakeholders and Interests: Lecturers want to sign up for an account to utilize the chatbot web application.				
Brief Description: This use case explains how lecturers sign up for the chatbot web application.				
Trigger: Lecturers want to use the chatbot.				
Relationships:  Association : Lecturers  Include : N/A  Extend : Login  Generalization: N/A				
Normal Flow of Events:  1. The web application will show a login page. 2. The users need to sign up for an account for login. 3. The users need to select lecturers/trainers for the sign up option. 4. The users fill in their username, email, password, confirm password and lecturer code. 5. The web application will verify the email, password, confirm password and lecturer code. 6. The users done signing up for an account.				
Sub-flows:  4.1 If users does not fill in a proper email address, the web application will output error and ask for correct email address.  4.2 If users does not fill in matching password for password and confirm password row, the web application will output error and ask for matching passwords.  4.3 If users does not fill in correct educator code, the web application fill output error and ask for correct educator code.				



4.4 If users fill in proper email address, matching passwords and educator code, the web application will prompt successful sign up message and admins can login afterwards.
Alternate/Exceptional Flows:

#### 4.4.2.2 Upload files

Table 4.15: Upload files use case description

Use Case Name: Upload files		ID: UC11	Importance High	Level:
Primary Actor: Lecturers		Use Case Type: Detailed, Essential		
Stakeholders and Interests: Lecturers want to upload new syllabus file the chatbot web application.				
Brief Description: This use case explains how lecturers upload CTFL syllabus file for the chatbot web application.				
Trigger: Lecturers want to upload latest CTFL syllabus file.				
Relationships: <div>Association : Lecturers</div> <div>Include : N/A</div> <div>Extend : N/A</div> <div>Generalization: N/A</div>				
Normal Flow of Events: <div>1. The lecturers visit the upload file page.</div> <div>2. The lecturers need to select a file for upload.</div> <div>3. The lecturers can choose to leave a message for the admins.</div> <div>4. The lecturers can upload a file after selecting a file.</div>				

Sub-flows: -
Alternate/Exceptional Flows: 2a. Lecturers can leave the page or click reset to not upload any file.

#### 4.4.3 Use case descriptions for Admin

For the same use cases, such as login, input questions will not be explained in this section as the above sections have already demonstrated those use cases.

##### 4.4.3.1 Sign up

Table 4.16: Sign up (admin) use case description

Use Case Name: Sign up		ID: UC12	Importance High	Level:
Primary Actor: Admin		Use Case Type: Detail, Essential		
Stakeholders and Interests: Admin that need to sign up for an account.				
Brief Description: This use case explains how admin sign up for the chatbot web application.				
Trigger: Admin want to use or manage the web application.				
Relationships: <div>Association : Admin</div> <div>Include : N/A</div> <div>Extend : Login</div> <div>Generalization: N/A</div>				

<p>Normal Flow of Events:</p> <ol style="list-style-type: none"> <li>1. The web application will show a login page.</li> <li>2. The admin need to sign up for an account for login.</li> <li>3. The admin need to select admin for the sign up option.</li> <li>4. The admin fill in their username, email, password, confirm password and admin code.</li> <li>5. The web application will verify the email, password, confirm password and admin code.</li> <li>6. The admin done signing up for an account.</li> </ol>
<p>Sub-flows:</p> <ol style="list-style-type: none"> <li>1.1 If admin does not fill in a proper email address, the web application will output error and ask for correct email address.</li> <li>1.2 If admin does not fill in matching password for password and confirm password row, the web application will output error and ask for matching passwords.</li> <li>1.3 If admin does not fill in correct admin code, the web application fill output error and ask for correct admin code.</li> <li>1.4 If admin fill in proper email address, matching passwords and admin code, the web application will prompt successful sign up message and admins can login afterwards.</li> </ol>
<p>Alternate/Exceptional Flows:</p>

#### 4.4.3.2 Dashboard Management

Table 4.17: Dashboard Management use case description

Use Case Name: Dashboard Management		ID: UC13	Importance High	Level:
Primary Actor: Admin		Use Case Type: Detailed, Essential		
Stakeholders and Interests: Admin login to the web application and directly view the dashboard.				
Brief Description: This use case explains how Admin can view and utilize the dashboard.				

Trigger: After successful login for Admin type users.
Relationships: Association : Admin Include : N/A Extend : Deactivate/ activate account, Delete account, View uploaded files, Download uploaded files, Delete uploaded files Generalization: N/A
Normal Flow of Events: 1. The Admin login into the web application. 2. The web application will show the dashboard directly. 3. The dashboard will have two tables, list of registered users and uploaded files. 4. The Admin can choose to manage any of the items in the dashboard.
Sub-flows: -
Alternate/Exceptional Flows:

#### 4.4.3.3 Deactivate/ activate account

Table 4.18: Deactivate/ activate account use case description

Use Case Name: Deactivate/ activate account	ID: UC14	Importance Level: Moderate
Primary Actor: Admin	Use Case Type: Detailed	
Stakeholders and Interests: Admin need to perform deactivation or activation of a registered account.		

Brief Description: This use case explains how Admin can deactivate or activate an account.
Trigger: Admin needs to deactivate or activate an account.
Relationships: Association : Admin Include : N/A Extend : N/A Generalization: N/A
Normal Flow of Events: <ol style="list-style-type: none"> <li>1. The Admin visit the dashboard page.</li> <li>2. The Admin click on the edit button on the registered users list.</li> <li>3. The modal confirms whether to deactivate an active account or activate a deactivated account.</li> <li>4. The Admin can choose to deactivate or deactivate an account with the button provided in the modal.</li> </ol>
Sub-flows: -
Alternate/Exceptional Flows: 3a. Admin cancels the edit action in the modal, no account will be deactivated or activated and will be returned to dashboard page.

#### 4.4.3.4 Delete account

Table 4.19: Delete account use case description

Use Case Name: Delete account	ID: UC15	Importance Moderate	Level:
Primary Actor: Admin	Use Case Type: Detailed		

Stakeholders and Interests: Admin need to perform deletion of a registered account.
Brief Description: This use case explains how Admin can delete an account.
Trigger: Admin needs to delete an account.
Relationships: Association : Admin Include : N/A Extend : N/A Generalization: N/A
Normal Flow of Events: <ol style="list-style-type: none"> <li>1. The Admin visit the dashboard page.</li> <li>2. The Admin click on the delete button on the registered users list.</li> <li>3. The modal confirms whether to delete an account.</li> <li>4. The Admin can choose to delete an account by clicking the delete button in the modal.</li> </ol>
Sub-flows: -
Alternate/Exceptional Flows: <ol style="list-style-type: none"> <li>3a. Admin cancels the delete action in the modal, no account will be deleted and will be returned to dashboard page.</li> </ol>

#### 4.4.3.5 View uploaded files

Table 4.20: View uploaded files use case description

Use Case Name: View uploaded files	ID: UC16	Importance Moderate	Level:
Primary Actor: Admin	Use Case Type: Detailed		

Stakeholders and Interests: Admin want to view the content of the uploaded files in the dashboard.
Brief Description: This use case explains how Admin can view an uploaded file.
Trigger: Admin needs verify the content of uploaded files.
Relationships: Association : Admin Include : N/A Extend : N/A Generalization: N/A
Normal Flow of Events: 1. The Admin visit the dashboard page. 2. The Admin click on the hyperlink, which is the uploaded file name. 3. The browser open new tab to open the uploaded file. 4. The Admin can view the content of the uploaded file.
Sub-flows: -
Alternate/Exceptional Flows:

#### 4.4.3.6 Download uploaded files

Table 4.21: Download uploaded files use case description

Use Case Name: Download uploaded files	ID: UC17	Importance Moderate	Level:
Primary Actor: Admin	Use Case Type: Detailed		
Stakeholders and Interests: Admin wishes to download the uploaded file to save in database.			

Brief Description: This use case explains how Admin can download an uploaded file.
Trigger: Admin wants to download a selected file.
Relationships: Association : Admin Include : N/A Extend : N/A Generalization: N/A
Normal Flow of Events: <ol style="list-style-type: none"> <li>1. The Admin visit the dashboard page.</li> <li>2. The Admin click on the download button on the uploaded files list.</li> <li>3. The modal confirms whether to download the file.</li> <li>4. The Admin can choose to download the file by clicking the download button in the modal.</li> </ol>
Sub-flows: -
Alternate/Exceptional Flows: <ol style="list-style-type: none"> <li>3a. Admin cancels the download action in the modal, no files will be downloaded and will be returned to dashboard page.</li> </ol>

#### 4.4.3.7 Delete uploaded files

Table 4.22: Delete uploaded files use case description

Use Case Name: Delete uploaded files	ID: UC18	Importance Moderate	Level:
Primary Actor: Admin	Use Case Type: Detailed		



Stakeholders and Interests: Admin need to perform deletion of an uploaded file.
Brief Description: This use case explains how Admin can delete an uploaded file.
Trigger: Admin needs to delete an uploaded file.
Relationships: Association : Admin Include : N/A Extend : N/A Generalization: N/A
Normal Flow of Events: <ol style="list-style-type: none"> <li>1. The Admin visit the dashboard page.</li> <li>2. The Admin click on the delete button on the uploaded files list.</li> <li>3. The modal confirms whether to delete an uploaded file.</li> <li>4. The Admin can choose to delete the uploaded file by clicking the delete button in the modal.</li> </ol>
Sub-flows: -
Alternate/Exceptional Flows: <ol style="list-style-type: none"> <li>3a. Admin cancels the delete action in the modal, no files will be deleted and will be returned to dashboard page.</li> </ol>

#### 4.5 Flowchart of A chatbot for teaching software testing CTFL syllabus web application

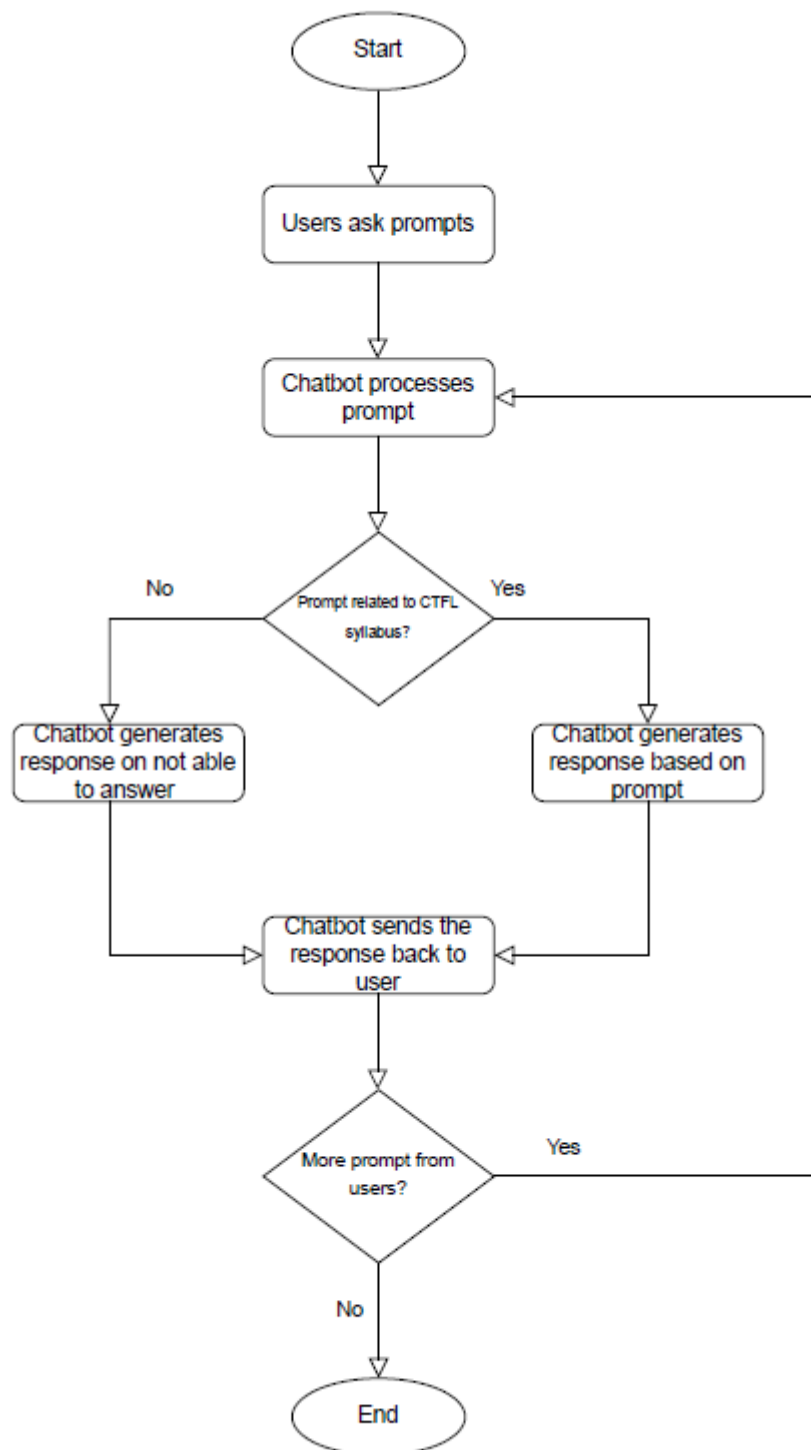


Figure 4.7: Flowchart

## CHAPTER 5

### SYSTEM DESIGN

#### 5.1 Introduction

In this chapter, the design of the web application of this project will be shared. The designs like system architecture design, ERD diagram and UI design will all be demonstrated.

#### 5.2 System Architecture Design

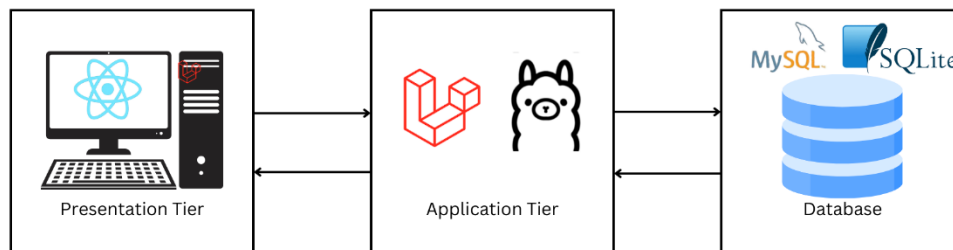


Figure 5.1: System Architecture Diagram

The above image states how the web application developed in this project works. The front-end of the web application is rendered by React, providing a dynamic view for users while utilizing the chatbot and provides a real-time update on the chat page. React will be handling the input or prompts from users, then send the prompt to back-end for generating response. React is a great front-end tool to provide users a good usage experience.

The back-end of this project is supported by Laravel, upon receiving a prompt or question from users in the front-end, Laravel will handle the request and manage the flow of the database and the Ollama model. With the routing and middleware provided by Laravel, all request will be authenticated and errors will be handled properly. Afterwards, when receive a prompt from front-end, the Ollama model is called to analyze the prompt, then check which part in the syllabus is the answer to the prompt.

Lastly, the database of this project includes MySQL and SQLite. MySQL is used to save all the chats, users, uploaded files and so on. While SQLite is the main tool for vector database, which was used to store the CTFL syllabus required for the model to refer and search for answers. Upon retrieving the correct record in the database, the model will tailor an answer and generate a response back to the user, allowing a smooth flow of conversation between users and the chatbot.

### 5.3 Entity Relationship Diagram (ERD)

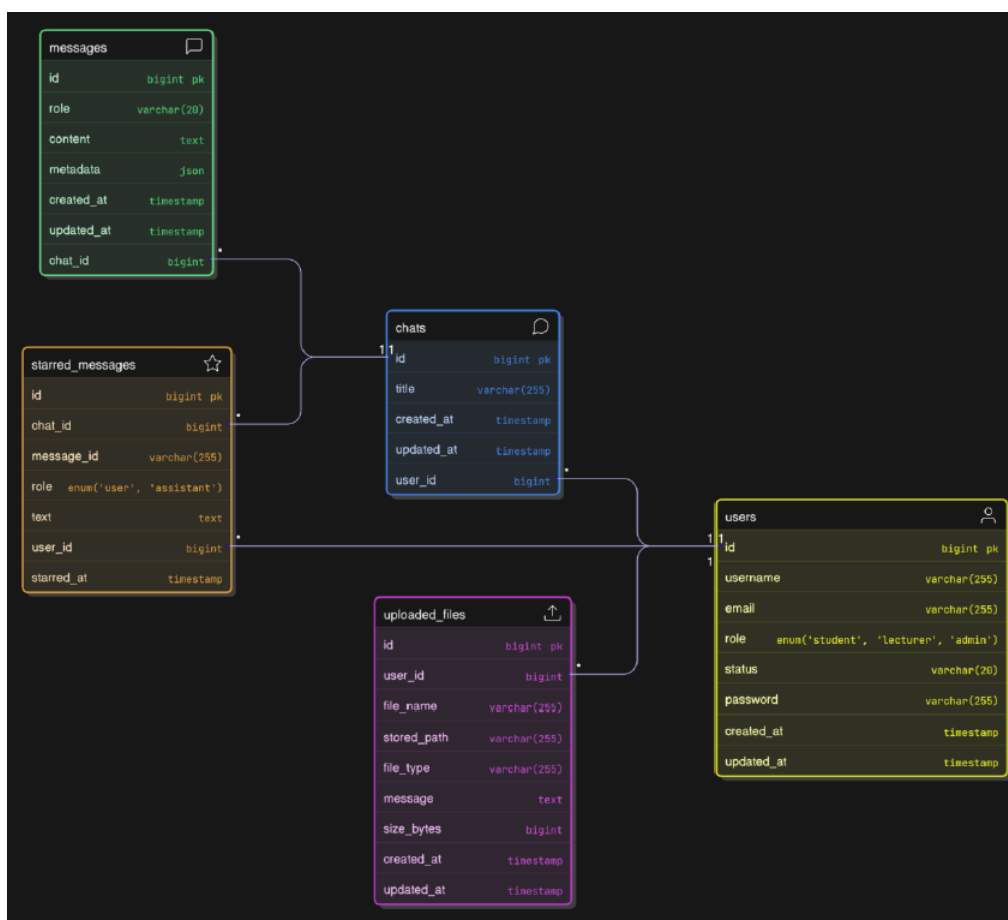


Figure 5.2: ERD

#### 5.3.1 Data Dictionary

Table 5.1: Data dictionary for users table

Column Name	Key	Data Type	Description

id	Primary	bigint	The unique id of each user
username	-	varchar(255)	The username of each user
email	-	varchar(255)	The email of each user
role	-	enum('student', 'lecturer', 'admin')	The role of each user
status	-	varchar(20)	The status of the user's account
password	-	varchar(255)	The password of the user's account
created_at	-	timestamp	User account's created time
updated_at	-	timestamp	User account's updated time

Table 5.2: Data dictionary for chats table

Column Name	Key	Data Type	Description
id	Primary	bigint	The unique id for each chat
user_id	Foreign	bigint	The chat belongs to which user
title	-	varchar(255)	The title of the chat
created_at	-	timestamp	The chat's created time
updated_at	-	timestamp	The chat's updated time

Table 5.3: Data dictionary for messages table

Column Name	Key	Data Type	Description
id	Primary	bigint	The unique id of each message

role	-	varchar(20)	The role of the user for the message
content	-	text	The content of the message
metadata	-	json	Additional structured details of the message in JSON.
chat_id	Foreign	bigint	The chat where the message belongs to
created_at	-	timestamp	The message's created time
updated_at	-	timestamp	The message's updated time

Table 5.4: Data dictionary for starred\_messages table

Column Name	Key	Data Type	Description
id	Primary	bigint	The unique id of each starred message
chat_id	Foreign	varchar(255)	The chat where contains the starred message
message_id	Foreign	varchar(255)	The starred message
role	-	enum('student', 'lecturer', 'admin')	The role of the user which starred the message
text	-	text	The content of the starred message
user_id	Foreign	varchar(255)	The user that starred the message
starred_at	-	timestamp	User account's created time

Table 5.5: Data dictionary for uploaded\_files table

Column Name	Key	Data Type	Description
id	Primary	bigint	The unique id of the uploaded file
user_id	Foreign	bigint	The id of the user which uploaded the file
file_name	-	varchar(255)	The name of the uploaded file
stored_path	-	varchar(255)	The stored path of the file
file_type	-	varchar(255)	The type of the uploaded file
message	-	text	The message left by users before upload a file
size_bytes		bigint	The size of the uploaded file
created_at	-	timestamp	The time of the file being uploaded
updated_at	-	timestamp	The time of the file being updated

#### 5.4 System User Interface (UI) Design

The design of user interface of this project will be demonstrated in this section. It will include all pages in this project.

### 5.4.1 Login UI

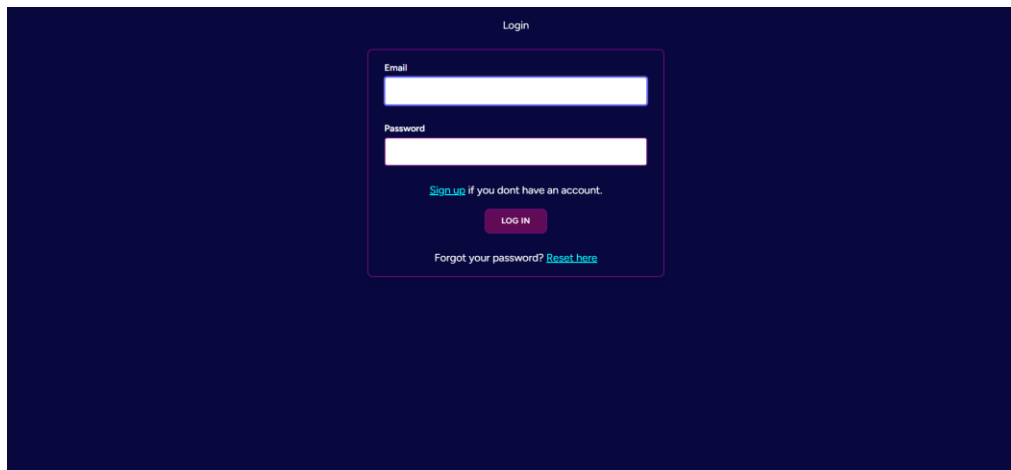


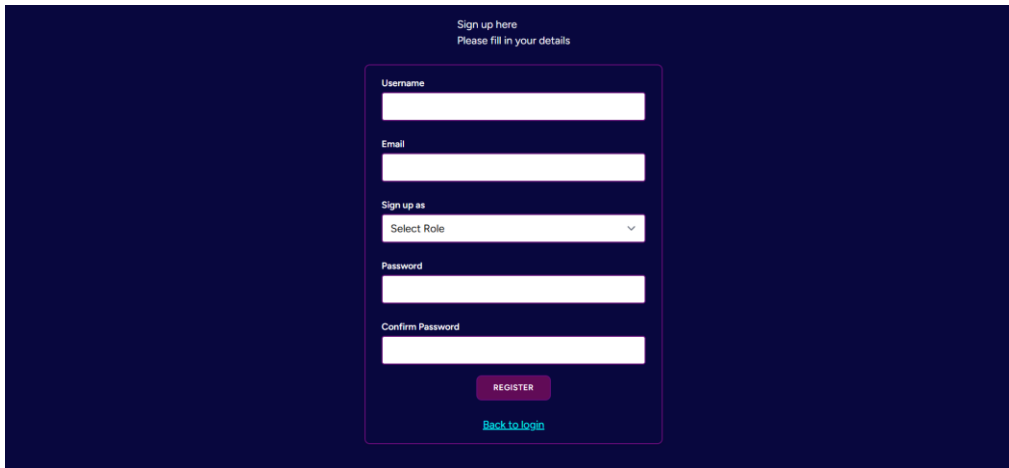
Figure 5.3: Login page UI

The login will require users to fill in email and correct password to login. After users filled in the credentials, users can click the log in button to login into the registered account.. New users are required to sign up for a new account for login, in this login page, there is a sign up link provided, new users can visit the registration page by clicking the link.

On top of that, there is also a hyperlink for the registered users, which serve as resetting password for login. There could be times when users accidentally forgotten the password, with the reset password link available, users can reset password to login back again into the registered account without having to lose all the chat histories. All links comes with light blue color and is underlined, indicating the words are links that can be accessed.



### 5.4.2 Registration UI



Sign up here  
Please fill in your details

Username

Email

Sign up as  
Select Role

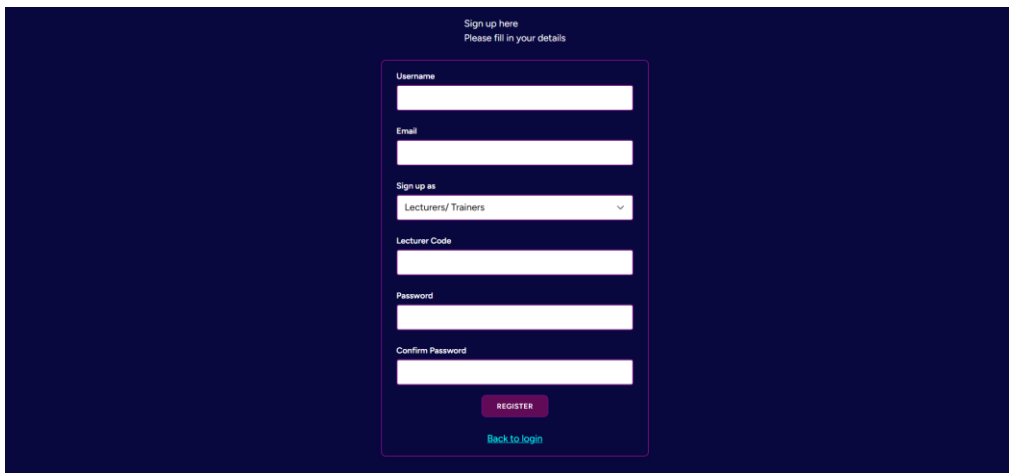
Password

Confirm Password

REGISTER

[Back to login](#)

Figure 5.4: Registration page UI



Sign up here  
Please fill in your details

Username

Email

Sign up as  
Lecturers/Trainers

Lecturer Code

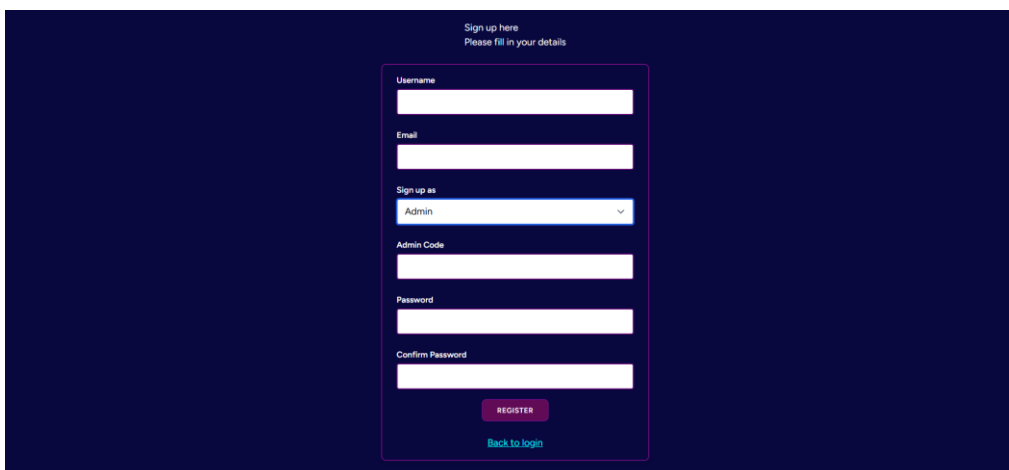
Password

Confirm Password

REGISTER

[Back to login](#)

Figure 5.5: Registering for Lecturers/Trainers role



Sign up here  
Please fill in your details

Username

Email

Sign up as  
Admin

Admin Code

Password

Confirm Password

REGISTER

[Back to login](#)

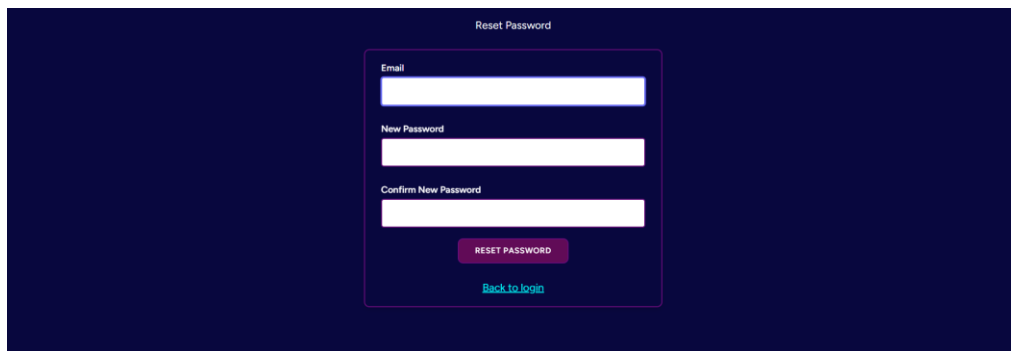
Figure 5.6: Registering for Admin role

The above figures shows the registration page after accessing the sign up link provided in the login page, new users can fill in credentials for registering a new account for login. New users are required to fill in username, email, password, confirm password, and new users are required to select a role for registration. When students or software testers wishes to create a new account, users may select the “Student/Software Testers” role.

In figure 5.4 and 5.5, there is one extra column which is the code. This code will only be shown when users select “Lecturers/Trainers” or “Admin” role. Correct lecturer code or admin code must be provided to register lecturers or admin role account. Lecturers or Trainers can get this code from the admin.

After filling in the credentials correctly, new users may click on the register button to register a new account, and will be directed back to login page. If new users want to return back to login page without registering a new account, users may access the back to login link provided in the page.

### 5.4.3 Forget Password UI



The image shows a 'Reset Password' form with a dark blue background. The form is centered and contains three white input fields labeled 'Email', 'New Password', and 'Confirm New Password'. Below these fields is a red button labeled 'RESET PASSWORD' and a blue link labeled 'Back to login'.

Figure 5.7: Forget Password page UI

This is the reset password page which can be accessed in the login page as well. This page can allow existing users to reset the account password if users ever forget password. Users have to fill in a registered email to reset the password, else the reset password action will not be completed. After successfully resetting password, users will be directed back to login page for login. Same as previous page, users can return to login page without having to reset the password by accessing the link provided.

### 5.4.4 Chat Page UI

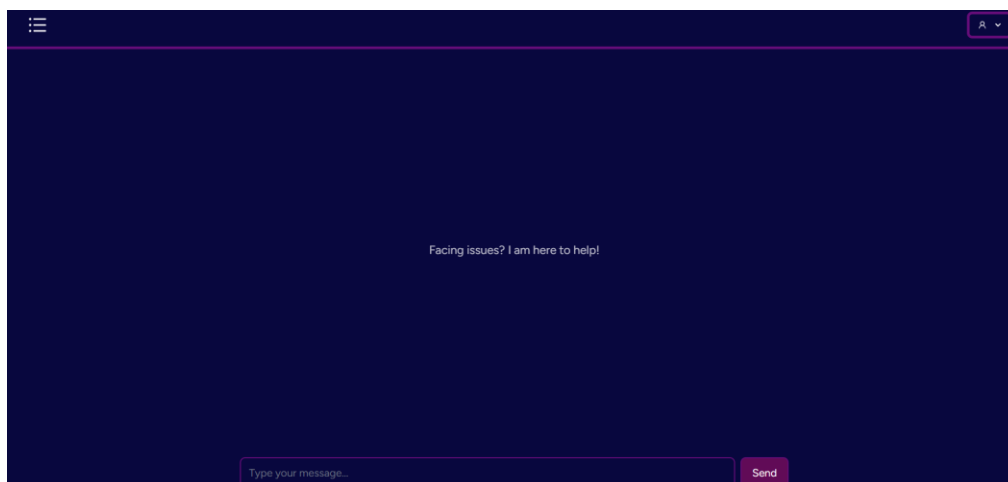


Figure 5.8: Chat page UI

After logging in, a chat page is the first thing to be seen. Users can start to chat with the chatbot that has been integrated into this project's web application and start learning the software testing CTFL syllabus.

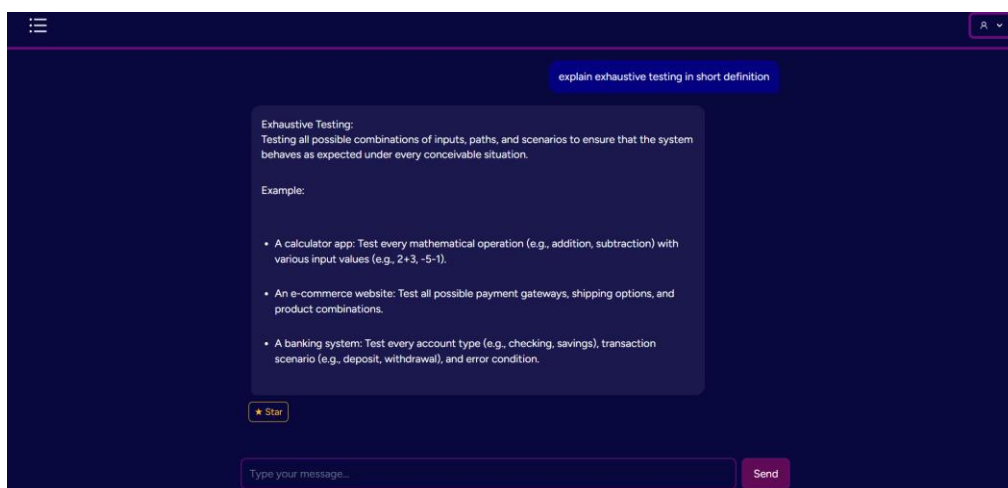


Figure 5.9: Example chat in chat page

Users may ask questions in the chat input box provided in the bottom of the page, and send the message through the enter button on keyboard or click the send button right beside the chat input box. In a case where users think that the response from the chatbot is useful, users can star the message by clicking on the star button provided.

On the top left and top right corner of the page, there are two different icons. The top left icon is a clickable icon for opening the sidebar, users can see chat histories and other actions. As for the top right icon is also a clickable icon for opening a dropdown menu, allowing users to visit different pages.

#### 5.4.5 Sidebar UI

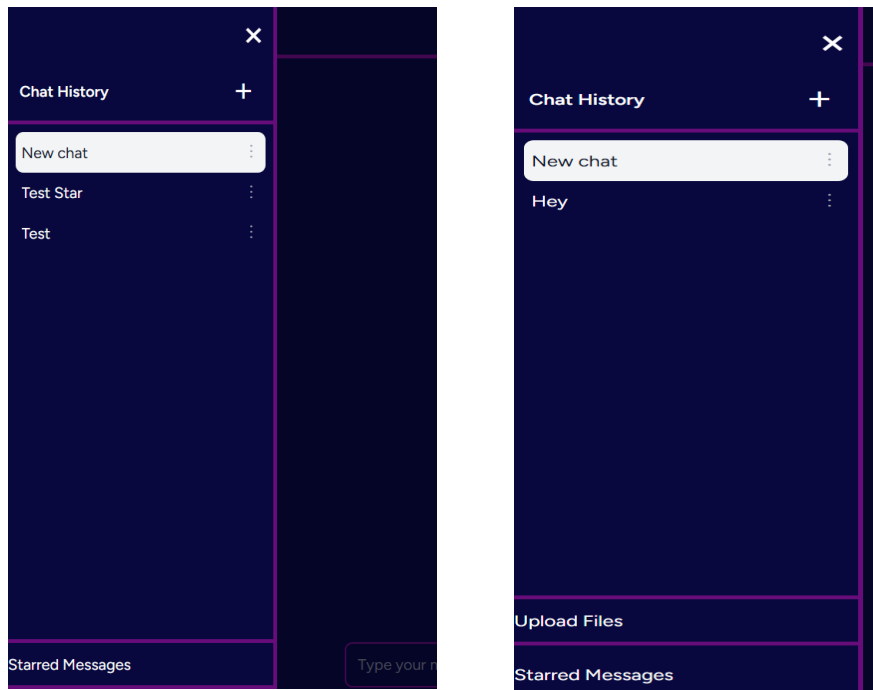


Figure 5.10: Sidebar UI

Above figures is the sidebar, allowing users to view the previous chat history lists. At the bottom of the sidebar, users can access to the starred messages page, this is a page where users can view all the messages have been starred. Aside from that, only “Lecturer/ Trainers” role users can access to the upload file page, which allows users to upload the syllabus file for admins to update the RAG knowledge base for the chatbot.



Figure 5.11: Sidebar actions

Other than viewing chat history list and access to other pages, the sidebar also provide users different sets of actions, such as add new chat, rename the chat or delete the chat. By simply click on the “+” button, users can choose to add new chat with the box provided, after confirmation, users can successfully add a new chat and will be directed to the new chat page.

To manage the existing chat histories, users can click on the three dots beside each chat history lists. Users can choose to either rename the chat, or completely delete the whole chat. Lastly, the “x” button is for users to close the sidebar.

#### 5.4.6 Profile Dropdown Menu UI

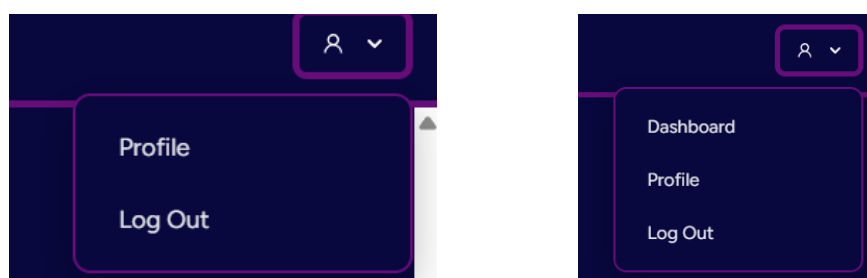


Figure 5.12: Profile icon dropdown

By accessing the profile icon on the top right of the page, users will see a dropdown menu. Users can select to visit the profile page, or to log out of this web application. As for “Admin” role users, the dropdown menu will have one extra section which is for visiting the dashboard page.

#### 5.4.7 Starred Message UI

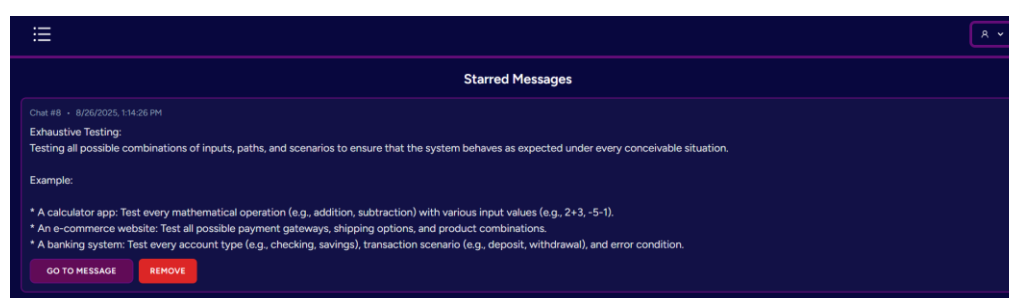


Figure 5.13: Starred Message page UI

After users star a message in the chat page, users may visit this page through the sidebar. This page will list out all the messages that have been starred by the user, allowing users to view the important response from the chatbot, saving the time to do exhaustive searching by scrolling through the chat. Users can visit back to the message by clicking the “go to message” button, this button will direct users back to the starred message immediately. To remove the message, users can also remove the message by clicking on the remove button.

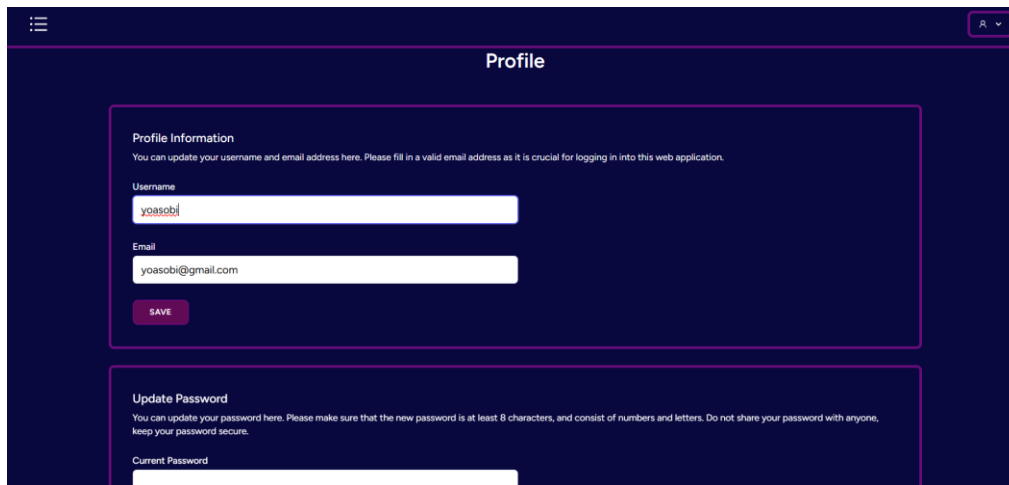
#### 5.4.8 Upload File UI

Figure 5.14: Upload file page UI

Upload file page is only accessible by lecturers/trainers role users, this page allows users to upload the syllabus file to admin, since lecturers or trainers are the one who can receive the materials first. Users may access the upload file button to select a file from the device for upload, after chosen one file, the name of the file will be stated and users can upload the file by clicking the upload file button. If there are no attached files, users cannot click the upload.

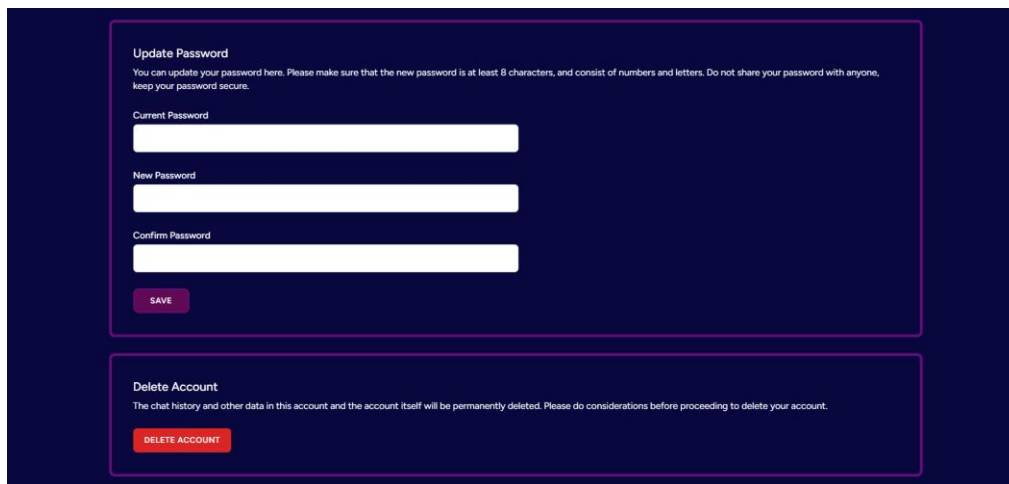
Additionally, users can also leave a message for the admin, such as stating what the uploaded file is, reducing confusion for admin users. If users want to remove the selected file, users may either select a new file or click on the reset button provided.

### 5.4.9 Profile Page UI



The image shows the upper portion of a profile page. At the top, there is a dark blue header with a hamburger menu icon on the left and a user profile icon with a dropdown arrow on the right. Below the header, the word "Profile" is centered. The main content area has a light blue background and contains two white-bordered boxes. The first box, titled "Profile Information", includes a sub-header and a paragraph: "You can update your username and email address here. Please fill in a valid email address as it is crucial for logging in into this web application." It features two input fields: "Username" with the value "yoasobi" and "Email" with the value "yoasobi@gmail.com". A red "SAVE" button is at the bottom of this box. The second box, titled "Update Password", has a sub-header and a paragraph: "You can update your password here. Please make sure that the new password is at least 8 characters, and consist of numbers and letters. Do not share your password with anyone, keep your password secure." It starts with a "Current Password" input field.

Figure 5.15: Upper part of profile page UI

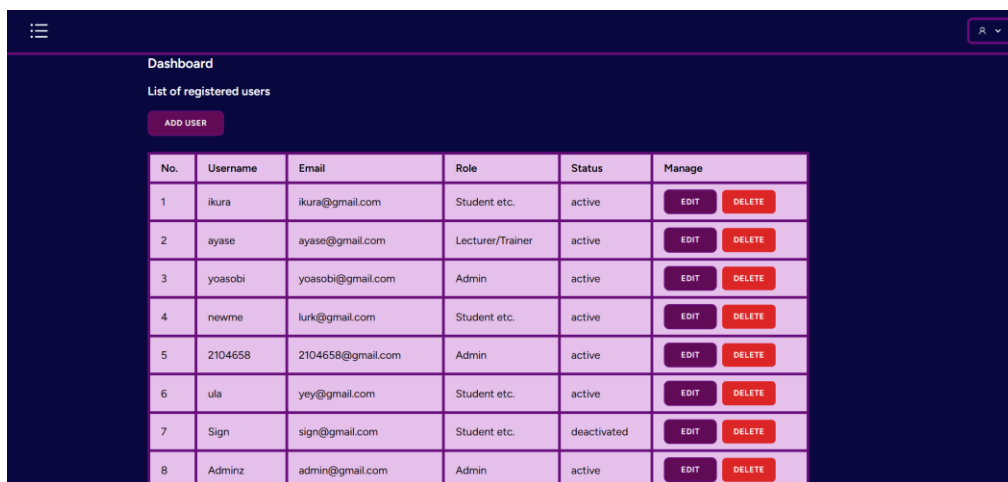


The image shows the lower portion of the profile page. It continues the "Update Password" section with two more input fields: "New Password" and "Confirm Password". A red "SAVE" button is positioned below these fields. Below the "Update Password" box is another white-bordered box titled "Delete Account". It contains a sub-header and a paragraph: "The chat history and other data in this account and the account itself will be permanently deleted. Please do considerations before proceeding to delete your account." At the bottom of this box is a red "DELETE ACCOUNT" button.

Figure 5.16: Lower part of profile page UI

The above figures demonstrate the profile page, which is accessible by all types of users. In this page users can either update the personal information such as email, username or password. In the bottom of the page, users can also delete account, deleting the account also means removing all the data such as account data, chat histories.

### 5.4.10 Dashboard Page UI



No.	Username	Email	Role	Status	Manage
1	ikura	ikura@gmail.com	Student etc.	active	EDIT DELETE
2	ayase	ayase@gmail.com	Lecturer/Trainer	active	EDIT DELETE
3	yoasobi	yoasobi@gmail.com	Admin	active	EDIT DELETE
4	newme	lurk@gmail.com	Student etc.	active	EDIT DELETE
5	2104658	2104658@gmail.com	Admin	active	EDIT DELETE
6	ula	yey@gmail.com	Student etc.	active	EDIT DELETE
7	Sign	sign@gmail.com	Student etc.	deactivated	EDIT DELETE
8	Adminz	admin@gmail.com	Admin	active	EDIT DELETE

Figure 5.17: Upper part of Dashboard page UI



11	user4	user4@gmail.com	Student etc.	active	EDIT DELETE
12	student222	student123@gmail.com	Student etc.	active	EDIT DELETE
13	lecturer3	lecturer3@gmail.com	Lecturer/Trainer	active	EDIT DELETE
14	fatimah	audah@utar.edu.my	Lecturer/Trainer	active	EDIT DELETE

No.	From	Uploaded file	Message	Manage
1	fatimah	<a href="#">ISTQB_CTFL_Syllabus-v4.0.pdf</a>	-	DOWNLOAD DELETE
2	lecturer3	<a href="#">ISTQB_CTFL_Syllabus-v4.0.pdf</a>	UAT demo upload	DOWNLOAD DELETE
3	ayase	<a href="#">ISTQB_CTFL_Syllabus-v4.0.pdf</a>	CTFL syllabus	DOWNLOAD DELETE
4	ayase	<a href="#">UECS2363A1-2104658.pdf</a>	try upload	DOWNLOAD DELETE
5	ayase	<a href="#">Practical 3 (Wk4) Principles of Code Refactoring using Copilot.docx</a>	-	DOWNLOAD DELETE
6	ayase	<a href="#">EYPI-rubrics.pdf</a>	hi	DOWNLOAD DELETE

Figure 5.18: Lower part of Dashboard page UI

The dashboard page contains two tables, list of registered users and uploaded files. The list of registered users will list out all the registered users, including username, email, and the status of the account. Admin can choose to edit the account's status, whether to deactivate or activate the account. Deactivated accounts will not be able to log in again until activated. Admin can also delete any registered accounts.

As for the uploaded files table, the table will state the uploaded files, uploaded by who and message. Admins can view the content of the files by



clicking the filename. After viewing the content, admins can choose to delete the file or download the file and prepare for update the RAG knowledge base.

#### 5.4.11 Add User UI

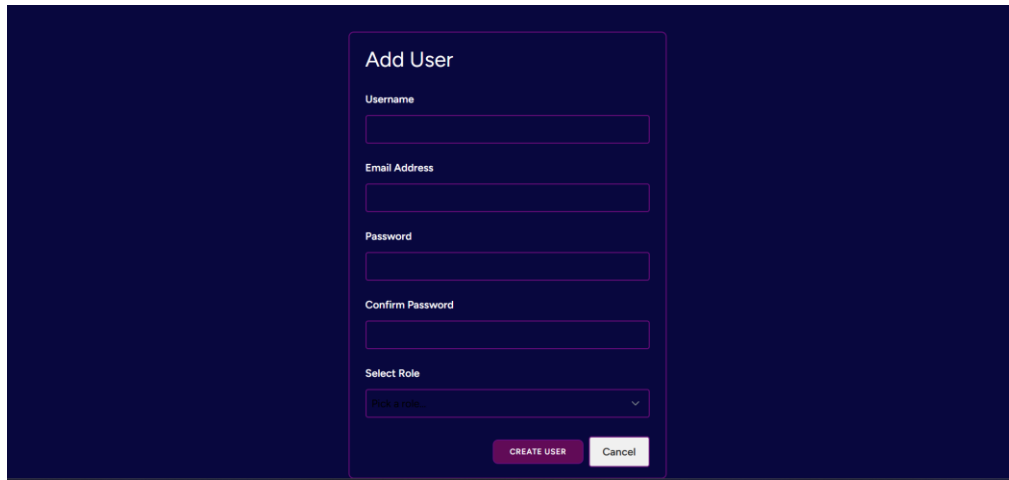
The image shows a web application interface with a dark blue background. A white modal form titled "Add User" is centered on the screen. The form contains five input fields: "Username", "Email Address", "Password", "Confirm Password", and "Select Role". The "Select Role" field is a dropdown menu with "Admin" selected. At the bottom of the form are two buttons: a red "CREATE USER" button and a white "Cancel" button with a black border. The form is set against a dark blue background with a subtle grid pattern.

Figure 5.19: Add user page UI

In the dashboard page, there is one add user button at the top of the dashboard page. Admins can use this function to add one new user into this web application. Admin is required to fill in credentials for the user, then click on the add user button to add one new account or cancel the action by clicking the cancel button.

## CHAPTER 6

### IMPLEMENTATION

#### 6.1 Introduction

In this chapter will be discussing how the features are implemented in this project. All of the important modules in this project will be showed and explained.

#### 6.2 Implementation of front-end

##### 6.2.1 Account Registration Module

```
export default function Login({ status, canResetPassword }) {  
  const { data, setData, post, processing, errors, reset } = useForm({  
    email: '',  
    password: '',  
    remember: false,  
  });  
  
  const submit = (e) => {  
    e.preventDefault();  
  
    post(route('login'), {  
      onFinish: () => reset('password'),  
    });  
  };  
};
```

Figure 6.1: Code snippet in Login.jsx (1)

The code snippet above manages login form state and handles submission, sending data to the server and clearing the password field after each attempt. The useForm hook is utilized to handle the entire form state and submission process. Three fields were set up which includes email, password, and a remember option, which the remember option can be utilized to develop a remember me option in the future. However, in this project the remember option is not used and is saved for future works.

After that, there is one submit function. Inside the function, by calling the e.preventDefault(), the browser's default submit action will be stopped, allowing the process to be handled entirely with JavaScript. The

`post(route('login'), {...})` call then sends the form data to the server's login route, and once the request is complete, the `onFinish` callback clears the password field with `reset('password')`.

```
<div>
  <InputLabel htmlFor="email" value="Email" className='input-label' />

  <TextInput
    id="email"
    type="email"
    name="email"
    value={data.email}
    className="text"
    autoComplete="email"
    isFocused={true}
    onChange={(e) => setData('email', e.target.value)}
  />

  <InputError message={errors.email} />
</div>
```

Figure 6.2: Code snippet in Login.jsx (2)

This snippet of code showcase a sample field in the login page, an input field for the email address. The `TextInput` component will create an input field which is fully connected to the `useForm` state, the `value={data.email}` keeps the input's value in sync with the `data.email` field managed by `useForm`, `onChange={(e) => setData('email', e.target.value)}` updates that field every time the user types and `InputError` component displays any validation error message stored in `errors.email`, users will immediately see error message if the email is invalid.

Same goes to the password field, this field will have same setup as the email input field as mentioned above.

```

export default function Register() {
  const { data, setData, post, processing, errors, reset } = useForm({
    username: '',
    email: '',
    password: '',
    password_confirmation: '',
    role: '',
    role_code: '',
  });

  const submit = (e) => {
    e.preventDefault();

    post(route('register'), {
      onSuccess: () => {
        router.post(
          route('logout'),
          {},
          {
            preserveScroll: true,
            onSuccess: () => {
              router.visit(route('login'));
            },
          },
        );
      },
      onFinish: () => reset('password', 'password_confirmation'),
    });
  };
};

```

Figure 6.3: Code snippet in Register.jsx (1)

Similar to the Login.jsx, in the register part also uses useForm hook to handle the form state and submission process. The hook sets up a few more fields compared to login, which is username, password\_confirmation, role and role\_code. Upon successful registration of an account, users will be logged out first and visit login page.

```

{
  (data.role === 'lecturer' || data.role === 'admin') && (
    <div className="section">
      <InputLabel
        htmlFor="role_code"
        value={
          data.role === 'lecturer'
            ? 'Lecturer Code'
            : 'Admin Code'
        }
        className='input-label'
      />

      <TextInput
        id="role_code"
        name="role_code"
        type="text"
        value={data.role_code}
        className="text"
        onChange={(e) =>
          setData('role_code', e.target.value)
        }
        required
      />

      <InputError message={errors.role_code} />
    </div>
  )
}

```

Figure 6.4: Code snippet in Register.jsx (2)

The role\_code input field will only be shown when users select to register as Lecturers/Trainers or Admin role. Depending on the role selected, the role code field name will change between lecturer code or admin code. When users key in invalid role code, there will be an error message prompted to provide feedback for users.

Upon filling in all valid credentials during the registration process, users will be able to register for a new account to login afterwards.

```

export default function ForgetPassword({ status }) {
  const { data, setData, post, processing, errors, reset } = useForm({
    email: '',
    password: '',
    password_confirmation: '',
  });

  const submit = (e) => {
    e.preventDefault();
    post(route('forget.password.store'), {
      onSuccess: () => reset('password', 'password_confirmation'),
    });
  };
}

```

Figure 6.5: Code snippet in ForgetPassword.jsx

As usual, the code snippet begins with useForm hook and setup fields for reset password. The field required for a successful reset password process includes email, password and password\_confirmation. The email field is needed so users cannot simply reset any unregistered email's account password. Upon submitting the reset password form, the data will be send to the server's password reset route using the post method.

### 6.2.2 Chatbot Module

```

export default function Chatbot() {
  const { props } = usePage();
  const cleanAssistantText = (s = '') =>
    s.replace(/^[^s*[\.-]]s*$/gm, '').replace(/^[^s*[\.-]]s*$/gm, '').replace(/^[^s*[\.-]]s*$/gm, '').trim();

  // Get the chatId from Inertia props, or fall back to URL parsing
  const getChatIdFromUrl = () => {
    const m = window.location.pathname.match(/\/chats\/(\d+)/);
    return m ? m[1] : 'default';
  };

  const chatId = props?.chat?.id ?? getChatIdFromUrl();
  // Use a per-chat storage key
  const storageKey = `chat_messages_${chatId}`;

  // util
  const sanitize = (arr) =>
    Array.isArray(arr)
      ? arr.filter(m => m && typeof m.text === 'string' && m.text.trim().length > 0)
      : [];

  const [loading, setLoading] = useState(false);

  // Load messages for current chat
  const [messages, setMessages] = useState(() => {
    try {
      const saved = localStorage.getItem(storageKey);
      return saved ? sanitize(JSON.parse(saved)) : [];
    } catch {
      return [];
    }
  });
}

```

Figure 6.6: Code snippet in Chatbot.jsx (1)

```

    return [];
  }
});

const [input, setInput] = useState('');
const bottomRef = useRef(null);

// If the user navigates to a different /chats/{id}, reload that chat's history
useEffect(() => {
  try {
    const saved = localStorage.getItem(storageKey);
    setMessages(saved ? sanitize(JSON.parse(saved)) : []);
  } catch {
    setMessages([]);
  }
}, [storageKey]);

// 5) Save the chat's messages whenever they change
useEffect(() => {
  bottomRef.current?.scrollIntoView({ behavior: 'smooth' });
  try {
    localStorage.setItem(storageKey, JSON.stringify(sanitize(messages)));
  } catch {}
}, [messages, storageKey]);

const toOllamaMessages = (arr) => {
  const msgs = arr.map(m => ({
    role: m.sender === 'user' ? 'user' : 'assistant',
    content: m.text || '',
  }));
});

```

Figure 6.7: Code snippet in Chatbot.jsx (2)

```

if (!msgs.length || msgs[0].role !== 'system') {
  msgs.unshift({
    role: 'system',
    content: 'You are an ISTQB CTFL tutor. Answer concisely; max 3 short examples when helpful.',
  });
}
return msgs.slice(-20);
};

const handleSend = async (e) => {
  e.preventDefault();
  const text = (input || '').trim();
  if (!text || loading) return;

  const next = [...messages, { sender: 'user', text }];
  setMessages(next);
  setInput('');
  setLoading(true);

  try {
    const res = await fetch('/api/llm/chat', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json', 'Accept': 'application/json' },
      body: JSON.stringify({
        messages: toOllamaMessages(next),
        options: {
          temperature: 0.2,
          top_p: 0.9,

```

Figure 6.8: Code snippet in Chatbot.jsx (3)

```

        repeat_penalty: 1.15,
        num_predict: 220,
        num_ctx: 4096,
        seed: 42,
      },
    )},
  });
  if (!res.ok) throw new Error(`${res.status} ${await res.text()}`);

  const data = await res.json();
  const assistantText =
    data.content ??
    data.message?.content ??
    (data.choices?.[0]?.message?.content ?? '');

  setMessages(m => [...m, { sender: 'assistant', text: cleanAssistantText(assistantText) }]);
} catch (err) {
  console.error(err);
  setMessages(m => [...m, { sender: 'assistant', text: 'Error connecting to model' }]);
} finally {
  setLoading(false);
}
};

```

Figure 6.9: Code snippet in Chatbot.jsx (4)

The code snippets from Chatbot.jsx handles the entire client-side logic for the chat page, from loading chat histories, sending new messages and showing the model's generated response.

The component begins by pulling any page properties with `usePage()` and defining a helper named `cleanAssistantText`. This function tidies the model's response by removing unnecessary bullet characters and extra blank lines. Then, the function will check which chat session is active and tries to read a `chat.id` from props. In a case that the `chat.id` isn't available, the URL path will be parsed to extract an ID. That ID is used to create a unique `storageKey` such as `"chat_messages_12"` so each chat has own saved history.

The loading will check whether a message is being sent, messages is responsible for holding the current chat history and is initialized by attempting to read and sanitize any JSON stored under that storage key in `localStorage`. `input` keeps the user's current text entry, and `bottomRef` points to the end of the chat list so the view can scroll there automatically.

The code is continued with two different `useEffect` hooks, both of the hooks are responsible for keeping the conversation history. The first `useEffect`



hook will re-load the saved history whenever the user visit to a different chat, avoiding the risk of losing previous chat histories. Then, a second hook is used to scroll to the bottom and write the updated, sanitized messages back to `localStorage`, preventing the chat disappear available after reloading the page.

Prior sending data to the server, a helper `toOllamaMessages` is defined to converts the local message objects into the format the backend model expects. This tells each message to a role of either user or assistant, set up a system prompt describing the assistant's behavior, and limits the history to the last 20 messages.

Finally, the `handleSend` function executes when the user submits the form. This prevents the default browser submission, trims the input, and, if valid and not already loading, appends the user's message to the state and clears the input box. When the server responds, the assistant's text is cleaned with `cleanAssistantText`, and will be added to the message list. If the request to the model fails, an error message will be send to the user.

### 6.2.3 Star Message module

```
const buildStarSet = (ids) =>
  new Set((ids || []).map(mid => `${chatId}:${String(mid)}`));

// hydrate from server (chat.show returns `starredIds`)
const [starredSet, setStarredSet] = useState(buildStarSet(props.starredIds));

// if user navigates to another chat, refresh the set
useEffect(() => {
  setStarredSet(buildStarSet(props.starredIds));
}, [chatId, props.starredIds]);

// give each message a stable client id if it doesn't have one
const ensureId = (m) => m.id ?? (m._gid ?? (m._gid = crypto.randomUUID()));

const toggleStar = (msg) => {
  const id = ensureId(msg);
  const key = `${chatId}:${id}`;

  // optimistic toggle
  setStarredSet(prev => {
    const next = new Set(prev);
    next.has(key) ? next.delete(key) : next.add(key);
    return next;
  });
};
```

Figure 6.10: Star message code snippet from Chatbot.jsx (1)

```
router.post(
  route('messages.star.toggle', chatId),
  {
    message_id: id,
    role: msg.sender === 'user' ? 'user' : 'assistant',
    text: msg.text || '',
  },
  {
    preserveScroll: true,
    preserveState: true,
    onError: () => {
      setStarredSet(prev => {
        const next = new Set(prev);
        next.has(key) ? next.delete(key) : next.add(key);
        return next;
      });
    },
  }
);
```

Figure 6.11: Star message code snippet from Chatbot.jsx (2)

The star message module will be implemented in both `Chatbot.jsx` and `Starred.jsx`, the code snippet in `Chatbot.jsx` will be discussed first.

Firstly, a helper function `buildStarSet` is defined to accept an array of messages' ID, then will return Set that combines the current `chatId` with a message's ID. This ease the process to check whether a message is starred and ensures each key is unique. The state variable `starredSet` is then initialised, using any `starredIds` sent from the server so the interface immediately reflects which messages are already starred when the page loads.

A `useEffect` hook is implemented to make sure when users navigates to a different chat or update the list of starred messages, the UI will reflects the correct starred messages for the current chat. There is also an `ensureId` function that guarantees every message has a stable identifier. If ID is missing, a temporary id will be created with `crypto.randomUUID()` so that even unsaved messages can be starred.

The `toggleStar` function is defined to handles whenever users clicks the star icon, this will determines the key for the message and performs an optimistic update. This will allow the star button to change the state as the function will automatically add or remove the key from the `starredSet`. To store the starred message, a POST request will be send to the `messages.star.toggle` route. If this fails, the `onError` callback will handle the change, providing the front-end stays the same with back-end data.

```

export default function Starred({ auth, starredMessages }) {

  const { props } = usePage();
  const initial = Array.isArray(props.items) ? props.items : [];
  const [items, setItems] = useState(initial);

  // Keep in sync if the server sends updated items later
  useEffect(() => {
    setItems(Array.isArray(props.items) ? props.items : []);
  }, [props.items]);

  const removeOne = async (id) => {
    // Optimistic UI
    setItems(prev => prev.filter(x => x.id !== id));
    router.delete(route('starred.destroy', id), {
      preserveScroll: true,
      onError: () => {
        // rollback on error
        setItems(Array.isArray(props.items) ? props.items : []);
      },
    });
  };
};

```

Figure 6.12: Code snippet in Starred.jsx

The code snippet above comes from Starred.jsx, which is the page to view all the previously starred messages.

A `usePage()` hook is utilized to pull the props, then an initial array is initialized. In the case where the `props.items` is a valid array, that particular array becomes the starting value. Else will use an empty array. Afterwards, a `useState` will store the list in `items` and provides the `setItems` function to update the list as the user interacts with the page.

Then, in the `useEffect` hook, whenever the `props.items` changes, the effect starts to run and will reset the `items`' state to match the latest data. Ensuring the UI always synchronized, even if the list is changed outside of user actions.

The `removeOne` function handles removing a starred message in the starred message page instead of removing in the chat page. Before removing

anything, the code will update the UI first. Then `router.delete` is called to send a DELETE request to the `starred.destroy` route on the server, passing the starred message's ID to remove the selected starred message. The `onError` callback will restore the list if there are any errors removing the starred message.

### 6.2.4 Upload file module

```
export default function Upload() {
  const { data, setData, post, processing, progress, errors, reset, clearErrors } = useForm({
    file: null,
    message: '',
  });

  const [successMessage, setSuccessMessage] = useState('');
  const fileRef = useRef(null);

  useEffect(() => {
    if (!successMessage) return;
    const id = setTimeout(() => setSuccessMessage(''), 4000);
    return () => clearTimeout(id);
  }, [successMessage]);

  useEffect(() => {
    // clear file-specific error as soon as a new file is chosen
    if (data.file && errors.file) clearErrors('file');
  }, [data.file, errors.file, clearErrors]);
}
```

Figure 6.13: Code snippet in Upload.jsx (1)

```
const submit = (e) => {
  e.preventDefault();
  post(route('lecturer.upload.store'), {
    forceFormData: true,
    preserveScroll: true,
    onSuccess: () => {
      setSuccessMessage('File uploaded successfully!');
      reset('file', 'message');
      if (fileRef.current) fileRef.current.value = '';
    },
  });
};

const handleReset = () => {
  reset('file', 'message');
  if (fileRef.current) fileRef.current.value = '';
};
```

Figure 6.14: Code snippet in Upload.jsx (2)

In the beginning of the Upload.jsx, some fields like file and messages are set up. The use form will manage the form data such as current form values, update form fields, submitting the form and so on. There is also a state prepared for the success message, which is used for storing a temporary success message after a successful upload of file.

The first useEffect is used for clearing the success message shown after a successful upload, the default time set for the message is 4 seconds. After 4 seconds the success message will be removed from the page. The second useEffect is responsible for clearing any file error whenever a new file is selected.

To allow lecturer role users to upload the file, a submit function is defined. The uploaded file will be send to the route lecturer.upload.store, to save the uploaded file in the database. A success message will be shown when the file is uploaded successfully, details were mentioned in above paragraphs. Another handleReset function is defined for allowing users to reset the selected file before upload. Users can either select a new file for upload or can directly click the reset to reset everything.

## 6.3 Implementation of back-end

### 6.3.1 Registration process

```
class RegisteredUserController extends Controller
{
    /**
     * Display the registration view.
     */
    public function create(): Response
    {
        return Inertia::render('Auth/Register');
    }

    /**
     * Handle an incoming registration request.
     *
     * @throws \Illuminate\Validation\ValidationException
     */
    public function store(Request $request): RedirectResponse
    {
        $request->validate([
            'username' => 'required|string|max:255',
            'email' => 'required|string|lowercase|email|max:255|unique:'.User::class,
            'password' => ['required', 'confirmed', Rules\Password::defaults()],
            'role' => ['required'],
            'role_code' => ['nullable', 'string'],
        ]);

        if ($request->role === 'admin' && $request->role_code !== env('ADMIN_CODE')) {
            throw ValidationException::withMessages([
                'role_code' => ['The admin code you entered is incorrect.'],
            ]);
        }
    }
}
```

Figure 6.15: Code snippet in RegisteredUserController.php (1)

```
        if ($request->role === 'lecturer' && $request->role_code !== env('LECTURER_CODE')) {
            throw ValidationException::withMessages([
                'role_code' => ['The lecturer code you entered is incorrect.'],
            ]);
        }

        $user = User::create([
            'username' => $request->username,
            'email' => $request->email,
            'password' => Hash::make($request->password),
            'role' => $request->role,
        ]);

        event(new Registered($user));

        return redirect() -> route('login');
    }
}
```

Figure 6.16: Code snippet in RegisteredUserController.php (2)

The controller above will handle the registration process for new users, allowing users to register a new account.

The create function will create one view for the registration page, displaying the registration form for users to fill in. The second function is store, the function will request credentials from users, such as username, email, password, role and role\_code. The function also checks for role\_code when user selects “Lecturers/ Trainers” or “Admin” role, the default role code is defined in the .env file, when users fill in the role code the function will check with the code defined in the .env. When users fill in an incorrect role code, one error message will be prompted to provide feedback to users.

Upon successful completion of registration process, the credentials of the users will be saved, creating a new user account. Lastly, users are directed back to the login page.

### 6.3.2 Login process

```
class AuthenticatedSessionController extends Controller
{
    /**
     * Display the login view.
     */
    public function create(): Response
    {
        return Inertia::render('Auth/Login', [
            'canResetPassword' => Route::has('password.request'),
            'status' => session('status'),
        ]);
    }

    /**
     * Handle an incoming authentication request.
     */
    public function store(LoginRequest $request): RedirectResponse
    {
        $request->authenticate();

        $request->session()->regenerate();

        $user = Auth::user();

        if ($user->status !== 'active') {
            Auth::logout();

            throw ValidationException::withMessages([
                'email' => 'Your account has been deactivated. You may contact any admins for more information or help.',
            ]);
        }
    }
}
```

Figure 6.17: Code snippet for AuthenticatedSessionController.php (1)



```

    }

    if ($user -> role === 'admin') {
        return redirect()->intended('dashboard');
    } else if (in_array($user->role, ['student', 'lecturer'])) {
        return redirect()->route ('chatbot');
    }
}

/**
 * Destroy an authenticated session.
 */
public function destroy(Request $request): RedirectResponse
{
    Auth::guard('web')->logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

    return redirect('/login');
}
}

```

Figure 6.18: Code snippet for AuthenticatedSessionController.php (2)

The create function in this controller is to show the login page for users, either first time visiting, after registering a new account or reset password. The canResetPassword is responsible for informing the front-end there is a route to reset password, allowing the front-end login page can show the link to access. In addition, the status will pass any status message from the session.

The store function will handle the login from users. This function starts by validating the credentials provided by the users, then logs in the user into the web application. Then, session is regenerated to avoid the risk of session-fixation attacks, and then the function gets the current authenticated user. The function is continued with a condition, to check the account's status of the user. If the account is deactivated, the user will be logged out and can see the error message provided.

Users will be directed to different pages based on the account's role after logging in. Only admin role users will be directed to the dashboard page directly, other role users will be directed to the chat page instead.

### 6.3.3 Reset password process

```
class ForgetPasswordController extends Controller
{
    //
    public function store(Request $request)
    {
        $data = $request->validate([
            'email' => ['required', 'email', 'exists:users,email'],
            'password' => ['required', 'confirmed', 'min:8'],
        ]);

        $user = User::where('email', $data['email'])->firstOrFail();

        $user->forceFill([
            'password' => Hash::make($data['password']),
            'remember_token' => Str::random(60),
        ])->save();

        event(new PasswordReset($user));

        return redirect()->route('login')->with('status', 'Password updated. Please log in.');
```

Figure 6.19: Code snippet in ForgetPasswordController.php

This controller contains only one function, store. To reset password, users have to fill in registered email address and new password. The function will check whether the email address has been registered in the system, if the email is not registered, this function will not allow users to reset password.

Upon filling in the correct email and valid password, the password of the user's account will be updated, triggering the PasswordReset event. After successful reset password process, the users will be directed back to login page, with the success message displayed.

### 6.3.4 Chat page

```
class ChatController extends Controller
{
    //
    public function index()
    {
        $userId = Auth::id();

        $latest = Chat::where('user_id', $userId)
            ->orderByDesc('updated_at')
            ->first();

        if ($latest) {
            return redirect()->route('chats.show', $latest);
        }

        // If user has no chats yet, create one and go there
        $chat = Chat::create([
            'user_id' => $userId,
            'title' => 'New chat',
        ]);

        return redirect()->route('chats.show', $chat);
    }

    public function entry()
    {
        $chat = Chat::where('user_id', Auth::id())
            ->orderByDesc('updated_at')
            ->first() ?? Chat::create(['user_id' => Auth::id(), 'title' => 'New chat']);
    }
}
```

Figure 6.20: Code snippet in ChatController.php (1)

```

    return redirect()->route('chats.show', $chat);
}

public function store(Request $request)
{
    $title = trim((string) $request->input('title', 'New chat'));

    $chat = Chat::create([
        'user_id' => Auth::id(),
        'title'    => $title ?: 'New chat',
    ]);

    return redirect()->route('chats.show', $chat);
}

public function show(Chat $chat)
{
    abort_unless($chat->user_id === Auth::id(), 403);

    // messages in chronological order
    $messages = $chat->messages()
        ->orderBy('created_at')
        ->get(['id', 'role', 'content as text', 'created_at']);

    // sidebar list
    $chats = Chat::where('user_id', Auth::id())
        ->orderByDesc('updated_at')
        ->get(['id', 'title', 'updated_at']);
}

```

Figure 6.21: Code snippet in ChatController.php (2)

```

        $starredIds = StarredMessage::where('user_id', Auth::id())
            ->where('chat_id', $chat->id)
            ->pluck('message_id')
            ->all();

        return Inertia::render('Chat/Chatbot', [
            'chat'      => $chat->only(['id', 'title']),
            'messages' => $messages,
            'chats'     => $chats,
            'starredIds' => $starredIds,
        ]);
    }

    public function update(Request $request, Chat $chat)
    {
        $validated = $request->validate([
            'title' => ['required', 'string', 'max:120'],
        ]);

        $chat->update(['title' => $validated['title']]);

        return back()->with('success', 'Chat renamed.');
```

Figure 6.22: Code snippet in ChatController.php (3)

```

        $chat->delete();
        return back()->with('success', 'Chat deleted.');
```

Figure 6.23: Code snippet in ChatController.php (4)

The ChatController handles the behaviour and actions for the chat page. The index function gets the id of the authenticated users first, then search for the latest chat of the user and directs to the chats.show route, showing users the latest chat page. In a case where the users are newly registered, new users do not have any chat page available yet, the function will create one new chat, with a default “New Chat” title. Then new users will be directed to the latest created

“New Chat” page. This allows every users can directly view the chat page of the web application.

The entry function is almost identical to the index function. The function will search for the newest chat from the authenticated user, then direct the user to that chat page. If there are no chats from the authenticated user, a new chat will be created for the user and direct to that page.

The store function will allow users to create a new chat instead of created automatically by the web application. In the sidebar of the web application, there is a “+” button for users to create a new chat, when user create a new chat, this store function will be called. The new chat will have a default title of “New Chat” as well, and is linked to the authenticated user, so only that user can access to that particular chat page. After creating a new chat, users will be directed to that page immediately.

To allow users to view all the chat lists, the show function is defined. If the chat does not belong to a specific user, the system will abort 403 error. This show function will list of the chat lists in time order, depending on which is the newest, the function also retrieves any starred messages’ id from the user. All data is then passed to the front-end to render the chat interface with messages, the sidebar list of chats, and indicators for starred messages.

The update function will handle the update of chat name. The function will request a new title from the users, up to 120 words max. When users confirm to update the new title, the function will update the title in the database and return a success message.

Lastly, the destroy function allows users to delete a specific chat. This function will check whether the chat belongs to the authenticated users first, if not the system will abort 403 error. After checking, users can delete the chat and is returned with a success message.

### 6.3.5 Star message

```
class MessageController extends Controller
{
    //
    public function starredIndex()
    {
        $items = StarredMessage::where('user_id', Auth::id())
            ->orderByDesc('starred_at')
            ->get(['id', 'chat_id', 'message_id', 'role', 'text', 'starred_at']);

        return Inertia::render('Chat/Starred', [
            'items' => $items,
        ]);
    }

    public function toggleStar(Request $request, Chat $chat)
    {
        abort_unless($chat->user_id === Auth::id(), 403);

        $data = $request->validate([
            'message_id' => ['required', 'string', 'max:191'],
            'role'       => ['required', 'in:user,assistant'],
            'text'       => ['required', 'string'],
        ]);

        $userId = Auth::id();

        $existing = StarredMessage::where([
            'user_id' => $userId,
            'chat_id' => $chat->id,
            'message_id' => $data['message_id'],
```

Figure 6.24: Code snippet in MessageController.php (1)

```

    ]->first();

    if ($existing) {
        $existing->delete();
        return back()->with('starred', false);
    }

    StarredMessage::create([
        'user_id' => $userId,
        'chat_id' => $chat->id,
        'message_id' => $data['message_id'],
        'role' => $data['role'],
        'text' => $data['text'],
        'starred_at' => now(),
    ]);

    return back()->with('starred', true);
}

public function starredDestroy($id)
{
    $item = StarredMessage::where('user_id', Auth::id()->findOrFail($id);
    $item->delete();
    return back();
}
}

```

Figure 6.25: Code snippet in MessageController.php (2)

To implement the star message module, a message controller is created to handle the logics including star or remove starred message. The controller is started with the starredIndex function, this function aids to retrieves all the starred messaged from the authenticated user. The function utilizes the StarredMessage model to find the records where user\_id matches with the authenticated user, then sort the starred message by the latest to the oldest. The columns being chosen includes id, chat\_id, message\_id, role, text and starred\_at. In the end, the function will render the front-end starred message page by passing all the starred messages list as item for displaying in the page.

On top of that, the function toggleStar is defined for allowing users to star a message or remove a star message. The function checks if the chat belongs to the user first as usual, and will abort 403 if the chat does not belongs to the user. Afterwards the function will check the request, whether the request contains the needed fiels such as message\_id, role and text. Then the user's id is being retrieved to validate the status of the message, either the message is starred or not. In a case where users star the same message again, the function



will remove the record of the starred message, which allow users to undo a star message action. Users can star a message if there are no record of that particular message in the database, storing into database and can be used to display in the starred messages page.

The last function `starredDestroy` is utilized to remove a starred message directly by id. The function will search in database to confirm the message is belong to the logged-in user, if not found 404 will be returned. If the message's record is found in the database, the record will be deleted and return back to the page, allowing users to remove the starred message in the starred message page.

### 6.3.6 Upload file

```
class LecturerController extends Controller
{
    //
    public function create()
    {
        return Inertia::render('Lecturer/Upload');
    }

    public function store(Request $request)
    {
        $data = $request->validate([
            'file' => ['required', 'file', 'max:20480', 'mimes:pdf,txt,doc,docx,ppt,pptx,xls,xlsx,csv'],
            'message' => ['nullable', 'string', 'max:2000'],
        ]);

        $file = $data['file'];
        $path = $file->store('uploads/lecturer', 'public');

        $record = File::create([
            'user_id' => $request->user()->id,
            'file_name' => $file->getClientOriginalName(),
            'stored_path' => $path,
            'file_type' => $file->getClientMimeType(),
            'message' => $data['message'] ?? null,
            'size_bytes' => $file->getSize(),
        ]);

        return back()->with('success', 'File uploaded successfully.');
```

Figure 6.26: Code snippet in `LecturerController.php`

In this controller, the `create` function will allow front-end to render a page for upload file page. Generating a page for lecturers and trainers role users to upload the syllabus file.

The `store` function will handle the upload action of a file by user. Firstly, request will be validated, any file is required to upload, and allowing all types

of files to upload. However, the message is optional, users can upload a file without leaving a message for the admin. Then, a path is defined for storing all the uploaded files by the users. Other than defining a path for storing the file, the file will also be stored into the database. Users will receive a success message upon every successful upload of a file.

### 6.3.7 Chatbot

```
class LLMController extends Controller
{
    //
    public function chat(Request $request)
    {
        // Accept either a single prompt OR a messages array
        $data = $request->validate([
            'prompt' => ['nullable', 'string'],
            'messages' => ['nullable', 'array'],
            'messages.*.role' => ['required_with:messages', 'in:system,user,assistant'],
            'messages.*.content' => ['required_with:messages', 'string'],
            'options' => ['nullable', 'array'],
        ]);

        // Build messages if only a prompt was provided
        $messages = $data['messages'] ?? null;
        if (!$messages) {
            $prompt = trim($data['prompt'] ?? '');
            if ($prompt === '') {
                return response()->json(['error' => 'Provide either prompt or messages.'], 422);
            }
            $messages = [
                ['role' => 'system', 'content' => 'You are an ISTQB CTFL tutor. Answer concisely; up to 3 short examples.'],
                ['role' => 'user', 'content' => $prompt],
            ];
        }

        // Ollama endpoint + defaults
        $ollamaBase = rtrim(env('OLLAMA_HOST', 'http://127.0.0.1:11434'), '/');
    }
}
```

Figure 6.27: Code snippet in LLMController.php (1)

```

$payload = [
    'model' => env('OLLAMA_MODEL'),
    'messages' => $messages,
    'stream' => false, // set true later for future streaming
    'options' => array_merge([
        'temperature' => 0.2,
        'top_p' => 0.8,
        'repeat_penalty' => 1.2,
        'num_predict' => 120, // hard cap: 80-100 words
        'num_ctx' => 4096,
        'seed' => 42,
    ], $data['options'] ?? []),
];

try {
    $resp = Http::timeout(180)->post($ollamaBase.'/api/chat', $payload);

    if ($resp->failed()) {
        \Log::warning('Ollama failed', ['status' => $resp->status(), 'body' => $resp->body()]);
        return response($resp->body(), $resp->status())
            ->header('Content-Type', 'application/json');
    }

    $json = $resp->json();
    return response()->json([
        'role' => 'assistant',
        'content' => data_get($json, 'message.content', ''),
        'raw' => $json,
    ]);
}

```

Figure 6.28: Code snippet in LLMController.php (2)

```

    });
} catch (\Throwable $e) {
    \Log::error('Ollama call exception', ['e' => $e->getMessage()]);
    return response()->json(['error' => 'Connection error to Ollama'], 502);
}

private function messagesToPrompt(array $messages): string
{
    $parts = [];
    foreach ($messages as $m) {
        $role = $m['role'] ?? 'user';
        $content = $m['content'] ?? '';
        if ($role === 'system') $parts[] = "<|system|>\n{$content}";
        elseif ($role === 'assistant') $parts[] = "<|assistant|>\n{$content}";
        else $parts[] = "<|user|>\n{$content}";
    }
    $parts[] = "<|assistant|>\n";
    return implode("\n\n", $parts);
}

```

Figure 6.29: Code snippet in LLMController.php (3)

This controller is crucial as this controls how the LLM in the web application behaves. The chat function will validate the request from users first, to allow users to ask a prompt or a structured message array, each message need to include a role and text content field. If an empty message is send, the system will return an error response to the user. One instruction is defined to the model

to indicate what is the model's role and what the model should do, and a user message with the prompt is also defined. This allows the LLM to perform better and conduct a nice conversation with the user.

To make the LLM respond to the user's prompt, the function defined a POST request to Ollama server, read the URL and the model name from the .env file. The \$payload part will combine the conversation messages, then define a message generation settings, such as:

- temperature controls the randomness of the message, low temperature reduces randomness.
- top\_p considers the probability mass for upcoming tokens, to balance coherence.
- repeat\_penalty reduces repeating in generating messages.
- num\_predict controls how much the tokens are to generate.
- num\_ctx is the context window size in tokens, this can limit how much text the models think.
- seed fix one random seed for reproducible outputs.

Then, the request will be send to the endpoint with 3 minutes timeout, if there are any failure on the request, a warning message and the status code will be returned. Upon success request, the function parses the JSON and return with a simple JSON response, which includes the chatbot's text, raw Ollama response. In a case where there are any exception, the function will return 502 error and an error message.

In the end, the last function messagesToPrompt is responsible to display the conversation array into a single string prompt if needed. The function will iterate through every message, put a sign based on the message role, then tag one assistant tag for the LLM to generate response to the users.

## **CHAPTER 7**

### **TESTING**

#### **7.1 Introduction**

In this chapter, several testing is conducted after the project is completed with development. Testing is needed to ensure this project successfully implemented the requirements stated in the previous chapters and to fix the bugs and error discovered during testing. 4 types of tests were conducted following the V-Model methodology, including unit testing, integration testing, system testing and User-Acceptance Testing (UAT).

## 7.2 Unit Testing

Unit testing is carried out to validate each module in this project is working properly, without bugs that would sabotage the usage. Test cases were designed to validate each module can work without any issue. For unit tests will only cover necessary and vital modules, other modules such as profile page will not be tested.

Table 7.1: Account Registration Module Test Cases

Project Title:					A chatbot for teaching software testing CTFL Syllabus			
Module:					Account Registration Module			
Test Designed Date:					20/8/2025			
Test Executed Date:					22/8/2025			
Test Designed By:					Teh Wing Fung			
Test Executed By:					Teh Wing Fung			
Test Case #	Test Scenario	Test Case	Procedure	Test Data	Expected Results	Post-Conditions	Actual Results	Status

UT-001	To test the registration functionality	Register a valid Student/Software Testers role account	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: student 2. Email: student@gmail.com 3. Sign up role: Students/Software Testers 4. Password: 1234abcd 5. Confirm Password: 1234abcd	A Student/Software Testers role account will be created.	Successfully registered an account and will be directed to login page.	An account is registered, and user is directed to login page.	PASS
UT-002	To test the registration functionality	Register a Student/Software Testers role account without filling in username	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: null 2. Email: student2@gmail.com 3. Sign up role: Students/Software Testers 4. Password: 1234abcd 5. Confirm Password: 1234abcd	Sign up process will fail. An error message will show that users need to fill in the username field.	The registration failed, users cannot create an account.	No account is registered. Username field has an error message requesting users to fill	PASS

							in a username.	
UT-003	To test the registration functionality	Register a Student/ Software Testers role account without filling in valid email address	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: user3 2. Email: not an email 3. Sign up role: Students/ Software Testers 4. Password: 1234abcd 5. Confirm Password: 1234abcd	Sign up process will fail. An error message will show that users need to fill in the email field.	The registration failed, users cannot create an account.	No account is registered. Email field indicates users need to add "@" symbol in the email address.	PASS



UT-004	To test the registration functionality	Register a Student/ Software Testers role account without filling in valid password format	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: user4 2. Email: user4@gmail.com 3. Sign up role: Students/ Software Testers 4. Password: 1234 5. Confirm Password: 1234	Sign up process will fail. An error message will show that users need to fill in the password field.	The registration failed, users cannot create an account.	No account is registered. An error message is shown: "The password field must be at least 8 characters."	PASS
UT-005	To test the registration functionality	Register a Student/ Software Testers role account without	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: user5 2. Email: user5@gmail.com 3. Sign up role: Students/ Software Testers 4. Password: 1234abcd	Sign up process will fail. An error message will show that users need to fill in the password field.	The registration failed, users cannot create an account.	No account is registered. An error message is shown: "The password field	PASS

		filling in matching passwords		5. Confirm Password: 1234			confirmation does not match.”	
UT-006	To test the registration functionality	Register a valid Lecturers/ Trainers role account.	1. Click on the ‘Sign Up’ link on the login page 2. Fill in the test data	1. Username: lecturer 2. Email: lecturer@gmail.com 3. Sign up role: Lecturers/ Trainers 4. Lecturer code: lecturer246 5. Password: 1234abcd 6. Confirm Password: 1234abcd	Sign up process for lecturer/ trainers role account will be successful.	Successfully registered an account and will be directed to login page.	An account is registered, and user is directed to login page.	PASS

UT-007	To test the registration functionality	Register a Lecturers/ Trainers role account without filling in valid lecturer code.	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: lecturer2 2. Email: lecturer2@gmail.com 3. Sign up role: Lecturers/ Trainers 4. Lecturer code: lecturer123 5. Password: 1234abcd 6. Confirm Password: 1234abcd	The sign-up process will fail and will have error message in the lecturer code field.	The registration failed, users cannot create an account.	No account is registered. An error message in the lecturer code field is shown: "The lecturer code you entered is incorrect."	PASS
UT-008	To test the registration functionality	Register a Lecturers/ Trainers role account without filling in	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: lecturer3 2. Email: lecturer3@gmail.com 3. Sign up role: Lecturers/ Trainers 4. Lecturer code: null 5. Password: 1234abcd 6. Confirm Password:	The sign-up process will fail and will have an error message that ask to fill in the lecturer code field.	The registration failed, users cannot create an account.	No account is registered. Users are requested to fill in a lecturer code.	PASS

		lecturer code.		1234abcd				
UT-009	To test the registration functionality	Register a Lecturers/ Trainers role account without filling in username	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: null 2. Email: lecturer4@gmail.com 3. Sign up role: Lecturers/ Trainers 4. Lecturer code: lecturer246 5. Password: 1234abcd 6. Confirm Password: 1234abcd	Sign up process will fail. An error message will show that users need to fill in the username field.	The registration failed, users cannot create an account.	No account is registered. Username field has an error message requesting users to fill in a username.	PASS

UT-010	To test the registration functionality	Register a Lecturers/ Trainers role account without filling in valid email address	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: lecturer5 2. Email: not an email 3. Sign up role: Lecturers/ Trainers 4. Lecturer code: lecturer246 5. Password: 1234abcd 6. Confirm Password: 1234abcd	Sign up process will fail. An error message will show that users need to fill in the email field.	The registration failed, users cannot create an account.	No account is registered. Email field indicates users need to add "@" symbol in the email address.	PASS
UT-011	To test the registration functionality	Register a Lecturers/ Trainers role account without filling in valid	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: lecturer6 2. Email: lecturer6@gmail.com 3. Sign up role: Lecturers/ Trainers 4. Lecturer code: lecturer246 5. Password: 1234	Sign up process will fail. An error message will show that users need to fill in the password field.	The registration failed, users cannot create an account.	No account is registered. An error message is shown: "The password field must be	PASS

		password format		6. Confirm Password: 1234			at least 8 characters.”	
UT-012	To test the registration functionality	Register a Lecturers/ Trainers role account without filling in matching passwords	1. Click on the ‘Sign Up’ link on the login page 2. Fill in the test data	1. Username: lecturer7 2. Email: lecturer7@gmail.com 3. Sign up role: Lecturers/ Trainers 4. Lecturer code: lecturer246 5. Password: 1234abcd 6. Confirm Password: 1234	Sign up process will fail. An error message will show that users need to fill in the password field.	The registration failed, users cannot create an account.	No account is registered. An error message is shown: “The password field confirmation does not match.”	PASS

UT-013	To test the registration functionality	Register a valid Admin role account.	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: Admin 2. Email: admin1@gmail.com 3. Sign up role: Admin 4. Lecturer code: admin123 5. Password: 1234abcd 6. Confirm Password: 1234abcd	Sign up process for admin role account will be successful.	Successfully registered an admin role account and will be directed to login page.	An admin role account is registered, and user is directed to login page.	PASS
UT-014	To test the registration functionality	Register an Admin role account without filling in valid admin code.	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: Admin2 2. Email: admin2@gmail.com 3. Sign up role: Admin 4. Admin code: admin220 5. Password: 1234abcd 6. Confirm Password: 1234abcd	The sign-up process will fail and will have error message in the admin code field.	The registration failed, users cannot create an account.	No account is registered. An error message in the lecturer code field is shown: "The admin code	PASS

							you entered is incorrect.”	
UT-015	To test the registration functionality	Register an Admin role account without filling in admin code.	1. Click on the ‘Sign Up’ link on the login page 2. Fill in the test data	1. Username: Admin3 2. Email: admin3@gmail.com 3. Sign up role: Admin 4. Admin code: null 5. Password: 1234abcd 6. Confirm Password: 1234abcd	The sign-up process will fail and will have an error message that ask to fill in the admin code field.	The registration failed, users cannot create an account.	No account is registered. Users are requested to fill in an admin code.	PASS
UT-016	To test the registration functionality	Register an Admin role account without filling in username	1. Click on the ‘Sign Up’ link on the login page 2. Fill in the test data	1. Username: null 2. Email: Admin4@gmail.com 3. Sign up role: Admin 4. Admin code: admin123 5. Password: 1234abcd	Sign up process will fail. An error message will show that users need to fill in the username field.	The registration failed, users cannot create an account.	No account is registered. Username field has an error message requesting	PASS



				6. Confirm Password: 1234abcd			users to fill in a username.	
UT-017	To test the registration functionality	Register an Admin role account without filling in valid email address	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: Admin5 2. Email: not an email 3. Sign up role: Admin 4. Admin code: admin123 5. Password: 1234abcd 6. Confirm Password: 1234abcd	Sign up process will fail. An error message will show that users need to fill in the email field.	The registration failed, users cannot create an account.	No account is registered. Email field indicates users need to add "@" symbol in the email address.	PASS

UT-018	To test the registration functionality	Register an Admin role account without filling in valid password format	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: Admin6 2. Email: admin6@gmail.com 3. Sign up role: Admin 4. Admin code: admin123 5. Password: 1234 6. Confirm Password: 1234	Sign up process will fail. An error message will show that users need to fill in the password field.	The registration failed, users cannot create an account.	No account is registered. An error message is shown: "The password field must be at least 8 characters."	PASS
UT-019	To test the registration functionality	Register an Admin role account without filling in matching passwords	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: Admin7 2. Email: admin7@gmail.com 3. Sign up role: Admin 4. Admin code: admin123 5. Password: 1234abcd 6. Confirm Password: 1234	Sign up process will fail. An error message will show that users need to fill in the password field.	The registration failed, users cannot create an account.	No account is registered. An error message is shown: "The password field confirmation	PASS

							does not match.”	
UT-020	To test the registration functionality	Register an account without selecting a role	1. Click on the ‘Sign Up’ link on the login page 2. Fill in the test data	1. Username: user6 2. Email: user6@gmail.com 3. Sign up role: null 4. Password: 1234abcd 5. Confirm Password: 1234abcd	Sign up process will fail. An error message will show that users need to select a role.	The registration failed, users cannot create an account.	No account is registered. Users are required to select a role in the “Sign up as” field.	PASS
UT-021	To test the login functionality	Login into Students/ Software Testers account using correct credentials	1. Fill in the email and password field using the test data	1. Email: student@gmail.com 2. Password: 1234abcd	Successful login into the web application using correct credentials.	Users are logged in into the web application and are shown with the chat page.	Users can view the chat page after the login.	PASS

UT-022	To test the login functionality	Login into Students/ Software Testers account using incorrect email	1. Fill in the email and password field using the test data	1. Email: false@gmail.com 2. Password: 1234abcd	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login, an error message is shown: "These credentials do not match our records."	PASS
UT-023	To test the login functionality	Login into Students/ Software Testers account using incorrect password	1. Fill in the email and password field using the test data	1. Email: student@gmail.com 2. Password: 1234ab	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login, an error message is shown: "These credentials do not match our records."	PASS

UT-024	To test the login functionality	Login into Lecturers/ Trainers account using correct credentials	1. Fill in the email and password field using the test data	1. Email: lecturer@gmail.com 2. Password: 1234abcd	Successful login into the web application using correct credentials.	Users are logged in into the web application and are shown with the chat page.	Users can view the chat page after the login.	PASS
UT-025	To test the login functionality	Login into Lecturers/ Trainers account using incorrect email	1. Fill in the email and password field using the test data	1. Email: lectur@gmail.com 2. Password: 1234abcd	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login, an error message is shown: "These credentials do not match our records."	PASS

UT-026	To test the login functionality	Login into Lecturers/ Trainers account using incorrect password	1. Fill in the email and password field using the test data	1. Email: lecturer@gmail.com 2. Password: 1234ab	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login, an error message is shown: "These credentials do not match our records."	PASS
UT-027	To test the login functionality	Login into Admin account using correct credentials	1. Fill in the email and password field using the test data	1. Email: admin1@gmail.com 2. Password: 1234abcd	Successful login into the web application using correct credentials.	Users are logged in into the web application and are shown with the chat page.	Users can view the chat page after the login.	PASS

UT-028	To test the login functionality	Login into Admin account using incorrect email	1. Fill in the email and password field using the test data	1. Email: adm@gmail.com 2. Password: 1234abcd	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login, an error message is shown: "These credentials do not match our records."	PASS
UT-029	To test the login functionality	Login into Admin account using incorrect password	1. Fill in the email and password field using the test data	1. Email: admin1@gmail.com 2. Password: 1234ab	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login, an error message is shown: "These credentials do not match our records."	PASS

UT-030	To test the login functionality	Login without fill in the email address field	1. Leave email field blank. 2. Fill in password field	1. Email: null 2. Password: 1234abcd	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login and shown with: "The email field is required."	PASS
UT-031	To test the login functionality	Login without filling password field	1. Fill in an email address. 2. Leave the password field blank	1. Email: student@gmail.com 2. Password: null	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login and shown with: "The password field is required."	PASS



UT-032	To test the login functionality	Login without filling any field	1. Leave both email and password field blank	1. Email: null 2. Password: null	Login process will fail.	Cannot login into the web application, an error message is shown.	Users cannot login and shown with: "The email field is required." and "The password field is required."	PASS
UT-033	To test the login functionality	Login using a deactivated account	1. Use a deactivated email address and fill in correct password	1. Email: sign@gmail.com 2. Password: 1234abcd	Login process will fail.	Cannot login into the web application, an error message will be shown	Users cannot login and shown with: "Your account has been deactivated. You may	PASS

							contact any admins for more information or help.”	
UT-034	To test the reset password functionality	Reset password with valid and matching passwords	1. Click on the ‘Reset here’ link 2. Fill in registered email address 3. Fill in matching and valid passwords	1. Email: student123@gmail.com 2. New password: 1234abcde 3. Confirm new password: 1234abcde	Users will be able to reset the password.	Users will visit back login page.	Users can reset password and directed to login page with “Password updated. Please log in.”	PASS

UT-035	To test the reset password and login functionality	After reset password, users will login using new password	1. Log in by filling in the new password in the password field.	1. Email: student123@gmail.com 2. Password: 1234abcde	Users will be able to log in using new password.	Users are logged in into the web application and are shown with the chat page.	Users can view the chat page after the login.	PASS
UT-036	To test the reset password functionality	Reset password without filling email address	1. Click on the 'Reset here' link 2. Leave email field blank 3. Fill in matching and valid passwords	1. Email: null 2. New password: 1234abcde 3. Confirm new password: 1234abcde	Reset password action will fail.	No password will be reset; a message will request users to fill in email field.	Users cannot reset password, the system prompt "Please fill out this field" in the email field.	PASS

UT-037	To test the reset password functionality	Reset password without filling password	1. Click on the 'Reset here' link 2. Fill in email field 3. Leave password fields blank	1. Email: student123@gmail.com 2. New password: null 3. Confirm new password: null	Reset password action will fail.	No password will be reset; a message will request users to fill in password field.	Users cannot reset password, the system prompt "Please fill out this field" in the password field.	PASS
UT-038	To test the reset password functionality	Reset password without filling confirm new password	1. Click on the 'Reset here' link 2. Fill in email and new password field	1. Email: student123@gmail.com 2. New password: 1234abcd 3. Confirm new password: null	Reset password action will fail.	No password will be reset; a message will request users to fill in confirm new	Users cannot reset password, the system prompt "Please fill out this field" in the	PASS

			3. Leave confirm new password field blank			password field.	confirm new password field.	
UT-039	To test the reset password functionality	Reset password without filling matching passwords	1. Click on the 'Reset here' link 2. Fill in email and new password field 3. Fill in mismatch password in	1. Email: student123@gmail.com 2. New password: 1234abcd 3. Confirm new password: 1234abcde	Reset password action will fail.	No password will be reset; an error message will be shown	Users cannot reset password, the system prompt "The password field confirmation does not match." in the	PASS

			confirm new password field				password field.	
UT-040	To test the reset password functionality	Reset password without filling in the registered email address	1. Click on the 'Reset here' link 2. Fill in unregistered email address 3. Fill in matching and valid passwords	1. Email: student12@gmail.com 2. New password: 1234abcd 3. Confirm new password: 1234abcd	Reset password action will fail.	No password will be reset; an error message will be shown.	Users cannot reset password, the system prompt "The selected email is invalid." in the email field.	PASS

UT-041	To test the reset password functionality	Reset password without filling any field	1. Click on the 'Reset here' link 2. Leave every field blank and click reset password button	1. Email: null 2. New password: null 3. Confirm new password: null	Reset password action will fail.	No password will be reset; an error message will be shown.	Users cannot reset password; the system prompt a message to fill in the email field.	PASS
UT-042	To test the registration functionality	Register an account with registered email address	1. Click on the 'Sign Up' link on the login page 2. Fill in the test data	1. Username: test same 2. Email: student@gmail.com 3. Sign up role: Students/ Software Testers 4. Password: 1234abcd 5. Confirm Password:	No account will be created	An error message will be prompted to indicate user that the email is not available	No account is registered, there is an error message: "The email has already been taken."	PASS

				1234abcd				
--	--	--	--	----------	--	--	--	--



Table 7.2: Chatbot Module Test Cases

Project Title:					A chatbot for teaching software testing CTFL Syllabus			
Module:					Chatbot			
Test Designed Date:					20/8/2025			
Test Executed Date:					22/8/2025			
Test Designed By:					Teh Wing Fung			
Test Executed By:					Teh Wing Fung			
Test Case #	Test Scenario	Test Case	Procedure	Test Data	Expected Results	Post-Conditions	Actual Results	Status
UT-043	To test the functionality of the chatbot of the web application	Ask the chatbot about who the chatbot is	1. Login using any registered account 2. Ask the chatbot who the chatbot is	Prompt: “Who are you”	The chatbot should take some time to generate a valid response.	After some time, the chatbot should respond with who chatbot is and what is	The chatbot responded with: “I’m a qualified ISTQB (International Software Testing Qualifications Board) Certified	PASS

						the chatbot's ability.	Test Manager and Tutor, specializing in teaching software testing methodologies, frameworks, and best practices. I help individuals prepare for the ISTQB CTFL (Certified Tester Foundation Level) certification exam."	
--	--	--	--	--	--	------------------------	---	--

UT-044	To test the functionality of the chatbot of the web application	Ask the chatbot questions about CTFL syllabus	1. Login using any registered account 2. Ask one prompt which is related to CTFL syllabus	Prompt: “Why is Testing Necessary?”	The chatbot should take some time to generate a valid response.	After some time, the chatbot should generate a valid response.	The chatbot responded with: “Testing is necessary because it helps achieve the agreed-upon goals within the set scope, time, quality, and budget constraints (1.2). Testing also contributes to success by providing a cost-effective means of detecting defects and indirectly contributing to	PASS
--------	---	---	--	-------------------------------------	---	--	---	------

							higher quality test objects (1.2.1).”	
UT-045	To test the functionality of the chatbot of the web application	Ask the chatbot questions that are out of CTFL syllabus	1. Login using any registered account 2. Ask one prompt which is not related to	Prompt: “what is a song”	The chatbot should take some time to handle the prompt.	After some time, the chatbot should provide a response indicating the question is not related	The chatbot replied with: “A simple question! In the context of ISTQB CTFL, tissues (or facial tissues) aren't directly related to software testing. However, if I had to make a connection:	PASS

			CTFL syllabus			to CTFL syllabus.	<ul style="list-style-type: none"><li>• <b>Error Handling:</b> Just as we blow our nose with a tissue after making an error in code, a well-designed error handling mechanism can help mitigate issues and prevent errors from propagating.</li><li>• <b>Test Case Failure:</b> When a test case fails, it's like blowing your nose -</li></ul>	
--	--	--	------------------	--	--	----------------------	---	--

							<p>you need to clean up the mess (i.e., identify and fix the issue) before moving on. Please keep in mind these examples are quite weak connections!</p> <p>”</p>	
--	--	--	--	--	--	--	---	--

Table 7.3: Quiz Module Test Cases

Project Title:					A chatbot for teaching software testing CTFL Syllabus			
Module:					Quiz Module			
Test Designed Date:					21/8/2025			
Test Executed Date:					23/8/2025			
Test Designed By:					Teh Wing Fung			
Test Executed By:					Teh Wing Fung			
Test Case #	Test Scenario	Test Case	Procedure	Test Data	Expected Results	Post-Conditions	Actual Results	Status
UT-046	To test the chatbot's ability to generate a quiz	Ask the chatbot to generate a quiz from the syllabus	1. Login using any registered account 2. Ask the chatbot to generate a quiz	Prompt: "can you ask me a question to test me?"	The chatbot should take some time to generate a valid response.	After some time, the chatbot should generate a question.	The chatbot responded with: "Here's your first question: What is the primary purpose of the	PASS

							<p>Test Environment Management (TEM) process in the ISTQB CTFL?</p> <p>A) To define and manage test environments</p> <p>B) To create and maintain test data</p> <p>C) To identify and report defects found during testing</p> <p>D) To develop and</p>	
--	--	--	--	--	--	--	--	--



							execute test scripts Please choose your answer. ”	
UT-047	To test the chatbot's ability to respond to user's answer after the quiz	Respond with an answer to the previous question	1. Ask the chatbot for a question 2. Answer the question provided by the chatbot	Question: “Here's your first question: What is the primary purpose of the Test Environment Management (TEM) process in the ISTQB CTFL? A) To define and manage test	The chatbot should take some time to generate a valid response.	After some time, the chatbot should generate a response.	The chatbot replied: “That's incorrect, I'm afraid. The correct answer is A) To define and manage test environments. TEM is a	PASS

				<p>environments B)          To create and maintain test data          C) To identify and report defects found during testing D) To develop and execute test scripts          Please choose your answer.          ”          Prompt: “is the answer D?”</p>			<p>process that ensures the environment in which tests are performed meets the required standards and configurations.          Here's another question:          What type of testing involves executing the same sequence of steps multiple times</p>	
--	--	--	--	--	--	--	--	--

							<p>to identify any variations or differences?</p> <p>A) Equivalence partitioning B) State transition testing C) Boundary value analysis D) Repetitive testing</p> <p>Please choose your answer.</p> <p>”</p>	
--	--	--	--	--	--	--	--	--

UT-048	To test the chatbot's ability to respond to user's answer after the quiz	Respond to the previous question with another prompt that is not relevant to the question	1. Ask the chatbot for a question 2. Answer with other things	Question: "What type of testing involves executing the same sequence of steps multiple times to identify any variations or differences? A) Equivalence partitioning B) State transition testing C) Boundary value analysis D) Repetitive testing Please choose your answer.	The chatbot should take some time to generate a valid response.	After some time, the chatbot should inform the user that the answer is not related to the question.	The chatbot replied with: "That's not relevant to our ISTQB CTFL study session! Let's get back on track. You didn't answer the previous question, so I'll give you another chance. What type of testing involves executing the	PASS
--------	--	---	--	---	---	---	--	------

				” Prompt: “what is a song”			same sequence of steps multiple times to identify any variations or differences? A) Equivalence partitioning B) State transition testing C) Boundary value analysis D) Repetitive testing Please choose your answer. ”	
--	--	--	--	----------------------------------	--	--	---	--

Table 7.4: Star Message Module Test Cases

Project Title:					A chatbot for teaching software testing CTFL Syllabus			
Module:					Star Message Module			
Test Designed Date:					21/8/2025			
Test Executed Date:					23/8/2025			
Test Designed By:					Teh Wing Fung			
Test Executed By:					Teh Wing Fung			
Test Case #	Test Scenario	Test Case	Procedure	Test Data	Expected Results	Post-Conditions	Actual Results	Status
UT-049	To test the star message functionality	Click the “star” button below the message box of the chatbot	1. Ask the chatbot a prompt 2. Wait for the response from chatbot	The response from the chatbot	The star button under the message box should turn yellow.	The message will be successfully starred.	The “star” button turned yellow, indicating the message	PASS

			3. Star the message from the chatbot				has been starred.	
UT-050	To test the star message functionality	View the starred message after star	1. Star a message from the chatbot 2. Open sidebar and select “Starred Messages” 3. View the starred	Starred messages in the starred messages page	The starred message page should store the starred message.	Users can view the starred message on that page.	The starred message is displayed on the starred message page and can be	PASS

			message in that page				viewed clearly.	
UT-051	To test the star message functionality	Visit the starred message by clicking the “Go to message” button	1. Select one starred message 2. Click “Go to message” button	Any starred messages	There should be a “Go to message” button within all the starred messages.	Users will be directed to the starred message after clicking the button.	Users are directed to the starred message after clicking the “Go to message” button.	PASS



UT-052	To test the star message functionality	Remove the starred message in the chat	<ol style="list-style-type: none"> <li>1. Select one starred message</li> <li>2. Click on the “star” button</li> <li>3. After that visit starred message page to check whether the message is removed</li> </ol>	Any starred messages in the chat page	The star button should change back to the original color gray after removing as starred message.	The message will not be starred anymore and cannot be viewed in starred message page.	The “star” button become gray, and the previous starred message is not in starred message page.	PASS
--------	--	--	--	---------------------------------------	--	---	---	------

UT-053	To test the star message functionality	Remove the starred message in the starred message page	<ol style="list-style-type: none"> <li>1. Select one starred message in the page</li> <li>2. Click the “remove” button</li> <li>3. Return to the chat page to view the result</li> </ol>	Any starred messages in the starred message page	The message should be removed	In the chat page, the “star” button will become back to gray color.	The message is removed in the page, in the chat page the “star” button is not yellow anymore.	PASS
--------	--	--	--	--	-------------------------------	---	---	------

Table 7.5: Upload file Module Test Cases

Project Title:					A chatbot for teaching software testing CTFL Syllabus			
Module:					Upload file			
Test Designed Date:					21/8/2025			
Test Executed Date:					23/8/2025			
Test Designed By:					Teh Wing Fung			
Test Executed By:					Teh Wing Fung			
Test Case #	Test Scenario	Test Case	Procedure	Test Data	Expected Results	Post-Conditions	Actual Results	Status
UT-054	To test the upload file functionality	Upload a file in the upload file page	1. Login into a lecturer role account 2. Visit the upload	Any file	The page should provide a button for users to add a file for upload.	A success message will be prompted after a successful upload.	File is uploaded and there is a success message: "File uploaded successfully!"	PASS

			file page 3. Click on the upload file and select a file for upload				”	
UT-055	To test the upload file functionality	Upload a file and add a message	1. Login into a lecturer role account 2. Visit the upload file page 3. Click	Message: “try upload”	The file and the message should be able to upload.	A success message will be prompted after a successful upload.	File is uploaded along with the message and there is a success message: “File uploaded successfully!”	PASS

			on the upload file and select a file for upload 4. Add a message in the box provided				”	
UT-056	To test the upload file functionality	Cannot click the “upload” button if no file is attached	1. Login into a lecturer role account 2. Visit the upload	Null	The upload button should be in a state where users cannot click.	The button cannot be clicked and no files is uploaded.	The “upload” button does not allow users to click since there are no	PASS

			file page 3. Try to click “upload” button without selecting any file for upload				attached files.	
UT-057	To test the upload file functionality	User cannot upload any files even added message	1. Login into a lecturer role account 2. Visit the upload	Message: “try upload”	The upload button should be in a state where users cannot click.	The button cannot be clicked and no files is uploaded.	The “upload” button does not allow users to click since there are no attached	PASS

		in the box provided	file page 3. Leave a message in the box 4. Try to click “upload” button without selecting any file for upload				files, even there are message attached in the box.	
--	--	------------------------	---	--	--	--	--	--

UT-058	To test the upload file functionality	Users click on the reset after selecting a file	<ol style="list-style-type: none"> <li>1. Login into a lecturer role account</li> <li>2. Visit the upload file page</li> <li>3. Select a file from device</li> <li>4. Click on the “Reset” button</li> </ol>	Any file	After reset, users cannot click the upload button.	The button cannot be clicked after clicking reset.	Before reset, the upload button can be clicked. After reset, the upload button cannot be clicked.	PASS
--------	---------------------------------------	---	--	----------	--	--	---	------



UT-059	To test the upload file functionality	Users click on the reset after adding a message	1. Login into a lecturer role 2. Visit the upload file page 3. Add a message in the box 4. Click on the "Reset" button	Message: "test reset"	Before reset, the message should be able to be viewed.	After reset, the message should disappear.	Before reset, the message is in the box. After reset, the message is gone.	PASS
--------	---------------------------------------	---	---	-----------------------	--	--	--	------

UT-060	To test the upload file functionality	View the uploaded files list on the dashboard page.	1. Login into an admin role account 2. Visit the upload file section	Uploaded files	Admins can view all the uploaded files and messages.	Admins can choose different sets of manage actions.	After logging in, admin users can see all the uploaded files and messages in the dashboard page. Admin can also view who is the uploader, have different manage actions.	PASS
--------	---------------------------------------	---	---	----------------	--	---	--	------

UT-061	To test the upload file functionality	View the uploaded files	<ol style="list-style-type: none"> <li>1. Login into an admin role account</li> <li>2. Visit the upload file section</li> <li>3. Click on the name of the file</li> </ol>	Any uploaded files	Admins should be able to view the content of the file.	Admins is directed to new tab to view the file.	After clicking the hyperlink (file name), admin users can view the content of the file.	PASS
--------	---------------------------------------	-------------------------	---	--------------------	--	---	---	------

UT-062	To test the upload file functionality	Download the uploaded files	<ol style="list-style-type: none"> <li>1. Login into an admin role account</li> <li>2. Visit the upload file section</li> <li>3. Click on the download button</li> </ol>	Any uploaded files	Admins should be asked whether to download the file.	Admins can select to download or do not download.	After clicking the download button, admin users are asked to confirm the download the file.	PASS
--------	---------------------------------------	-----------------------------	--	--------------------	--	---	---	------

UT-063	To test the upload file functionality	Delete the uploaded files	<ol style="list-style-type: none"> <li>1. Login into an admin role account</li> <li>2. Visit the upload file section</li> <li>3. Click on the delete button</li> </ol>	Any uploaded files	Admins should be asked whether to delete the file.	Admins can select to delete or do not delete.	After clicking the download button, admin users are asked to confirm the deletion the file.	PASS
--------	---------------------------------------	---------------------------	--	--------------------	--	---	---	------

### **7.3 Integration Testing**

In this part, integration testing was carried out to verify each of the modules can work with each other without any issue. Certain test cases were designed and tested to validate the modules work properly with other modules.

Table 7.6: Integration testing test cases

Test Case ID	Test Case Description	Test Steps	Test Data	Status
IT-001	To determine if users can chat with the chatbot after logging in	1. Login using a registered account. 2. Send a prompt to the chatbot. 3. Wait for chatbot's response.	1. Email: 2104658@gmail.com Password: 1234abcd 2. Prompt: "Define test objective"	PASS
IT-002	To determine if users can star a message sent by the chatbot	1. Send a prompt to the chatbot. 2. Wait for chatbot's response. 3. Star a response from the chatbot.	Response generated by the chatbot	PASS
IT-003	To determine if registered users can be showed in the admin dashboard	1. Register one new account. 2. Login to an admin role account. 3. Verify the existence of the newly registered account.	1. Email: test@gmail.com 2. Dashboard registered users list	PASS
IT-004	To determine if uploaded files can be showed in the admin dashboard	1. Login into one lecturer role account. 2. Upload one file in the upload file page. 2. Login to an admin role account. 3. Verify the existence of the newly uploaded file.	Any uploaded files and Dashboard uploaded files list	PASS

IT-005	To determine if admins can download or delete an uploaded file	<ol style="list-style-type: none"> <li>1. Login into one admin account.</li> <li>2. Choose to download any uploaded files.</li> <li>3. Choose to delete any uploaded files.</li> </ol>	Any uploaded files available in the dashboard	PASS
IT-006	To determine if admins can view updated info of the users' account information	<ol style="list-style-type: none"> <li>1. Update email or username of a registered account.</li> <li>2. Login into one admin account.</li> <li>3. View the user's information in the list.</li> </ol>	Any registered users for the web application	PASS



## **7.4 System Testing**

System Testing is vital as it validates the whole system runs properly without any bugs affecting the usage. In this section, the result of system testing is showed.

Table 7.7: System testing test cases

Test Case ID	Test Case Description	Test Steps	Expected Results	Status
ST-001	To verify if users can register, login, and update own information without losing the account's data	<ol style="list-style-type: none"> <li>1. Register an account and login.</li> <li>2. Navigate to profile page and update password or email.</li> <li>3. Logout and login again to check if any data is lost.</li> </ol>	Changing any information such as email, username or password will not cause the account's data to lose. Users still can login and view created chat and chat histories.	PASS
ST-002	To verify users can navigate back to the starred message after creating many other chats	<ol style="list-style-type: none"> <li>1. Chat with the chatbot and star one response.</li> <li>2. Open new chats to continue chat with the chatbot.</li> <li>3. Visit starred messages page and select one starred message.</li> <li>4. Press the go to message button.</li> </ol>	Users will still be able to navigate back to the message after creating many chats.	PASS
ST-003	To verify the chatbot will keep on providing response	<ol style="list-style-type: none"> <li>1. Open new chat or visit back one existing chat.</li> <li>2. Chat with the chatbot continuously.</li> </ol>	The chatbot will keep on answering user's prompt as long the chatbot receive a prompt from users.	PASS

ST-004	To verify admins can manage users and the uploaded files in the dashboard page	<ol style="list-style-type: none"> <li>1. Login into admin account.</li> <li>2. Perform actions on registered users list.</li> <li>3. Perform actions on uploaded files list.</li> </ol>	The dashboard page should always be available for admins to add new users, edit or delete registered users. Moreover, admins can also view, download or delete uploaded files.	PASS
--------	--	--	--	------

### **7.5 User Acceptance Testing (UAT)**

To fully ensure that this project fully implemented the requirements and validate that the web application is working smoothly. A total of 8 participants are invited to participate in the UAT, 5 of the participants are students who learned software testing in university, where the remaining 3 participants are lecturers teaching in university. Both groups of participants will have different test cases to perform during the UAT.

Before conducting the UAT with the participants, a consent form was sent to the participants, to confirm that participants are clear that the results from the test will be used in this report for analysis. The consent forms will be attached in the appendix.

The UAT form and results are attached in the appendix. Based on the results from the participants, the web application can successfully pass all test cases and run smoothly, and also indicating the project is able to implement the requirements from users. The results proven that the project is rather stable and no obvious bugs affecting usage of users.

### **7.6 Analysis on the test results**

From the tests conducted, including unit testing, integration testing, system testing and UAT, the results of each tests indicated that the web application can perform properly without any issue.

During testing, the defects found were immediately fixed to prevent future users to have issues while using this chatbot web application. After fixing the defects formed during the development of the project, the web application can allow users to use without having bugs to sabotage the usage experience.

With all the testings conducted, a quality web application was developed.

## CHAPTER 8

### CONCLUSION AND FUTURE WORKS

#### 8.1 Introduction

This chapter will be covering the conclusion for this project, limitations and future works. These were all concluded throughout the process of developing this project, and will be discussed in this chapter.

#### 8.2 Limitations

Although the development of this project was successful, there are things in this project that are quite limited in terms of the performance.

1. Slow response generation: : The model for the chatbot run locally in a laptop, this causes the response generation time become longer. Users will need to wait quite some time to wait for a response from the chatbot.
2. Response from chatbot: The chatbot's response length is very limited, this causes the response could be short and could potentially cause confusion to users due to lack of context.

#### 8.3 Future works

As stated above, there are many things that can be improved for this project. The improvements are not limited to:

1. Faster response generation: In near future, the model can be deployed to cloud using cloud services to boost up the response generation time, making the response from the chatbot becomes faster.
2. Better UI: The UI of the web application could still be not user friendly enough for other users, this should be improved to allow users to navigate through different pages through intuition and leave no confusion.

3. Providing links: The chatbot in this project is limited to answering prompts from the users, the chatbot can be improved to search for the web and retrieve useful links, either from a website or YouTube link, provide to users along with the chatbot's response. This can provide users with an extra option to study or refer to.

#### **8.4 Conclusion**

The web application proposed in the previous final year project semester has been developed successfully, although the testings conducted has shown that this project's web application can work smoothly without issue, there are still rooms for improvements as there are limitations in the web application developed. The development process provided significant learning opportunities, aided in gaining personal growth. Overall, the opportunity to be able to develop this project has been invaluable, and will be served as a cornerstone for future works.

## REFERENCES

- AdinaTru. (2024, August 23). Discover the new Multi-Lingual, High-Quality PHI-3.5 SLMS. *TECHCOMMUNITY.MICROSOFT.COM*. <https://techcommunity.microsoft.com/blog/azure-ai-services-blog/discover-the-new-multi-lingual-high-quality-phi-3-5-slms/4225280>
- Admin. (2025, February 21). Certified Tester Foundation Level (CTFL) v4.0 - International Software Testing Qualifications Board. International Software Testing Qualifications Board. <https://www.istqb.org/certifications/certified-tester-foundation-level-ctfl-v4-0/>
- Bonnet, A. (2024, October 8). *From vision to edge: Meta's Llama 3.2 explained*. <https://encord.com/blog/llama-3-2-explained/>
- Essel, H. B., Vlachopoulos, D., Tachie-Menson, A., Johnson, E. E., & Baah, P. K. (2022). The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education. *International Journal of Educational Technology in Higher Education*, 19(57). <https://doi.org/10.1186/s41239-022-00362-6>
- GeeksforGeeks. (2024, March 7). *Agile Software development Software engineering*. GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering-agile-software-development/>
- GeeksforGeeks. (2024c, July 1). *Lean Software Development (LSD)*. GeeksforGeeks. <https://www.geeksforgeeks.org/lean-software-development-lsd/>
- GeeksforGeeks. (2024a, March 6). *SDLC VModel Software Engineering*. GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/>

Kahani, A. (2025, February 17). *AI Agents: Understanding the Thinking of Llama 3.2 3B*. FlowHunt. <https://www.flowhunt.io/blog/ai-agents-understanding-the-thinking-of-llama-3-2-3b/>

Kamath, O. (2024, August 1). *Gemma 2 2B Architecture: Innovations and applications*. Cody - the AI Trained on Your Business. <https://meetcody.ai/blog/gemma-2-2b-architecture-innovations-and-applications/>

Labadze, L., Grigolia, M., & Machaidze, L. (2023). *Role of AI chatbots in education: Systematic literature review*. *International Journal of Educational Technology in Higher Education*, 20(56). <https://doi.org/10.1186/s41239-023-00426-1>

Lee, D., & Yeo, S. (2022). Developing an AI-based chatbot for practicing responsive teaching in mathematics. *Computers & Education*, 191, 104646. <https://doi.org/10.1016/j.compedu.2022.104646>

Lutkevich, B., & Silverthorne, V. (2021, December 6). *Lean software development*. Search Software Quality. <https://www.techtarget.com/searchsoftwarequality/definition/lean-programming>

Oppermann, A. (2023, April 6). *What is the V-Model in software development?* Built In. <https://builtin.com/software-engineering-perspectives/v-model#:~:text=The%20V%2Dmodel%20is%20a,and%20outs%20of%20this%20methodology.&text=The%20V%2Dmodel%20is%20named,side%20represents%20the%20validation%20phase.>

*Retrieval-Augmented Generation (RAG) | Deepgram*. (n.d.). Deepgram. <https://deepgram.com/ai-glossary/retrieval-augmented-generation>



*Software Development Methodologies: Types and comparison.* (n.d.).  
<https://www.itransition.com/software-development/methodologies>

Szatmári, A., Gergely, T., & Beszédes, Á. (2023). ISTQB-based software testing education: Advantages and challenges. *Proceedings of the 2023 International Conference on Software Engineering Education and Training (CSEE&T)*. <https://doi.org/10.5281/zenodo.10044543>

## APPENDICES

### Appendix A: Survey Questions

# A Chatbot For Teaching Software Testing CTFL Syllabus

I am Teh Wing Fung, a final year software engineering student from Universiti Tunku Abdul Rahman (UTAR). This questionnaire aims to collect feedback from people who are studying the software testing CTFL syllabus, to develop a chatbot that is useful to aid in studying the syllabus.

The information collected through this form will not be shared to anyone and remain confidential, it will only be used for the project's future improvements and changes.

If there are any questions, you may contact:

Teh Wing Fung  
winggerteh@utar.my  
010-5619463

---

*\* Indicates required question*

1. Have you taken software testing course or CTFL syllabus before? \*

*Mark only one oval.*

- ☐ Yes  
☐ No

2. Do you take the software testing course or CTFL syllabus for undergraduate requirement or to get the certified foundation tester certification? \*

*Mark only one oval.*

- ☐ Yes  
☐ No

3. Do you have any working experience in software testing area? (QA, Software Testers) \*

*Mark only one oval.*

☐ Yes

☐ No

4. When you are to study the software testing course or CTFL syllabus, how do you get the syllabus and materials? \*

*Check all that apply.*

☐ Search from the web

☐ YouTube lessons

☐ International Software Testing Qualifications Board (ISQTB) official website

☐ University provided study materials

☐ Other: \_\_\_\_\_

5. Do you use AI chatbots (ChatGPT, Grok, Perplexity, etc.) to study? \*

*Mark only one oval.*

☐ Yes

☐ No

6. Do you think using a chatbot to help in study is a good idea? \*

*Mark only one oval.*

☐ Yes

☐ No

☐ Other: \_\_\_\_\_

*Skip to question 7*

## Section B

7. When you are studying the software testing syllabus, which of the following you find it hard? \*

*Check all that apply.*

- ☐ Understand concepts
- ☐ Remember different terms in software testing
- ☐ Providing examples on certain topics
- ☐ Other: \_\_\_\_\_

8. How did you get the quiz/exam questions to practice your understandings? \*

*Check all that apply.*

- ☐ Search from the Internet
- ☐ ISQTB official website
- ☐ Use chatbots (ChatGPT, Grok, Gemini) to generate sample questions
- ☐ University provided some exam questions
- ☐ Other: \_\_\_\_\_

9. When you are attempting to do the quiz/exam questions, what are the challenges you faced? \*

*Check all that apply.*

- ☐ Unable to understand complex concepts
- ☐ Not familiar with some terms used in the questions
- ☐ Questions are not the same as in textbook/ study materials
- ☐ Other: \_\_\_\_\_

## Section C

10. From the below features, which feature you would like a chatbot to have to help in teaching? \*

*Check all that apply.*

- ☐ Quiz generation (to test understanding)
- ☐ Star messages (no need to search through chat history to find certain response from before)
- ☐ Provide useful explanations when generating responses
- ☐ Other: \_\_\_\_\_

11. How long do you think a chatbot should generate a response? \*

*Mark only one oval.*

- ☐ Less than 30 seconds
- ☐ 30 seconds or more
- ☐ 1 - 2 minutes
- ☐ Other: \_\_\_\_\_

12. Do you have any suggestion or comment for a chatbot that can be used for teaching software testing CTFL syllabus? \*

---

---

---

---

---

## Appendix B: Consent forms for UAT

### Consent & Recording Release Form

---

I agree to participate in the study conducted and recorded by A chatbot for teaching software testing CTFL syllabus.

I understand and consent to the use and release of the recording by A chatbot for teaching software testing CTFL syllabus. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by A chatbot for teaching software testing CTFL syllabus without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 10/9/2025

Please write your name: Wong Zhen Feng

---

Please sign your name: \_\_\_\_\_

*Feng*

Thank you!

I appreciate your participation!

## Consent & Recording Release Form

---

I agree to participate in the study conducted and recorded by A chatbot for teaching software testing CTFL syllabus.

I understand and consent to the use and release of the recording by A chatbot for teaching software testing CTFL syllabus. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by A chatbot for teaching software testing CTFL syllabus without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: \_\_10 Sep 2025\_\_

Please write your name: \_\_\_\_\_ Chua Kek Yang

\_\_\_\_\_

Please sign your name: \_\_\_\_\_ *Chua Kek Yang*

\_\_\_\_\_

Thank you!

I appreciate your participation!

## Consent & Recording Release Form

---

I agree to participate in the study conducted and recorded by A chatbot for teaching software testing CTFL syllabus.

I understand and consent to the use and release of the recording by A chatbot for teaching software testing CTFL syllabus. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by A chatbot for teaching software testing CTFL syllabus without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 11/09/2025

Please write your name: Alston Ho Song Lin

Please sign your name:

A handwritten signature in black ink, appearing to be 'Alston Ho Song Lin', written over a horizontal line.

Thank you!

I appreciate your participation!



## Consent & Recording Release Form

I agree to participate in the study conducted and recorded by A chatbot for teaching software testing CTFL syllabus.

I understand and consent to the use and release of the recording by A chatbot for teaching software testing CTFL syllabus. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by A chatbot for teaching software testing CTFL syllabus without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 11 September 2025

Please write your name: Goh Kah Xie

Please sign your name:



Thank you!

I appreciate your participation!

## Consent & Recording Release Form

---

I agree to participate in the study conducted and recorded by A chatbot for teaching software testing CTFL syllabus.

I understand and consent to the use and release of the recording by A chatbot for teaching software testing CTFL syllabus. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by A chatbot for teaching software testing CTFL syllabus without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: \_\_11/9/2025 \_\_\_\_\_

Please write your name: \_\_\_\_FOO JIA  
YU\_\_\_\_\_

Please sign your name:



Thank you!

I appreciate your participation!

## Consent & Recording Release Form

---

I agree to participate in the study conducted and recorded by A chatbot for teaching software testing CTFL syllabus.

I understand and consent to the use and release of the recording by A chatbot for teaching software testing CTFL syllabus. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by A chatbot for teaching software testing CTFL syllabus without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 9/9/2025 \_\_\_\_\_

Please write your name: Fatimah Zaki \_\_\_\_\_

\_\_\_\_\_

Please sign your name: fatimah \_\_\_\_\_

\_\_\_\_\_

Thank you!

I appreciate your participation!

## Consent & Recording Release Form

---

I agree to participate in the study conducted and recorded by A chatbot for teaching software testing CTFL syllabus.

I understand and consent to the use and release of the recording by A chatbot for teaching software testing CTFL syllabus. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by A chatbot for teaching software testing CTFL syllabus without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: \_\_ \_16/9/2025\_\_

Please write your name: \_\_\_\_\_Ts Dr Lee Chen Kang\_\_\_\_\_

Please sign your name: \_\_\_\_\_*Lee Chen Kang*\_\_\_\_\_

---

Thank you!

I appreciate your participation!

## Consent & Recording Release Form

---

I agree to participate in the study conducted and recorded by A chatbot for teaching software testing CTFL syllabus.

I understand and consent to the use and release of the recording by A chatbot for teaching software testing CTFL syllabus. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by A chatbot for teaching software testing CTFL syllabus without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 17/09/2025 \_\_\_\_\_

Please write your name: Gunavathi Duraisamy

\_\_\_\_\_

Please sign your name: guna

\_\_\_\_\_

Thank you!

I appreciate your participation!

## Appendix C: UAT forms and results

Participant #			1	
Testing Date			10/9/2025	
No.	Module Being Tested	Test Scenario	Results	Comments from participants
1	Account registration	<p>1. Participant will sign up for a new account for student role.</p> <p>2. Participant is able to register one student role account.</p> <p>3. Participant can login using correct credentials.</p>	Pass	None
2	Account registration	<p>1. Participant will log out using the registered account.</p> <p>2. Participant will login using incorrect credentials after sign up.</p> <p>3. Participant will not be able to login to the web application due to wrong credentials.</p>	Pass	None
3	Chatbot	1. Participant will ask a prompt related to CTFL syllabus.	Pass	None

		2. The chatbot in the web application will generate response based on the prompt from the participant.		
4	Chatbot, Quiz	<p>1. Participant will ask the chatbot to generate a quiz based on CTFL syllabus.</p> <p>2. The chatbot will generate a quiz based on the prompt from participants.</p> <p>3. Participants can provide answers in the chat input box.</p> <p>4. The chatbot will provide answers and validate score of participant's answer.</p>	Pass	None

Participant #			2	
Testing Date			10/9/2025	
No.	Module Being Tested	Test Scenario	Results	Comments from participants
1	Account registration	1. Participant will sign up for a new account for student role.	Pass	None

		<p>2. Participant is able to register one student role account.</p> <p>3. Participant can login using correct credentials.</p>		
2	Account registration	<p>1. Participant will log out using the registered account.</p> <p>2. Participant will login using incorrect credentials after sign up.</p> <p>3. Participant will not be able to login to the web application due to wrong credentials.</p>	Pass	None
3	Chatbot	<p>1. Participant will ask a prompt related to CTFL syllabus.</p> <p>2. The chatbot in the web application will generate response based on the prompt from the participant.</p>	Pass	None
4	Chatbot, Quiz	1. Participant will ask the chatbot to generate a quiz based on	Pass	None



		<p>CTFL syllabus.</p> <p>2. The chatbot will generate a quiz based on the prompt from participants.</p> <p>3. Participants can provide answers in the chat input box.</p> <p>4. The chatbot will provide answers and validate score of participant's answer.</p>		
--	--	--	--	--

Participant #			3	
Testing Date			10/9/2025	
No.	Module Being Tested	Test Scenario	Results	Comments from participants
1	Account registration	<p>1. Participant will sign up for a new account for student role.</p> <p>2. Participant is able to register one student role account.</p> <p>3. Participant can login using correct credentials.</p>	PASS	None
2	Account registration	1. Participant will log out using the	PASS	None

		<p>registered account.</p> <p>2. Participant will login using incorrect credentials after sign up.</p> <p>3. Participant will not be able to login to the web application due to wrong credentials.</p>		
3	Chatbot	<p>1. Participant will ask a prompt related to CTFL syllabus.</p> <p>2. The chatbot in the web application will generate response based on the prompt from the participant.</p>	PASS	Provided decent example
4	Chatbot, Quiz	<p>1. Participant will ask the chatbot to generate a quiz based on CTFL syllabus.</p> <p>2. The chatbot will generate a quiz based on the prompt from participants.</p> <p>3. Participants can provide answers in the</p>	PASS	Chatbot helped to check my answers correctly

		chat input box.  4. The chatbot will provide answers and validate score of participant's answer.		
--	--	--	--	--

Participant #			4	
Testing Date			10/9/025	
No.	Module Being Tested	Test Scenario	Results	Comments from participants
1	Account registration	1. Participant will sign up for a new account for student role.  2. Participant is able to register one student role account.  3. Participant can login using correct credentials.	PASS	no
2	Account registration	1. Participant will log out using the registered account.  2. Participant will login using incorrect credentials after sign up.  3. Participant will not be able to login to the web	PASS	no

		application due to wrong credentials.		
3	Chatbot	<p>1. Participant will ask a prompt related to CTFL syllabus.</p> <p>2. The chatbot in the web application will generate response based on the prompt from the participant.</p>	PASS	no
4	Chatbot, Quiz	<p>1. Participant will ask the chatbot to generate a quiz based on CTFL syllabus.</p> <p>2. The chatbot will generate a quiz based on the prompt from participants.</p> <p>3. Participants can provide answers in the chat input box.</p> <p>4. The chatbot will provide answers and validate score of participant's answer.</p>	PASS	no

Participant #	5
Testing Date	10/9/2025

No.	Module Being Tested	Test Scenario	Results	Comments from participants
1	Account registration	<p>1. Participant will sign up for a new account for student role.</p> <p>2. Participant is able to register one student role account.</p> <p>3. Participant can login using correct credentials.</p>	Pass	No comment
2	Account registration	<p>1. Participant will log out using the registered account.</p> <p>2. Participant will login using incorrect credentials after sign up.</p> <p>3. Participant will not be able to login to the web application due to wrong credentials.</p>	Pass	No comment
3	Chatbot	<p>1. Participant will ask a prompt related to CTFL syllabus.</p> <p>2. The chatbot in the web application will generate response</p>	Pass	No comment

		based on the prompt from the participant.		
4	Chatbot, Quiz	<p>1. Participant will ask the chatbot to generate a quiz based on CTFL syllabus.</p> <p>2. The chatbot will generate a quiz based on the prompt from participants.</p> <p>3. Participants can provide answers in the chat input box.</p> <p>4. The chatbot will provide answers and validate score of participant's answer.</p>	Pass	No comment

Participant #			6	
Testing Date			9/9/2025	
No.	Module Being Tested	Test Scenario	Results	Comments from participants
1	Account registration	<p>1. Participant will sign up for a new account for lecturer role.</p> <p>2. Participant will key in the correct lecturer code</p>	PASS	None

		<p>for registering the account.</p> <p>3. Participant is able to register one lecturer role account.</p> <p>4. Participant can login using correct credentials.</p>		
2	Account registration	<p>1. Participant will log out using the registered account.</p> <p>2. Participant will login using incorrect credentials after sign up.</p> <p>3. Participant will not be able to login to the web application due to wrong credentials.</p>	PASS	None
3	Chatbot	<p>1. Participant will ask a prompt related to CTFL syllabus.</p> <p>2. The chatbot in the web application will generate response based on the prompt from the participant.</p>	PASS	Best is that the chatbot can provide citations. For instance, state the version of response, whether its from version 4.0 or newer versions.
4	Chatbot, Quiz	1. Participant will ask the	PASS	None

		<p>chatbot to generate a quiz based on CTFL syllabus.</p> <p>2. The chatbot will generate a quiz based on the prompt from participants.</p> <p>3. Participants can provide answers in the chat input box.</p> <p>4. The chatbot will provide answers and validate the correctness of participant's answer.</p>		
5	Upload file	<p>1. Participants will visit the upload materials page in the web application.</p> <p>2. Participants can upload a selected file.</p> <p>3. One individual will log in as admin to check if the material is uploaded.</p>	PASS	None

Participant #	7
Testing Date	16/9/2025



No.	Module Being Tested	Test Scenario	Results	Comments from participants
1	Account registration	<p>1. Participant will sign up for a new account for lecturer role.</p> <p>2. Participant will key in the correct lecturer code for registering the account.</p> <p>3. Participant is able to register one lecturer role account.</p> <p>4. Participant can login using correct credentials.</p>	Pass	
2	Account registration	<p>1. Participant will log out using the registered account.</p> <p>2. Participant will login using incorrect credentials after sign up.</p> <p>3. Participant will not be able to login to the web application due to wrong credentials.</p>	Pass	
3	Chatbot	1. Participant will ask a prompt related	Pass	

		<p>to CTFL syllabus.</p> <p>2. The chatbot in the web application will generate response based on the prompt from the participant.</p>		
4	Chatbot, Quiz	<p>1. Participant will ask the chatbot to generate a quiz based on CTFL syllabus.</p> <p>2. The chatbot will generate a quiz based on the prompt from participants.</p> <p>3. Participants can provide answers in the chat input box.</p> <p>4. The chatbot will provide answers and validate the correctness of participant's answer.</p>	Pass	
5	Upload file	<p>1. Participants will visit the upload materials page in the web application.</p> <p>2. Participants can upload a selected file.</p>	Pass	

		3. One individual will log in as admin to check if the material is uploaded.		
--	--	--	--	--

Participant #			8	
Testing Date			17/08/2025	
No.	Module Being Tested	Test Scenario	Results	Comments from participants
1	Account registration	<p>1. Participant will sign up for a new account for lecturer role.</p> <p>2. Participant will key in the correct lecturer code for registering the account.</p> <p>3. Participant is able to register one lecturer role account.</p> <p>4. Participant can login using correct credentials.</p>	<p>Pass</p> <p>Pass</p> <p>Pass</p> <p>Pass</p>	Ok, no issues
2	Account registration	<p>1. Participant will log out using the registered account.</p> <p>2. Participant will login using incorrect credentials after sign up.</p>	<p>Pass</p> <p>Pass</p> <p>Pass</p>	Ok, no issues

		3. Participant will not be able to login to the web application due to wrong credentials.		
3	Chatbot	<p>1. Participant will ask a prompt related to CTFL syllabus.</p> <p>2. The chatbot in the web application will generate response based on the prompt from the participant.</p>	<p>Pass</p> <p>Pass</p>	Ok, the chatbot able to answer the questions asked and it is providing answers related to CTFL syllabus.
4	Chatbot, Quiz	<p>1. Participant will ask the chatbot to generate a quiz based on CTFL syllabus.</p> <p>2. The chatbot will generate a quiz based on the prompt from participants.</p> <p>3. Participants can provide answers in the chat input box.</p> <p>4. The chatbot will provide answers and validate the correctness of</p>	<p>Pass</p> <p>Pass</p> <p>Pass</p>	Ok, the chatbot can guide if then answer for quiz is incorrect and gives some hints.

		participant's answer.		
5	Upload file	<p>1. Participants will visit the upload materials page in the web application.</p> <p>2. Participants can upload a selected file.</p> <p>3. One individual will log in as admin to check if the material is uploaded.</p>	<p>Pass</p> <p>Pass</p> <p>Pass</p>	Able to upload documents where it can be used to train the model later.