

COMPUTING JOBS MONITORING DASHBOARD IN MALAYSIA
BY
TAN ZHEN WEI

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 2022

REPORT STATUS DECLARATION FORM

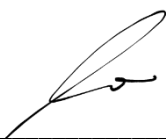
Title: COMPUTING JOBS MONITORING DASHBOARD
IN MALAYSIA

Academic Session: JUNE 2022

I TAN ZHEN WEI
(CAPITAL LETTER)

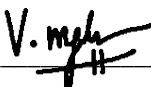
declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



Tan Zhen Wei

Verified by,



(Supervisor's signature)

Address:

E1-10.JALAN PP16
PANGSAPURI PUTRA INDAH SEK 2
43300 SERI KEMBANGAN, SELANGOR

Dr Mogana a/p Vadiveloo

Supervisor's name

Date: 09/09/2022

Date: 09/09/2022

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

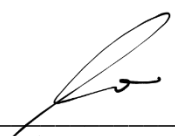
Date: 09/09/2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Tan Zhen Wei (ID No: 19ACB06234) has completed this final year project/ dissertation/ thesis* entitled Computing Jobs Monitoring Dashboard In Malaysia under the supervision of Dr. Mogana a/p Vadiveloo (Supervisor) from the Department of Computer Science , Faculty/Institute* of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,



 (Tan Zhen Wei)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**COMPUTING JOBS MONITORING DASHBOARD IN MALAYSIA**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :  _____

Name : Tan Zhen Wei

Date : 09/09/2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr Mogana a/p Vadiveloo for all the guidance that she has provided me for the project. It has allowed me to be exposed to the technologies which are related to web scraping and to develop an interactive dashboard. Lastly, I am deeply grateful to my parents, whose endless support and encouragement enabled me to persevere to complete this project.

ABSTRACT

This project proposed computing jobs monitoring dashboard in Malaysia and the dashboard will analyze and visualize the scraped data to help job seekers to better understand the current job market in the IT industry. The main motivation to propose this project is that there is vast amount of data available in online job recruitment platform but however, no tools or software are available to analyze that data into meaningful representation to job seekers. This project will focus on scraping data about computing jobs, this is because the IT industry changes and grows rapidly year by year, yet there is no data analysis and statistics about the related industry in Malaysia. Therefore, in this work, a computing jobs monitoring dashboard is proposed to solve the aforementioned issues. The proposed dashboard is able to automatically extract relevant data from online job recruitment platform such as JobStreet and Indeed, analyze the extracted data and visualize them in an interactive manner. The scraped data includes job title, company, location, salary, job requirements, qualifications, years of relevant job experience and application link. Apart from that, the Logistic Regression was used to classify the jobs into different computing jobs categories and a custom Named Entity Recognition (NER) model was built to extract the Information and Communication Technology (ICT) skills from each job requirements. The dashboard displays useful information for job seekers, including popular programming languages and skills, distribution of job opportunities, etc. The proposed dashboard is an interactive dashboard that provide users with several filtering options to view relevant data and information based on certain filtering criteria. In this work, BeautifulSoup has been used to program web scraping scripts and WayScript is used as the main development platform to automate the data scraping and storing them in Azure Blob Storage. In addition to that, the front end of this project is a highly interactive dashboard is developed using Plotly's Dash framework.

Table of Contents

TITLE PAGE	I
REPORT STATUS DECLARATION FORM	II
FYP THESIS SUBMISSION FORM.....	III
DECLARATION OF ORIGINALITY	IV
ACKNOWLEDGEMENTS	V
ABSTRACT.....	VI
LIST OF FIGURES	XI
LIST OF TABLES	XIII
LIST OF ABBREVIATIONS	XIV
CHAPTER 1: INTRODUCTION.....	1
1.1 Background Information.....	1
1.2 Problem Statement.....	1
1.3 Motivation.....	2
1.4 Project Objectives	3
1.5 Project Scope and Direction.....	4
1.6 Contributions.....	5
1.7 Report Organization.....	6
CHAPTER 2: LITERATURE REVIEW	7
2.0 Introduction.....	7
2.1 Online Job Recruitment Platform in Malaysia.....	7

2.1.1 Jobstreet.com	7
2.1.2 Indeed.com.....	8
2.2 Web Scraping and Data Analysis	10
2.2.1 Python Library	10
2.2.2 Web Data Extractor.....	14
2.3 Dashboard	16
2.3.1 D3.js	16
2.3.2 Tableau.....	17
2.3.3 Plotly's Dash.....	18
2.4 Limitation of The Reviewed Tools	20
2.4.1 Web Scraping and Data Analysis	20
2.4.2 Dashboard	21
2.5 Critical Remark.....	22
CHAPTER 3: SYSTEM DESIGN	25
3.1 Overview.....	25
3.2 System Architecture.....	25
3.3 Use Case Diagram.....	27
3.4 Use Case Description.....	28
3.5 Activity Diagram	34
CHAPTER 4: METHODOLOGY AND TOOLS	36
4.1 System Methodology	36
4.2 System Requirement	37
4.2.1 Hardware Requirements.....	37
4.2.2 Software Requirements	38
4.3 User Requirements.....	39
4.4 Non-functional Requirements.....	40

4.5 Verification Plan	40
4.5.1 Filtering Data	40
4.5.2 Filtering and Sorting Job Postings in Data Table	41
4.5.3 Regenerate a New Chart when Further Filtering is Performed.....	42
4.6 Implementation Issues and Challenges	43
4.7 Timeline	45
4.7.1 – Timeline of the FYP1.....	45
4.7.2 – Timeline of the FYP2.....	46
CHAPTER 5: SYSTEM IMPLEMENTATION	48
5.1 WayScript	48
5.2 Azure Blob Storage.....	52
5.3 Web Scraping	55
5.4 Data Cleaning.....	56
5.5 Data Analysis	58
5.5.1 Multiclass Classification.....	58
5.5.2 Named-Entity Recognition (NER).....	66
5.6 Dashboard	70
5.6.1 Overview of Dashboard	70
5.6.2 Data Summary in the Dashboard	73
5.6.3 Data Visualization in the Dashboard	74
5.6.4 Filtering in the Dashboard	81
CHAPTER 6: CONCLSION	84
6.1 Project Review	84
6.2 Novelties	85
6.3 Future Work.....	86
REFERENCES.....	87

FINAL YEAR PROJECT WEEKLY REPORT	91
POSTER.....	97
PLAGIARISM CHECK RESULT.....	98
FYP2 CHECKLIST.....	105

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	Computer science and IT job postings in JobStreet	4
Figure 2.1	Overview of Jobstreet.com	8
Figure 2.2	Overview of Indeed.com	9
Figure 2.3	Overview of the process of web scraping and data analysis using Python	10
Figure 2.4	Scraping data from Shopee using BeautifulSoup and request	13
Figure 2.5	The scraped data is embedded in different HTML tags	13
Figure 2.6	Creating a session in Web Data Extractor	15
Figure 2.7	Web scraping performance of Web Data Extractor	15
Figure 2.8	Options provided by Tableau for the import format and the database selection	17
Figure 2.9	Overview of Tableau	18
Figure 3.1	System architecture of the proposed computing jobs monitoring dashboard	25
Figure 3.2	Use case diagram of the proposed computing jobs monitoring dashboard	27
Figure 3.3	Activity diagram for the proposed interactive dashboard	34
Figure 4.1	Agile model	37
Figure 4.2	Timeline of the FYP1	45
Figure 4.3	Timeline of the FYP2	46
Figure 5.1	Create Liar in the WayScript	48
Figure 5.2	Steps to start development	49
Figure 5.3	Configuration of Cron	50
Figure 5.4	Cron schedule epression	50
Figure 5.5	Create requirements.txt	51
Figure 5.6	Deploy the script	51
Figure 5.7	The storage account has been deployed	52
Figure 5.8	Create a new container	53

Figure 5.9	Access Key	53
Figure 5.10	Placing the access key in the web scraping	54
Figure 5.11	The scraped data stored in “webscraping” blob	54
Figure 5.12	The flowchart of the web scraping	55
Figure 5.13	The process of the data cleaning	56
Figure 5.14	The preprocessing of the training model	58
Figure 5.15	The weights for each class	59
Figure 5.16	Grid search returns well-performed hyperparameters	63
Figure 5.17	The accuracy of Logistic Regression	63
Figure 5.18	The accuracy of each training model in the evaluation data	65
Figure 5.19	Data annotation	66
Figure 5.20	NER training pipeline	68
Figure 5.21	The precision, recall and f1-score of the NER model	69
Figure 5.22	The precision, recall and f1-score of the NER model in the evaluation data	69
Figure 5.23	Overview of the dashboard (1)	70
Figure 5.24	Overview of the dashboard (2)	71
Figure 5.25	The modal named About Dashboard	72
Figure 5.26	Download data from the data table	73
Figure 5.27	Choropleth Map	74
Figure 5.28	Pie Chart	75
Figure 5.29	Bar Chart	76
Figure 5.30	WordCloud	76
Figure 5.31	Funnel Chart	77
Figure 5.32	Tree Map	78
Figure 5.33	Expand the Tree Map	79
Figure 5.34	Bubble Chart	80
Figure 5.35	Settings in dashboard	81
Figure 5.36	Alert message	81
Figure 5.37	The code for filtering data in dashboard	83

LIST OF TABLES

Table Number	Title	Page
Table 2.1	Information about libraries from	12
Table 2.2	Critical Remark of Python Library and Web Data Extractor	22
Table 2.3	Critical Remark of D3.js, Tableau and Plotly'sDash	23
Table 3.1	Use Case Description for "View About Dashboard" Use Case	28
Table 3.2	Use Case Description for "Filtering Data" Use Case	29
Table 3.3	Use Case Description for "View Data Summary" Use Case	30
Table 3.4	Use Case Description for "View Data Visualization" Use Case	32
Table 4.1	Hardware requirements	37
Table 4.2	Software Requirements	38
Table 4.3	Verification Plan for Filtering Data	40
Table 4.4	Verification Plan for Filtering and Sorting Job Postings in Data Table	41
Table 4.5	Verification Plan for Regenerate a New when Further Filtering is Performed	42
Table 5.1	The Description of Each Hyperparameter	61
Table 5.1	The Hyperparameter used and Accuracy of Each Training Model	62

LIST OF ABBREVIATIONS

<i>CLI</i>	Command-Line Interface
<i>CSV</i>	Comma-Separated Values
<i>CSS</i>	Cascading Style Sheets
<i>CPU</i>	Central processing unit
<i>D3</i>	Data-Driven Documents
<i>DOM</i>	Document Object Model
<i>GIF</i>	Graphics Interchange Format
<i>HTML</i>	Hypertext Markup Language
<i>HTTPS</i>	Hypertext Transfer Protocol Secure
<i>ICT</i>	Information and Communications Technology
<i>IDE</i>	Integrated Development Environment
<i>IP</i>	Internet Protocol
<i>JPG</i>	Joint Photographic Expert Group
<i>JSON</i>	JavaScript Object Notation
<i>NER</i>	Named Entity Recognition
<i>NLP</i>	Natural Language Processing
<i>OS</i>	Operating System
<i>PNG</i>	Portable Network Graphics
<i>SDLC</i>	Software Development Life Cycle
<i>SVG</i>	Scalable Vector Graphics
<i>URL</i>	Uniform Resource Locator
<i>WEKA</i>	Waikato Environment for Knowledge

	Analysis
<i>XLSX</i>	Excel Microsoft Office Open XML Format Spreadsheet file
<i>XML</i>	Extensible Markup Language

CHAPTER 1: INTRODUCTION

1.1 Background Information

Nowadays, citizens in Malaysia use online job recruitment platforms to find the right job. Passive and active job seekers also tend to prefer online applications to traditional application methods, because it saves time and money and makes it possible to browse through a wider range of job offers [1]. There is no denying the fact that online job recruitment platforms are replacing traditional media such as newspapers and flyers to find the relevant and suitable jobs. Some of the best-known online recruitment platforms in Malaysia are JobStreet, Indeed, LinkedIn, and etc. The advantages of these online job recruitment platforms include providing the latest information on employment, job filtering, information on active industry and many more. In addition, these online job recruitment platforms allow employers to post job vacancy and assist them to choose the potential candidate for the job in a short period of time. On the other hand, job seekers can also provide their working experiences at their user profile created at the job recruitment platforms. By this, the employers can browse these details to stream the potential candidate for interviews. Hence, online job recruitment platforms act as a two-way bridge between employers and candidates, meeting the needs of both parties.

1.2 Problem Statement

Although the online job recruitment platforms in Malaysia provide a lot of straightforward and easy to grasp information for jobs vacancies. However, there are only a few information that job seekers are mainly interested such as the salary details, job location, job requirements, and etc. Obviously, other information such as the company's culture, overview, size, and etc. are the secondary concerns. In this case, the online job recruitment platforms are unable to filter the primary information that the job seekers are not interested in for a particular job vacancy posting.

The online job recruitment platforms have a large number of job postings. Unfortunately, these platforms do not utilize this vast information or data for further

CHAPTER 1: INTRODUCTION

analysis in order to provide useful information to the job seekers. As the objective of online portals is not tied to research but rather to providing a platform on which demand and supply meet, data are seldom stored and used as an input to analyze labor market trends and developments [2]. For example, the average range salary for a particular job, the primary skills required for a particular job, the demand for a particular industry in the state, etc. are all valuable, helpful, and informative for the job seekers in Malaysia.

Nowadays, it is an information age, and the job market offers many computing jobs and there is a great demand for high-tech professionals specializing in the field of information technology. Along with the fast development in information and communication technology (ICT), job skills required by ICT industries are also evolving very rapidly [3]. Besides, it also makes the market demand for different computing jobs and the average salary will be adjusted in every year. All of the above information is important for job seeker to get a realistic view of the computing jobs in Malaysia. However, there are no tools or software available in Malaysia to monitor the online job recruitment platforms in order to extracting relevant, useful, and informative data for those interested job seekers, programmers, or graduates.

1.3 Motivation

The aim of this project is to develop a computing job monitoring dashboard in Malaysia. The backend of this computing job monitoring dashboard enables monitoring of online job recruitment platforms available in Malaysia and extracting the useful and informative data for the job seekers for further analysis. It would be quite of a challenge to extract the relevant data successfully from the online job recruitment platforms as they are vast. However, this becomes one of the motivations to develop the job monitoring dashboard in this project. A dashboard is a platform of data visualization in order guide the users to interpret and understand the data efficiently. In this project, the proposed dashboard is able to visualize the useful and informative data that the job seekers are interested in.

CHAPTER 1: INTRODUCTION

1.4 Project Objectives

1. To automatically extract relevant data from online job recruitment platform

As the job postings in JobStreet and Indeed contain a lot of information, it is necessary to filter some of the minor data. This project focused on extracting eight categories of data including job title, company, location, qualifications, salary, job's requirements, years of relevant job experience and application link. Due to the large data set that needed to be extracted, the ability to automate and manipulate large data sets is extremely important in the process of data scraping. It is also important to ensure that automated scraping of data is in high accuracy and recovery rate to ensure the integrity of the data being extracted.

2. To analyze the extracted data by generating some valuable and meaningful information.

Another objective of this project is to analyze the extracted data to generate valuable and meaningful information. This dashboard will provide different categories of analysis to ensure that different user groups, such as graduates, higher education institutions and companies, can find out what they want to know about the current computing job market analysis.

In addition, the section on data analysis is focusing on data statistics. The data statistics includes the distribution of a computing job, analysis of the main skills required for a computing job, popular qualifications, the type of computing job that are popular in the current job market, the lesser-known computing job.

3.To visualize analyzed data in an interactive dashboard

The data that has been analyzed is mostly presented in words and figures. The intention of visualizing the data is to make it easier for the user to understand the analyzed data in an intuitive and visual way. In addition, the project also aims to develop a highly interactive and centralized monitorable dashboard. The visualized data will also be placed in the dashboard according to different categories so that users can more easily navigate through the information they want to know.

1.5 Project Scope and Direction

1. Focuses on scraping data on computing work in the field of computer science and IT

This project will divide all computing jobs into two categories, which are computer science and information technology (IT). The reason for scraping mainly only the above two categories of computing jobs is that there are a large and consistent number of job postings on the online job recruitment platforms such as Jobstreet and Indeed. Large and stable data is required to facilitate and support data analysis and to gain a better understanding and generate new insights into the computing job market in Malaysia. Figure 1.1 shows the statistics of Job street's data for computer science and IT. There are 5,371 computer science and 18,820 IT of job postings on Jobstreet, so this amount of data is conducive to data analysis.

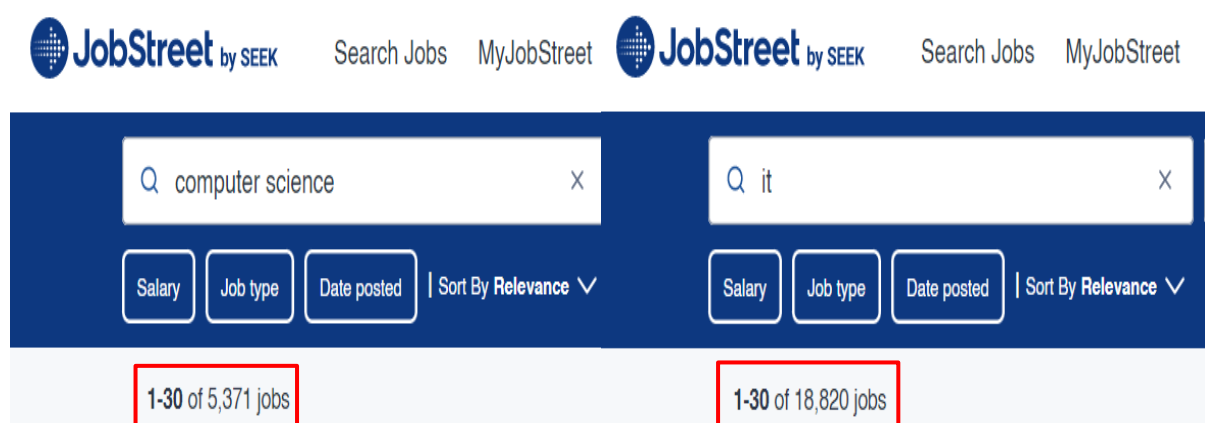


Figure 1.1: Computer science and IT job postings in JobStreet

2. Using BeautifulSoup for data scraping and WayScript for automated scraping

One of the final deliverables of this project was to complete a program that would automatically scrape data from an online job recruitment platform. BeautifulSoup will be the main method of scraping data in this project. In addition, the scraping script programmed in BeautifulSoup will be deployed to WayScript for automated scraping. The project now is planning that the web scraping script will be triggered every 2 weeks in WayScript. This means that the data in the dashboard will be updated every 2 weeks, giving users latest view of computing job market trends and analysis. In addition, section 2.2.1.2 of chapter 2 introduces BeautifulSoup and section 5.1 in Chapter 5 describes the WayScript setup.

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 1: INTRODUCTION

3. JobStreet and Indeed were chosen as the source of scraping data

The online job recruitment platform like JobStreet and Indeed were chosen as the source of scraping data for this project because it has a large and updated pool of job postings. Besides, the information in job postings is valuable because it reflects the current state of the market and trends for particular jobs. JobStreet and Indeed have been selected as the job recruitment platforms for scraping relevant data in this project which provides reliable and authentic information as well as a consistent format for job postings. Specifically, the project will only scrape some important data in the job postings and not all information will be scrapped.

4. Using Plotly's Dash to build an interactive dashboard and understandable visualization of data.

This project will use the Dash framework to build a highly interactive dashboard and Plotly for data visualization. The proposed dashboard allows users to explore different data by selecting different filtering options. In addition, the dashboard provides different charts such as maps, bar charts, pie charts, etc. to visualize the data and allow the user to analyze the data more intuitively.

1.6 Contributions

The expected main contribution of this project is the development of computing job monitoring dashboard that aims as a reference platform to the job seekers in Malaysia for choosing the suitable computing job as per their qualifications. Due to the rapid evolution of information technology, higher education institutions can use this dashboard to adapt future course material to ensure that graduates' skills are aligned with today's information technology industry. Apart from that, this dashboard is useful to the students to prepare themselves for information technology industry opportunities, such as understanding which computing jobs are in highest demand, which programming languages are a priority to learn, etc. Moreover, this proposed dashboard is useful for IT industry in Malaysia to perform market analysis in order to analyze companies or computing jobs that have the potential to grow in the nearest future.

CHAPTER 1: INTRODUCTION

1.7 Report Organization

The details of this report are shown in the following chapters. In Chapter 2, the online job recruitment platforms, web scraping and data analysis, and approaches to program interactive dashboard are reviewed. The system architecture, use case diagram and descriptions and activity diagram are discussed in Chapter 3. Chapter 4 is about the system methodology, system requirements, user requirements, non-functional requirements, verification plan and the implementation issues and challenges of this project. Besides that, the Chapter 5 has discussed about system implementation including the web scraping, data cleaning, data analysis and the final deliver dashboard. Finally, this report will be concluded with Chapter 6.

CHAPTER 2: LITERATURE REVIEW

2.0 Introduction

The purpose of literature review is to review the existing online job recruitment platform in Malaysia, to explore and evaluate the existing web scraping and data analysis tools and the existing dashboards. In addition, this literature review will list out the advantages and disadvantages of the reviewed web scraping tools and dashboards.

2.1 Online Job Recruitment Platform in Malaysia

2.1.1 Jobstreet.com

One of the online media used by today's society to meet the needs of job vacancy information is Jobstreet.com [4]. Jobstreet.com serves as facilitator of matching and employment communication between job seekers and companies in Malaysia and other Southeast Asian countries [4]. In addition, Jobstreet.com provides job seekers with the latest job postings. Each job posting has detailed information such as salary, location, job description, company registration number, and other information. This platform provides reliable, clear and highly transparent information on job opportunities to job seekers. By this, it helps the job seeker to gain confidence to use the platform. Besides, job seekers can use the job filters provided by Jobstreet.com to find the right job for them. This can be filtered by keywords, location and job specialization as shown in Figure 2.1. Job seekers can also submit their resume directly to any company via this platform, which indirectly saves the job seekers's time. In addition, the platform provides job seekers with real reviews of companies. The benefit of this feature is that it allows job seekers to have a more comprehensive understanding of any company before making a better career decision. Therefore, Jobstreet.com is a reputable and trusted online job recruitment platform in Malaysia.

However, Jobstreet.com does not provide further data analysis or market analysis that would enable the job seekers to better understand the current job market in Malaysia. These data analyses or market analyses could also be used as important reference

CHAPTER 2: LITERATURE REVIEW

material for the job seekers to be able to find suitable jobs. In addition, job seekers are not able to filter details that are not of their interest in this job searching platform.

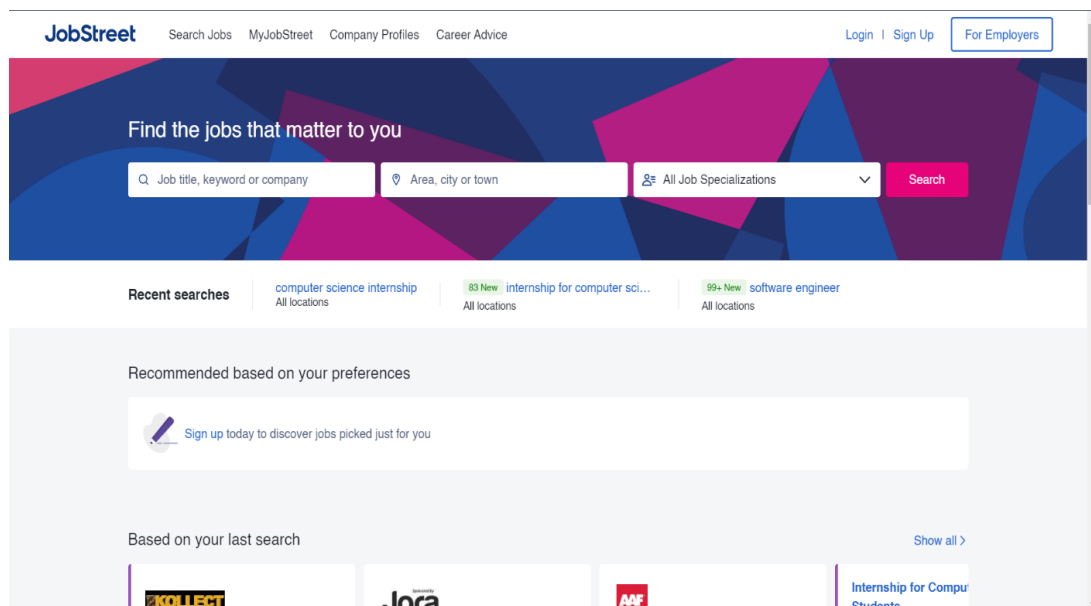


Figure 2.1: Overview of Jobstreet.com

2.1.2 Indeed.com

Indeed.com is a reputable online recruitment platform for both job seekers and employers in over 60 countries. Indeed.com's job search engine provides job seekers with many filtering options, such as finding the right job by company, job location, job posting time, job type and salary estimate. In addition, Indeed.com offers a resume builder for job seekers. In this way, job seekers can efficiently and easily fill out their resumes in prepared templates and send them to the companies. Job seekers can also view company reviews, including reviews from current employees, company ratings and job happiness, ensuring that the job information provided by Indeed.com is transparent and unbiased. To enable employers to reduce the cost of finding the right person for the job, Indeed.com offers a candidate management tool. This tool helps employers effectively manage their recruitment pipeline. The features include screening candidates, grouping candidates by status, and candidate matching. In addition, candidate assessment and interview scheduling can all be done on Indeed.com. As a result, Indeed.com is well received by both job seekers and employers.

CHAPTER 2: LITERATURE REVIEW

Indeed.com claims that 250 million people visit the platform every month, that over 3,000,000 companies use Indeed for recruitment and that it has 60 million stored resumes. All of these statistics are exciting, especially as this allows different fields to study and analyze the data and generate different insights. For this study, we focused on Indeed's employee reviews of Fortune 50 companies, with permission from Indeed to gather and analyze the data [5]. This paper uses millions of employee reviews on Indeed.com as a data source to analyze potential themes related to salient factors of employee satisfaction.

In fact, Indeed.com has no intention to share this reliable and huge data to the public. Besides, Indeed.com also does not share information on the platform that job seekers want to know, such as market analysis, career distribution, average salary, etc. The above information could help job seekers better understand their chosen career and market demand. However, most online recruitment platforms are not willing to provide this valuable information, and Indeed.com is certainly no exception.

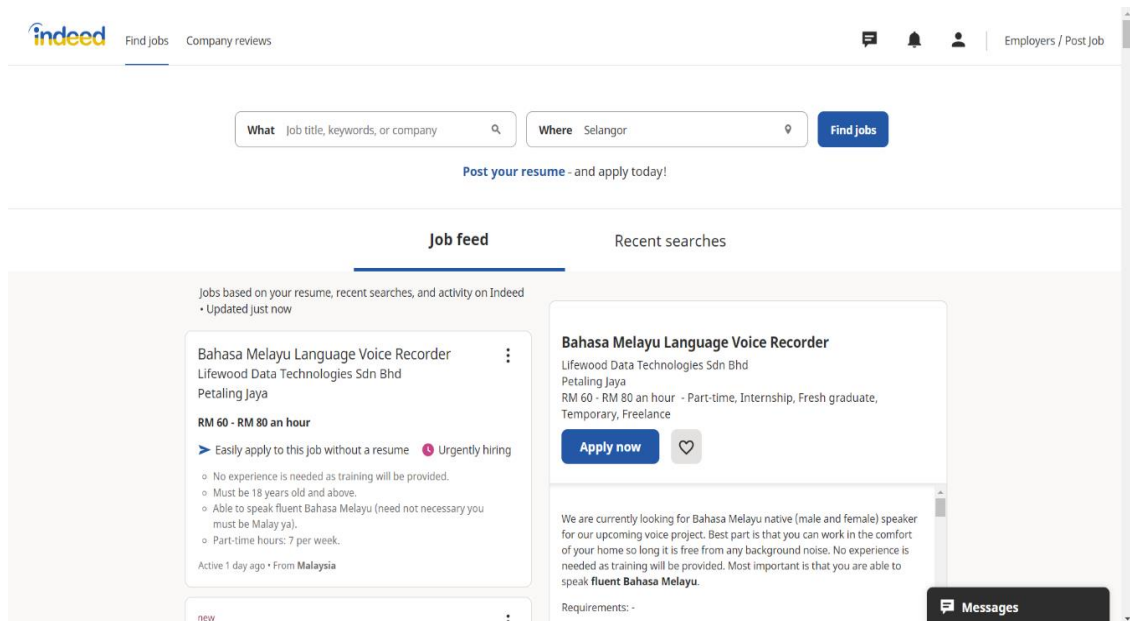


Figure 2.2: Overview of Indeed.com

2.2 Web Scraping and Data Analysis

2.2.1 Python Library

Python can be used for entire processes, including web scraping, data analysis, visualization, and so forth [6]. This paper has used Python for web scraping Indeed.com (an online job platform) to collect data about the computer science industry including job titles, salaries, skill requirements, and other relevant data [6]. In addition, Natural Language Processing (NLP) techniques are used to analyze the data. Python is a well-known high-level programming language, and Python provides variety libraries to perform many complex tasks in a simple and efficient way. For example, the authors use Python's libraries for web crawling, data analysis, data pre-processing, and data visualization. Figure 2.3 below shows an overview of the web scraping and data analysis processes using Python.

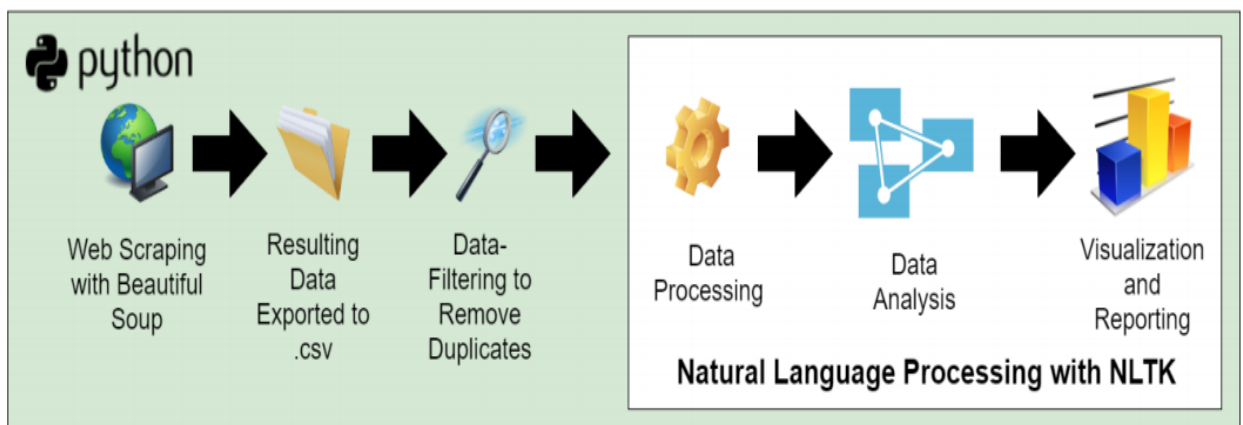


Figure 2.3: Overview of the process of web scraping and data analysis using Python

Referring to Figure 2.3 as for the web scraping, the authors have proposed to use BeautifulSoup and lxml from Python's libraries to extract the data from HTML and XML files. This is because the main data to be crawled is from web pages, so the aforementioned two Python libraries can be used to extract data from the web pages effectively. In addition, these two libraries can be used to program a script that only scrapes data for a certain target instead of scraping all the data that exists in the web page, as this can improve the performance of the web scraping and reduces the scale of the scraping data. Later, the scraped data from the web page is exported to a usable format such as CSV, XLSX and JSON.

CHAPTER 2: LITERATURE REVIEW

The exported data requires data cleaning and data processing to reduce data noise (a large amount of meaningless information). Regular-Expression Tokenizer, Brown Corpus, WordNetLemmatizer and Stopwords are utilized to further process the data, which includes the removal of common English words, symbols, and etc. A well-developed data cleaning and processing can improve the accuracy in data analysis models. Different data analysis models can be chosen for different purposes. This paper focused on the use of NLP techniques to process and analyze scraped data in a meaningful way, especially when the scraped data is a textual or linguistic data. The final step is to visualize the results of the data analysis by using charts or histograms for better representation and understanding by the user. In this work, it used WordCloud, Matplotlib and other libraries provided by Python to visualize the data.

2.2.1.1 Pandas

Pandas is the Python library that provides integrated, intuitive routines for performing common data manipulations and analysis on such data sets [7]. Besides, pandas also a tool for working with structured data sets common to statistics, finance, social sciences, and many other fields [7]. In particular, pandas is a library often used in data science. Table 1 presents information about the more famous libraries for data science from GitHub. According to Table 2.1, pandas is used by 693k other repositories in GitHub, second only to NumPy. Pandas is also a high-level data structures which pandas' core data structure is the DataFrame and it able to handle heterogeneous tabular data structures in a more streamlined way. Moreover, pandas have excellent performance in data alignment, missing data statistics, merging, groupby and other data manipulation functions. Therefore, pandas is also often used to filter, clean and aggregate data before doing further analysis on it. Pandas has many different functions to access diverse data sources (CSV, JSON, spreadsheets, database tables, and many more) allow to focus on the actual data processing instead of data loading and formatting [8]. Pandas supports a total of 14 different file formats, so it allows data scientists or programmers to not worry about pandas' compatibility issues.

Table 2.1 Information about libraries from

Library	Stars	Forked	Contributors	Used by
NumPy	19.9k	6.7k	1286	963k
pandas	33.1k	14.1k	2541	693k
Matplotlib	15.2k	6.3k	1143	501k
SciPy	9.3k	4.2k	1112	458k
scikit-learn	49.4k	22.8k	2284	317k
TensorFlow	164k	86.5k	3082	182k
PyTorch	54.7k	15.1k	2184	112k
Keras	54.7k	19k	998	N/A
Caffe	32.3k	19k	269	N/A

Note:

- 1) Library: Name of the library from GitHub
- 2) Stars: Indicate that someone like the library
- 3) Forked: The number of fetch updates from or submit changes to the original library with pull requests
- 4) Contributors: The number of contributed something back to the library
- 5) Used by: The number of repositories that depend on the library
- 6) N/A: Not available
- 7) Data is from March19th, 2022

2.2.1.2 Beautiful Soup

Beautiful Soup is a Python package based on the foundation of HTML/XML analytics engine, used for extracting, analyzing, and editing information in the Document Object Model (DOM) of webpages [9]. Beautiful Soup is very useful and efficient for web scraping. This is because it uses a simplified and straightforward approach to allow beginners to quickly scrape data on the target's webpages. A simple web scraping program can be built using the Beautiful Soup and requests library. For example, Figure 2.4 shows that it only takes 3 lines of code to scrape data from Shopee webpage. The web scraping program in Figure 2.4 uses the `get()` function to send a GET request to the specified URL, and the `BeautifulSoup()` function to parse the returned HTML or XML file and store it in a data structure. Since Beautiful Soup extracts information from the DOM of a webpage, the content returned in the parsed HTML or XML file will contain HTML tags and metadata. As shown in Figure 2.5, the scraped data or content is embedded in different HTML tags (red frames indicate HTML tags and metadata). Thus, Beautiful Soup can extract single or multiple occurrences of HTML tag by using `find_all()` or `find()` functions and uses the `get_text()` function to extract the content in that particular HTML tag. Combining the data responses from experimental users, using Beautiful Soup for information retrieval achieved an accuracy of nearly 100% [9]. This also indicates that Beautiful Soup has reliable and accurate scraping

CHAPTER 2: LITERATURE REVIEW

data on the webpage, and it also provides multiple functions to ensure that relevant scraping data can be accurately extracted.

```
1 from bs4 import BeautifulSoup
2 import requests
3
4 #Make an HTTP request to get HTML content via the specific URL
5 url = 'https://shopee.com.my/Leather-Watches-col.1048348'
6 response = requests.get(url)
7
8 #Create a BeautifulSoup object and define the parser
9 soup = BeautifulSoup(response.text, 'html.parser')
```

Figure 2.4: Scraping data from Shopee using BeautifulSoup and request

```
<html>
<head>
<link href="//cf.shopee.com.my/" rel="preconnect"/>
<link href="//de0.shopeemobile.com/shopee/" rel="preconnect"/>
<link href="//cv.shopee.com.my/" rel="preconnect"/>
<meta charset="utf-8"/>
<meta content="width=device-width,initial-scale=1,maximum-scale=1,minimum-scale=1,user-scalable=no,viewport-fit=cover" name="viewport"/>
<meta content="aca16d7f62cbf93e6be58fe4a4db35e292e66e1c" name="shopee:git-sha"/>
<meta content="rw-v4.1.10" name="shopee:version"/>
<link as="style" data-modern="true" href="https://de0.shopeemobile.com/shopee/shopee-mobileall-live-sg/assets/entry-modules.fbd8ca58505b0c644706.css" rel="preload"/>
<link as="style" data-modern="true" href="https://de0.shopeemobile.com/shopee/shopee-mobileall-live-sg/assets/bundle.eb60c8768486254510f9.css" rel="preload"/>
<link data-modern="true" href="https://de0.shopeemobile.com/shopee/shopee-mobileall-live-sg/assets/webpack-runtime.d0382f0d1909e4d82e7f.js" rel="modulepreload"/>
<link data-modern="true" href="https://de0.shopeemobile.com/shopee/shopee-mobileall-live-sg/assets/entry-modules.d9d72cace76305452323.js" rel="modulepreload"/>
<link data-modern="true" href="https://de0.shopeemobile.com/shopee/shopee-mobileall-live-sg/assets/bundle.c7998a9601adc23c25da.js" rel="modulepreload"/>
<link data-modern="true" href="https://de0.shopeemobile.com/shopee/shopee-mobileall-live-sg/assets/modules.49ef89b6d6ab78cafd97.js" rel="modulepreload"/>
```

Figure 2.5: The scraped data is embedded in different HTML tags

CHAPTER 2: LITERATURE REVIEW

2.2.2 Web Data Extractor

Web Data Extractor is a powerful and easy-to-use application which helps user automatically extract specific information from web pages [10]. This tool allows the user to change the web scraping settings according to the user preferences. This feature reduces the searching scope of web scraping and increase the rate of obtaining the useful scraping data. Users can select any of the data sources provided by the tool, including websites, search engines and URL list. Besides, the user can further filter the selection of the target website, the search engine to be used and the settings in the URL list. The maximum scraping depth for this tool is 100 pages, which is generally sufficient to scrape the required data. This tool able to extract different data such as URL, domains, Meta Tags, emails, phones, proxies, and so on . The extraction of custom data has also been added for increasing the variety of data that can be extracted. The tool also has good data processing performance, such as the ability to filter duplicate data and user-defined keywords. Figure 2.6 shows the setup for scraping computer science internship data.

This tool also has excellent web scraping performance. As shown in Figure 2.7, it is found that the tool was able to successfully scrape 85 emails and 338 phone numbers in 1 minute from 193 websites related to computer science internships. At the end, the scraped data can be exported in CSV or other formats. In summary, Web Data Extractor is a tool that can manipulate large data sets and perform well in web scraping.

CHAPTER 2: LITERATURE REVIEW

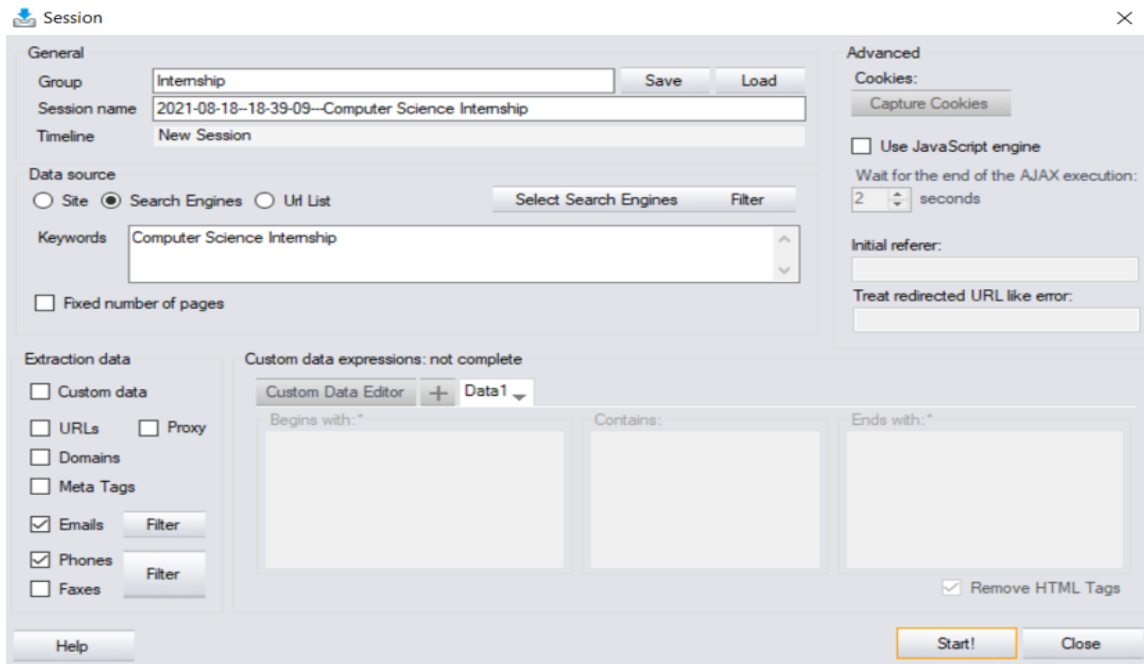


Figure 2.6: Creating a session in Web Data Extractor

The screenshot shows the 'Stored Sessions' window in Web Data Extractor. A table displays the results of a scraping session. The table has columns for Email, Name, Url, Title, Host, Keywords density, and Keywords. The status bar at the bottom indicates 'Processing time: 00:01:03.323', 'Sites processed: 193 / 649', 'Downloaded: 61,938 KB', and 'Avg. Speed: 1,000 KB/s'.

Email (85)	Phone (3383)	Name	Url	Title	Host	Keywords density	Keywords
nidhi.shama@gras...		nidhi.shama	https://gras.com/internship	Online Internship Training ...	gras.com	0	internship, science, ...
HTIU@usdoj.gov		HTIU	https://www.justice.gov/criminal-ceo...	Computer Forensics Intems...	justice.gov	0	computer, internsh...
complaints@intems...		complaints	https://intemshala.com/internships/s...		intemshala.com	2	science, internsh...
helpdesk@intemsh...		helpdesk	https://intemshala.com/internships/s...		intemshala.com	2	science, internsh...
Lker2@pride.hofst...		Lker2	https://cs.hofstra.edu/docs/pages/g...	How to install Unity I Comp...	cs.hofstra.edu	0	computer, science, ...
me@email.com		me	https://www.careercliff.com/letter-for...	7 Steps to Write Letter for I...	careercliff.com	0	internship
rbanderson@sdsu...		rbanderson	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
mike@seasats.com		mike	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
info@linkmunch.com		info	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
miguel@enstal.com		miguel	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
Charlie@denroomm...		Charlie	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
ecury@burwood.c...		ecury	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
sreyes@bikmtm.com		sreyes	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
Robotics Internsh...		Robotics Intems...	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
fpentieri@globel...		fpentieri	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
collegecareers@v...		collegecareers	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
EPamintuan@sdsu...		EPamintuan	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
careers@ipsgroup...		careers	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
gene@gtlconsulti...		gene	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
christine.knights@...		christine.knights	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
eric.gerhardt@sdc...		eric.gerhardt	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
jimpoet@hotmail.com		jimpoet	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
jswager@helixelec...		jswager	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
anara@sdsu.edu		anara	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
ggallo@teksystems...		ggallo	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
lmorrison@calamp...		lmorrison	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
tim.ortiz@neocortex...		tim.ortiz	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
tim.ortiz@neocortex...		tim.ortiz	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
jalane@sdsu.edu		jalane	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
carol.c.nelepovitz...		carol.c.nelepovitz	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
carol.c.nelepovitz...		carol.c.nelepovitz	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
SRDI_Recruiting@...		SRDI_Recruiting	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
tingtingchen@cpp...		tingtingchen	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
frang@cpp.edu		frang	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
set2@cornell.edu		set2	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
equalopportunity@...		equalopportunity	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
kindgel@visa.com		kindgel	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
tim@centrail.com		tim	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
careers@dronecita...		careers	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...
astalio@calstate...		astalio	https://cs.sdsu.edu/current-job-anno...	Computer Science Depart...	cs.sdsu.edu	2	computer, science, ...

Figure 2.7: Web scraping performance of Web Data Extractor

CHAPTER 2: LITERATURE REVIEW

2.3 Dashboard

2.3.1 D3.js

Data-Driven Documents (D3) is a JavaScript library for manipulating documents based on data allowing us to bind arbitrary data to a Document Object Model (DOM) [11]. This paper used WEKA for clinical data mining [11]. WEKA is an open-source software package that assembles different data mining and model building algorithms. However, it was found that WEKA has performed poorly in terms of data visualization. For example, the scatter plot of the Plot Matrix in WEKA did not clearly label the data. Therefore, have used D3.js to optimize WEKA's shortcomings in visualization and design a dashboard for physicians to analyze clinical data [11].

D3 has a high degree of flexibility and provides many tools for manipulating data. In addition, data can be mapped to HTML structures or SVG documents. The SVG format can render images of any size without degrading their quality unlike PNG, GIF or JPG formats that are able to degrade the image quality for its sizes. In this case, D3 has better visualization than other tools. D3 can also be used with CSS to design personalized dashboards and jQuery to trigger different event handling. The advantage of D3 is that it can support different data formats, including CSV, JSON and GeoJSON. Apart from that, D3 is able to perform well in terms of interaction and animation with large data sets [11]. In this case, D3 box plots to display data for numerical attributes, bar and pie charts to display data for nominal attributes in this dashboard design [11]. As a result, the D3 can handle any dashboard design with excellent interactivity.

CHAPTER 2: LITERATURE REVIEW

2.3.2 Tableau

Tableau is a tool that focuses on business intelligence and is also an excellent visualization tool [12]. One of the reasons most markets choose this tool for data visualization is that Tableau's drag-and-drop functionality allows users to build a complete and interactive dashboard. The drag-and-drop feature reduces the time spent on creating dashboards and allows users who are without programming background able to easily manipulate large datasets in a short period of time. In addition, Tableau supports the import and export of different types of formats such as PDF, JSON, CSV, etc. and can connect to different types of databases including MySQL, Oracle, etc as shown in Figure 2.8. This gives the users the maximum flexibility to determine which formats and databases they prefer to connect to, rather than limiting their options.

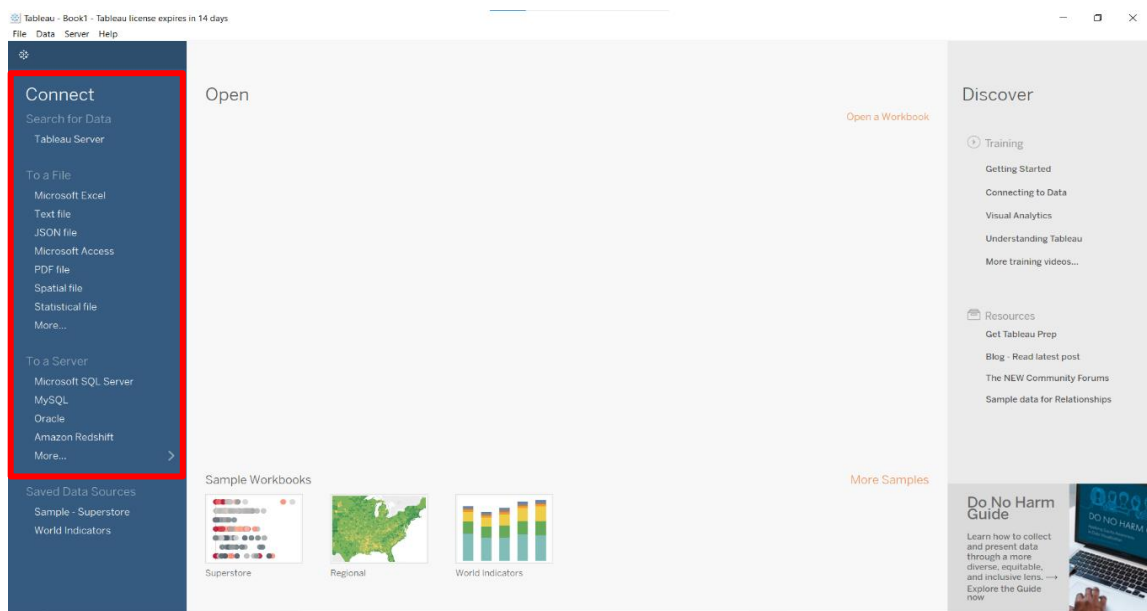


Figure 2.8: Options provided by Tableau for the import format and the database selection

Besides, Tableau offers 24 different visualization models, including bubble charts, maps, heat maps and more as illustrated in Figure 2.9. The above visualization models are generally sufficient for large data sets. Users can also select the attributes to be visualized from different datasets and create instantly. These actions can be performed with the drag and drop function as stated in the earlier paragraph. Tableau also offers the user a choice of common data analysis models, such as average line, median with 95% CI, total, and etc. This function also gives the user an initial insight into the content analysis and statistics of the data. One of the most surprising aspects of Tableau is its

CHAPTER 2: LITERATURE REVIEW

outstanding performance in data manipulation, such as the ability to process and visualize hundreds of thousands of pieces of datasets in a second. Therefore, this performance is ideal for visualizing the scraped data after web scraping and composing a dashboard in a short period of time.

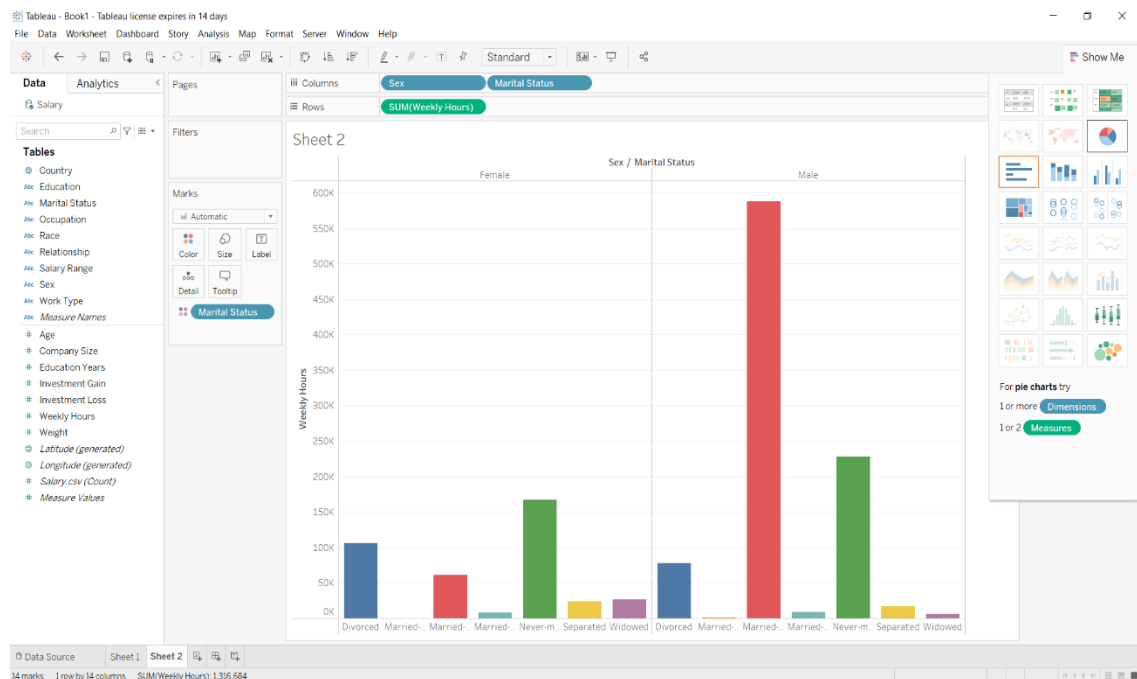


Figure 2.9: Overview of Tableau

2.3.3 Plotly's Dash

Plotly's Dash is a library that empowers data scientists to create interactive web applications declaratively in Python [13]. A highly interactive dashboard can be built using the Dash framework and the Plotly library. As well, the dashboard is programmed using only the pure Python programming language, which reduces the need for developers to learn other programming languages and increases development efficiency.

Dash is a python framework for building interactive web apps. The biggest benefit of using Dash framework to develop dashboards is that developers only need to program in Python and do not need to have any knowledge of front-end programming languages such as HTML, CSS and JavaScript. The Dash application is made up of a layout module and a callback module. The layout's building block refers to the interface of this

CHAPTER 2: LITERATURE REVIEW

dashboard. Where Dash provides Dash HTML component and Dash Core component to develop a custom dashboard interface. Dash HTML component is used to create and design HTML content and elements such as slider, checkboxes, radio button are created using Dash Core components. In addition, callbacks are used instead of JavaScript to make this dashboard interactive.

Plotly is a Python library to analyze and visualize data [14]. Plotly is very well known in the field of data science and is often used to visualize data. Besides, Plotly generates attractive and clean charts and allows users to customize any of them. Plotly offers more than 40 different charts, so it can visualize different types of data and present them in a diverse way. The charts provided include line chart, bar chart, histograms, scatter plot, bubble chart, heatmaps etc. Alternatively, developers can use the `plotly.express` or `plotly.graph_objects` classes to create charts. The `plotly.express` is usually used to generate a chart by changing some of parameters, while graphs that require more manipulation and customization are generated using `plotly.graph_objects`.

CHAPTER 2: LITERATURE REVIEW

2.4 Limitation of The Reviewed Tools

2.4.1 Web Scraping and Data Analysis

Using a python library like BeautifulSoup for web scraping requires an analysis of the target website's architecture and its anti-scraping mechanisms. This is to ensure that a well-designed and automated web scraping system is in place. The aim of anti-scraping application is to keep the increase in page-loading time to a minimum, without compromising on the security [15]. The main anti-scraping mechanisms include tracking the frequency of requests from the same IP, Captchas, login access, etc. These are the challenges of designing web scraping scripts using python. It is also important to ensure that each library used is interoperable and connected. For example, the data scraped about the IT industry using BeautifulSoup must ensure that the Random Forest Classifier (a machine learning algorithm) is able to use the scraped data to build a reliable and accurate salary prediction model in the IT industry.

The most obvious limitation of web scraping tools such as Web Data Extractor Pro is unable to scrape data on the websites with sophisticated anti-scraping mechanisms. This is because existing web scraping tools are only suitable for use on common websites where the defense mechanisms are not that high. Another limitation of these web scraping tools is that some attributes cannot be fully scraped, which results in missing values and reduces the integrity of the data. Although existing web scraping tools provides data processing features, there is still a failure to thoroughly filter some data noise. This can cause a reduction in the accuracy of data analysis .

CHAPTER 2: LITERATURE REVIEW

2.4.2 Dashboard

A limitation of D3 is that D3 does not provide any prebuilt library for data visualizations for users to use. This results in the heavy coding required to create a simple visualization. At the same time, it takes more time to develop a dashboard. Besides, the use of D3 for dashboards also requires knowledge of other programming languages such as HTML, CSS, jQuery, JavaScript etc.[16]. This is to ensure the dashboard to have good interactivity and performance. The final limitation of using D3 is that D3's performance degrades when dealing with large data sets in gigabytes, whereas visualization tools such as Tableau can still perform well [16].

The most obvious limitation of visualization tools such as Tableau is the inability to design a personalized and highly interactive dashboard. This is because most visualization tools only provide simple visualization models. In addition, Tableau does not support importing or exporting to SVG format. The advantage of SVG format is highly compressible, lightweight and can be rendered at any size without compromising its quality . Lastly, Tableau does not support predictive analytics or relational data mining [16].

Plotly's Dash is an easy to program and powerful framework for creating interactive web applications, especially dashboards. However, it still has the following limitations. Dash has performance limitations which are likely the callbacks in the code itself [17]. Server callback is the important component in Dash for creating an interactive dashboard. Dash's interactivity mechanism requires the client to make requests to the server, including updating any charts. This make Dash is less efficient than executing JavaScript code in the browser. Also, Dash has its limitations with loading the visualizations [17]. Moreover, Dash itself has some problems with plotting a large amount of `plotly.graph_objects` [18]. Lastly, it is difficult for developers to categorize and manage the code of Dash applications. All the modules such as the layout of app and the callback module, were written in a single PY file. Thus, the project became more complex and larger, it resulted in code that looked messy and unmanageable.

2.5 Critical Remark

This section summaries the advantages and disadvantages of the methods and software reviewed in sections 2.2 to 2.3.

Table 2.2: Critical Remark of Python Library and Web Data Extractor

	Python Library	Web Data Extractor
Strengths	<ul style="list-style-type: none"> • Python's libraries can be applied to web crawling, data analysis, data pre-processing and data visualization • Great data cleaning performance 	<ul style="list-style-type: none"> • Allows the user to change the web scraping settings according to the user preferences • Can select any of the data sources for web scraping • Able to extract different data
Weaknesses	<ul style="list-style-type: none"> • Requires a detailed analysis of the target website's architecture and its anti-scraping mechanisms • Must ensure that each library used is interoperable and connected 	<ul style="list-style-type: none"> • Unable to scrape data on the websites with sophisticated anti-scraping mechanisms • Some data cannot be fully scraped • Poor data pre-processing performance

Table 2.3: Critical Remark of D3.js, Tableau and Plotly’s Dash

	D3.js	Tableau	Plotly’s Dash
Strengths	<ul style="list-style-type: none"> • High degree of flexibility and provides many tools for manipulating data • Data can be mapped to HTML structures or SVG documents • Can handle any dashboard design with excellent interactivity 	<ul style="list-style-type: none"> • No need programming background • Provide drag-and-drop feature to develop a dashboard • Provide simple visualization models and data analysis models • Outstanding performance in data manipulation 	<ul style="list-style-type: none"> • The dashboard is programmed using only the pure Python • Dash’s callbacks are used instead of JavaScript • Improve development efficiency • A highly interactive and customizable dashboard can be built
Weaknesses	<ul style="list-style-type: none"> • Does not provide any prebuilt library for data visualizations • Heavy coding required to create a simple visualization • Requires knowledge of other programming languages • Performance degrades when 	<ul style="list-style-type: none"> • Inability to design a personalized and highly interactive dashboard • Does not support importing or exporting to SVG format • Does not support predictive analytics or relational data mining 	<ul style="list-style-type: none"> • Dash’s callbacks is less efficient than executing JavaScript code in the browser • Limitations with loading the visualizations • Having some problems with plotting a large number of charts • Difficult to categorize and manage the code of Dash applications

CHAPTER 2: LITERATURE REVIEW

	dealing with large data sets in gigabytes		
--	---	--	--

CHAPTER 3: SYSTEM DESIGN

3.1 Overview

This chapter will explain the flow of the system and each module in the project by illustrating different design diagrams such as the system architecture, use case diagram and activity diagram. The use case description is a good way to outline the behavior of the system from the user's perspective when responding to a request, which also helps to explain well how users perform certain tasks in the proposed dashboard.

3.2 System Architecture

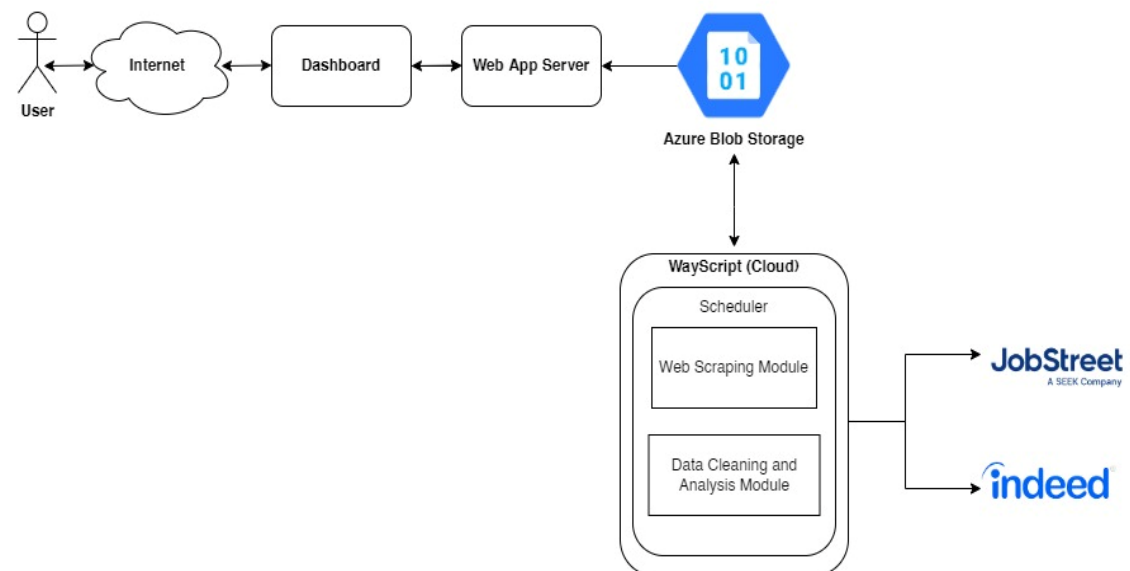


Figure 3.1: System architecture of the proposed computing jobs monitoring dashboard

Figure 3.1 shows the system architecture of the proposed computing jobs monitoring dashboard. The dashboard is the main component of the front-end, which interacts with the user. The back-end of the system includes the web application server for launching the dashboard, listening and responding to services, all scraped data or information is stored in Azure Blob storage, and WayScript is responsible for performing web scraping, data cleaning and analysis in an automated manner.

CHAPTER 3: SYSTEM DESIGN

The user can access the deployed dashboard by entering the URL of the dashboard in any browser. The user's request will be received by the listening port of the web application server, which will launch the dashboard and return the data and user interface of the dashboard to the user. The data transmitted between the browser and the website will be encrypted via Hypertext Transfer Protocol Secure (HTTPS) to improve the security of the data transmission.

In this project, Heroku (Cloud Platform as a Service) will be used as a web application server. This means programmed dashboard will be deployed to this server. The tasks of this web application server include managing requests from clients and retrieving information from storage. Besides that, the dashboard is designed as sever-side callback which any dashboard interaction triggered by the user needs to be done on this server. For example, when the user needs to select or filter certain data in the dashboard, an event is triggered and request for data filtering to the web app server. This does the corresponding action and returns the data or result to the client. Therefore, the client and server side always maintain a two-way connection, which is the key to the client's request and the server's response to achieve an interactive dashboard.

Apart from that, Azure Blob Storage will store scraped data by the online job recruitment platform. In this project, Azure Blob Storage will store two types of data, including the raw data that collected and data that has been cleaned and analyzed. The cleaned and analyzed data will be the data source of the dashboard. In addition, the data in the storage will be overwritten when latest scraped data has uploaded. This is to ensure that the data stored in the repository is up to date.

Web scraping, data cleaning and analysis are all done automatically in WayScript. Furthermore, above mentioned two modules are programed in python and deployed on the WayScript platform. WayScript provides a time trigger service which as a task scheduler, and these two modules will be executed automatically according to the time set in the time trigger. The web scraping script will be triggered at 1:00 am on the 12th and 26th of each month to scrape the required data from JobStreet and Indeed and store the results in Azure Blob Storage. Besides that, but the scraped data is also automatically stored in storage after WayScript performs data cleaning and analysis.

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

3.3 Use Case Diagram

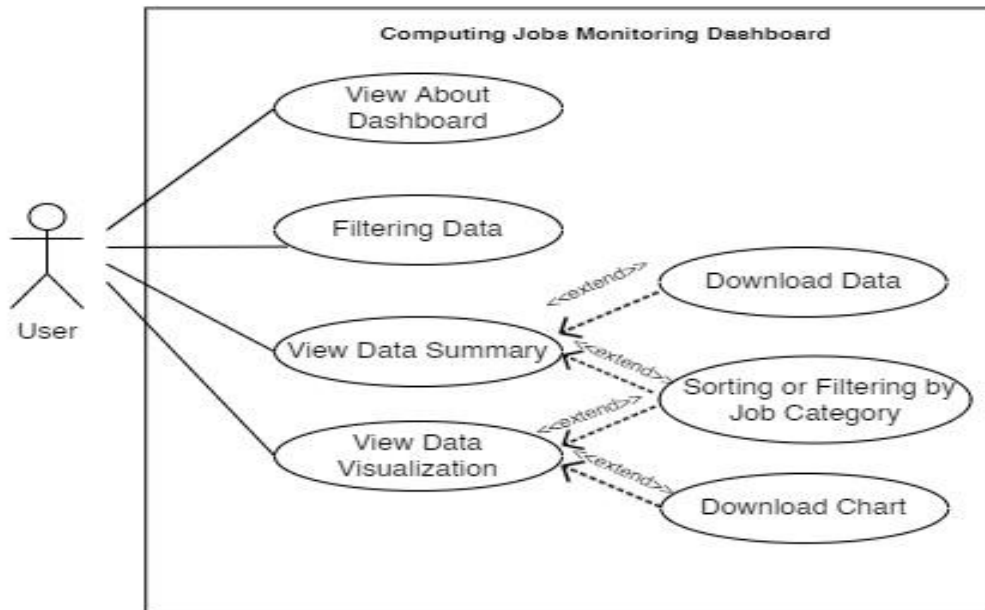


Figure 3.2 Use case diagram of the proposed computing jobs monitoring dashboard

Figure 3.2 is a use case diagram that shows the tasks that users can perform on the dashboard.

CHAPTER 3: SYSTEM DESIGN

3.4 Use Case Description

Table 3.1 Use Case Description for “View About Dashboard” Use Case

Use Case ID	UC001	Use Case Name	View About Dashboard
Primary Actor	User		
Brief Description	Users can view the dashboard's introduction to more understanding about the purpose, services offered and features of the computing jobs monitoring dashboard.		
Trigger	User clicks on the "About Dashboard" navigation option.		
Precondition	User has accessed the dashboard.		
Scenario Name	Step	Action	
Main Flow	1	User accessed the dashboard.	
	2	User clicks on the "About Dashboard" navigation option.	
	3	System shows the description of the dashboard.	
	4	User clicks the “Close” button.	
	5	System closes the “About Dashboard" and navigate to the previous page	

Table 3.2 Use Case Description for “Filtering Data” Use Case

Use Case ID	UC002	Use Case Name	Filtering Data
Primary Actor	User		
Brief Description	Users can filter certain jobs by selecting different filtering options, such as job field, state, and expected salary. The system will display the filtered results and regenerate the chart in the "Data Visualization" tab.		
Trigger	User clicks on the "Filtering" navigation option.		
Precondition	User has accessed the dashboard.		
Scenario Name	Step	Action	
Main Flow	1	User accessed the dashboard.	
	2	User clicks on the "Filtering" navigation option.	
	3	System request for input of filtering options such as job field, state, and expected salary.	
	4	User inputs filter option and clicks the “Submit” button.	
	5	System filters the jobs based on the input of filter values.	
	6	System displays the filtered results on the “Data Summary” tab, including the updated data summary and data tables.	
	7	User clicks on the “Data Visualization” tab.	
	8	System regenerates the charts based on the filtered results .	
Alternate Flow – Cannot Find Any Results	5.1	System returns null values after querying the data based on the filtering options selected by the user.	
	5.2	System prompts a warning message “No results matched your filter. Change Filter?” on the “Data Summary” tab.	
	5.3	System will not display the contents of the data table.	
	5.4	User clicks on the “Data Visualization” tab.	
	5.5	System prompts a warning message “No results matched your filter. Change Filter?” on the “Data Visualization” tab.	
	5.6	System will not display any chart and display “No Results” message.	

Table 3.3 Use Case Description for “View Data Summary” Use Case

Use Case ID	UC003	Use Case Name	View Data Summary
Primary Actor	User		
Brief Description	Users can view the data summary on the “Data Summary” tab. The content of data summary includes the number of companies and jobs, last update date of the scraped data and data table.		
Trigger	User clicks on the "Data Summary" tab.		
Precondition	User has accessed the dashboard.		
Scenario Name	Step	Action	
Main Flow	1	User accessed the dashboard.	
	2	User clicks on the "Data Summary" tab.	
	3	System displays the content of data summary.	
	4	System perform Sub Flow based on the tasks performed by the user.	
Sub Flow – Filtering Jobs on the Data Table by Job Category	4a.1	System request for input to filter jobs by job category.	
	4a.2	User selects the job category from the drop-down list.	
	4a.3	System validates the input of the job category value.	
	4a.4	System queries jobs based on the job category selected by the user.	
	4a.5	System updates the content of data table.	
Sub Flow – Sorting Jobs on the Data Table by Salary Range	4b.1	System request for input to sort jobs by salary range.	
	4b.2	User select the salary range from the drop-down list.	
	4b.3	System validates the input of the salary range value.	
	4b.4	System sorts of jobs according to the job salary range selected by the user.	
	4b.5	System updates the content of data table.	
Sub Flow – Download Computing Jobs Recruitment Information on the	4c.1	User clicks the “Download” button on the data table.	
	4c.2	System validates the data on the data table.	
	4c.3	System downloads the computing jobs recruitment information in csv format.	

CHAPTER 3: SYSTEM DESIGN

Data Table		
Alternate Flow – Cannot Find Any Results	4a.4.1	System returns null values after querying the data based on the job category selected by the user.
	4a.4.2	System will not display the contents of the data table.

Table 3.4 Use Case Description for “View Data Visualization” Use Case

Use Case ID	UC004	Use Case Name	View Data Visualization
Primary Actor	User		
Brief Description	Users can view different charts on the “Data Visualization” tab. The “Data Visualization” page provides 7 different types of charts to generate some interesting and new insight about the different aspect of job computing market in Malaysia. Besides that, user can further re-generate charts by filtering job category.		
Trigger	User clicks on the "Data Visualization" tab.		
Precondition	User has accessed the dashboard.		
Scenario Name	Step	Action	
Main Flow	1	User accessed the dashboard.	
	2	User clicks on the "Data Visualization" tab.	
	3	System displays the charts.	
	4	System perform Sub Flow based on the tasks performed by the user.	
Sub Flow – Re-generate the Charts by filtering job category	4a.1	System request for input to filter jobs by job category.	
	4a.2	User selects the job category from the drop-down list.	
	4a.3	System validates the input of the job category value.	
	4a.4	System queries jobs based on the job category selected by the user.	
	4a.5	System generates a new chart based on the filtered results.	
	4a.6	System displays the new chart.	
Sub Flow – Download the Chart	4b.1	User clicks the “Camera” icon in the upper right corner of each chart.	
	4b.2	System downloads the chart in png format.	
Alternate Flow – Cannot Find Any Results	4a.4.1	System returns null values after querying the data based on the job category selected by the user.	
	4a.4.2	System will not display the chart and prompt “No Results” message.	

CHAPTER 3: SYSTEM DESIGN

Alternate Flow – Unable to Download the Chart	4b.1.1	The system verified that the chart does not exist or that there was a problem generating it.
	4b.1.2	System will display error message.

3.5 Activity Diagram

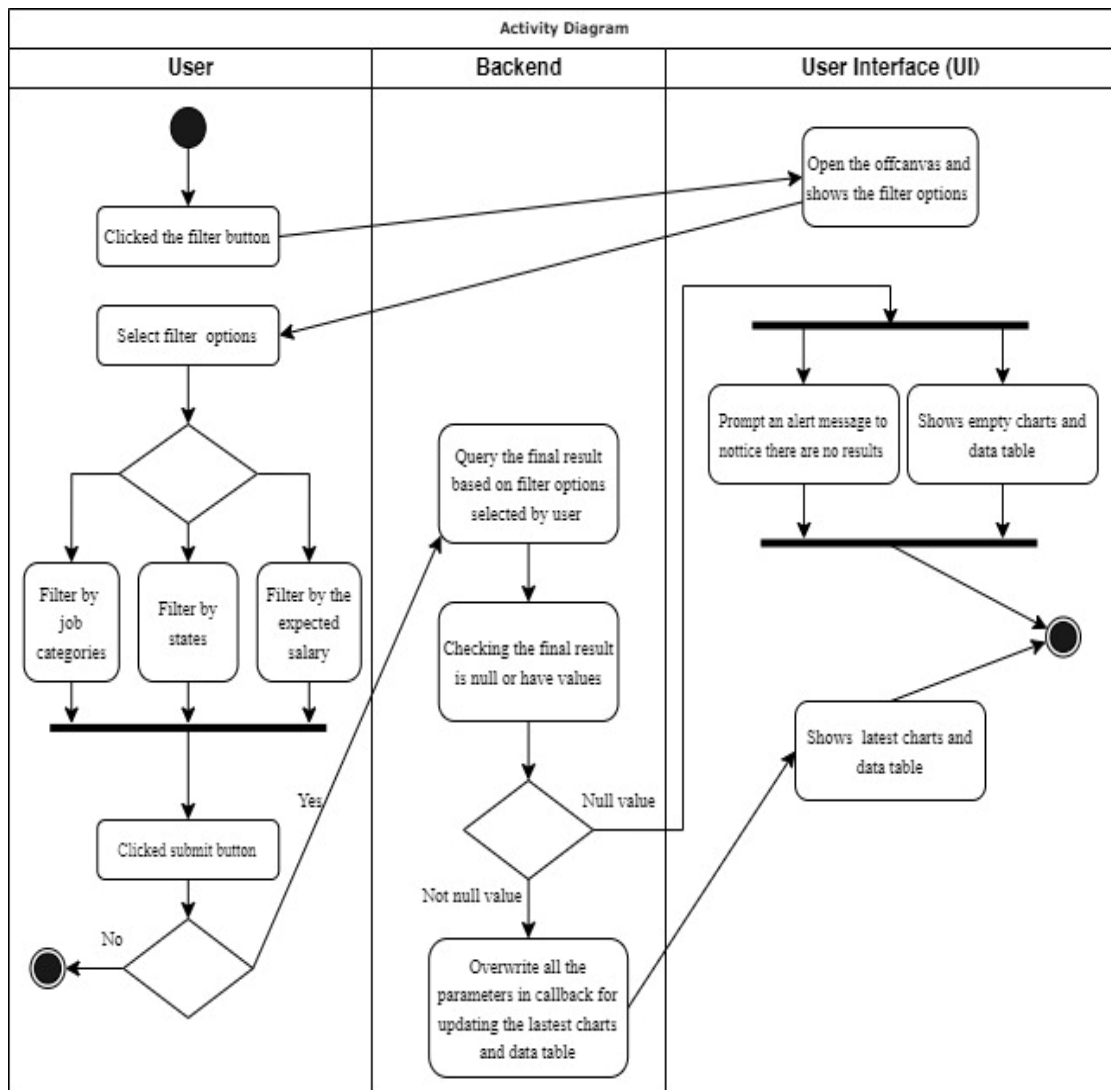


Figure 3.3 : Activity diagram for the proposed interactive dashboard

One of the goals of the project is to create an interactive dashboard. The interactive dashboard updates the data or content in the chart based on the filtering options selected by the user so that user have a new insight of data. Plotly's Dash was selected for this project to program an interactive dashboard. Besides, Plotly's Dash provides callback modules to implement interactive dashboards. Figure 3.3 shows the activity diagram of the interactive dashboard. The activity diagram is composed of 3 different modules, which are the user, the backend which is mainly the logical layer for handling queries and the user interface (UI).

CHAPTER 3: SYSTEM DESIGN

When the user wants to filter different options to generate charts with new insights, the offcanvas (a sidebar component) in the UI is triggered to allow the user to select the filtering options. This interactive dashboard provides the user with 3 different filtering options to query the data. The filtering options provided include job field, state and expected salary to allow the user to query the jobs information they want to know. There is also a submit button on the filtering interface for the user to determine the final selected filtering option. Once the user has clicked the submit button, the backend will query the appropriate data based on the filter options selected by the user. Conversely, if the submit button is not clicked, no event will be triggered.

The result of the query is either return the latest filtered data or no data found (null value returned). When the final filtered data is returned by the backend, all parameters in the callback module are overwritten to update and display the latest charts and data tables and display it in the UI. If the returned data is empty, the UI will prompt a warning message to inform the user that no data is queried and asks the user to select another filter option. In addition, all data tables and charts in the UI display blank data when no data is being queried.

CHAPTER 4: METHODOLOGY AND TOOLS

4.1 System Methodology

The project uses the agile model as the development methodology. The term agile stands for 'moving quickly' [19]. The agile model is an innovative software development model that allows for frequent changes. Besides, the agile model's Software Development Life Cycle (SDLC) takes an iterative approach to deliver the final product as illustrated in Figure 4.1.

The SDLC in the agile model can be divided into five different phases including planning, analysis, design, implementation, and testing. Any relevant system or user requirements are gathered and analyzed during the planning and analysis phase. The previous phases are studied for further system design during the design phase. Once a preliminary design is in place, the system needs to be programmed and developed in the implementation phase. In addition, the developed system needs to be tested and debugged during the testing phase. Once the testing phase is complete, the initial version of the product will be delivered and the first cycle of the SDLC will be ended. Users will review the initial version of the system and ensure that it meets their requirements. Any improvements required in the initial version of the system will be documented for reference in the planning and analysis phase of the next round of SDLC. Then, the next round of SDLC can then be started to improve the previous product and deliver a new version of the system.

Hence, the agile development model takes an iterative approach to building systems and increments the functionality of each build. In the end, a final version of the system is delivered that meets the needs of the user.

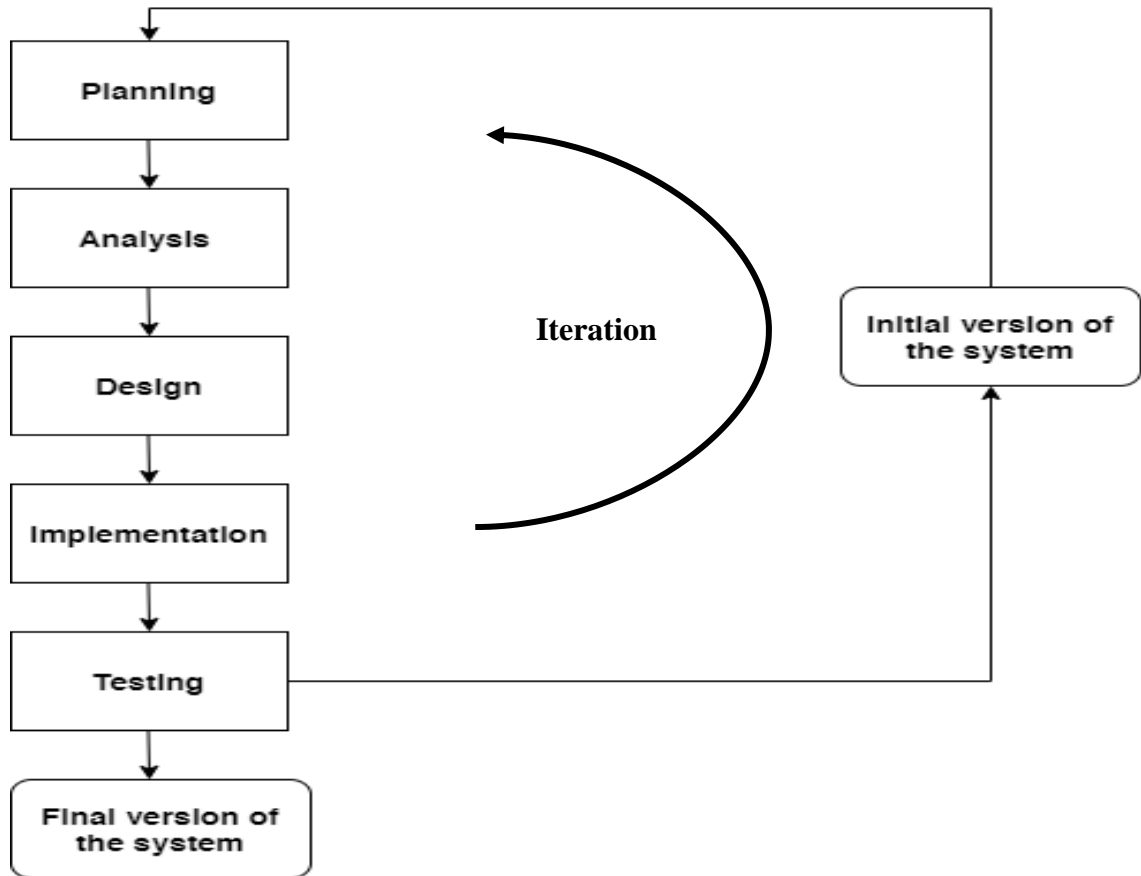


Figure 4.1: Agile model

4.2 System Requirement

4.2.1 Hardware Requirements

Table 4.1 Hardware requirements

Hardware component	Specification
Processor	Intel Core i5-8250U CPU (1.60GHz)
Memory (RAM)	20 GB RAM
Disk space	475 GB SSD
Display	15-inch Laptop Screen

4.2.2 Software Requirements

Table 4.2 Software requirements

Software component	Specification
Operating System	Windows 10 (64-bit)
Development Tool	Visual Studio Code Jupyter Notebook Google Colab
Development Platform	Microsoft Azure WayScript Heroku

4.2.2.1 Development Tool

4.2.2.1.1 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows users to create and share documents containing code, equations, visualizations and text. Over 40 programming languages are supported, including Python, R and more.

4.2.2.1.2 Visual Studio Code

Visual Studio Code is a lightweight and powerful cross-platform source code editor. In addition, it can run on Mac OS X, Windows and Linux.

4.2.2.1.3 Google Colab

Google Colab is a hosted Jupyter notebook service and provides free access to GPU/TPU computing resources. Besides that, programmed Colab notebooks can be stored on Google Drives.

CHAPTER 4: METHODOLOGY AND TOOLS

4.2.2.2 Development Platform

4.2.2.2.1 Microsoft Azure

Microsoft Azure is a cloud computing service operated by Microsoft. More than 200 products and cloud services are available on the Azure cloud platform.

4.2.2.2.2 WayScript

WayScript is a rapid scripting environment built for developers. Developers may use their preferred programming language to construct and operate apps in cloud, automate tasks, create internal tools, and integrate with databases and APIs.

4.2.2.2.3 Heroku

Heroku is a platform that provides users with the ability to rapidly deploy applications. It has the benefit of allowing developers to reduce the cost of building and maintaining the underlying system. In addition, to adapt the hardware and other resources required to the needs of the project.

4.3 User Requirements

- The dashboard must show all scraped computing job postings.
- The dashboard must provide variety filtering options such as job field, states and expected salary to filter certain jobs.
- The dashboard must provide different types of chart to illustrate the computing jobs market in Malaysia.
- The dashboard shall allow user to download the computing job postings.
- The dashboard shall allow user to download the chart.
- The dashboard shall allow user to further filter the jobs in data table or chart by job category.
- The dashboard shall allow user to further sort the jobs in data table by salary range.

CHAPTER 4: METHODOLOGY AND TOOLS

- The dashboard must show a description of the dashboard, such as purpose, services provided and features.
- The dashboard be synchronized to update the data in “Data Summary” and “Data Visualization” tabs to ensure that the data displayed on both sides is the same and correct.

4.4 Non-functional Requirements

- The user interface of dashboard must load within 5 seconds.
- The chart or images in dashboard must load within 5 seconds.
- The service call for the web application must respond within 5 seconds.
- The scripts of web scraping, data cleaning and analysis must be executed automatically at 1:00 am on the 12th and 26th of each month.

4.5 Verification Plan

4.5.1 Filtering Data

Table 4.3 Verification Plan for Filtering Data

Procedure Number	P1
Method	Testing
Applicable Requirements	The dashboard can update data and charts based on the filtering options selected by the user.
Purpose/Scope	To improve the user experience and the robustness and accuracy of the query system.
Item Under Test	Filtering Data
Precautions	The backend must successfully download the scraped data from Azure Blob Storage.
Limitations	None
Equipment/Facilities	Laptop
Data recording	None
Acceptance Criteria	The system must query the data based on the filtering options selected by the user and display accurate data and charts in the dashboard.
Procedure	<ol style="list-style-type: none">1. The dashboard provides 3 different filtering options, including jobs categories, job location and expected salary.2. Select different combinations of filtering options, which include the following 8 combinations:<ol style="list-style-type: none">I. No selection of any

CHAPTER 4: METHODOLOGY AND TOOLS

	<p>II. Select job category</p> <p>III. Select job location</p> <p>IV. Select expected salary</p> <p>V. Select job category and job location</p> <p>VI. Select job category and expected salary</p> <p>VII. Select job location and expected salary</p> <p>VIII. Select all filter options</p> <p>3. The query system can return the appropriate data according to the above 8 combinations of filtering options.</p> <p>4. The dashboard must update the chart based on the data returned by the backend.</p> <p>5. Conversely, if there are no query results, the dashboard must let the user know about it.</p>
Troubleshooting	Iterate this procedure
Post-Test Activities	None

4.5.2 Filtering and Sorting Job Postings in Data Table

Table 4.4 Verification Plan for Filtering and Sorting Job Postings in Data Table

Procedure Number	P2
Method	Testing
Applicable Requirements	The job posting in data table can be further filtering and sorting.
Purpose/Scope	To increase the interactivity of dashboard by further filter and sort job posting in the data table.
Item Under Test	Filtering and Sorting Job Postings in Data Table
Precautions	The backend must return at least one record of job posting to data table.
Limitations	None
Equipment/Facilities	Laptop
Data recording	None
Acceptance Criteria	The system must filter or sort the data based on the filtering or sorting options selected by the user and display accurate job posting in data table.
Procedure	<p>1. The data table can be filtering by job category and sorting by the salary range. There have 10 different jobs categories for filtering job postings, user can select one filtering option in each time. The sequence of job posting to display according to the ascending or descending of job salary.</p> <p>2. There are three criteria will be occurred in this data table:</p>

CHAPTER 4: METHODOLOGY AND TOOLS

	<ol style="list-style-type: none"> I. Filtering by job category first, then perform the sorting. II. Sorting by job range, then perform the filtering. III. Does not perform any actions. <p>3. The query system can return the appropriate data according to the above three criteria.</p> <p>4. If there are no query results returned, the data table will not show any content.</p>
Troubleshooting	Iterate this procedure
Post-Test Activities	None

4.5.3 Regenerate a New Chart when Further Filtering is Performed

Table 4.5 Verification Plan for Regenerate a New when Further Filtering is Performed

Procedure Number	P3
Method	Testing
Applicable Requirements	The chart can be regenerated when user need to filter by job category.
Purpose/Scope	To generate new insights in different perspectives by regenerating charts in different job categories.
Item Under Test	Regenerate a New Chart when Further Filtering is Performed
Precautions	Initial chart must be generated for further filtering and regenerate a new chart.
Limitations	None
Equipment/Facilities	Laptop
Data recording	None
Acceptance Criteria	The system must filter or sort the data based on the filtering or sorting options selected by the user and display accurate job posting in data table.
Procedure	<ol style="list-style-type: none"> 1. Some charts provide filtering by job category to regenerate a new chart. There have 10 different jobs categories for filtering, user can select one filtering option in each time. 2. The query system can return the appropriate data according to the filter options selected by user. 3. The backend based on the filtered result to regenerate a new chart.

CHAPTER 4: METHODOLOGY AND TOOLS

	4. If there are no query results returned, the chart will not be displayed and shows “No Results” message.
Troubleshooting	Iterate this procedure
Post-Test Activities	None

4.6 Implementation Issues and Challenges

Several issues and challenges were encountered while developing this project. The first issue is that the data scraped from the online recruitment platform has the problem of unbalanced classification. Unbalanced classification means that the distribution of the data in the classes is biased or skewed. This poses a challenge to build predictive modelling in machine learning and is unable to provide users with more comprehensive and reliable data statistics. For example, the data about computer science that scraped from JobStreet shows that Kuala Lumpur has 669 job records, but some states such as Pahang and Kelantan have less than 20 records. As a result, the data analysis and statistics in the preliminary dashboard are biased.

In addition, the proposed dashboard codes look messy and unmanageable at the moment. This is because all the code is programmed in one PY file, including the interface and logic layers of the dashboard. This is one of the limitations of developing dashboards using Dash framework as it is difficult to separate the code into different modules. Besides, Dash's official documentation does not give a solution to the forementioned problems, thus developers have to find their own solutions.

Another problem encountered when designing and developing the user interface of the proposed dashboard is choosing the appropriate chart to display the different types of data. For example, bar charts and pie charts are suitable for representing categorical data, while histogram and dot plots are used for numeric data. Hence, developing the user interface for the dashboard requires much effort and time in order to investigate the charts to be used to represent the analyzed data in order to create a better user experience.

The anti-crawling mechanism is one of the challenges of this project as it required scraping job posting automatically from Jobstreet and Indeed. The cyber law in Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 4: METHODOLOGY AND TOOLS

Malaysia does not explicitly state the web scraping is illegal, but almost well-established website has their own anti-crawler mechanism to prevent third party stealing their information without authorization. Indeed, has a well and dynamic anti-crawling mechanism that makes scraping data are challenging in this project. Their anti-crawling mechanism is changing the html structure of its site or the corresponding class IDs time to time, and above process is done automatically and dynamically. This makes it necessary to constantly modify the web scraping script according to the changes in Indeed's website.

After deploying the dashboard in Heroku, it was found that the response time of the dashboard was within 10 seconds, which is not a good performance compared to the response time must within 5 seconds that was set in the non-functional requirements. There are two reasons cause the longer response time which are the web application server and database were placed in different region, and the project uses server-side callback for implement the high interaction in dashboard.

Since Heroku is a free platform for deploying dashboards, it only offers a web application server option for Europe and the US. However, the Azure Blob Storage is set up in the region of Southeast Asia. This makes it necessary for the European or US web application server to connect to the blob storage in Southeast Asia in order to retrieve the required data when launching the dashboard. As a result, the above process takes a little time, but has no significant impact on the overall response time.

The second major factor causing longer response times is that the dashboard uses server-side callbacks instead of JavaScript to make the dashboard interactive. Server-side callbacks are one of the features of the Dash framework, which allows programmers to use python programming to implement complex functionality on the web. However, the disadvantage is that any event triggered by the user, such as filtering data, has to make a request to the web application server. The above mentioned process is very time consuming because any calculation has to be done by the server before the result can be returned to the user. Another alternative way is to user client-side callback like JavaScript which the server will return appropriate script to the client side. All calculation will be done by client side, this will reduce the response time.

CHAPTER 4: METHODOLOGY AND TOOLS

Although Plotly is a comprehensive open-source library for generating charts, a few minor faults have been found. For example, the data in the bar chart will not be displayed when the user keeps zooming in, so these minor errors are required to be resolved.

4.7 Timeline

4.7.1 – Timeline of the FYP1

Progress \ Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Program web scraping script	■	■												
Set up Azure Blob Storage		■												
Program automation web scraping and set up WayScript		■												
Data cleaning		■	■											
Prototype interface of dashboard			■											
Study Plotly's Dash for building a dashboard			■	■	■									
Program interface of dashboard				■	■									
Program the logic layer of the dashboard						■	■							
Visualize the data and shows in dashboard							■	■						

CHAPTER 4: METHODOLOGY AND TOOLS

Refactor code that program in FYP1														
Study NER and Machine Learning for data analysis														
Program the training model														
Completed the NER and classification model														
Testing final analysis result														
Finalize all function in dashboard														
Deploy dashboard and system testing														
Write report														
Complete and submit final report														
Presentation														

Figure 4.3 : Timeline of the FYP2

The main tasks to be completed in the starting timeline of FYP 2 include coding the web scraping script for Indeed, reprogram the data cleansing module, program the logic layer of the dashboard, and refactor code. Then, it is expected to spend a total of 6 weeks to learn the Named Entity Recognition (NER) and the machine learning to code the training model for the data analysis. The training models and all functions in the proposed dashboard have to be completed in week 10. Later, the dashboard can be deployed for web testing. This is followed by the FYP 2 report write up. Finally, the FYP 2 presentation and demonstration are to be conducted in between week 13 and 14.

CHAPTER 5: SYSTEM IMPLEMENTATION

5.1 WayScript

WayScript is a platform that provides visual programming to help developers to build software tools and automate workflows. This project will use WayScript to implement a backend that can automatically scrape data from a specified online recruitment platform within a certain period of time. Not only that, but the scraped data cleaning and analysis also will also be done on this platform. This is for the dashboard can provide the latest analytics to job seekers. The project will take a total of about 3 hours and 40 minutes to scrape for computer science and Information Technology (IT) data on JobStreet and Indeed. The data cleaning and analysis modules will complete all processes within 10 minutes. Therefore, WayScript's offer of 100 free hours a month of automation services was perfect for this project.

The following are the steps to WayScript setup:

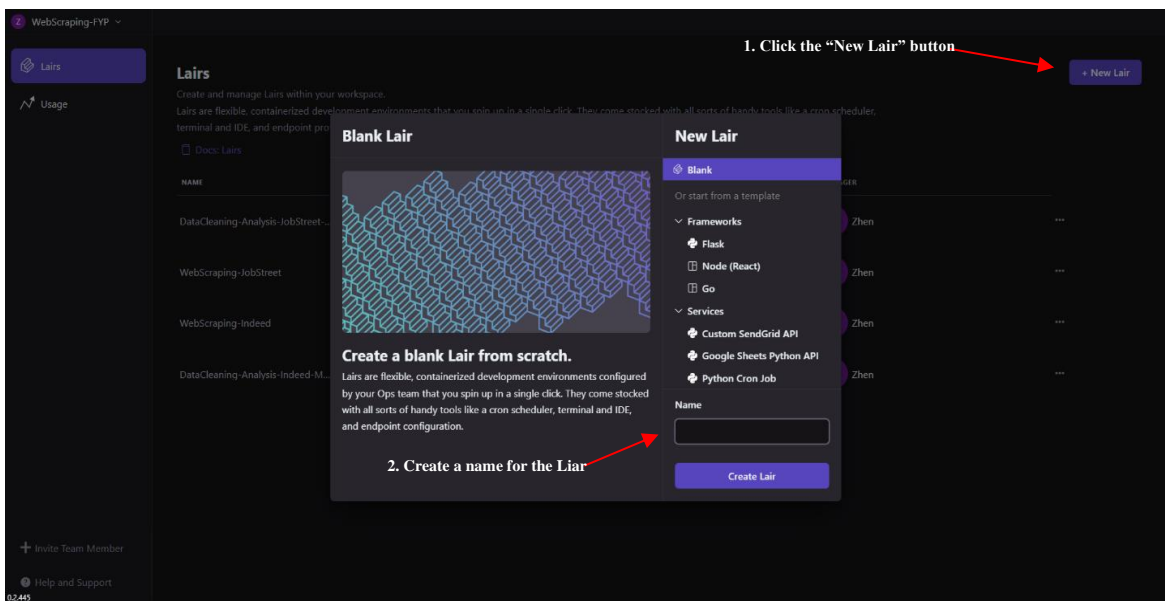


Figure 5.1 : Create Lair in the WayScript

The WayScript team has launched a new development environment called Lair, which offers a flexible and containerized development environment. Furthermore, the development environment comes with a variety of convenience tools such as an Integrated Development Environment (IDE), cron scheduler and endpoint configuration. The latest version of WayScript also provides Lair owners able to invite

CHAPTER 5: SYSTEM IMPLEMENTATION

other developers to contribute to the project. The first step is to click the “New Liar” button and create a name for the Liar as shown in Figure 5.1.

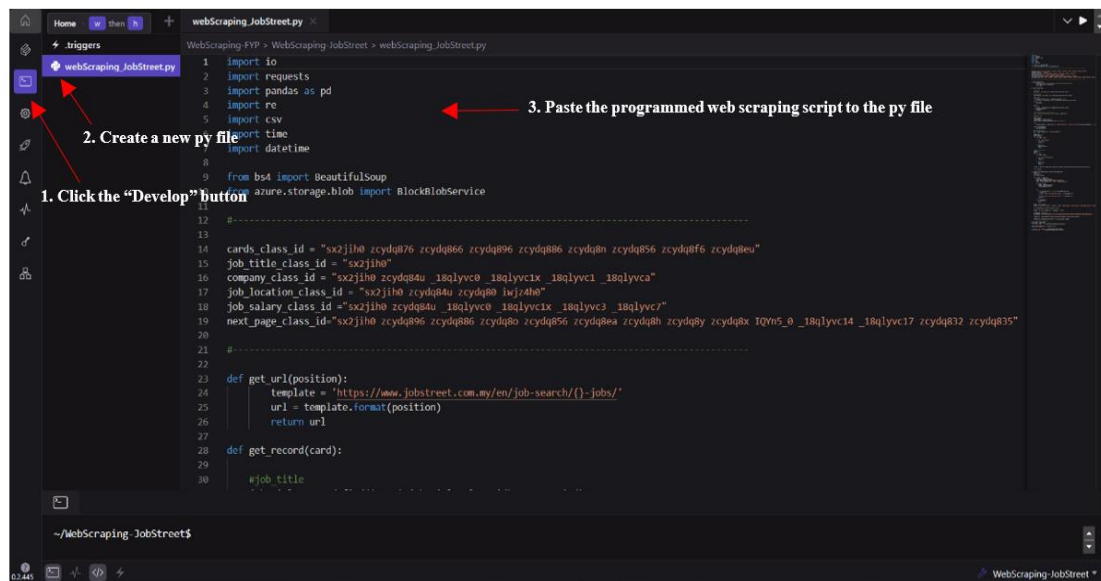


Figure 5.2 : Steps to start development

The following steps will be used as an example of deploying web scraping, which is the same as the deployment steps for the data cleaning and analysis module. Once Liar has been successfully created, the user will navigate to the development environment. The user needs to click on the second button named "Develop" in the left sidebar to start developing the project. After that, create a new py file named “py webScraping_JobStreet.py” and paste the programmed web scraping script into it. Figure 5.2 shows the steps mentioned above.

CHAPTER 5: SYSTEM IMPLEMENTATION

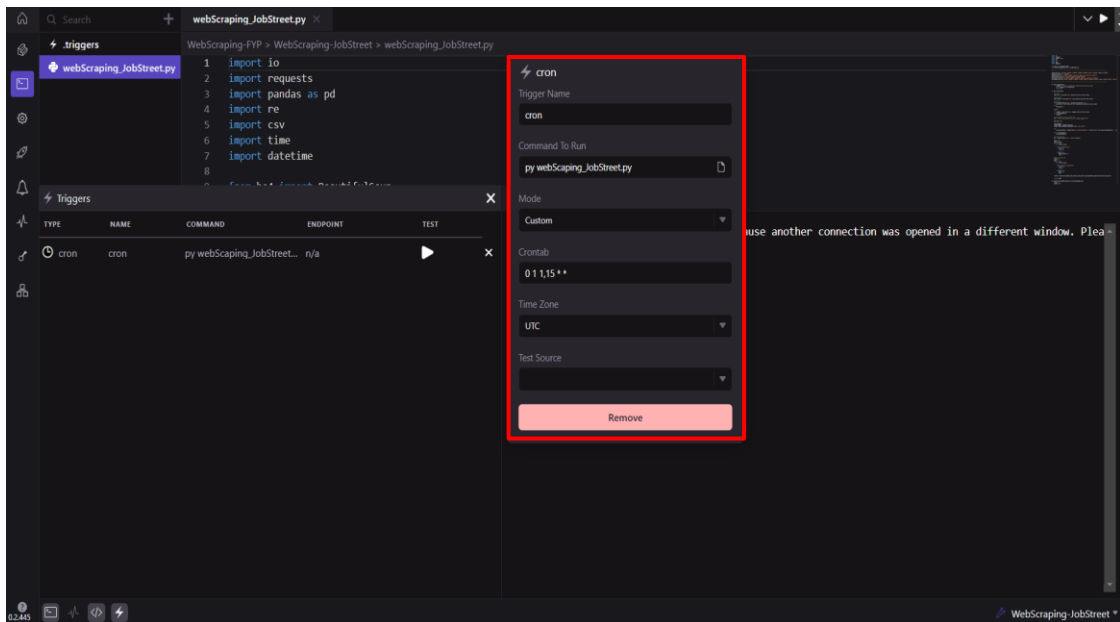


Figure 5.3 : Configuration of Cron

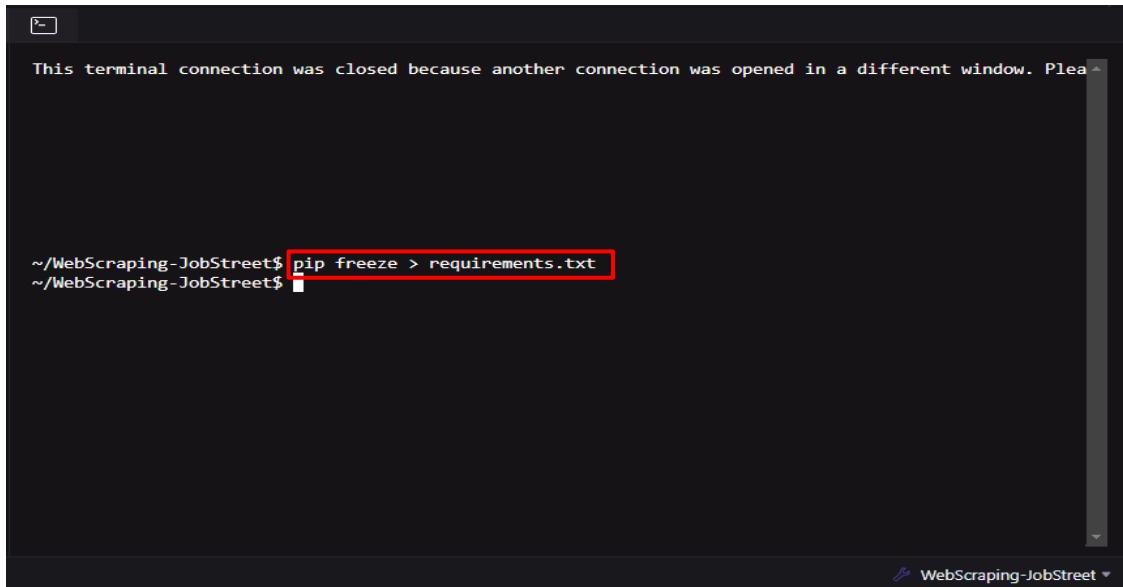


Figure 5.4 : Cron schedule expression

Select cron in the "triggers" section, as shown in Figure 5.3. The "Command To Run" section requires a properly formatted input (file_type file_name) to ensure that the particular file is executed. In this example, "py webScraping_JobStreet.py" is filled in the input box. Next, select "custom" to execute the script every time. The project plans that this data scraping script will be executed every two weeks. Therefore, insert "0 1 12,26 * *" in the crontab line. At the end of the setup, select UTC for the time zone option.

CHAPTER 5: SYSTEM IMPLEMENTATION

The command of "0 1 12,26 * *" is the cron schedule expression which means the script will be triggered at 1am on the 12th and 26th dates of each month. Figure 5.4 shows the meaning of this command.



```
This terminal connection was closed because another connection was opened in a different window. Plea ~  
  
~/WebScraping-JobStreet$ pip freeze > requirements.txt  
~/WebScraping-JobStreet$
```

Figure 5.5 : Create requirements.txt

Fill out the code shown in Figure 5.5 in the terminal to automatically generate requirements.txt. Requirements.txt will list all the packages required for this project.

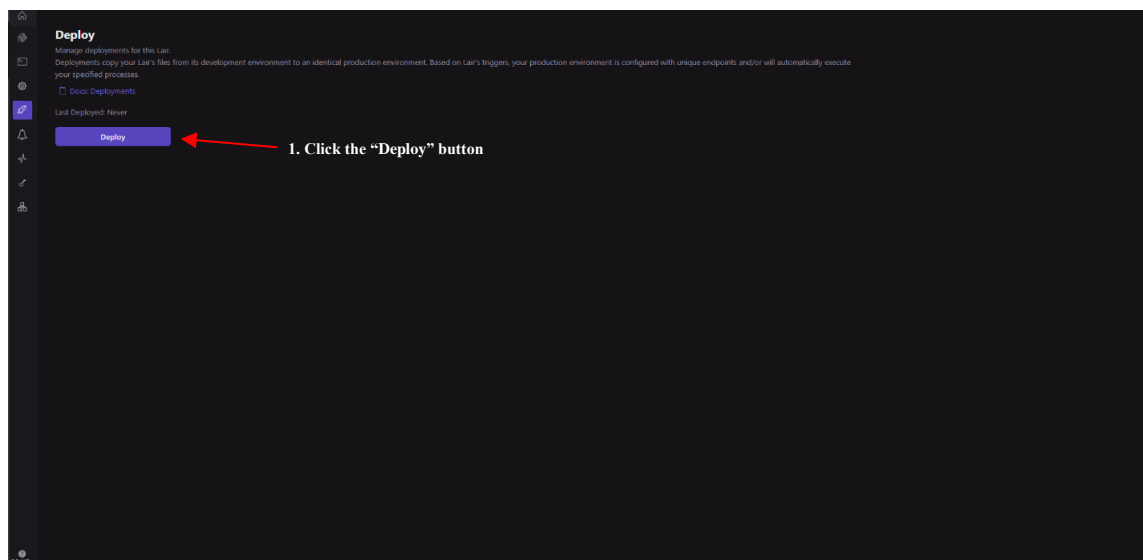


Figure 5.6 : Deploy the script

CHAPTER 5: SYSTEM IMPLEMENTATION

Clicking on the fifth button named "Deploy" on the left sidebar and will navigate to the final deployment page. Lastly, click the "Deploy" button as shown in Figure 5.6 to deploy the project that needs to be executed automatically to the cloud.

5.2 Azure Blob Storage

Microsoft Azure is a cloud computing platform that offers a variety of cloud services, including computing, storage, networking, and analytics. Azure Blob Storage is a cloud storage service from Microsoft Azure. It mainly stores unstructured data, including text, images or binary data. The data scraped by this project on the target online recruitment platform is stored in csv format. Therefore, Azure Blob Storage is well suited as the back-end storage data platform for this project.

The figure below shows the step to setup the Azure Blob Storage:

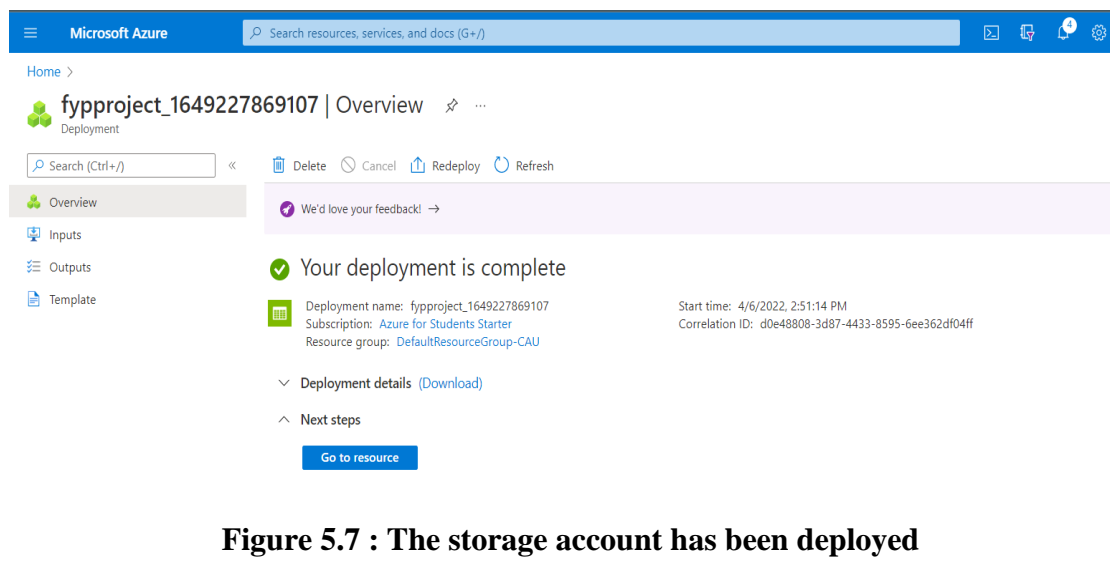


Figure 5.7 : The storage account has been deployed

A storage account needs to be created. For example, the account created here is called fyproject. The purpose of creating a storage account is to allow the user to select different data stores, including containers, file shares, queues and tables. Figure 5.7 shows the storage account has been deployed.

CHAPTER 5: SYSTEM IMPLEMENTATION

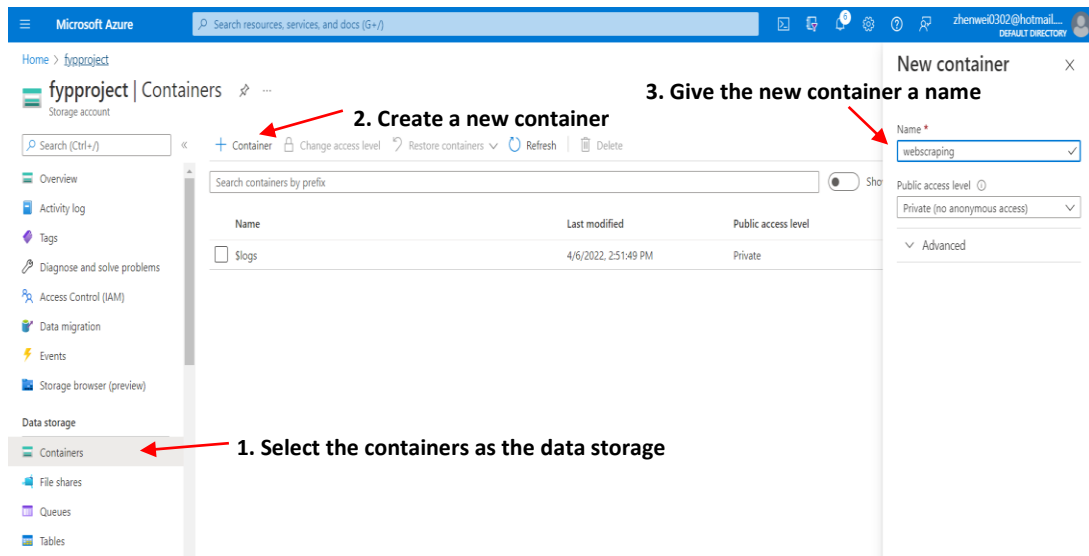


Figure 5.8 : Create a new container

Navigate to the storage account that has been created and search for the data storage in the right sidebar and select containers. After that, create a new container to store the scraped data. The name of the container created here is webscraping. Figure 5.8 shows the steps for creating a new container.

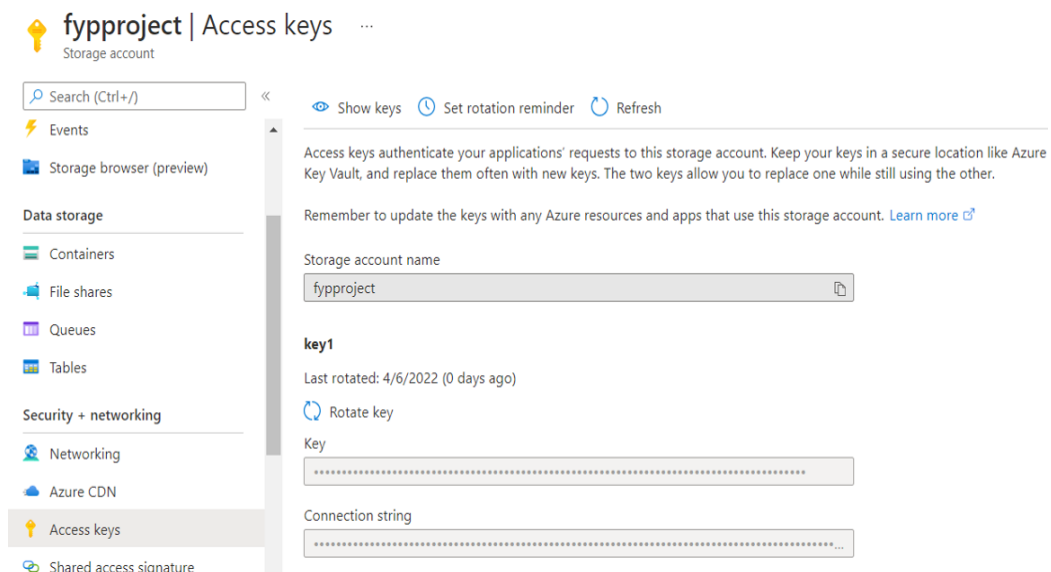


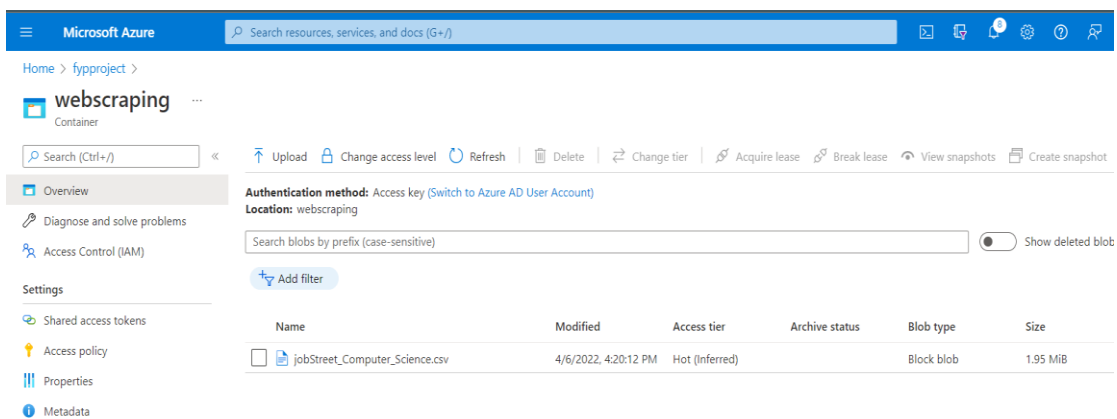
Figure 5.9 : Access Key

Later, copy key 1 from the access key's page. The purpose of the access key is for the data scraping script to access the specified storage account and container to store the latest scraped data. Figure 4.10 shows the access key's page.

CHAPTER 5: SYSTEM IMPLEMENTATION

```
130     output = io.StringIO()
131     title = ['JobTitle', 'Company', 'Location', 'Salary', 'Requirements', 'Qualification', 'YearsofExperience']
132
133     df = pd.DataFrame(records, columns = title)
134
135     output = df.to_csv(index=False, encoding = "utf-8")
136
137     accountName = "fypproject"
138     accountKey = "0000000" #paste the access key - key 1
139
140     blobService = BlockBlobService(account_name=accountName, account_key=accountKey)
141
142     blobService.create_blob_from_text('webscraping', 'jobStreet_Computer_Science.csv', output)
143     # blobService.create_blob_from_text(container_name, file_name, content)
```

Figure 5.10: Placing the access key in the web scraping



The screenshot displays the Microsoft Azure portal interface for a storage account named 'fypproject'. The selected container is 'webscraping'. The authentication method is 'Access key'. The location is 'webscraping'. A search bar for blobs is present. A table lists the blobs in the container:

Name	Modified	Access tier	Archive status	Blob type	Size
jobStreet_Computer_Science.csv	4/6/2022, 4:20:12 PM	Hot (Inferred)		Block blob	1.95 MiB

Figure 5.11 : The scraped data stored in “webscraping” blob

Figure 5.10 shows placing the access key of the storage account in the web scraping script. The 140th line of code assigns parameters about the name and access key of the storage account to the built-in function of the BlockBlobService. This allows the specified Azure storage account to be connected. Then, line 142th line of the code is about stores the scraped data in a container named webscraping and the uploaded file name is jobStreet_Computer_Science.csv. While Figure 5.11 shows that the latest scraped data is uploaded to Azure Blob Storage.

5.3 Web Scraping

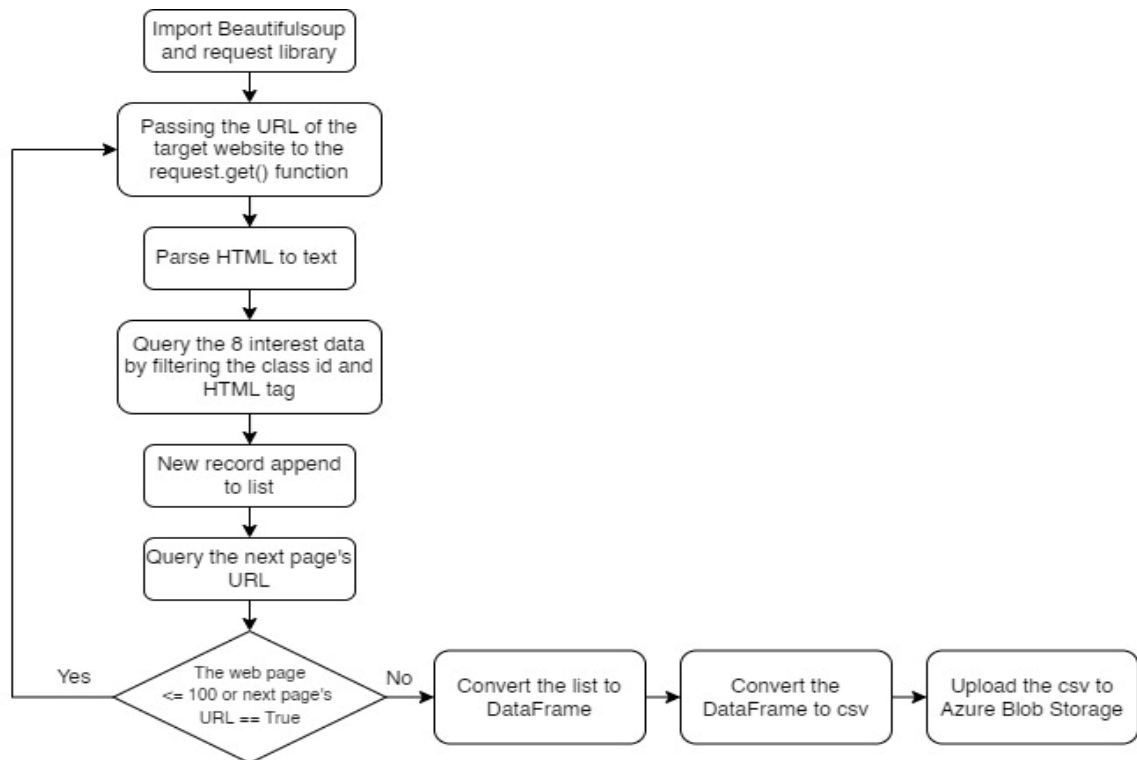


Figure 5.12 : The flowchart of the web scraping

Figure 5.12 shows the flowchart of the web scraping. This project on the focus is to scrap JobStreet and Indeed data on computer science and IT.

The entire web scraping program requires to import BeautifulSoup and request library. First of all, assign the link of the website that needs to be scraped to the “get” method of the request library. The get method will return a response object. Later, the html content in the response object is parsed as text to facilitate querying the targeted data. This project focuses on scraping 8 different types of data which include job title, company, location, salary, job requirements, qualifications, years of relevant job experience and application link. The data to be scraped above are embedded with the corresponding class IDs and HTML tags. Therefore, the web scraping program is required to query the class id and HTML tag of the data and store the targeted data into the corresponding variable. Then, the 8 different variables will be appended to a list which mean that the information of a job posting was successfully stored in the list. After that, the program will start querying the next page’s URL.

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 5: SYSTEM IMPLEMENTATION

Next, the program will check if the 100th page is scraped. This project only scrapes the first 100 pages of JobStreet, because the information on the job posting after 100 pages are incomplete and outdated. If the program has not scraped 100 pages of job posting yet, it will start querying the next page's URL and assign it back to the get method of the request library to scrape the next page of data. However, the web scraping process only will be terminated if the URL of the next page is not found in Indeed. This means that the Indeed's web scraping script has no restrictions for scraping pages.

The scraped data stored in the list is converted into a DataFrame when the program has completed the process of scraping the data. As the final format to be stored is csv, the DataFrame is converted to csv format and uploaded to Azure Blob Storage.

5.4 Data Cleaning

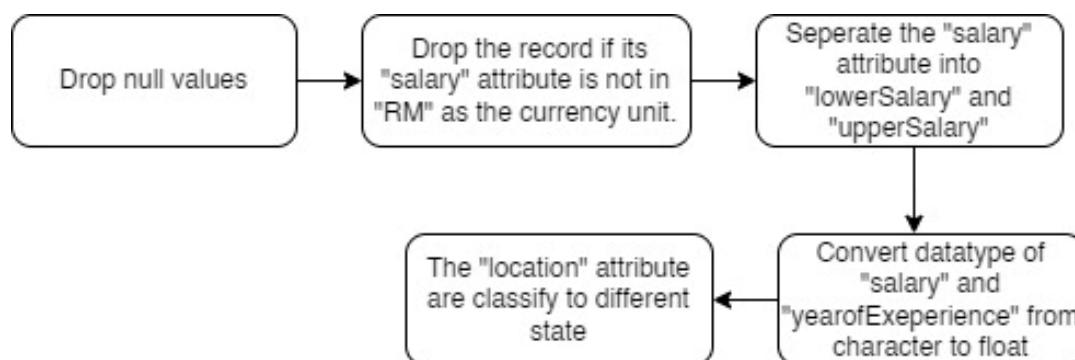


Figure 5.13 : The process of the data cleaning

The scraped data was obtained from two different online job recruitment platforms. Therefore, data cleaning enables the standardization of scraped data, including the data type and the format in which the data is presented. Data cleaning is also responsible for filtering out noisy or invalid data in order to produce more accurate information during data analysis.

Figure 5.13 shows the process of data cleaning. The first step in data cleaning is to drop the null values. The records will be dropped while has null values for job title, location, salary and job requirements. In addition, the salary attribute in the record does not

CHAPTER 5: SYSTEM IMPLEMENTATION

appear in "RM" as the currency unit will be removed, which is to ensure that the data scraped is in Malaysia. Since the salary in JobStreet has a lower and upper range, the data cleaning process will split this salary data format into two separate attributes. Also, convert the data type of salary and years of experience from character to float.

The final data cleaning is according to the job location to classify to different state. The backend has a total 3891 records about different location and its state to classify the job location attribute in scraped data. The purpose to classify the location to different state is because the state as a filter options in dashboard. Besides that, the program will sort and group the references data and the job location data by the first letter of the alphabet. For example, "C" is the first letter of Cyberjaya, so the searching algorithm only search the reference group that first letter is "C". This proposed search method will improve the efficiency of classifying the job location into different states instead of searching sequentially.

CHAPTER 5: SYSTEM IMPLEMENTATION

5.5 Data Analysis

There are two parts of this project that needed further analysis to generate meaningful insights. The first is to categorize all the job postings into 10 different computing jobs categories so that users could search for the job postings they are interested in by jobs categories on the dashboard. Furthermore, the program needs to identify the Information and Communications Technology (ICT) skills mentioned in each job description to analysis the ICT skills needed by the job seeker for the specific job. The classification of computing job will implement the machine learning. Since identifying ICT skills in job description texts is a natural language processing, the Named-Entity Recognition (NER) techniques will be implemented in this project .

5.5.1 Multiclass Classification

5.5.1.1 Pre-Processing

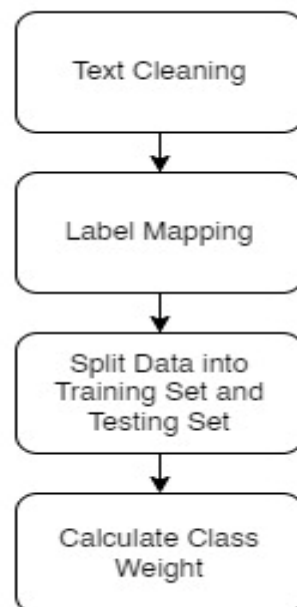


Figure 5.14 : The preprocessing of the training model

Figure 5.14 shows the preprocessing of the training model. There are two attributes in the training data, including the job title and its computing job category. The job category including manager, information systems engineer, developer/programmer, IT analyst, IT support, internship/trainee, others, IT quality control/asurance, IT architect, IT educator.

CHAPTER 5: SYSTEM IMPLEMENTATION

The first step of the preprocessing is the text cleaning of the job title, which includes filtering symbols, numbers, and lowercase letters. This is because text cleaning is very helpful to improve the accuracy of the training model. Since the training model cannot directly classify job titles in string format, it is necessary to map 10 different computer job categories as integers from 0 to 9, such as "0" for managers and "9" for IT educators. Besides that, the data will be divided into 80% for training and 20% for testing.

The training data has severe data imbalance, such as the manager sample has 1803 records, while the IT educator has only 43 records. Therefore, the `class_weight` function is used to solve the data imbalance problem. Classes with fewer records will be assigned more weight, which means the training model must focus more on these classes. Figure 5.15 shows the weights for each class, such as label 9 indicates that the IT Educator class received a weight of 11.45.

```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.utils import class_weight
    import numpy as np

    X = df.JobTitle
    y = df.job_label_code
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 1050)

    class_weight = class_weight.compute_class_weight('balanced',
                                                    classes = np.unique(y_train),
                                                    y= y_train)

    class_weight = dict(enumerate(class_weight.flatten(), 0))

    for label,weight in class_weight.items():
        print(label,weight)

0 0.284472049689441
1 0.44804347826086954
2 0.568551724137931
3 1.543820224719101
4 1.5613636363636363
5 2.0923857868020304
6 2.264835164835165
7 9.16
8 11.14054054054054
9 11.45
```

Figure 5.15 : The weights for each class

CHAPTER 5: SYSTEM IMPLEMENTATION

5.5.1.2 Training Model

This project will use different training models and compare the accuracy of their results on predictive testing and evaluation datasets. The training model include Logistic Regression [20], Naïve Bayes Classifier [21], XGBoost [23] and Support Vector Machines (SVM) [25] algorithms. In addition, a total of 5153 training data were used to for training using the models mentioned above.

The following is the overview of each training model:

5.5.1.2.1 Logistic Regression

A multi-class Logistic Regression method is proposed to transform the multi-classification problem into a binary classification problem [20]. Multi-class Logistic Regression method is proposed to identify all kinds of faults accurately [20].

5.5.1.2.2 Naïve Bayes Classifier

A Naïve Bayes classifier is a simple probabilistic based method, which can predict the class membership probabilities [21]. Naïve Bayes classifier can easily handle missing attribute values by simply omitting the corresponding probabilities for those attributes when calculating the likelihood of membership for each class [22].

5.5.1.2.3 XGBoost

XGBoost is an ensemble machine learning system combining a series of decision trees of which each learns from the prior one and influences the next one [23]. It improves the traditional gradient boosting decision tree (GBDT) algorithm with respect to computing speed, generalization performance, and scalability [24].

5.5.1.2.4 Support Vector Machines (SVM)

SVMs are classification prediction tools that use Machine Learning theory as a principled and very robust method to maximize predictive accuracy for detection and classification [25]. SVM classification is based on the idea of decision hyperplanes that determine decision boundaries in input space or high dimensional feature space [25].

CHAPTER 5: SYSTEM IMPLEMENTATION

Grid search will be used for all training models to tune the accuracy of the model. The grid search loops over all the candidate hyperparameters and returns the best performing hyperparameters for the model. Table 5.1 shows the description of each candidate hyperparameter. Table 5.2 shows the hyperparameters used and final accuracy of each training model in the test data, with Logistic Regression achieves the highest accuracy as compared to other mentioned models.

While Figure 5.16 shows the hyperparameters returned by the grid search for the logistic regression model. Moreover, Figure 5.17 shows that the accuracy of Logistic Regression is 0.989 after using the hyperparameters returned by the grid search, while the accuracy of the original model is only 0.945.

Table 5.1 The Description of Each Hyperparameter

Hyperparameter	Description
C value	C value is the inverse of regularization strength. The larger value of C specify lower regularization, which means that the algorithm of the training model is more prone to overfit.
Alpha	Alpha is a vector weights of training instances. The higher value of alpha, which means the training instance are more importance for the model.
Penalty	L1 and L2 will be candidate hyperparameter in the penalization. The penalty is to deduce the weights or coefficients of unimportant variables. This is to ensure that the training model is not complicated and solve the overfitting problem, which is also the method of regularization. L1: The regularization method called Lasso Regression, penalizes the total of the weights' absolute values. L2: The regularization method called Ridge Regression, penalizes the sum of squares of the weights.
Solver	The solver is to find the weight of parameter that minimize the cost function. There has 3 solvers have been proposed, including newton-cg, lbfgs and liblinear. newton-cg: This is the newton method which implement the Hessian matrix. lbfgs: Limited memory Broyden Fletcher Goldfarb Shanno (lbfgs), which uses the gradient evaluations to approximate the second derivative matrix updates.

CHAPTER 5: SYSTEM IMPLEMENTATION

	liblinear : Library for Large Linear Classification (liblinear), which implement the coordinate decent algorithm.
vect_ngram_range	The ngram will extract the words in the data based on the lower bound and the boundary of the n-value range. For example, ngram_range(1,1) means unigrams.
tfidf__use_idf	Inverse Document Frequency (IDF) is used to identify the common words in the data, minimizing the weight of frequently occurring words such as "of", "as", and "the".

Table 5.2 The Hyperparameter used and Accuracy of Each Training Model

Model Name	Hyperparameter	Accuracy
Logistic Regression	c : 1000.0 penalty : l1 clf_slover : liblinear vect_ngram_range : (1,1) tfidf_use_idf : True	0.989
Naïve Bayes Classifier	alpha : 0.01 vect_ngram_range : (1,2) tfidf_use_idf : False	0.872
XGBoost	alpha : 0.01 vect_ngram_range : (1,1) tfidf_use_idf : False	0.966
Support Vector Machines (SVM)	alpha : 0.001 clf_penalty : l2 clf_slover : liblinear vect_ngram_range : (1,1) tfidf_use_idf : False	0.982

CHAPTER 5: SYSTEM IMPLEMENTATION

Grid Search

```
[ ] from sklearn.model_selection import GridSearchCV
import warnings

warnings.filterwarnings('ignore')

params={
    'clf_penalty' : ['l1','l2'],
    'clf_C'       : np.logspace(-3,3,7),
    'clf_solver'  : ['newton-cg', 'lbfgs', 'liblinear'],
    'vect_ngram_range': [(1, 1), (1, 2)],
    'tfidf_use_idf': (True, False),
}

grid = GridSearchCV(lr,          # model
                   param_grid = params, # hyperparameters
                   scoring='accuracy', # metric for scoring
                   cv=5,          # number of folds
                   n_jobs=-1,)

grid.fit(X_train,y_train)

print("Tuned Hyperparameters :", grid.best_params_)

Tuned Hyperparameters : {'clf_C': 100.0, 'clf_penalty': 'l1', 'clf_solver': 'liblinear', 'tfidf_use_idf': True, 'vect_ngram_range': (1, 1)}
```

Figure 5.16 : Grid search returns well-performed hyperparameters

↳ Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn import metrics
import warnings

warnings.filterwarnings('ignore')

lr = Pipeline([('vect', CountVectorizer(stop_words='english')),
              ('tfidf', TfidfTransformer()),
              ('clf', LogisticRegression(class_weight = class_weight)),
              ])

lr.fit(X_train,y_train)
y_pred = lr.predict(X_test)

print("Logistic Regression(without GridSearch) Accuracy:", metrics.accuracy_score(y_test, y_pred))

#-----

lr_gs = Pipeline([('vect', CountVectorizer(stop_words='english',ngram_range = (1,1))),
                 ('tfidf', TfidfTransformer(use_idf = True)),
                 ('clf', LogisticRegression(class_weight = class_weight,
                                           C = 100.0,
                                           penalty = 'l1',
                                           solver = 'liblinear'))),
                 ])

lr_gs.fit(X_train,y_train)
y_pred_gs = lr_gs.predict(X_test)

print("Logistic Regression (with GridSearch) Accuracy:", metrics.accuracy_score(y_test, y_pred_gs))

Logistic Regression(without GridSearch) Accuracy: 0.944713870029098
Logistic Regression (with GridSearch) Accuracy: 0.9893307468477207
```

Figure 5.17 : The accuracy of Logistic Regression

CHAPTER 5: SYSTEM IMPLEMENTATION

5.5.1.3 Evaluation

In the final stage, the accuracy of all training models in the evaluation data will be compared. The evaluation data has a total of 666 different jobs and is a completely new data to test the accuracy of the training models to detect whether the training models will be overfitted and underfitted. Figure 5.18 shows the accuracy of each training model in the evaluation data. The accuracy of Logistic Regression is 0.985, which is the best performance among all training models.

There are several reasons why Logistic Regression has the best performance compared to other above mentioned models. First, Logistic Regression is less prone to overfitting when training low-dimensional datasets. Low-dimensional datasets are specified by a small number of classes in the data. For example, the training data in this model has only 10 computing jobs categories for job classification. In addition, the penalized hyperparameters mentioned above also can avoid overfitting. Logistic Regression allows the training model to be easily updated where the classification of unknown data is performed fast.

Therefore, in this work, Logistic Regression will be used as a model for multiclass classification.

CHAPTER 5: SYSTEM IMPLEMENTATION

▼ Evaluation

```
[ ] dff = pd.read_csv('evaluation_data.csv')

label_map = {
    'Manager': 0,
    'Information Systems Engineer': 1,
    'Developer/Programmer': 2,
    'IT Analyst': 3,
    'IT Support': 4,
    'Internship/Trainee': 5,
    'Others': 6,
    'IT Quality Control/Assurance': 7,
    'IT Architect': 8,
    'IT Educator': 9,
}

dff['job_label_code'] = dff['job_label'].map(label_map)
dff['job_label_code'] = dff.job_label_code.astype(int)

X = dff.JobTitle
Y = dff.job_label_code

y_pred_svm_gs = clf_svm_gs.predict(X)
y_pred_lr_gs = lr_gs.predict(X)
y_pred_nb_gs = naivebayes_gs.predict(X)
y_pred_xgb_gs = xgboost_gs.predict(X)

print("    Model Name      |      Evaluation's Accuracy\n")
print("1) Logistic Regression      ", metrics.accuracy_score(Y, y_pred_lr_gs))
print("2) SVM                        ", metrics.accuracy_score(Y, y_pred_svm_gs))
print("3) Xgboost Classifier         ", metrics.accuracy_score(Y, y_pred_xgb_gs))
print("4) Naive Bayes Classifier     ", metrics.accuracy_score(Y, y_pred_nb_gs))
```

Model Name	Evaluation's Accuracy
1) Logistic Regression	0.984984984984985
2) SVM	0.9819819819819819
3) Xgboost Classifier	0.9654654654654654
4) Naive Bayes Classifier	0.8408408408408409

Figure 5.18 : The accuracy of each training model in the evaluation data

CHAPTER 5: SYSTEM IMPLEMENTATION

5.5.2 Named-Entity Recognition (NER)

Named Entity Recognition (NER) is one of the fundamental tasks in Natural Language Processing (NLP) [26]. The main goal of NER is to process the unstructured text and classify it into relevant entities such as people, organizations, currencies, etc. For example, UTAR is classified as an organization rather than people or other entity. The NER model is trained in supervised machine learning approach and is trained on an annotated dataset. At the end, a trained statistical NER model will be delivered to predict and assign the exact entity to any unannotated datasets. The spaCy is an open-source software library for advanced natural language processing and will be used to train a custom entity recognition model to identify all ICT skills.

5.5.2.1 Pre-Processing

The source of the training dataset is from Kaggle and the dataset named "StackSample: 10% Stack Overflow quizzes", which contains 10% questions and answers from the Stack Overflow website. The "Question.csv" will be selected as the training data from this dataset, which has 1.3 million data. The "Title" attribute of the training data has listed different ICT skills, which very appropriate for training this NER model.

After data cleaning, a total of 46,508 records were used as the final training data, with 20% of the data used for testing and the rest for training. Then, the ICT skills of each record were annotated using the `Matcher()` function. The following is the annotation format that all records should follow:

(data, {"entities":[{"initial position of the entity, end position of the entity, entity name}]})

Figure 5.19 shows an example of such a data annotation.

```
[ ] print(DATA[0])
```

```
("Kite also plans to add support for Google's Go programming language in the future.", {'entities': [(44, 46, 'ICT_SKILLS')]})
```

Figure 5.19 : Data annotation

CHAPTER 5: SYSTEM IMPLEMENTATION

5.5.2.2 Training NER Model

The latest version of spaCy provides developers to train the NER models through the spaCy command line interface (CLI) rather than program complex module for training. Developers simply provide the CLI with a configuration file, training data and a test dataset to start training the model. It improves training efficiency and reduces development costs.

There are several parameter's values were changed in the configuration file to fine tune the hyperparameter, including dropout, learning rate and the evaluation's frequency. The dropout was set to 0.2 to discard a portion of the neural network units which can reduces overfitting. In this [27] it mentioned that, using smaller dropout value of 0.2 is a good starting point to build a training model [27]. In addition, in this work, the NER model has a learning rate of 0.001 and the default optimizer used is Adam optimizer. According to [28], the Adam optimizer with learning rates of 0.001 performed well in observing the relationship between training time and model size for multiple learning rates [28]. Moreover, the learning rate is one of the key hyperparameters to determines whether the neural network will converge to a global minimum. Furthermore, developers shall adjust the desired evaluation frequency for viewing the evaluation results earlier or later. The evaluation frequency is set to 128 which specifies the CLI should display the evaluation's result every 128 steps , as shown by the "#" attribute in Figure 5.20.

The spaCy will keep training the NER model until the loss is minimized and have a high accuracy. As shown in Figure 5.20, the loss of the training model is decreasing at each epoch, which is a good sign that the model is learning from the data.

Precision, recall and f-score were used as evaluation measures. The precision reflects the ability of the model to identify negative samples, while recall is to identify the positive samples [29]. The f1-score is the combination of the precision and recall. The higher the f-score, the more robust the model is [29]. .Figure 5.21 shows the results of this trained NER model on the testing data with a precision of 99.82, a recall of 99.83, and an f1-score of 99.82.

CHAPTER 5: SYSTEM IMPLEMENTATION

```

✓ [20] |python -m spacy train config.cfg --output ./output --paths.train ./train.spacy --paths.dev ./test.spacy
2022-09-03 07:52:44.662022: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA
✓ Created output directory: output
i Saving to output directory: output
i Using CPU

===== Initializing pipeline =====
[2022-09-03 07:52:45.402] [INFO] Set up nlp object from config
[2022-09-03 07:52:45.413] [INFO] Pipeline: ['tok2vec', 'ner']
[2022-09-03 07:52:45.418] [INFO] Created vocabulary
[2022-09-03 07:52:45.420] [INFO] Finished initializing nlp object
[2022-09-03 07:54:36.049] [INFO] Initialized pipeline components: ['tok2vec', 'ner']
✓ Initialized pipeline

===== Training pipeline =====
i Pipeline: ['tok2vec', 'ner']
i Initial learn rate: 0.001
E # LOSS TOK2VEC LOSS NER ENTS_F ENTS_P ENTS_R SCORE
-----
0 0 0.00 480.83 0.00 0.00 0.00 0.00
0 128 257.32 14687.20 96.97 97.26 96.69 0.97
0 256 174.43 1494.94 98.51 98.46 98.56 0.99
0 384 130.05 1009.66 98.87 98.67 99.06 0.99
0 512 107.93 794.75 99.19 99.29 99.09 0.99
0 640 110.98 749.68 99.39 99.32 99.46 0.99
0 768 118.28 718.11 99.38 99.16 99.60 0.99
0 896 118.32 746.48 99.60 99.60 99.61 1.00
0 1024 119.19 696.30 99.67 99.69 99.65 1.00
0 1152 122.40 623.40 99.71 99.77 99.65 1.00
0 1280 129.67 640.07 99.73 99.73 99.72 1.00
0 1408 103.32 518.00 99.75 99.74 99.76 1.00
0 1536 99.29 509.40 99.73 99.70 99.75 1.00
1 1664 88.51 375.67 99.73 99.73 99.73 1.00
1 1792 97.78 364.68 99.77 99.83 99.71 1.00
1 1920 91.05 395.67 99.75 99.75 99.76 1.00
1 2048 108.78 381.78 99.78 99.77 99.79 1.00
1 2176 97.16 376.98 99.67 99.56 99.79 1.00
1 2304 77.24 343.90 99.81 99.84 99.78 1.00
1 2432 84.23 318.71 99.75 99.65 99.85 1.00
1 2560 91.65 320.54 99.77 99.76 99.77 1.00
2 2688 91.28 324.83 99.82 99.84 99.80 1.00
2 2816 95.68 256.96 99.79 99.80 99.78 1.00
2 2944 128.96 232.08 99.81 99.85 99.78 1.00
2 3072 113.27 246.77 99.81 99.81 99.81 1.00
2 3200 113.85 282.17 99.83 99.84 99.81 1.00
2 3328 131.11 263.67 99.83 99.86 99.80 1.00
2 3456 124.58 263.04 99.82 99.81 99.82 1.00
2 3584 148.38 233.57 99.82 99.85 99.80 1.00
2 3712 132.43 236.43 99.79 99.73 99.85 1.00
3 3840 163.12 227.02 99.80 99.76 99.84 1.00
3 3968 187.93 176.66 99.81 99.79 99.83 1.00
3 4096 177.88 194.87 99.82 99.84 99.81 1.00
3 4224 203.04 195.62 99.81 99.78 99.84 1.00
3 4352 208.28 186.98 99.75 99.63 99.87 1.00
3 4480 226.80 209.36 99.82 99.81 99.83 1.00
3 4608 209.68 160.50 99.81 99.83 99.80 1.00
3 4736 291.95 199.42 99.81 99.79 99.83 1.00
3 4864 255.70 173.25 99.82 99.81 99.83 1.00
✓ Saved pipeline to output directory
output/model-last

Note:
1) E : Epoch
2) # : Steps in the training model
3) LOSS TOK2VEC: The loss values for the token-to-vector
4) LOSS NER: The loss values for the named entity recognition
5) ENTS_F: F1-Score
6) ENTS_P: Precision
7) ENTS_R: Recall
8) SCORE: The overall source of the pipeline

```

Figure 5.20 : NER training pipeline

CHAPTER 5: SYSTEM IMPLEMENTATION

```
[ ] !python -m spacy evaluate /content/output/model-last test.spacy

2022-09-03 09:10:59.618116: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] t
i Using CPU

===== Results =====

TOK      100.00
NER P    99.82
NER R    99.83
NER F    99.82
SPEED    26406

===== NER (per type) =====

          P      R      F
ICT_SKILLS  99.82  99.83  99.82
```

Figure 5.21 : The precision, recall and f1-score of the NER model

5.5.2.3 Evaluation of the NER Model

A total of 3000 newly records in evaluation data were used to evaluate the NER training model. The evaluations results of the model was also satisfactory, with Figure 5.22 showing a precision of 97.85, a recall of 98.85, and an f1-score of 98.35.

```
[ ] !python -m spacy evaluate /content/output/model-last evaluation.spacy

i Using CPU

===== Results =====

TOK      100.00
NER P    97.85
NER R    98.85
NER F    98.35
SPEED    22830

===== NER (per type) =====

          P      R      F
ICT_SKILLS  97.85  98.85  98.35
```

Figure 5.22 : The precision, recall and f1-score of the NER model in the evaluation data

CHAPTER 5: SYSTEM IMPLEMENTATION

5.6 Dashboard

5.6.1 Overview of Dashboard

The dashboard for this project was built from the Dash framework and utilizes Plotly to visualize the data. Figure 5.23 and Figure 5.24 show the user interface of the dashboard. The entire user interface of dashboard consists of 3 components, which are the navigation bar, the data summary, and the data visualization. The analysis of the data is distinguished by two tabs for data summary and data visualization, which are designed for users to have different viewpoint to analyze the data. The navigation bar has provided a modal called “About Dashboard” and settings. Figure 5.25 shows the modal named “About Dashboard”, which is designed to provide the user with more information based on a description of the purpose and function of the dashboard. In addition, the settings are to allow the user to select multiple filtering options to analyze the interested data.

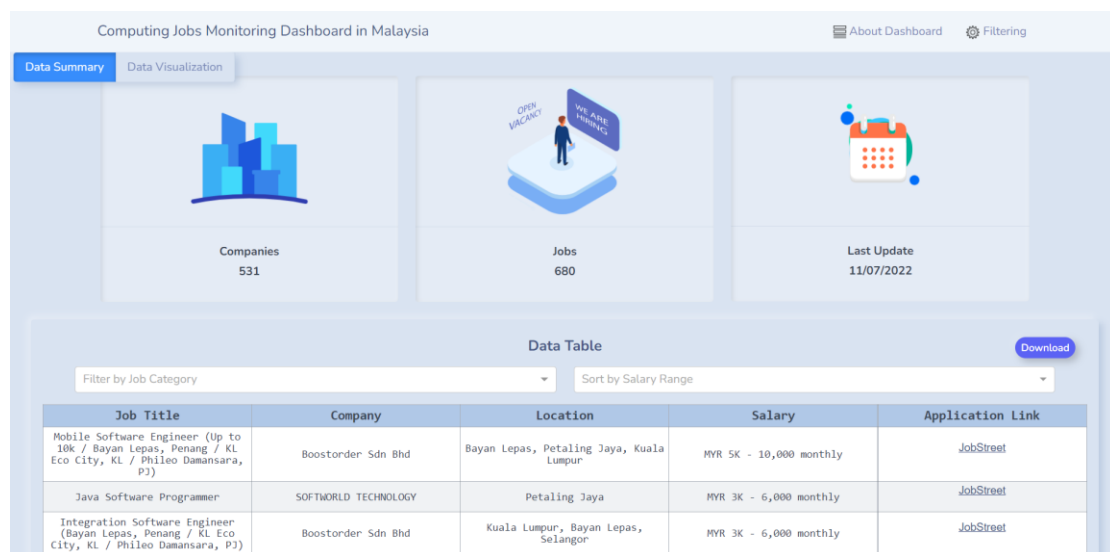


Figure 5.23 : Overview of the dashboard (1)

CHAPTER 5: SYSTEM IMPLEMENTATION

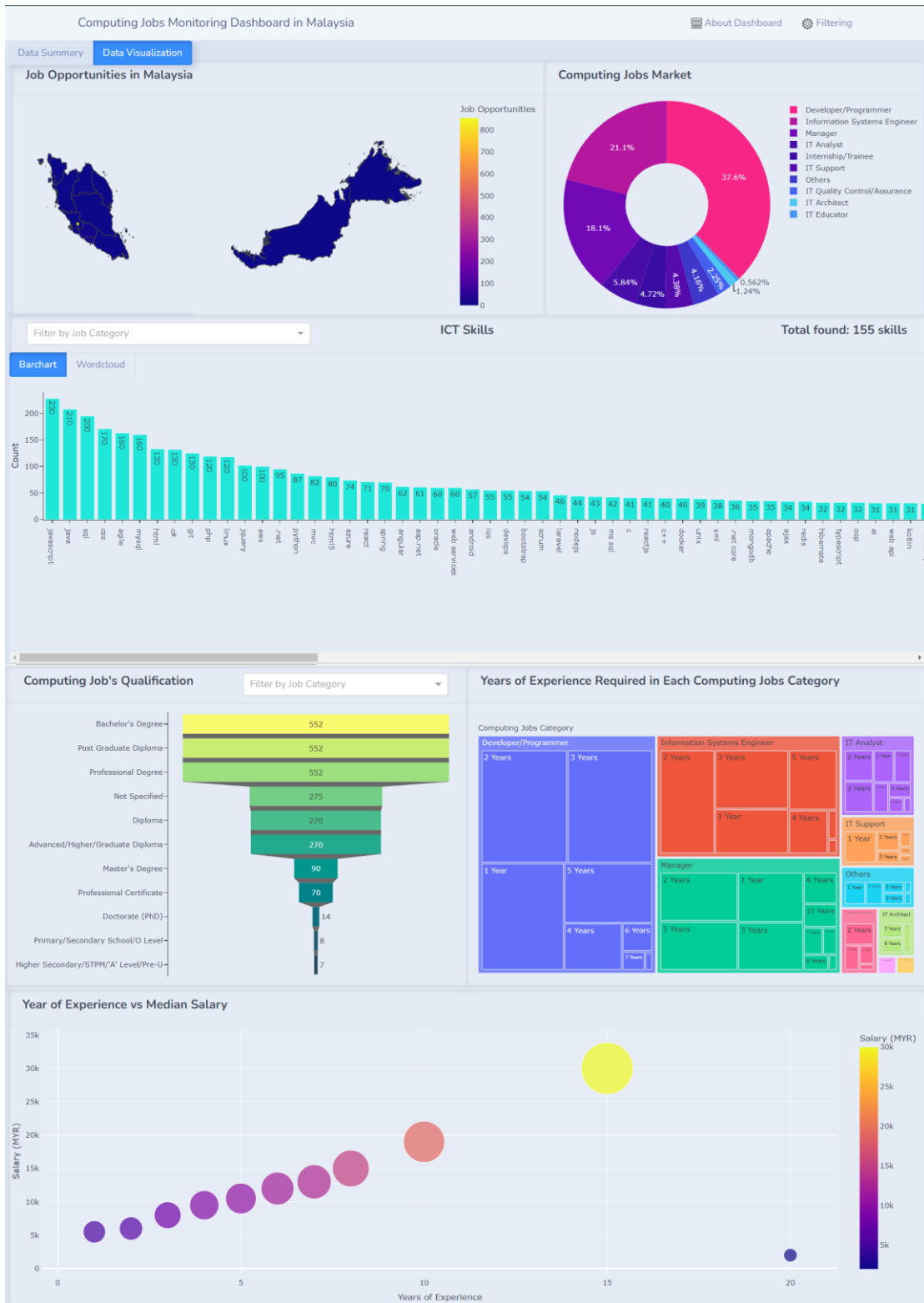


Figure 5.24 : Overview of the dashboard (2)

CHAPTER 5: SYSTEM IMPLEMENTATION

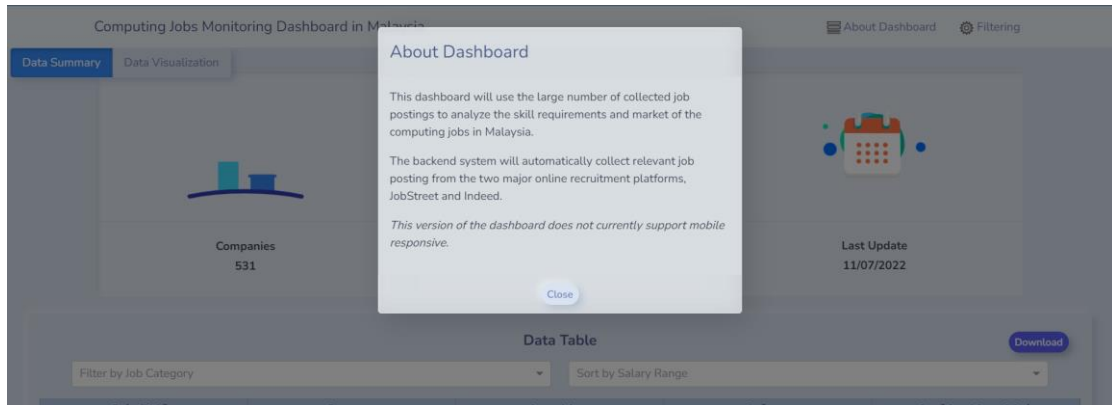
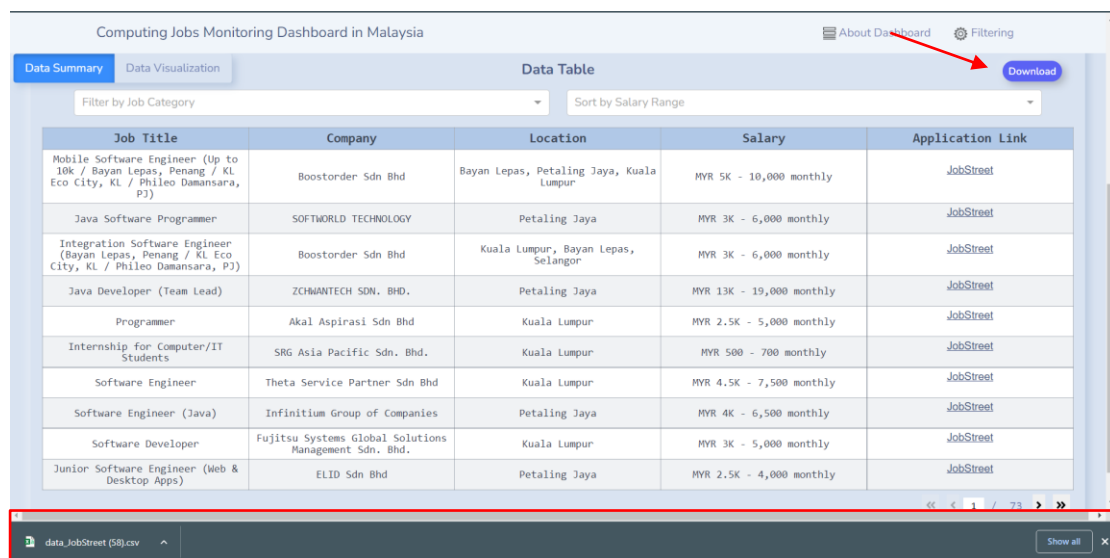


Figure 5.25 : The modal named About Dashboard

CHAPTER 5: SYSTEM IMPLEMENTATION

5.6.2 Data Summary in the Dashboard

The user interface of the data summary can be referred to Figure 5.23. The data summary is to enable the user to understand the data in a superficial way. The data summary includes the number of companies on the market, the number of jobs offered, and the date of the data was scraped. In addition, the data table is also provided in the data summary. The data table gives the user a more intuitive view of the information they want to know. Furthermore, the data table provides only five attributes that most job seekers are interested to know, including job title, company, location, salary and application link. The data table also provides users with the ability to filter and sort in order to view jobs of interest. The jobs in the data table can be further filtered by 10 different computing job categories. Apart from this, the sorting of the job details displayed in the data table can be sorted by salary range, such as high to low salary or low to high salary to view job information. Figure 5.26 shows that the user can download data from the data table. Finally, all the statistical data in the data summary and the data in the data table will change according to the filtering options selected by the user in the settings.



Computing Jobs Monitoring Dashboard in Malaysia

About Dashboard Filtering

Data Summary Data Visualization Data Table Download

Filter by Job Category Sort by Salary Range

Job Title	Company	Location	Salary	Application Link
Mobile Software Engineer (Up to 10k / Bayan Lepas, Penang / KL Eco City, KL / Phileo Damansara, PJ)	Boostorder Sdn Bhd	Bayan Lepas, Petaling Jaya, Kuala Lumpur	MYR 5K - 10,000 monthly	JobStreet
Java Software Programmer	SOFTWORLD TECHNOLOGY	Petaling Jaya	MYR 3K - 6,000 monthly	JobStreet
Integration Software Engineer (Bayan Lepas, Penang / KL Eco City, KL / Phileo Damansara, PJ)	Boostorder Sdn Bhd	Kuala Lumpur, Bayan Lepas, Selangor	MYR 3K - 6,000 monthly	JobStreet
Java Developer (Team Lead)	ZCHWANTECH SDN. BHD.	Petaling Jaya	MYR 13K - 19,000 monthly	JobStreet
Programmer	Akal Aspirasi Sdn Bhd	Kuala Lumpur	MYR 2.5K - 5,000 monthly	JobStreet
Internship for Computer/IT Students	SRG Asia Pacific Sdn. Bhd.	Kuala Lumpur	MYR 500 - 700 monthly	JobStreet
Software Engineer	Theta Service Partner Sdn Bhd	Kuala Lumpur	MYR 4.5K - 7,500 monthly	JobStreet
Software Engineer (Java)	Infinium Group of Companies	Petaling Jaya	MYR 4K - 6,500 monthly	JobStreet
Software Developer	Fujitsu Systems Global Solutions Management Sdn. Bhd.	Kuala Lumpur	MYR 3K - 5,000 monthly	JobStreet
Junior Software Engineer (Web & Desktop Apps)	ELID Sdn Bhd	Petaling Jaya	MYR 2.5K - 4,000 monthly	JobStreet

data_jobStreet (58).csv Show all

Figure 5.26 : Download data from the data table

5.6.3 Data Visualization in the Dashboard

Figure 5.24 also shows the user interface of the data visualization. Data visualization means using graphs such as bar charts, pie charts or maps etc. to present the data. The benefit of data visualization is that it is easier for users to identify strongly relevant parameters visually and graphically. The final dashboard will use 7 different charts to show the analyzed data and other valuable and meaningful information. These 7 different charts are choropleth maps, pie chart, bar chart, WordCloud, funnel chart, tree map and bubble chart. Each chart is carefully selected before being applied to ensure that the chart fits certain data and conveys information to the user in an intuitive and understandable way. Besides that, these charts will be regenerated based on information of interest to the user and can be downloaded.

5.6.3.1 Choropleth Map - Job Opportunities in Malaysia

The choropleth map is a map with areas filled in with colors to show its density and other statistical measures, and the map will be applied to show the job opportunities in Malaysia. The dashboard's backend will calculate the number of computing jobs available in each state to generate a choropleth map. The states shown on the map with brighter colors mean that more jobs are available in that area. Conversely, those areas filled with darker colors mean that only few jobs are available. Thus, the statistics on the map can be clearly defined by the color range. When the user hovers over a specific state, the name of the state, the number of jobs available, and the minimum and maximum wages are displayed. The figure below shows the choropleth map.

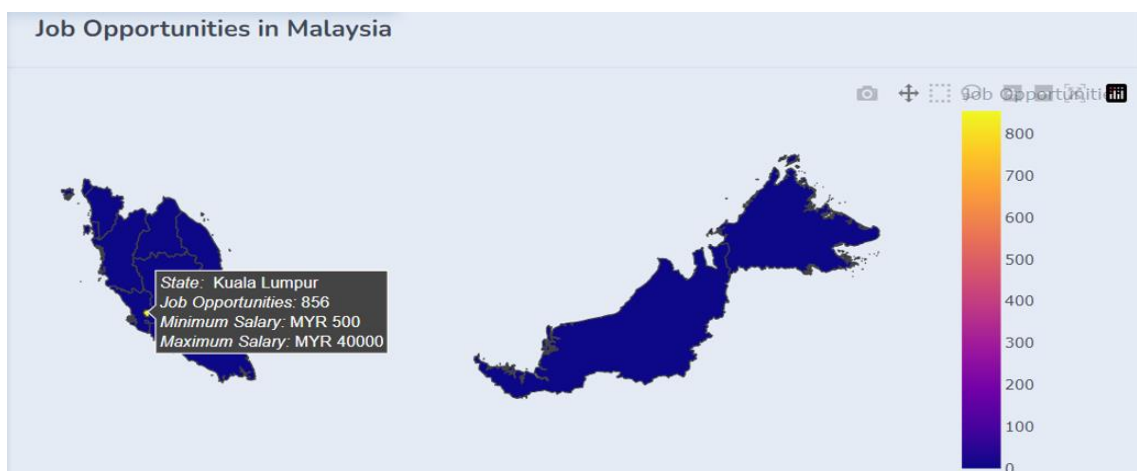


Figure 5.27 : Choropleth Map

5.6.3.2 Pie Chart- Computing Jobs Market

Figure 5.28 shows a pie chart of the computer jobs market. The pie chart is divided into 10 different computing jobs categories by percentage share in the market. The job categories are all included manager, information systems engineer, developer/programmer, IT analyst, IT support, internship/trainee, others, IT quality control/assurance, IT architect, IT educator. Hovering over a particular portion of the pie chart will display the job categories and the total number of available jobs. Users can learn more about current market trends for each computer job category, including which types of computing jobs are in high demand.

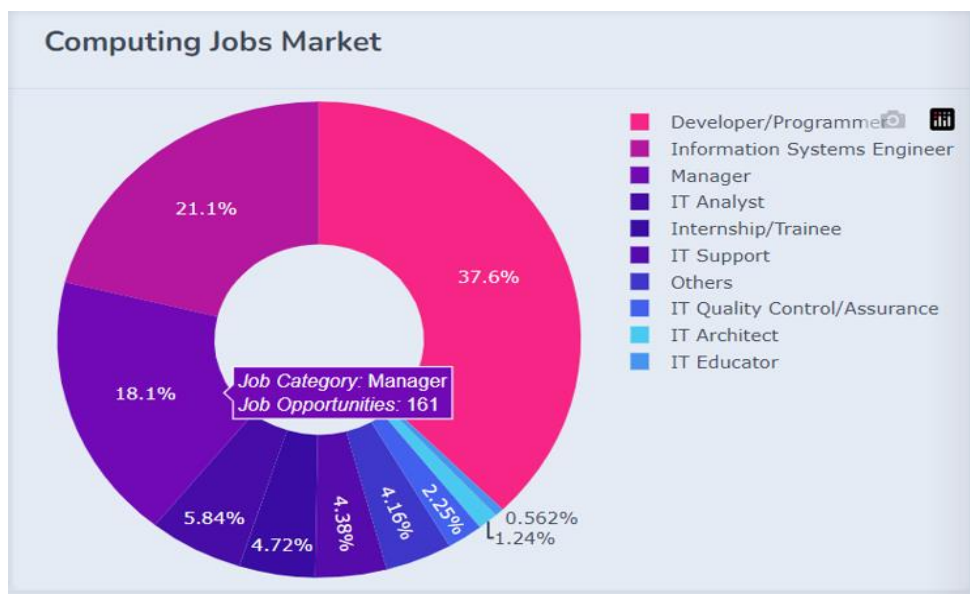


Figure 5.28 : Pie Chart

CHAPTER 5: SYSTEM IMPLEMENTATION

5.6.3.3 Bar Chart and WordCloud – ICT Skills

This dashboard will provide two different data visualizations to present ICT skills in demand in the market, which are bar chart and WordCloud. Bar chart can display relative numbers or ratios for multiple categories, so that users can get a good overview of that frequency distribution. Therefore, bar chart is a good choice for presenting multiple types of ICT skills. Furthermore, users able to scroll the horizontal axis of the bar chart when there are too many ICT skills required to plot. WordCloud is another way of presenting frequency distributions, using a collection of words to form a picture. In WordCloud, a large font size for an ICT skill means that the skill appears frequently, which allows users to quickly understand which ICT skills are important. Users can also view the ICT skills required for different job categories by selecting the filtering options provided. Thus, both bar chart and WordCloud will update the data simultaneously. Besides that, the number of ICT skills queried in the backend is also displayed in the upper right corner of the chart. Figures 5.29 and 5.30 show the ICT skills displayed by bar chart and WordCloud, respectively.

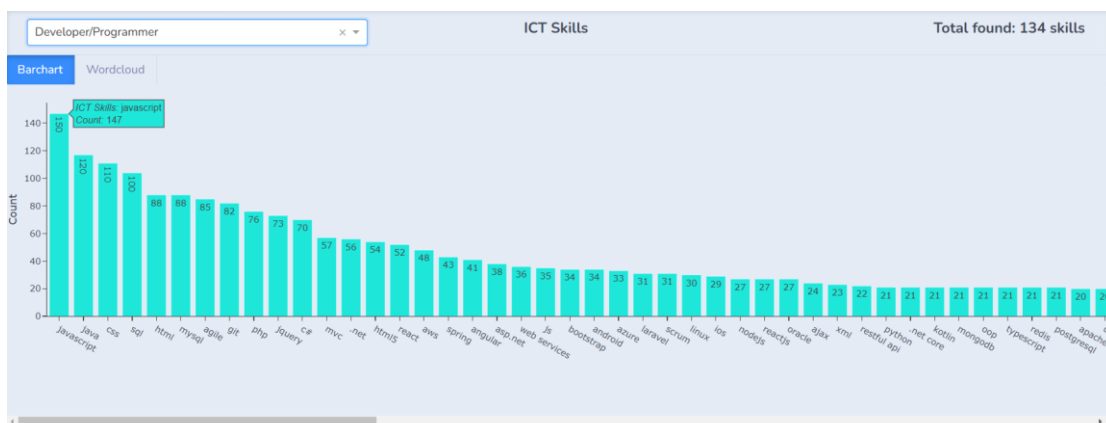


Figure 5.29 : Bar Chart

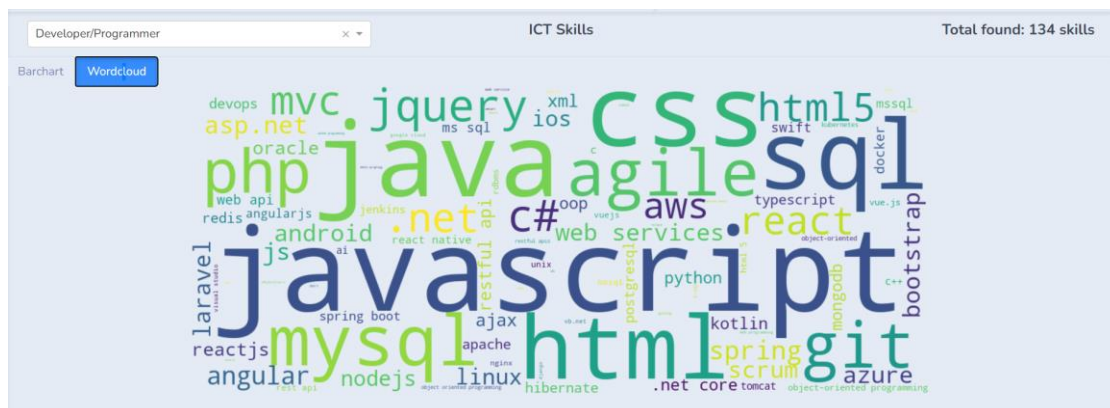


Figure 5.30 : WordCloud

5.6.3.4 Funnel Chart – Computing Job’s Qualification

Figure 5.31 shows the funnel chart for the qualification of each computing jobs. The funnel chart’s characteristic is hierarchical arrangement of the categories of interest from highest to lowest frequencies, which will eventually form a funnel shape. This characteristic gives the chart a clear frequency distribution, allowing users to quickly understand what level of education a job seeker needs to land a specific job in the market. The education level and the number of jobs requiring that level of education be displayed when the user hovers over the funnel chart. Apart from that, the funnel chart can be regenerated when the user needs further information about the level of education required for each job category.

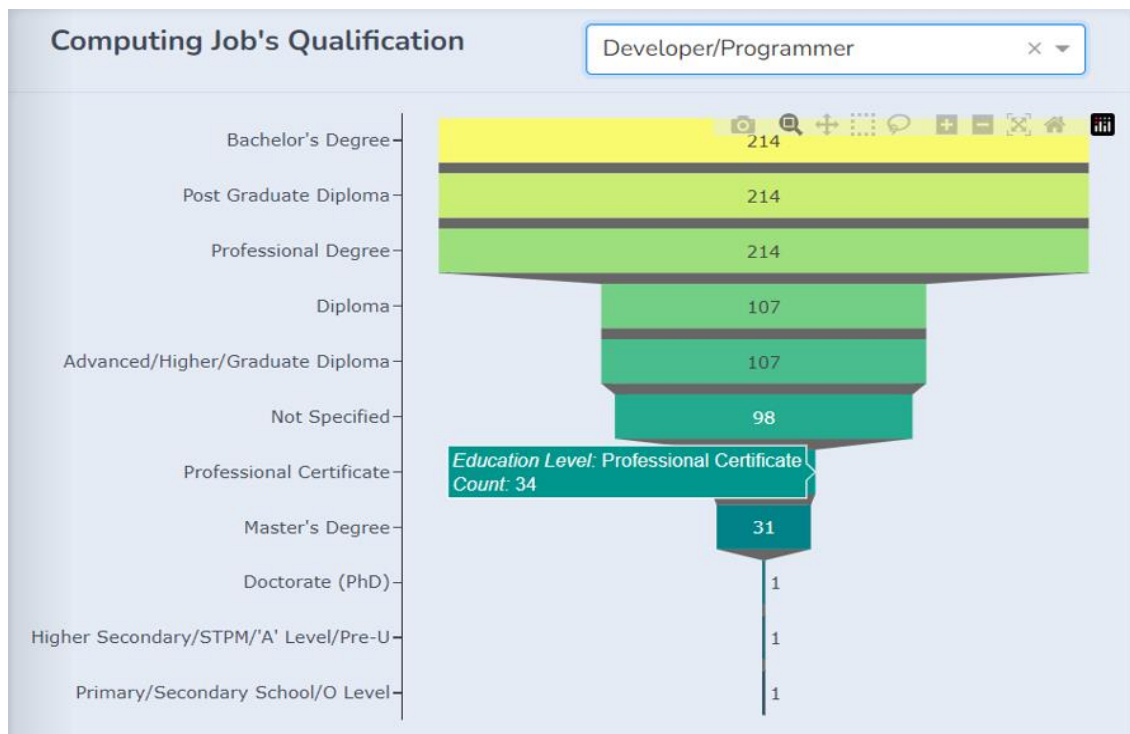


Figure 5.31 : Funnel Chart

5.6.3.5 Tree Map – Years of Experience Required in Each Computing Jobs Category

The tree map's characteristic is the data is represented as rectangle, the branches and sub-branches of the data are displayed. Furthermore, the area of the rectangles in the tree diagram is proportional to the number of each category. In this project, the tree diagram shows the years of experience for each computing jobs categories, as shown in Figure 5.32. When the area of rectangles for showing certain of years of experience is large, it means that there are more jobs available for that job category for those years of experience. Figure 5.33 shows that when the user clicks on the corresponding job category, the image expands to show more data. As with the other charts, the user can hover over the funnel chart to view the number of jobs available in each job category or years of experience.

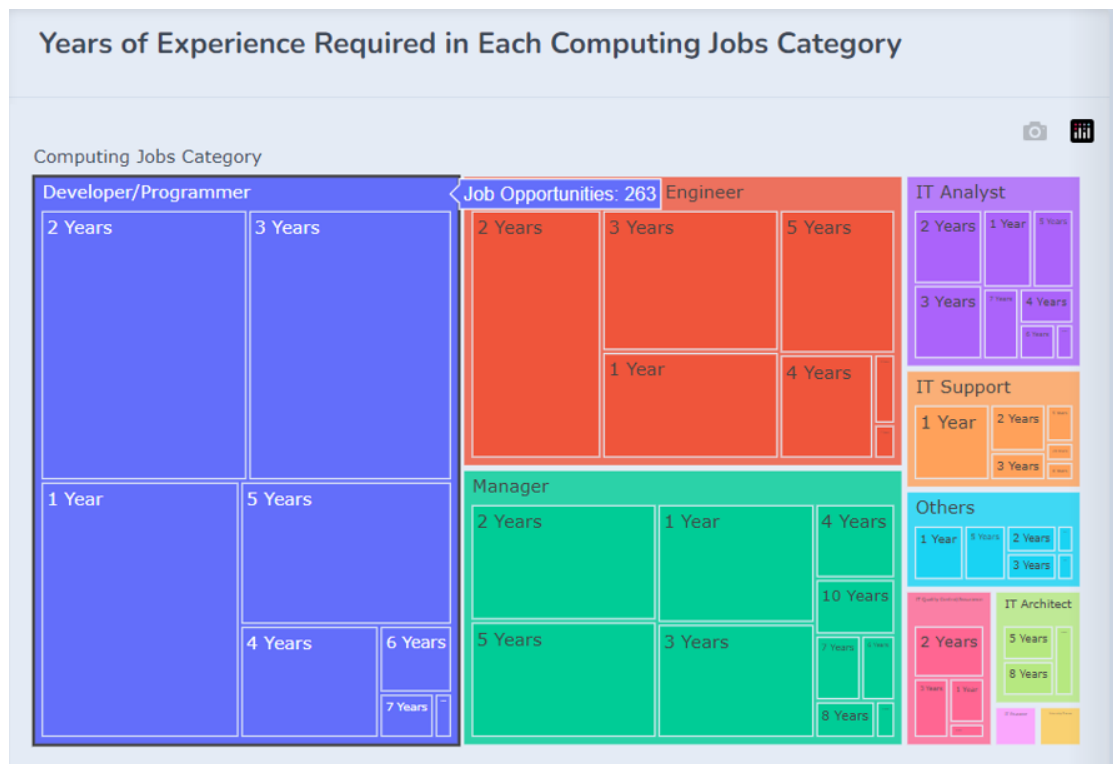


Figure 5.32 : Tree Map

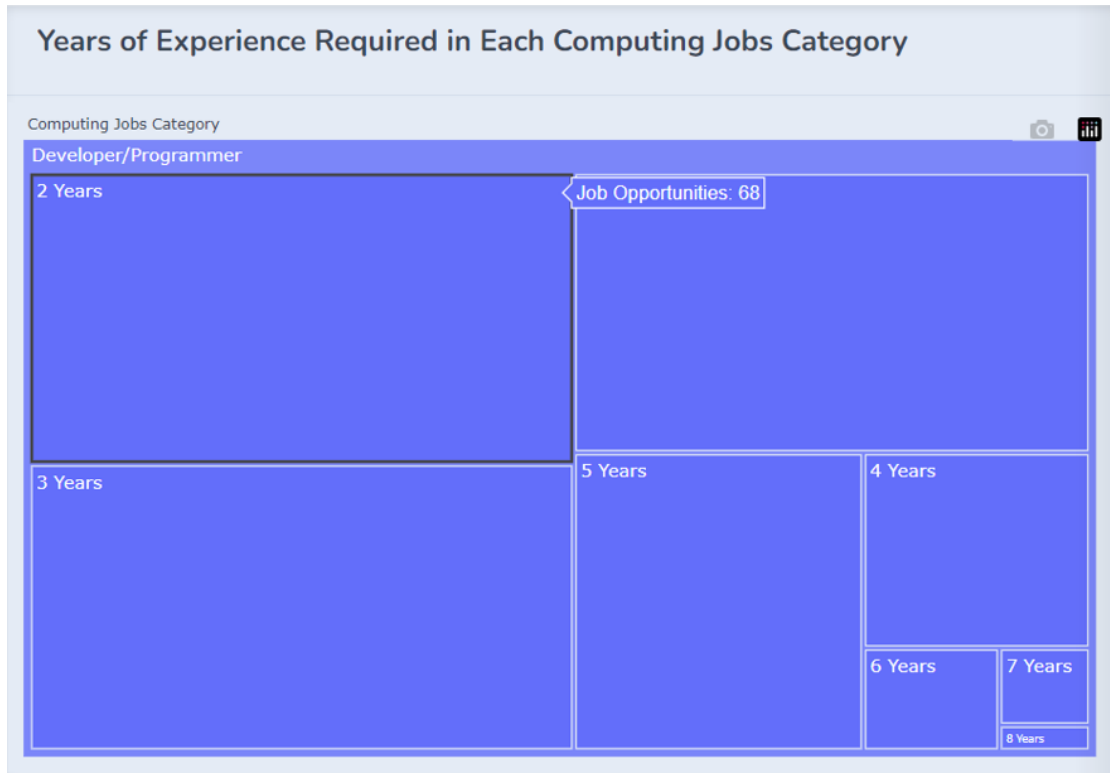


Figure 5.33 : Expand the Tree Map

CHAPTER 5: SYSTEM IMPLEMENTATION

5.6.3.6 Bubble Chart – Year of Experience vs Median Salary

Figure 5.34 shows the relationship between years of experience and median salary using a bubble chart. The main purpose of this chart is to determine whether years of experience have a positive effect on salary, since the common perception is that salary increases with years of experience. When the area of the bubble is large, it means that a high salary is offered for the corresponding number of years of experience. Years of experience, median salary and the number of computing jobs offered in that experience will be displayed when hovering over a bubble.

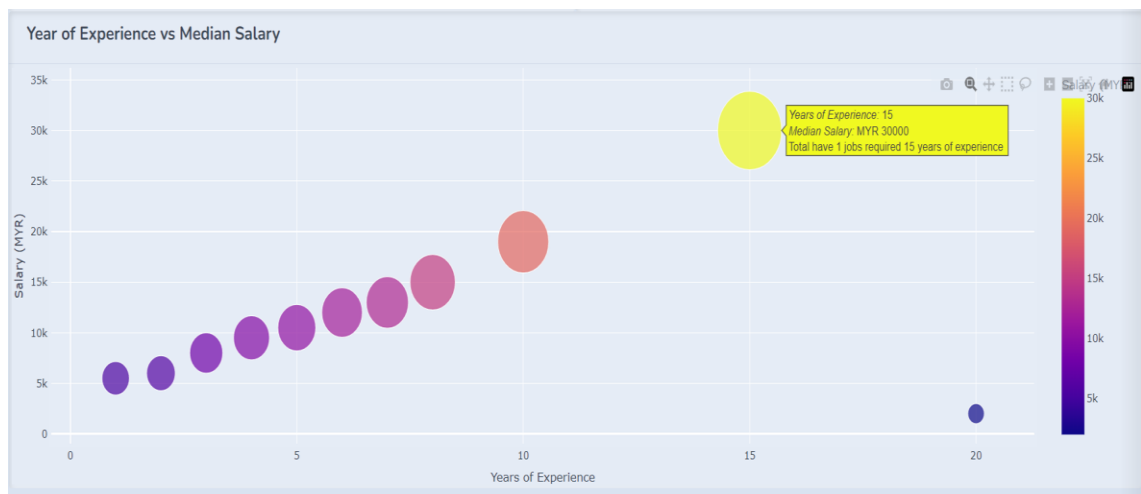


Figure 5.34 : Bubble Chart

CHAPTER 5: SYSTEM IMPLEMENTATION

5.6.4 Filtering in the Dashboard

As shown in Figure 5.35, the settings provide a number of filter options to the user, including job field, states and expected salary. The backend will query the appropriate data and update the data in the dashboard through multiple filtering options selected by the user. In addition, the system must also ensure that the data in both tabs of the data summary and data visualization must be updated simultaneously. If the backend cannot find any data based on the filtering options selected by the user, then an alert message is prompted to the user informing that no results were returned. Besides, all data and graphs are displayed blank. Figure 5.36 shows the dashboard interface appears an alert message when the backend does not query any corresponding data.

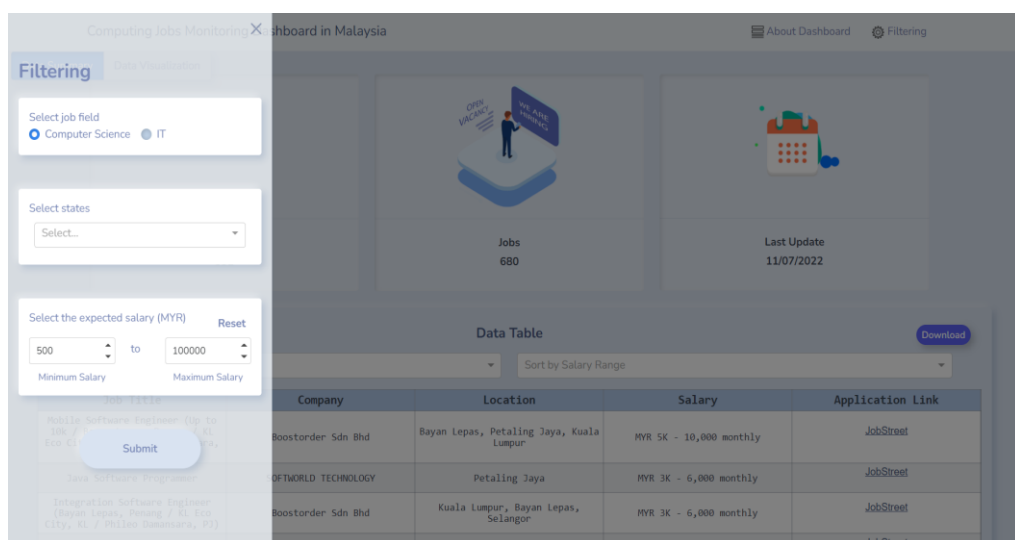


Figure 5.35 : Settings in dashboard

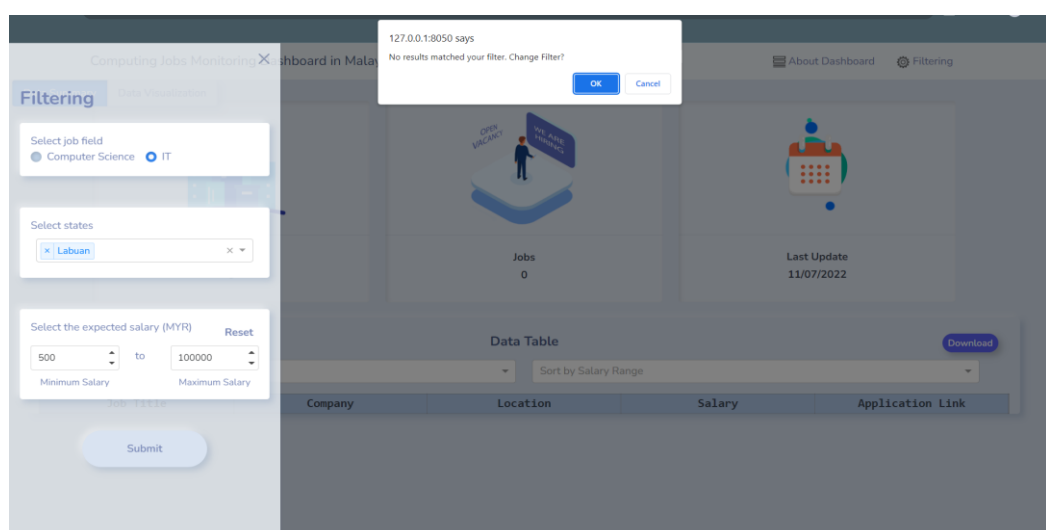


Figure 5.36 : Alert message

CHAPTER 5: SYSTEM IMPLEMENTATION

Figure 5.37 shows the filtering module for the dashboard, using callbacks provided by the Dash framework to filter and update the data on the dashboard. 835 to 840 lines of code are used to define which html components will be used for the callbacks, as well as to define the output, input, and state of the callbacks. The state that allows additional values to be passed without triggering the callback. The input of this callback is the submit button that need user to click it for triggering the filter module. Besides that, the output is return the filtered result in JSON format and stored it to user's browser. The reason for storing it the in the browser is to ensure that both the "Data Summary" and "Data Visualization" pages have the same data as a reference so that the data on each page can be synchronized and updated.

The code on lines 843 to 850 is the function named `value_job_categories`. It is responsible for returning data on the computer science or IT that the user has chosen to view. The code on line 858 will use this function and assign it to a parameter named `jobCopy`. Furthermore, line 852 code will detect if the submit button has been clicked. If it is not clicked, then it will raise `dash.exceptions.PreventUpdate` and not perform any data updates.

When the callback detects that the submit button has been clicked, lines 858 to 872 codes will be executed. Its main purpose is to filter the data by executing the filter option selected by the user. Eventually, the final filtered data will be parsed `DataFrame` to dictionary and returned it to a HTML component with the ID "update-dataframe". Also, an error handling added to lines 874 to 876 codes so that this callback will not perform any updates when an error occurs.

CHAPTER 5: SYSTEM IMPLEMENTATION

```
834 #Filtering
835 @app.callback(Output('update-dataframe', 'data'),
836               [State('state-dropdown', 'value'),
837                State('minSalary-input', 'value'),
838                State('maxSalary-input', 'value'),
839                State('job-categories-radiobutton', 'value')],
840               Input('submit-val', 'n_clicks'))
841 def func1(state_value,minSalary_value,maxSalary_value,job_categories_value,n_clicks):
842
843     def value_job_categories(job):
844         if job == "IT":
845             jobCopy = job_it.copy()
846
847         else:
848             jobCopy = job_cs.copy()
849
850         return jobCopy
851
852     if n_clicks is None:
853         raise dash.exceptions.PreventUpdate
854
855     else:
856         try:
857
858             jobCopy = value_job_categories(job_categories_value)
859
860             if state_value is None:
861                 state_value=""
862
863             if (len(state_value)>0):
864                 jobCopy = jobCopy[(jobCopy.StateCategory.isin(state_value)) & (jobCopy.SalaryUpperBound <= maxSalary_value) &
865                                 (minSalary_value <= jobCopy.SalaryLowerBound) == True]
866
867             elif(len(state_value)==0):
868                 jobCopy = jobCopy[(jobCopy.SalaryUpperBound <= maxSalary_value) & (minSalary_value <= jobCopy.SalaryLowerBound) == True]
869
870             jobCopy.reset_index(inplace=True,drop=True)
871
872             return jobCopy.to_dict('records')
873
874         except Exception as e:
875             print(e)
876             return dash.no_update
```

Figure 5.37 : The code for filtering data in dashboard

CHAPTER 6: CONCLSION

6.1 Project Review

Web Scraping is one of the methods that many companies will use to collect the required data and further analyze it into useful information. Currently, there is no platform or software on the market to analyze the computing jobs market in Malaysia in real time. Although systems or platforms with similar functionality may exist, they are used internally and not shared with the public due to the high value of the data being analyzed. The value of analyzing computing job postings is also mentioned in the contribution section in Chapter 1, such as the ICT skills required in the market or choosing the suitable computing job with their respective qualifications. Certainly, the beneficiary group of this project is large, including students, programmers, educational institutions or market analysts.

Therefore, this project will display analyzed job postings on a dashboard that will be shared to the public for free, and users can browse on the Internet at any time. Most importantly, users can get real-time analysis of computing jobs market in Malaysia without having to spend more time for browsing job postings on online job recruitment platforms to get the analysis.

Apart from that, all the objectives mentioned in Chapter 1 have been achieved. The dashboard provides users with different filtering options to further select job postings of interest, such as filtering data by job field, computing jobs categories, states, and expected salary. Not only that, but the filters provided also increases the interactivity of the dashboard with the user. The user is able to autonomous select the information to be viewed to generate quick and multiple insights. At the same time, the system's backend will automatically collect the data needed from the target site to ensure that all data is up to date.

CHAPTER 6: CONCLUSION

6.2 Novelties

The novelty of this project is mainly in the web scraping and data analysis process. During the planning stage of development, various software or platforms were tried to scrape data from the online job recruitment platform, but most of them did not perform as well as expected. The main problems were unable to scrape the specified data accurately and unstable performance. Besides that, some web scraping software does not provide task scheduling, which means that the systems cannot automatically scraping data at a given time, which is unfulfilled the requirements of the project.

Therefore, this project is to program a custom web scraping to scrape the real time job posting form different targeted website such as JobStreet and Indeed to build the proposed dashboard. The advantage of this approach is that developer can autonomously decide which data to scrape and keep optimize the code and performance of the web scraping script. The programmed web scraping scripts will be deployed to WayScript to execute automatically and store the latest scraped data in a database.

Furthermore, the project implemented machine learning and NER in the data analysis part. The main task of machine learning was to classify job postings into 10 different computing jobs categories by job title. Logistic regression was chosen as the classification technique because it achieved highest accuracy as compared to other machine learning techniques used. NER was used to extract ICT skills from the descriptive text of the job requirements. This project is not interested in the details of each job requirement, but in the listed ICT skills. The extracted ICT skills will be used to analyze which ICT skills are in high demand in the market.

In this project, there is no development cost for both front-end and back-end development. The proposed dashboard is deployed in Heroku for free and there is no charge for Azure Blob storage if the storage is small. Apart from that, WayScript offers 100 hours of free runtime per month, which is sufficient for web scraping. Therefore, the project can continue to run and serve users without any other costs.

CHAPTER 6: CONCLUSION

6.3 Future Work

There are many improvements that could be made, including scraping other online job recruitment platforms, not just JobStreet and Indeed. This improvement could provide users with more job posting on the dashboard. Another benefit is that the data imbalance can be effectively addressed by providing comprehensive training data for classification and NER models to improve accuracy.

Error handling can be added to the web scraping script so that appropriate actions can be taken in case of any errors occurs. This is because any possible error can occur during web scraping that causes it to not perform properly, such as website updates anti web scraping mechanism, server failure, and etc. At this point, error handling in the script can notify the administrator or developer of the error and make changes or concerns about it.

Finally, improvement on the file structure in dashboard development is required instead of writing all the code in one PY file. This makes it challenging for developers perform maintenance as the whole program code looks messy. Therefore, the file structure must be hierarchical, and code or modules for different purposes must be developed in different files.

REFERENCES

REFERENCES

- [1] U.C. Okolie and I.E.Irabor.(2017). E-recruitment: practices, opportunities and challenges. Presented at European Journal of Business and Management, 9(11). [Online]. Available: <https://core.ac.uk/download/pdf/234627826.pdf>
- [2] L.M. Kureková, M. Beblavý and A. Thum-Thysen.(2015). Using online vacancies and web surveys to analyse the labour market: a methodological inquiry. Presented at IZA Journal of Labor Economics. [Online]. Available: <https://link.springer.com/article/10.1186/s40172-015-0034-4>
- [3] T. Mohammad Akhriza, Y. Ma and J. Li, “Revealing the Gap Between Skills of Students and the Evolving Skills Required by the Industry of Information and Communication Technology”, in *International Journal of Software Engineering and Knowledge Engineering* , 2017, pp. 675–698, doi: 10.1142/s0218194017500255
- [4] K. Fauziah, and A.Triyanto.(2018). The influence of Jobstreet.com toward the fulfillment of job vacancy information needs. Presented at Library Philosophy and Practice, 2018. [Online]. Available: https://www.researchgate.net/publication/332574844_The_influence_of_Jobstreetcom_toward_the_fulfillment_of_job_vacancy_information_needs
- [5] B. Sainju, C. Hartwell and J. Edwards.(2021). Job satisfaction and employee turnover determinants in Fortune 50 companies: Insights from employee reviews from Indeed. com. Presented at Decision Support Systems. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0167923621000920>
- [6] S. Lunn, J. Zhu, and M. Ross, “Utilizing Web Scraping and Natural Language Processing to Better Inform Pedagogical Practice” in *IEEE Frontiers in Education Conference (FIE)*, Oct. 2020, doi: 10.1109/fie44824.2020.9274270
- [7] W. McKinney.(2011). pandas: a foundational Python library for data analysis and statistics. Presented at Python for high performance and scientific computing, 14(9). [Online]. Available: https://www.dlr.de/sc/portaldata/15/resources/dokumente/pyhpc2011/submissions/pyhpc2011_submission_9.pdf
- [8] S. Hagedorn, S. Kläbe and K.U Sattler.(2021). Putting Pandas in a Box. Presented at CIDR. [Online]. Available: https://www.researchgate.net/publication/355736963_Putting_Pandas_in_a_Box
- [9] C. Zheng, G.He. and Z.Peng, “A Study of Web Information Extraction Technology Based on Beautiful Soup” in *JCP 10(6)*, 2015, pp 31-387, doi: 10.17706/jcp.10.6.381-387
- [10] B. Liu, Y. Peng, Y. Zou, J. Wang and T. Jiang, “Web-Weka Meets D3. js in Web Based Medical Data Mining”, in *2015 8th International Symposium on Computational Bachelor of Computer Science (Honours) Faculty of Information and Communication Technology (Kampar Campus), UTAR*

REFERENCES

Intelligence and Design (ISCID), Dec. 2015, pp. 180-183, doi: 10.1109/ISCID.2015.309

[11] N.R. Haddaway.(2015). The use of web-scraping software in searching for grey literature. Presented at Grey J, 11(3). [Online].Available: https://www.researchgate.net/profile/Neal-Haddaway/publication/282658358_The_Use_of_Web-scraping_Software_in_Searching_for_Grey_Literature/links/5c3af240458515a4c721fea1/The-Use-of-Web-scraping-Software-in-Searching-for-Grey-Literature.pdf

[12] B. Jena, “An Approach for Forecast Prediction in Data Analytics Field by Tableau Software” in *International Journal of Information Engineering & Electronic Business*, 2019, pp 19-26, doi: 10.5815/ijieeb.2019.01.03.

[13] S. Hossain, “Visualization of bioinformatics data with dash bio” , in *Proceedings of the 18th Python in Science Conference*, 2019, pp. 126-133, doi: 10.25080/majora-7ddc1dd1-012

[14] S.M. Ali, N. Gupta, Nayak, G.K. and R.K. Lenka, “Big data visualization: Tools and challenges”, in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)* , Dec. 2016, pp. 656-660, doi: 10.1109/IC3I.2016.7918044.

[15] A. Haque and S. Singh, “Anti-scraping application development” in *2015 international conference on advances in computing, communications and informatics (ICACCI)* , Aug. 2015, pp. 869-874, doi: 10.1109/ICACCI.2015.7275720.

[16] S. Slater, S. Joksimović, V. Kovanovic, R.S. Baker and D. Gasevic, “Tools for educational data mining”, in *Journal of Educational and Behavioral Statistics* , 2016, pp. 85-106, doi: 10.3102/1076998616666808

[17] F. Blom, M. Heine, I. van den Heuvel, C. Reep, D. Steenhof and M. Burch. An Interactive Visualization Tool for Dynamic Graphs, Node-Link Diagrams, and Adjacency Matrices.[Online].Available: https://www.researchgate.net/profile/Michael-Burch/publication/331641305_An_Interactive_Visualization_Tool_for_Dynamic_Graphs_Node-Link_Diagrams_and_Adjacency_Matrices/links/5c852cc092851c695069d7a9/An-Interactive-Visualization-Tool-for-Dynamic-Graphs-Node-Link-Diagrams-and-Adjacency-Matrices.pdf

[18] J. Martens, J. Kalisvaart, T. Quadt, J. Niederle, L. Snijder and M. Burch, DynaVis: The Visualization Tool for Dynamic Graph Data.[Online].Available: https://www.researchgate.net/profile/Michael-Burch/publication/332292744_DynaVis_The_Visualization_Tool_for_Dynamic_Graph_Data/links/5cac84a24585158cc21a54bd/DynaVis-The-Visualization-Tool-for-Dynamic-Graph-Data.pdf

REFERENCES

- [19] S. Balaji and M.S. Murugaiyan, 2012. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. Presented at International Journal of Information Technology and Business Management 2012.[Online].Available: <https://mediaweb.saintleo.edu/Courses/COM430/M2Readings/WATERFALLVs%20V-MODEL%20Vs%20AGILE%20A%20COMPARATIVE%20STUDY%20ON%20SDLC.pdf>
- [20] Zhang, H., Chen, P. and Wang, Q., "Fault diagnosis method based on EEMD and multi-class logistic regression", in *2018 3rd International Conference on Smart City and Systems Engineering (ICSCSE)*, Dec. 2018, pp. 859-863, doi:10.1109/ICSCSE.2018.00185
- [21] Chen, J., Huang, H., Tian, S. and Qu, Y., "Feature selection for text classification with Naïve Bayes" in *Expert Systems with Applications*, 2009, pp.5432-5435, doi:10.1016/j.eswa.2008.06.054
- [22] Farid, D.M., Zhang, L., Rahman, C.M., Hossain, M.A. and Strachan, R., "Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks" in *Expert systems with applications*, 2014, pp.1937-1946, doi:10.1016/j.eswa.2013.08.089
- [23] Chen, X., Yu, D., Fan, X., Wang, L. and Chen, J., " Multiclass Classification for Self-Admitted Technical Debt Based on XGBoost" in *IEEE Transactions on Reliability*, 2021, doi:10.1109/TR.2021.3087864
- [24] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in Proc. Adv. Neural Inf. Process. Syst. 30: Annu. Conf. Neural Inf. Process. Syst., Long Beach, CA, USA, 2017, pp. 3146–3154.
- [25] Chamasemani, F.F. and Singh, Y.P., "Multi-class support vector machine (SVM) classifiers--an application in *hypothyroid detection and classification*" in *2011 sixth international conference on bio-inspired computing: theories and applications*, 2011, pp. 351-356, doi:10.1109/BIC-TA.2011.51
- [26] Alsaaran, N. and Alrabiah, M., "Arabic named entity recognition: A BERT-BGRU approach" in *Comput. Mater. Contin.*, 2021, pp.471-485, doi:10.32604/cmc.2021.016054
- [27] Mack, D., "How to pick the best learning rate for your machine learning project", [freecodecamp.org](https://www.freecodecamp.org/news/how-to-pick-the-best-learning-rate-for-your-machine-learning-project-9c28865039a8). [Online]. Available: <https://www.freecodecamp.org/news/how-to-pick-the-best-learning-rate-for-your-machine-learning-project-9c28865039a8>

REFERENCES

- [28] Pavan.S, "What is Dropout Regularization? Find out :)", Kaggle.com, [Online]. Available:<https://www.kaggle.com/code/pavansanagapati/what-is-dropout-regularization-find-out/notebook>
- [29] Meng, L., McWilliams, B., Jarosinski, W., Park, H.Y., Jung, Y.G., Lee, J. and Zhang, J., "Machine learning in additive manufacturing: a review" in Jom, 2020, pp.2363-2377, doi:10.1007/s11837-020-04155-y

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2022	Study week no.: Week 2
Student Name & ID: Tan Zhen Wei (19ACB06234)	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Computing Jobs Monitoring Dashboard Malaysia	

1. WORK DONE

- Refactored all program code in FYP1.
- Completed Indeed's web crawling script.
- Completed Indeed's data cleaning module.

2. WORK TO BE DONE

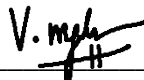
- Finding information about the multiclass classification's approach, so that this project able to classify all computing jobs into different jobs categories.
- Learning the NER to identify the ICT skills in each job description.

3. PROBLEMS ENCOUNTERED

The accuracy of classify the computing jobs by using word similarity is poor. Therefore, more times needed to study and learn other methods for multiclass classification.

4. SELF EVALUATION OF THE PROGRESS

I learned a lot about NER, which is interesting to me. I am trying to start programming a custom NER model in the next few weeks.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2022	Study week no.: Week 4
Student Name & ID: Tan Zhen Wei (19ACB06234)	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Computing Jobs Monitoring Dashboard Malaysia	

1. WORK DONE

- Collected sufficient data for training NER model.
- Programmed initial NER model.

2. WORK TO BE DONE

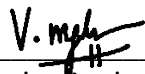
- Learning the multiclass classification references given by supervisor.
- Start training the multiclass classification model.
- Keep improving the accuracy of the NER model.

3. PROBLEMS ENCOUNTERED

Training the NER model is time consuming, taking almost two hours to complete the process. Even when running this process with Goggle Colab's GPU, there was no significant reduction in training time.

4. SELF EVALUATION OF THE PROGRESS

Training the NER model is time consuming, so it requires me to make sure that everything is valid and confirmed before training the model.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2022	Study week no.: Week 6
Student Name & ID: Tan Zhen Wei (19ACB06234)	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Computing Jobs Monitoring Dashboard Malaysia	

1. WORK DONE

- Trained NER model.
- Collected data to train a multi-class classification model.
- Learned various types of machine learning.

2. WORK TO BE DONE

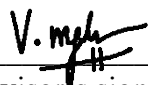
- Start to use different machine learning methods to train the classification model, including Logistic Regression, Naïve Bayes Classifier, XGBoost and Support Vector Machines (SVM).
- Keep improving the accuracy of the training model.

3. PROBLEMS ENCOUNTERED

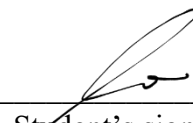
- The initial training models for each method perform well, but still require keep tuning to ensure high accuracy.
- There is an imbalance data for each classes, so it is necessary to resolve the problem.

4. SELF EVALUATION OF THE PROGRESS

I really learned a lot about different of machine learning, which was interesting to me.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2022	Study week no.: Week 8
Student Name & ID: Tan Zhen Wei (19ACB06234)	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Computing Jobs Monitoring Dashboard Malaysia	

1. WORK DONE

- Trained multi-class classification model
- Deploy data cleaning and data analysis models to WayScript.

2. WORK TO BE DONE

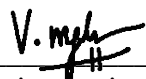
- Start programming the user interface of the dashboard
- Start plotting different charts and applying them to the dashboard
- Study Heroku's documentation to deploy the dashboard

3. PROBLEMS ENCOUNTERED

A lot of time is spent on selecting an appropriate chart to present certain data. This is because it is important to ensure that the user can easily get information from the chart.

4. SELF EVALUATION OF THE PROGRESS

Development is inefficient and requires a good time management to make sure everything is on track.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2022	Study week no.: Week 10
Student Name & ID: Tan Zhen Wei (19ACB06234)	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Computing Jobs Monitoring Dashboard Malaysia	

1. WORK DONE

- Finalized all modules
- Program the dashboard
- Deploying the dashboard

2. WORK TO BE DONE

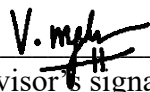
- Keep improving the dashboard, including user experience and functionality.
- Modify the design of the dashboard based on suggestions from my supervisor
- Add application links to the dashboard to allow users to easily apply for jobs.
- Ensure front-end and back-end are well connected and transfer data to each other.

3. PROBLEMS ENCOUNTERED

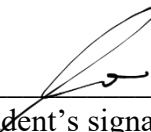
Keep improving the dashboard and amend some details. This is because the first impression and experience is also a key factor to determine the success of the product.

4. SELF EVALUATION OF THE PROGRESS

Deploying dashboards to the public has been a great experience for me, and I really appreciate Heroku because it makes it easy for developers to deploy their work.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: June, 2022	Study week no.: Week 12
Student Name & ID: Tan Zhen Wei (19ACB06234)	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Computing Jobs Monitoring Dashboard Malaysia	

1. WORK DONE

- Finalize every model and dashboard
- Report writing on progress

2. WORK TO BE DONE

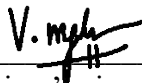
- Complete the FYP2 report

3. PROBLEMS ENCOUNTERED

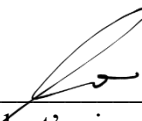
-

4. SELF EVALUATION OF THE PROGRESS

- Learned a lot about the development of frontend and backend.
- Improve programming problem solving during develop this project.



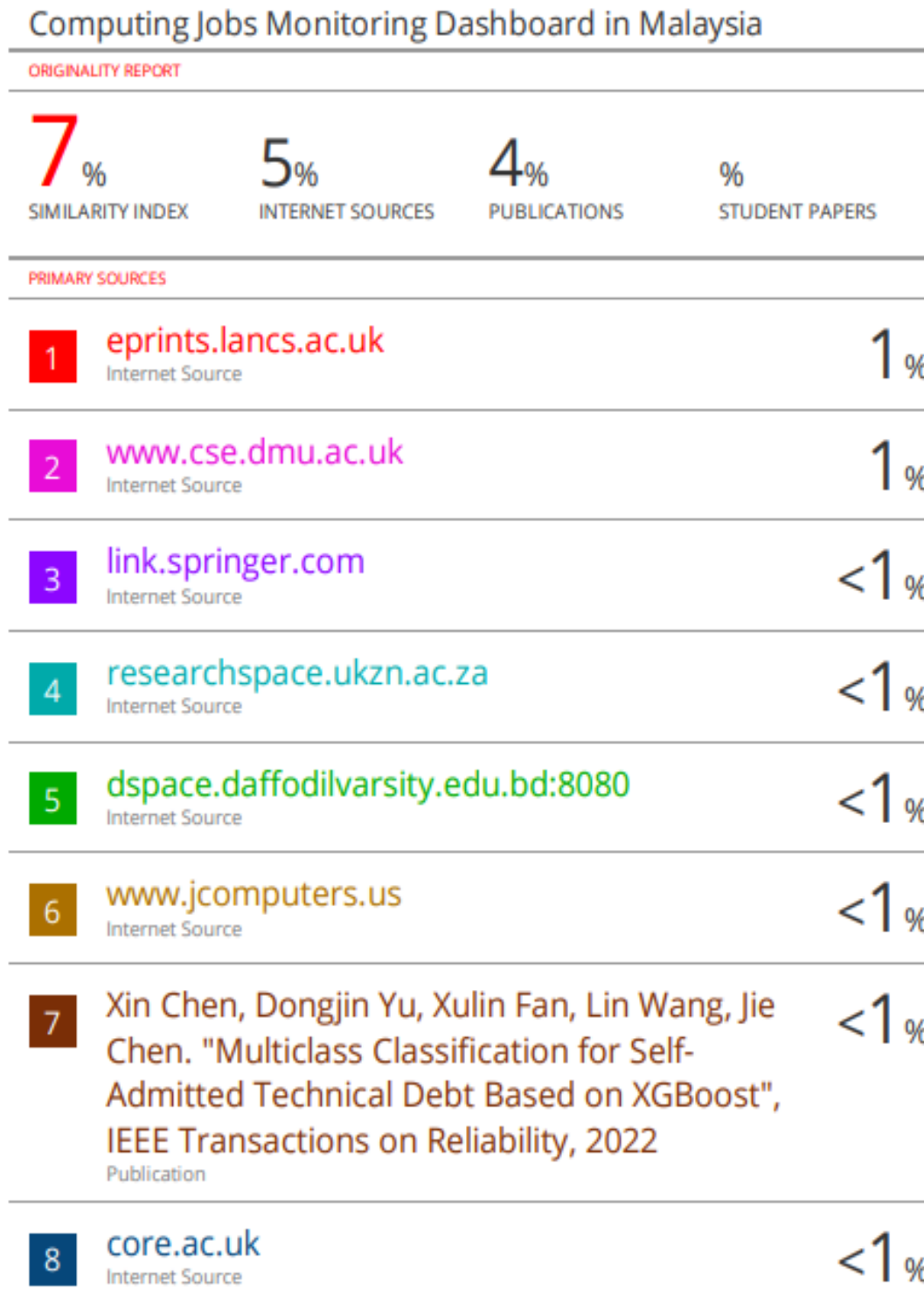
Supervisor's signature



Student's signature

PLAGIARISM CHECK RESULT

PLAGIARISM CHECK RESULT



PLAGIARISM CHECK RESULT

9	myassignmenthelp.com Internet Source	<1 %
10	www.ukessays.com Internet Source	<1 %
11	Srinivasa K G, Muralidhar Kurni. "A Beginner's Guide to Learning Analytics", Springer Science and Business Media LLC, 2021 Publication	<1 %
12	Hui Zhang, Ping Chen, Qiang Wang. "Fault Diagnosis Method Based on EEMD and Multi-Class Logistic Regression", 2018 3rd International Conference on Smart City and Systems Engineering (ICSCSE), 2018 Publication	<1 %
13	www.db-thueringen.de Internet Source	<1 %
14	Ivan Miguel Pires, Faisal Hussain, Nuno M. Garcia, Eftim Zdravevski. "Improving Human Activity Monitoring by Imputation of Missing Sensory Data: Experimental Study", Future Internet, 2020 Publication	<1 %
15	Bishal Sainju, Chris Hartwell, John Edwards. "Job satisfaction and employee turnover determinants in Fortune 50 companies: Insights from employee reviews from Indeed.com", Decision Support Systems, 2021	<1 %

PLAGIARISM CHECK RESULT

Publication		
16	slides.com Internet Source	<1 %
17	www.idr.iitkgp.ac.in Internet Source	<1 %
18	www.trustradius.com Internet Source	<1 %
19	"The Semantic Web – ISWC 2020", Springer Science and Business Media LLC, 2020 Publication	<1 %
20	www.worldscientific.com Internet Source	<1 %
21	Afzalul Haque, Sanjay Singh. "Anti-scraping application development", 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015 Publication	<1 %
22	Fereshteh Falah Chamasemani, Yashwant Prasad Singh. "Multi-class Support Vector Machine (SVM) Classifiers -- An Application in Hypothyroid Detection and Classification", 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications, 2011 Publication	<1 %

PLAGIARISM CHECK RESULT

23	vdocuments.site Internet Source	<1 %
24	Gunarathne, Thilina, Bingjing Zhang, Tak-Lon Wu, and Judy Qiu. "Scalable parallel computing on clouds using Twister4Azure iterative MapReduce", Future Generation Computer Systems, 2012. Publication	<1 %
25	conference.scipy.org Internet Source	<1 %
26	kau.diva-portal.org Internet Source	<1 %
27	buuic.buu.ac.th Internet Source	<1 %
28	www.scinapse.io Internet Source	<1 %
29	David J. Cox, Bryan Klapes, John Michael Falligant. "Scaling N from 1 to 1,000,000: Application of the Generalized Matching Law to Big Data Contexts", Perspectives on Behavior Science, 2021 Publication	<1 %
30	www.researchgate.net Internet Source	<1 %
31	trap.ncirl.ie Internet Source	<1 %

PLAGIARISM CHECK RESULT

32	M Nijila, M T Kala. "Extraction of Relationship Between Characters in Narrative Summaries", 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), 2018 Publication	<1 %
33	www.dailymail.co.uk Internet Source	<1 %
34	downloads.hindawi.com Internet Source	<1 %
35	www.duke.edu Internet Source	<1 %
36	Maryam KafiKang, Abdeltawab Hendawi. "Drug-Drug Interaction Extraction from Biomedical Text using Relation BioBERT with BLSTM", Cold Spring Harbor Laboratory, 2022 Publication	<1 %
37	repository.usd.ac.id Internet Source	<1 %
38	Andreas François Vermeulen. "Chapter 10 Transform Superstep", Springer Science and Business Media LLC, 2018 Publication	<1 %
39	kr.mathworks.com Internet Source	<1 %
	mail.grossarchive.com	

PLAGIARISM CHECK RESULT

40	Internet Source	<1 %
41	pastebin.com Internet Source	<1 %
42	www.tech.dmu.ac.uk Internet Source	<1 %
43	Herman Tang. "Engineering Research", Wiley, 2021 Publication	<1 %
44	eclipse.genesys.shef.ac.uk Internet Source	<1 %
45	oa.upm.es Internet Source	<1 %
46	origin.geeksforgeeks.org Internet Source	<1 %
47	www.kaggle.com Internet Source	<1 %
48	Lin, C.T.. "A neural fuzzy system for image motion estimation", Fuzzy Sets and Systems, 20000901 Publication	<1 %

PLAGIARISM CHECK RESULT

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



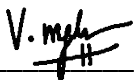
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Tan Zhen Wei
ID Number(s)	19ACB06234
Programme / Course	Bachelor in Computer Science (Honours)
Title of Final Year Project	Computing Jobs Monitoring Dashboard in Malaysia

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u>7</u> % Similarity by source Internet Sources: <u>5</u> % Publications: <u>4</u> % Student Papers: <u>0</u> %	No comments.
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.



Signature of Supervisor

Name: Dr Mogana a/p Vadiveloo

Date: 09/09/2022

Signature of Co-Supervisor

Name: _____

Date: _____

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

FYP2 CHECKLIST



UNIVERSITI TUNKU ABDUL RAHMAN

**FACULTY OF INFORMATION & COMMUNICATION
TECHNOLOGY (KAMPAR CAMPUS)
CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	19ACB06234
Student Name	Tan Zhen Wei
Supervisor Name	Dr Mogana a/p Vadiveloo

TICK (√)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
×	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
×	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 09/09/2022