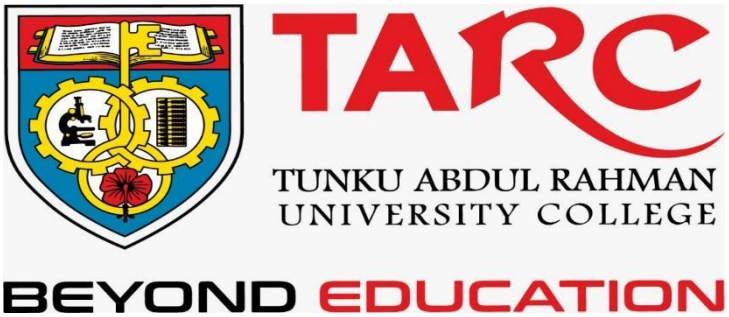
# Web scraping and data analytics (ranking, sorting and student activities) for UIP databases.

By

Kong Mun Jun




FACULTY OF COMPUTING AND
INFORMATION TECHNOLOGY

TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE
KUALA LUMPUR

ACADEMIC YEAR
2021/2022

# Web scraping and data analytics (ranking, sorting and student activities) for UIP databases

By

Kong Mun Jun

Supervisor: Dr. Lim Siew Mooi

A project report submitted to the
Faculty of Computing and Information Technology
in partial fulfillment of the requirement for the
Bachelor of Computer Science (Honours) in Data Science.

**Department of Computer Science and Embedded Systems**
Faculty of Computing and Information Technology
Tunku Abdul Rahman University College
Kuala Lumpur

# Declaration

The project submitted herewith is a result of my own efforts in totality and in every aspect of the project works. All information that has been obtained from other sources had been fully acknowledged. I understand that any plagiarism, cheating or collusion or any sorts constitutes a breach of TAR University College rules and regulations and would be subjected to disciplinary actions.

Kong Mun Jun

**Bachelor of Computer Science (Honours) in Data Science**

ID: 20WMR08867

# Abstract

In this globalised era, graduating from university is like a standard procedure for getting an active job in society. University also increased in numbers and became standard for students to achieve their high-level studies in every industry. However, information from every university's website is rarely standardised. Some even fragmented, which caused the third party such as students and parents to have a hard time doing research with every university. This phenomenon will bring students into an unfavourable situation where they might choose the wrong university for themselves due to the difficulties in investigating the aspects and benefits of each university.

This project aims to develop educators in Universities to convert their areas of expertise into prototypes, curriculum, industry-friendly collaboration models and develop new areas of research with AI-enabled engines. This project will discover each educator and his/her specialisation and achievement, then create a personalised AI expert system.

# Acknowledgement

I want to give my most tremendous gratitude to my supervisor, Dr. Lim Siew Mooi, for allowing me to collaborate with MyFinB, a potential Artificial Intelligence (AI) company, in doing this final year project (FYP). I also very much appreciate Dr Lim for giving me and my team members, Lee Kah Wei, Chin Qian Ying, Ng Yi Xuan, Koay Mei Siew, Eng Kay Tze and Lee Wei Kang a lot of guidance in helping all of us in completing this FYP project. Last but not least, I would like to thank MyFinB team, who gave a lot of advice and fantastic ideas for this project.

# Table of Contents

**List of Figures**

**List of Tables**

# Chapter 1

# Introduction

# Introduction

This project is involved in big data analytics to help universities and industries create a linked connection among Universities, industries, and Investors. With the established relationships, the universities' published research and uncommercialised patents can be found by the investors and enterprises to turn these universities works into commercialised in the community. The data sets used in this project involve universities, governments agencies and industries.

This project includes data crawling from various websites, segmenting raw data into data clusters, and inserting them into an online database followed by data analyses and displaying the informative dashboards. This project involves a few groups of team members.

## 1.1 Background

MyFinB (MFB) is one of the award-winning AI companies that frequently plays an AI-as-a-Service (AIaaS) platform for various industries. They focus on building a new proprietary analysis platform with excellent tools to transform structures and unstructured data into a readable and user-friendly format for the third party.

## 1.2 Problem Statement

With the increasing number of universities in Malaysia, it takes time for third parties such as teenagers, parents, and lecturers to explore and research the universities with every aspect such as the Courses, Subject material, Social activities, Portfolios etc. However, some information that we can find on websites is fragmented into broken pieces where the data are not close together.

Some university information is outdated, such as the alumni of the university and facilities developments. Therefore, these fragmented data became the core problem of a third party having difficulties getting accurate real-time information.

This core problem, the fragmented data issues in some universities, affected two main phenomena in society. Firstly, the third party had a hard time producing a better choice in targeting the most suitable university for every individual.

Constructive suggestions and feedback from all parties, including students, are vital for improvement purposes.

The second phenomenon is that the connection between the industry, universities and investors is weak. It is believed that some universities have numerous published researches and non-commercialised patents that need more engagement and adoption by the industries. Moreover, technology is getting more contact with educators' teaching methods, which change students' learning experience norms. Hence, enterprises need to have a strong connection with the educators to ensure the enhanced learning scope and contents to be learnt by the students in universities.

## 1.3 Company Background of Industry Collaborator


Figure 1.3.1 UIP, CE.A.I., and MyFinB Logo

MyFinB is a risk analytic platform that uses artificial intelligence to process assessments and insights to the users and industries. With the extensive demand for Artificial Intelligence in the next few years, they start providing a platform by gathering all different patents and research for an efficient way for the community and investors to understand and look through all educations data.

## 1.4 Project Background

The characteristics of this project are based on analysing a vast amount of various universities data that look less informative and turn them into meaningful information by segmentation in findings insights from the pieces of information of universities.

### 1.4.1 Target Market

The target market of this project is the Universities researchers, the academic community, Centres of Excellence of Universities, Government agencies and Ministries, SMEs and Corporation in 15 industry Groups, Investors, and Undergraduates and Postgraduate students.

1. University researchers and the academic community

They play a role of contributing and sharing their research or patent ideas to the platform as the platform tends to accept as much as possible of researches and patents to display and commercialise the education materials to the public

2. Centres of Excellence of Universities

Centres of Excellence in Universities are targeted. They could assist the project outside of the platform to filter out the best tea of experts to help out with the research or uncommercialised patents.

3. Government agencies and Ministries

Government is one of the target markets. Therefore, we are looking for agencies and ministries to plan their projects for future industries and educational institutions for a better wisdom level in the country.

4. SMEs and Corporations in 15 industry groups

Small Medium Enterprise plays a role of invest in various patents to be part of the contributors and use the patent technology or ideas in the community for the increment of their business and economics.

5. Investors (venture capitalists and private equity groups)

We need investors as the platform are a long-lasting project and venue for the education community. It cannot be denied that a minority of the researchers or patents need long-lasting equity to continue and succeed in their researches. Hence, Investors from venture capitalists and private equity groups are one of the target markets we would capture.

6. Undergraduates and Postgraduate students

We want undergraduates and post-graduate students to be part of the platform as they could easily find their interest in researches from the platform and build up their passion from it. They also can analyse the universities from the activities and information from the platform and make a good decision to pursue their studies in the right university institutions.

### 1.4.2 Competition / Contribution

Currently, very few similar purposes and characteristic platforms exist in Malaysia. However, other comparable aspects that are unforeseen would be the accelerators and incubators that are catering more when the start-up's phase.

The comparison will be used during competing will be the difference of UIPFuture platform and other accelerators and incubators platform. In addition, UIPFuture platform will also be juxtaposed with the Crowdsourcing platform as they invite expertise into various projects.
The UIPFuture platform contributes to academic researchers and lecturers in every university that they are collaborating.

## 1.5 Objectives

The Objectives of the project is evident as below:

a. To crawl various information from the university, government agency and industry websites into Microsoft Excel format.
b. To create a system that acts as a centralised database to gather pieces of universities information.
c. To store the scrap data in Microsoft Excel format. Then extract all Microsoft Excel data into a centralised online database.
d. To provide convenient, efficient and accurate visualisation results to the third party.

### 1.5.1 Project Related Questions

1) What are the quantity and the quality of the resources that can be collected from universities?
2) What will be the main focus of the students when making the decision?
3) Will the platform be accurate for academic researchers when making the decision?
4) How to retrieve information from institutions that is unavailable to the public?
5) How fast will the system respond and finish analysing the results?

### 1.5.2 Project Aims and Sub-Objectives

The project aims to provide a convenient platform for academic researchers' universities research, and uncommercialised patents bring into real life and various industries to be commercialised.

The project sub-objectives are as below:

a. To improve the skills of web scraping websites and extract from codes instead of displays.

b. To bring up the work and contributions of educators' researches.

c. To simplify the workflow and figure out the effectiveness of the platform.

## 1.6 Project Scopes

In this project, UIPFuture platform will be developed. During the development phase, several works and scope in this research will be focused as follow:

### 1.6.1 Dataset

The dataset used in this project is the set of data of some vital information in about 70 public and private higher institutions, industries, and government agencies.

### 1.6.2 Programming Language

The programming languages used in this project are HTML, CSS, JavaScript and Python. HTML, CSS and JavaScript are used in building the user interface; Python is used to scrap data and analysis.

## 1.7 Project Plan

UIP, an AI-as-a-Service (AIaaS) platform, will be built for educators in Universities to convert and commit their areas of expertise into prototypes, curriculum, industry-friendly collaboration models, and develop new research areas with an AI-enabled engine.

UIP will be created and designed to showcase every select UIP researches with various categories into one single platform. All researches will be categorised as Agriculture, Board Reporting, Business and et cetera. In the platform, the user will be able to explore the UIP projects based on their interest.

The project also will undergo 4 phases as below:

### 1.7.1 Phase 1 Research and Prototypes

Various researches are selected and prepared to insert into the platform to have a chance to collaborate with industries and develop a research prototype from the research itself. The researchers will have the opportunity to co-own the prototype when it is produced.

### 1.7.2 Phase 2 Website Scrapping Module

We will scrap all information we need from the Universities, Government agencies, and Industries during this phase. However, understanding more of the connection between the three of them is required to build a highly usable platform.

### 1.7.3 Phase 3 Data Segmentation Module

We will gather all data we got from phase 2 while creating a centralised online database to insert all the data into the platform. The raw data we collected from phase 2 will also convert into various data clusters to become meaningful for us in the project.

### 1.7.4 Phase 4 Dashboard Module

At this phase, we will create a dashboard to export the data that can assist the users. Before being viewed by the specific user, the data will be analysed by various methods to display meaningful insights in different formats such as bar graphs, top-ranking charts, et cetera.

## 1.8 Project Organisation and Timeline

Table 1.8.1: Table of Project Timeline

| Timeline | Mile Stone | | Mile Stone Goals |
|---|---|---|---|
| 16/03/2021 - 30/03/2021 | Web Scraping | Universities Staff Module | Scrap all the related information and produce output in .csv format. |
| 25/05/2021 - 29/05/2021 | | Government Agencies Module | |
| 19/04/2021 - 30/04/2021 | UI/UX development | UIPFuture website Module | Create a website as a platform to display various researches |
| | | UIP projects Module | Gather and Categorised researches which collaborated with MyFinB |
| | Database development | Push all the scraped information into one central database with auto search and filter functions. | |
| | Database UI/UX development | Development of UI/UX for the database to produce information needed and data flowchart. | |
| | Dashboard Design and Development | Design and development of the dashboard of the RoboAdvisor AI-based system. | |

## 1.9 Advantages and Contributions

The finding of this research will benefit the community, such as the universities, industries and investors. The output of this research will be seeking linked relationships and collaboration towards each of the universities. Academic researchers will also find it easy and convenient to obtain the required information. The fund can be obtained when investors find their desired research projects on the proposed platform.

# Chapter 2

# Literature Review

## 2.1 Literature Review


Figure 2.1 UIP Logo

UIP, known as University - Industry Partnerships, is an AI-as-a-Service (AIaaS) platform for educators in Universities to convert their expertise and professionals into various prototypes, curriculum, industry-friendly collaboration models, and develop new areas or research with AI-enabled engine. The research will help UIP expand its coverage and research will be developed into fully-ready AI-based expert systems for industry adoption and commercialisation.

The UIP platform will help the educators design their own personalised AI expert system based on their specialisation and achievements from their education biography, then transform their research prototype into complete commercialisation tools. Moreover, the education community may need not spend enormous resources and credits to complete their prototypes such as infrastructure, human capital expertise, capital expenditure costs, et cetera.

## 2.2 Feasibility Study

### 2.2.1 Technical Feasibility

Web Scraping

To develop this platform, Web Scraping method is vital in getting all sorts of information from various websites. Unlike screen scraping, which only copies pixels displayed on-screen, Web Scraping uses specific bots to extract data and information underlying HTML codes in the websites. It is undeniable that web scraping has been widely used in current digital businesses when they require data harvesting. Once the data is scrapped, it will be stored in a database for future uses.

Data in databases from scrapping ministry websites, agencies, and universities will be used to analyse the contents so the platform can perform some analytic on the specific tasks. Bots types such as ParseHub, Selenium and BeautifulSoup are used for Web Scrapping as they are fully customisable to the following items:

1) Recognise the uniqueness of HTML site structures
2) Easy to extract and transform all sorts of content
3) Assisting to store scrapped data
4) Data from APIs are managed to be scraped and extracted

### 2.2.2 Operational Feasibility

Design thinking is a must process for creative problem-solving in the platform to allow the platform to perform at its best. Design thinking has a human-centred core that quickly encourages organisations to focus on the people they create to lead to better products, services and internal processes. The "standard of procedure" to create a solution with design thinking is to figure out a party's desire from a human point of view with technologically feasible and economically viable.

The challenge with these creative tools is to starts with making the right actions and understanding the right questions. After that, all the technique embraces the most straightforward mindset shifts and tackles the problems from different directions.

Design thinking is best to address problems where multiple spheres of business and society issues, logic and emotions, rational and creative, human needs, and economic demands collide at the intersections between systems and individuals.

To go through the phases of design thinking, we have to go back and forth with the questions to lead us to new innovative solutions. The stages of design thinking such as below:

- Frame a Question. By identifying the driving question which able to inspires others for creative solutions.

- Gather Inspiration. They are inspiring new thoughts to discover what the community need.
- Generate Ideas. Segment past apparent solutions to get ideas breakthrough from a state of point.
- Make Ideas Tangible. By building rough and straightforward prototypes to learn how to improvise and make ideas better.
- Test to Learn. Refining ideas by gathering various feedback from others and then experimenting with it afterwards.
- Share the Story. Craft and create a scenario that can inspire others to be attracted to actions.

## 2.3 Chapter Summary and Evaluation

In this chapter, I talked about the company background and its goals and the mission that got the company started in the first place. We also discussed the project background and why we have proposed this project. A few of the reasons are to provide the education with a way to gather large amounts of data in a platform and analyse them to gather various insights of what resources they could get for their educational research and prototype. This chapter also discussed the uniqueness of the proposed system that we are developing to create an education analysis system that allows the users to provide insights to make better decisions and establish the relationship in the education community.

# Chapter 3

# Methodology and Requirements Analysis

## 3.1 Methodology

Various methodologies used in the education system exists in the market. The process that we are using to apply to this system is the Evolutionary Prototype Model. More will be discussed in the following introduction and fact-gathering methods.

### 3.1.1 Software Development Model

The evolutionary prototype model is the most suitable for this project as the company will develop and refine the system in the future. The prototype model allows us to build, debug, and refine the prototype until it can be defined as an acceptable and commercialised prototype. It also would assist in creating a firm base for the future development of the finalised system.

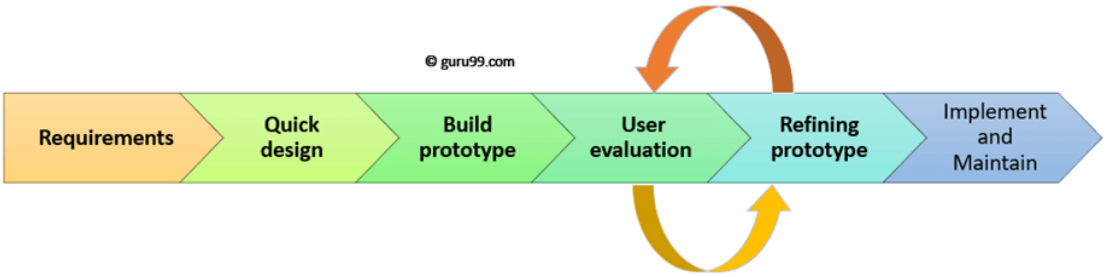



Diagram 3.1.1 The Phase of Prototyping Model

The reason for choosing prototype model methodology is that evolutionary prototyping is a suitable type of prototype for our software development projects. With evolutionary prototyping, the project will be improved and developed better and better day by day with feedback. Moreover, projects involving AI are challenging in framing different specifications; evolutionary prototypes will be much valuable and more accessible to be built in this kind of exploratory programming.

## 3.2 Requirement Analysis

### 3.2.2 External Interface

**Graphical User Interface**

The system's design will probably be towards the use of icons or other visualisable indicators alongside text for more easily understandable of the functions of an icon. The font chosen should quickly catch the attention of the users and be easy to be seen. To avoid being overwhelming, the font could be used, such as Serif (e.g. Times New Roman) and

Mono-space (e.g. Courier). The theme colour used in the user interfaces needs to be eye-pleasing to allow a comfortable screening and not distract users from essential things.

**Hardware Interface**

Desktop and computer will be used to retrieve and receive data information. Therefore, pointing devices such as Mouse and TouchPad are necessary to achieve controllable features for the end-user. In addition, A stable internet connection is required because the system developed is a web application system.

**Software Interface**

The web browser should install the later version to support the latest version of the web application, which python3.8 and HTML5 supported.

### 3.2.3 Functional Requirements

Web Scraping Module

1. This module needs to scrap efficiently with the specific keywords based on requirements and store them into the data after scraping.

Data Pattern Analysis Module

1. This module needs to filter the information which is useless to the user.
2. This module needs to sort the information in different formats to allow users to observe from different views quickly.
3. This module needs to be able to form various insights for the users.

Data Mining Module

1. This module needs to be able to extract a group of information into excel or csv format.

Data Segmentation Module

1. This module needs to segment raw data into a group of data clusters for different purposes.
2. This module needs to be able to store the real-time segmented data into a database.

Dashboard Module

1. This module needs to export data from the database and convert it into an editable format for the users.
2. This module needs to allow users to have the right of manipulating the data features.

### 3.2.4 Non Functional Requirements

- Correctness
  - The information achieves from the web scraping module must be the same as the data shown on the websites.
  - The web scraping module must always detect the latest update of the websites and scrap down the newest information from the websites.
  - The information used for data mining must be the same as the information in excel format.
  - The data mining module must not leave out any information that is from the excel file.
- Readability
  - The information scrap from the website must be filtered into a readable format for the users.
  - The analysed result from the dashboard module must be shown in a readable format to the users.
- Reliability
  - The system must not be losing connection more than 0.5% of the day.
  - With the massive information in the system, the user-interface display must show various details in an easy-to-understand way for the users.
- Usability

- The system must provide guidance or tutorial to lead the users to understand the mechanism of the system. Thus, users can have fast progress for their final results with guidance.
- The system needs to be created with various simple modules and avoid complex modules to confuse users.

- Maintainability
  - The structure of programming needs to be understandable by inserting various simple explanations in comments. Hence, future developers easily maintain and upgrade the system.
- User-Friendly
  - The system's user interface needs to assist the users to quickly understand and simplify various important information from massive information on the display.

### 3.2.5 Development Environment

Python programming language will be used as the development language for this project. The python language will help us scrape out specific information from websites based on keywords. Data extracted from excel files also will be assisted with Python language. This project will be developed in web applications and used by devices such as Desktop PC and laptops.

**Programming Language**

1. Python 3.9
2. HTML/CSS
3. JavaScript

**Other Software**

1. Microsoft Excel
2. Visual Studio Code
3. Google Colab

**Operating System**

1. Microsoft Windows 10

### 3.2.6 Operation Environment

**Desktop / Laptop**

Table 3.2.6.1: Table of Desktop/Laptop Environment

| Minimum Hardware and Software Requirement | |
|---|---|
| Processor (CPU) | Intel Core i3 and above |
| Operating System | Windows 7 and above |
| Memory | 4GB RAM or Higher |
| Storage | 4GB or higher space available |
| Other non-operating-system programs | Web Browser such as Internet Explorer version 6.0 or Google Chrome |
| Required Equipment | Mouse and Keyboard |
| Database | - |
| Development Tools | Visual Studio Code, Google Colab |

## 3.3 Chapter Summary and Evaluation

In this chapter, we have explained the reason for implementing prototyping as our software development model. Furthermore, the development environment and operational environments have displayed the minimum requirement of hardware and software components that will be used to develop this proposed project. Last but not least, non-functional requirements and functional requirements are listed, which can help the reader to understand the functions of the system more accessible.

# Chapter 4

# System Design

## 4.1 User Interface(UI) Design

We planned to develop UIPFuture as a web application. This web application will use the colour green and white as the system theme because green brings quietness, trust, and confidence towards the web application. It brings calm feelings of renewal for the user. With the white theme, the UI's background would allow users to pay more attention to the interface due to its simplicity, and it brings the feeling of wholesomeness towards the whole web application. This primary theme colour allows users to feel comfortable when viewing the entire interface. It is also the most suitable theme for allowing users to think calmly and read every interested research to make every decision.

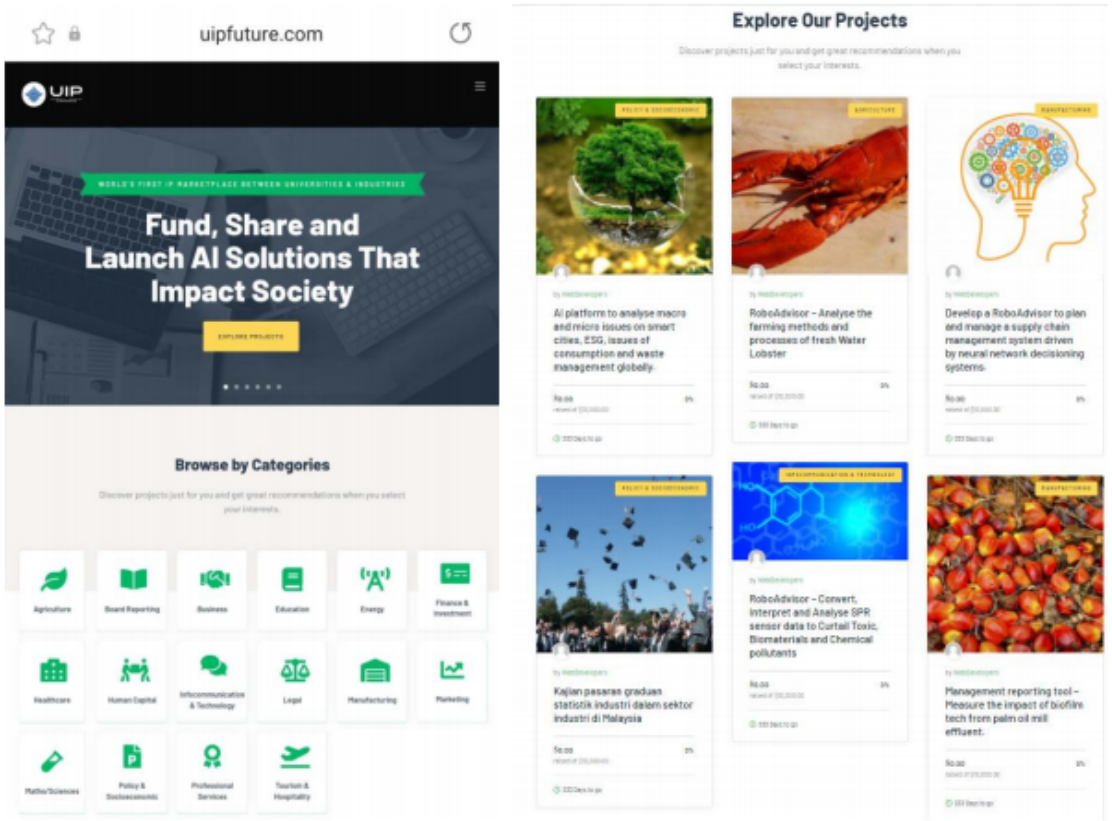



Figure 4.1.1 Main Page

Figure 4.1.1 show the main page of our UIPFuture. This page will show the logo, navigation bar from the header and every category of the researches.


Figure 4.1.2 UIPFuture Logo

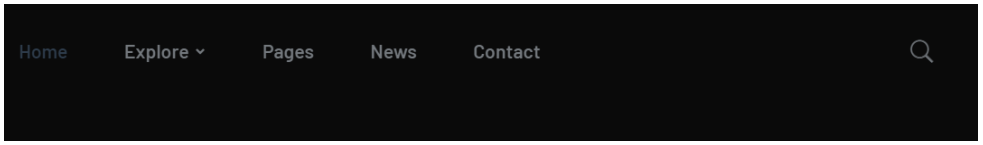



Figure 4.1.3 UIPFuture Navigation Bar

The logo and navigation bar will be shown in the header of the website. The user will be directed to the main page with the "Home" button shaded. Other than that, users can navigate to other pages from the navigation bar, such as the researches page from Explore, Pages, news related education from news and contact us for any queries from contact.

## 4.2 Security Design

### Login

The data provided by users such as students, lecturers, and industries that will be used in the education analysis is much valuable. Hence, the data will be secured and protected to avoid data leakage to any other outsiders. We created a login function for every user to protect the data so data will be tracked with user's ID if leaked happens. Users also need to log in to UIPFuture to explore the website and use the dashboard anytime as long as the ID and password remain. Therefore, those users without ID will not steal and leak any information from the website.

## 4.3 Chapter Summary and Evaluation

In this chapter, we presented the user interface design of our UIPFuture look like, so the users could preview the system to determine satisfaction. We also explained the security design applied by us for the system to avoid data breaches and leakage.

# Chapter 5

# System Implementation and Testing

## 5.1 Implementation and Testing

In this chapter, the implementation, and testing of this prototype will be discussed in details in this chapter. The implementation of the prototype will be discussed which aims to develop various features to assist in achieving the initial objective of the project. While, the testing for the prototype will be discussed in this chapter which aims to self-evaluate whether the prototype hit the user requirement and features implemented successfully.

This chapter will be summarized as follows:

- Coding Implementation
- Testing
- Software Methodologies
- Test Cases

## 5.2 Coding Implementation

The prototype are integrated with a few different modules listed as below:

- Data Scraping Module
- Data Gathering Module
- Data Cleaning Module
- Sort and Filter Data Module
- User Interface Module

### 5.2.1 Data Scraping Module

In this module I was incharge in scraping academic universities staff information and also government ministry information. The method that were used for web scraping are the Beautiful Soup and Selenium.

5.2.1.1 BeautifulSoup Data Scraping (BSDS)

By using beautiful soup for data scraping, various library are needed to install before continuing the following tasks. Library that were used in this part are the requests, bs4, csv, and pandas which are shown in Figure 5.1.

```
import requests
from bs4 import BeautifulSoup
import csv
import pandas as pd
```

Figure 5.1: Library used in BSDS

After import the libraries, the module will begin with creating an variable(URL) to input the link of the web pages that I would like to scrap with, then insert it into the the requests library to fetch the page content. Then, it will create a parse Tree object(soup) with the assistance of BeautifulSoup and Python built-in "html" parser.

```
URL = "https://www.kbs.gov.my/pej-menteri-kbs"
page = requests.get(URL)

soup = BeautifulSoup(page.content, "html.parser")
```

Figure 5.2: Fetching html parser in BSDS

In the website HTML page, I have to find the container's attribute that have all the data needed to scrapped. In this specific website, the data I needed are in the div container with the value of "jsn-mainbody". After parsing the div container into the variable (results), I can just find the elements I need by mentioning the placement of the data and the class name to get closer to the data I would like to collect.

```
#find div id
results = soup.find(id="jsn-mainbody")
#print(results.prettify())

job_elements = results.find_all("tr", {"class": ["fabrik_row oddRow0", "fabrik_row oddRow1"]})
```

Figure 5.3: HTML container in BSDS

After specify the direction of the data needed to collect, I created an array variable (gov_list) to store all of the data that I will collect soon. Then I create a loop to find all the data with the class name and store all into the each variable (name_element, jawatan_element, cawangan_element, tel_element, emel_element) . Before storing into an array (gov), I did some data cleaning by removing and stripping off some blanks spaces and left only words, also inserting the email format "@kbs.gov.my" into every emel_element. Lastly, I will append the gov variable into the gov_list variable that we created earlier.

```
gov_list=[]

for job_element in job_elements:
    nama_element = job_element.find("td", class_="direktori_staf_kbs___nama fabrik_element fabrik_list_1_group_1")
    jawatan_element = job_element.find("td", class_="direktori_staf_kbs___jawatan fabrik_element fabrik_list_1_group_1")
    cawangan_element = job_element.find("td", class_="hidden-phone direktori_staf_kbs___cawangan_unit fabrik_element fabrik_list_1_group_1")
    tel_element = job_element.find("td", class_="direktori_staf_kbs___no_telefon fabrik_element fabrik_list_1_group_1")
    emel_element = job_element.find("td", class_="hidden-phone direktori_staf_kbs___emel fabrik_element fabrik_list_1_group_1")
    nama_element.text.strip()
    jawatan_element.text.strip()
    cawangan_element.text.strip()
    tel_element.text.strip()
    emel_element.text.strip() + '@kbs.gov.my'
    gov = [nama_element, jawatan_element, cawangan_element, str(tel_element), emel_element]
    gov_list.append(gov)
```

Figure 5.4: Storing data scrapped in BSDS

Lastly, all the data in gov_list will be output into a csv file (BStest.csv) for better visualization and storing purposes.

```
f= open('BStest.csv', 'w')
csv_writer = csv.writer(f)
for i in gov_list:
    csv_writer.writerow(i)

f.close()
```

Figure 5.5: Data output to csv in BSDS

5.2.1.1 Selenium Data Scraping (SDS)

By using selenium for data scraping, various library are needed to install before continuing the following tasks. Library that were used in this part are the selenium, csv, numpy, and pandas which are shown in Figure 5.6.

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
import pandas as pd
import numpy as np
import csv
```

Figure 5.6: Library used in SDS

After import the libraries, a master dataframe (df) are created in the beginning before scrapping and webdriver will be launching the Chrome in headfull mode controlled by the Python programming while browsing the website inserted in the GET function.

```
#master Dataframe
df=pd.DataFrame(columns=['Name','Position','Department','Tel No.','Email'])

driver = webdriver.Chrome('C:\chromedriver')
driver.get('https://www.kbs.gov.my/pej-menteri-kbs')
```

Figure 5.7: Browsing website in webdriver in SDS

To extract out the data from the website, I first inspect the html format in the website, find the elements of the data, and store it with the method of XPath.

```
names = driver.find_elements(By.XPATH, '//td[@class="direktori_staf_kbs___nama fabrik_element fabrik_list_1_group_1"]')
positions = driver.find_elements(By.XPATH, '//td[@class="direktori_staf_kbs___jawatan fabrik_element fabrik_list_1_group_1"]')
departments = driver.find_elements(By.XPATH, '//td[@class="hidden-phone direktori_staf_kbs___cawangan_unit fabrik_element fabrik_list_1_group_1"]')
tels = driver.find_elements(By.XPATH, '//td[@class="direktori_staf_kbs___no_telefon fabrik_element fabrik_list_1_group_1"]')
emails = driver.find_elements(By.XPATH, '//td[@class="hidden-phone direktori_staf_kbs___emel fabrik_element fabrik_list_1_group_1"]')
```

Figure 5.8: XPath method in SDS

Array variable (names_list, positions_list, departments_list, tels_list, emails_list) also created to store all the elements (names, positions, department, tels, emails) found. Then the data will be store into a list to combine all the data together.

```
names_list = []
for n in range(len(names)):
    names_list.append(names[n].text)

positions_list = []
for n in range(len(positions)):
    positions_list.append(positions[n].text)

departments_list = []
for n in range(len(departments)):
    departments_list.append(departments[n].text)

tels_list = []
for n in range(len(tels)):
    tels_list.append(tels[n].text)

emails_list = []
for n in range(len(emails)):
    emails_list.append(emails[n].text + '@kbs.gov.my')

#list of each gov names, position, department, contact, email
data_tuples = list(zip(names_list[1:], positions_list[1:], departments_list[1:], tels_list[1:], emails_list[1:]))
```

Figure 5.9: Storing data scrapped in list in SDS

List will then being convert into dataframe for easy storing into csv then being output into a csv file (Seleniumtest.csv).

```
#create Dataframe of each tuple
temp_df = pd.DataFrame(data_tuples, columns = ['Name','Position','Department','Tel No.','Email'])

#appends to master dataframe
df = df.append(temp_df)
arr = df.to_numpy()
print (arr)

driver.close()

f = open('Seleniumtest.csv', 'w')
csv_writer = csv.writer(f)
for i in arr:
    csv_writer.writerow(i)

f.close()
```

Figure 5.10: Data output to csv in SDS

### 5.2.2 Data Gathering Module (DGM)

In this module, I was incharge in gathering the categorized data from industry projects into one combined data table.

Firstly, I imported the pandas library to read every csv file into the Jupyter Notebook

```
import pandas as pd
agridata = pd.read_csv('Agriculture.csv')
infodata = pd.read_csv('Info&Tech.csv')
legaldata = pd.read_csv('Legal.csv')
manudata = pd.read_csv('Manufacturing.csv')
marketdata = pd.read_csv('Marketing.csv')
mathdata = pd.read_csv('Math_Sciences.csv')
policydata = pd.read_csv('Policy & Socioeconomic.csv')
professdata = pd.read_csv('Professional Services.csv')
tourismdata = pd.read_csv('Tourism & Hospitality.csv')
```

Figure 5.11: Read csv in DGM

Since the header for some data columns are not understandable, I have to edit the column name into a more understanding names for the convenience of data analysis in the future. Each file had been edited into the same column order and names for the next procedure which is to append and combine all the file into one easily.

```
agridata['Category'] = 'Agriculture'
del agridata['pic']
agridata.rename(columns={'description MyFinB Role': 'MyFinB'}, inplace=True)
agridata.rename(columns={'Full Name for Topic': 'Full Topic'}, inplace=True)
agridata.rename(columns={'Tech': 'Science & Technology Driver'}, inplace=True)
agridata.rename(columns={'Driver': 'Socio-Economic Driver'}, inplace=True)
agridata.head()
```

Figure 5.12: Change column name in DGM

After all the file names are edited, a frame is created to insert all the data into one variable and manipulate the axis and position of the list into dataframe by using concat function.

```
frames = [agridata, infodata, legaldata, manudata, marketdata, mathdata, policydata, professdata, tourismdata]


fulldf = pd.concat(frames)
fulldf.head(30)
```

Figure 5.13: Combine data in DGM

Lastly, the dataframe will be stored into a csv file (FYPProject.csv) and import to google drive as cloud storage.

```
fulldf.to_csv('FYPProject.csv', index = False)


from google.colab import  drive
drive.mount('/drive')
fulldf.to_csv('/drive/My Drive/FYPProject.csv', index = False)
```

Figure 5.14: Store csv into drive in DGM

### 5.2.3 Data Cleaning Module (DCM)

Similar to the previous module, where I am in charge in the same part which is the industry project where I start to clean the data for better analysis results when comes to data analytic.

As usual, I import the pandas library to read the csv file from google drive.

```
import pandas as pd
df = pd.read_csv('FYPProject.csv')
```

Figure 5.15: Read data in DCM

I started to clean the data by removing the pending project which does not yet exist or not progressing in the current. Then resetting the index of the data for better looking. I also then remove the useless symbol in the data itself to be convenient when categorizing the data during the analysis phase.

```
# Filter all rows where project still pending
df.drop(df[df['MyFinB'] == 'Pending'].index, inplace = True)


fypdata = df.reset_index(drop=True)
```

Figure 5.16: Remove pending project in DCM

```
for r in (("-", ""), ("\n", " , ")):
  fyp1['Socio-Economic Driver'] = fyp1['Socio-Economic Driver'].str.replace(*r)
  fyp1['Science & Technology Driver'] = fyp1['Science & Technology Driver'].str.replace(*r)
```

Figure 5.17: Remove useless symbol in data in DCM

Lastly, I store the data into a csv file (CleanFYP.csv) into google drive as well.

```
fypdata.to_csv('CleanFYP.csv', index = False)


from google.colab import  drive
drive.mount('/drive', force_remount=True)
fypdata.to_csv('/drive/My Drive/CleanFYP.csv', index = False)
```

Figure 5.18: Store data in drive in DCM

### 5.2.4 Sort and Filter Data Module (SFDM)

In this module, I also take charge of the Industry Project part for simulating and creating the sorting and function feature.

As usual, pandas library are imported to allow python to read the csv file (CleanFYP.csv)

```
import pandas as pd
df = pd.read_csv('CleanFYP.csv')
```

Figure 5.19: Read csv file in SFDM

Filter function are created in a way that it will allow users to input the specific keyword they would like to filter and which column of the data they want to use the filter.

```
#Search specific words
word = input("Enter keyword: ")
```

```
Enter keyword: Bioscience
```

```
abc = df[df['Science & Technology Driver'].str.contains(word)]
```

Figure 5.20: Filter function in SFDM

While the sort function, is created in a way that it will receive the input from the user which column they would like to sort and decide the order they would like to sequence, either ascending or descending.

```
#Sort by words
sortword = input("Enter column to sort: ")
rowseq = input("Ascending or Descending? :(A/D)")
```

```
Enter keyword to sort: Topic
Ascending or Descending? :(A/D)a
```

```
if (rowseq == 'A' or rowseq == 'a'):
  seq = True
else:
  seq = False
```

```
rslt_df = df.sort_values(by = sortword, ascending = seq)
```

Figure 5.21: Sort function in SFDM

### 5.2.5 User Interface Module (UIM)

In this module, I am responsible for creating the user interface for the university data.

The website begins with a simple header with the title of Academic Search and continues with a search function for the universities table.

```
<h1 style="text-align: center; color: black;">Academic Search</h1>

<br>

<p>Key in your input to filter the table:</p>
<input type="text" id="myinput" placeholder="Search..." title="Type in something">
```

Figure 5.22: Coding of design in UIM

Variable (tableData) are function to simulate the database and created in the javascript which it is also connected with the html.

```
var tableData = [
    {university: 'University Teknologi Malaysia', department: 'Fakulti Alam Bina & Ukur', position: 'Pensyarah Kanan', name: 'Dr. Abdul Rahim Bin Abdul Hamid', email: 'abdul.rahim@utm.my'},
    {university: 'University Teknologi Malaysia', department: 'Fakulti alam bina & ukur', position: 'PENSYARAH KANAN (DS52)', name: 'Sr Dr. Abd Wahid Bin Rasib', email: 'abdwahid@utm.my'},
    {university: 'University Teknologi Malaysia', department: 'Fakulti alam bina & ukur', position: 'PENSYARAH KANAN (DS52)', name: 'Gs. Dr. Abd. Halim Bin Hamzah', email: 'halimhamzah@utm.my'},
    {university: 'University Teknologi Malaysia', department: 'Fakulti alam bina & ukur', position: 'PENSYARAH KANAN (DS51)', name: 'Dr. Abdul Rahim Bin Abdul Hamid', email: 'abdul.rahim@utm.my'},
    {university: 'University Teknologi Malaysia', department: 'Fakulti alam bina & ukur', position: 'PROFESOR MADYA (DS54)', name: 'Prof. Madya Sr. Abdul Wahid Bin Kamarulzaman', email: 'ab_wahid@utm.my'},
    {university: 'University Teknologi Malaysia', department: 'Fakulti alam bina & ukur', position: 'PENSYARAH KANAN (DS52)', name: 'Sr Dr. Abdullah Hisam Bin Omar', email: 'abdullahhisham@utm.my'},
```

Figure 5.23: Table data backend in UIM

In the user interface itself, the appearance of the website to the users is as shown below.

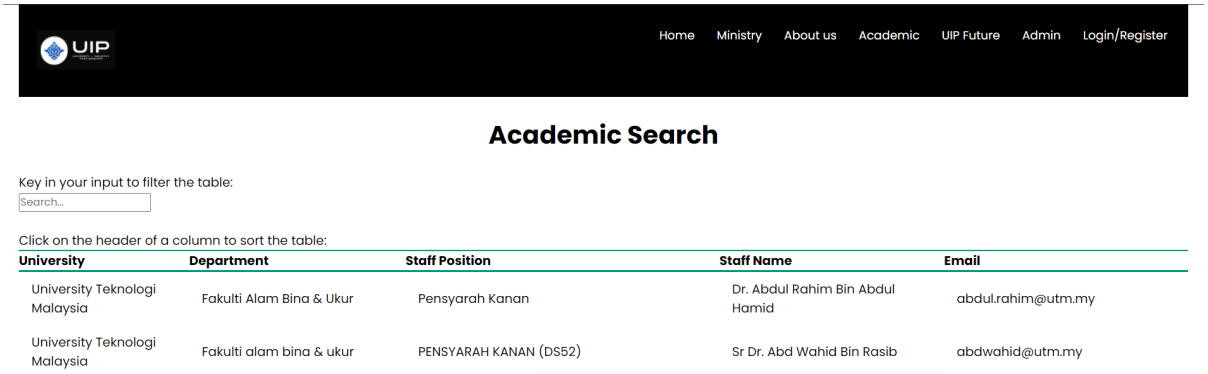



Figure 5.24: Frontend design in UIM

When users search the specific keyword, it will activate the filter function and the specific data will be highlighted to alert the user.

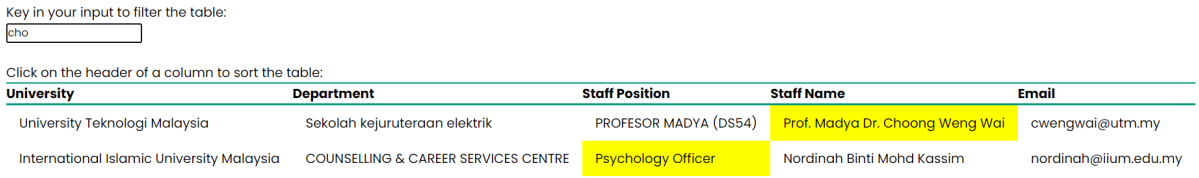
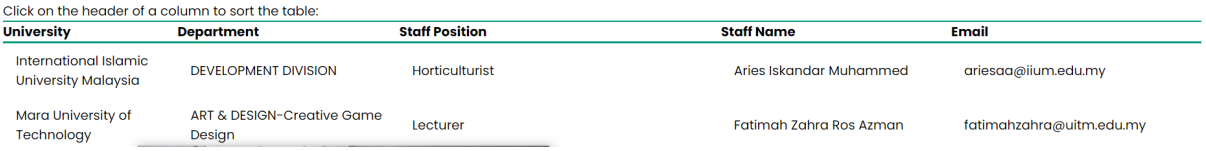


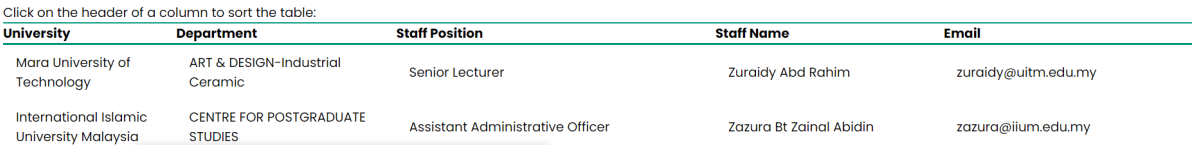

Figure 5.25: Highlight when filter in UIM

However, the user will need to click the table column name to use the sort function in the order of ascending or descending.


Click on the header of a column to sort the table:

| University | Department | Staff Position | Staff Name | Email |
|---|---|---|---|---|
| International Islamic University Malaysia | DEVELOPMENT DIVISION | Horticulturist | Aries Iskandar Muhammed | ariesaa@iium.edu.my |
| Mara University of Technology | ART & DESIGN-Creative Game Design | Lecturer | Fatimah Zahra Ros Azman | fatimahzahra@uitm.edu.my |

Figure 5.26: Ascending order when sort in UIM


Click on the header of a column to sort the table:

| University | Department | Staff Position | Staff Name | Email |
|---|---|---|---|---|
| Mara University of Technology | ART & DESIGN-Industrial Ceramic | Senior Lecturer | Zuraidy Abd Rahim | zuraidy@uitm.edu.my |
| International Islamic University Malaysia | CENTRE FOR POSTGRADUATE STUDIES | Assistant Administrative Officer | Zazura Bt Zainal Abidin | zazura@iium.edu.my |

Figure 5.27: Descending order when sort in UIM

## 5.3 Software Testing Methodologies

Software Testing is the procedure of evaluating and verifying the particular software product or application does what it is supposed and expected to do, also satisfy the user requirements(IBM, n.d.). This process is very important and required during every system development as it could prevent bugs issues, reduce excessive development costs, and improve performance in the system before it is published to the public. In system testing, it is categorized as Functional Testing and Non-functional Testing.

Functional Testing involves testing the system application against the business requirements and guaranteeing each part of the software will behave as expected by using several types of test cases (Aebersold, n.d.). For instance, Unit testing, Integration Testing, and System Testing are used to test the functions and incorporate every module and function in the system.

However, Non-functional Testing is a method that incorporates every functional testing but mainly focuses on the operational aspects of a piece of software (Aebersold, n.d.). For exemplification, Performance testing to test the software performance, Security testing aimed to secure the system and data information in it, Usability testing to measure the user-friendliness, and Compatibility testing to gauge the application working in each different environment.

## 5.4 Test Cases for Data Scraping Module

| | |
|---|---|
| **Test Case ID: TC_01** | **Test Designed by: Kong Mun Jun** |
| **Test Priority (Low/Medium/High): Low** | **Test Designed date: 20-10-2021** |
| **Module Name: Data Scraping** | **Test Executed by:** |
| **Test Title:** Verify the data scraping process | **Test Execution date:** |
| **Description:** To test whether the data crawled are correct in each columns | |

| |
|---|
| **Pre-conditions:** |
| **Dependencies:** |

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|---|---|---|---|---|---|---|
| 1 | Enter python file | | | | | |
| 2 | Upload the excel file | gov.csv | Uploaded and not missing any data | | | |

## 5.5 Test Cases for Data Grouping Module

| | |
|---|---|
| **Test Case ID: TC_02** | **Test Designed by: Kong Mun Jun** |
| **Test Priority (Low/Medium/High): Med** | **Test Designed date: 20-10-2021** |
| **Module Name: Data Grouping** | **Test Executed by:** |
| **Test Title:** Verify the data grouping process | **Test Execution date:** |
| **Description:** To test whether all the data are combined | |

| |
|---|
| **Pre-conditions:** connected to Google Drive |
| **Dependencies:** |

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|---|---|---|---|---|---|---|
| 1 | Standardize the columns in every file | “Tech” to “Technology” | Combined data in correct column | | | |
| 2 | Upload the excel file | GroupFYP.csv | Uploaded and not missing any data | | | |

## 5.6 Test Cases for Data Cleaning Module

| | |
|---|---|
| **Test Case ID: TC_03** | **Test Designed by:** Kong Mun Jun |
| **Test Priority (Low/Medium/High): Med** | **Test Designed date: 20-10-2021** |
| **Module Name: Data Cleaning** | **Test Executed by:** |
| **Test Title:** Verify the data cleaning and process | **Test Execution date:** |
| **Description:** To test whether the data cleaned and processed are accurate | |

**Pre-conditions:** connected to Google Drive
**Dependencies:**

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|---|---|---|---|---|---|---|
| 1 | Remove meaningless data | Project-"Pending" | Data removed | Data successfully removed | | |
| 2 | Remove useless symbol | "-Water & Food\n" | "\n" replace to " , " | Symbol successfully removed | | |
| 3 | Upload the excel file | CleanFYP.csv | Data uploaded and cleaned | | | |

## 5.7 Test Cases for Sort&Filter Function Module

| | |
|---|---|
| **Test Case ID: TC_04** | **Test Designed by: Kong Mun Jun** |
| **Test Priority (Low/Medium/High): Med** | **Test Designed date: 20-10-2021** |
| **Module Name: Data Sort&Filter** | **Test Executed by:** |
| **Test Title:** Verify the sorting and filter function | **Test Execution date:** |
| **Description:** To test whether the data able to sort or filter based on conditions | |

**Pre-conditions:** connected to Google Drive
**Dependencies:**

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|---|---|---|---|---|---|---|
| 1 | Input specific columns | MyFinB | Column exist | Column exist | | |
| 2 | Input ascending or descending | a | Column in ascending order | Column in correct order | | |
| | | d | Column in descending order | Column in correct order | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | Input specific keyword | Bioscience | Rows with “Bioscience” keyword | Row successfully appeared | | |

## 5.8 Test Cases for UI Website Module

| | |
|---|---|
| **Test Case ID: TC_05** | **Test Designed by: Kong Mun Jun** |
| **Test Priority (Low/Medium/High): Med** | **Test Designed date: 20-10-2021** |
| **Module Name: UI test cases** | **Test Executed by:** |
| **Test Title:** Verify the UI feature | **Test Execution date:** |
| **Description:** To test whether the UI can provide the correct feature to the users. | |

| |
|---|
| **Pre-conditions:** |
| **Dependencies:** |

| Step | Test Steps | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|---|---|---|---|---|---|---|
| 1 | Click into the module | Click from upper tab | Able to direct to Module | Module appeared successfully | | |
| 2 | Input value to search box | Mara | Rows with “Mara” appear | Rows successfully executed | | |
| 3 | Filter data in table | Clickable header column | Data reorder in alphabetic order | Data successfully reorder | | |

## 5.9 Chapter Summary and Evaluation

Software testing is a vital process and play as an important role to sustain the quality of the system before it is publish to third party. It is a part of the development cycle of a system project to identify every defects and issues in the system to achieve the user requirement expected by third parties such as clients, stakeholders, developers, and testers.

In the nutshell, the testing will be assigned to assure the prototype has minimum deficiency and achieve further improvisation.

# Chapter 6

# System Deployment

## 6.1 System Specification

Throughout the whole process of system development and documentation process, the prototype was completed by the author with his laptop. The following specification is the specifications of the laptop:

**HP Pavilion 15**

- **Processor:** Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
- **RAM:** 20.0 GB (19.9 GB usable)
- **System Type:** 64-bit operating system, x64-based processor
- **VGA:** NVIDIA GeForce MX150
- **Hard Disk:** 1TB
- **SSD:** 225GB

In order to complete the prototype, this is the software used:

- **Python**

  Python is used as the development environment for this project. It is mainly used to assist the project with the process of web scraping data information by keywords and extracting the data out for further tasks from every excel file. All the extracted data will be gathered and cleaned with the process of data cleaning. The python software used the desktop PC or laptops as the development platform.

- **Visual Code Studio**

  Visual Code Studio is an open-source code editor application where it allows multiple languages to be installed and used. For example, Visual Code was used to install python to be used in this project. It also allows different languages to be integrated together.

- **Google Colab**

  Google Colab is an open-source code editor as well with the simulation of Jupyter notebook environment that runs entirely free in the cloud. It does not require any setup or installation in order to start using it for projects as Colab is able to support most of the popular machine learning libraries which are able to load easily in the notebook(tutorialspoint, n.d.).

- **Programming Language**
  1. Python 3.9
  2. HTML5/CSS
  3. JavaScript

- **Other Software**
  1. Microsoft Office 365 Words
  2. ParseHub
  3. Visual Studio Code

- **Operating System:** Microsoft Windows 10

## 6.2 Minimum Software and Hardware Specification

Table 6.2.1: Table of Software and Hardware Specification

**Desktop/Laptop**

| Minimum Software and Hardware Specification | |
|---|---|
| Processor | Intel Core i3 or above |
| Operating System | Windows 7 or above |
| Memory | 4GB RAM or higher |
| Storage | 4GB or Higher space the available |
| Other non-operating-system programs | Web Browser such as Microsoft Edge or Google Chrome |
| Required equipment | Mouse and keyboard |
| Development Tools | Visual Studio Code 1.62 |

## 6.3 Deployment Steps

### 6.3.1 On-site Deployment

This project is run with Visual Studio Code and frameworks and drivers are required to install before the deployment. The version of python used in Visual Studio Code is the latest version that is Python 3.9.7.

In order to run and test the prototype on other computers, there are a few procedures that are needed as the following:

- Visual Studio Code to execute the codes
  - Libraries required to install: selenium, bs4, requests, pandas, numpy, csv in python terminal with the following format “pip install [libraries]”

### 6.3.2 Installation and Configuration

#### 6.3.2.1 Web Scraping Module

1. Check the version of web browser in google chrome by clicking the hamburger button at the upper right, navigating to the help button then lastly‘About Google Chrome’

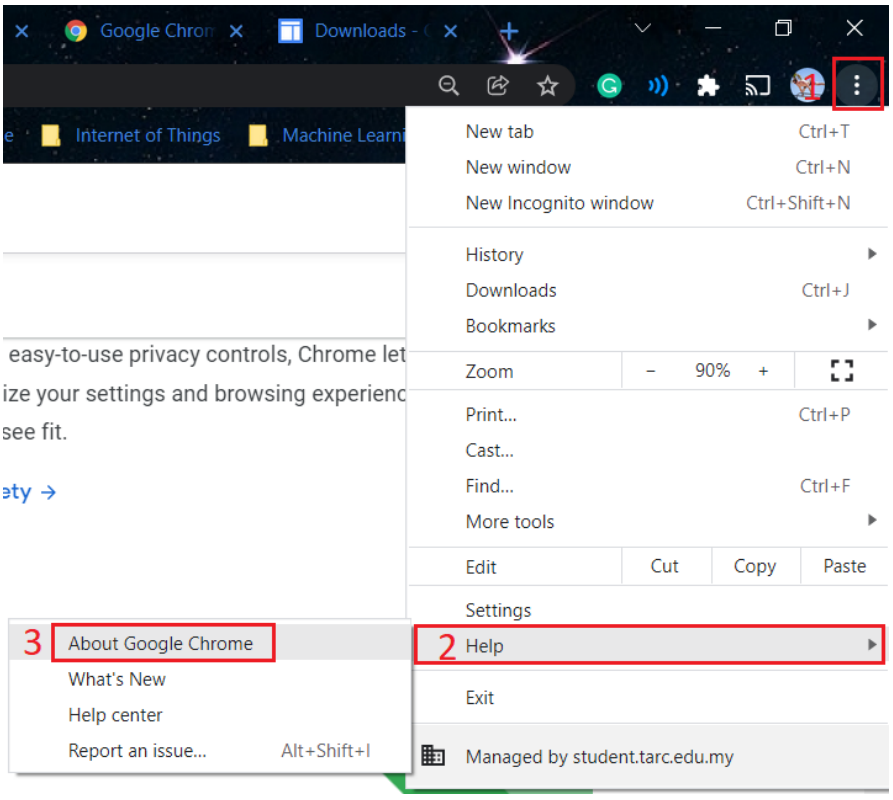



Figure 6.1: Google Chrome Version Check

2. Download the driver from this link: ChromeDriver - WebDriver . You are required to proceed the step 1 in order to install the correct version of webdriver.

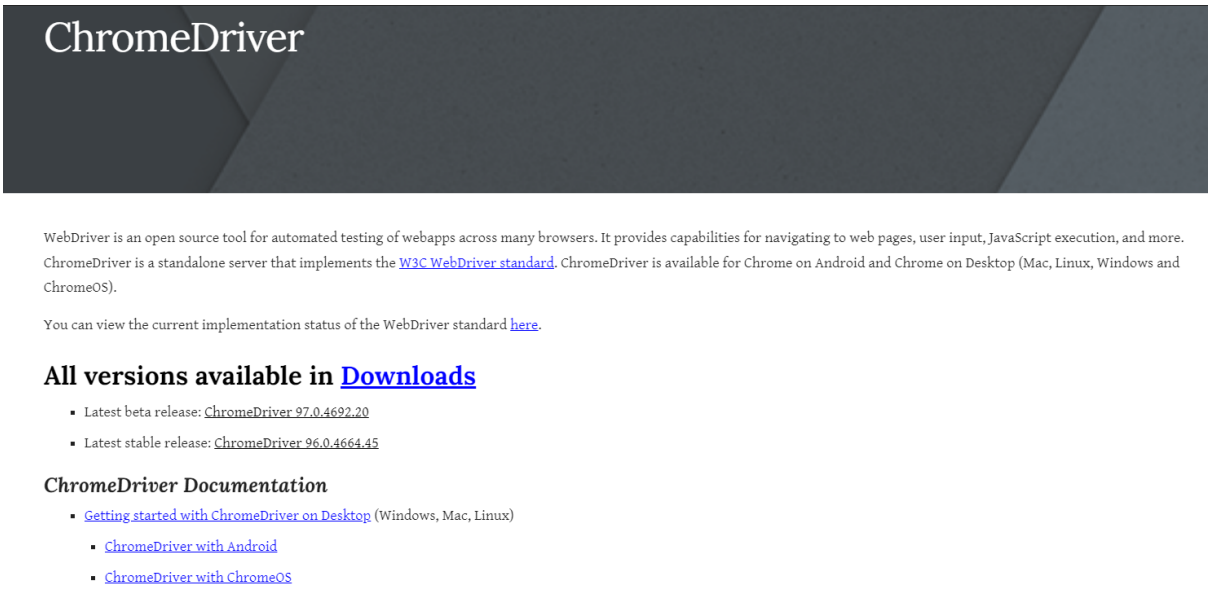



Diagram 6.2: Google Chrome WebDriver Installation

3. Use pip install to install selenium, bs4, requests one by one

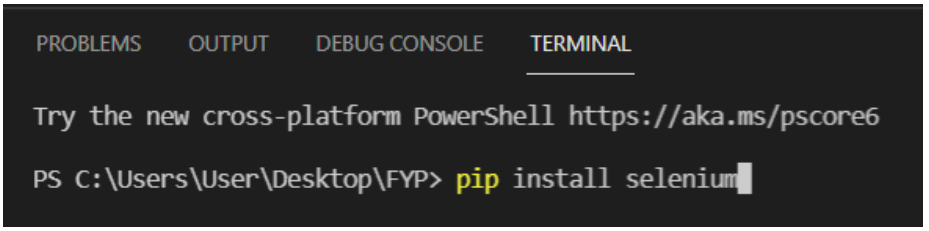



Diagram 6.3: selenium installation


Diagram 6.4: beautifulsoup4 installation


Diagram 6.5: requests installation

## 6.4 Chapter Summary and Evaluation

In this chapter, Software and Hardware requirements as well as installation and configuration are listed and needed to be a references in order to successfully deploy the system. Procedures and guidance in aiding the installation and configuration were also listed down while various comments in the codes existed to allow third-party easily understood the project. Software tools, programming languages, other software, and operating system are also listed in this chapter as a reference.

# Chapter 7

# Discussion and Conclusion

# 7 Discussion and Conclusion

This chapter aim to evaluate the project and explain the projects in terms of summary throughout the whole period of the project. Also, this chapter will introduce conclusions to wrap up the project.

## 7.1 Summary

Initially, the students, parents, universities, governments, industries, and experts face some difficulties which are information from every sort of website related to academics are mostly fragmented and hardly connect to each other. This issue leads to time-consuming for each party when they would like to compare or filter every academic data to make the last decision in their actions and future. Hence, developing a plan to gather all academic-related data information together can play a vital role in providing quick and accurate information and analysis.

Therefore, this proposed project is able to overcome the problem that is currently faced by them which is developing a centralized database to gather all information from every website such as industry websites, universities websites, and government websites and output the results in a faster and accurate method. Besides, the database system is also able to show the dashboards from previous data to give a reference to the management.

During the system development cycle, the model decided to use for building the project is prototyping. This is a matter of fact that every member involved and contributed to the project has a limited time period to achieve the final user requirements of the project. Due to the new and not-exist-yet idea for this project, each of us barely understand the needs of the users and hard to achieve it due to the limitation of knowledge, thus, the throwaway prototype is proposed to demo the solution for the users to receive more feedback from the users for further improvisation.

During the development stage, the Google Colab, Visual Studio Code, and ParseHub, these software tools are decided to use in the development of the system. These tools are decided to be used as the Google Colab have all the import library stored for Python programming language and act as a better simulated Jupyter Notebook; the Visual Studio Code is also a free open source platform which supports ample programming languages; the ParseHub also an open-source platform which assists web scrapper beginner to finish their task easily.

The testing approach in this project is known as the black box testing, it allows the user to test the system without consuming time in understanding the code while the developer team is able to analyze the user behavior for further improvisation in the system.

## 7.2 Conclusion

In overall, UIPFuture, the centralized database system is considered a successful system which had achieved hitting the requirements of the users. It has all the data information and also the features and functions required by the users. The UI design is also created in a simpler way which shows the user-friendliness in the system itself.

However, limitations of the systems and issues did appear in this new prototype system and enhancement can be applied in the future. Fortunately, the system is still able to fulfill the user requirement with the limitations in the system.

## References/Bibliography


1. Augustyn, A., 2021, supply and demand | Definition, Example, & Graph. [online] Encyclopedia Britannica, Viewed on 17 March 2021 <https://www.britannica.com/topic/supply-and-demand>.
2. Beacon - Main Site, 2019, Higher Ed Sites: Understanding Your Most Common Target Audiences, Part I: The Millennial, viewed on 14 Sep 2021, <https://www.beacontechnologies.com/blog/2015/06/higher-ed-sites-understanding-your-most-common-target-audiences-part-i-the-millennial.aspx#:~:text=Given%20this%20definition%2C%20higher%20education>. [Accessed 14 Sep. 2021].
3. Bishop, R., 2021, Why Alumni Are Important for Higher Ed Institutions | Signal Vine. [online], viewed on 17 March 2021, <https://www.signalvine.com/alumni/why-alumni-are-important-for-higher-ed-institutions#:~:text=Alumni%20serve%20many%20valuable%20roles,and%20career%20opportunities%20to%20students.>
4. Hull, M., n.d., 4 Types of Prototyping, viewed on 06/09/2021, <https://www.andplus.com/blog/4-types-of-prototyping>.
5. IDEO U, 2017, What is Design Thinking? [online] IDEO U, viewed 06 Sep 2021, <https://www.ideou.com/blogs/inspiration/what-is-design-thinking>.
6. Love, J., 2021, Leaderonomics.com. , Viewed on 17 March 2021, <https://www.leaderonomics.com/articles/personal/necessity-competition-leadership>.
7. MyFinB, n.d., AI Projects – MyFinB, viewed on 14 Sep 2021, <https://myfinb.com/uip/ai-projects/>.
8. Perkmann, M., Tartari, V., McKelvey, M., Autio, E., Broström, A., D'Este, P., Fini, R., Geuna, A., Grimaldi, R., Hughes, A., Krabel, S., Kitson, M., Llerena, P., Lissoni, F., Salter, A. and Sobrero, M. (2013). Academic engagement and commercialisation: A review of the literature on university–industry relations. Research Policy, [online] 42(2), pp.423–442, viewed on 17 March 2021, <https://www.sciencedirect.com/science/article/pii/S0048733312002235>.
9. Salleh, M.S. and Omar, M.Z., 2013, University-industry Collaboration Models in Malaysia. Procedia - Social and Behavioral Sciences, 102, pp.654–664.

10. Smith, A, n.d., 7 Fundamental Steps to Complete a Data Analytics Project, viewed on 17 March 2021, <https://blog.dataiku.com/2019/07/04/fundamental-steps-data-project-success>.
11. Statista, 2021, Malaysia: higher education enrollment | Statista. [online], Viewed on 17 March 2021, <https://www.statista.com/statistics/794845/students-in-public-higher-education-institutions-by-gender-malaysia/#:~:text=Students%20in%20public%20higher%20education,Malaysia%202012%2D2019%2C%20by%20gender&text=In%202019%2C%20around%20291.53%20thousand,enrolled%20in%20public%20higher%20institutions>.
12. UIPFuture, n.d., Home, viewed on 14 Sep 2021, < https://uipfuture.com/>.
13. www.wordstream.com, n.d., 6 Tips to Choose a Stunning Website Color Scheme, viewed on 29 Oct 2021, <https://www.wordstream.com/blog/ws/2019/10/03/website-color-scheme>.
14. IBM, 2019, What is software testing?, viewed on 11 Nov 2021, <https://www.ibm.com/topics/software-testing>.
15. Aebersold, K. (2019). *Software Testing Methodologies, viewed on 11 Nov 2021,* <https://smartbear.com/learn/automated-testing/software-testing-methodologies/>.
16. Coursera., n.d., What Is Python Used For? A Beginner's Guide, viewed on 11 Nov 2021, <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>.
17. www.tutorialspoint.com, n.d., *Google Colab - What is Google Colab? - Tutorialspoint*, viewed on 11 Nov 2021, <https://www.tutorialspoint.com/google_colab/what_is_google_colab.htm>.
18. www.pluralsight.com. (n.d.). *Python BeautifulSoup Web Scraping | Pluralsight*, viewed on 11 Nov 2021, <https://www.pluralsight.com/guides/extracting-data-html-beautifulsoup>.
19. Pfalzgraf, 2020, How to Use Selenium to Web-Scrape with Example, viewed on 11 Nov 2021, <https://towardsdatascience.com/how-to-use-selenium-to-web-scrape-with-example-80f9b23a843a>.