STUDENT RESOURCE EXCHANGE: A WEB-BASED SYSTEM FOR SHARING EDUCATIONAL RESOURCES AMONG STUDENTS

BY

SAW HUI LOO

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Honours) at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment 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.

ACKNOWLEDGEMENTS

Firstly, I want to sincerely thank Dr. Jasmina Khaw Yen Min, my supervisor, for her help, support, and encouragement during this study. Her kindness has been invaluable in assisting me in completing this project.

I also want to express my sincere gratitude to my friends and family for their constant help and continuous encouragement. Their support and motivation helped me a lot.

Lastly, I would like to express my deepest appreciation to everyone who helped complete this project successfully, whether directly or indirectly.

ABSTRACT

This project focuses on the development of the Student Resource Exchange, a web-based platform designed to enhance the sharing and accessibility of educational resources among students. The platform addresses key limitations found in existing resource-sharing platforms, such as the lack of automated content analysis, insufficient content classification, and inadequate support for personalized learning. To overcome these issues, the system integrates Optical Character Recognition (OCR) for extracting text from images, Whisper for video transcription, and Google Gemini for generating concise video summaries. Natural Language Processing (NLP) techniques, such as TF-IDF with Logistic Regression for hate speech detection and BERT-based sentiment analysis, ensure safe and meaningful interaction within the platform. For content classification, the Google Gemini API with structured prompt engineering is applied to automatically organize materials into relevant academic courses. In addition, the platform introduces AI-powered study tools, including automated question generation and flashcards, as well as a built-in smart calendar planner that applies conflict detection and priority-based scheduling algorithms to help students manage study sessions and deadlines effectively. This system aims to provide a more efficient, user-friendly, and engaging platform for students to exchange educational resources, ultimately fostering a more collaborative and accessible learning environment.

Area of Study: Web Development

Keywords: Optical Character Recognition (OCR), Natural Language Processing (NLP), AI-powered Study Tools, Hate Speech Detection, Smart Calendar Planner

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

OCR	Optical Character Recognition
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
TF-IDF	Term Frequency-Inverse Document Frequency
LLM	Large Language Model
AI	Artificial Intelligence
ASR	Automatic Speech Recognition
NLTK	Natural Language Toolkit
CORS	Cross-Origin Resource Sharing
API	Application Programming Interface
PHP	Hypertext Preprocessor
HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheets
PIL	Python Imaging Library

Chapter 1
Introduction

In recent years, digital technology has significantly transformed individuals' interactions with information. This transformation has mainly been highest in the education sector with digital mediums increasingly being utilized to acquire, exchange, and manage information. In this digital learning environment, there is reduced usage of hardbound textbooks by students; however, students are actively utilizing electronic materials such as e-books, lecture slides, videos, and many other multi-media resources. The transition to digital learning has brought about the demand for electronic mediums via which students are able to exchange learning resources quickly and efficiently.

Although many websites develop to provide platforms for sharing study resources, many are still suffering from significant drawbacks. The majority of platforms are function primarily as repositories for uploading and downloading files, offering limited support for organizing, analyzing, or interpreting the content. As a result, students often encounter difficulties when searching for relevant resources, particularly when materials are provided in different formats such as documents, images, or videos. Moreover, existing platforms generally lack automated tools for extracting meaningful information from uploaded content, such as converting images into editable and searchable text or summarizing the contents of video lectures. These limitations hinder students' ability to fully utilize the resources available to them and reduce the overall efficiency of digital learning.

During the COVID-19 pandemic, the use of web-based platforms for educational resource sharing significantly increased. Before the COVID-19 pandemic, digital learning platforms were primarily used as supplementary tools in education, providing additional resources and enhancing traditional classroom experiences. However, the pandemic necessitated a rapid shift, transforming these digital platforms from supplementary resources into the central mode of instruction. The outbreak of the pandemic forced more than 190 education systems worldwide to implement school closures to combat COVID-19, the impact on over 1.5 billion students was immense [1]. Research by Wong [2] highlights how the pandemic has accelerated the shift

towards online distance learning. The study underscores the challenges educational institutions face in adapting to this rapid transition, including issues related to readiness and the effectiveness of online learning solutions [2]. The pandemic has revealed some weaknesses in the current education system, which may not have been well-adapted to the new changes [2]. This situation underscored the urgency of developing effective digital platforms to support remote education and address these emerging challenges [2].

As a result of enhanced digitalization and challenges raised by the pandemic, several sites like Scribd [8], CourseHero [7], Studocu [6], Docsity [9], and OneClass [10] have emerged as resources for students to access and share educational materials. These platforms provide a wealth of content that can enhance learning experiences, offering lecture notes, past exams, and study guides. However, with all their advantages, there are remaining disadvantages these platforms are yet to rectify. Access to premium content is often restricted, the quality and accuracy of uploaded materials can vary significantly, and some resources may be outdated or incomplete. Consequently, students may still struggle to efficiently find reliable and relevant learning materials.

Thus, there is a need to create a user-friendly web system to disseminate learning resources to students. The platform will feature several key components to facilitate user experience and organization of resources. Optical Character Recognition (OCR) will be utilized to extract text from images so students can convert multimedia resources to editable and retrievable materials. Transcriptions of videos can be developed using Whisper tools or downloading transcripts from YouTube, with a summary being generated using Gemini AI. The automatically uploaded contents will belong to respective courses and thus can be quickly sorted and found by students. There can be further interaction with peers by students using a comment section to enhance feedback and interaction. In addition to this, there is a smart calendar system to facilitate students' timetable import and task setting with AI-powered study schedule recommendations based on deadlines. Lastly, there is a functionality to create questions to develop quizzes and flashcards and to facilitate an AI chat functionality based on contents uploaded.

1.1 Problem Statement and Motivation

In this section, the following problem statements are defined based on a review of existing web-based educational platforms, highlighting key areas where improvements are needed:

i. Lack of Text Extraction and Video Content Summarization

The existing platforms often lack advanced features for text extraction and video content summarization from multimedia files such as images and videos. For instance, these platforms generally do not have the capability to convert user-uploaded images into editable and searchable text formats. Despite platforms like Studocu [6] offering a wealth of learning materials for user reference, they lack automated content analysis and text extraction features. This deficiency adds steps for users when uploading educational materials. Text extraction is particularly useful for extracting text from images. For instance, if users have handwritten learning materials on paper and need to upload them as Word documents, this feature can significantly reduce the manual typing workload.

ii. Lack of Automated Content Classification for User-Uploaded Content

Many current educational platforms do not offer automated content classification and categorization for user-uploaded materials. As an example, even though CourseHero [7] provides a wide range of courses, it still lacks features that can automatically classify different types of user-uploaded content into their corresponding categories. This absence of automated classification can make it difficult for users to manage and retrieve educational resources efficiently, reducing overall user experience. Currently, the existing platforms do not provide automated content classification for user-uploaded content. As a result, users need to manually fill in detailed information when they want to upload learning materials. This manual process becomes time-consuming, especially while uploading a high number of files. For instance, on a platform like CourseHero [7], users must enter detailed information for each upload and then wait for the content to be processed and unlocked, which further consumes their time.

iii. Lack of Interaction and Collaboration Features

Some of the existing educational platforms lack interactive and collaborative features such as commenting systems, calendar planner feature, and personalized AI learning tools. For example, even though Scribd [8] provides access to huge numbers of documents, it does not provide features like commenting. This will limit opportunities for users to communicate and provide feedback. This absence reduces engagement and creates delays in communication. Similarly, platforms such as Studocu [6], CourseHero [7], Scribd [8], Docsity [9], and OneClass [10] do not provide built-in calendar planner feature, making it difficult for students to manage their workload effectively and often leading to poor time management. Additionally, platforms like Scribd [8] and OneClass [10] lack personalized learning support such as flashcards, quizzes, and AI chatbot, preventing students from testing their knowledge or revising efficiently, which forces reliance on external applications and results in a fragmented learning experience.

1.2 Objectives

The main goal of this project is to implement a web-based system that allows students to share their educational resources among each other. The system aims to solve the limitations of existing platforms by integrating some useful educational features and functionalities. Key objectives include:

- **To develop and implement a content extraction and summarization system:** Utilizing OCR for text extraction from images and integrating Whisper for video transcription and Google Gemini for generating video summaries, enabling efficient management of multimedia educational materials.
- **To create an automated course classification system:** Integrating Large Language Models (LLMs) using Google Gemini API to analyze the extracted content even in different file format, utilizing structured prompt engineering for automated educational content classification and course categorization.
- **To develop collaboration and personalized learning support features:** Implementing a commenting system with BERT-based sentiment analysis and TF-IDF hate speech detection for automated comment moderation, developing a built-in calendar planner for managing study schedules and deadlines, and integrating AI-powered tools such as automated question generation, flashcards, and AI chat to enhance active learning and personalized revision.

1.3 Project Scope and Direction

The scope of this project includes the design, development, and implementation of a web-based platform for sharing educational resources among students. The system aims to enhance the accessibility, organization, and interaction of educational materials. Specific tasks and features within this scope include:

- **Text Extraction and Content Summarization:** Implementing Optical Character Recognition (OCR) for extracting text from images, using Whisper for video transcription, and integrating Google Gemini for generating concise video summaries.
- **Automated Content Classification:** Implementing a prompt-engineered Large Language Model system that automatically categorizes user-uploaded materials into relevant course categories by leveraging Gemini's pre-trained language understanding capabilities, reducing manual classification effort and improving content organization.
- **Interactive Comment System:** Implementing a commenting feature with machine learning-based sentiment analysis (BERT transformer model) and hate speech detection (TF-IDF + Logistic Regression**)**, enabling users to engage in discussions, provide feedback, and collaborate effectively within the platform.
- **Smart Calendar Planner**: Implementing a rule-based scheduling system with conflict detection and greedy scheduling algorithms that automatically organizes study sessions, detects scheduling conflicts, calculates task progress, and provides alternative time recommendations.
- **AI-Powered Questions Generator**: Implementing a comprehensive question generation system that leverages Gemini's Large Language Model to automatically generate flashcards, quizzes, and interactive AI chatbot conversations from user-uploaded materials, providing intelligent answer validation and personalized learning experiences.

- **User Management and Security:** Implementing secure user authentication and data protection mechanisms using PHP within the XAMPP environment, ensuring that educational resources are shared in a secure and controlled environment.
- **Database Management and Storage:** Using MySQL within the XAMPP environment to store and manage user information, educational resources, chat logs, and extracted content, ensuring reliable data organization, retrieval, and security.
- **User Interface Design:** Creating a user-friendly interface that is intuitive, responsive, and accessible across various devices, focusing on enhancing the overall user experience.

1.4 Contributions

The primary contribution of this work is to develop a web-based system for sharing educational resources among students, addressing key limitations found in existing platforms. By implementing features like text extraction and content summarization, the system allows users to efficiently convert multimedia content, such as images and videos, into editable and searchable formats [3]. Additionally, the integration of automated content classification will simplify the process for users uploading materials, improving overall efficiency. Moreover, the system will implement interactive features, such as a commenting system, which can further enhance user engagement and collaboration, creating a more dynamic learning environment. Sentiment analysis and hate speech detection are incorporated into the interactive comment system to ensure constructive discussions and maintain a safe environment.

In addition, the system integrates a smart calendar planner with conflict detection and scheduling algorithms to support effective study planning and time management. An AI-powered questions generator is included to produce personalized learning materials such as flashcards, quizzes, and interactive chatbot assessments from user-uploaded content. Overall, the system provides a convenient and complete platform for users to manage, share, and access study resources, while supporting time management through the smart calendar planner and facilitating revision and test preparation using the AI-powered questions generator.

1.5 Report Organization

Chapter 1: Introduces the project background, problem statements, motivation, project objectives, project scope as well as the contributions from the project.

Chapter 2: Literature review of existing educational resource sharing platforms. Includes the limitations of the existing systems and the proposed solutions to these limitations.

Chapter 3: System methodology. System architecture design, use case diagrams, use case descriptions, and activity diagrams were included in this chapter

Chapter 4: Contains the system design including the system flowcharts for all major modules and database SQL commands for data storage management.

Chapter 5: Includes the system implementation where the hardware setup, software setup, installation of required libraries, algorithms, and system operation were recorded.

Chapter 6: Includes system evaluation and discussion about the unit testing results, project challenges, and objectives evaluation.

Chapter 7: Conclusion of overall report and recommendation.

Chapter 2
Literature Review

2.1 Review of Existing Educational Resource Sharing Platforms

Based on the research from Internet, there are some existing educational resource sharing platforms for students to share their learning materials such as Scribd [8], Course Hero [7], Docsity [9], Studocu [6] and OneClass [10].

2.1.1 Course Hero [7]

Course Hero is a widely recognized educational platform designed to help students share and access a vast array of study materials. The platform primarily operates as a subscription-based service where users can upload and download resources comprising practice questions, study aids, textbook solutions, and lecture notes. Below are some of the key features of the Course Hero website, with brief explanations of each:

i. Subscription-Based Access

Course Hero operates on a subscription model, allowing users to unlock comprehensive access to its extensive library of educational materials. Subscribers pay a recurring fee to enjoy full benefits, which include unlimited downloads of study resources, access to textbook solutions, and 24/7 homework help. For those who prefer not to subscribe, Course Hero offers a limited free access option, which can be expanded by uploading study materials to the platform. This tiered access ensures that users who contribute resources can earn unlocks and benefit from additional features without necessarily having to pay for a subscription.

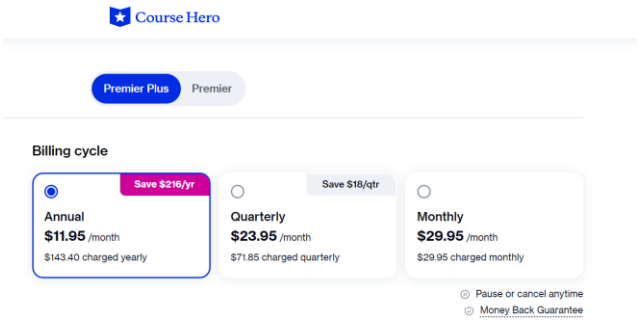


Figure 2.1.1 Course Hero Subscription-Based Access Feature [7]

ii. 24/7 Homework Help

Course Hero provides feature of homework assistance through its 24/7 Homework Help. This service connects students with qualified tutors who offer real-time help and answer questions related to various subjects. Users can submit their questions and receive detailed explanations, guidance, and solutions from experts at any time of day or night. This feature is particularly beneficial for students needing support outside regular study hours or those seeking immediate clarification on challenging topics.

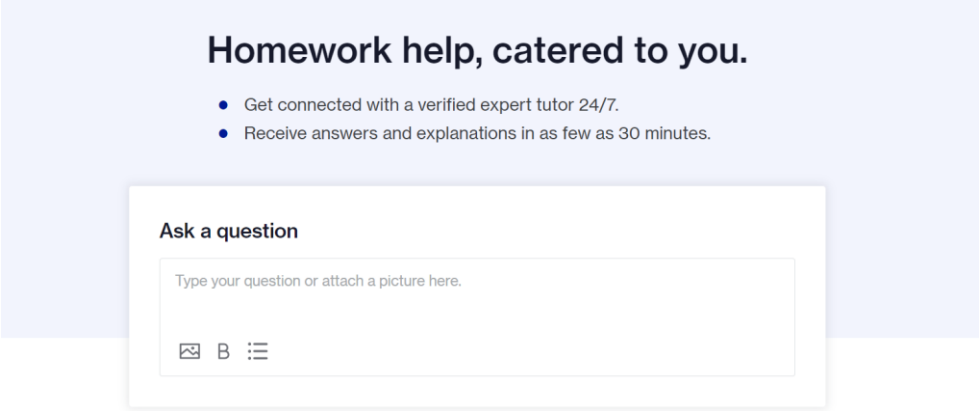


Figure 2.1.2 Course Hero 24/7 Homework Help Feature [7]

iii. Course Hero Quizzes

Course Hero offers a variety of quizzes designed to help students assess their understanding of course material. These quizzes cover a wide range of subjects and topics, providing users with an opportunity to test their knowledge and gauge their academic progress. By incorporating interactive quizzes, Course Hero helps students reinforce their learning and identify areas where they may need additional study.

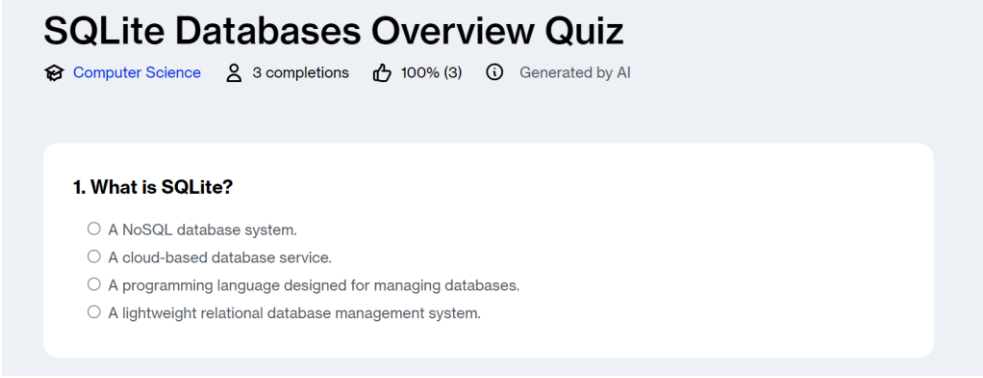


Figure 2.1.3 Course Hero Quizzes Feature [7]

iv. Textbook Solutions and Explanations

Course Hero provides detailed textbook solutions and explanations to help students understand complex concepts and solve challenging problems. Users can access step-by-step solutions to textbook exercises across various subjects, which aids in clarifying difficult topics and enhancing their learning experience.

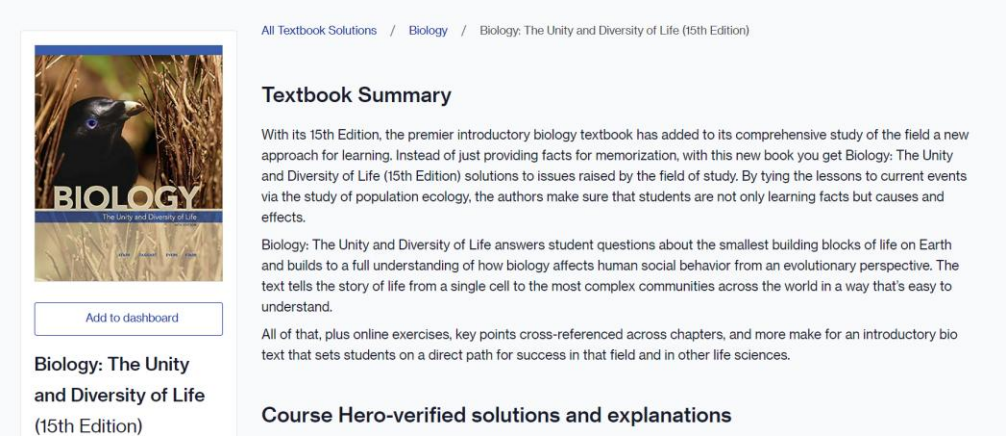


Figure 2.1.4 Course Hero Textbook Solutions and Explanation Feature [7]

v. Grammar Checker

Course Hero includes a grammar checker tool that helps students improve their writing by identifying and correcting grammatical errors. This feature supports users in producing well-written academic papers and assignments by providing suggestions for improving sentence structure, punctuation, and overall writing clarity.

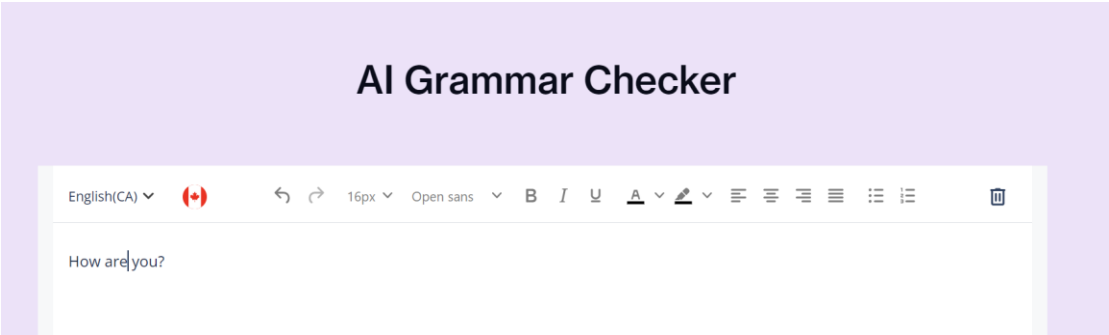


Figure 2.1.5 Course Hero Grammar Checker Feature [7]

2.1.2 Scribd [8]

Scribd is a well-established digital library and educational platform that offers students access to a broad range of reading materials and academic resources. The platform operates on a subscription-based model, allowing users to explore and share a diverse collection of documents, including e-books, audiobooks, academic papers, and more. Below are some of the key features of the Scribd website, with brief explanations of each:

i. Diverse Document Types

Scribd provides access to a wide range of document types, including academic, professional, cultural, hobbies & crafts, and personal growth materials. This extensive library allows users to explore various fields of interest and gain valuable insights across multiple areas of knowledge.

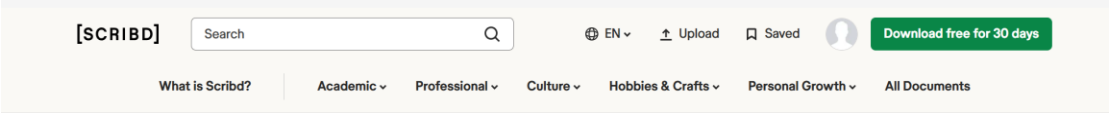


Figure 2.1.6 Scribd Diverse Document Types [8]

ii. Document Upload and Sharing

Users can upload their own documents to Scribd, making it easy to share educational resources with a global audience. This feature facilitates the exchange of study materials and helps users contribute to the Scribd community.

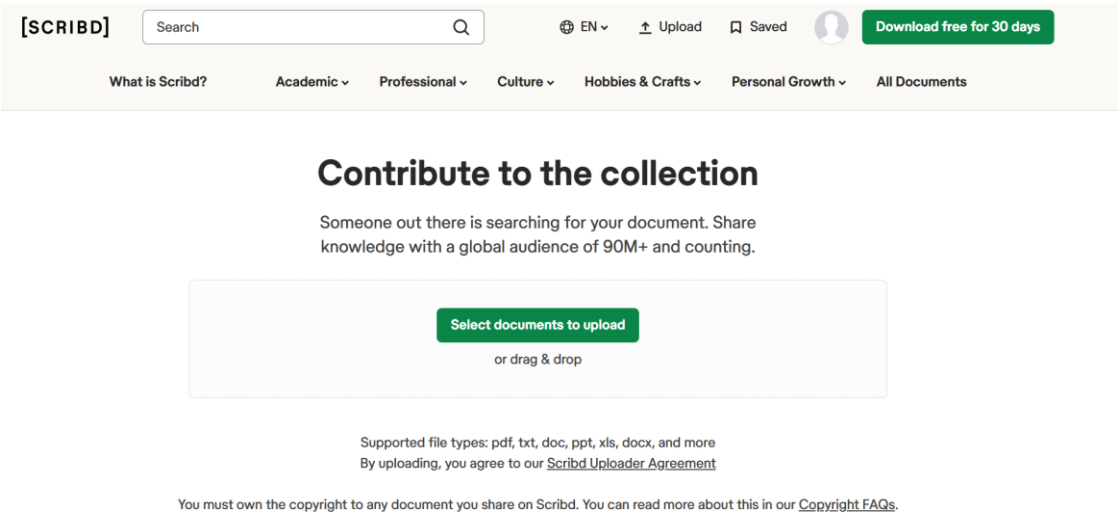


Figure 2.1.7 Scribd Document Upload and Sharing Feature [8]

iii. Language Options

Scribd offers multi-language support, allowing users to select their preferred language for site navigation and personalized recommendations. This feature enhances accessibility and user experience by providing content and interface options in multiple languages, including English, Spanish, Portuguese, German, French, Russian, Italian, Romanian, and Indonesian.

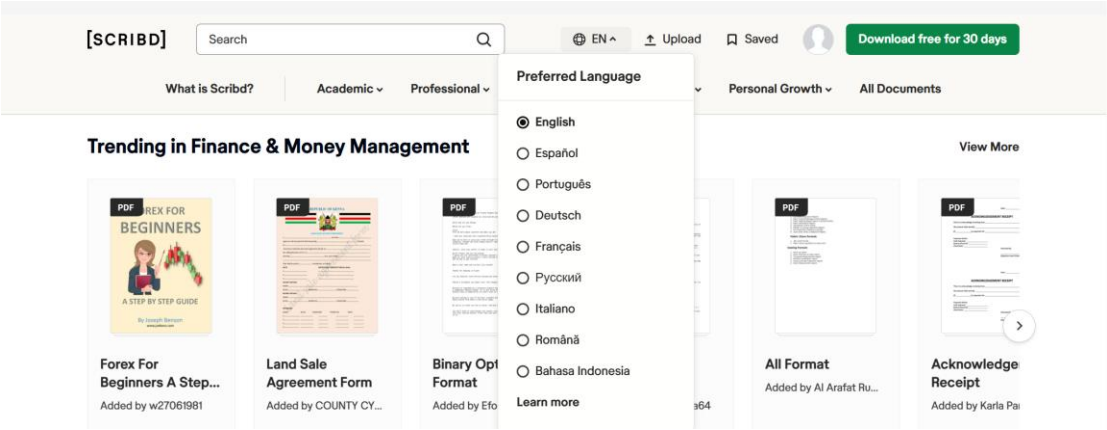


Figure 2.1.8 Scribd Language Options [8]

2.1.3 Docsity [9]

Docsity is an educational resource-sharing platform that helps students access, exchange, and share study materials. It allows users to upload their notes, summaries, and academic resources, creating a collaborative learning environment. Key features include a "Ask Community" feature for peer support, a note-selling system and others:

i. Note Selling and Buying

Docsity provides users with the ability to upload their study notes, which can earn points. These points can be exchanged for various rewards, such as discounts, premium access, or other benefits. In certain regions, users can even sell their notes directly for money, offering an additional incentive to contribute to the platform.

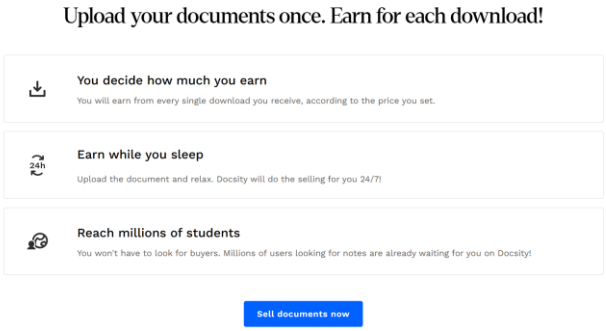


Figure 2.1.9 Docsity Note Selling and Buying Feature [9]

ii. Ask Community

Docsity include a "Ask a Question" section where students can ask academic questions and seek help from peers or verified contributors. This feature encourages collaboration and allows users to benefit from collective knowledge, creating a peer-to-peer support system that aids in both learning and problem-solving.

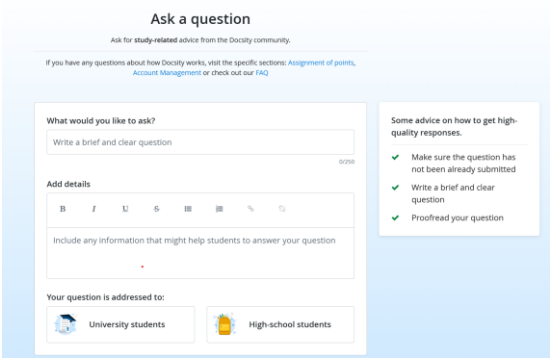


Figure 2.1.10 Docsity Ask Community Feature [9]

iii. User Ratings and Reviews

Docsity allows users to rate uploaded documents based on quality and usefulness. Higher-rated notes appear more prominently, helping users find better study materials.

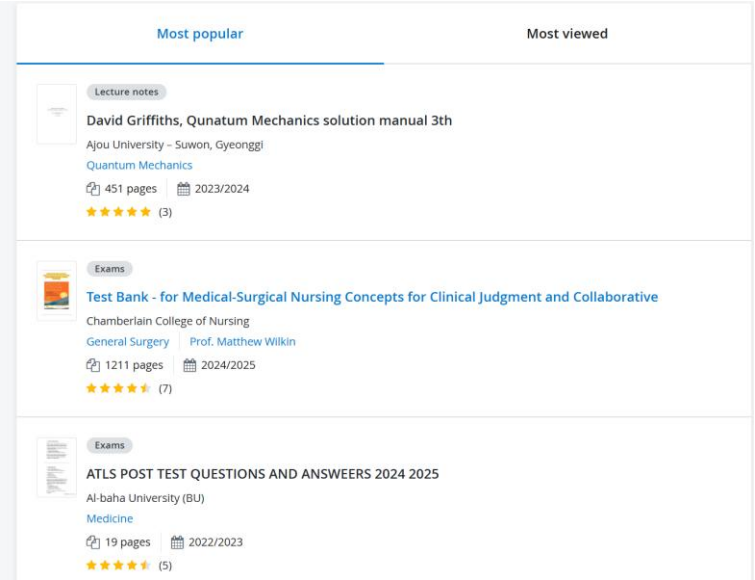


Figure 2.1.11 Docsity Rating and Reviews Feature [9]

2.1.4 Studocu [6]

Studocu is a comprehensive educational resource sharing platform designed to facilitate the exchange of study materials among students. The platform offers a variety of features aimed at enhancing the accessibility and quality of educational resources. Below are some key features of Studocu:

i. Study Groups

Studocu offers a feature for users to join or create study groups. This allows students to connect with peers who share similar academic interests or are enrolled in the same courses. Users can browse and search for groups, participate in discussions, and collaborate on academic projects.

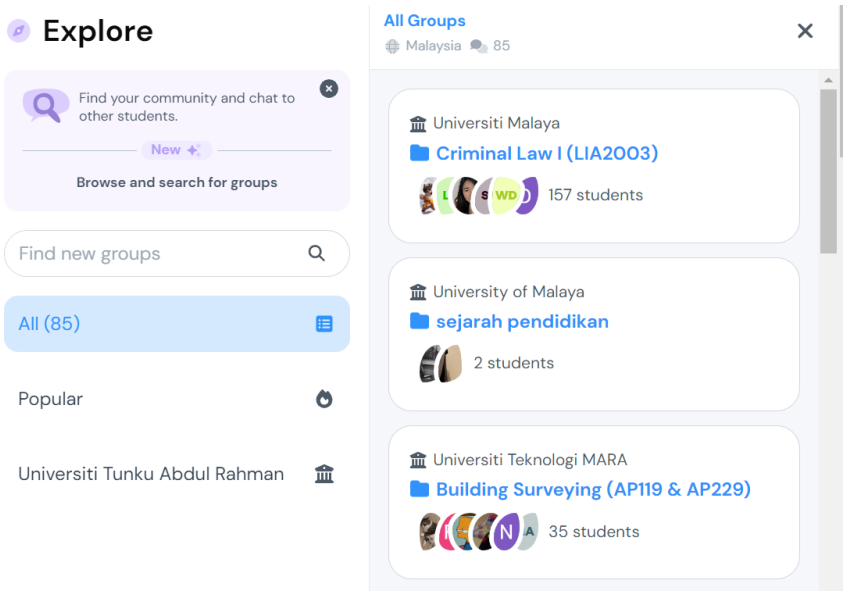


Figure 2.1.12 StuDocu Study Groups Feature [6]

ii. Blog

Studocu maintains a blog that covers a variety of topics relevant to students. The blog includes articles on job tips, lifestyle, news and updates, student life, study tips, travel, and more.

Figure 2.1.13 StuDocu Blog Feature [6]

iii. Ask AI

Studocu features an “Ask AI” tool that allows users to submit study questions and receive instant answers. This feature leverages AI technology to provide quick and accurate responses to academic inquiries, helping students get the information they need efficiently.

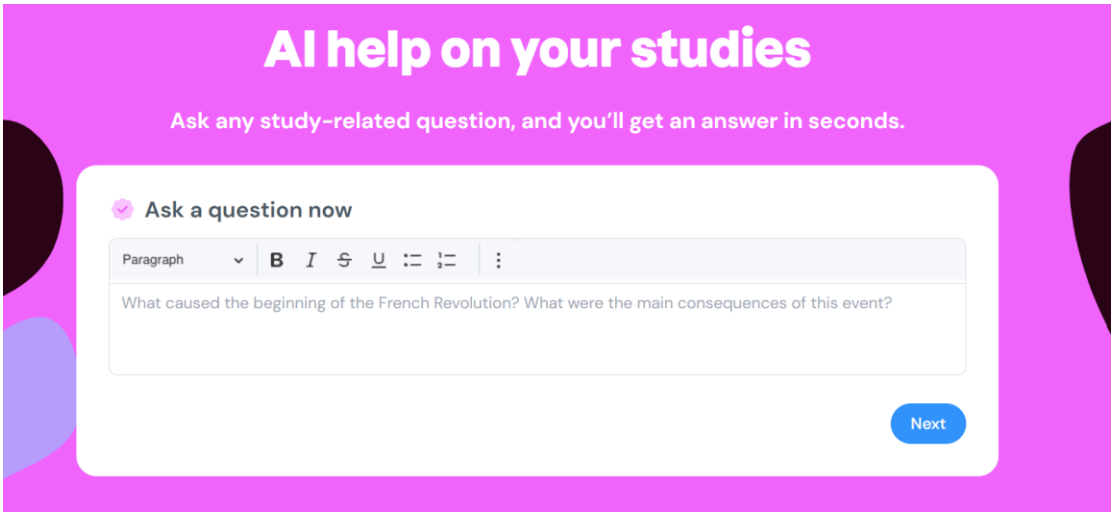


Figure 2.1.14 StuDocu Ask AI Feature [6]

iv. Study Tools and Features

Studocu includes various tools designed to assist students in their studies. These include features like flashcards, quizzes, and note-taking tools that help users consolidate their learning and improve their academic performance.

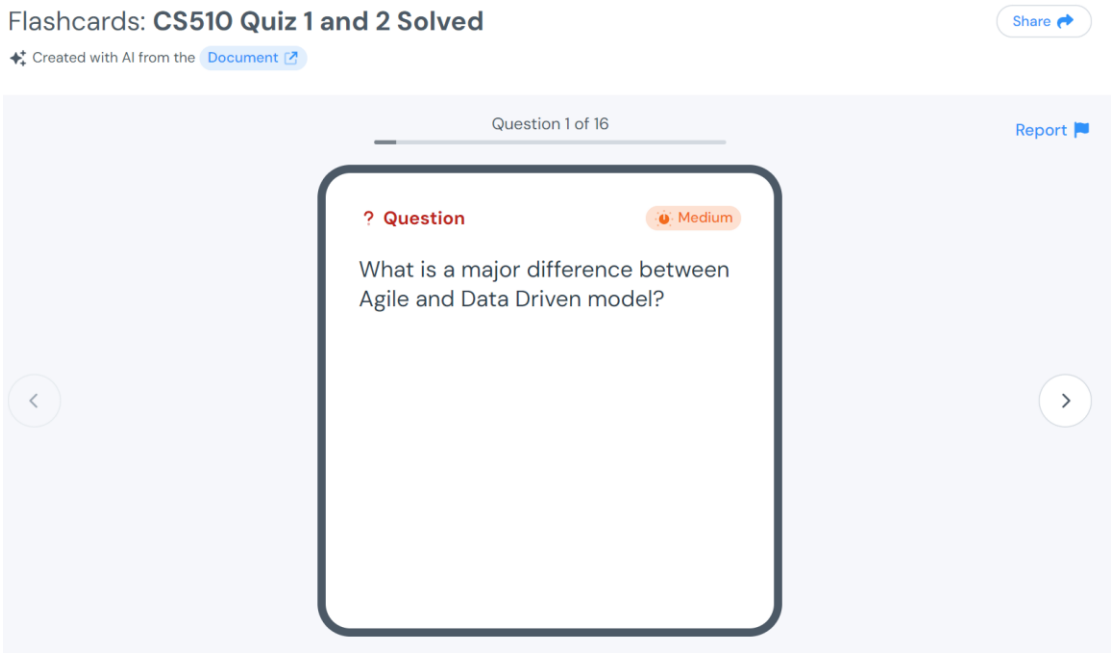


Figure 2.1.15 StuDocu Study Tools Feature [6]

2.1.5 OneClass [10]

OneClass is an educational platform designed to enhance students' learning experiences by providing access to a broad range of academic resources. The platform focuses on delivering study materials, interactive tools, and a supportive community to help students excel in their studies. Below are some of the key features of OneClass:

i. Community Q&A Feature

OneClass enables users to ask academic questions and receive answers from the community. This feature allows students to seek help with specific problems or topics and get responses from other users, fostering collaborative problem-solving and peer-to-peer assistance.

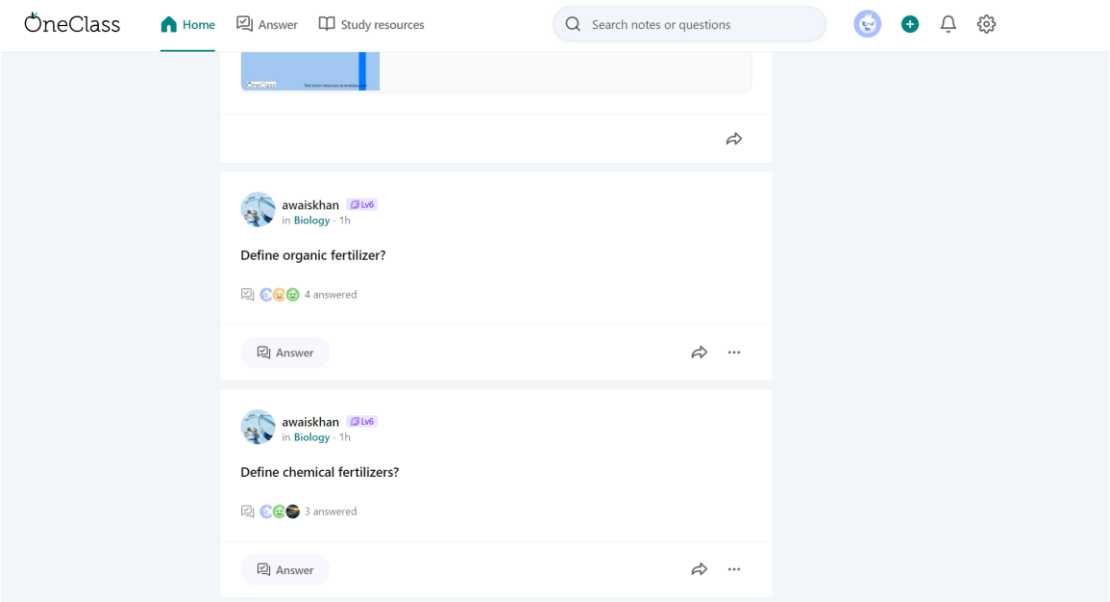


Figure 2.1.16 OneClass Community Q&A Feature [10]

ii. Class Notes

OneClass offers a feature where students can access a repository of class notes shared by their peers. These notes are detailed and cover various topics discussed during lectures. This feature allows users to supplement their own notes with additional perspectives and insights, providing a more comprehensive understanding of the course material.

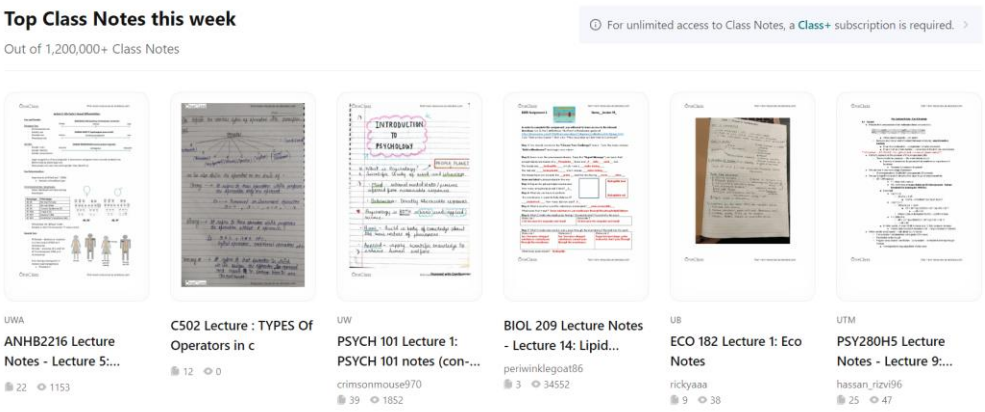


Figure 2.1.17 OneClass Class Notes Feature [10]

iii. Textbook Solutions

Another valuable feature of OneClass is its collection of detailed solutions to textbook problems. This feature enables students to verify their answers against provided solutions, ensuring accuracy and learning from any mistakes. The solutions include step-by-step explanations, offering insights into problem-solving methods and techniques.

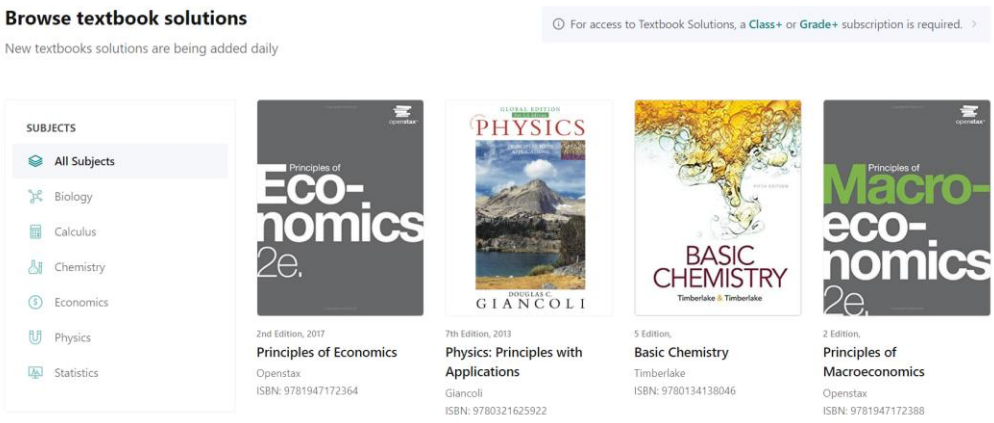


Figure 2.1.18 OneClass Textbook Solutions Feature [10]

2.2 Limitation of Existing Educational Resource Sharing Platforms

The limitation across these platforms such as [6], [7], [8], and [10] is that many of their advanced features are locked behind a subscription-based model. Users are often required to upgrade to a premium account to access additional functionalities, which can limit the overall user experience and accessibility. This subscription model may create barriers for some students, affecting their ability to fully utilize the resources available. For [7], users can either pay or upload notes to gain access to the locked documents. If they choose to upload notes, they must submit 10 notes to gain access. The process of uploading and verifying these notes can be time-consuming, which may negatively impact on the user experience.

Additionally, a significant limitation of websites such as [7] and [8] is the lack of a commenting feature. This absence prevents users from engaging in meaningful interactions with the content and other users. Without the ability to comment, users are unable to ask questions for clarification, provide feedback, or share insights about the materials. This limitation restricts collaborative learning and impedes the development of a supportive academic community. The lack of interactive elements like comments can lead to a less dynamic and engaging user experience, diminishing the overall value of the platform for students seeking a more collaborative and interactive learning environment.

Another notable drawback is the absence of a text extraction feature on these platforms such as [6], [7], [8], [9] and [10]. Users cannot easily convert and manipulate text from documents for personal use or further study, such as converting images into editable and searchable text formats. This limitation means that users cannot easily search or edit content presented in image or video formats, potentially reducing the utility and accessibility of the resources provided. This lack of functionality can hinder users' ability to interact with and utilize the materials effectively.

Furthermore, the current platforms do not offer automated content classification for user-uploaded materials such as [6], [7], [8], [9] and [10]. They lack the ability to automatically categorize or filter different types of content, such as videos, documents, and images. This absence of automation requires users to manually sort and organize their uploads, leading to a disorganized and less efficient user experience. The inability

to automatically classify and categorize content into relevant course categories can impede users from finding and utilizing the materials effectively.

In addition, platforms such as [6], [7], [8], [9], and [10] do not include a Smart Calendar Planner. The absence of Smart Calendar Planner prevents students from efficiently organizing their study schedules and tracking deadlines. Furthermore, platforms such as [8] and [10] lack flashcards, quiz generators, and AI chatbot that allow users to ask questions about uploaded content. This lack of interactive study tools and AI chatbot features limits students' ability to actively engage with the learning materials and customize their study experiences.

For platforms that offer flashcard or quiz generation, access is often limited. Platforms such as [9] allow users to generate flashcards and quizzes from uploaded content, but these features are locked behind a subscription plan, requiring users to pay in order to unlock the features. Similarly, for platform [7], flashcards are only available to users who upgrade to a premium plan. These restrictions make it harder for students to fully use the platforms' study tools without paying for a subscription.

In summary, these limitations, including restricted access due to subscription-based models, the absence of interactive features like commenting, insufficient text extraction capabilities, and the requirement for manual content classification, significantly affect the usability and effectiveness of educational resource-sharing platforms. Addressing these challenges could enhance user experience, making learning environments more flexible, accessible, and collaborative for students.

2.3 Summary of the Existing System

Platform	Advantages	Weaknesses
Course Hero [7]	• Extensive library of educational materials. • 24/7 Homework Help with tutors. • Interactive quizzes for self-assessment. • Detailed textbook solutions and explanations. • Grammar checker tool.	• Subscription-based model restricts access. • Requires uploading 10 notes to unlock content. • No commenting feature for user interactions. • No Smart Calendar Planner. • Flashcards and quizzes are only available after premium upgrade.
Scribd [8]	• Diverse document types. • Multi-language support. • Document upload and sharing	• Subscription-based models limit access to many resources. • No commenting or interactive features for engagement. • No automated text extraction from uploaded documents. • No automated content classification. • No flashcard or quiz generation. • No AI chatbot. • No Smart Calendar Planner.
Docsity [9]	• Revenue sharing for contributors. • Ask Community feature. • User ratings feature.	• Subscription-based model. • No text extraction features. • No automated content classification. • Flashcard and quiz generation locked behind subscription. • No Smart Calendar Planner.

Studocu [6]	• Study groups feature. • AI-powered instant study help. • Study tools like flashcards and quizzes.	• Subscription-based model limits access and premium features. • No automated classification of content uploads. • No Smart Calendar Planner.
OneClass [10]	• Community Q&A for academic questions. • Access to class notes from peers. • Textbook solutions with step-by-step explanations.	• Subscription required for full access to resources. • No text extraction feature. • No content classification automation. • No flashcard or quiz generation. • No AI chatbot. • No Smart Calendar Planner.

Table 2.1 Review on the Advantages and Weaknesses of Existing Platforms

2.4 Proposed Solutions

This project aims to propose a set of innovative solutions to address the limitations found in existing educational resource-sharing platforms. The focus is on enhancing user experience, improving accessibility, and fostering a more interactive and efficient learning environment. The proposed solutions include:

1. **Enhanced Access and Usability**: To mitigate the issues associated with subscription-based models, the project proposes a tiered access system that offers free features. This approach ensures that the functionalities are available to all users. This model aims to lower barriers for students and improves overall resource utilization.

2. **Interactive Features for Engagement**: Introducing interactive features such as commenting to address the lack of user engagement on current platforms. These features will enable users to ask questions, provide feedback, and engage in collaborative learning, thereby enhancing the educational value and fostering a supportive academic community.

3. **Text Extraction and Transcription**: To address the absence of text extraction capabilities, the project proposes integrating advanced tools for text extraction and transcription. These tools will allow users to convert images and videos into editable and searchable text formats, enhancing the flexibility and usability of the resources provided. This functionality will enable users to interact with the content more effectively.

4. **Automated Content Classification**: Implementing automated content classification features will streamline the organization of user-uploaded materials. The system will automatically categorize and filter content such as videos, documents, and images into appropriate course categories. This automation will improve the efficiency of content management and enhance users' ability to find and utilize relevant materials.

5. **Enhanced User Feedback Mechanisms**: To address the current lack of comprehensive feedback mechanisms, the project proposes integrating features for user ratings, reviews, and comments on uploaded materials. This will allow users to share insights, rate the quality of resources, and provide constructive feedback. Enhanced feedback mechanisms will improve the overall quality of shared resources and support a more interactive learning environment.

6. **Smart Calendar Planner Integration**: To address the absence of a Smart Calendar Planner, the project proposes integrating a calendar and scheduling feature that allows students to organize their study schedules and track deadlines efficiently. This feature will help students improve time management and ensure timely completion of tasks.

7. **Interactive Study Tools**: To address the lack of flashcards, quizzes, and AI chatbot, the project proposes integrating tools to generate flashcards and quizzes from user-uploaded content. An AI chatbot will also be implemented to answer questions about uploaded materials, enhancing engagement and supporting personalized learning.

By implementing these solutions, the project aims to create a more inclusive, interactive, and user-friendly educational resource-sharing platform that better meets the needs of students and enhances their learning experience.

Chapter 3

System Methodology/Approach

3.1 System Design Diagram

3.1.1 System Architecture Diagram

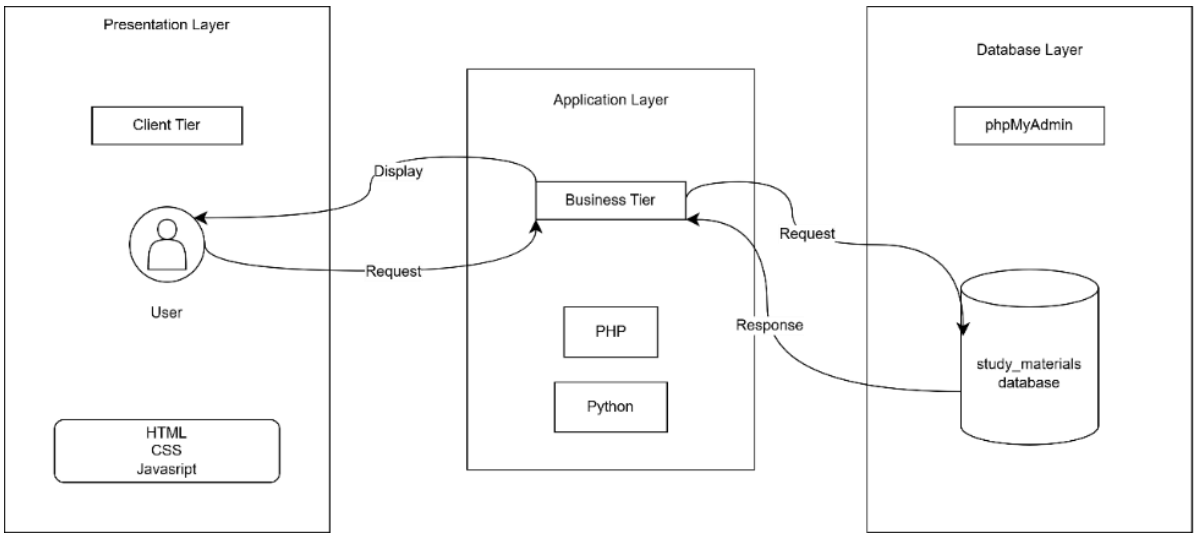


Figure 3.1.1 System Architecture Diagram

Figure 3.1.1 shows the structure of the Student Resource Exchange System. This system is organized into three main layers: Presentation, Application, and Database Layer. The Presentation Layer is where the users directly interact with the system. This presentation layer is developed using HTML, CSS, and JavaScript to create a user-friendly web interface. In this interface, users can perform actions such as clicking buttons or submitting forms. After performing action, requests will be generated and sent to the Application Layer. The results will be returned and shown to the user in this Presentation Layer once the requests are processed.

The Application Layer functions as the core logic of the system. This layer uses PHP to handle backend operations such as user registration, login, file uploads, and handling study materials. In addition, Python is integrated to support advanced features aligned with the project objectives. The features include Optical Character Recognition (OCR) which is used to convert images into editable text, video summarization which is used to generate transcripts and summaries, and automated course classification using machine learning algorithms. This layer also manages interactive features such as the commenting system with sentiment analysis and hate speech detection. The Smart

Calendar Planner allows AI-Scheduled task creation and conflict detection to avoid overlapping tasks. Other than that, the AI Study Tools use Natural Language Processing (NLP) and Large Language Models (LLMs) to generate flashcards, quizzes and an AI chatbot used for answering user questions. Overall, the Application Layer handles user requests, executes algorithms, integrates external APIs, ensures data integrity, and delivers meaningful responses to the Presentation Layer.

The Database Layer serves as the system's storage unit and is managed using phpMyAdmin. This layer contains a database where all important data is stored, such as user accounts, uploaded study materials, extracted text from images, video summaries, scheduled task, quizzes, flashcards and so on. The Database Layer is essential because it ensures that student resources are kept safe and available even if users log out or the system restarts. The arrows shown in the diagram represent the flow of information: "Request" arrows indicate actions initiated by the user, while "Response" and "Display" arrows represent the processed data returned to the user interface.

3.1.2 Use Case Diagram

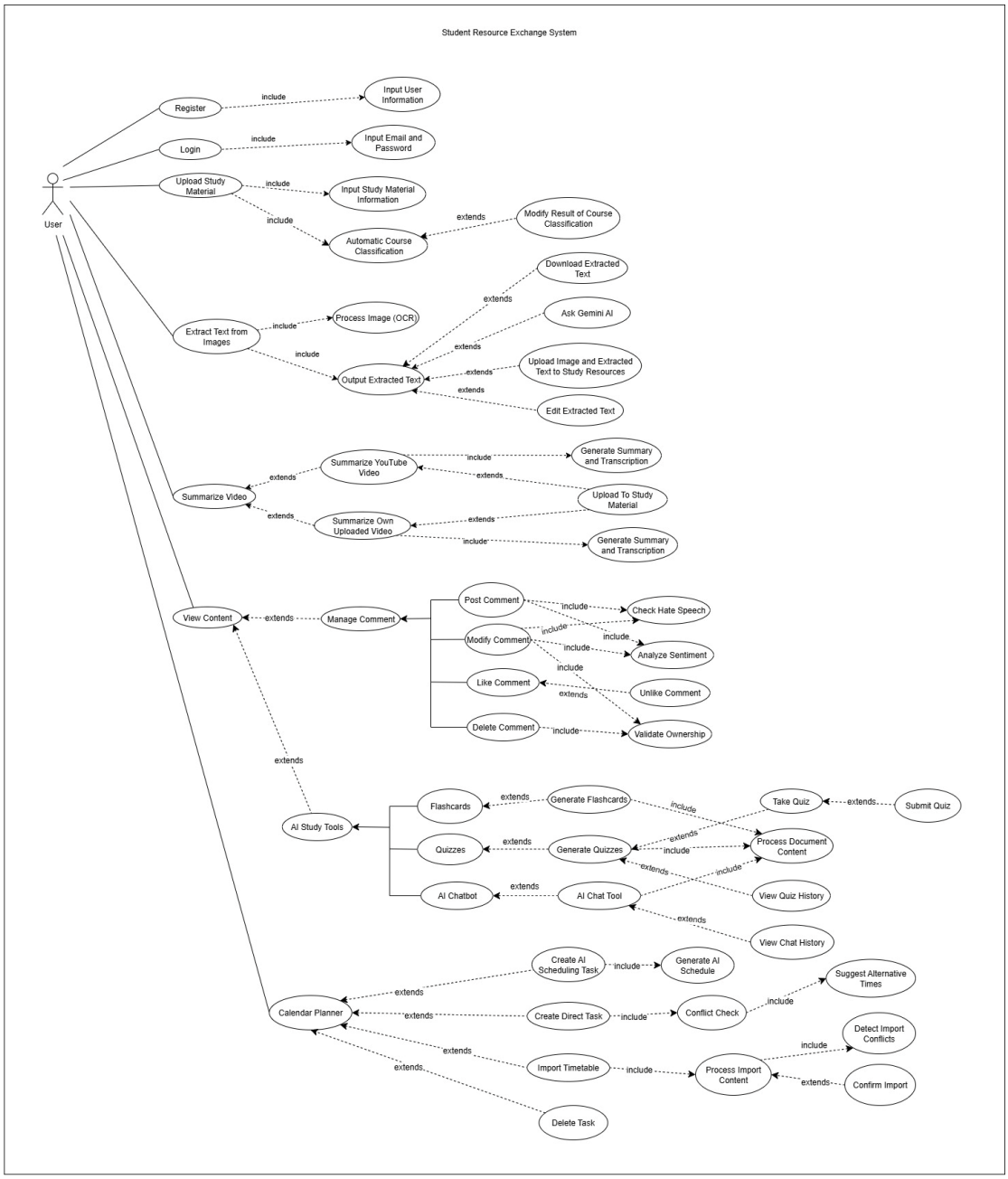


Figure 3.1.2 Use Case diagram

3.1.3 Use Case Description

Use Case Name	Register	Use Case ID	UC01
Primary Actor	User		
Brief Description	Allows a new user to create an account in the system.		
Trigger	User selects the "Register" option on the login page.		
Normal Flow of Events	1. System displays the registration form. 2. User provides required information. 3. System validates the information. 4. System creates a new user account. 5. System confirms successful registration to the user.		
Sub-flows	2a: Input User Information 1. System prompts for username, email, password, and confirm password. 2. User inputs the required information. 3. System validates input format and uniqueness.		
Exception Flows	E1: Email Already Exists 1. System detects email is already registered. 2. System displays error message. 3. User must use different email address. E2: Invalid Input Format 1. System detects invalid email format or weak password. 2. System highlights invalid fields. 3. User corrects the information.		

Table 3.3.1 Use Case Description for "Register"

Use Case Name	Login	Use Case ID	UC02
Primary Actor	User		
Brief Description	Allows an existing user to authenticate and access the system.		
Trigger	User selects the "Login" option.		
Normal Flow of Events	1. User selects the "Login" option. 2. System displays the login form. 3. User inputs their email and password. 4. User submits the login form. 5. System validates the credentials. 6. System grants access and redirects the user to the homepage.		
Sub-flows	2a: Input Email and Password 1. System prompts for email and password. 2. User enters email and password. 3. System validates user input.		
Exception Flows	E1: Invalid Email or Wrong Password 1. System detects incorrect email or password. 2. System displays error message. 3. User re-enters email or password. E2: Forgot Password 1. User selects the "Forgot Password" option. 2. System prompts the user to enter their email address. 3. System sends a password reset link to the provided email. 4. User clicks the reset link and creates a new password. 5. User returns to the login page to authenticate with the new password.		

Table 3.3.2 Use Case Description for "Login"

Use Case Name	Upload Study Material	Use Case ID	UC03
Primary Actor	User		
Brief Description	Allows user to upload study materials to the system.		
Trigger	User selects the "Upload Study Material" option.		
Normal Flow of Events	1. System displays an upload interface. 2. User selects files to upload. 3. User enters details for the uploaded materials. 4. System processes file and extracts content information. 5. System sends content to AI classification service. 6. System displays AI classification results and alternative suggestions. 7. User reviews the material details. 8. User submits the upload form. 9. System validates all required fields and processes the upload 10. System stores the uploaded materials. 11. System confirms successful upload.		
Sub-flows	5a: Automatic Course Classification 1. System analyzes uploaded content. 2. AI service suggests appropriate course categories. 3. System presents classification results to user.		
Alternative Flows	7a: Modify Result of Course Classification 1. User reviews AI classification suggestions. 2. User can accept or modify the course suggestions. 3. System updates classification based on user input.		

Exception Flow	E1: AI Classification Service Unavailable 1. System displays warning about unavailable classification service. 2. System enables manual course selection without AI assistance. 3. User manually selects appropriate course from dropdown. E2: Missing Required Information 1. System highlights missing required fields. 2. Users must complete all required information before submission.

Table 3.3.3 Use Case Description for "Upload Study Material"

Use Case Name	Extract Text from Image	Use Case ID	UC04
Primary Actor	User		
Brief Description	Allows user to extract text content from images.		
Trigger	User selects the "OCR Tool" option.		
Normal Flow of Events	1. System displays image upload interface. 2. User uploads image. 3. System processes the image. 4. System displays extracted text. 5. User reviews extracted text. 6. System provides options for further actions.		
Sub-flows	3a: Process Image (OCR) 1. System applies OCR algorithms to the uploaded image. 2. System converts recognized text to editable text. 4a: Output Extracted Text 1. System displays the extracted text in editable format. 2. System provides options for user to save or further process the text.		

Alternative Flows	5a: Ask Gemini AI 1. User enters questions and send to Gemini AI. 2. System connects to Gemini AI. 3. AI provides answers for the extracted text. 5b: Upload Image and Extracted Text to Study Resources 1. User chooses to save the image and extracted text as study resources. 2. System prompts for material details. 3. System saves the image and text to the user's study resources. 5c: Edit Extracted Text 1. User selects to edit the extracted text. 2. System provides text editing interface. 3. User makes desired changes. 4. System saves the edited text.

Table 3.3.4 Use Case Description for "Extract Text From Image"

Use Case Name	Summarize Video	Use Case ID	UC05
Primary Actor	User		
Brief Description	Allows user to generate summaries of video content.		
Trigger	User selects video summarization option.		
Normal Flow of Events	1. System displays video summarization interface. 2. User provides video to summarize. 3. System processes the video content. 4. System generates summary and transcript. 5. System displays the result. 6. User reviews the generated content.		
Sub-flows	3a: Generate Summary and Transcription 1. System extracts audio from the video. 2. System converts audio to text. 3. System analyzes text to generate a concise summary. 4. System displays both the transcription and summary.		
Alternative Flows	2a: Summarize YouTube Video 1. User provides a YouTube URL. 2. System validates the URL. 3. System processes the YouTube video content. 4. System generates summary and transcript. 2b: Summarize Own Uploaded Video 1. User uploads a video file. 2. System processes the uploaded video. 3. System generates summary and transcript. 5a: Upload To Study Material 1. Users choose to save the video summary as study material. 2. System prompts for material details. 3. System saves the video summary to the study materials.		

Exception Flows	E1: Invalid YouTube URL 1. System cannot access the provided YouTube URL. 2. System displays error message 3. User provides correct YouTube URL.

Table 3.3.5 Use Case Description for "Summarize Video"

Use Case Name	View Content	Use Case ID	UC06
Primary Actor	User		
Brief Description	Allows users to view study materials, manage comments, and access AI Study Tools.		
Trigger	User selects a study material or content item to view.		
Normal Flow of Events	1. System displays the requested study material. 2. System shows associated comments with sentiment indicators. 3. User views the content and available features. 4. User can interact with content through various actions.		
Sub-flows	2a: Manage Comments 1. User chooses to post, like, modify, or delete comments. 2. System performs hate speech detection and sentiment analysis on comments. 3. System processes approved comments. 4. System updates new comment information. 2b: Post Comment 1. User writes a new comment. 2. System performs hate speech detection. 3. System performs sentiment analysis 4. System saves approved comment. 5. System displays comment with sentiment indicator. 2c: Like Comment 1. User clicks like button on existing comment. 2. System records the like interaction. 3. System updates like count display. 2d: Modify Comment 1. System validates comment ownership 2. User clicks on edit button on their own comment. 3. System allows comment editing. 4. User makes changes and saves. 5. System re-analyzes modified comment for hate speech and sentiment.		

	2e: Delete Comment 1. System validates comment ownership. 2. User clicks delete button on their own comment. 3. Systems confirm deletion action and remove comment.
Alternative Flows	2f: Unlike Comment 1. User clicks unlike button on previously liked comment. 2. System removes like interaction. 3. System updates like count display. 3a: AI Study Tools Access 1. User selects AI study tools options. 2. System provides access to flashcards, quizzes, or AI chat tools. 3. User interacts with AI study tool.
Exception Flows	E1: Comment Contains Hate Speech 1. System detects offensive content in comment. 2. System blocks comment submission. 3. System displays error message. 4. User re-enter the comment.

Table 3.3.6 Use Case Description for "View Content"

Use Case Name	Calendar Planner Create Task	Use Case ID	UC07
Primary Actor	User		
Brief Description	Allows users to create and manage study tasks in their personal calendar planner.		
Trigger	User clicks on “Calendar Planner” button.		
Normal Flow of Events	1. System displays the calendar planner interface. 2. User selects option to create a new task. 3. System displays task creation form. 4. User enters task details. 5. System validates the task information. 6. System saves the task to user’s calendar. 7. System displays successful task creation message.		
Sub-flows	4a: Create AI Scheduling Task 1. User clicks on “Create Task” button. 2. System analyzes user’s existing schedule. 3. AI suggests suitable time slots for the task. 4. User reviews and accepts or modifies AI suggestions. 2a: Create Direct Task 1. User clicks on a timeslot column in the calendar view. 2. System checks for schedule conflicts. 3. System allows direct scheduling if no conflicts exist. 4b: Import Timetable 1. User clicks on button “Import Timetable” to import existing timetable. 2. System displays import form. 3. User provides timetable data. 4. System processes and analyzes imported content. 5. System displays extracted schedule information. 6. User clicks on “Import Schedule” to confirm import. 7. System processes and imports the schedule.		

Alternative Flows	4c: Delete Task 1. User selects existing task to delete. 2. System confirms deletion action. 3. System removes task and associated sessions 2b: Suggest Alternative Times 1. System analyzes available time slots. 2. System generates alternative scheduling options. 3. System presents suggestions to user. 4. User selects preferred alternative.
Exception Flows	E1: Schedule Conflict 1. System detects conflicting tasks at same time. 2. System highlights the conflict to user. 3. System prevents conflicting schedule creation. 4. System automatically suggests alternative times. 5. User must choose different timeslot or resolve conflict.

Table 3.3.7 Use Case Description for "Calendar Planner Create Task"

Use Case Name	AI Study Tools	Use Case ID	UC08
Primary Actor	User		
Brief Description	Allow user to generate flashcards, quizzes, and ask question with AI chatbot.		
Trigger	User selects AI Study Tools from content view.		
Normal Flow of Events	1. User accesses AI Study Tools. 2. System processes document content for AI analysis. 3. System displays AI Study Tools. 4. User selects specific tool (Flashcards/Quiz/Chat). 5. System presents interactive study interface. 6. User engages with AI-generated study content.		
Sub-flows	3a: Generate Flashcards 1. System analyzes document content. 2. AI extracts key concepts and creates question-answer pairs. 3. System presents interactive flashcard interface. 4. User studies using flashcard feature. 3b: Generate Quizzes 1. System processes document content. 2. AI creates various question types. 3. System presents quiz interface with questions. 4. User takes quiz. 3c: AI Chatbot 1. System displays AI chat interface. 2. User asks questions about document content. 3. AI provides answers based on document. 4. System stores conversation history.		
Alternative Flow	5a: Take Quiz 1. User starts generated quiz. 2. System presents questions one by one. 3. User submits answers. 4. System provides immediate feedback and scoring.		

	5b: Submit Quiz 1. User completes all quiz questions. 2. System calculates final score. 3. System saves quiz results. 4. System provides performance analysis. 5c: View Quiz History 1. User accesses quiz history section. 2. System displays past quiz attempts with scores. 3. User can review previous performance.

Table 3.3.8 Use Case Description for "AI Study Tools"

3.1.4 Activity Diagram

UC01 – Register Activity Diagram

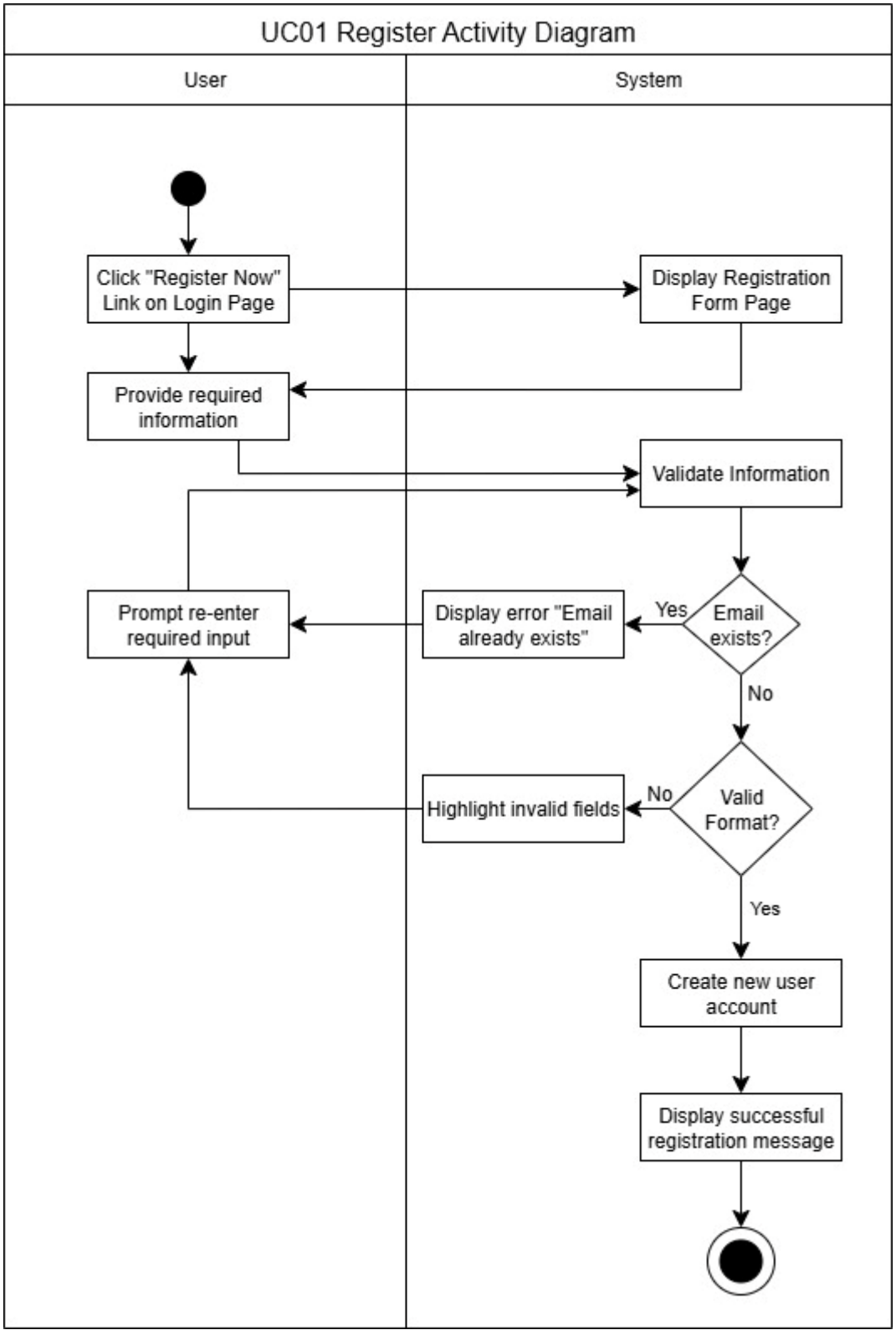


Figure 3.1.3.1 UC01 Register Activity Diagram

UC02 - Login Activity Diagram

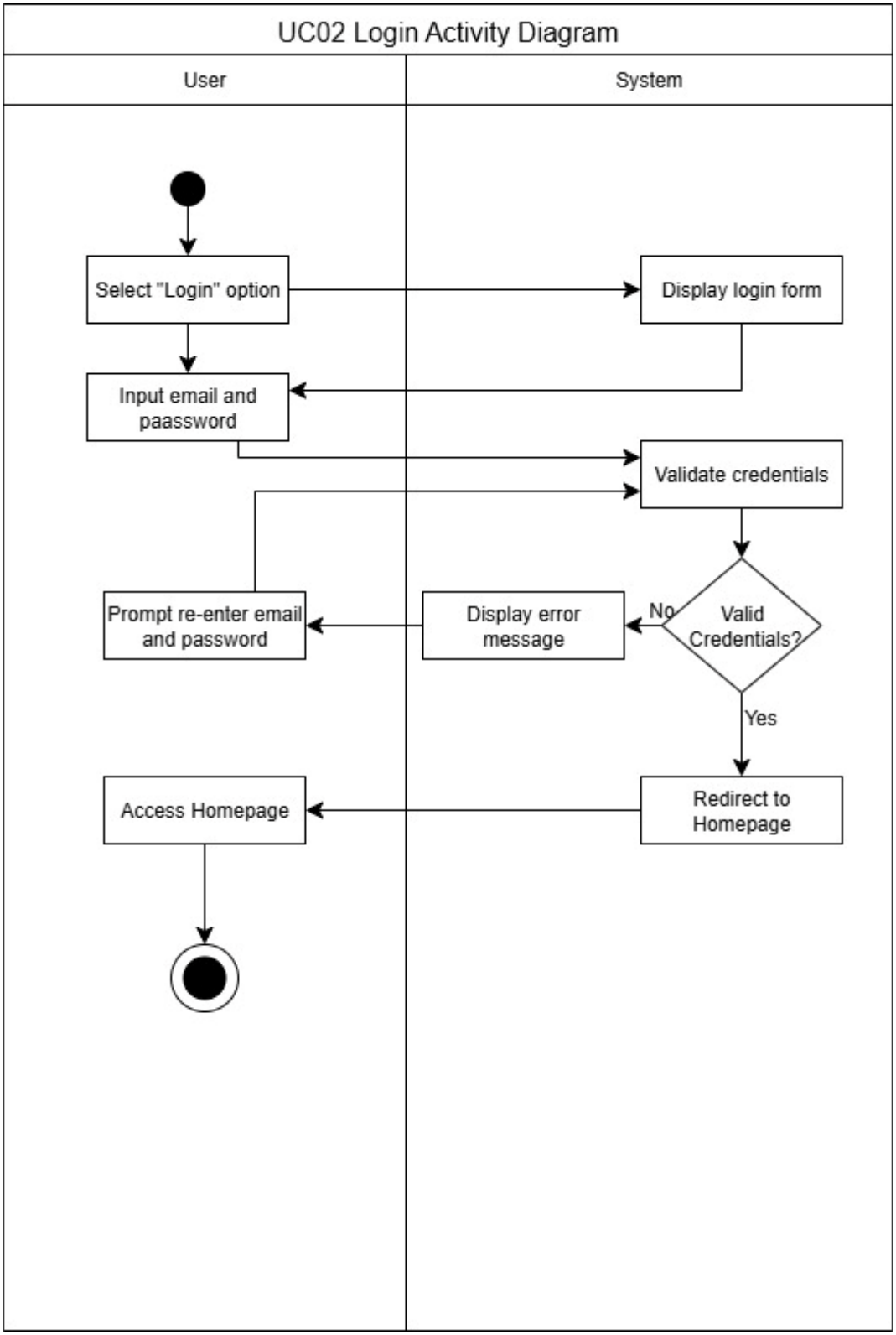


Figure 3.1.3.2 UC02 Login Activity Diagram

UC03 – Upload Study Material Activity Diagram

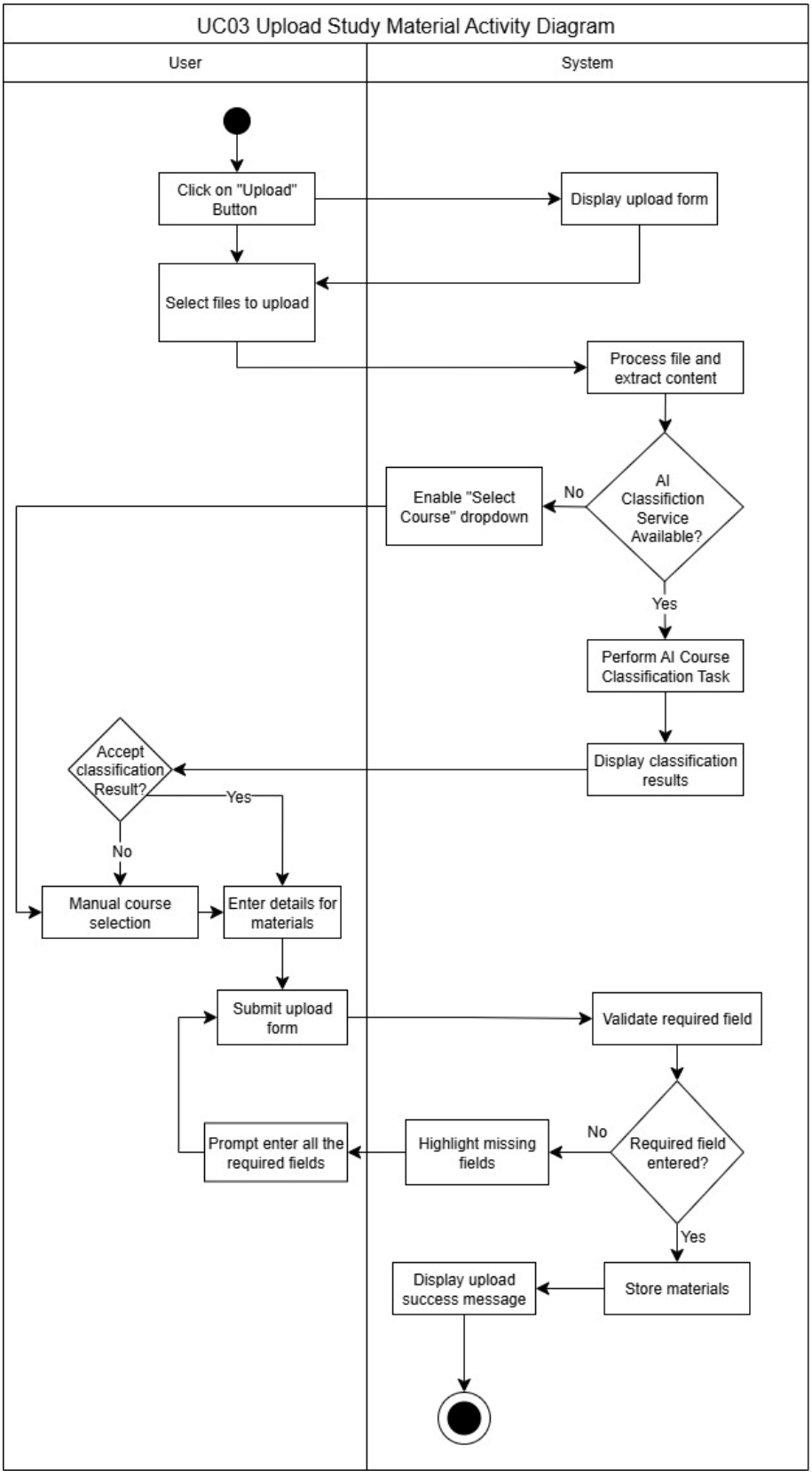


Figure 3.1.3.3 UC03 Upload Study Material Activity Diagram

UC04 – Extract Text from Image Activity Diagram

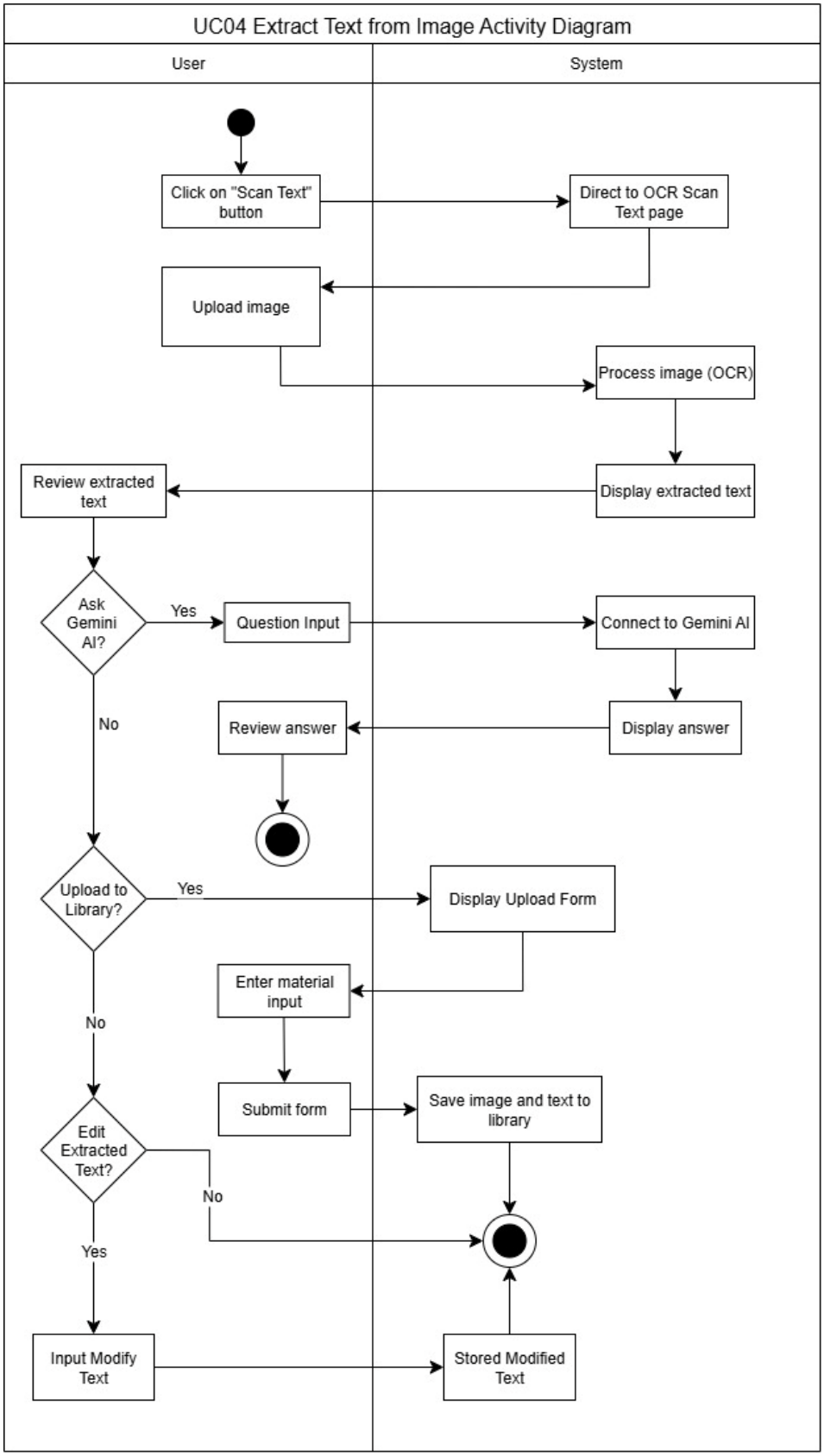


Figure 3.1.3.4 UC04 – Extract Text from Image Activity Diagram

UC05 – Summarize Video Activity Diagram

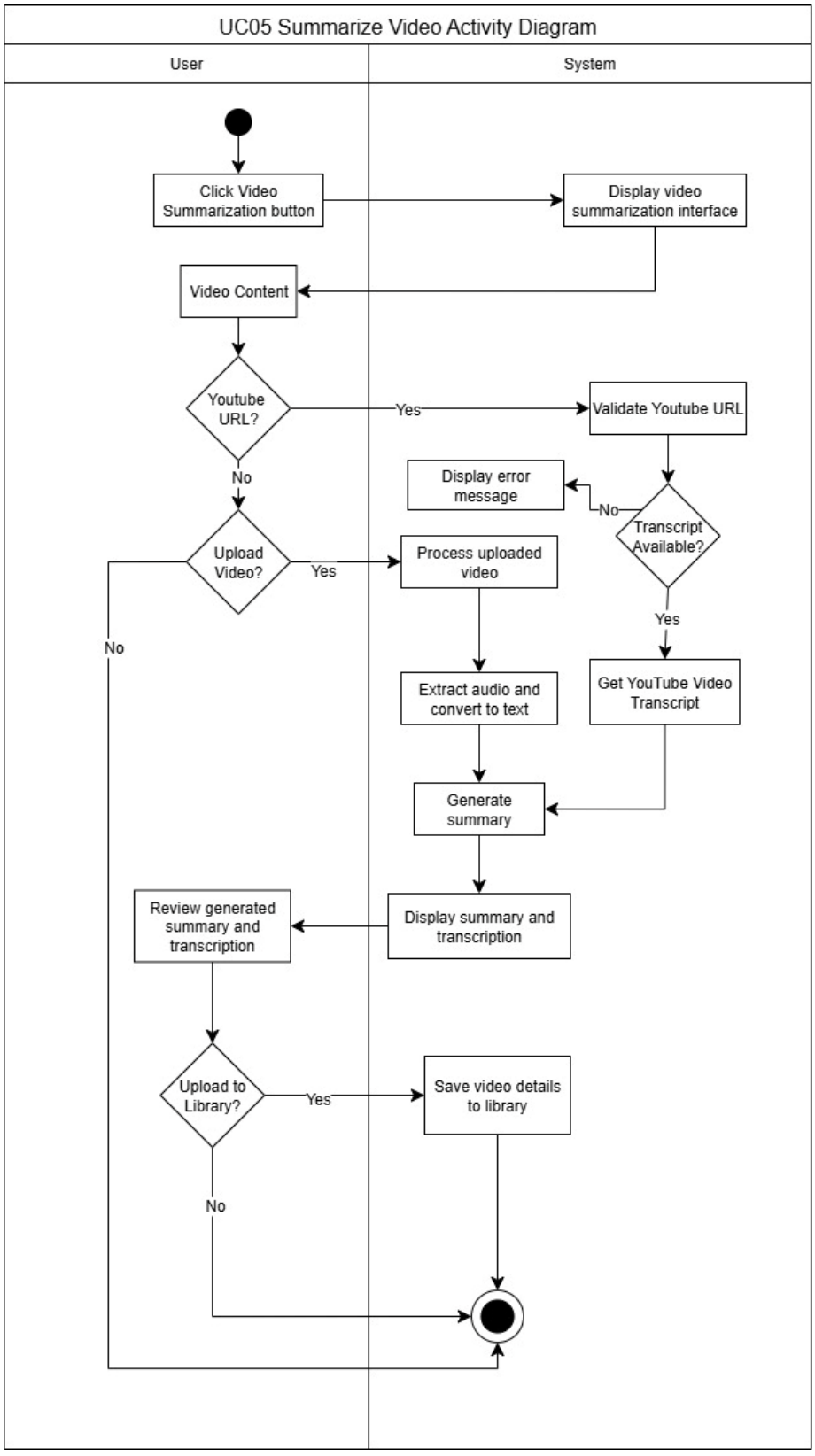


Figure 3.1.3.5 UC05 – Summarize Video Activity Diagram

UC06 – View Content Activity Diagram

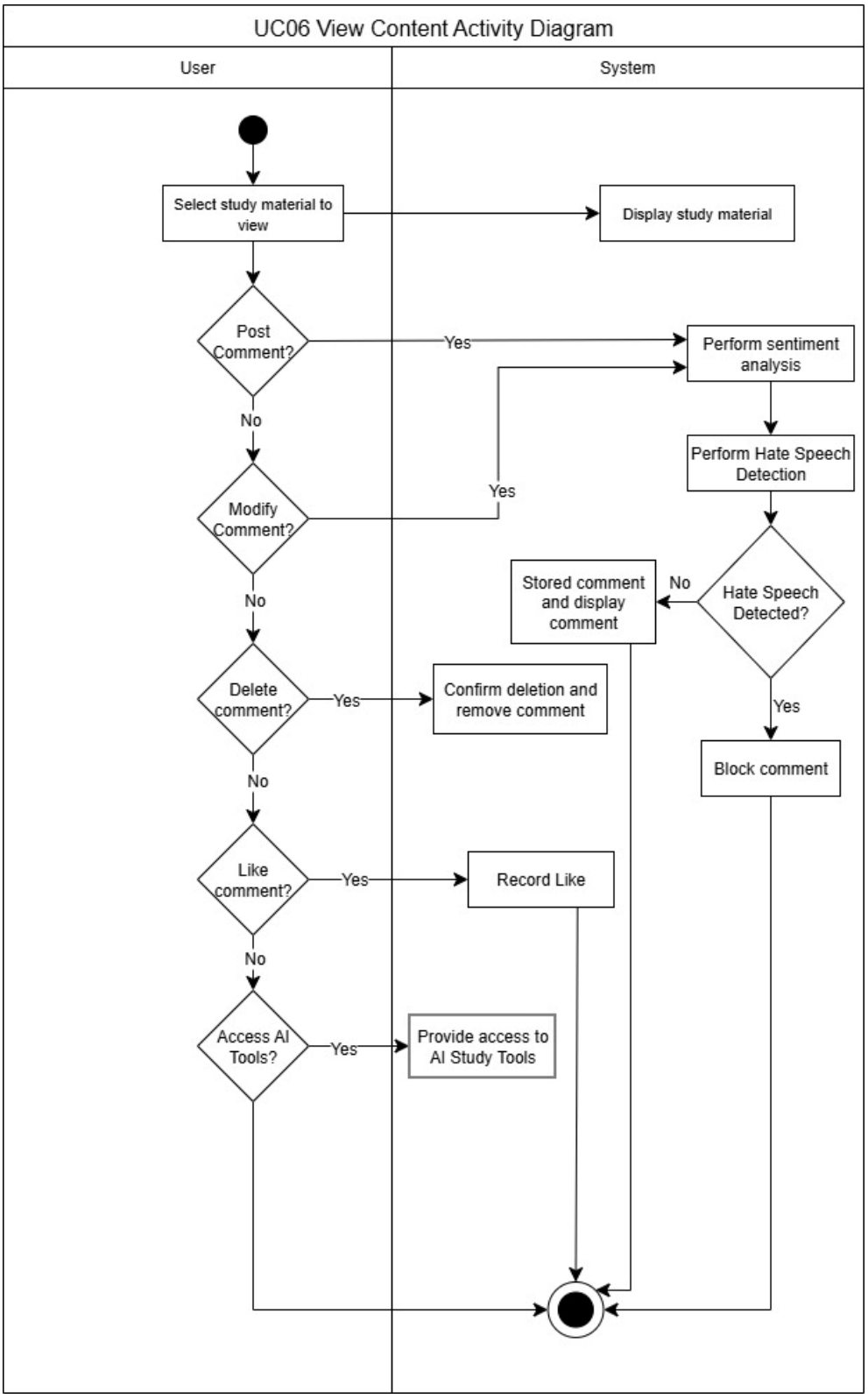


Figure 3.1.3.6 UC06 – View Content Activity Diagram

UC07 – Calendar Planner Activity Diagram

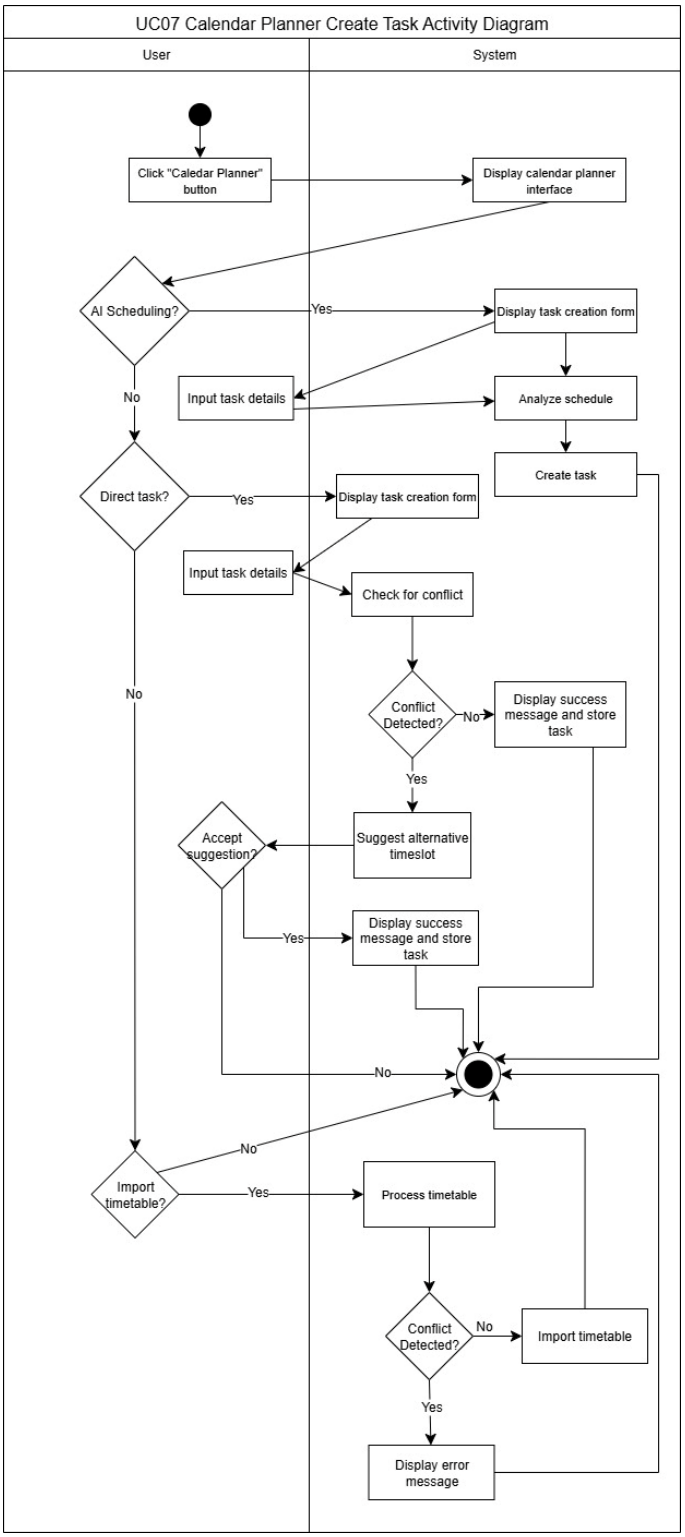


Figure 3.1.3.7 UC07 – Calendar Planner Activity Diagram

UC08 – AI Study Tools Activity Diagram

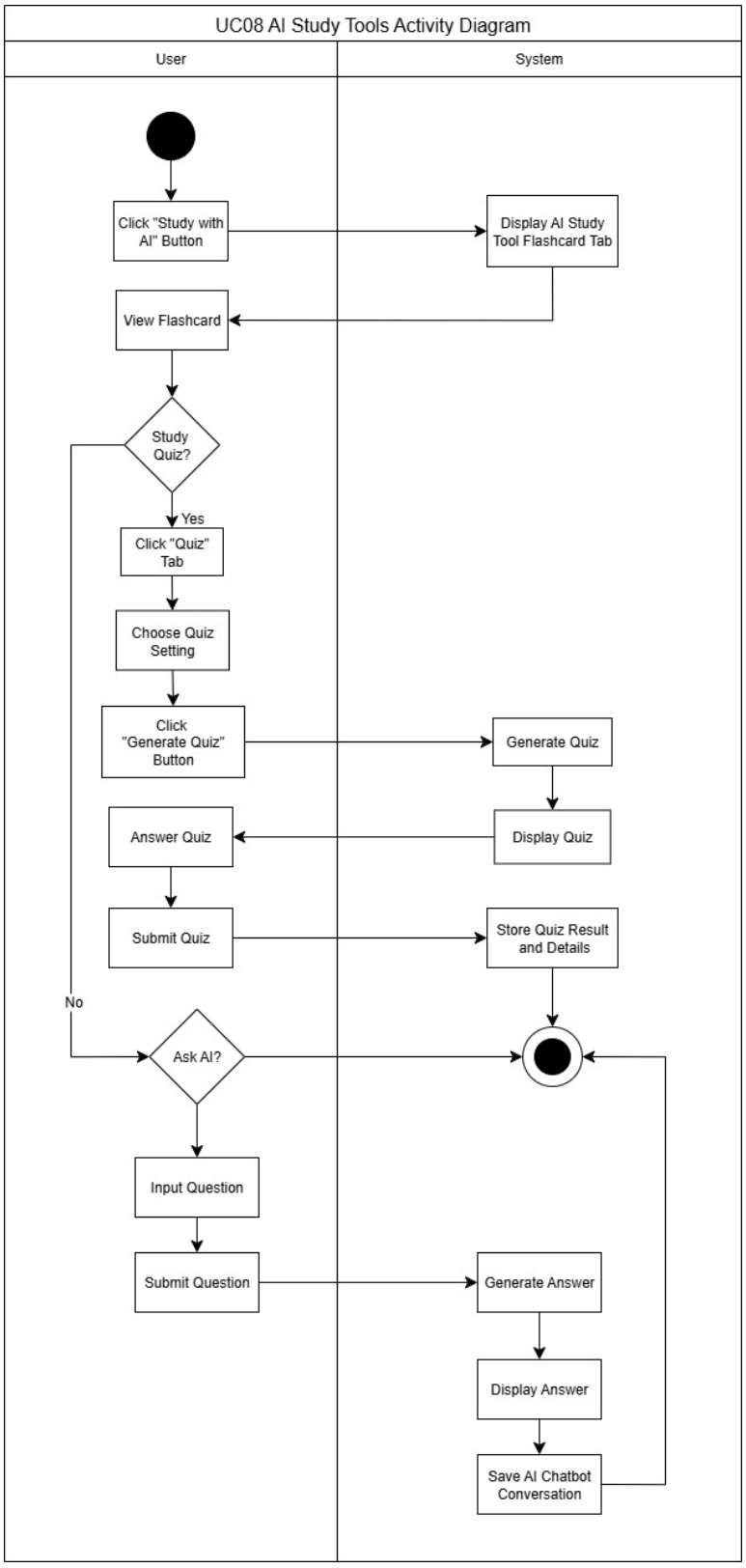


Figure 3.1.3.8 UC08 – AI Study Tools Activity Diagram

Chapter 4

System Design

4. 1 System Flowchart

4.1.1 System Flowchart for Login and Registration

Figure 4.1.1 shows the system flowchart for Login and Registration Modules. At first, users need to register an account if they are new users and have not created an account yet. To create an account, users need to fill in all the required information in the Registration page form. After successfully creating an account, the users' information will be stored in the study_materials database. If users have registered an account but forgot the password, then they need to go to the Forgot Password page and enter the registered email address. After that, the user will receive a password reset email, and they are required to fill in the new password, the reset password will be stored in the study_materials database after submitting the new password. In order to log in to the website, users are required to input their email address and password, and the system will check whether the email address exists in the database or not. If yes, then the system will further verify whether the provided password is correct or wrong. If both the email address and password are correct, then the user will be directed to the Homepage. Else, an error message will be displayed to the user, and they are required to re-enter the correct credentials.

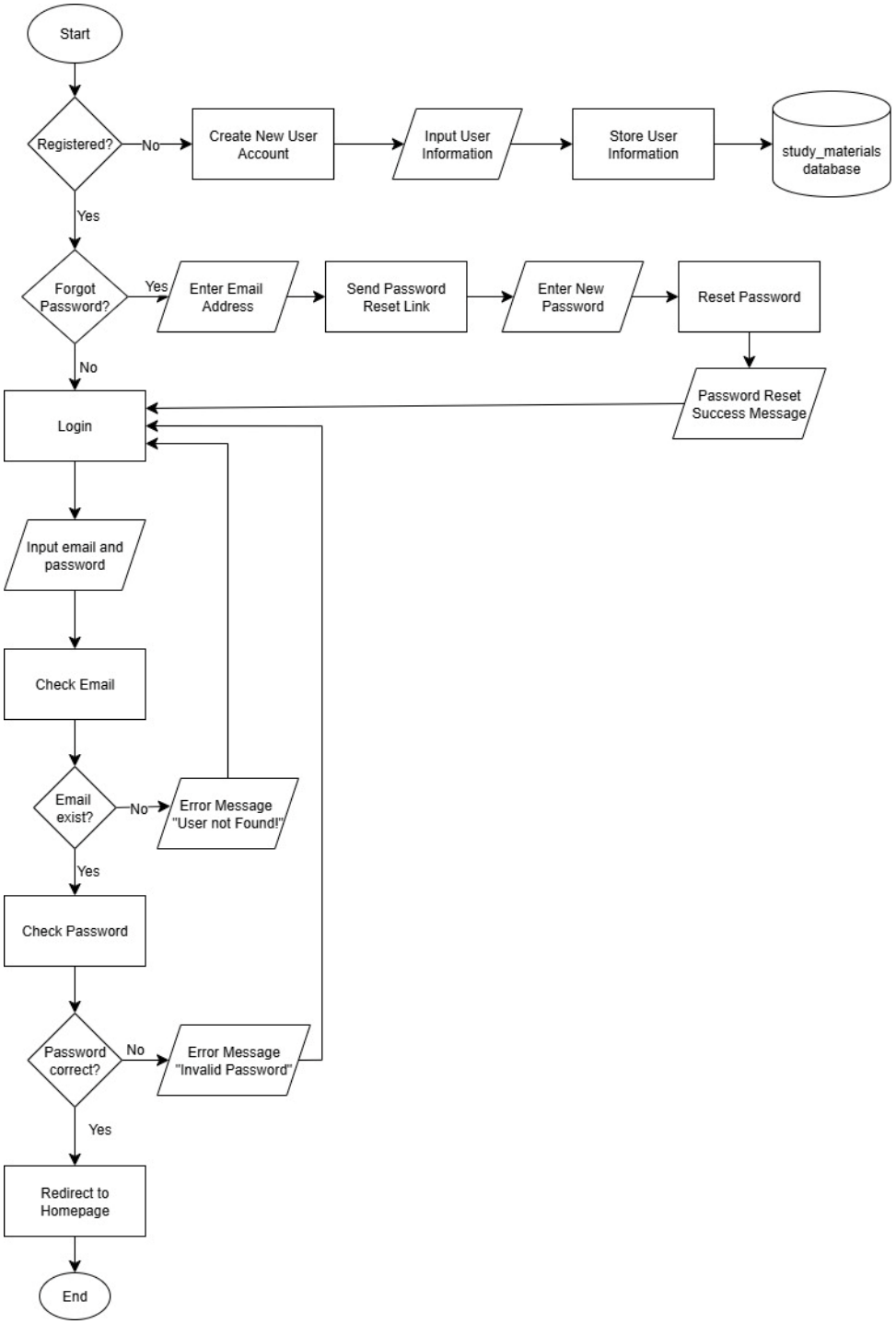


Figure 4.1.1 System Flowchart for Login and Registration

4.1.2 System Flowchart for Upload Material

Figure 4.1.2 shows the system flowchart for Upload Material. To upload a material, the system will first display the upload material form. After that, the users need to fill in all the required material information. Then the system will perform the AI Course Classification Task, which will automatically classify the user-uploaded material into its corresponding course. The AI Course Classification result will then be displayed to the users. If the user is satisfied with the AI Classification result, then they can submit the upload form, else the users need to manually select the course and submit the upload form. After the user submits the upload form, the system will process the upload form and store the material in the study_materials database.

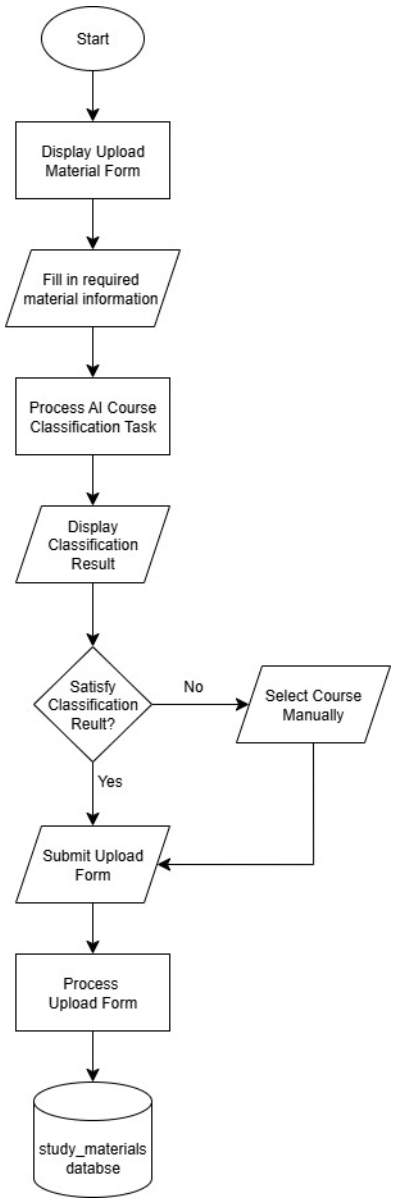


Figure 4.1.2 System Flowchart for Upload Material

4.1.3 System Flowchart for Comment System

Figure 4.1.3 shows the system flowchart for the commenting system. The system will first display the view content comment interface and users can perform multiple actions on this page. If users want to post a comment, then they need to input the comment into the comment field and click the submit button. After that, the system will analyze the comment sentiment and detect comment hate speech. If no hate speech is detected, then the system will store the comment in the study_materials database, else the system will display an error message and the comment cannot be posted. The user can also like the comment by clicking the “Like” button. After clicking the “Like” button, the system will then update the like count and save it into the study_materials database. In addition, the users can also delete their own comment by clicking the “Delete” comment button. After clicking the “Delete” button, a confirmation message will appear. If the user confirms to delete the comment, the system will then remove the comment from the study_materials database. Users can also click the “Edit” button to edit their own comment. After clicking the “Edit" button, the users need to input the modified comment and click the “Submit Button”. The system will then re-analyze the modified comment with sentiment analysis and hate speech detection. If no hate speech is detected, the comment will be successfully modified and saved in the study_materials database, else an error message will be displayed.

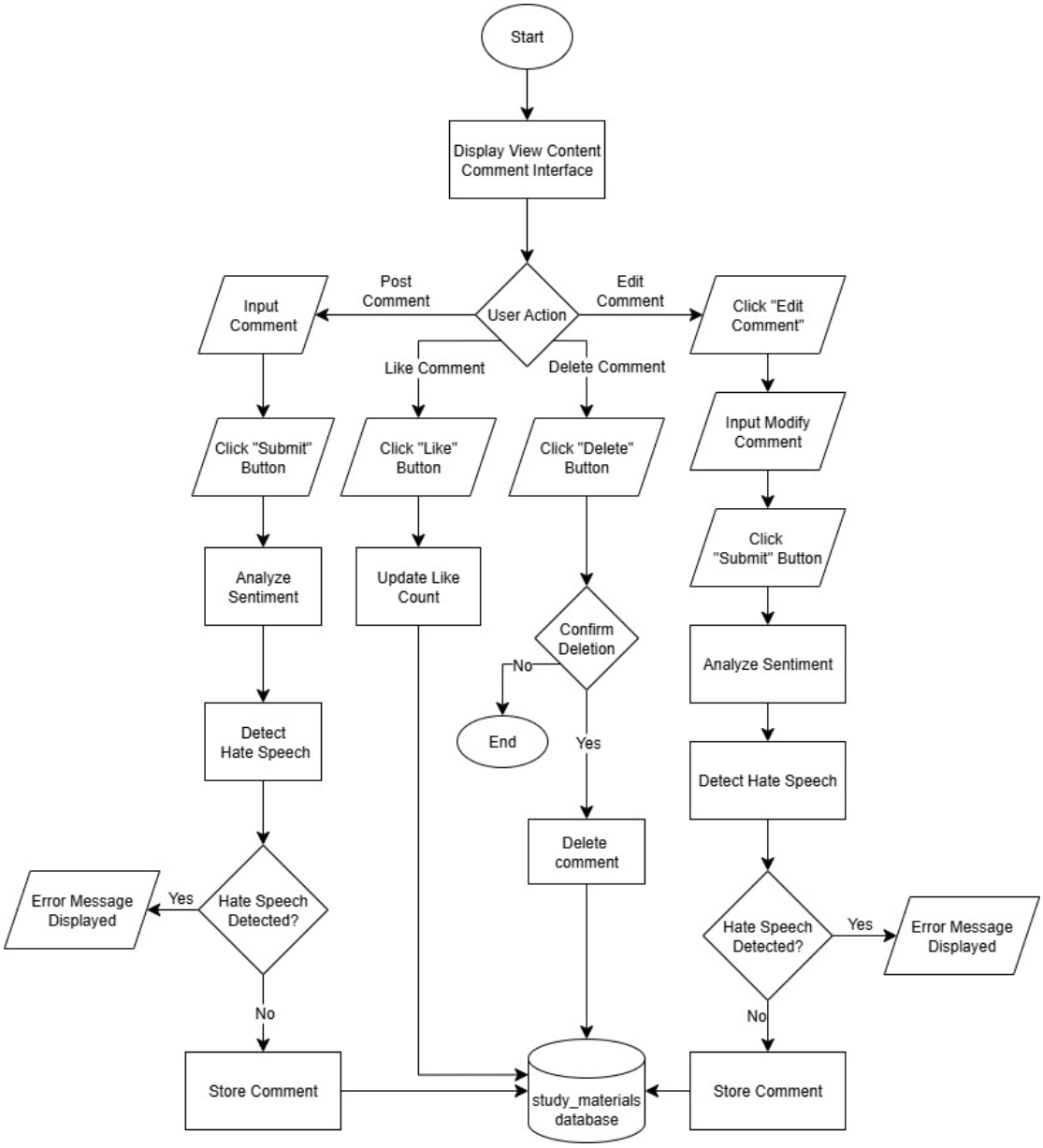


Figure 4.1.3 System Flowchart for Comment System

4.1.4 System Flowchart for Image-to-Text (OCR)

Figure 4.1.4 shows the system flowchart for Image-to-Text (OCR). At first, the system will display the Image-to-Text page. After that, users need to upload an image with text to extract. After uploading an image, the system will process the image and display the extracted text result to users. After successfully extracting the text, the user can perform several actions. If users want to upload the image with extracted text to the Library page, then they need to click the “Upload Material” button. After that, the system will direct the user to an upload form with image and extracted text preview. Since the image and extracted text are the material to upload, the AI Course Classification Task will perform automatically and display the classification result to the users. If users are satisfied with the AI Classification result, then they can submit the upload form after filling in all the required material information, else the users need to manually select the course, and the system will store the material in the study_material database.

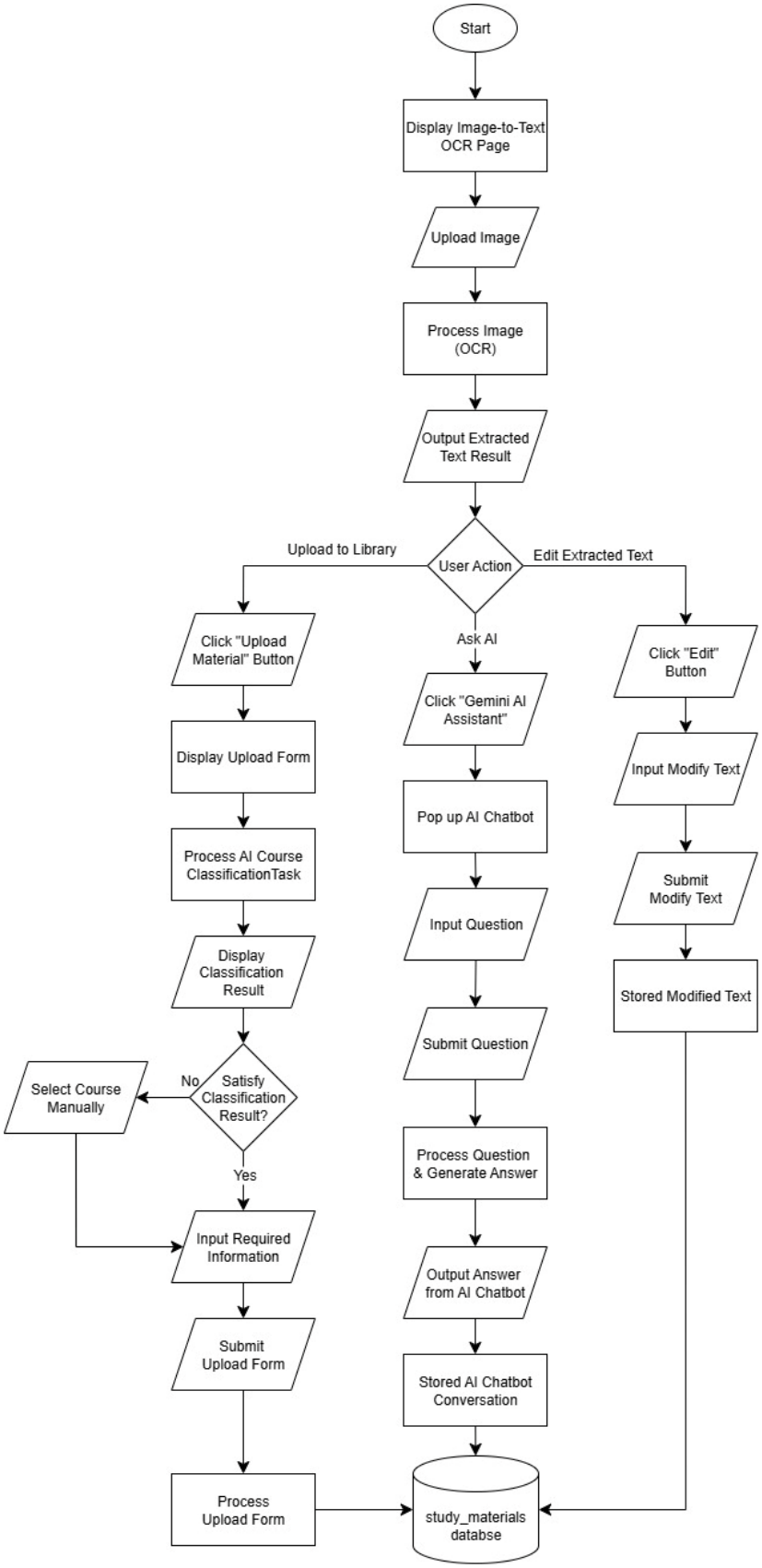


Figure 4.1.4 System Flowchart for Image-to-Text (OCR)

4.1.5 System Flowchart for Upload Video Summarizer

Figure 4.1.5 shows the Upload Video Summarizer system flowchart. First, the system will display the Upload Video Summarizer Page. The users need to upload a video and the system will then process the user-uploaded video and extract the video transcription. If the video transcription is successfully extracted, then the system will generate a video summary and output the video summary and transcription to users, else the system will display an error message “Unable to Generate Summary”. After displaying the result to users, users can perform several tasks. If users want to upload the video together with its video summary and transcript, then the users need to click the “Save to Study Resources” button. After that, the system will display an upload form with video, summary, and transcription preview. Since the material to upload is the video, the AI Course Classification will start automatically. After classification is done, the result will be displayed to users. If the users are satisfied with the classification result, then the user can submit the upload form after filling in all the required material information, else the users need to manually select the course and submit the upload form. After submitting the upload form, the system will process the upload form and store the material in the study_materials database..

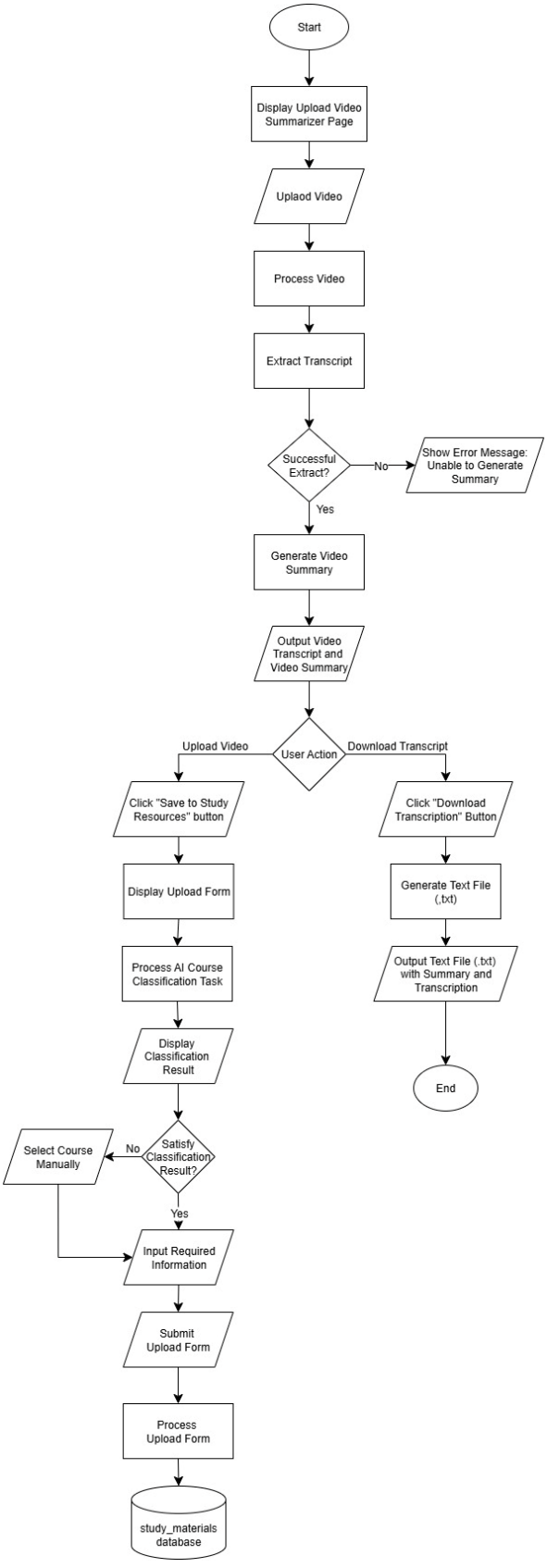


Figure 4.1.5 System Flowchart for Video Summarizer

4.1.6 System Flowchart for YouTube Video Summarizer

Figure 4.1.6 shows the system flowchart for YouTube Video Summarizer. First, the system will display the YouTube Video Summarizer Page. Users need to input a YouTube URL link. After that, the system will then process the YouTube URL link. The system will check whether the input YouTube URL link is valid or not. If yes, then the system will continue to extract the video ID and fetch the video transcription. After that, the system will check whether transcription is available or not. If yes, the system will generate the video summary and display the result of the summary and video transcription. If users want to upload the YouTube video to the Library page, then the users need to click the "Save to My Study Resources" button. The system will then display an upload form to the users. Since the material to upload is the video, the AI Course Classification will start automatically. After classification is done, the result will be displayed to the users. If the users are satisfied with the classification result, then the users can submit the upload form after filling in all the required material information. Else, the users need to manually select the course and submit the upload form. After submitting the upload form, the system will process the upload form and store the material in the study_materials database.

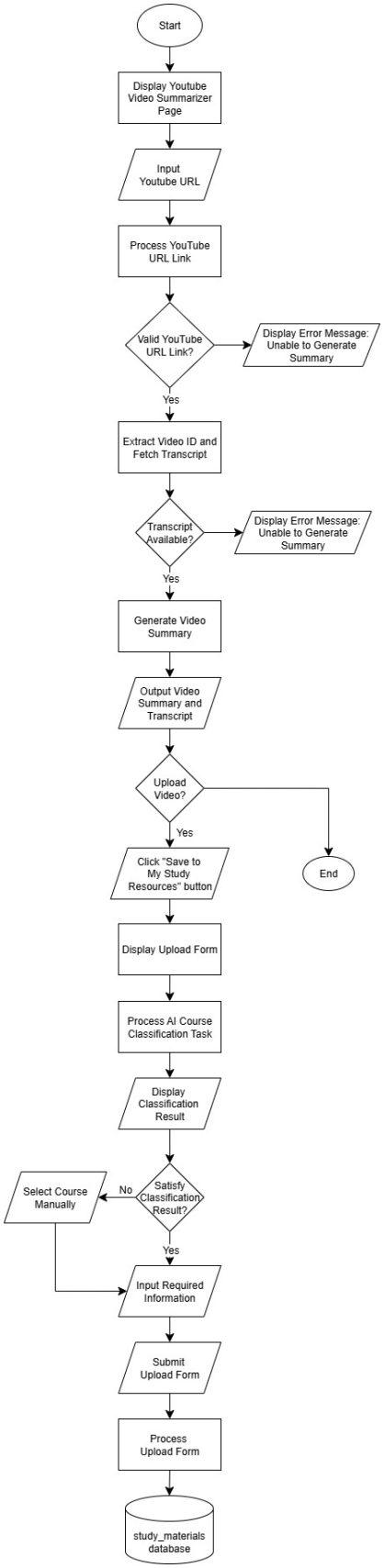


Figure 4.1.6 System Flowchart for YouTube Video Summarizer

4.1.7 System Flowchart for Calendar Planner

Figure 4.1.7 shows the system flowchart for Calendar Planner. The system will first display the Calendar Planner page. Users can perform different actions. If users want to create an AI-schedule task, then users need to click on the "+New Task" button. After that, the system will display a Create New Task form. Users are required to fill in all the required task details and click the "Create Schedule Task" button. The system will then process the upload form and schedule. The system will be based on the information provided in the upload form to allocate the task into a suitable timeslot, and the created task will be saved into the study_materials database.

Users can also create a direct task by clicking the specific timeslot column in the calendar view. The system will display a Create Direct Task form, and users are required to fill in the required task details and click the "+Create Task" button. The system will check whether the task created detected conflict or not. If yes, then the system will display an error message and display an alternative timeslot suggestion to users. Else, the task will be successfully created and stored in the study_materials database.

Users can import a timetable by clicking the "Import Timetable" button. The system will display the Import Timetable form. Users need to upload a timetable. The system will process the imported timetable and detect whether the timetable detects conflict with existing tasks or not. If yes, then an error message will be displayed. Else, the system will display the Imported Timetable Timeslot Preview and users need to click on the "Import Timetable" button. The system will then add the task and save it to the study_materials database.

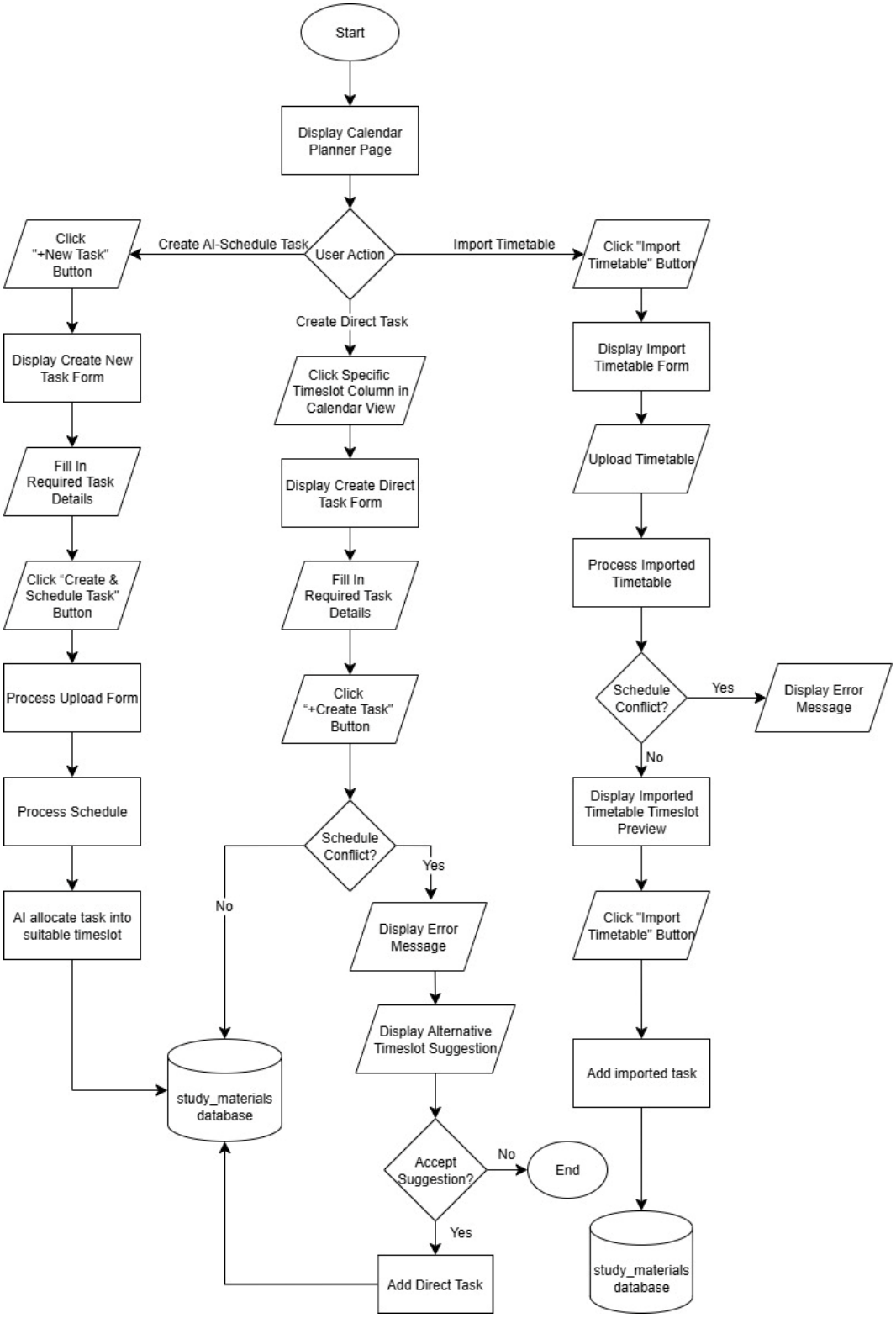


Figure 4.1.7 System Flowchart for Calendar Planner

4.1.8 System Flowchart for AI Study Tools

Figure 4.1.8 shows the system flowchart for AI Study Tools. The system will first display the View Material Page. To access the AI Study Tools page, the users need to click on the "Study with AI" button in the View Material Page. After that, the system will process the material, extract material content, process the extracted content, and generate a flashcard. Then the system will display the flashcard and save the generated flashcard to the study_materials database.

Users can click on the "Quiz" tab to be directed to the Quiz Tab interface. To generate a quiz, the users need to first select the quiz settings and click the "Generate New Quiz" button. The system will process the material, extract material content, process the extracted content, and generate a quiz. After displaying the generated quiz, the users can choose whether to answer the generated quiz or not. If yes, then users need to answer all of the questions and submit the quiz. Then the system will store the quiz result into the study_materials database.

Users can click on the "AI Assistant" tab and the system will direct users to the AI Assistant Tab interface. To ask questions, users are required to input the question and submit the question. Then the system will generate the answer and display the answer. The AI Chatbot conversation will also be saved to the study_materials database.

The users can click on the "Quiz History" tab to review previous quizzes. The system will direct users to the Quiz History Tab interface. The users can click on the "Review Answer" button. The system will display previous quiz answers and explanations.

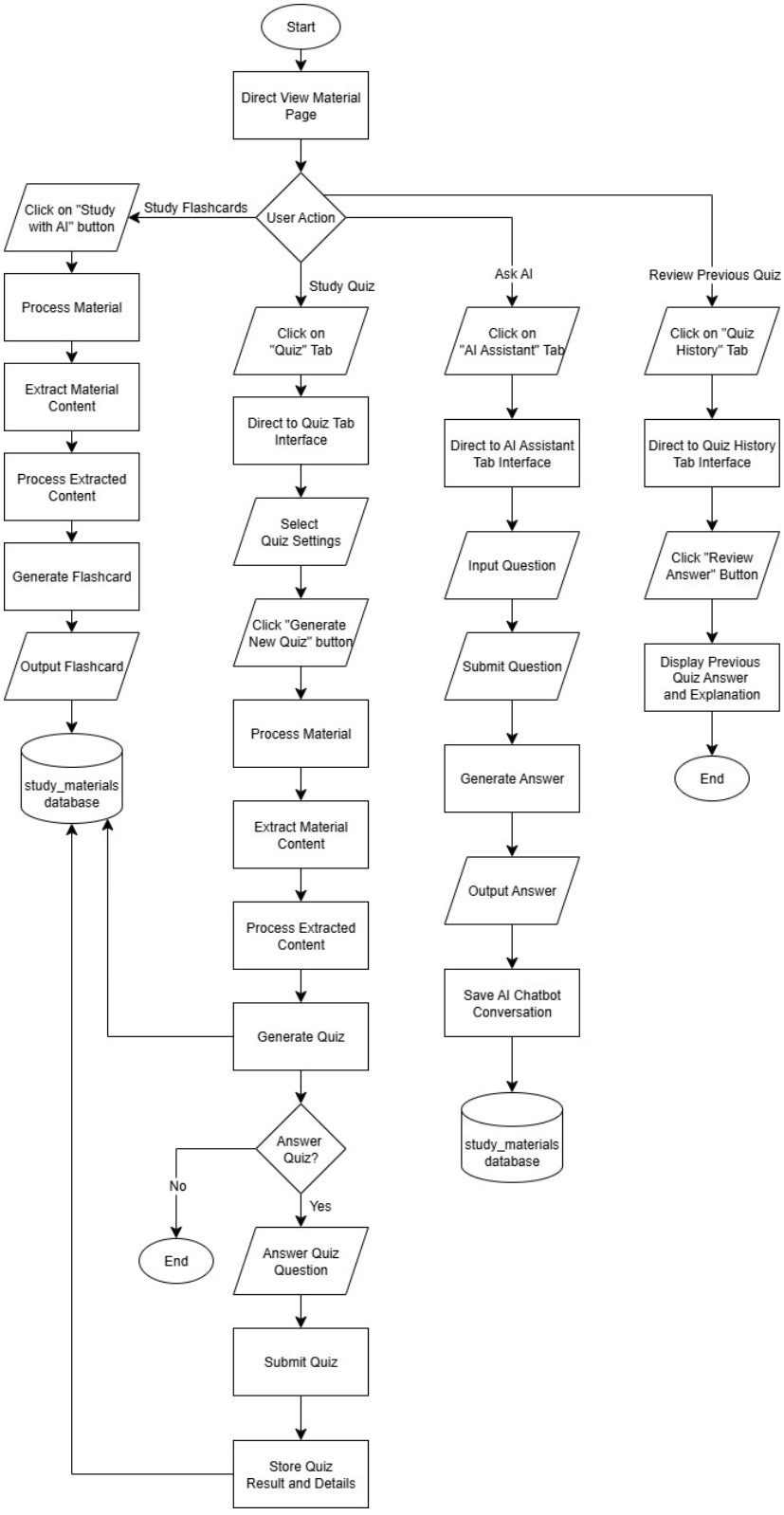


Figure 4.1.8 System Flowchart for AI Study Tools

4.2 Database SQL Command

The system uses phpMyAdmin with MariaDB as the database management system. The database is named as "study_materials", acts as the main storage for the platform's data. It's structured with 23 tables to efficiently store and manage all user and platform data such as user accounts, uploaded materials, and learning resources.

The database was created using the following SQL command:

```
CREATE DATABASE study_materials;
```

After creating the database, tables were created using the CREATE TABLE statements. The following are the SQL commands used to create the tables:

1. Login & Registration Module

- users: Keeps users' information like username, email, password, and profile picture.

```
CREATE TABLE `users` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`username` varchar(100) NOT NULL,
`email` varchar(255) NOT NULL,
`password` varchar(255) NOT NULL,
`profile_picture` varchar(500) DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
PRIMARY KEY (`id`),
UNIQUE KEY `username` (`username`),
UNIQUE KEY `email` (`email`)
)
```

- user_sessions: : Tracks active user sessions and authentication tokens.

```
CREATE TABLE `user_sessions` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`user_id` int(11) NOT NULL,
`session_token` varchar(255) NOT NULL,
`expires_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
PRIMARY KEY (`id`),
UNIQUE KEY `unique_token` (`session_token`),
KEY `user_id` (`user_id`),
CONSTRAINT `user_sessions_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE
)
```

2. Library Module

- materials: Stores uploaded resources such as documents, images, and videos, along with course details, extracted text, video transcripts, and summaries.

```
CREATE TABLE `materials` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `course_id` int(11) NOT NULL,
  `course_code` varchar(20) DEFAULT NULL,
  `subject_name` varchar(255) DEFAULT NULL,
  `title` varchar(255) NOT NULL,
  `description` text DEFAULT NULL,
  `file_name` varchar(255) DEFAULT NULL,
  `file_path` varchar(500) DEFAULT NULL,
  `file_data` longblob DEFAULT NULL,
  `file_type` varchar(50) NOT NULL,
  `file_category` enum('image','document','video','other') NOT NULL DEFAULT
'other',
  `file_size` int(11) NOT NULL,
  `upload_date` datetime NOT NULL,
  `is_private` tinyint(1) DEFAULT 0,
  `ocr_text` text DEFAULT NULL,
  `summary` text DEFAULT NULL,
  `transcript` text DEFAULT NULL,
  `youtube_id` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  KEY `course_id` (`course_id`),
  KEY `idx_file_category` (`file_category`),
  KEY `idx_user_course` (`user_id`,`course_id`),
  KEY `idx_upload_date` (`upload_date`),
  KEY `idx_course_code` (`course_code`),
  KEY `idx_subject_name` (`subject_name`),
```

```
  CONSTRAINT `materials_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES
`users` (`id`),
  CONSTRAINT `materials_ibfk_2` FOREIGN KEY (`course_id`)
REFERENCES `courses` (`id`)
)
```

- documents: Store processed text content from uploaded files, serving as the foundation for AI-powered study tools.

```
CREATE TABLE `documents` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `filename` varchar(255) NOT NULL,
  `original_filename` varchar(255) NOT NULL,
  `file_type` varchar(100) NOT NULL,
  `file_size` int(11) NOT NULL,
  `content` longtext NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `idx_user_created` (`user_id`,`created_at`),
  CONSTRAINT `documents_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES
`users` (`id`) ON DELETE CASCADE
)
```

- courses: The courses table provides a classification system for organizing study materials by courses.

```
CREATE TABLE `courses` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `course_name` varchar(255) NOT NULL,
  `category` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `course_name` (`course_name`)
)
```

3. Video Summarizer Module

- video_uploads: Stores uploaded videos informations.

```
CREATE TABLE `video_uploads` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `original_filename` varchar(255) NOT NULL,
  `server_filename` varchar(255) NOT NULL,
  `file_path` varchar(500) NOT NULL,
  `file_size` bigint(20) NOT NULL,
  `file_type` varchar(100) NOT NULL,
  `upload_timestamp` timestamp NOT NULL DEFAULT current_timestamp(),
  `transcription_status` enum('pending','processing','completed','failed') DEFAULT
'pending',
  `transcription_result` text DEFAULT NULL,
  `transcription_word_count` int(11) DEFAULT NULL,
  `transcription_duration` decimal(10,2) DEFAULT NULL,
  `summary` text DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `video_uploads_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`)
)
```

- video_summaries: Stores AI-generated summaries of uploaded videos.

```
CREATE TABLE `video_summaries` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `video_id` varchar(255) NOT NULL,
  `youtube_link` varchar(255) NOT NULL,
  `summary` text DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `video_summaries_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`)
)
```

4. Comments & Interaction Module

- comments: Stores user comments with sentiment scores and labels.

```
CREATE TABLE `comments` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `file_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `comment_text` text NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NULL DEFAULT NULL,
  `sentiment_score` int(11) DEFAULT 3 COMMENT 'Sentiment score from 1 (very negative) to 5 (very positive)',
  `sentiment_label` varchar(20) DEFAULT 'neutral' COMMENT 'Sentiment label: very_negative, negative, neutral, positive, very_positive',
  PRIMARY KEY (`id`),
  KEY `idx_file_id` (`file_id`),
  KEY `idx_user_id` (`user_id`),
  KEY `idx_created_at` (`created_at`),
  CONSTRAINT `comments_ibfk_1` FOREIGN KEY (`file_id`) REFERENCES `materials` (`id`) ON DELETE CASCADE,
  CONSTRAINT `comments_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE
)
```

- comment_likes: Store likes comments by users.

```
CREATE TABLE `comment_likes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `comment_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `created_at` datetime NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  UNIQUE KEY `unique_user_comment_like` (`comment_id`,`user_id`),
  KEY `idx_comment_likes_comment_id` (`comment_id`),
  KEY `idx_comment_likes_user_id` (`user_id`),
  CONSTRAINT `comment_likes_ibfk_1` FOREIGN KEY (`comment_id`)
REFERENCES `comments` (`id`) ON DELETE CASCADE,
  CONSTRAINT `comment_likes_ibfk_2` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE

    )
```

- content_likes: Store likes on library materials by users.

```
CREATE TABLE `content_likes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `material_id` int(11) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  UNIQUE KEY `unique_like` (`user_id`,`material_id`),
  KEY `material_id` (`material_id`),
  CONSTRAINT `content_likes_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE,
  CONSTRAINT `content_likes_ibfk_2` FOREIGN KEY (`material_id`)
REFERENCES `materials` (`id`) ON DELETE CASCADE

)
```

5. Smart Calendar Planner Module

- tasks: Stores created tasks information.

```
CREATE TABLE `tasks` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `task_name` varchar(255) NOT NULL,
  `duration` decimal(5,2) NOT NULL COMMENT 'Total duration in hours',
  `min_duration` decimal(4,2) NOT NULL COMMENT 'Minimum time per session in hours',
  `max_duration` decimal(4,2) NOT NULL COMMENT 'Maximum time per session in hours',
  `deadline` date NOT NULL,
  `preferred_start_date` date NOT NULL,
  `priority_level` enum('Low','Medium','High','Urgent') NOT NULL,
  `task_type` varchar(100) NOT NULL,
  `description` text DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
  `is_direct_scheduled` tinyint(1) DEFAULT 0,
  `is_imported` tinyint(1) DEFAULT 0,
  PRIMARY KEY (`id`),
  KEY `idx_tasks_user_id` (`user_id`),
  KEY `idx_tasks_user_deadline` (`user_id`,`deadline`),
  KEY `idx_tasks_priority` (`user_id`,`priority_level`),
  KEY `idx_tasks_user_type` (`user_id`,`is_direct_scheduled`,`is_imported`),
  CONSTRAINT `tasks_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE
)
```

- task_categories: Define the categories for tasks with color codes for display.

```
CREATE TABLE `task_categories` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `category_name` varchar(100) NOT NULL,
  `color_code` varchar(7) DEFAULT '#3498db',
  PRIMARY KEY (`id`),
  UNIQUE KEY `category_name` (`category_name`)
)
```

- scheduled_sessions:

```
CREATE TABLE `scheduled_sessions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `task_id` int(11) DEFAULT NULL,
  `session_date` date NOT NULL,
  `start_time` time NOT NULL,
  `end_time` time NOT NULL,
  `duration` decimal(4,2) NOT NULL,
  `is_completed` tinyint(1) DEFAULT 0,
  `session_type` enum('work','break','task','habit','direct','imported') DEFAULT
'work',
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON
UPDATE current_timestamp(),
  `habit_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_sessions_user_date` (`task_id`,`session_date`),
  KEY `idx_sessions_date_time` (`session_date`,`start_time`),
  KEY `idx_sessions_completion` (`is_completed`,`session_date`),
  KEY `idx_habit_id` (`habit_id`),
  KEY `idx_session_date_user` (`session_date`),
  KEY `idx_session_type_enhanced` (`session_type`),
  CONSTRAINT `scheduled_sessions_ibfk_1` FOREIGN KEY (`task_id`)
REFERENCES `tasks` (`id`) ON DELETE CASCADE,
  CONSTRAINT `scheduled_sessions_ibfk_2` FOREIGN KEY (`habit_id`)
REFERENCES `habits` (`id`) ON DELETE CASCADE
)
```

- habits: Stores user habits.

```
CREATE TABLE `habits` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `habit_name` varchar(255) NOT NULL,
  `duration` decimal(4,2) NOT NULL,
  `preferred_time` time DEFAULT NULL,
  `repeat_days` varchar(20) NOT NULL,
  `category` varchar(100) DEFAULT 'Personal',
  `priority` enum('Low','Medium','High') DEFAULT 'Medium',
  `description` text DEFAULT NULL,
  `is_active` tinyint(1) DEFAULT 1,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON
UPDATE current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `idx_user_active` (`user_id`,`is_active`),
  CONSTRAINT `habits_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES
`users` (`id`) ON DELETE CASCADE
)
```

- schedule_changes: Records changes made to scheduled sessions with old and new times.

```
CREATE TABLE `schedule_changes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `session_id` int(11) NOT NULL,
  `old_date` date NOT NULL,
  `new_date` date NOT NULL,
  `old_start_time` time DEFAULT NULL,
  `new_start_time` time DEFAULT NULL,
  `change_reason` varchar(100) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `session_id` (`session_id`),
  CONSTRAINT `schedule_changes_ibfk_1` FOREIGN KEY (`session_id`)
REFERENCES `scheduled_sessions` (`id`) ON DELETE CASCADE
)
```

6. Questions Generator Module

- flashcards: Stores flashcards generated from user uploaded documents with front question content and back answer.

```
CREATE TABLE `flashcards` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `document_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `front` text NOT NULL,
  `back` text NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  KEY `idx_document_user` (`document_id`,`user_id`),
  CONSTRAINT `flashcards_ibfk_1` FOREIGN KEY (`document_id`)
REFERENCES `documents` (`id`) ON DELETE CASCADE,
  CONSTRAINT `flashcards_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES
`users` (`id`) ON DELETE CASCADE

)
```

- quizzes: Manages quizzes including title, difficulty, and associated documents.

```
CREATE TABLE `quizzes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `document_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `title` varchar(255) NOT NULL,
  `difficulty_level` enum('easy','medium','hard') DEFAULT 'medium',
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  KEY `idx_document_user` (`document_id`,`user_id`),
  CONSTRAINT `quizzes_ibfk_1` FOREIGN KEY (`document_id`)
REFERENCES `documents` (`id`) ON DELETE CASCADE,
  CONSTRAINT `quizzes_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES
`users` (`id`) ON DELETE CASCADE
)
```

- quiz_questions: Stores individual quiz questions.

```
CREATE TABLE `quiz_questions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `quiz_id` int(11) NOT NULL,
  `question_type` enum('mcq','true_false','fill_blank','short_answer') NOT NULL,
  `question` text NOT NULL,
  `options` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_bin
DEFAULT NULL CHECK (json_valid(`options`)),
  `correct_answer` text NOT NULL,
  `explanation` text DEFAULT NULL,
  `question_order` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_quiz_order` (`quiz_id`,`question_order`),
  CONSTRAINT `quiz_questions_ibfk_1` FOREIGN KEY (`quiz_id`)
REFERENCES `quizzes` (`id`) ON DELETE CASCADE
)
```

- quiz_attempts: Records users' quiz attempts, scores, and answers.

```
CREATE TABLE `quiz_attempts` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `quiz_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `score` decimal(5,2) NOT NULL,
  `total_questions` int(11) NOT NULL,
  `correct_answers` int(11) NOT NULL,
  `answers` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_bin NOT
NULL CHECK (json_valid(`answers`)),
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `quiz_id` (`quiz_id`),
  KEY `idx_user_created` (`user_id`,`created_at`),
  CONSTRAINT `quiz_attempts_ibfk_1` FOREIGN KEY (`quiz_id`)
REFERENCES `quizzes` (`id`) ON DELETE CASCADE,
  CONSTRAINT `quiz_attempts_ibfk_2` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE
)
```

- quiz_attempt_responses: Stores detailed responses for each question in a quiz attempt.

```
CREATE TABLE `quiz_attempt_responses` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `attempt_id` int(11) NOT NULL,
  `question_id` int(11) NOT NULL,
  `user_answer` text DEFAULT NULL,
  `is_correct` tinyint(1) DEFAULT 0,
  `question_order` int(11) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `question_id` (`question_id`),
  KEY `idx_attempt_responses` (`attempt_id`,`question_order`),
  CONSTRAINT `quiz_attempt_responses_ibfk_1` FOREIGN KEY (`attempt_id`)
REFERENCES `quiz_attempts` (`id`) ON DELETE CASCADE,
  CONSTRAINT `quiz_attempt_responses_ibfk_2` FOREIGN KEY
(`question_id`) REFERENCES `quiz_questions` (`id`) ON DELETE CASCADE
)
```

7. AI Chat Module

- chat_conversations: Stores chat histories between users and the Gemini AI assistant.

```
CREATE TABLE `chat_conversations` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `document_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `message` text NOT NULL,
  `response` text NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  KEY `idx_document_user_created` (`document_id`,`user_id`,`created_at`),
  CONSTRAINT `chat_conversations_ibfk_1` FOREIGN KEY (`document_id`)
REFERENCES `documents` (`id`) ON DELETE CASCADE,
  CONSTRAINT `chat_conversations_ibfk_2` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE
)
```

- gemini_conversations: Store conversation between the user and the Gemini AI, along with the extracted text and the original image.

```
CREATE TABLE `gemini_conversations` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `ocr_text` text NOT NULL,
  `image_path` varchar(255) NOT NULL,
  `created_at` datetime NOT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_user_id` (`user_id`),
  CONSTRAINT `gemini_conversations_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE
)
```

- gemini_messages: Stores AI chat history during AI-assisted learning sessions.

```
CREATE TABLE `gemini_messages` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `conversation_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `sender` enum('user','gemini') NOT NULL,
  `message` text NOT NULL,
  `sent_at` datetime NOT NULL,
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  KEY `idx_conversation_id` (`conversation_id`),
  CONSTRAINT `gemini_messages_ibfk_1` FOREIGN KEY (`conversation_id`)
REFERENCES `gemini_conversations` (`id`) ON DELETE CASCADE,
  CONSTRAINT `gemini_messages_ibfk_2` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE
)
```

4.3 Timeline

Activity/Week	1	2	3	4	5	6	7	8	9	10	11	12
Design the website layout and features.												
Set up XAMPP and create the database.												
Build login and register system.												
Start OCR coding for image text extraction.												
Integrate OCR with Gemini AI												
Start coding for YouTube video transcription and summarization.												
Build transcription for user-uploaded videos.												
Coding for summarization for user-uploaded videos.												
Write report												

Table 4.3.1 Timeline for FYP 1

Activity/Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Implement commenting system with sentiment analysis													
Integrate hate speech detection for comments													
Implement AI course classification system													
Redesign Platform UI													
Develop smart calendar planner with AI scheduling													
Implement timetable import and create direct task functionality.													
Implement Smart Calendar Planner with Conflict Detection													
Build AI study tools generate flashcards													
Build AI study tools generate quizzes and AI Chabot													
System Testing and Fixing Bugs													
Write report													

Table 4.3.2 Timeline for FYP 2

Chapter 5

System Implementation

5. 1 Hardware Setup

In this project, a computer is the only required hardware component. The computer is used for the entire development process, including coding the OCR functionality with pytesseract, implementing video summarization using Google Gemini API, and developing the transcript extraction features with Whisper and YouTube Transcript API. Since the Student Resource Exchange is a web-based system, all testing and deployment are performed directly on the computer without requiring additional hardware devices. The system processes uploaded images and videos on the server side, eliminating the need for external capture devices during development or deployment.

Description	Specifications
Model	REALME Book 14” Laptop
Processor	Intel Core i5-1135G7
Operating System	Windows 10
Graphic	Intel Iris Xe Graphics
Memory	8GB DDR4 4266Mhz RAM
Storage	512GB PCIe SSD

Table 5.1 Specifications of laptop

5.2 Software Setup

Before starting the development of the Student Resource Exchange System, the following softwares are required to be installed and used on the laptop:

1. Python

   Python is a popular high-level programming language that's great for web development, machine learning, and data science since it's easy to learn and understand [19]. It provides various frameworks and libraries that support everything from web application development to data analysis and machine learning.

   In this project, Python will be used to implement the key features. For example, the Optical Character Recognition (OCR) Tools to extract text from images and video transcription will be generated using Whisper. Tools like pytesseract will be used for text extraction, while Whisper will handle video transcription. These tools will allow the system to extract and transcribe content from multimedia materials, providing valuable insights to users.

2. Visual Studio Code

   Visual Studio Code (VS Code) is a lightweight, open-source, and highly customizable source code editor developed by Microsoft [20]. It is known for its speed, rich features, and flexibility which make it a popular choice for developers. VS Code supports multiple programming languages, integrates Git version control, and provides excellent debugging tools [20].

   For this project, Visual Studio Code is the primary integrated development environment (IDE) for Python development. Visual Studio Code will also be used to handle front-end code with HTML, CSS, and JavaScript. The editor supports useful extensions that enhance the development process.

3. XAMPP

   XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It provides an easy-to-install distribution of Apache, MySQL, PHP, and Perl. XAMPP helps to simplify the process of setting up a local development environment for web applications [21]. In this project, XAMPP will be used to start Apache and MySQL. By starting them, it provides the necessary server and database environment for managing and hosting the web application locally. Apache will handle the web server functionalities and MySQL will manage the database, ensuring smooth data storage and retrieval for the application.

5.3 Installation of Required Libraries

Other than installing the necessary software, some external Python libraries also need to be installed to develop the main features of this project.

1. python-dotenv: This library is used to load environment variables from a .env file into the application [22]. In this project, this library is used to load the Gemini API key that needed to access Google's Generative AI services more secule. This python-dotenv library is to keep sensitive information like API keys separate from the main code. This will help to improve the project security and maintainability.

```
(venv) C:\Users\huils2114\Desktop\FYP>pip install python-dotenv
Collecting python-dotenv
  Downloading python_dotenv-1.1.0-py3-none-any.whl.metadata (24 kB)
Downloading python_dotenv-1.1.0-py3-none-any.whl (20 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.1.0
```

Figure 5.3.1 Installation of python-dontenv Library

2. google-generativeai: This library provides access to Google's Generative AI services, including the Gemini AI model. In this project, it is used for multiple purposes including asking questions about text extracted from images, generating summaries from videos and generating flashcards and quizzes from user-uploaded materials. It also allows users to ask questions about both extracted text and selected study resources. Additionally, the Gemini 1.5 Flash model is also used to intelligently classify uploaded content into most relevant course categories, ensuring better organization of study materials.

```
(venv) C:\Users\huils2114\Desktop\FYP>pip install google-generativeai
Collecting google-generativeai
  Downloading google_generativeai-0.8.5-py3-none-any.whl.metadata (3.9 kB)
Collecting google-ai-generativelanguage==0.6.15 (from google-generativeai)
  Using cached google_ai_generativelanguage-0.6.15-py3-none-any.whl.metadata (5.7 kB)
Collecting google-api-core (from google-generativeai)
  Using cached google_api_core-2.24.2-py3-none-any.whl.metadata (3.0 kB)
Collecting google-api-python-client (from google-generativeai)
  Downloading google_api_python_client-2.168.0-py3-none-any.whl.metadata (6.7 kB)
Collecting google-auth>=2.15.0 (from google-generativeai)
  Downloading google_auth-2.39.0-py2.py3-none-any.whl.metadata (6.2 kB)
Collecting protobuf (from google-generativeai)
  Downloading protobuf-6.30.2-cp310-abi3-win_amd64.whl.metadata (593 bytes)
```

Figure 5.3.2 Installation of google-generativeai Library

3. pytesseract: This is a Python wrapper for Google's Tesseract-OCR Engine [23]. It is used in this project to perform OCR, which extracts text from images. This allows the system to turn text in uploaded images into editable and searchable digital text.

```
(venv) C:\Users\huils2114\Desktop\FYP>pip install pytesseract
Collecting pytesseract
  Using cached pytesseract-0.3.13-py3-none-any.whl.metadata (11 kB)
Collecting packaging>=21.3 (from pytesseract)
  Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Collecting Pillow>=8.0.0 (from pytesseract)
  Downloading pillow-11.2.1-cp310-cp310-win_amd64.whl.metadata (9.1 kB)
Using cached pytesseract-0.3.13-py3-none-any.whl (14 kB)
Downloading packaging-25.0-py3-none-any.whl (66 kB)
Downloading pillow-11.2.1-cp310-cp310-win_amd64.whl (2.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.7/2.7 MB 12.9 MB/s eta 0:00:00
Installing collected packages: Pillow, packaging, pytesseract
Successfully installed Pillow-11.2.1 packaging-25.0 pytesseract-0.3.13
```

Figure 5.3.3 Installation of pytesseract Library

4. Pillow (PIL): Pillow is a Python Imaging Library which used for opening, editing, and saving images into different file formats [24]. In this project, Pillow is used to process images. For example, Pillow is used to adjust the image brightness, contrast, and apply filters to improve the quality of the images. The image with good quality help make the text extraction more accurate.

```
(venv) C:\Users\huils2114\Desktop\FYP>pip install pytesseract
Collecting pytesseract
  Using cached pytesseract-0.3.13-py3-none-any.whl.metadata (11 kB)
Collecting packaging>=21.3 (from pytesseract)
  Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Collecting Pillow>=8.0.0 (from pytesseract)
  Downloading pillow-11.2.1-cp310-cp310-win_amd64.whl.metadata (9.1 kB)
Using cached pytesseract-0.3.13-py3-none-any.whl (14 kB)
Downloading packaging-25.0-py3-none-any.whl (66 kB)
Downloading pillow-11.2.1-cp310-cp310-win_amd64.whl (2.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.7/2.7 MB 12.9 MB/s eta 0:00:00
Installing collected packages: Pillow, packaging, pytesseract
Successfully installed Pillow-11.2.1 packaging-25.0 pytesseract-0.3.13

(venv) C:\Users\huils2114\Desktop\FYP>pip install Pillow
Requirement already satisfied: Pillow in c:\users\huils2114\desktop\fyp\venv\lib\site-packages (11.2.1)
```

Figure 5.3.4 Installation of Pillow (PIL) Library

5. youtube-transcript-api: This library allows users to retrieve transcripts from YouTube videos [25]. In this project, it is used to fetch the text transcript of a YouTube video, which can then be summarized using the Gemini AI.

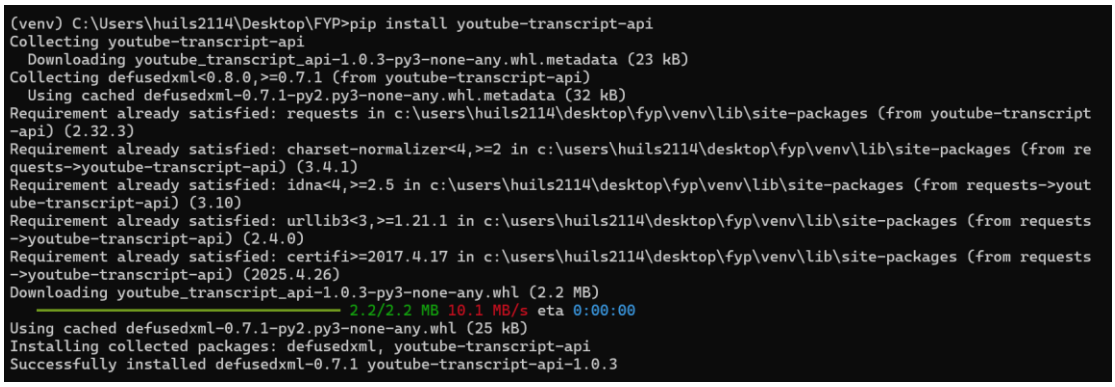


Figure 5.3.5 Installation of youtube-transcript-api Library

6. whisper: Whisper is an automatic speech recognition (ASR) system developed by OpenAI [26]. It is used in this project to transcribe audio from videos into text. The transcribed text can then be processed and summarized with the help of Gemini AI, allowing the system to handle spoken content from videos.

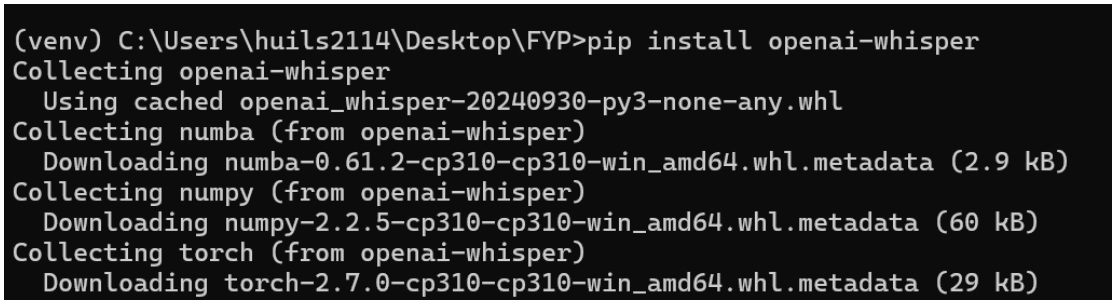


Figure 5.3.6 Installation of whisper Library

7. flask: Flask is a lightweight web framework for Python that supports the development of web applications and APIs [27]. In this project, it serves as the primary backend framework, providing REST API endpoints to handle file uploads, process classification requests, communicate with the PHP frontend through HTTP requests.

```
(venv) C:\Users\huils2114\Desktop\FYP2>pip install flask
Collecting flask
  Downloading flask-3.1.2-py3-none-any.whl (103 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 103.3/103.3 kB 3.0 MB/s eta 0:00:00
Collecting werkzeug>=3.1.0
  Using cached werkzeug-3.1.3-py3-none-any.whl (224 kB)
Collecting blinker>=1.9.0
  Using cached blinker-1.9.0-py3-none-any.whl (8.5 kB)
Collecting itsdangerous>=2.2.0
  Using cached itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting markupsafe>=2.1.1
  Using cached MarkupSafe-3.0.2-cp310-cp310-win_amd64.whl (15 kB)
Collecting click>=8.1.3
  Using cached click-8.2.1-py3-none-any.whl (102 kB)
Collecting jinja2>=3.1.2
  Using cached jinja2-3.1.6-py3-none-any.whl (134 kB)
Collecting colorama
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: markupsafe, itsdangerous, colorama, blinker, werkzeug, jinja2, click, flask
Successfully installed blinker-1.9.0 click-8.2.1 colorama-0.4.6 flask-3.1.2 itsdangerous-2.2.0 jinja2-3.1.6 markupsaf
e-3.0.2 werkzeug-3.1.3
```

Figure 5.3.7 Installation of flask Library

8. flask-cors: The flask-cors library is used to enable Cross-Origin Resource Sharing (CORS) [28]. Foe example, this library allows the frontend and backend to interact even if they are on different domains. In this project, flask-cors is used to allow the PHP frontend to make API requests to the Flask backend without being blocked by the browser.

```
(venv) C:\Users\huils2114\Desktop\FYP2>pip install flask-cors
Collecting flask-cors
  Using cached flask_cors-6.0.1-py3-none-any.whl.metadata (5.3 kB)
Requirement already satisfied: flask>=0.9 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from flask-cors)
 (3.1.2)
Requirement already satisfied: Werkzeug>=0.7 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from flask-co
rs) (3.1.3)
Requirement already satisfied: blinker>=1.9.0 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from flask>=
0.9->flask-cors) (1.9.0)
Requirement already satisfied: click>=8.1.3 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from flask>=0.
9->flask-cors) (8.2.1)
Requirement already satisfied: itsdangerous>=2.2.0 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from fl
ask>=0.9->flask-cors) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from flask>=0
.9->flask-cors) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from flas
k>=0.9->flask-cors) (3.0.2)
Requirement already satisfied: colorama in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from click>=8.1.3-
>flask>=0.9->flask-cors) (0.4.6)
Using cached flask_cors-6.0.1-py3-none-any.whl (13 kB)
Installing collected packages: flask-cors
Successfully installed flask-cors-6.0.1
```

Figure 5.3.8 Installation of flask-cors Library

9. pydantic: Pydantic is a library used for checking and validating data [29]. It provides the BaseModel class, which makes sure that incoming data is in the correct format before the backend processes it. In this project, Pydantic is used to validate requests sent to the backend API. This will make the system more reliable and preventing errors from invalid data.

```
(venv) C:\Users\huils2114\Desktop\FYP2>pip install pydantic
Collecting pydantic
  Downloading pydantic-2.11.9-py3-none-any.whl.metadata (68 kB)
Collecting annotated-types>=0.6.0 (from pydantic)
  Using cached annotated_types-0.7.0-py3-none-any.whl.metadata (15 kB)
Collecting pydantic-core==2.33.2 (from pydantic)
  Using cached pydantic_core-2.33.2-cp310-cp310-win_amd64.whl.metadata (6.9 kB)
Collecting typing-extensions>=4.12.2 (from pydantic)
  Downloading typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Collecting typing-inspection>=0.4.0 (from pydantic)
  Using cached typing_inspection-0.4.1-py3-none-any.whl.metadata (2.6 kB)
Downloading pydantic-2.11.9-py3-none-any.whl (444 kB)
Using cached pydantic_core-2.33.2-cp310-cp310-win_amd64.whl (2.0 MB)
Using cached annotated_types-0.7.0-py3-none-any.whl (13 kB)
Downloading typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Using cached typing_inspection-0.4.1-py3-none-any.whl (14 kB)
Installing collected packages: typing-extensions, annotated-types, typing-inspection, pydantic-core, pydantic
Successfully installed annotated-types-0.7.0 pydantic-2.11.9 pydantic-core-2.33.2 typing-extensions-4.15.0 typing-ins
pection-0.4.1
```

Figure 5.3.9 Installation of pydantic Library

10. PyPDF2: PyPDF2 is a library used to read and extract text from PDF files [30]. In this project, PyPDF2 is used to extract readable text content from PDF files uploaded by students, enabling the AI system to analyze academic papers, lecture notes, and textbooks for course classification.

```
(venv) C:\Users\huils2114\Desktop\FYP2>pip install PyPDF2
Collecting PyPDF2
  Using cached pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Using cached pypdf2-3.0.1-py3-none-any.whl (232 kB)
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1
```

Figure 5.3.10 Installation of PyPDF2 Library

11. python-docx: This library is used to read and work with Microsoft Word (.docx) documents [31]. In this project, python-docx is used to extract text content from Word documents uploaded by students, such as assignments, research papers, and study notes, making them analyzable by the AI classification system.

```
(venv) C:\Users\huils2114\Desktop\FYP2>pip install PyPDF2
Collecting PyPDF2
  Using cached pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Using cached pypdf2-3.0.1-py3-none-any.whl (232 kB)
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1

(venv) C:\Users\huils2114\Desktop\FYP2>pip install python-docx
Collecting python-docx
  Downloading python_docx-1.2.0-py3-none-any.whl.metadata (2.0 kB)
Collecting lxml>=3.1.0 (from python-docx)
  Downloading lxml-6.0.1-cp310-cp310-win_amd64.whl.metadata (3.9 kB)
Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (fr
om python-docx) (4.15.0)
Downloading python_docx-1.2.0-py3-none-any.whl (252 kB)
Downloading lxml-6.0.1-cp310-cp310-win_amd64.whl (4.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.0/4.0 MB 6.2 MB/s  0:00:00
Installing collected packages: lxml, python-docx
Successfully installed lxml-6.0.1 python-docx-1.2.0
```

Figure 5.3.11 Installation of python-docx Library

12. python-pptx: This library is used to read and process Microsoft PowerPoint (.pptx) presentations [32]. In this project, it is used to extract text content from PowerPoint slides uploaded by students, including lecture presentations and project slides, for AI-powered course classification.

```
(venv) C:\Users\huils2114\Desktop\FYP2>pip install python-pptx
Collecting python-pptx
  Using cached python_pptx-1.0.2-py3-none-any.whl.metadata (2.5 kB)
Collecting Pillow>=3.3.2 (from python-pptx)
  Downloading pillow-11.3.0-cp310-cp310-win_amd64.whl.metadata (9.2 kB)
Collecting XlsxWriter>=0.5.7 (from python-pptx)
  Downloading xlsxwriter-3.2.9-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: lxml>=3.1.0 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (from python-ppt
x) (6.0.1)
Requirement already satisfied: typing-extensions>=4.9.0 in c:\users\huils2114\desktop\fyp2\venv\lib\site-packages (fr
om python-pptx) (4.15.0)
Using cached python_pptx-1.0.2-py3-none-any.whl (472 kB)
Downloading pillow-11.3.0-cp310-cp310-win_amd64.whl (7.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.0/7.0 MB 7.7 MB/s  0:00:00
Downloading xlsxwriter-3.2.9-py3-none-any.whl (175 kB)
Installing collected packages: XlsxWriter, Pillow, python-pptx
Successfully installed Pillow-11.3.0 XlsxWriter-3.2.9 python-pptx-1.0.2
```

Figure 5.3.12 Installation of python-pptx Library

13. pandas: Pandas is a powerful data manipulation and analysis library for Python that provides data structures like DataFrames [33]. It is used in this project to read the training dataset from CSV files (pd.read_csv("train.csv")) [37], manipulate the dataset by creating binary labels for toxic/non-toxic classification, and perform data preprocessing operations on the comment text. The dataset "train.csv" was obtained from the Kaggle website [37], which provides labeled comments for training and evaluation.

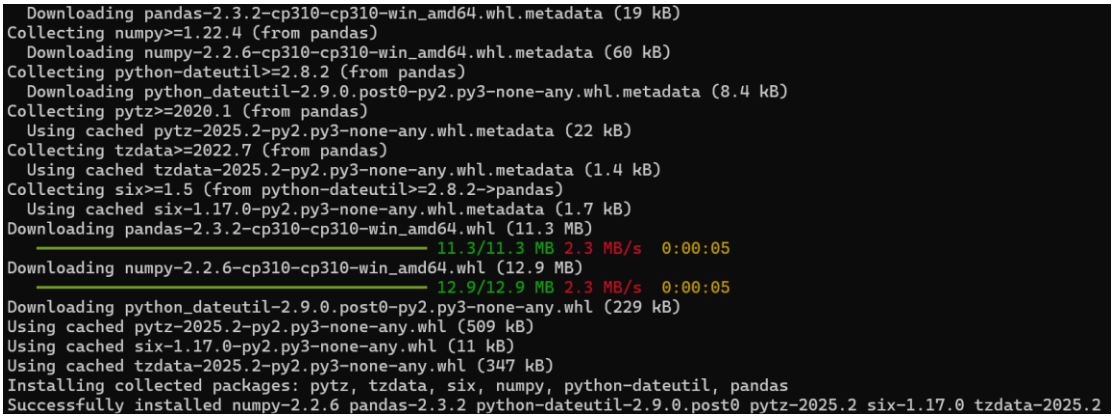


Figure 5.3.13 Installation of pandas Library

14. numpy: NumPy is a library used for numerical computing and handling arrays [34]. In this project, NumPy is used to convert pandas DataFrames into numpy arrays (np.array()) that are compatible with scikit-learn's machine learning algorithms for training the hate speech detection model.

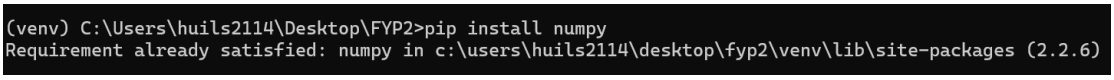


Figure 5.3.14 Installation of numpy Library

15. scikit-learn: Scikit-learn is a machine learning library for Python [35]. In this project, scikit-learn is used for two main components: TfidfVectorizer to convert text data into numerical feature vectors, and LogisticRegression to train the binary classification model that distinguishes between toxic and non-toxic comments.

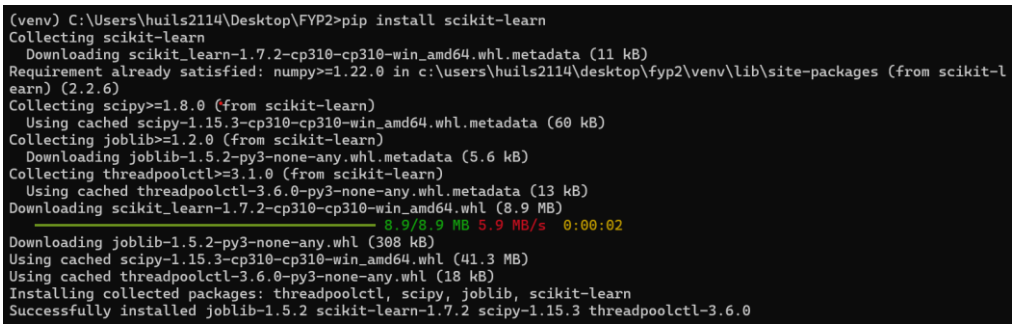


Figure 5.3.15 Installation of scikit-learn Library

16. nltk: The Natural Language Toolkit (NLTK) is a Python library used for text preprocessing and Natural Language Processing (NLP) [36]. In this project, NLTK is used to remove stopwords and apply stemming to words before training and predicting hate speech detection.

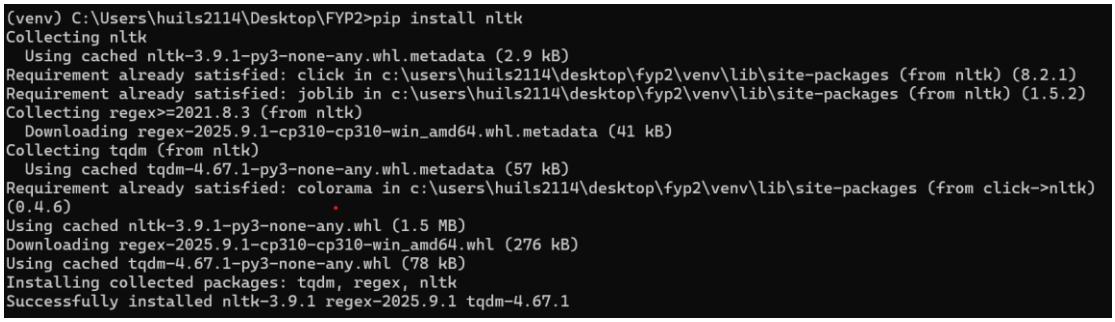


Figure 5.3.16 Installation of nltk Library

17. transformers: In this project, transformers is used to load the multilingual BERT-based sentiment analysis model. It enables the system to classify user comments into sentiment categories such as very negative, negative, neutral, positive, and very positive, and return the results in JSON format for the comment system.

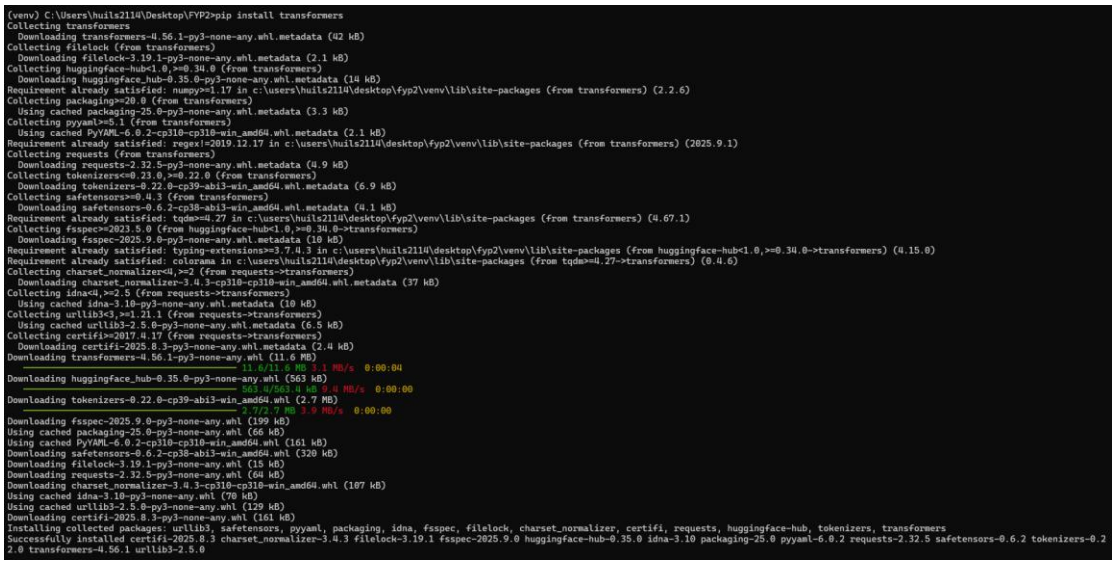


Figure 5.3.17 Installation of transformers Library

18. torch: In this project, PyTorch is required to run the BERT-based sentiment analysis model used in the comment system. It performs the underlying computations for tokenization, predictions, and probability scoring to ensure accurate sentiment classification.

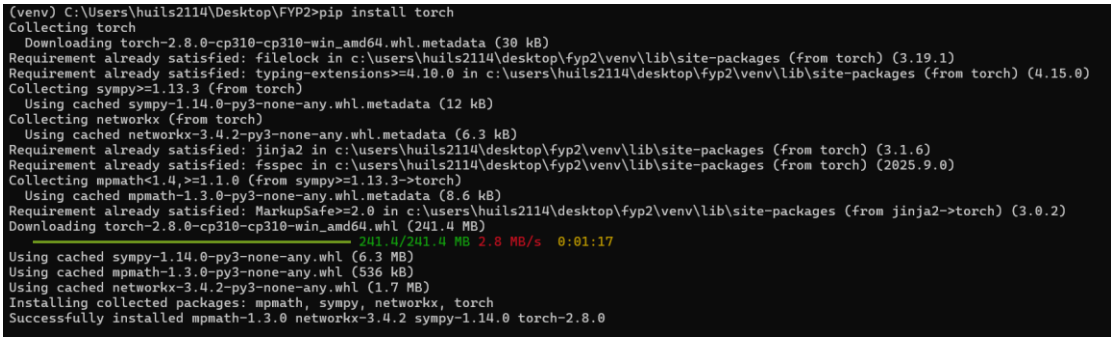


Figure 5.3.18 Installation of torch Library

5.4 Algorithms

This section outlines the key algorithms implemented in the system. The system combines custom-built algorithms for specific processing tasks with external AI services.

1. Optical Character Recognition (OCR)

The OCR module extracts text from images using the following algorithms:

- **Tesseract OCR with LSTM Neural Networks** for text recognition.
- **Image Preprocessing Algorithms**: Resizing, contrast enhancement, Gaussian filtering, and thresholding to optimize images quality for accurate text extraction.

2. Video Summarization

The Video Summarizer features generate summaries of videos using YouTube transcripts and the Google Gemini AI model. The algorithms applied include:

- **Automatic Speech Recognition (ASR)**: Using Whisper to convert spoken audio in videos into text.
- **Segmentation Algorithm**: The transcript is divided into segments to manage content efficiently.
- **Large Language Model with Prompt Engineering**: The **Google Gemini API** is used to generate clear and concise summaries from the transcript. A prompt is designed to give instructions to the model to focus on key points and main ideas and produce a summary of the video.

3. Comment System

The comment system uses machine learning and deep learning algorithms to detect hate speech and analyze sentiment.

- **TF-IDF + Logistic Regression for Hate Speech Detection**: The algorithm converts comment text into numerical feature vectors using TF-IDF vectorization after preprocessing, then classifies content as toxic or non-toxic using a trained Logistic Regression model. The dataset “train.csv” [37], obtained from Kaggle, is used for training and evaluation.
- **BERT-based Sentiment Analysis**: This algorithm analyzes the sentiment of user comments. Comments are tokenized and processed by a pre-trained BERT model, which predicts whether the sentiment is positive, neutral, or negative.

4. Course Classification System

The course classification system automatically organizes user-uploaded materials into relevant academic courses using Google Gemini 1.5 Flash with engineered prompts that analyze content context, extract key topics, and provide confidence-scored classifications with alternative course suggestions:

- **Large Language Model with Prompt Engineering**: Google Gemini 1.5 Flash model is used to classify content based on context. A prompt is given to guide the model and categorize the content into relevant courses.
- **LLM-Based Classification with Confidence Tiering**: The Google Gemini model generates classification confidence scores and alternative course suggestions as part of its structured output.

5. Smart Calendar Planner

The smart calendar planner manages scheduling, task progress evaluation, and time suggestions for study sessions. It uses several algorithms to ensure efficient, conflict-free scheduling.

- **Conflict Detection Algorithm**: This algorithm checks proposed sessions against existing tasks and habits using temporal overlap detection. Returns detailed conflict information including overlapping sessions, duration, and priority levels.
- **AI-Powered Scheduling Algorithm**: This algorithm uses Google Gemini API to generate optimal schedules by analyzing task requirements, existing commitments, and user preferences.
- **Priority-Based Greedy Scheduling Algorithm**: This algorithm implements greedy selection of time slots based on task priority levels.
- **Task Progress Calculation Algorithm**: The algorithm calculates the completion status of a task by counting completed sessions and summing completed hours, helping users track progress effectively.
- **Data Aggregation Algorithm**: This algorithm organizes imported timetables into structured groups based on course names.
- **Time Slot Suggestion Algorithm**: This algorithm checks proposed time slots against existing sessions. If an overlap is detected, it prevents creating the task and instead provides suggestions for available time slots , allowing users to schedule sessions without conflicts.

6. Questions Generator

The questions generator automatically produces flashcards and quizzes from user uploaded content using **Natural Language Processing (NLP)** and **Large Language Models (LLM)**.

- **Natural Language Processing (NLP):** The NLP extracts and analyzes text from user-uploaded content in multiple file formats, including PDF, DOCX and PPT, using libraries such as PyPDF2.
- **Large Language Model (LLM) – Google Gemini API:** This algorithm uses the Google Gemini API to generate flashcards and quizzes from preprocessed text. A prompt is given to guide the model to focus on key concepts, definitions, and main ideas in the content.
- **Fuzzy String-Matching Algorithm**: This algorithm evaluates the correctness of a user's answer.

5.5 System Operation

5.5.1 Login Page

Figure 5.1.1 shows the Login Page of the website “Student Resource Hub”. Users need to fill in the login credentials and by clicking the “Sign In” button, the user will be able to direct to the Home Page if the email and password are correct.

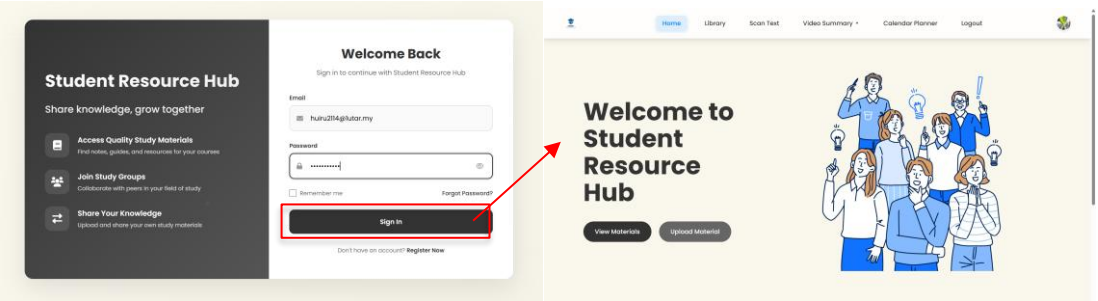


Figure 5.5.1 Login Page

5.5.2 Registration Page

Figure 5.1.2 shows the Registration Page interface of the website. User needs to fill in the register form to create a new user account. After filling in all the required fields, user needs to click the “Create Account” button. A success message will be displayed upon successful registration.

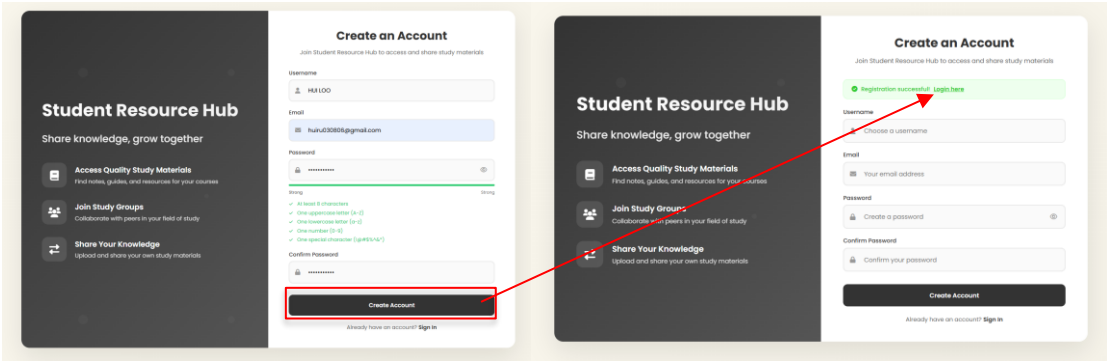


Figure 5.5.2 Registration Page

5.5.3 Library Page

Figure 5.5.3 shows the Library Page interface with all the materials uploaded by the users. Users can view the materials uploaded by clicking the “View” button and they will be directed to the View Material Page.

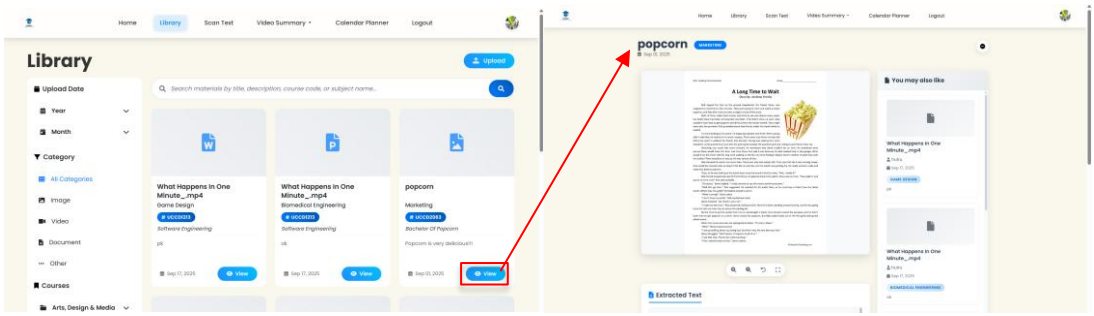


Figure 5.5.3 Library Page View Button

For Library Page, users are able to search for materials by entering the materials title, course code or course name. Figure 5.5.4 shows the search function of the Library Page.

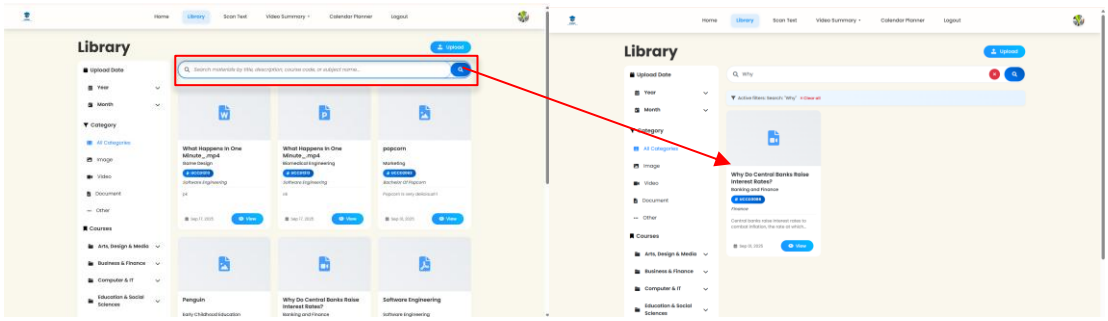


Figure 5.5.4 Library Page Search Function

In addition to search function, the Library Page allows users to find the materials by applying filters from the left sidebar. Figure 5.5.5 shows the filtering functions, where the “Image” category and “Marketing” course are selected to filter the materials.

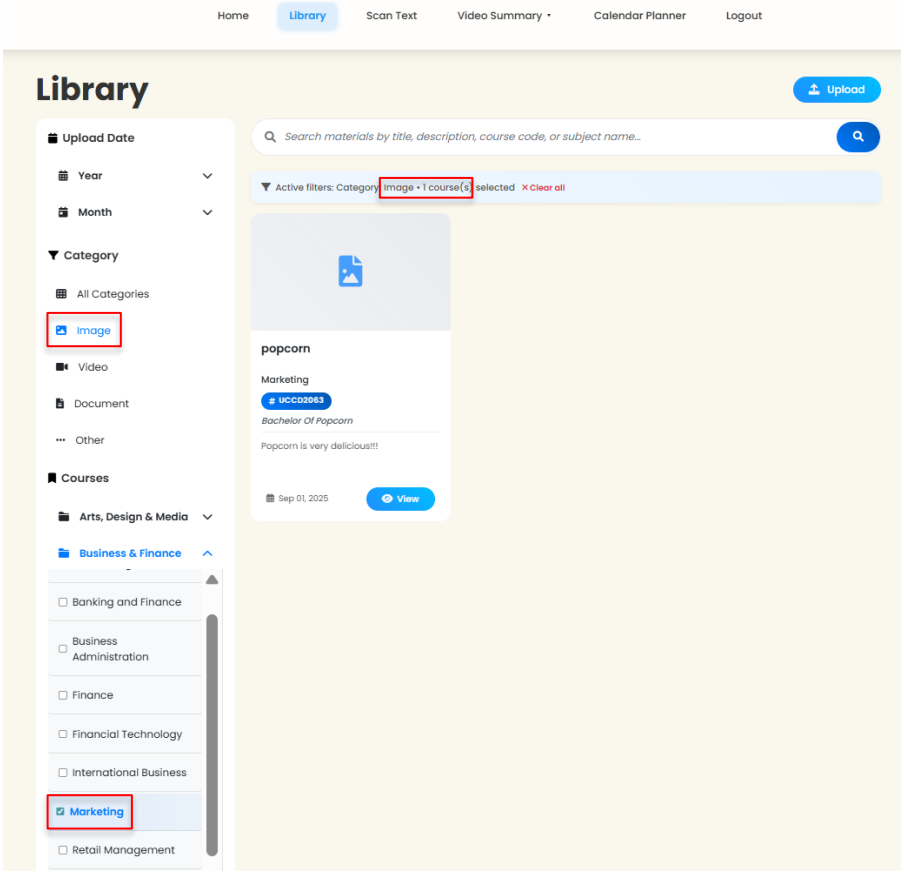


Figure 5.5.5 Library Page Filtering Function

5.5.4 Upload Materials Page

Figure 5.5.6 shows the "Upload" button in the Library Page. By clicking the "Upload" button, users will be directed to the Upload Materials Page shown in Figure 5.5.7, with an upload form.

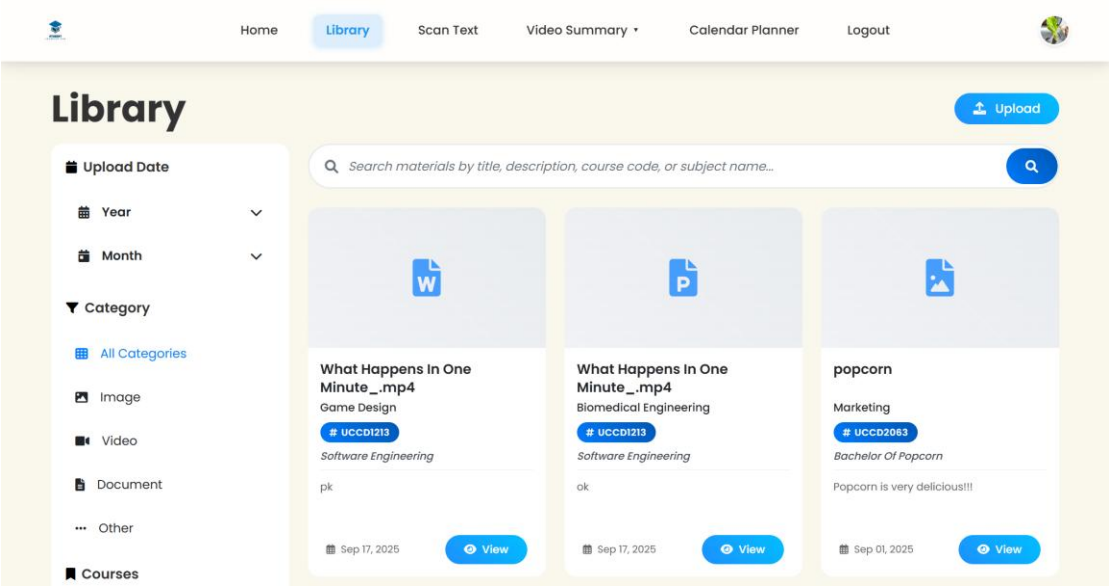


Figure 5.5.6 Library Page "Upload" Button

Figure 5.5.7 shows the upload form to upload materials. Users are required to fill in the materials details including material title, course code, subject name and material file. After uploading a file, the AI course classification feature will be enabled. Users must also select the privacy settings: Public or Private. By default, the privacy setting is set to Public.

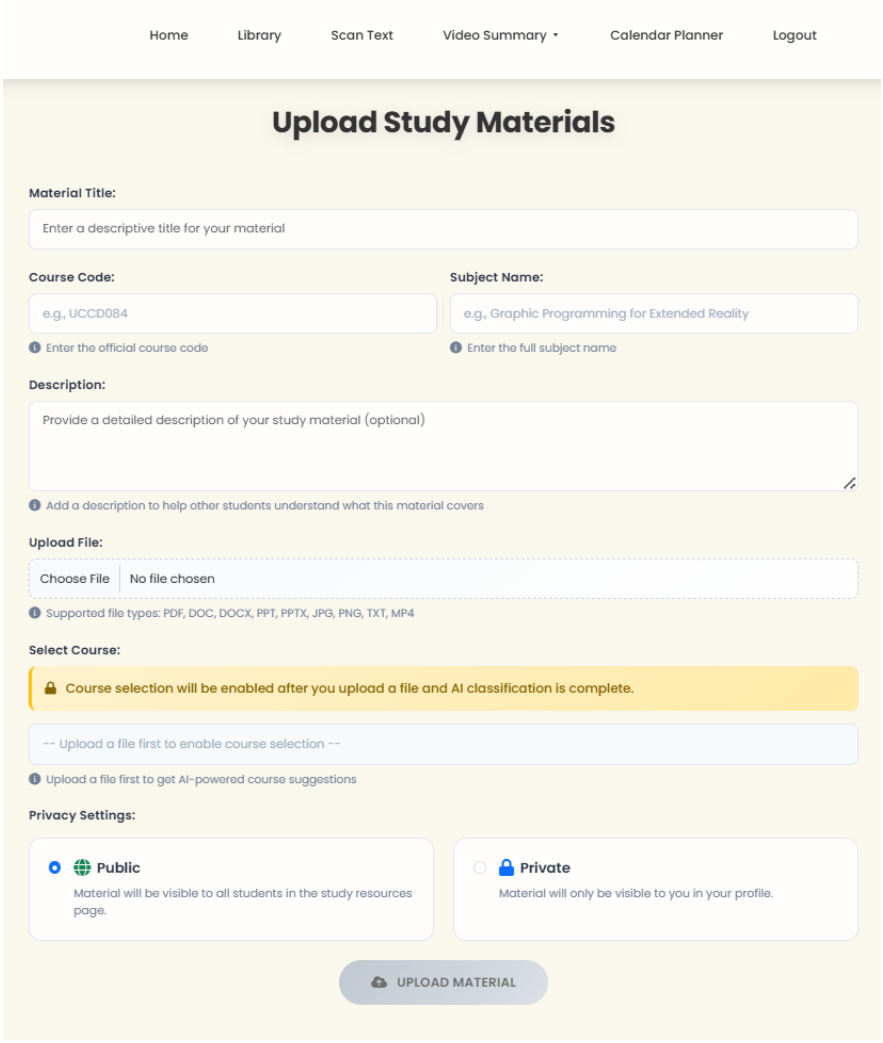


Figure 5.5.7 Upload Form

Figure 5.5.8 shows the AI course classification result. After users upload a material, the AI course classification is enabled, and the classification result is shown along with the course suggestions. After classification is complete, the "Select Course" dropdown will be enabled to let user to modify the course selection if they are not satisfied with the AI classification result. By clicking the "Upload Material" button, the material will be uploaded, and a success message will be displayed.

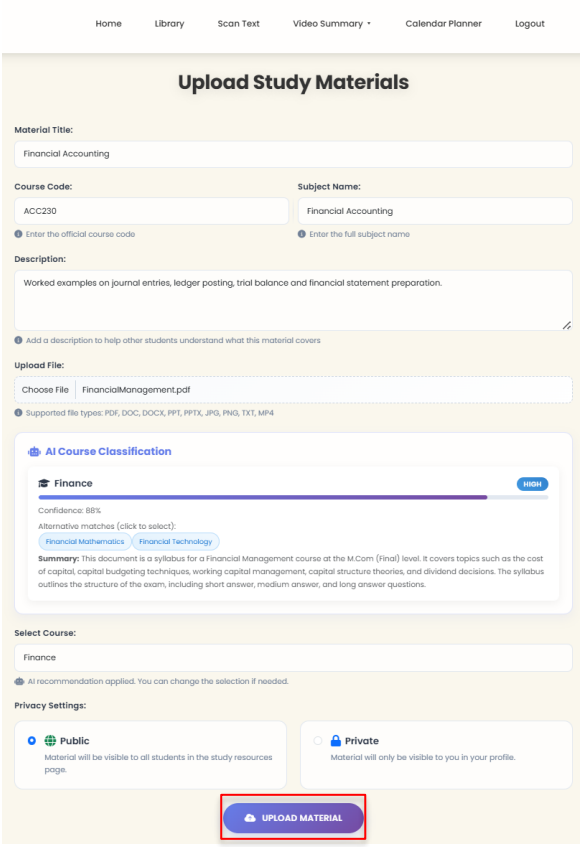


Figure 5.5.8 Upload Form with AI Course Classification Result

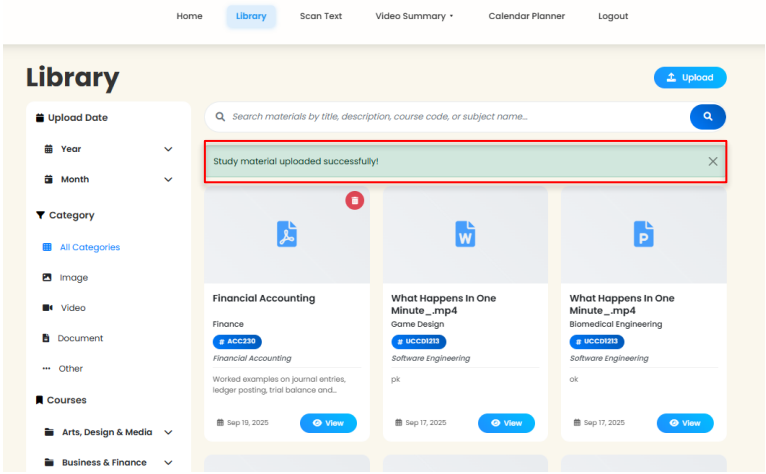


Figure 5.5.9 Upload Material Success Message

5.5.5 Optical Character Recognition (OCR)

Figure 5.5.10 shows the Optical Character Recognition (OCR) Interface with a "Choose Image" button. By clicking the "Choose Image" button, a file explorer will pop out. Figure 5.5.11 shows file explorer window where users can select an image file to extract text.

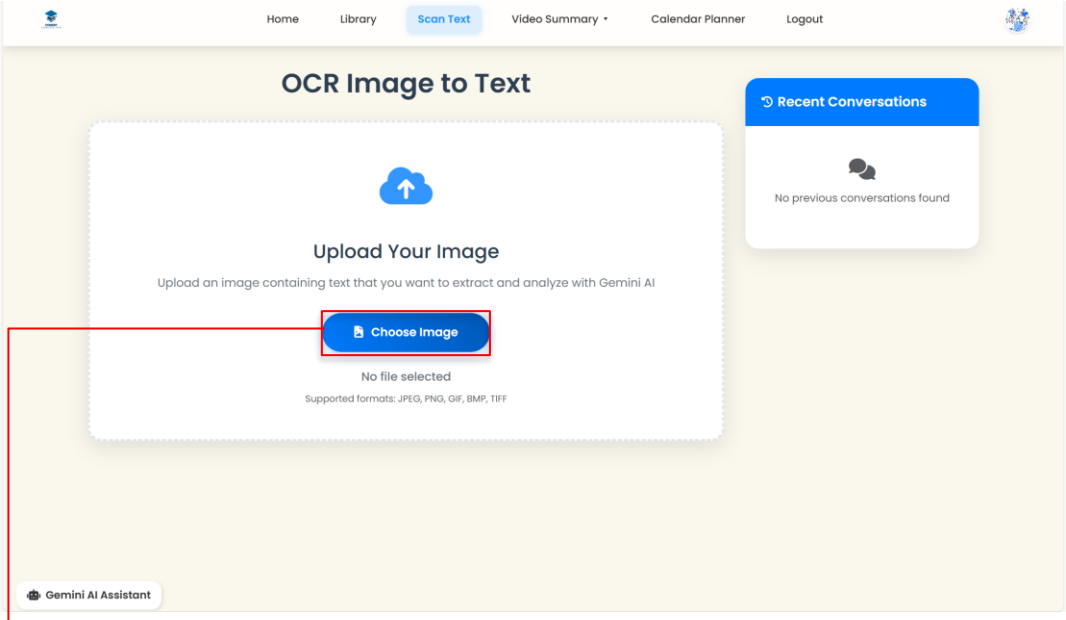


Figure 5.5.10 Optical Character Recognition (OCR) Interface

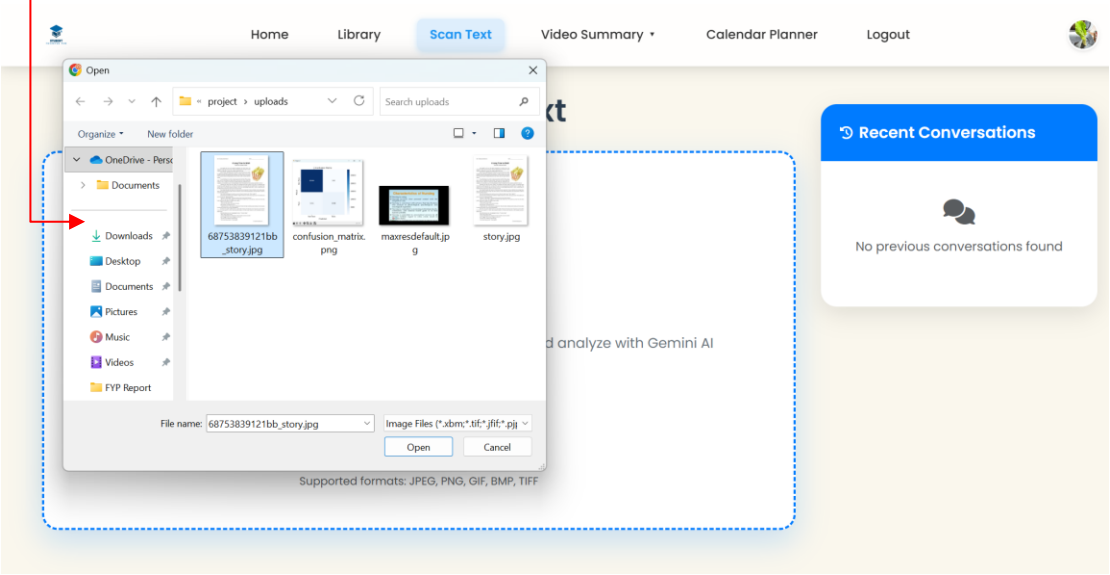


Figure 5.5.11 OCR Upload Image File

After uploading an image file, the OCR will begin to extract text from image. Figure 5.5.12 shows the OCR extraction result along with four buttons: “Edit”, “Copy Text”, “Download”, “Upload Material”. At the bottom of the page, the Gemini AI Assistant is displayed, allowing users to ask questions about the extracted text and get answers. The right sidebar shows the recent conversations which allow users to review their previous conversation with the Gemini AI assistant.

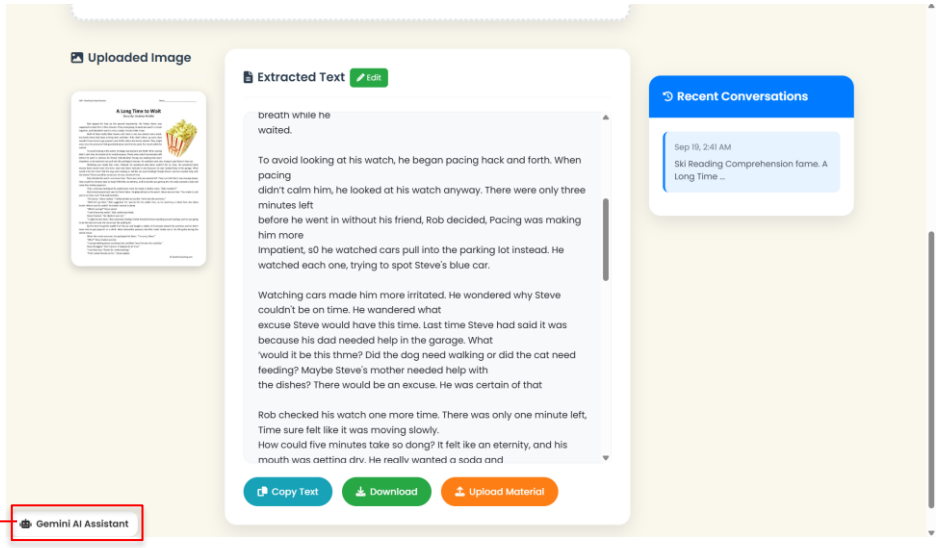


Figure 5.5.12 OCR Extraction Result

Figure 5.5.13 shows the pop-up Gemini AI Assistant chatbot. After clicking the Gemini AI Assistant, the AI chatbot will pop-up and users are allowed to ask questions.

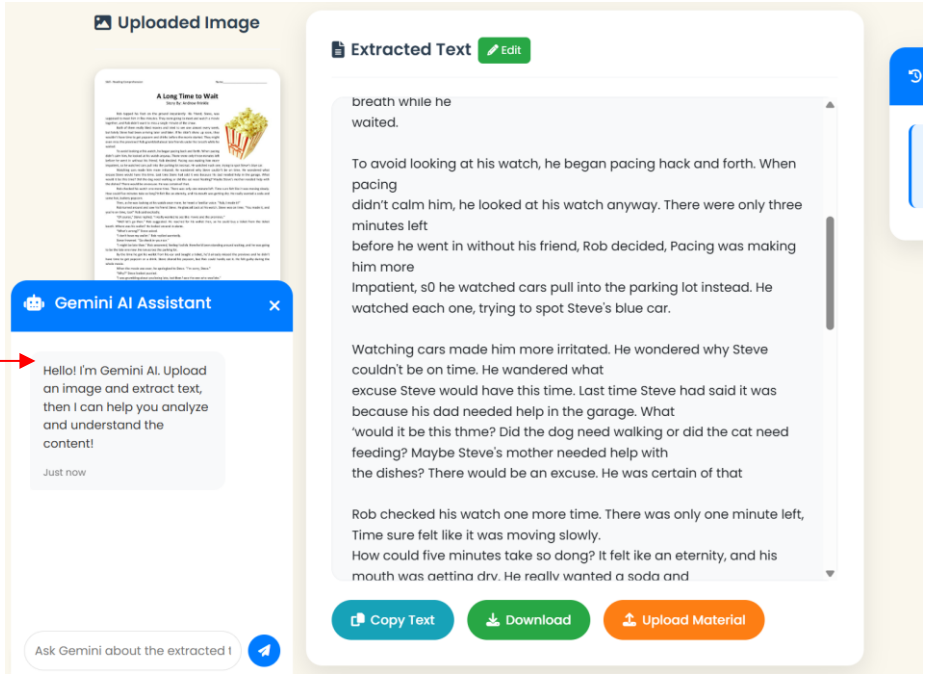


Figure 5.5.13 Gemini AI Assistant

Figure 5.5.14 shows a question asked by the user and the answer provided by the Gemini AI Assistant.

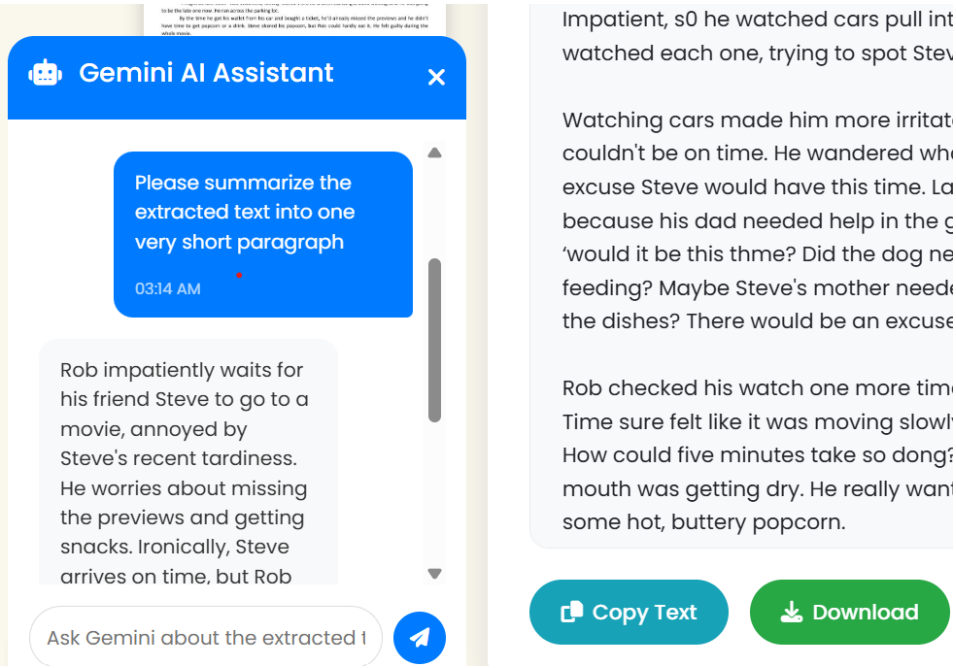


Figure 5.5.14 Gemini AI Assistant Question and Answer

If a user has multiple conversations with the AI chatbot on the different extracted text contents, they can track back recent conversations by clicking on the conversation they want to review. Figure 5.5.15 and Figure 5.5.16 shows the pop-up Gemini AI Assistant after the user clicks on a specific conversation from “Recent Conversation” sidebar on the right.

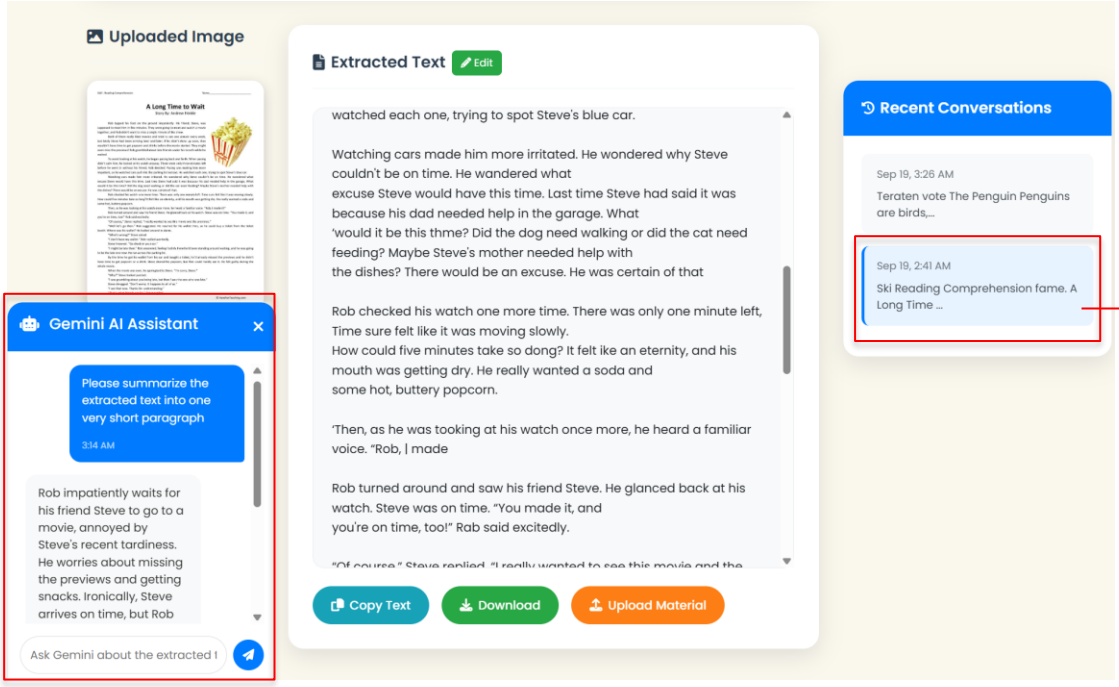


Figure 5.5.15 Recent Conversations with Gemini AI Assistant 1

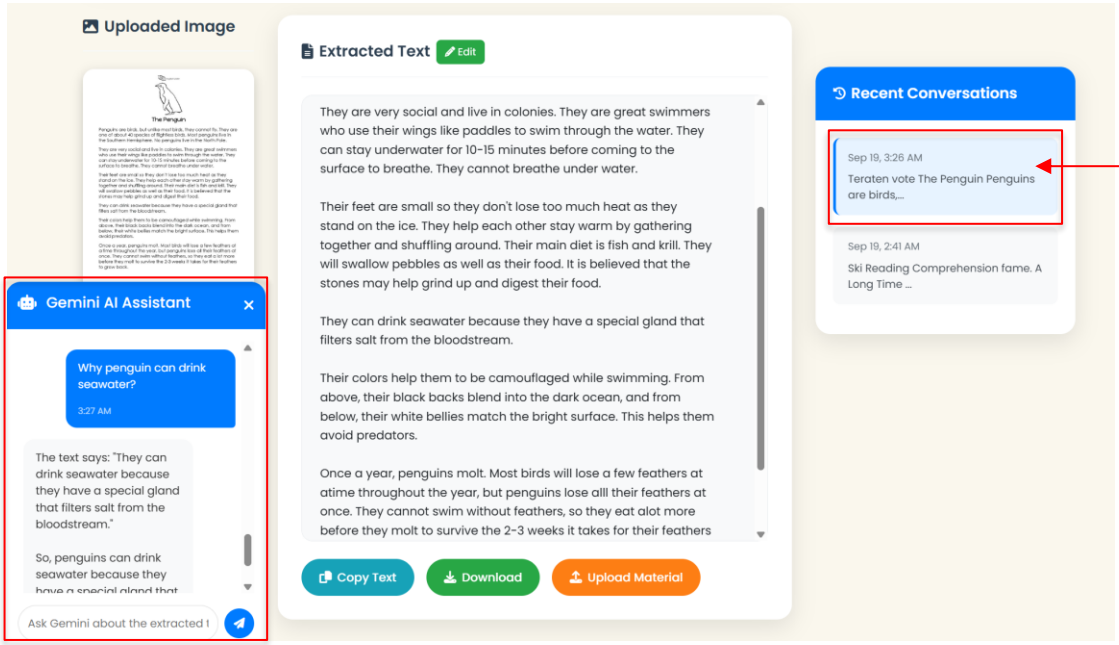


Figure 5.5.16 Recent Conversations with Gemini AI Assistant 2

5.5.6 Uploaded Video Content Summarization

Figure 5.5.16 shows the user interface for the Uploaded Video Content Summarization feature, which includes a drag and drop upload form, transcription tips sidebar, a step-by-step process guide, and a pro tip section for YouTube video summarization.

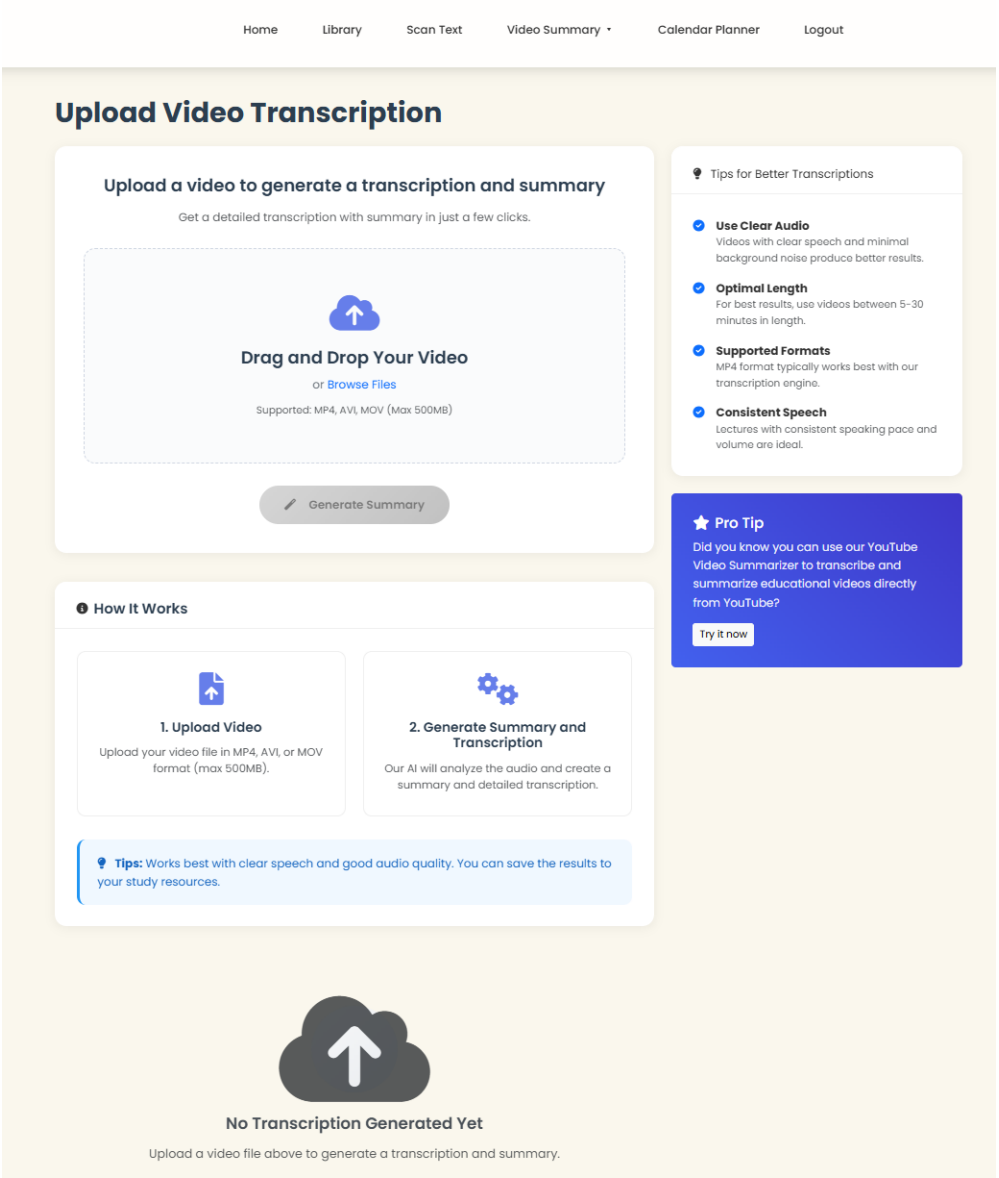


Figure 5.5.17 User Interface of Uploaded Video Content Summarization

To generate a summary and transcription from a video, users must first upload a video. By clicking the “Browse File” button, a file explorer window will appear, and users are allowed to upload a video file. Another way is that users can choose to drag and drop a video file into the upload form. Figure 5.5.17 shows the uploaded video file preview with the “Generate Summary” button enabled.

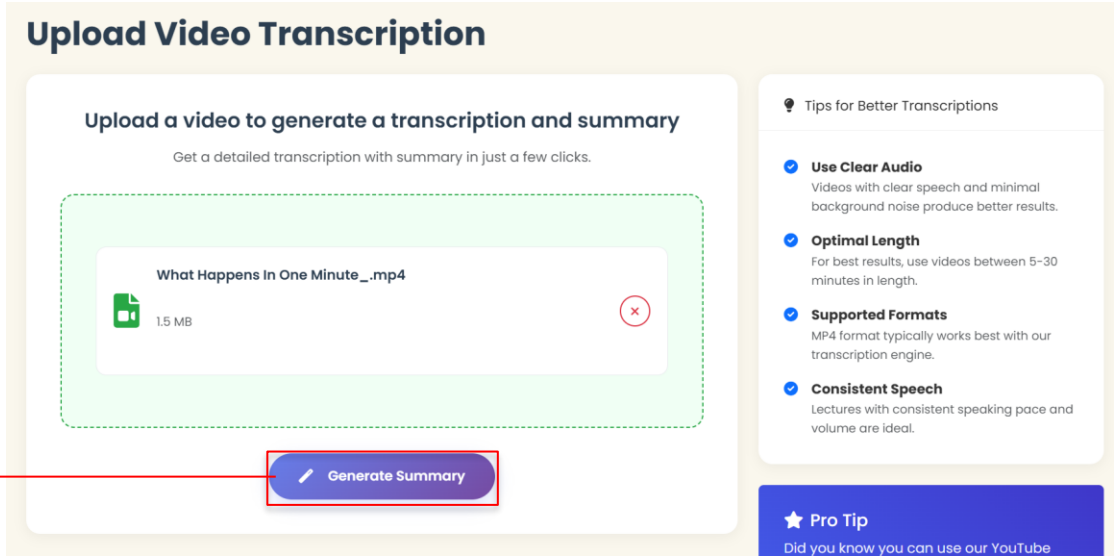


Figure 5.5.18 Upload Form Preview

By clicking the “Generate Summary" button, a loading screen will appear. Figure 5.5.18 shows the loading screen displayed while generating the summary and video transcript.

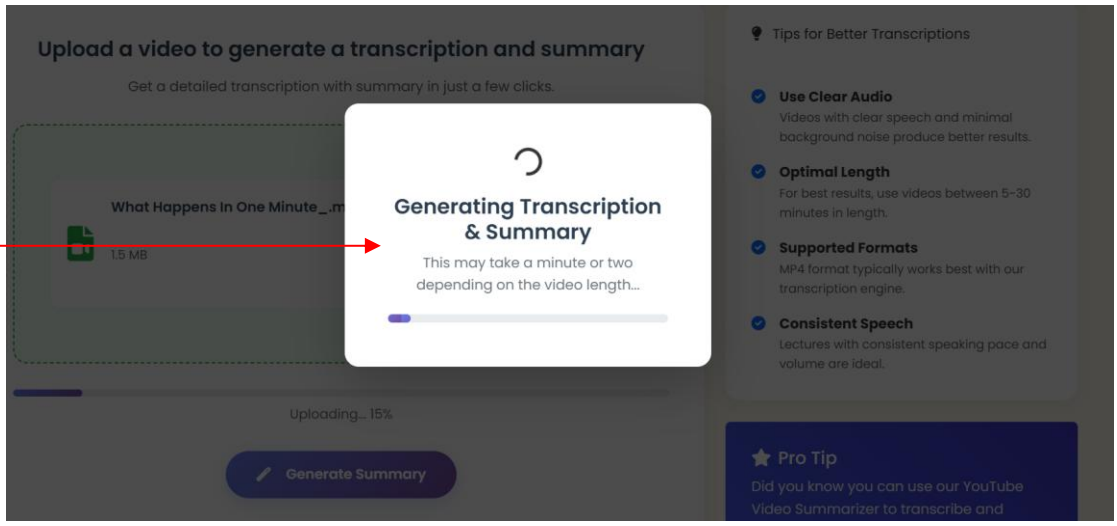


Figure 5.5.19 Loading Screen

Figure 5.5.19 shows the Uploaded Video Summarization result, which includes two tabs: Video Summary and Full Transcript. Users can choose to upload the video along with its summary and transcript into the Library page by clicking the “Save to Study Resources” button. Alternatively, users can download the video summary and transcript by clicking the “Download Transcription” button.

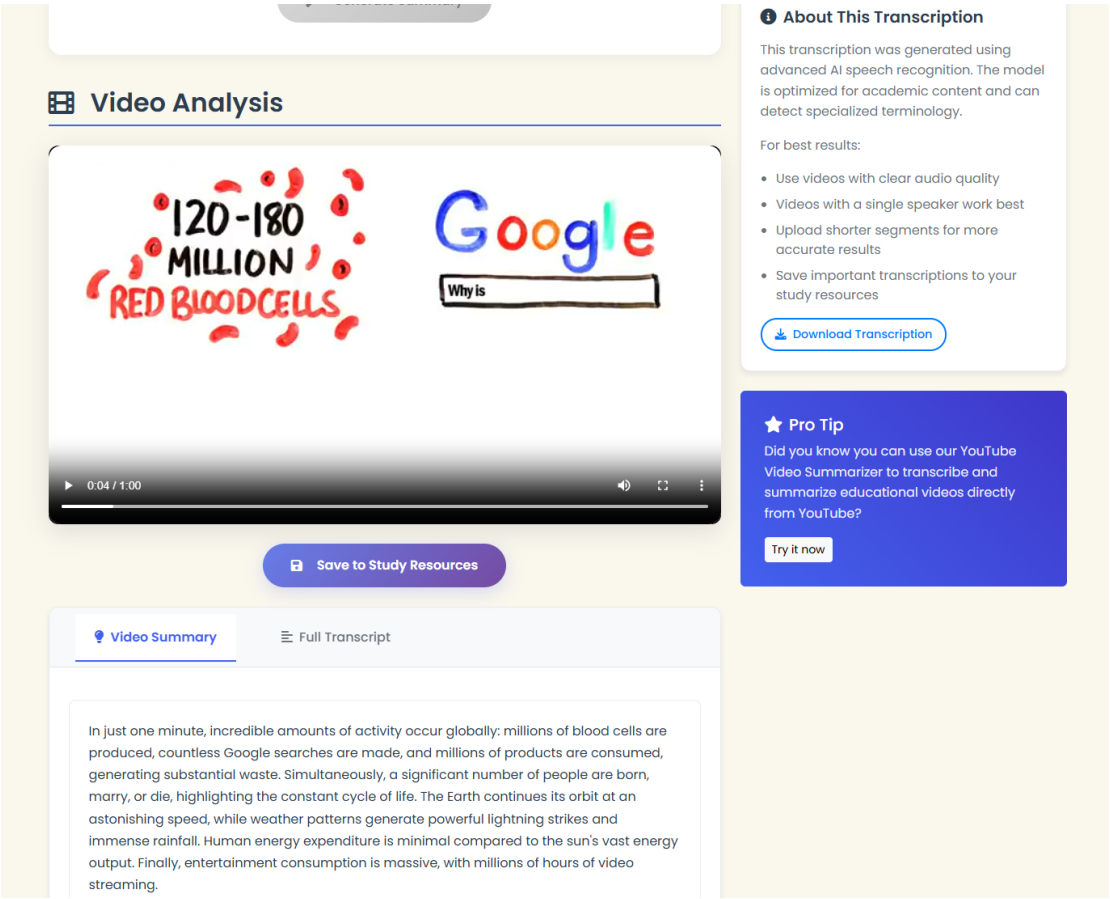


Figure 5.5.20 Uploaded Video Summarization Result Video Summary Tab

To view the Full Transcription, users can click the Full Transcript tab. Figure 5.5.20 shows the Uploaded Video Summarization Result with Full Transcript Tab.

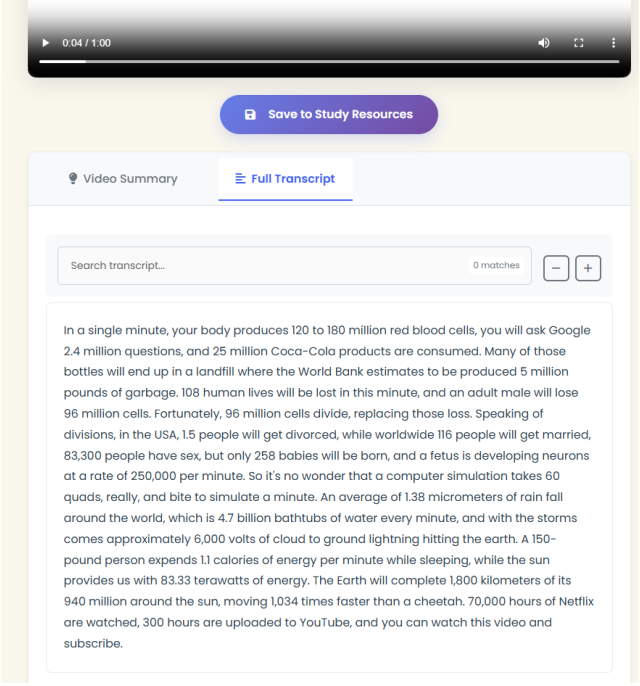


Figure 5.5.21 Uploaded Video Summarization Result Full Transcript Tab

By clicking the "Download Transcription" button, a text file with video summary and transcription will be downloaded. Figure 5.5.21 shows the downloaded text file and its content.

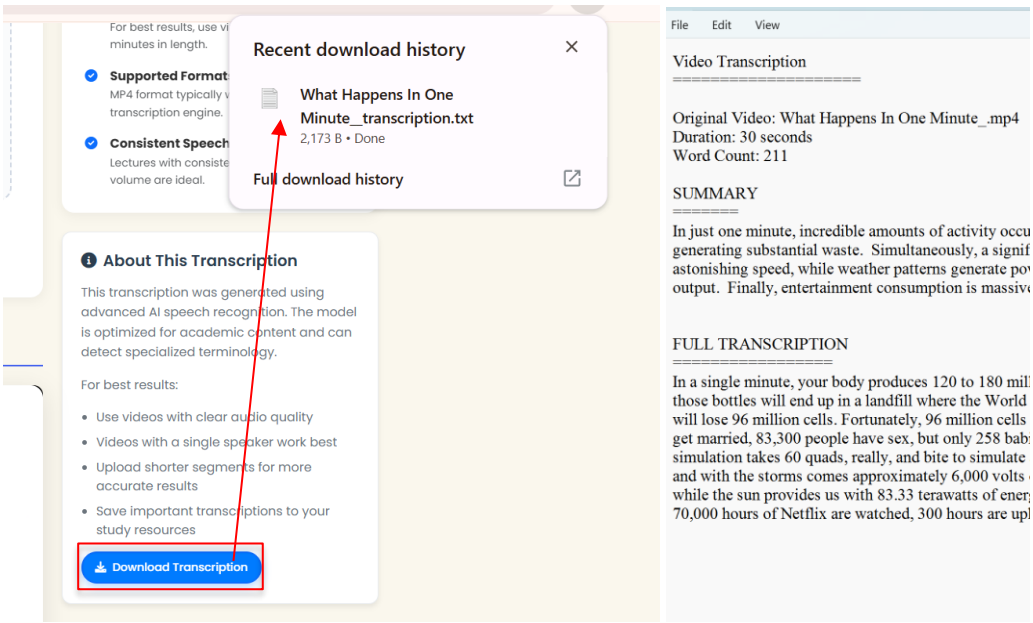


Figure 5.5.22 Uploaded Video Summarization Download Transcription

5.5.7 YouTube Video Content Summarization

Figure 5.5.22 shows the YouTube Video Content Summarizer interface with a YouTube URL link pasted. To generate a video summary and transcript, users need to paste a YouTube URL link into the input field and click the “Generate Summary” button.

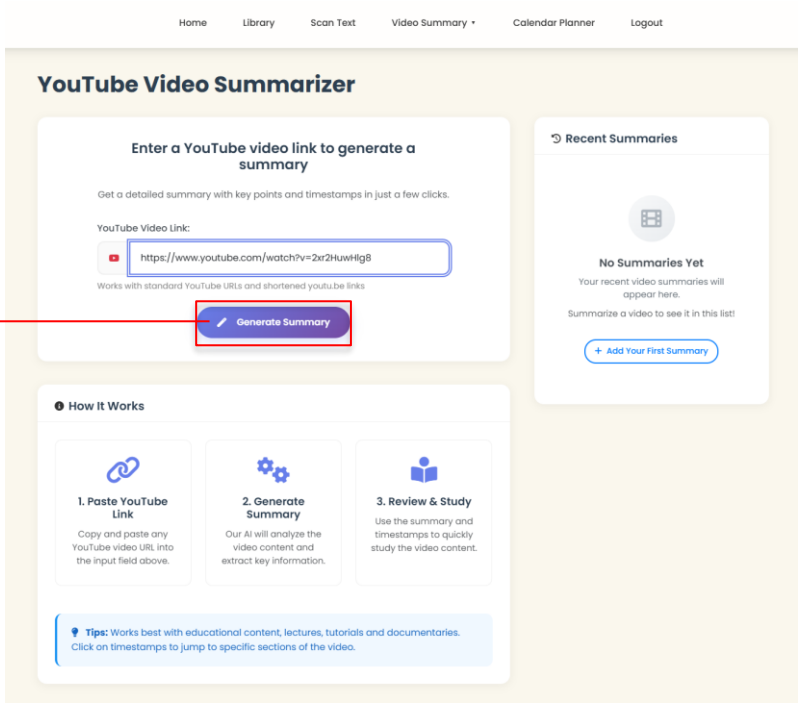


Figure 5.5.23 YouTube Video Content Summarization Interface

After clicking the “Generate Summary" button, a loading screen will appear. Figure 5.5.23 shows the loading screen while generating the YouTube video summary and transcript.

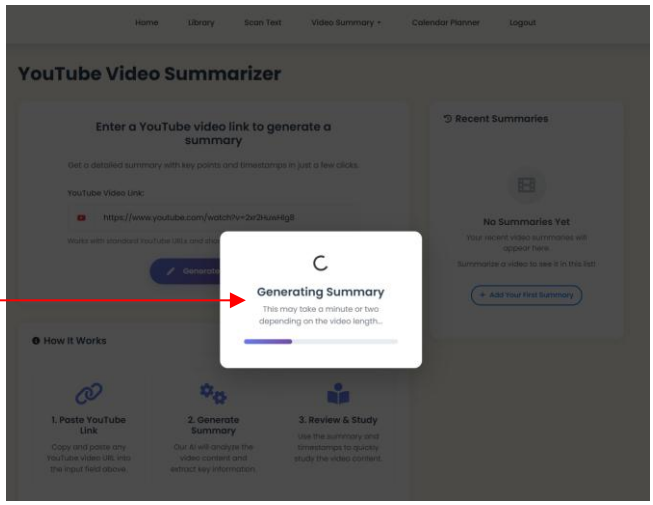


Figure 5.5.24 YouTube Video Content Summarization Loading Screen

Figure 5.5.24 shows the result of the YouTube video summary, including key timestamps and the video transcript. Users can click the “Play” button on key timestamps to jump directly for that specific video timestamps in the YouTube video. They can also click the “View” button to see the transcript for that specific timestamp. Additionally, users can upload the YouTube video along with its summary and transcript to the Library page by clicking the “Save to My Study Resources” button.

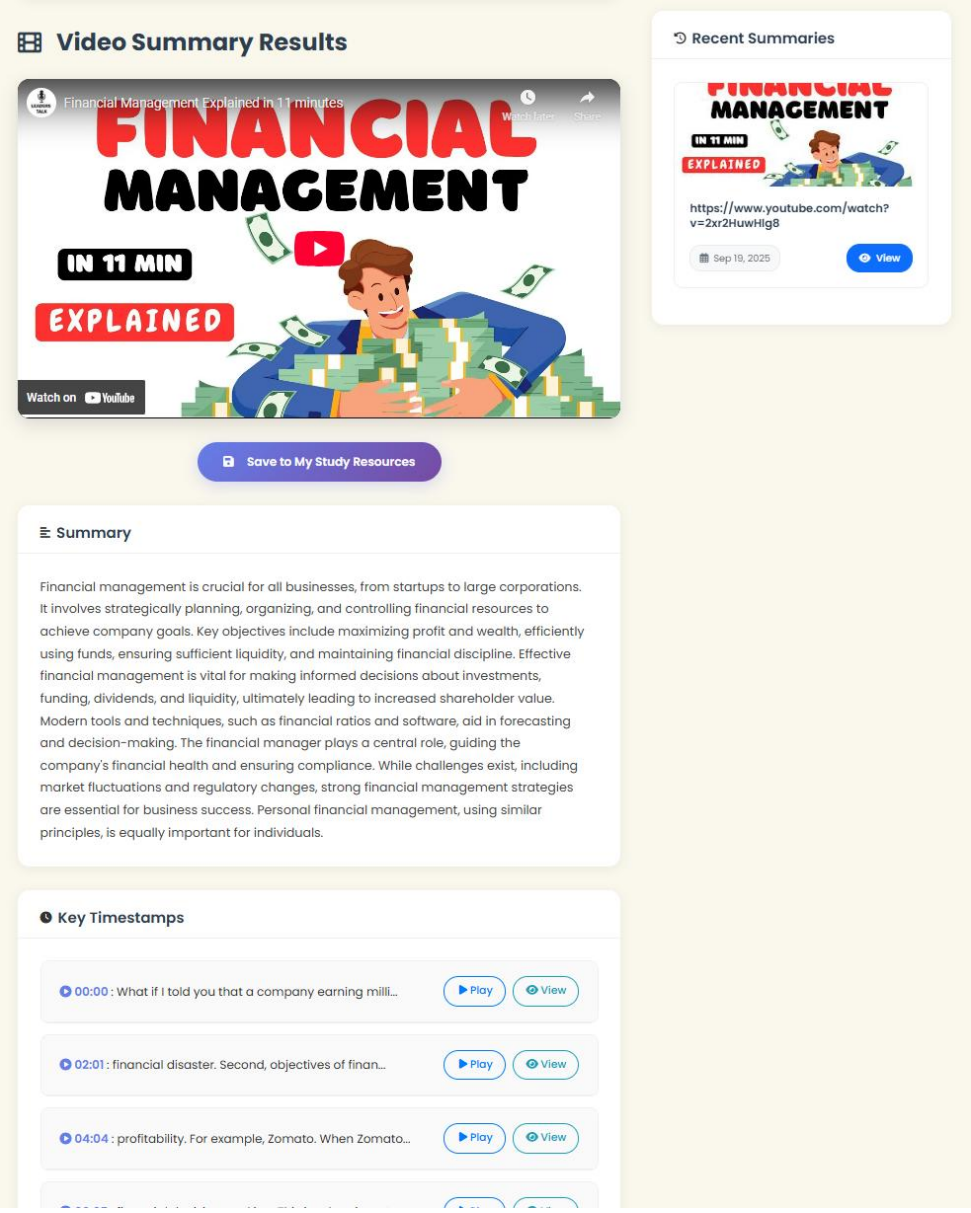


Figure 5.5.25 YouTube Video Summary Results

Figure 5.5.25 shows the transcript for a specific timestamp after the user clicks the "View" button.

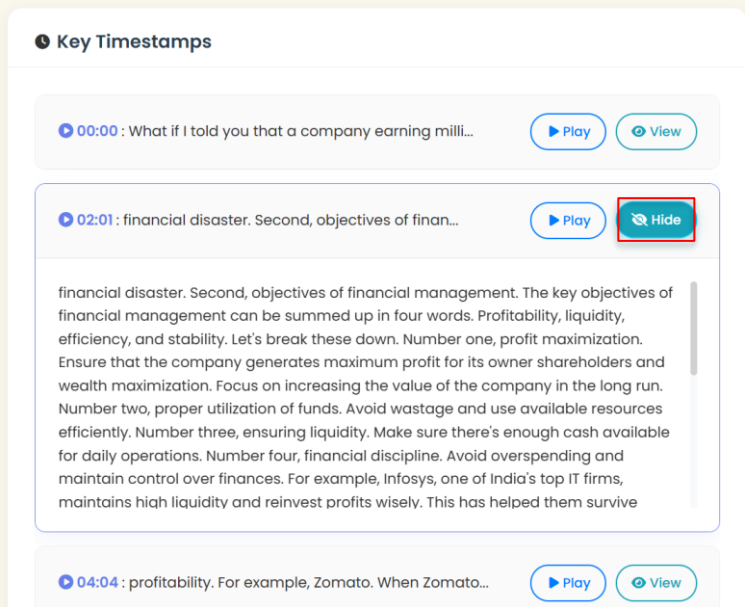


Figure 5.5.26 YouTube Video Summary Specific Timestamp Transcript

Users can re-generate the YouTube video summary and transcript by clicking the "View" button from the "Recent Summaries" sidebar on the right. Figure 5.5.26 shows the "Recent Summaries" sidebar with the YouTube video that been summarized and transcribed.

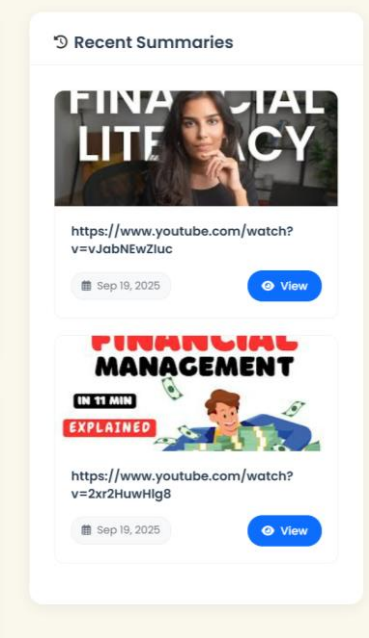


Figure 5.5.27 YouTube Video Summary Recent Summaries Sidebar

5.5.8 Commenting Feature

Figure 5.5.27 shows the interface of the View Content Page with the commenting feature filled with a comment. Users can enter their thoughts in the comment field and click the “Post Comment” button to submit their thoughts.

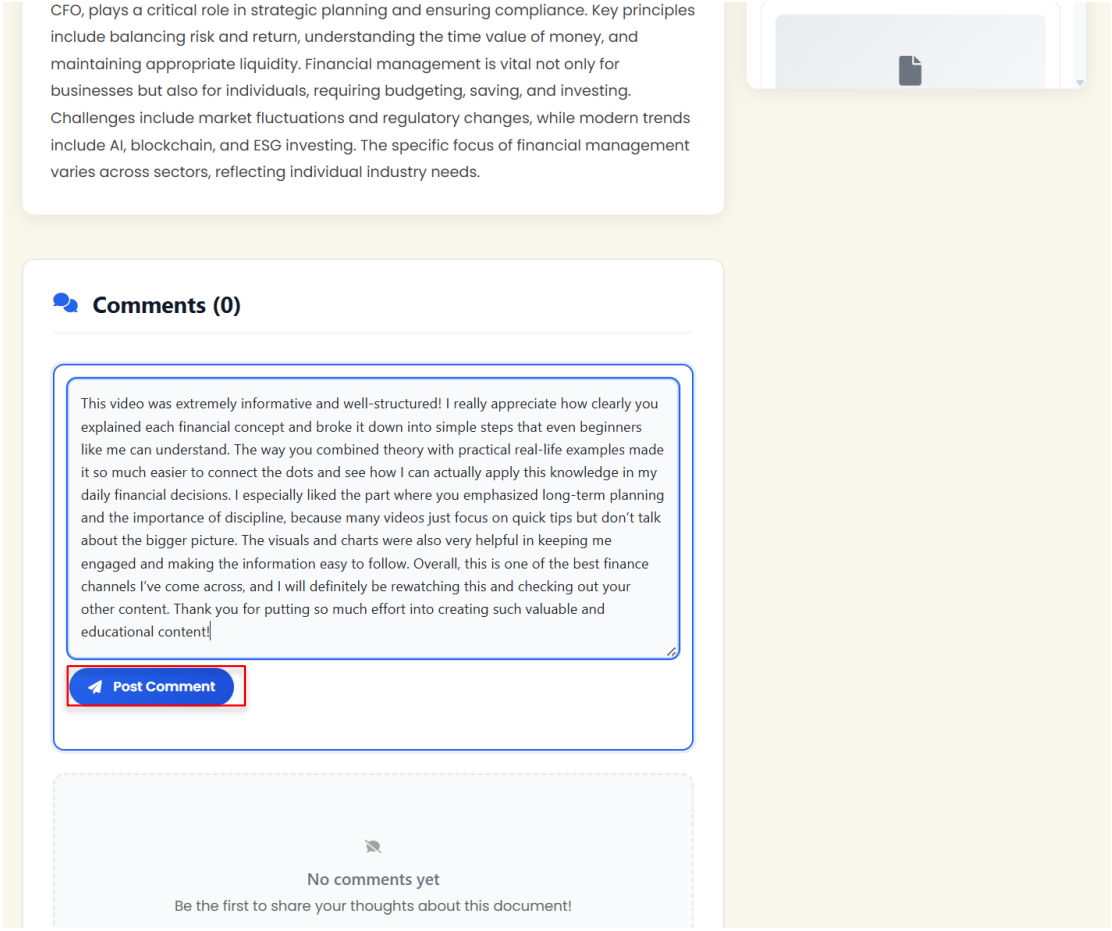


Figure 5.5.28 View Content Page Commenting Feature

After clicking the "Post Comment" button, the comment will be posted if no hate speech is detected. Figure 5.5.28 shows a positive comment uploaded along with the sentiment indicator. To manage the comment, there have three buttons: "Like", "Edit" and "Delete", which allow users to like comment, edit and delete their own comment.

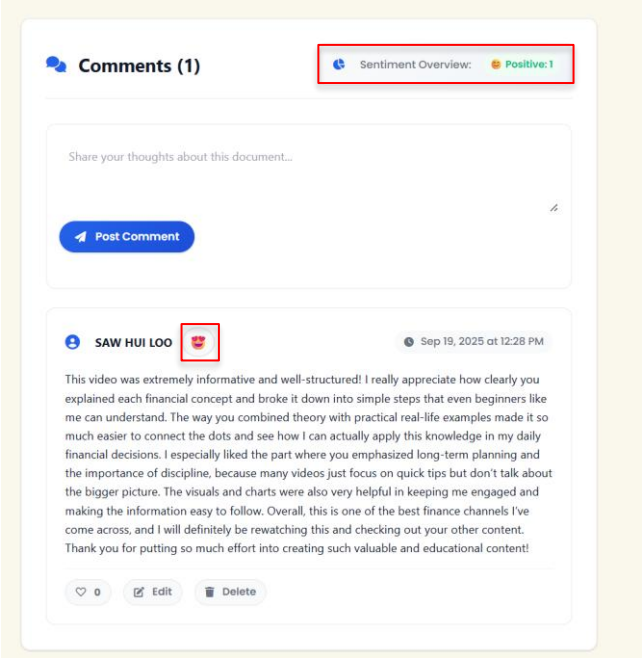


Figure 5.5.29 Uploaded Comment with Positive Sentiment Indicator

If users upload a negative comment without any hate speech detected, the comment will be posted with a negative sentiment indicator. Figure 5.5.29 shows a negative comment uploaded with a negative indicator.

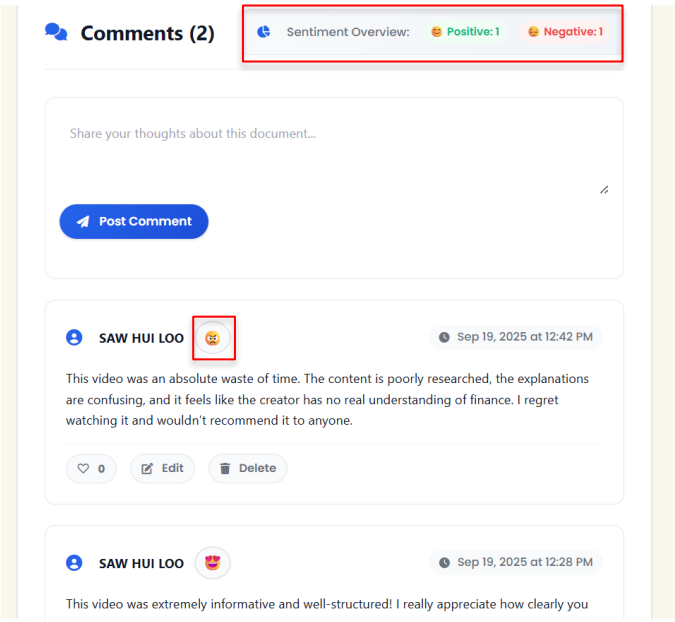


Figure 5.5.30 Negative Comment with Negative Indicator

If users want to upload a negative comment that includes hate speech, the comment will not be posted, and an error message will appear. Figure 5.5.30 shows a negative comment containing hate speech, while Figure 5.5.31 shows the comment upload failed with an error message displayed.

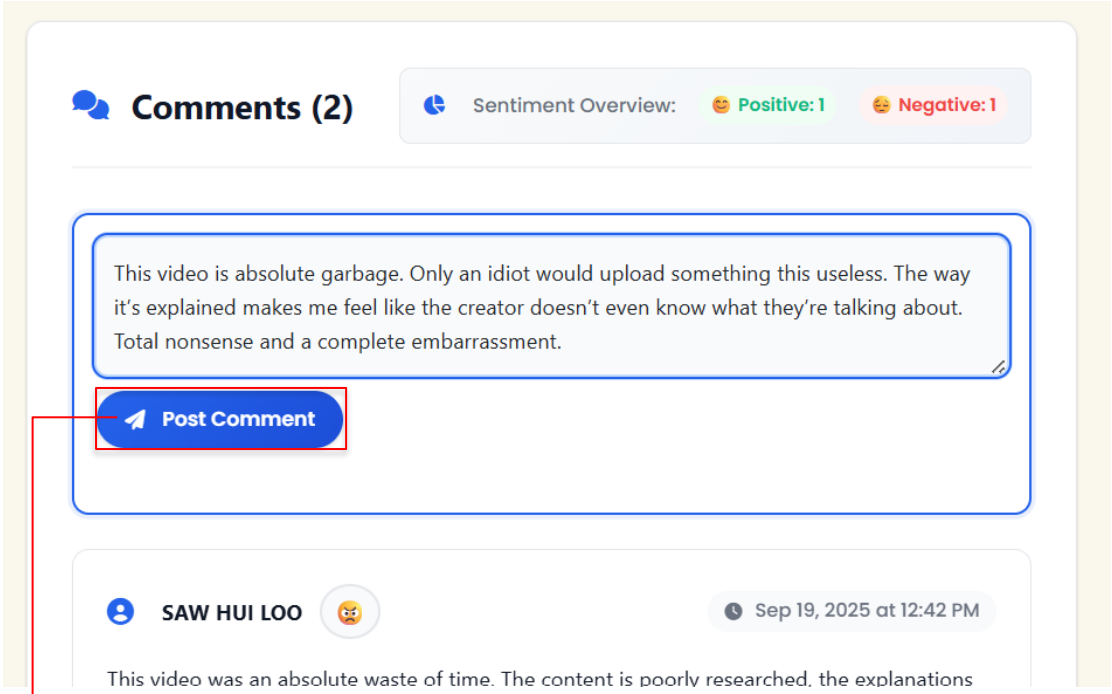


Figure 5.5.31 Negative Comment with Hate Speech

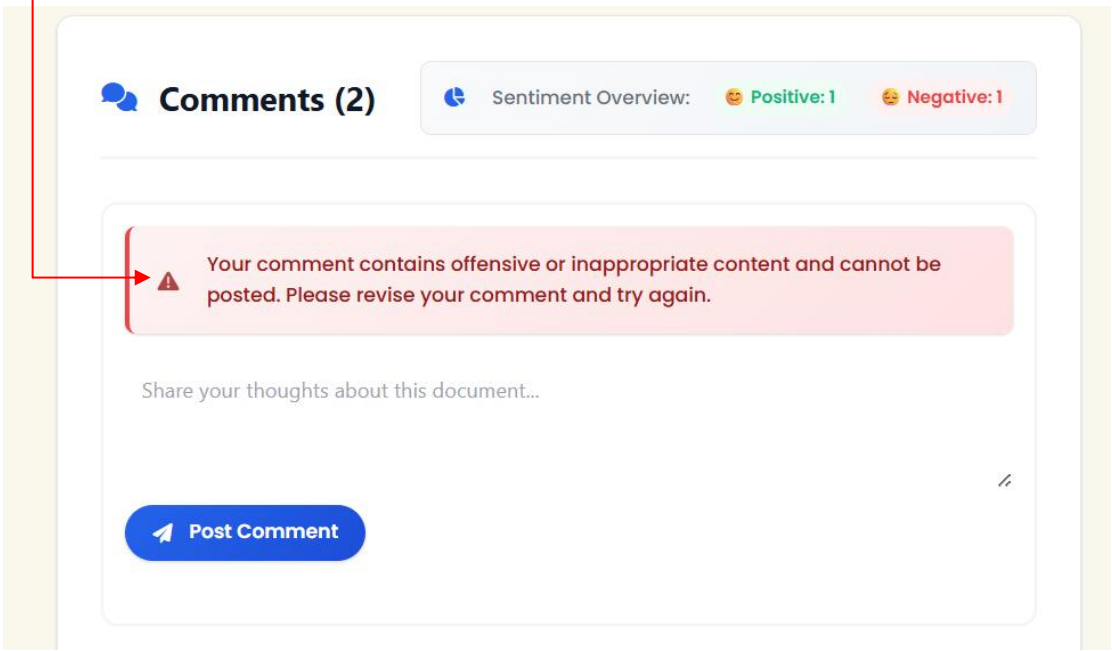


Figure 5.5.32 Comment Upload Failed

5.5.9 Smart Calendar Planner

Figure 5.5.32 shows the Smart Calendar Planner Interface. By clicking the "+New Task" button, users can create an AI-scheduled task, and the timeslot will be automatically allocated by the AI.

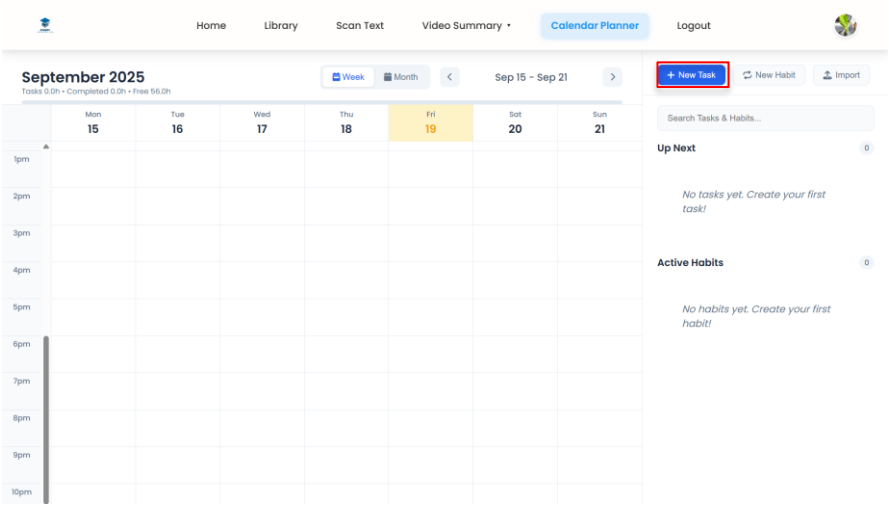
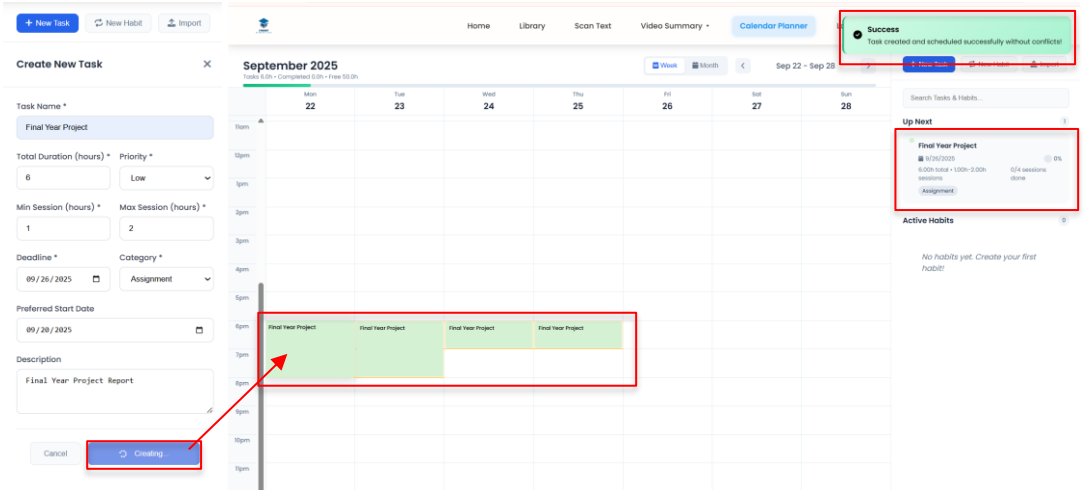


Figure 5.5.33 Smart Calendar Planner Interface

Figure 5.5.33 shows the Create New Task form with information filled in and the successfully AI-scheduled task created. To create an AI-schedule task, users need to fill in all the required information and click the "Create & Schedule" button. Once submitted, the task will be created, and a success message will be displayed.



Firgure 5.5.34 Create AI-Schedule Task

To create a task in specific timeslot, users need to click on the specific timeslot column in the calendar view, and the "Create Task" form will appear. Figure 5.5.34 shows the form with all task information filled in. After clicking the "+Create Task" button, a task with be created for the selected timeslot. Figure 5.5.35 shows the success message displayed after the task is successfully created.

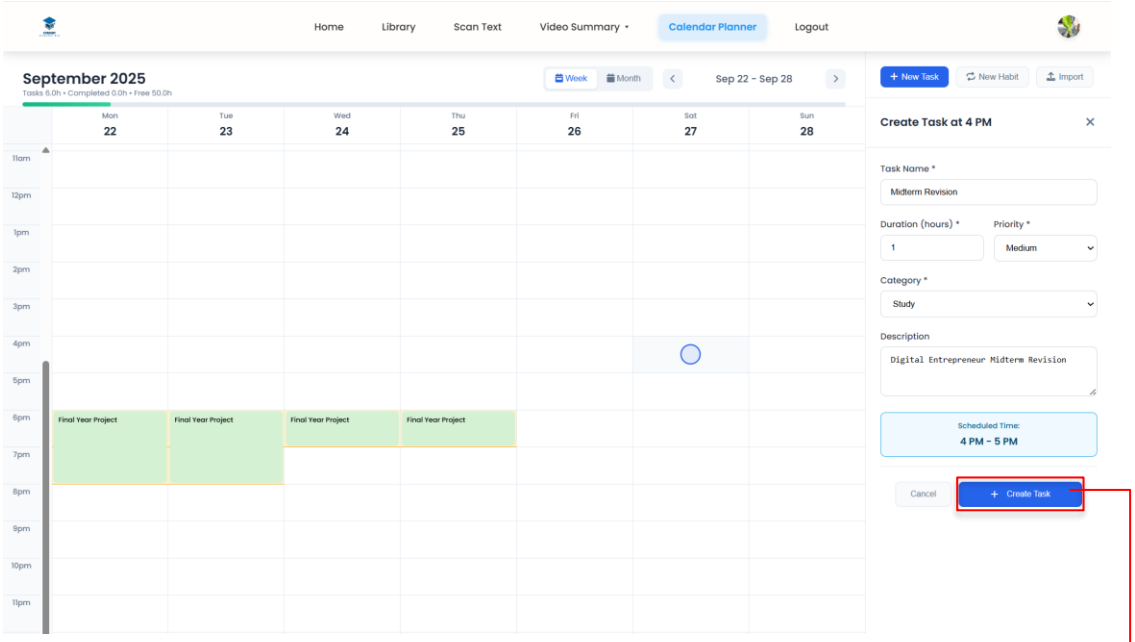


Figure 5.5.35 Create Task Form with Task Information Filled

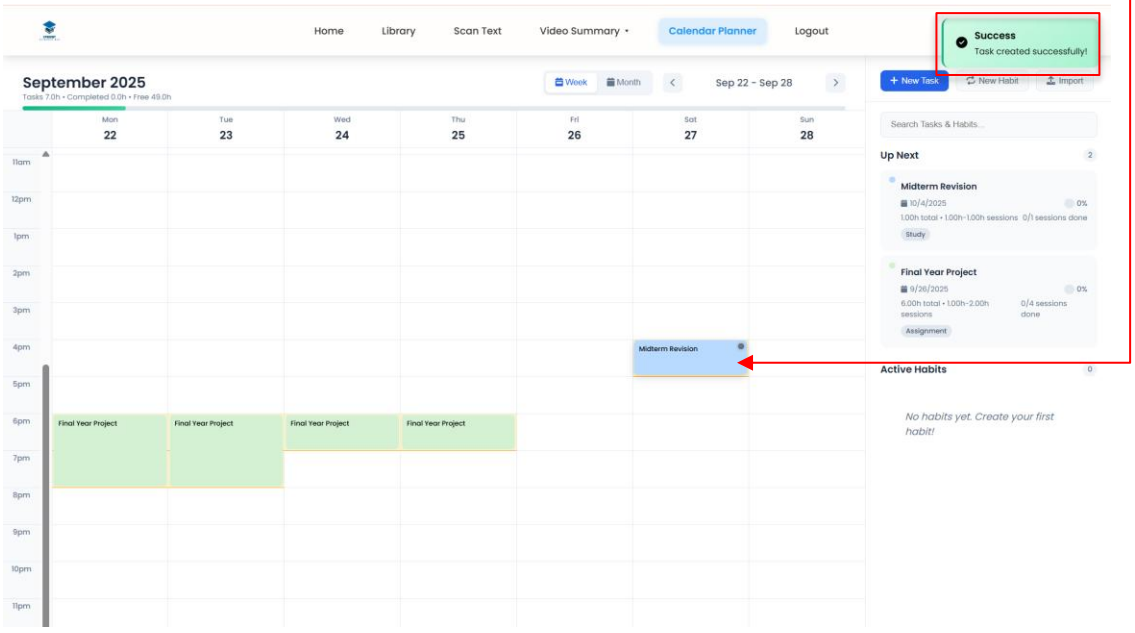


Figure 5.5.36 Task Created Success Message

Figure 5.5.36 shows the “Import” button. When clicked, an Import Timetable upload form pops up, which allows users to import their timetable into the calendar.

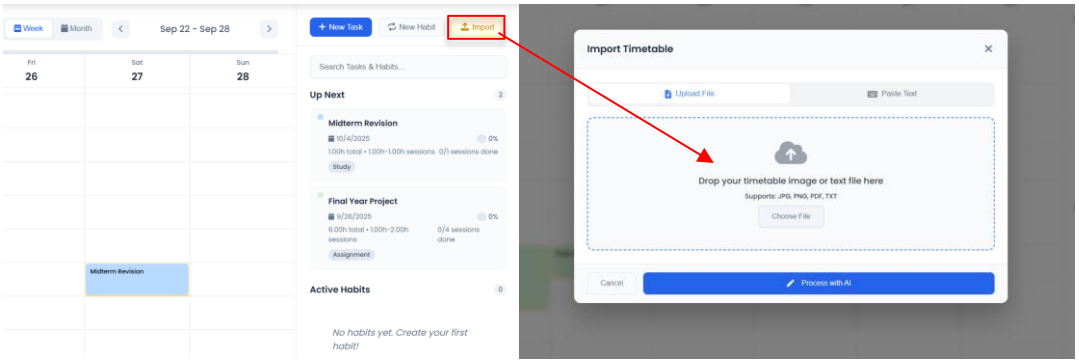


Figure 5.5.37 Import Timetable Upload Form

By clicking the “Choose File” button, a file explorer window will be pop out and user are able to upload their timetable for import. Figure 5.5.37 shows the file explorer window that appears after clicking the “Choose File” button and the uploaded image preview after upload an image.

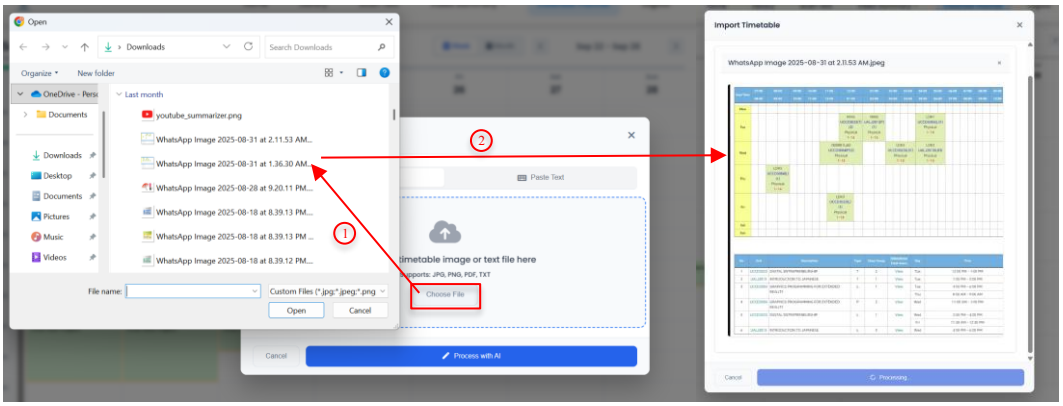


Figure 5.5.38 File Explorer Window and Uploaded Timetable Preview

Figure 5.5.38 shows the detected schedule from the uploaded timetable. Users can use the dropdown to select the start and end months for applying the timetable.

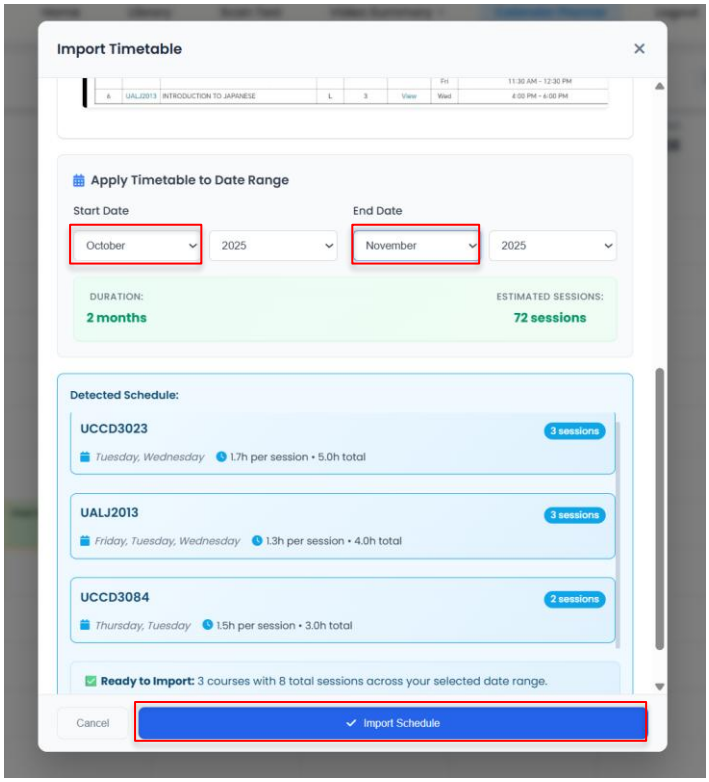


Figure 5.5.39 Detected Schedule from Uploaded Timetable

After clicking the "Import Schedule" button, the timetable will be successfully imported into the calendar, and a "Import Successful!" message will appear, as shown in Figure 5.5.38.

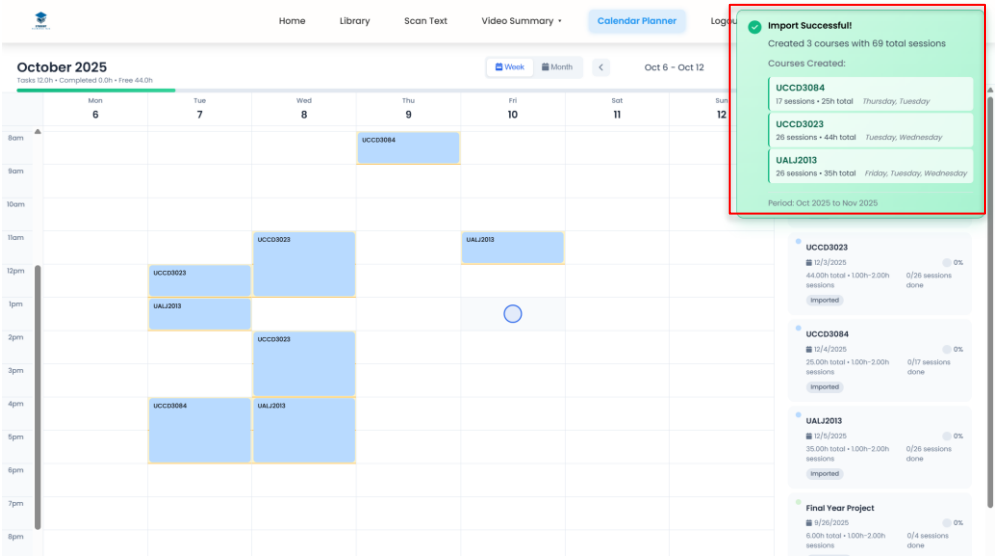


Figure 5.5.40 Timetable Import Successful Message

5.5.10 AI Study Tools

Figure 5.5.39 shows the View Material Page with a “Study with AI” button. After clicking the “Study with AI" button, users are directed to AI Study Tools Page.

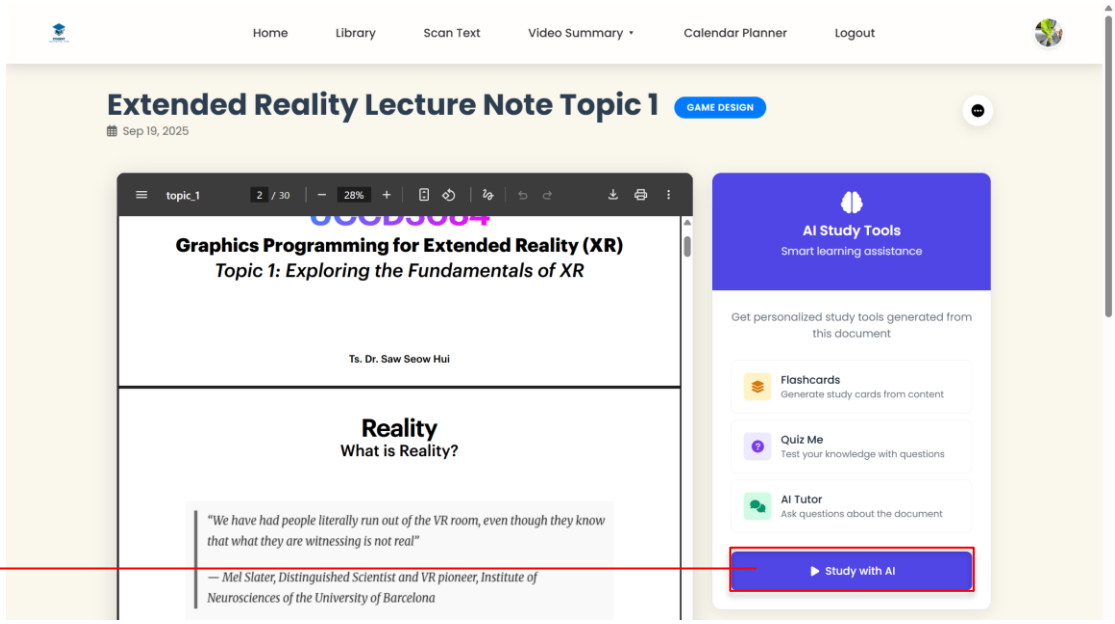


Figure 5.5.41 View Material Page with “Study with AI” Button

After clicking the “Study with AI” button, users will be directed to AI Study Tools page, specifically the Flashcards tab. Figure 5.5.40 shows the Flashcards displayed with questions.

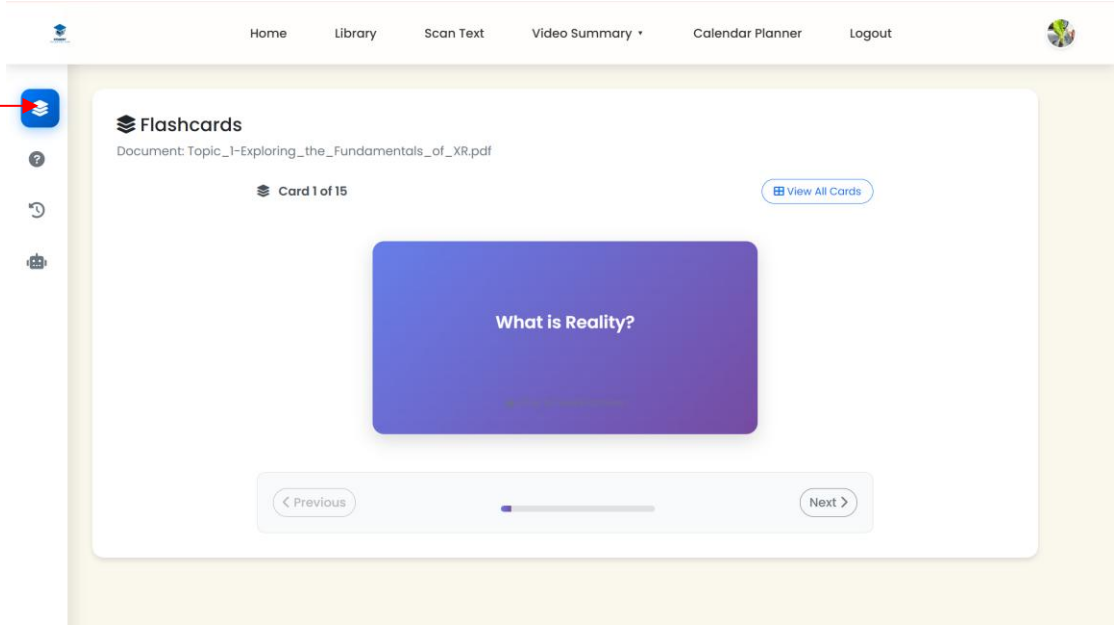


Figure 5.5.42 AI Study Tools Flashcards Tab

To view the flashcard answer, users need to click on the flashcard, and the answer will be displayed. Figure 5.5.41 shows the flashcard answer. To go to next question, users can click the “Next” button.

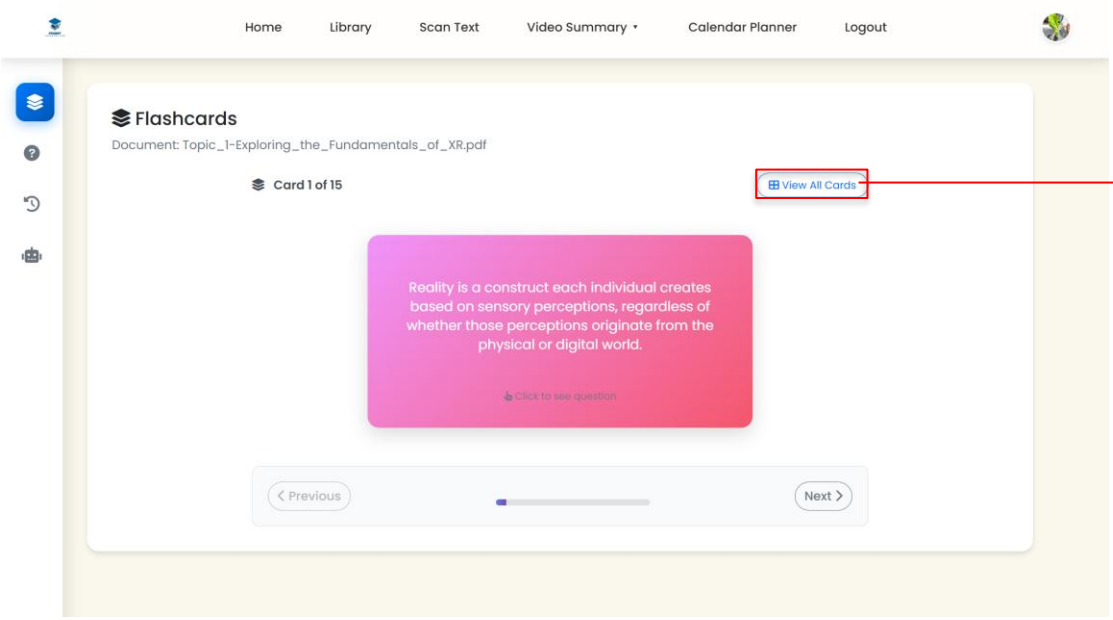


Figure 5.5.43 Flashcard Answer Display

If users want to view all the flashcards questions, they can click on the “View All Cards” button. Figure 5.5.42 shows all the flashcards, and users can also click on any card to view its answer.

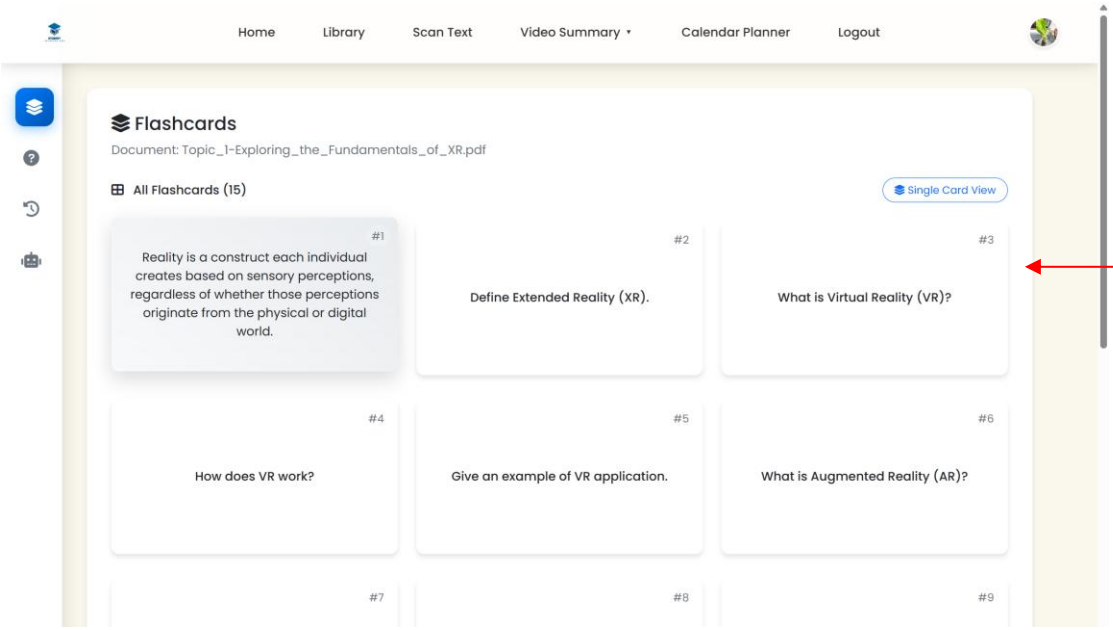


Figure 5.5.44 All Flashcards View

Figure 5.5.43 shows the AI Study Tools Quiz Tab, where users can select quiz, including the Number of Questions (10 Questions, 15 Questions and 20 Questions), Difficulty Level (Easy, Medium and Hard) and Question Type (Mixed, Multiple Choice, True/False and Fill in the Blank). To generate the quiz questions, users need to click the "Generate New Quiz" button, using either their selected settings or the default quiz settings.

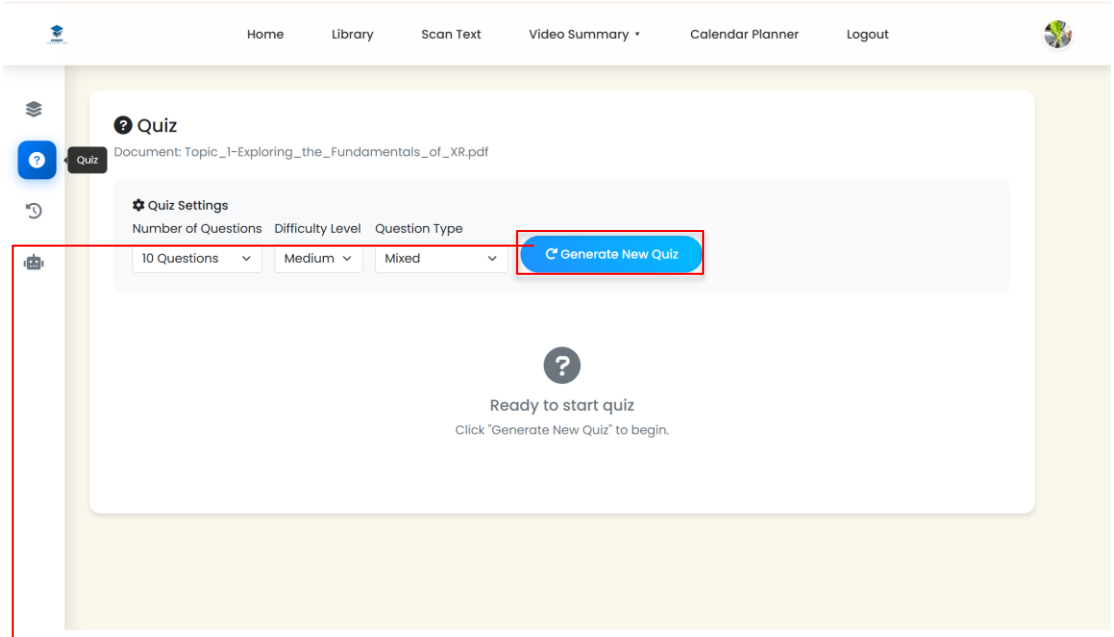


Figure 5.5.45 AI Study Tools Quiz Tab with Quiz Settings

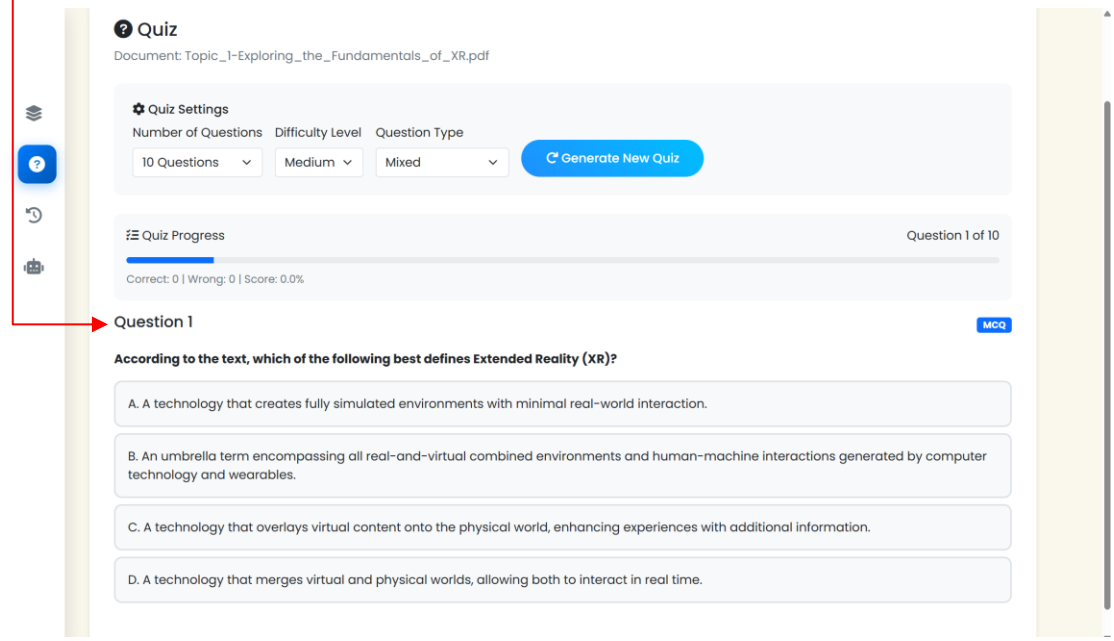


Figure 5.5.46 Generated Quiz

Figure 5.5.45 shows a question answered by the user along with its corresponding explanation.

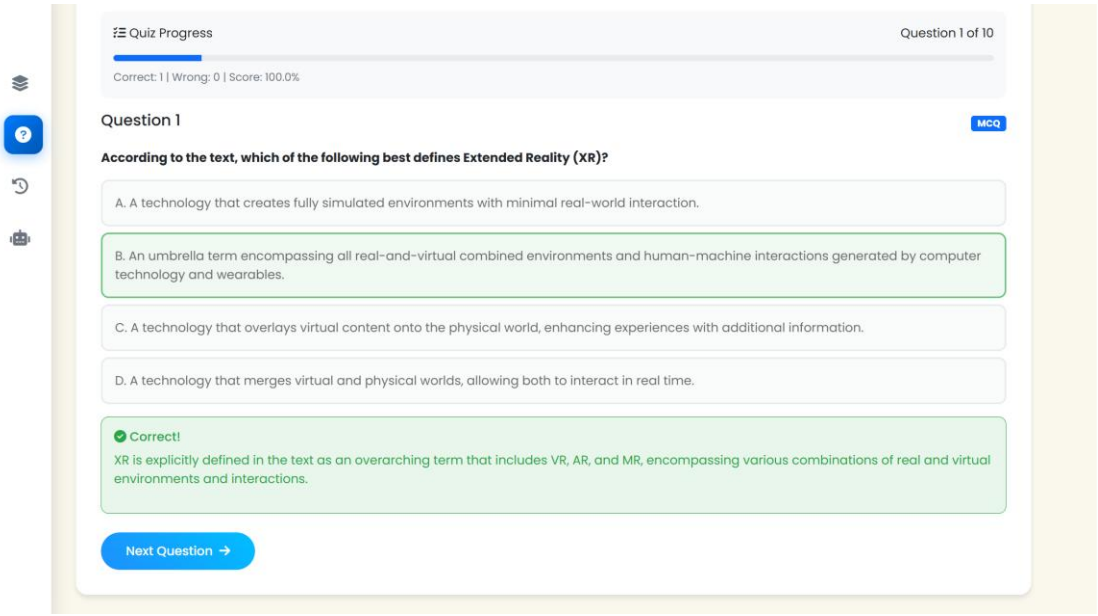


Figure 5.5.47 Answered Question with Explanation

After users have answered all the quiz questions, the result, showing correct and wrong answers, will be calculated and displayed. User can click the "Review Answers" button to view the quiz questions they have answered. Figure 5.5.46 shows the quiz result displayed and along with three buttons: "Take Another Quiz", "Review Answers" and "View History".

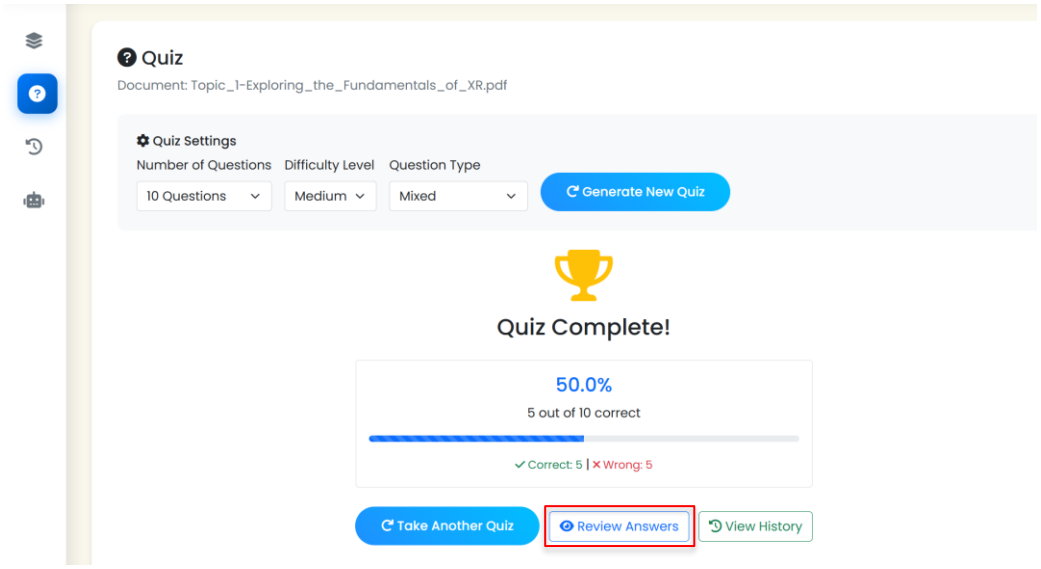


Figure 5.5.48 Quiz Results with Action Buttons

After clicking the “Review Answers” button, the Quiz Review page will be displayed, and user can review the quiz they have taken. Figure 5.5.47 shows the Quiz Review page.

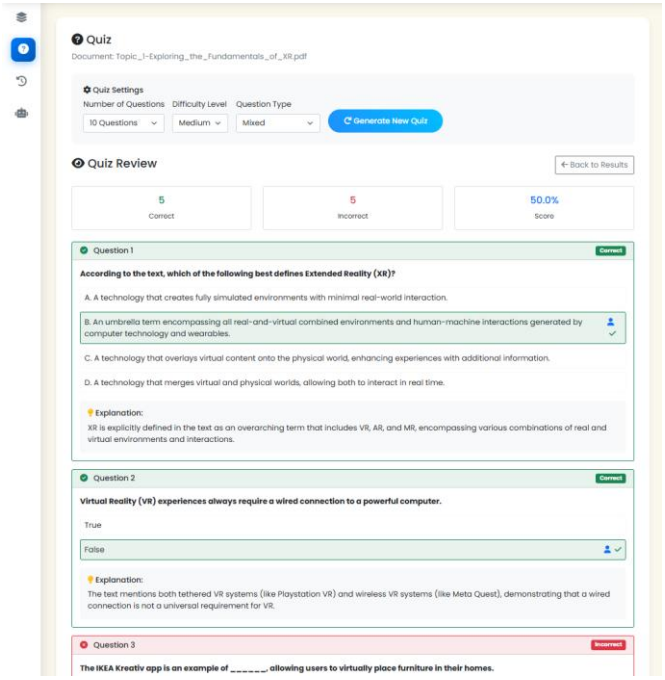


Figure 5.5.49 Quiz Review Page

Figure 5.5.48 shows the Quiz History Tab, which allows users to track their previously answered questions. By clicking the “Review Answers” button, users can review the questions, their answers and the corresponding explanations.

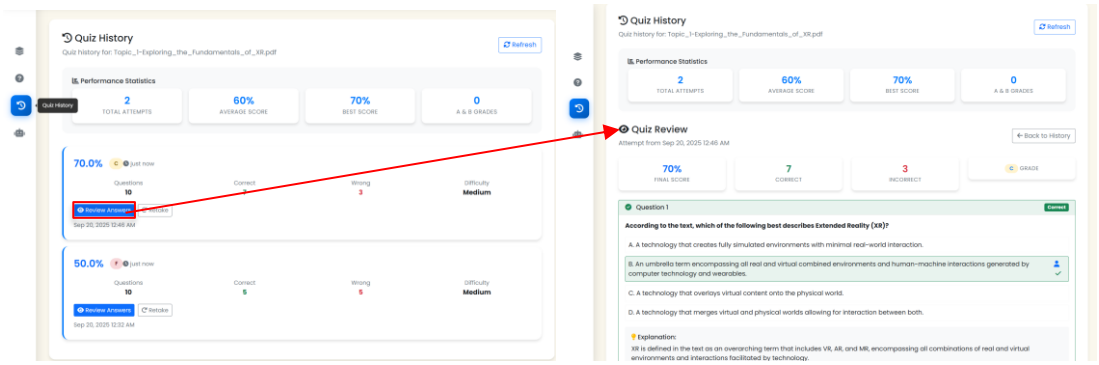


Figure 5.5.50 Quiz History Tab

Figure 5.5.49 shows the AI Assistant tab, displaying a question asked to the AI and the corresponding response provided.

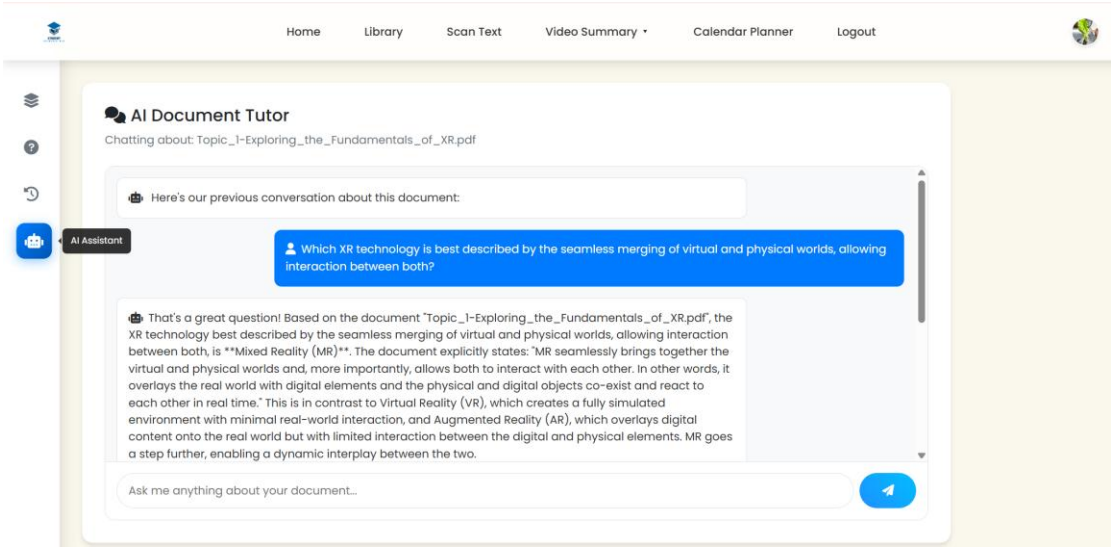


Figure 5.5.51 AI Assistant Tab

5.5.11 Forgot Password

Figure 5.5.50 shows the Login Page. If users forget password, they can click the "Forgot Password?" link to reset their password.

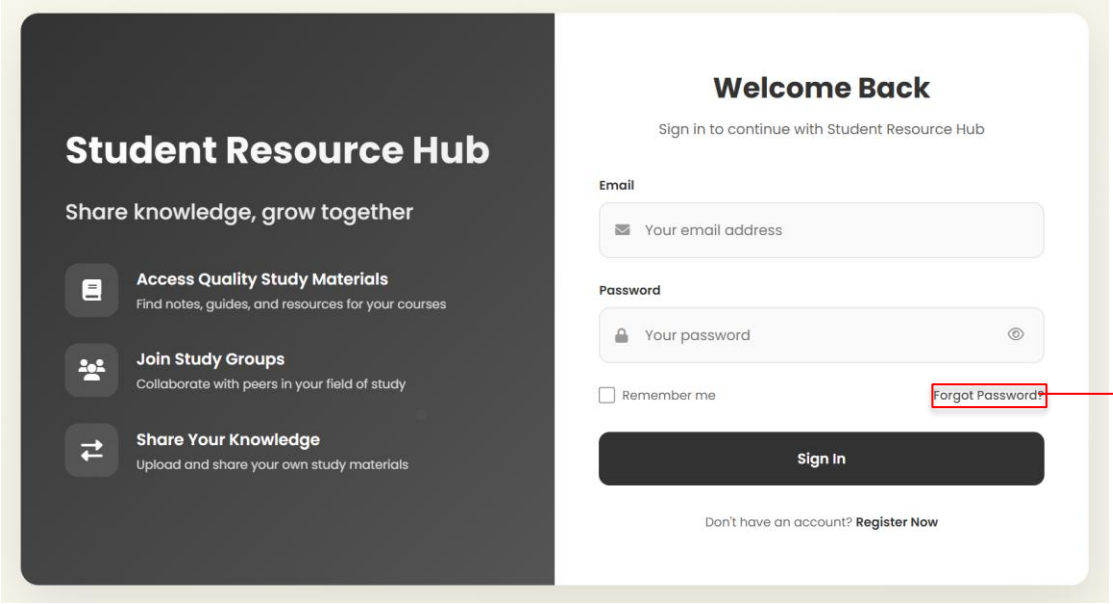


Figure 5.5.52 Login Page with Forgot Password Link

After clicking the "Forgot Password?" link, users will be directed to Forgot Password page and required to enter their email address. Figure 5.5.51 shows the Forgot Password page with an email address filled in. After filled in the email address, users need to click the "Send Reset Link" button.

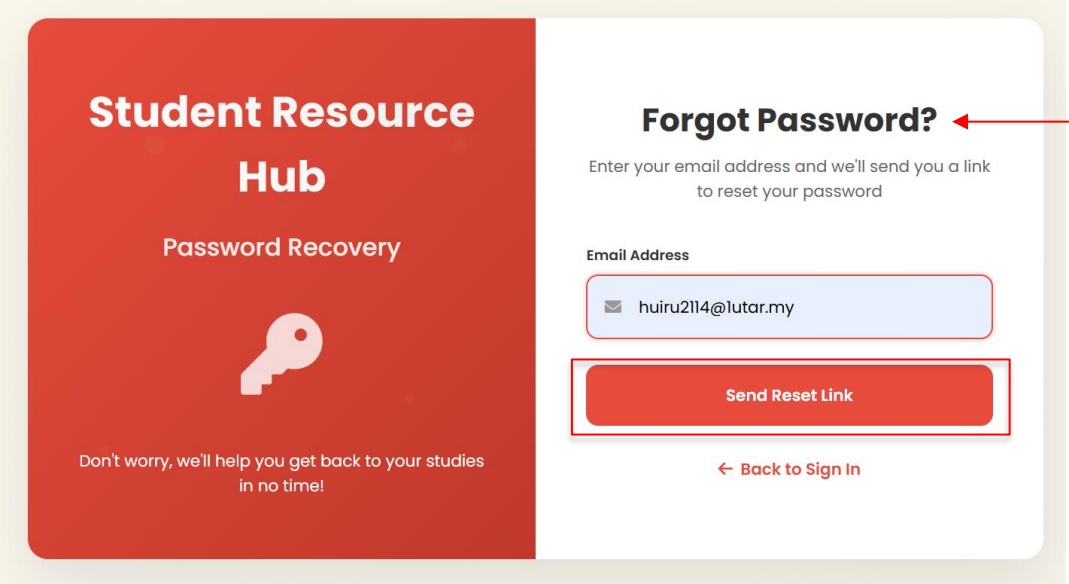


Figure 5.5.53 Forgot Password Page with Email Entered

After clicking the “Send Reset Link” button, a success message will appear, and a password reset link will be sent to the entered email address. Figure 5.5.52 shows the success message appeared after clicking the “Send Reset Link” button, and a password reset link will be sent to user’s email address. To reset password, users need to click on the “Reset Your Password” button in the email.

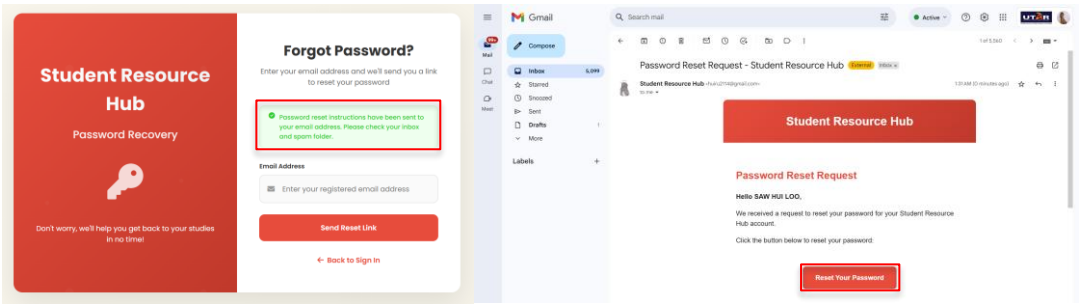


Figure 5.5.54 Password Reset Success Message and Reset Link Email

After clicking the “Reset Your Password” button, user will be directed to Create New Password Page. To reset password, users need to fill in a new password and click the “Update Password” button. Figure 5.5.53 shows the filled-in password reset form and success message displayed after a successful password reset.

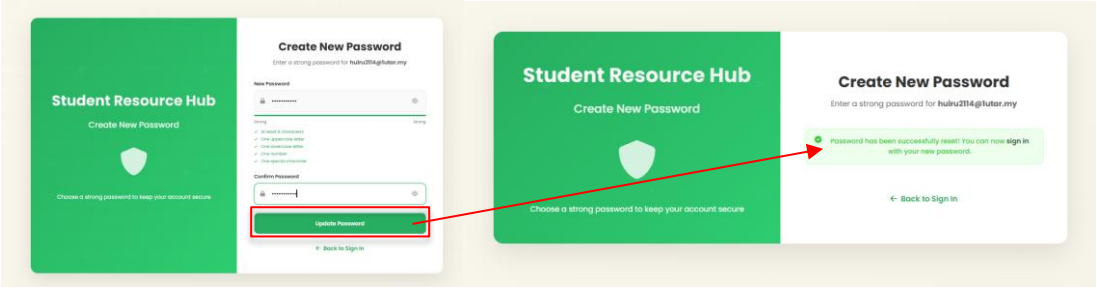


Figure 5.5.55 Password Reset Form and Reset Password Success Message

5.5 Implementation Issues and Challenges

During the implementation of this platform, there are several challenges encountered:

1. API Limitations

   One of the serious challenges during the implementation of this project was the limitation of the Gemini API key. This is because the platform developed relies heavily on Gemini API key to perform key tasks. For example, the AI Course Classification, OCR Tool Gemini AI Assistant, smart calendar planner, AI Study Tools for generating flashcards, quizzes and the AI Chatbot are required to use Gemini API key. Since all these features depend on the same API key, the usage limit was reached very quickly. Once the quota was exceeded, the system will be unable to function properly, which greatly affected testing and development progress.

   To address this challenge, one possible solution is to purchase a higher-tier Gemini API plan with increased usage limits. This would ensure that all features can operate smoothly without interruption and allow the system to handle more user requests efficiently.

2. Sentiment Analysis Misclassification

   The commenting system was developed to analyze the sentiment of users' comments and detect hate speech. However, the sentiment analysis model sometimes misclassified neutral comments as positive or negative, which affected the result. To address this issue, improvement needs to be implemented by fine-tuning with a large diverse dataset. This improvement would help ensure more reliable results when detecting the sentiment of users' comments.

5.6 Concluding Remark

In conclusion, the system implementation phase successfully integrated the required hardware, software, libraries, and algorithms to support the proposed features. Each module, including login and registration, content upload, commenting system, image-to-text extraction, video summarization, YouTube summarizer, calendar planner, and AI study tools, has been implemented and tested to ensure proper functionality. The system is now fully operational, allowing users to register, upload, analyze, and manage study materials effectively. This implementation provides the foundation for further testing and evaluation in the next chapter.

Chapter 6

System Evaluation And Discussion

6. 1 System Testing and Performance Metrics

To ensure that all system functionalities work correctly, system testing is important and cannot be omitted. In this chapter, unit testing applied to evaluate the system. This testing method ensures that all the system functionalities work as specified in the requirements.

6.1.1 Unit Testing

Unit testing is performed on all modules of the system. This is to ensure that each functionality works correctly. This testing focuses on the internal logic of each module and verifying the correct processing of valid inputs as well as the proper handling of invalid or unexpected inputs. The modules included in this unit testing are:

- Login Modules Unit Testing
- Registration Modules Unit Testing
- Library Module Unit Testing
- Upload Material Module Unit Testing
- Optical Character Recognition (OCR) Module Unit Testing
- Commenting Module Unit Testing
- Uploaded Video Content Summarization Module Unit Testing
- YouTube Video Content Summarization Module Unit Testing
- Smart Calendar Planner Module Unit Testing
- AI Study Tools Module Unit Testing
- Forgot Password Module Unit Testing

In unit testing, each module is tested individually to ensure their correctness, reliability, and proper handling of different input scenarios. Unit testing helps to verify that the system's core functionalities are working properly before integration and deployment.

6.1.1.1 Login Module Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Login	Successful Login	Users enter correct email address and password.	Directed to Homepage	Directed to Homepage	Pass
	Login with Unregistered Email	Users enter email not registered.	Display error message: User Not Found!	Display error message: User Not Found!	Pass
	Login with wrong password	Users enter wrong password.	Display error message: “Invalid password!”	Display error message: “Invalid password!”	Pass
	Login with empty fields	Leave email and password empty	Display error message: “Email and Password is required”	Display error message: “Email and Password is required”	Pass

Table 6.1.1 Login Module Unit Testing

6.1.1.2 Registration Module Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Registration	Successful Registration	User enters valid username, email and password.	Account created successfully; display message: "Registration successful"	Account created successfully; display message: "Registration successful"	Pass
	Registration with Existing Email	User enters an email that is already registered	Display error message: "Email already registered"	Display error message: "Email already registered"	Pass
	Registration with Invalid Email	User enters an invalid email format	Display error message: "Please enter a valid email address"	Display error message: "Please enter a valid email address"	Pass
	Registration with Empty Fields	User leaves required fields empty	Display error message: "All fields are required"	Display error message: "All fields are required"	Pass
	Registration with weak password	User enters weak password	Display error message: "Password must meet all requirements below"	Display error message: "Password must meet all requirements below"	Pass
	Registration with Password Not Match	User enters a password and a different confirm password	Display error message: "Passwords do not match"	Display error message: "Passwords do not match"	Pass

Table 6.1.2 Registration Module Unit Testing

6.1.1.3 Library Module Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Library	Upload Material	Users click "Upload" button.	Directed to Upload Material Page	Directed to Upload Material Page	Pass
	View Material Content	Users click "View" button.	Directed to View Content Page	Directed to View Content Page	Pass
	Search Material with Valid Keyword	User enters a valid keyword in the search bar	Display search results matching the keyword	Display search results matching the keyword	Pass
	Search Material with Invalid Keyword	User enters a keyword that does not match any material	Display message: "No materials found"	Display message: "No materials found"	Pass
	Filter Material by Category	User selects a file category from the filter	Display materials of the selected category	Display materials that of the selected category	Pass
	Filter Material by Upload Date	User selects a year or month for filtering materials	Display materials of the selected year or month	Display materials of the selected year or month	Pass
	Filter Material by Courses	User selects courses for filtering materials	Display materials of the selected courses	Display materials of the selected courses	Pass

Table 6.1.3 Library Module Unit Testing

6.1.1.4 Upload Material Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Upload Material	Open Upload Form	Users click "Upload" button.	Directed to Upload Material Page	Directed to Upload Material Page	Pass
	Upload File	Users upload material file.	AI Course Classification classifies the material	AI Course Classification classifies the material	Pass
	Change AI Classification Course	Users click "Select Course" dropdown	Display Select Course dropdown	Display Select Course dropdown	Pass
	Upload Material in Public	Users submit upload forms with public privacy setting.	Uploaded material displays on both Library page and user Profile Page	Uploaded material displays on both Library page and user Profile Page	Pass
	Upload Material in Private	Users submit upload forms with private privacy setting.	Uploaded material displays only on user Profile Page	Uploaded material displays only on user Profile Page	Pass
	Empty Upload Form	User does not fill in all the required fields	"Upload Material" button cannot be clicked	"Upload Material" button cannot be clicked	Pass
	AI Course Classification Service Unavailable	User uploads a material file but the AI Classification Service Unavailable	Display error message: "Classification service unavailable" and enable Select Course Dropdown	Display error message: "Classification service unavailable" and enable Select Course Dropdown	Pass

Table 6.1.4 Upload Material Unit Testing

6.1.1.5 Optical Character Recognition (OCR) Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Optical Character Recognition (OCR)	Open OCR Scan Text Page	Users click “Scan Text” button.	Directed to OCR Scan Text Page	Directed to OCR Scan Text Page	Pass
	Upload Image for OCR	Users click “Choose Image” button and upload an image file.	OCR scans the image and displays the extracted text on the page	OCR scans the image and displays the extracted text on the page	Pass
	Upload Image with No Text	User uploads an image file that contains no text	Display message “No text could be extracted from the image.”	Display message “No text could be extracted from the image.”	Pass
	Edit Extracted Text	Users click “Edit” button.	Display field that allows users to edit extracted text.	Display field that allows users to edit extracted text.	Pass
	Upload Image and Extracted Text to Library	Users click “Upload Material” button.	Display upload form with image preview and extracted text preview.	Display upload form with image preview and extracted text preview.	Pass
	Copy Extracted Text	Users click ”Copy Text” button	Display success message “Text copied to clipboard!”	Display success message “Text copied to clipboard!”	Pass
	Download Extracted Text	Users click “Download” button	Extracted text download as text file.	Extracted text download as text file.	Pass

	Gemini AI Assistant	Users click on “Gemini AI Assistant” button	Gemini AI Assistant chatbot pop up.	Gemini AI Assistant chatbot pop up.	Pass
	Ask Gemini Question	Users input questions to Gemini AI Chatbot	Answers provided and display to users	Answers provided and display to users	Pass
	View Recent Conversation with AI Chabot	Users click on specific conversation tab in “Recent Conversation” right sidebar	Gemini AI Chatbot pops up with previous conversation displayed.	Gemini AI Chatbot pops up with previous conversation displayed.	Pass
	View Uploaded Image Preview	Users click on uploaded image.	Display Uploaded Image Preview.	Display Uploaded Image Preview.	Pass

Table 6.1.5 Optical Character Recognition (OCR) Unit Testing

6.1.1.6 Uploaded Video Content Summarization Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Uploaded Video Content Summarization	Upload Video	Users click "Browse Files" button and upload a video file.	Display file preview in upload form.	Display file preview in upload form.	Pass
	Summarize video	Users click on "Generate Summary" button	Display video, generated summary and transcript	Display video, generated summary and transcript	Pass
	View Full Transcript	Users click on "Full Transcript" Tab.	Display full video transcript.	Display full video transcript.	Pass
	Upload to Library	Users click "Save to Study Resources" button.	Display upload form with video preview, generated summary and transcript.	Display upload form with video preview, generated summary and transcript.	Pass
	Download summary and transcript	Users click "Download Transcription" button	Generated summary and transcript download as text file.	Generated summary and transcript download as text file.	Pass
	Upload Unsupported File Format	Users upload a file with an unsupported format	Display error message: "Invalid file type. Please upload MP4, AVI, or MOV files."	Display error message: "Invalid file type. Please upload MP4, AVI, or MOV files."	Pass

Table 6.1.6 Uploaded Video Content Summarization Unit Testing

6.1.1.7 YouTube Video Content Summarization Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
YouTube Video Content Summarization	Summarize YouTube video	Users enter a valid YouTube URL link	Display YouTube video, generated summary and transcript	Display YouTube video, generated summary and transcript	Pass
	Enter invalid YouTube URL Link	Users enter invalid YouTube URL Link.	Display error message: "Unable to Generate Summary".	Display error message: "Unable to Generate Summary".	Pass
	Enter YouTube URL Link with no transcript provided	Users enter YouTube URL Link with no transcript available.	Display error message: "Unable to Generate Summary".	Display error message: "Unable to Generate Summary".	Pass
	Upload to Library	Users click "Save to My Study Resources" button.	Display upload form with video preview, generated summary and transcript.	Display upload form with video preview, generated summary and transcript.	Pass
	Play video with specific timestamp	Users click "Play" button at a specific key timestamp.	Play YouTube video at the specific timestamp clicked.	Play YouTube video at the specific timestamp clicked.	Pass
	Video Transcription for specific timestamp	Users click "View" button for transcription at a specific timestamp.	Display full transcription for the specific timestamp.	Display full transcription for the specific timestamp.	Pass

Table 6.1.7 YouTube Video Content Summarization Unit Testing

6.1.1.8 Commenting System Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Commenting System	Post positive comment	Users enter positive comments without hate speech and click "Post Comment" button.	Comment posted with positive sentiment indicator.	Comment posted with positive sentiment indicator	Pass
	Post negative comment	Users input negative comments without any hate speech and click "Post Comment" button.	Comment posted with negative sentiment indicator.	Comment posted with negative sentiment indicator.	Pass
	Hate Speech comment	Users input negative comments with hate speech and click "Post Comment" button	Display error message and failed to post comment.	Display error message and failed to post comment.	Pass
	Like comment	Users click "Like" button.	Like button toggle and update like count.	Like button toggle and update like count.	Pass
	Edit comment	Users click "Edit" button to modify comment input and "Save" button.	Display edits comment field and allow input comments.	Display edits comment field and allow input comments.	Pass
		Users modify comment input and "Save" button.	Display success message "Comment updated successfully".	Display success message "Comment updated successfully".	Pass

	Delete comment	Users click “Delete” button in comment interface.	Display deletes comment confirmation message.	Display deletes comment confirmation message.	Pass
		Users click “Delete” button in confirmation popup.	Display success message: “Comment deleted successfully!”	Display success message: “Comment deleted successfully!”	Pass

Table 6.1.8 Commenting System Unit Testing

6.1.1.9 Smart Calendar Planner Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Smart Calendar Planner	Calendar Planner interface	Users click on "Calendar Planner" button.	Directed to Calendar Planner Interface.	Directed to Calendar Planner Interface.	Pass
	Add AI Schedule Task	Users click on "+New Task" button.	Display Create New Task Form.	Display Create New Task Form.	Pass
		Users input all required information into Create New Task form and click "Create & Schedule" button.	Display success message and AI Schedule Task added to calendar view.	Display success message and AI Schedule Task added to calendar view.	Pass
		Users leave empty Create New Task form and click "Create & Schedule" button.	Display error message "Please fill out this field."	Display error message "Please fill out this field."	Pass
	Add Direct Task	Users click specific timeslot column in calendar view.	Display Create Direct Task Form.	Display Create Direct Task Form.	Pass
		Users input all required field in Create Direct Task form and click "+Create Task" button.	Display success message "Task created successfully" and created task display in calendar view.	Display success message "Task created successfully" and created task display in calendar view.	Pass

		Users leave empty Create Direct Task form and click "+Create Task" button.	Display error message "Please fill out this field."	Display error message "Please fill out this field."	Pass
	Import Timetable	Users click "Import" button	Display import timetable upload form.	Display import timetable upload form.	Pass
		Users upload timetable to the import timetable form.	Display uploaded file preview.	Display uploaded file preview.	Pass
		Users click "Process with AI" button.	Display detected schedule.	Display detected schedule.	Pass
		Users click "Import Schedule" button.	Display success message "Imported Successful!" and imported timetable added to calendar view.	Display success message "Imported Successful!" and imported timetable added to calendar view.	Pass
	Add Direct Task with Timeslot Conflict Detected	Users fill in Create Direct Task form with overlapping tasks.	Display error message "Time Conflict Detected" and suggest alternative timeslot.	Display error message "Time Conflict Detected" and suggest alternative timeslot.	Pass
	Import Timetable with Timeslot Conflict Detected.	Users import timetables with overlapping task.	Display error message and highlight overlapping timeslot.	Display error message and highlight overlapping timeslot.	Pass

Table 6.1.9 Smart Calendar Planner Unit Testing

6.1.1.10 AI Study Tools Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
AI Study Tools	AI Study Tools Interface	User clicks "Study with AI" button	Directed to AI Study Tools Page on Flashcards tab	Directed to AI Study Tools Page on Flashcards tab	Pass
	View Flashcard Answer	User clicks on a flashcard	Display flashcard answer.	Display flashcard answer.	Pass
	Next Flashcard	User clicks "Next" button	Display next flashcard question	Display next flashcard question	Pass
	View All Flashcards	User clicks "View All Cards" button	Display all flashcards	Display all flashcards	Pass
	AI Study Tools: Quiz	Users click on "Quiz Tab	Directed to AI Study Tools Page on Quiz tab.	Directed to AI Study Tools Page on Quiz tab	Pass
	Generate Quiz	User selects quiz settings and clicks "Generate New Quiz"	Display generated quiz	Display generated quiz	Pass
	Answer Quiz Question	User answers a quiz question	Display question, user's answer, and explanation	Display question, user's answer, and explanation	Pass
	View Quiz Result	User completes all quiz questions	Display quiz results with correct and wrong answers	Display quiz results with correct and wrong answers	Pass

	Review Quiz Answers	User clicks “Review Answers” button	Display answered quiz questions with explanations	Display answered quiz questions with explanations	Pass
	Take Another Quiz	User clicks “Take Another Quiz” button	Generate new quiz	Generate new quiz	Pass
	View Quiz History	User clicks “View History” button	Display previously answered quiz questions and answers	Display previously answered quiz questions and answers	Pass
	Ask AI Assistant	User enters question in AI Assistant tab	Display AI response	Display AI response	Pass

Table 6.1.10 AI Study Tools Unit Testing

6.1.1.11 Forgot Password Unit Testing

Use Case	Test Case	Input	Expected Output	Actual Output	Pass/Fail
Forgot Password	Forgot Password Page	User clicks "Forgot Password?" link on Login Page	Directed to Forgot Password Page	Directed to Forgot Password Page	Pass
	Send Password Reset Link	User enters registered email address and click "Send Reset Link" button.	Display success message and password reset link sent to users' email	Display success message and password reset link sent to users' email	Pass
	Click Reset Link in Email	User clicks "Reset Your Password" button in received email	Directed to Create New Password Page	Directed to Create New Password Page	Pass
	Reset Password Successfully	User enters new valid password and clicks "Update Password"	Display success message: "Password updated successfully"	Display success message: "Password updated successfully"	Pass
	Reset Password with Empty Fields	User leaves password field empty and clicks "Update Password"	Display error message: "Please enter a valid password"	Display error message: "Please enter a valid password"	Pass
	Reset Password with Mismatched Confirm Password	Users enter new password and different confirm password	Display error message: "Passwords do not match".	Display error message: "Passwords do not match".	Pass

	Reset Password with Weak Password	User enters a weak password and clicks “Update Password”	Display error message: “Please enter a valid password”	Display error message: “Please enter a valid password”	Pass

Table 6.1.11 Forgot Password Unit Testing

6.2 Project Challenges

Throughout the development of this project, several challenges were encountered. The challenges include:

- OCR Scan Text Tool Accuracy

```
C:\Users\huils2114\Desktop\ocr_testing>python evaluate_ocr.py ocr_script.py test_images ground_truth ocr_result
Processing advertisement1.jpg...
Character Accuracy: 68.66%
Word Accuracy: 94.69%
Processing Time: 2.09 seconds
--------------------------------------------------
Processing handwriting.jpg...
Character Accuracy: 53.09%
Word Accuracy: 87.50%
Processing Time: 0.80 seconds
--------------------------------------------------
Processing penguin1.jpg...
Character Accuracy: 82.40%
Word Accuracy: 96.00%
Processing Time: 1.80 seconds
--------------------------------------------------
Processing princess1.png...
Character Accuracy: 98.47%
Word Accuracy: 100.00%
Processing Time: 1.42 seconds
--------------------------------------------------
Processing test1.jpg...
Character Accuracy: 64.24%
Word Accuracy: 96.46%
Processing Time: 1.44 seconds
--------------------------------------------------
Processing test2.png...
Character Accuracy: 97.44%
Word Accuracy: 94.74%
Processing Time: 1.13 seconds
--------------------------------------------------
Processing test3.jpg...
Character Accuracy: 74.40%
Word Accuracy: 89.52%
Processing Time: 5.16 seconds
--------------------------------------------------
Processing test4.png...
Character Accuracy: 60.04%
Word Accuracy: 100.00%
Processing Time: 0.96 seconds
--------------------------------------------------

===== SUMMARY =====
Total Images Processed: 8
Average Character Accuracy: 74.84%
Average Word Accuracy: 94.86%
Average Processing Time: 1.85 seconds
Lowest Character Accuracy: 53.09%
Lowest Word Accuracy: 87.50%
Maximum Processing Time: 5.16 seconds

Detailed results saved to ocr_result\accuracy_results.csv
```

Figure 6.3.1 OCR Accuracy Result

Although the OCR Tool was successfully developed in this project, its accuracy is not as high as some existing OCR platforms. The tool was tested and achieved an average character accuracy of 74.84% and an average word accuracy of 94.86%. While these results are acceptable in some cases, however, the accuracy may not be sufficient when extracting for small text or text in difficult fonts, low quality images, or images with noise. The accuracy problems make the OCR tool less reliable and facing difficulties extracting images text in more accurate way. Therefore, improvement is needed, such as using better image processing methods.

- YouTube Video Summarizer

The second challenge faced during the development process was that the YouTube Video Summarizer relied on the availability of transcripts provided by the video uploaded. If the video uploader did not provide a video transcript, then users who use this YouTube Video Summarizer Tool could not generate the video transcription and summary because the system depends on the YouTube Transcript API to extract the captions. This is considered a challenge because many YouTube videos do not provide transcripts, which reduces the usefulness of this summarizer.

- Commenting System

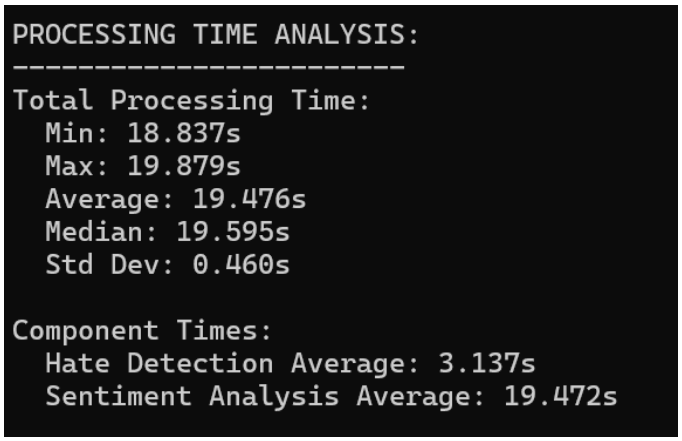


Figure 6.3.2 Commenting System Response Time

The third challenge during the implementation of this project is that the response time for posting a comment is quite long. Figure 6.3.2 shows that the minimum processing time for posting a comment is 18.837s and maximum time is 19.869s. The average time of posting a comment is 19.476s. This happens because the commenting system is developed to automatically detect hate speech and analyze the sentiment of comment. Therefore, when a comment is posted, the system needs to perform two separate tasks: sentiment analysis and hate speech detection. This causes the response time for posting comments to be longer.

6.3 Objectives Evaluation

The objectives of this project were evaluated as follows:

1. Content Extraction and Summarization

This objective was successfully achieved. OCR was implemented to extract text from images, Whisper was used to transcribe videos, and Google Gemini generated summaries. These components allowed multimedia educational materials to be managed more efficiently.

2. Automated Content Classification

The project met this objective by integrating Google Gemini API with structured prompt engineering. The system was able to analyze and classify content from different file formats into relevant course categories.

3. Collaboration and Personalized Learning Support Features

This objective was achieved. The commenting system with sentiment analysis and hate speech detection was implemented, ensuring safe and moderated discussions. A calendar planner was integrated to support study schedule management. Additionally, the AI Study Tools such as flashcards, quizzes, and AI chatbot were developed to help personalized learning.

In conclusion, the project successfully addressed its main objectives, though certain challenges such as computational requirements and content accuracy highlight opportunities for future improvements.

6.4 Concluding Remark

In conclusion, this chapter outlines the system testing of each module using unit testing, the challenges encountered during the development, and the evaluation of the objectives. The results of unit testing showed that all modules functioned correctly and met all the requirements specified. Although several challenges were faced, such as the limitation of accuracy of the OCR tool, reliance on YouTube transcripts for generating YouTube Video summaries, and the slow response time when posting comments, The overall system successfully achieved its main objectives. The implemented features include an OCR Tool for extracting text from images, automated course classification, video summarization, a commenting system, a built-in calendar planner, and personalized learning support tools such as AI Study Tools to generate flashcards and quizzes. These outcomes show that the project has successfully achieved the intended objectives.

Chapter 7
Conclusion

7.1 Conclusion

The Student Resource Exchange is an online platform developed for students which aims to make sharing educational resources easier and more useful. The purpose of implementing this platform is because many existing platforms were inconvenient to use. For example, the existing platforms are not able to extract text from image, the resources in existing platforms were poorly categorized, and lack of interactive features such as commenting system. These issues frustrated many users who use the platforms.

To address the issues, the platform integrated some technologies. For example, OCR can extract text from images; Whisper automatically converts audio from videos into text; and Google Gemini summarizes video content and automatically categorizes resources, saving significant time. Moreover, sentiment analysis and hate speech detection were also implemented to have a safe and constructive user interaction in the commenting system. In addition, the platform implemented AI-powered study tools, such as AI generated flashcards, quizzes and AI chatbot for personalized learning support. A Smart Calendar Planner also implemented to help students organize study schedules and deadlines more effectively.

The innovation of the project is that it combines different technologies into one platform. By automating important tasks, such as AI course classification task and AI scheduling task, it can make the flows of using the platform become more efficient. In conclusion, the platform solves the main problems found in the existing systems and successfully achieves its objectives which include integrating OCR technology, enabling user comments, automating resource classification, providing a smart calendar planner, and offering a comprehensive set of AI learning tools.

7.2 Recommendations

The recommendations for future work include improving the system with better OCR technology that can provide better accuracy for text extraction from images. By using advanced OCR engines like EasyOCR, the current 74%-character accuracy could be increased. The commenting system should also be improved with faster response time, as the current 19-second wait time is too slow for users. By adding backup speech recognition using Whisper API could help the YouTube video summarizer work even when videos do not have transcripts available. The system could also use multiple AI service providers instead of relying on Gemini API, which could prevent service interruptions when usage limits are reached. Overall, while the project has successfully met its objectives, these improvements would enhance the system's reliability and user experience for future development.

REFERENCES

[1] UNESCO, “UNESCO COVID-19 Education Response: How Many Students Are at Risk of Not Returning to School? Advocacy Paper,” in *UNESCO*, Jul. 2020. [Online]. Available: UNESCO COVID-19 education response: how many students are at risk of not returning to school? Advocacy paper - UNESCO Digital Library.

[2] S. Y. Wong, “Digitalization challenges in education during COVID-19: A systematic review,” in *Cogent Education*, vol. 10, no. 1, Jan. 2023. [Online]. Available: https://doi.org/10.1080/2331186X.2023.2198981.

[3] H. Anthony, “How Optical Character Recognition (OCR) reduces 90 percent of your document indexing workload?,” *GLOBODOX*, May 25, 2022. https://www.globodox.com/blog/how-optical-character-recognition-ocr-reduces-90-percent-of-your-document-indexing-workload/

[4] W. K. T. M. Gunarathne, T. K. Shih, Chalothon Chootong, Worapot Sommool, and Ankhtuya Ochirbat, “An Automated Learning Content Classification Model for Open Education Repositories: Case of MERLOT II,” *Journal of Internet Technology*, vol. 21, no. 5, pp. 1277–1288, Sep. 2020, Accessed: Aug. 25, 2024. [Online]. Available: https://jit.ndhu.edu.tw/article/view/2365

[5] Idaho State University, “What is OER?,” *Isu.edu*, 2019. [Online]. Available: http://isu.edu/itrc/open-educational-resources/what-is-oer/

[6] studocu, “StuDocu,” *StuDocu*, 2016. https://www.studocu.com/

[7] “CourseHero,” *Coursehero.com*, 2019. https://www.coursehero.com/

[8] Scribd, “Scribd - Read books, audiobooks, and more,” *Scribd*, 2008. https://www.scribd.com/

[9] “Notes, essays and summaries for university and high school students - Docsity,” *Docsity.com*, 2019. https://www.docsity.com/

[10] “OneClass - Better Grades Now,” *OneClass*, 2010. https://oneclass.com/

[11] “What is OCR (Optical Character Recognition)? | IBM,” *www.ibm.com*, Apr. 22, 2024. https://www.ibm.com/topics/optical-character-recognition

[12] B. Council, “Natural Language Processing (NLP) - Blockchain Council,” *Blockchain-council.org*, Jun. 15, 2023. https://blockchain-council.org/ai/natural-language-processing-nlp/?gad_source=1&gclid=Cj0KCQjw_sq2BhCUARIsAIVqmQtYrbNQRB8XXnjPmWOljLzk-32N_9cqsm5otrB0ZSw6yb5p7rOYPJsaAgPeEALw_wcB

[13] “Convolutional Neural Networks (CNN) and Deep Learning,” *Intel*, 2020. https://www.intel.com/content/www/us/en/internet-of-things/computer-vision/convolutional-neural-networks.html

[14] IBM, “What are Recurrent Neural Networks? | IBM,” *www.ibm.com*, 2023. https://www.ibm.com/topics/recurrent-neural-networks

[15] A. Chugh, “Deep Learning | Introduction to Long Short Term Memory,” *GeeksforGeeks*, Jan. 16, 2019. https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

[16] M. Banoula, “Introduction to Long Short-Term Memory(LSTM) | Simplilearn,” *Simplilearn.com*, Apr. 27, 2023. https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/lstm

[17] PubNub, “What are WebSockets?,” *PubNub*. https://www.pubnub.com/guides/websockets/

[18] N. Barney and S. G. Alexander , “What is sentiment analysis?,” *TechTarget*, Mar. 2023. https://www.techtarget.com/searchbusinessanalytics/definition/opinion-mining-sentiment-mining

[19] “High-Level Language - Glossary.” *DevX*, Dec. 2023, www.devx.com/terms/high-level-language/.

[20] Dominykas Jasiulionis, “What is VS Code: an overview of the popular code editor and its features,” *Hostinger Tutorials*, Dec. 20, 2024. https://www.hostinger.com/tutorials/what-is-vs-code

[21] Apache Friends, “XAMPP Installers and Downloads for Apache Friends,” *www.apachefriends.org*, 2022. https://www.apachefriends.org/

[22] GeeksforGeeks, “Using Python Environment Variables with Python Dotenv,” *GeeksforGeeks*, Mar. 19, 2024. https://www.geeksforgeeks.org/using-python-environment-variables-with-python-dotenv/

[23] F. Zelic, “Python OCR Tutorial: Tesseract, Pytesseract, and OpenCV,” *Nanonets*, Feb. 27, 2023. https://nanonets.com/blog/ocr-with-tesseract/

[24] S. Gruppetta, “Image Processing With the Python Pillow Library,” *Real Python*, Jan. 08, 2025. https://realpython.com/image-processing-with-the-python-pillow-library/

[25] J. Depoix, “youtube-transcript-api 1.0.3,” *PyPI*, Mar. 26, 2025. https://pypi.org/project/youtube-transcript-api/

[26] OpenAI, “Introducing Whisper,” *OpenAI*, Sep. 21, 2022. https://openai.com/index/whisper/

[27] GeeksforGeeks, “Introduction to Web development using Flask,” *GeeksforGeeks*, Oct. 16, 2018. https://www.geeksforgeeks.org/python/python-introduction-to-web-development-using-flask/

[28] C. Dolphin, “Flask-Cors: A Flask extension adding a decorator for CORS support,” *PyPI*, Jun. 11, 2025. https://pypi.org/project/Flask-Cors/

[29] Pydantic, “Pydantic,” *docs.pydantic.dev*. https://docs.pydantic.dev/latest/

[30] GeeksforGeeks, “Introduction to Python PyPDF2 Library,” *GeeksforGeeks*, Sep. 11, 2024. https://www.geeksforgeeks.org/python/introduction-to-python-pypdf2-library/ (accessed Sep. 17, 2025).

[31] S. Canny, “python-docx: Create, read, and update Microsoft Word .docx files.,” *PyPI*, Jun. 17, 2025. https://pypi.org/project/python-docx/

[32] GeeksforGeeks, “Creating and updating PowerPoint Presentations in Python using python pptx,” *GeeksforGeeks*, Aug. 15, 2020. https://www.geeksforgeeks.org/python/creating-and-updating-powerpoint-presentations-in-python-using-python-pptx/ (accessed Sep. 17, 2025).

[33] PyPi, “pandas,” *PyPI*, Jan. 30, 2020. https://pypi.org/project/pandas/

[34] Numpy, “NumPy,” *Numpy.org*, 2024. https://numpy.org/

[35] scikit-learn, “scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation,” *Scikit-learn.org*, 2019. https://scikit-learn.org/

[36] NLTK, “Natural Language Toolkit — NLTK 3.4.4 documentation,” *Nltk.org*, 2009. https://www.nltk.org/

[37] “jigsaw-toxic-comment-classification-challenge,” *www.kaggle.com*. https://www.kaggle.com/datasets/julian3833/jigsaw-toxic-comment-classification-challenge

POSTER

Developer: Saw Hui Loo
Supervisor: Dr. Jasmina Khaw Yen Min

STUDENT RESOURCE EXCHANGE SYSTEM

01. Introduction

The Student Resource Exchange System is a web-based platform for students to share educational resources with OCR text extraction, video summarization, automated course classification, built-in Smart Calendar Planner and AI study tools.

02. Methods

- Tesseract OCR with LSTM Neural Networks: Text Extraction from Images
- Image Preprocessing Algorithms: OCR Quality Enhancement
- Whisper ASR with Large Language Model: Video Transcription and Summarization
- LLM with Prompt Engineering: Automated Course Classification
- TF-IDF with Logistic Regression: Hate Speech Detection
- BERT-based Sentiment Analysis: Comment Sentiment Classification
- Conflict Detection Algorithm: Smart Calendar Planner
- Priority-Based Greedy Scheduling Algorithm: AI Task Scheduling
- Natural Language Processing (NLP): Content Analysis for Study Tools
- Fuzzy String-Matching Algorithm: Answer Validation for Quizzes

03. Results

- System Implementation: 11 functional modules completed
- Platform Features: Successfully integrated OCR, video summarization, AI Course Classification, commenting system, calendar planner, and AI Sudy Tools

04. Discussion

1. Strengths: Successfully developed 11 functional modules including OCR text extraction, video summarization, automated course classification, commenting system with sentiment analysis, smart calendar planner and AI study tools.
2. Limitations: OCR accuracy 74.84% insufficient for complex documents. Comment response time 19+ seconds impacts user experience.

05. Conclusion

Multiple algorithms including OCR, Whisper, BERT, TF-IDF, and conflict detection are applied to create a comprehensive educational platform with text extraction, video summarization, automated course classification, smart calendar planner, and AI study tools.