

Study Program Recommendation System



Chayanin Sutjaritvanichpongse

Pokpong Rujirachartkul

Somphon Ruengsri

Bachelor of Engineering in Software Engineering

Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

Academic Year 2020

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use



COPYRIGHT 2021

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE TECHNOLOGY LADKRABANG

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use

Thesis – Academic Year 2020

Bachelor of Engineering in Software Engineering

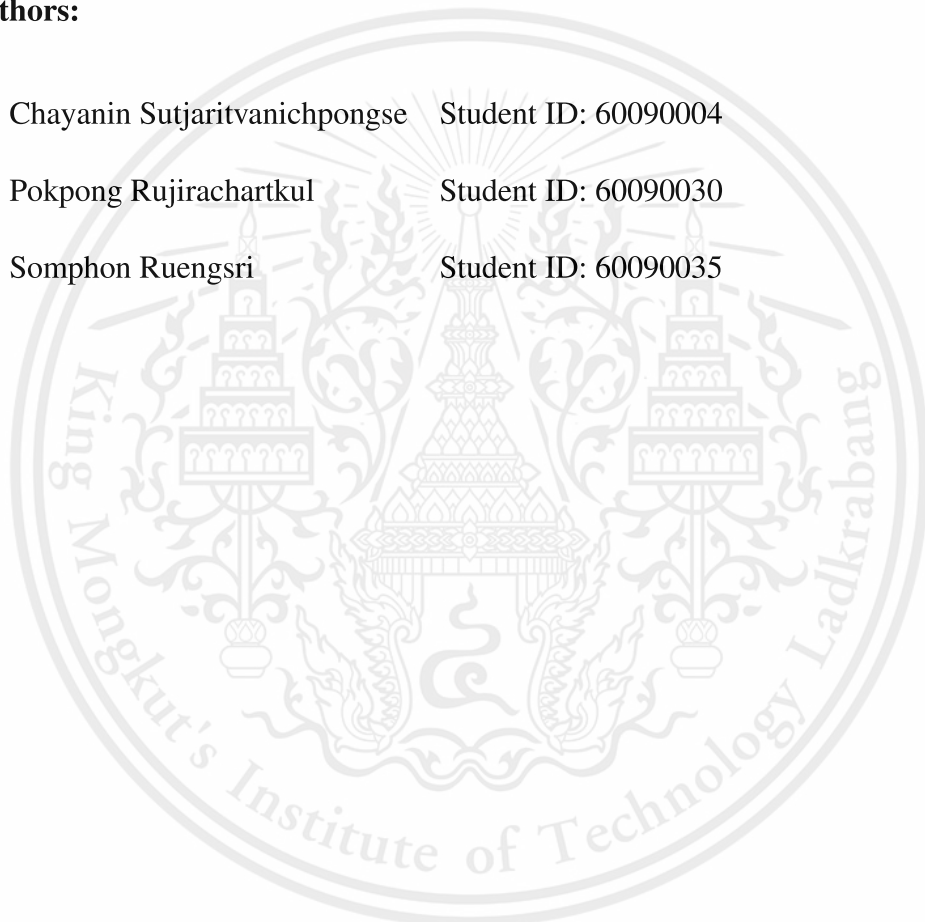
Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

Title: Study Program Recommendation System

Authors:

1. Chayanin Sutjaritvanichpongse Student ID: 60090004
2. Pokpong Rujirachartkul Student ID: 60090030
3. Somphon Ruengsri Student ID: 60090035



Approved for Submission

Approved for Submission

.....

.....

(Dr. Natthapong Jungteerapanich)
Advisor

(Asst.Prof.Dr. Chaiwat Nuthong)
Co-Advisor

Date ..30/6/2021

Date ..1/7/2021

Acknowledgment

We are sincerely thankful to to our advisors, Asst.Prof.Dr. Chaiwat Nuthong and Dr. Natthapong Jungteerapanich. We would like to express our very great gratitude. We really appreciated their valuable and constructive suggestions during the planning and development of this thesis. Their willingness to give their time so generously has been very much appreciated.

Also, we would like to extend our thanks to university staff and our colleagues including friends and classmates for all the support and help. Lastly, we would like to offer our special thanks to our parents and all other members of the family for their support and encouragement throughout our study and develop this thesis.

Abstract

The aim of the Study Program Recommendation System is to create a web application that recommends the student with lists of study programs that are suitable for them. The system recommends a high school graduate (or equivalence) with a list of the study programs that match their educational background, their Big Five personality label based on Big Five, and their preferences study programs. Our system requires each user to enter their educational information and take the Big Five personality test to be able to recommend the suitable study programs. After our system has gathered enough information of the educational and personality label data, we expect our system to be more accurate in providing study program recommendations. We expect that our recommendation system will reduce the dropout percentage of the students and also increase the mental health of the students, because they study in the programs that suit them.

Our Study Program Recommendation System is developed using Python, React JavaScript framework, and Django. On the client-side, the user interface is developed using React. On the server-side, the recommendation component along with the web services are developed using Python.

Table of contents

1	Objectives and Problem Description	14
1.1	Motivation	14
1.2	Objectives	14
1.3	Scope of Work	15
2	Related work	16
2.1	A Recommender for Improving the Student Academic Performance . . .	16
2.2	Recommender System for Predicting Student Performance	17
2.3	Developing an intelligent recommendation system for a private univer- sity in Thailand	18
3	Background Knowledge	20
3.1	Recommendation system	20
3.1.1	Collaborative filtering approach	20
3.1.2	Content based approach	20

3.2	Clustering Technique	21
3.3	Rule Based Machine Learning	21
3.3.1	Association rule learning	21
3.4	Supervised Learning Algorithm	22
3.4.1	Decision tree learning	23
3.4.2	Random forest	23
3.4.3	Regression Analysis	23
3.4.4	Linear Regression	23
3.5	Big Five personality traits	24
4	Software Analysis and Design	25
4.1	Requirements Analysis	25
4.1.1	Requirements	25
4.1.2	Use Case Diagram	27
4.1.3	Activity Diagrams	29
4.2	Software Design and Architecture	31
4.2.1	System Architecture	31
4.2.2	Database design	33
4.2.3	Software architecture	35
5	Implementation	38

5.1	Web Application	38
5.1.1	System Deployment	38
5.1.2	Client-Side Development	39
5.1.3	Server-Side Development	45
5.1.4	Big 5 Personality test scoring implementation	53
5.2	Recommendation System	54
5.2.1	Methodology	54
5.2.2	Classification method	54
5.2.3	Regression analysis	56
5.2.4	Algorithm selection	57
5.2.5	Dataset information	57
5.2.6	Classification	58
5.2.7	Regression	58
5.2.8	Train and test	58
5.2.9	Classification model	59
5.2.10	Regression model	59
5.2.11	Evaluate model performance	59
6	Results	60
6.1	Web Application	60
6.1.1	Home Page	60

6.1.2	Sign in/ Sign Up Page	63
6.1.3	Browse Program Page	67
6.1.4	Account Information Page	68
6.1.5	User Dashboard Page	70
6.1.6	Recommendation Programs Page	72
6.1.7	Educational Information Page	77
6.1.8	Personality Information Page	79
6.1.9	Admin Page	81
6.2	Experimental results	83
6.2.1	classification algorithm testing	83
6.2.2	Regression testing	84
7	Conclusion	86
7.1	What we have done	86
7.2	Problems that we faced	86
7.3	Possible improvements	87
	References	88

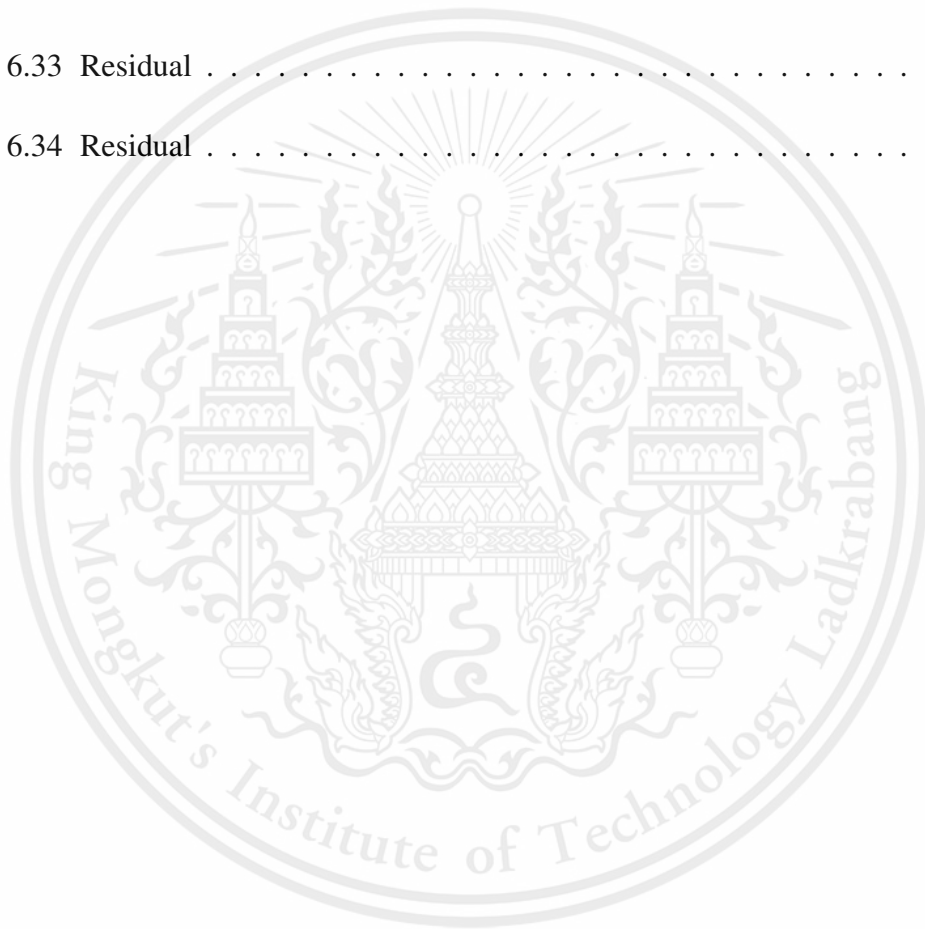
List of figures

4.1	Use Case Diagram	27
4.2	Activity Diagram	29
4.3	Admin Activity Diagram	30
4.4	Architecture Diagram version 4.0	31
4.5	Tool diagram	33
4.6	Table design and it's relations	34
4.7	Django project structure	35
4.8	Software architecture diagram	37
5.1	System deployment on Heroku	39
5.2	React-DOM render a props	40
5.3	JavaScript in Sign-In Component	40
5.4	HTML code in 'render()' function	41
5.5	Axios request 'POST' method to the server and handle returned JSON formatted data.	41

5.6	List of dependencies used in Client-side.	42
5.7	Upload file to Clouinary function.	43
5.8	delete file from Clouinary function.	44
5.9	Django restful-framework extension in INSTALLED-APPS	45
5.10	POST and GET method in 'login' and 'get-all-user' functions	46
5.11	APIs end-point in 'urls.py'	47
5.12	Client personal info class in 'models.py'	49
5.13	Client personal info table created using Django	50
5.14	Client db & Server db settings (Blurred out for security reason)	51
5.15	Email setting in 'setting.py' (Blurred out for security reason)	52
5.16	Usage of email service	52
5.17	Personality test scoring algorithm	53
6.1	Home Page	60
6.2	Home Page	61
6.3	User Dropdown	62
6.4	Sign Up page	63
6.5	Sign In page	64
6.6	Activate Account Email	64
6.7	Welcome	64
6.8	Forgot Password page	65

6.9	Email for reset password	65
6.10	Enter new password	66
6.11	Reset password completed	66
6.12	Browse Program Page	67
6.13	Account Information Page	68
6.14	Information update email notification	69
6.15	User Dashboard Page	70
6.16	Incomplete Recommendation Page	72
6.17	Complete label	72
6.18	Preference program	73
6.19	Preference program	74
6.20	Preference program	75
6.21	Preference program	75
6.22	Preference program	76
6.23	Education Information (All)	77
6.24	Education Information (Range)	78
6.25	Education Information (Step)	78
6.26	Personality Test Page	79
6.27	Personality Test Result	79
6.28	Study Program Recommendation Result Page	80

6.29 Admin Program List	81
6.30 Pop up input for new program	81
6.31 Pop up input for new program	81
6.32 Feedback from user	82
6.33 Residual	84
6.34 Residual	85



List of tables

2.1	Number of clusters and iterations used by the K-means and TwoStep Clustering Techniques	18
2.2	Number of cluster and iteration used by the K-means and Two Step Clustering Techniques	19
5.1	List of available APIs	48
6.1	Cross validation	83
6.2	Matrix confusion for Random Forest	83
6.3	Matrix Logistic regression	83
6.4	LinearDiscriminantAnalysis	84

Chapter 1

Objectives and Problem Description

1.1 Motivation

One of the most common problem that students entering universities have to face is choosing their study program at university. Many students do not know which study programs suit their personality and education background most. Our study Program Recommendation system will recommends the student with lists of study programs that match the student profile most. So the students will be able to know which study program that they are likely to succeed, this should decrease the rate of dropouts and increase the students' mental health as they are studying the programs that suit them.

1.2 Objectives

To develop a. web-based study program recommendation system that can recommend study programs for high school graduates (or equivalence) based on their educational background and personality traits. The system is expected to reduce dropout percentage and increase students satisfaction on their study programs. The data gathered by the system is expected to be useful for program planning and management.

1.3 Scope of Work

- The system to be developed is to be a web application that works on popular web browsers.
- The study programs to be recommended are undergraduate programs at KMITL. Depending on the data available, not all such programs may be used in the recommendation.
- Effectiveness of the recommendation algorithms will be evaluated and reported.



Chapter 2

Related work

This chapter provides a survey of related work.

2.1 A Recommender for Improving the Student Academic Performance

Kuyoro, et al [2] signifies that by recognizing the background factors can help to develop the optimal algorithm to improves student academic performance. He used multilayer perceptrons along with ten classification models to develop the optimal algorithm that helps in student performance prediction, recognize background factors that contribute to the performance and recommend essential measures to help the student at the end of their first year.

The data are taken from student record files from Babcock University, Nigeria. Within the period of 2001 and 2010, a total of eleven thousand and twenty-seven students (11,027). Seven thousand, five hundred (7,500) complete records of the enrolled students representing 68 percent of the entire population were randomly selected for the purpose of this study.

The result from the paper confirms that family background plays an important role in student's performance in their first year of university. Many of the respondents agree that negative home environment can lead to less concentration and stability that

results in poor performance in academic fields. The family background and environment are incredibly influenced by both culture and region factor. For instance, some region in the country of Nigeria places more value to education than the others. It is also observed that students who are from the region where education are compulsory tends to be more motivated than students from the other regions. The paper also points out that the respondents choose gender, average family income, ethnicity, religion, family size, parent's educational qualification, parents' marital status, parent's occupation, education sponsor, age at which the student entered the university, and secondary school certificate examination grade as the background factors are identifies in this paper.

2.2 Recommender System for Predicting Student Performance

Nguyen, et al [3] proposes a novel approach which uses recommender system techniques for educational data mining, especially for predicting student performance. They compare their recommender system techniques with traditional regression such as linear regression and logistic regression.

They have used two data sets from the Knowledge Discovery and Data Mining (KDD) Challenge in 2011. Each data set is split into a train and a test partition. The data consist of the log files of the interactions between students and computer-aided-tutoring systems. While students solve math related problems in the tutoring system, their activities, success and progress indicators are logged as individual rows in the data sets.

The best results on the "Algebra" dataset were achieved using only KC-rule. The reason for this is that, in this setting, there are fewer items (2,979) and the same number of ratings, thus, on average, each user has more ratings ($8,918,054/2,979 \approx 2,993$ ratings/user), which means that the algorithms have more data to learn from. Clearly, the proposed method, which employs recommender system techniques to predict the student performance, outperforms the traditional regression methods.

2.3 Developing an intelligent recommendation system for a private university in Thailand

Kanokwan [4] points out that many of the universities in Thailand are taking initiatives, like Student Recommendation System (SRM), to improve the counselling service for choosing a study program. The paper investigates the relationship within the data of student historical data and their final results by comparing between k-means and two step methods, with rough set and fuzzy logic. The paper describes the development of the experiments, and the proposed Intelligent Recommendation System framework.

The dataset used here consists of 3,500 student records from five academic years. The data includes records of student from the first year to graduation. The data is randomized and all private information has been removed to make sure no student is identified.

In the experiment, the results have shown that Two Step clustering technique has a maximum of 15 clusters and a minimum of 2 clusters. Table 2.1 shows that seven clusters have been used in order to compare with the results from the K-means method

	K-means clustering	Two Step Clustering
Number of cluster	7	7
Number of iteration	13	-

Table 2.1: Number of clusters and iterations used by the K-means and TwoStep Clustering Techniques

Table 2.2 shows the percentage of records which have been allocated to each cluster. The differences between the two methods are also shown in the same table.

	K-means clustering (%)	Two Step Clustering(%)
cluster 1	23.00	6.457
cluster 2	7.486	4.914
cluster 3	5.514	10.286
cluster 4	5.314	6.914
cluster 5	8.600	8.114
cluster 6	11.229	8.571
cluster 7	2.143	9.5457

Table 2.2: Number of cluster and iteration used by the K-means and Two Step Clustering Techniques

Chapter 3

Background Knowledge

3.1 Recommendation system

Recommendation system is a system used to predict the preference or interest of the user, by filtering through a large set of data and selecting the significant ones. In general, there are two main approaches used by recommendation systems: Collaborative filtering approach and content-based approach.

3.1.1 Collaborative filtering approach

The Collaborative filtering approach for recommender systems is a method that are based on the the interactions between the past records of users and items in order to produce recommendations. It filters out items that a user might like on the basis of the interaction by similar users. It works by searching in a large data set finding a smaller set of users which are similar to a particular user. It looks at the items they like and combines them to create a ranked list of suggestions and make predictions based on the estimated proximity.

3.1.2 Content based approach

Unlike collaborative filtering that only depends on the user-item interactions, the content based approach uses additional information about users and/or items. The

idea is to tag products using certain keywords and understand what the user likes, look up those keywords in the database and recommend different products with the same attributes that match the user's profile.

3.2 Clustering Technique

Clustering techniques are based on matching the profile of a user against a similar profiles of different users. It is the grouping of data so that the objects within the same cluster are as similar as possible while data at different clusters are as dissimilar as possible. The similarity measures on factors such as distance, connectivity and intensity.

3.3 Rule Based Machine Learning

Rule based uses machine learning methods to identify the rule. It has the same concept as a rule based system but has a different approach as rule based systems often need human knowledge to manually construct rules and curate a rule set whereas Rule based machine learning uses some form of learning to identify useful rules automatically. Examples of Rule-based machine learning approaches include learning classifier systems association rule learning, artificial immune systems, and any other method that relies on a set of rules, each covering contextual knowledge.

3.3.1 Association rule learning

Association rule learning is the rule based machine learning method to discover interesting relations in a large set of data items. This rule show how an item set occur in the transaction. It involves the use of machine learning to analyze data for patterns, co-occurrences in a database and also identifies frequent if-then associations which are association rules.

For example, let us consider the transaction table as below:

Transaction ID	Milk	Bread	Butter	Beer
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

First we find the Co-Occurrence of the table to count the frequency of each item appear, which is given as table below:-

	Milk	Bread	Butter	Beer
Milk	2	2	1	0
Bread	2	4	2	0
Butter	1	1	2	0
Beer	0	0	0	1

Thus from here we can find the association between each pair of items using 'IF condition Then result' For example

- 1) If Milk Then Bread
- 2) If Milk Then Butter

3.4 Supervised Learning Algorithm

A Supervised learning algorithm is a learning algorithm that is suited for the data that has the predicted outcome already. It maps the input to the output to create model. For supervised learning algorithm the data is usually divided into a train set and test set in the ratio out of 100 where the train data will be use for the computer to learn the data and the test data will check if the computer learn it correctly given it by percentage.

3.4.1 Decision tree learning

Decision tree learning is one of the supervised learning algorithms. It uses a decision tree to make the decision and predict the result based on what the decision tree is learned. The decision tree is learned by going through different attributes in the data in order to understand the association between the attribute and the output. It helps to split the data set based on different conditions.

3.4.2 Random forest

Random forest is an ensemble learning algorithm. It can be used for both regression and classification. It is done by constructing many decision trees at the training time. Each tree is responsible for predicting the outcome. The major outcome from all the decision trees is selected as the output of the prediction.

3.4.3 Regression Analysis

Regression analysis is a method to estimate the relationships between a dependent variable and independent variables. The dependent variable is the variable that depends on other factors and independent variables are the variables that stand alone and do not depend on other factors.

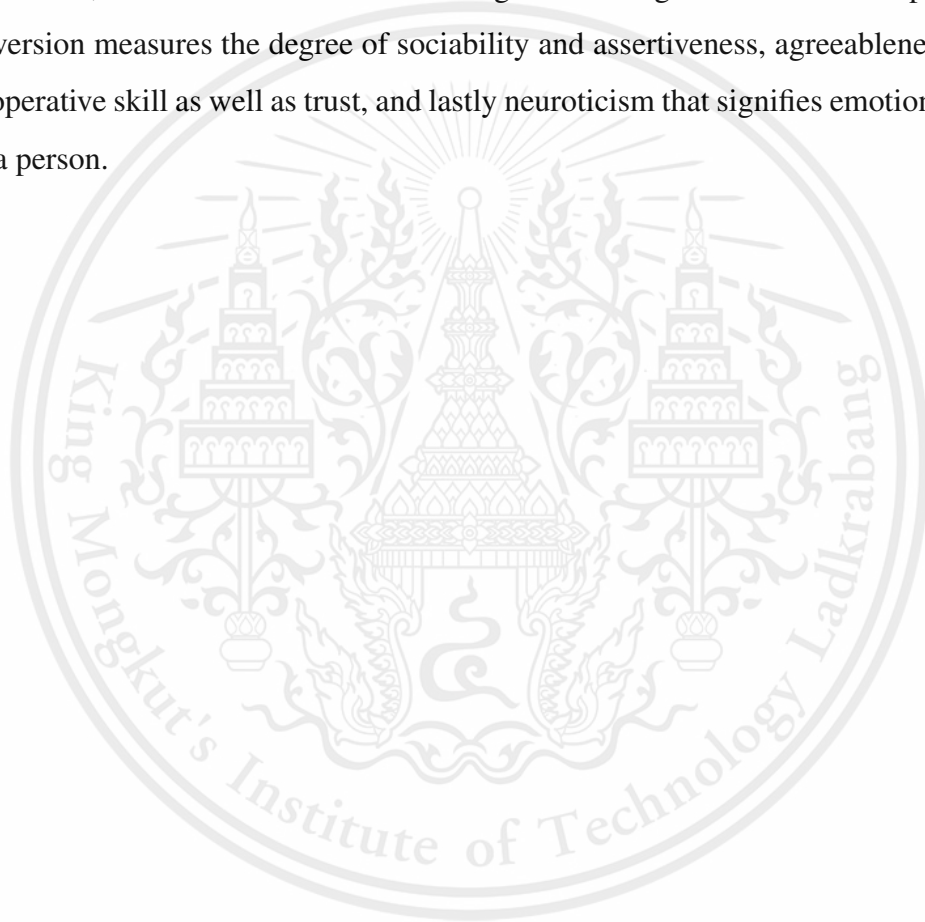
3.4.4 Linear Regression

Linear regression is a method to estimate the relationship between two variables by fitting the linear equation to the data in which one variable is an explanatory variable and the other is a dependent variable. For example, a linear equation for estimating the weight by using the height.

The most common method for Linear regression is least square regression. This method calculates the best fitting line by optimizing the mean square error of the vertical deviations from each data point to the line.

3.5 Big Five personality traits

According to the contemporary psychologists, personality can be divided into 5 major broad traits. The five personality dimensions are the key traits that help to build up a person's personality as a whole. However, there are some important note to be taken. Each of the five traits only shows the extreme of the polar ends of the personality. To be realistic, there are many people who falls between the range of the two extremes. The Big Five personality includes openness that shows the level of creativity and range of interest, conscientiousness tells the degree of thoughtfulness and competence, extraversion measures the degree of sociability and assertiveness, agreeableness signifies cooperative skill as well as trust, and lastly neuroticism that signifies emotional stability of a person.



Chapter 4

Software Analysis and Design

4.1 Requirements Analysis

4.1.1 Requirements

In order to achieve our objective to build the study program recommendation system, we need to define the requirements of the system. By dividing the requirement in to two separate categories, Functional and Non-Functional requirement.

Functional Requirements

Functional requirements that our system must offer. They can range from high-level abstract statements to low-level statements that typical readers can understand. Below is the list of our system functional requirements.

- System provides the user the ability to login and register.
- System provides the user the ability to take the test.
- System provides the user the ability to take the personality test if they do not know the personality type.
- System provides the user to view their result history.
- System provides users with the ability to view their study program and career information.

- System provides user the ability to view the personality type.
- System provides user the ability to view feedback.

Non-Functional Requirements

Non-Functional requirements define system attributes such as security, reliability, performance, maintainability, scalability and usability. To ensure the usability and effectiveness of the system, non-Functional requirements are as important as functional requirements. By failing to achieve any of these requirements, it can cause entire system to act abnormally. Listed below is our system non-functional requirements.

- The response time when submitting the request is less than 30ms.
- The website should be easily understood and user-friendly.
- The connection between client and server should be seamless with few or non re-interfacing.
- The visualization system should be reliable and prevent system crashes as much as possible.
- The recommendation system can recover data from unintended situations.
- The recommendation system should work on any web browser regardless of device and platform.
- The system should be able to run 24/7.
- The system should be able to keep the record of at least 1 million records.
- The system will keep the information of the user for further improvement of the algorithm.
- The system requires the internet to give the results.
- The system should support multiple users at the same time.
- The user interface should be implemented using React.JS.

- The system should be able to accurately recommend study programs to users.

4.1.2 Use Case Diagram

The following diagram is the user case diagram. It shows the capabilities of the users that can interact with the system based on the requirements.

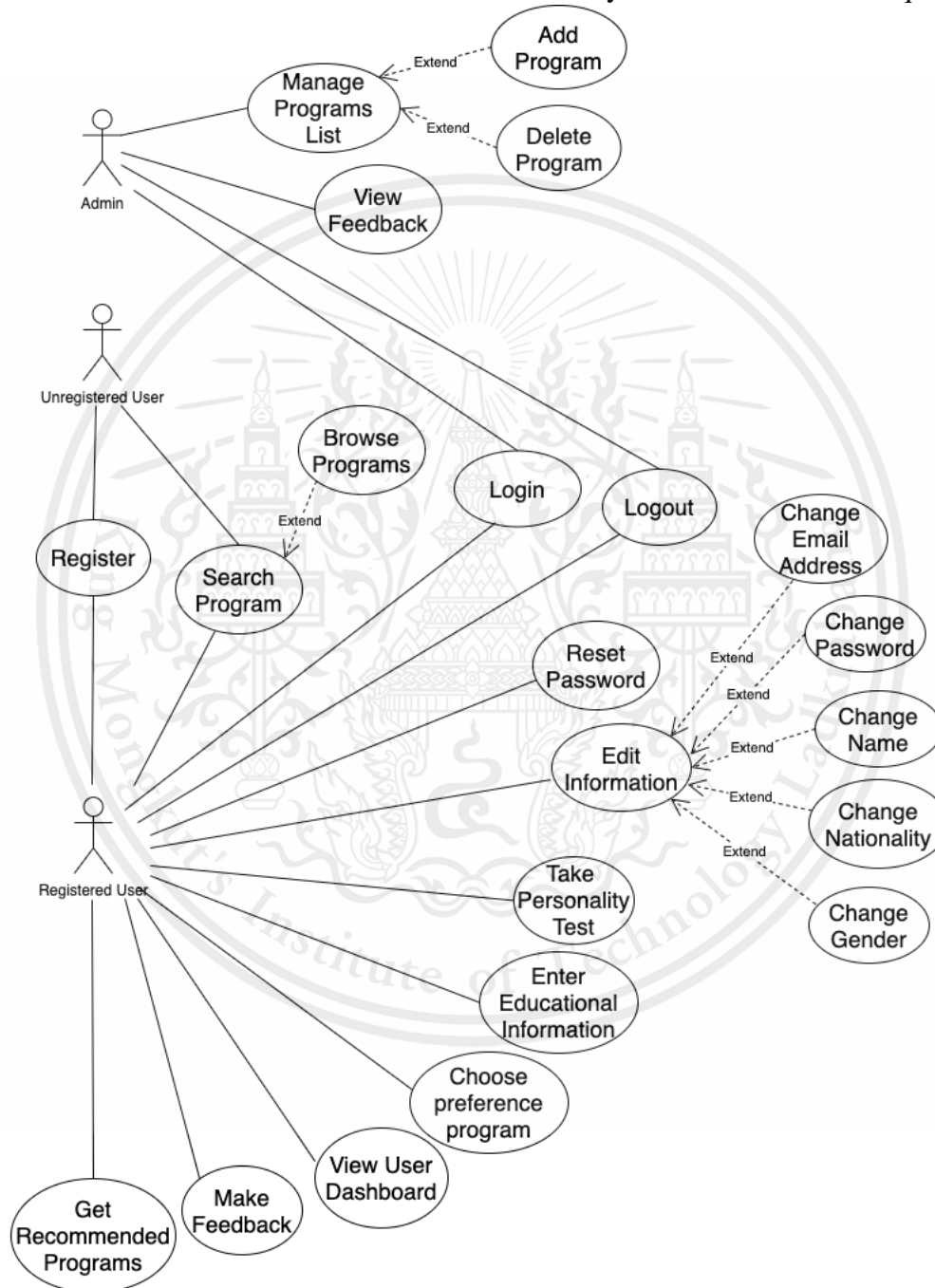
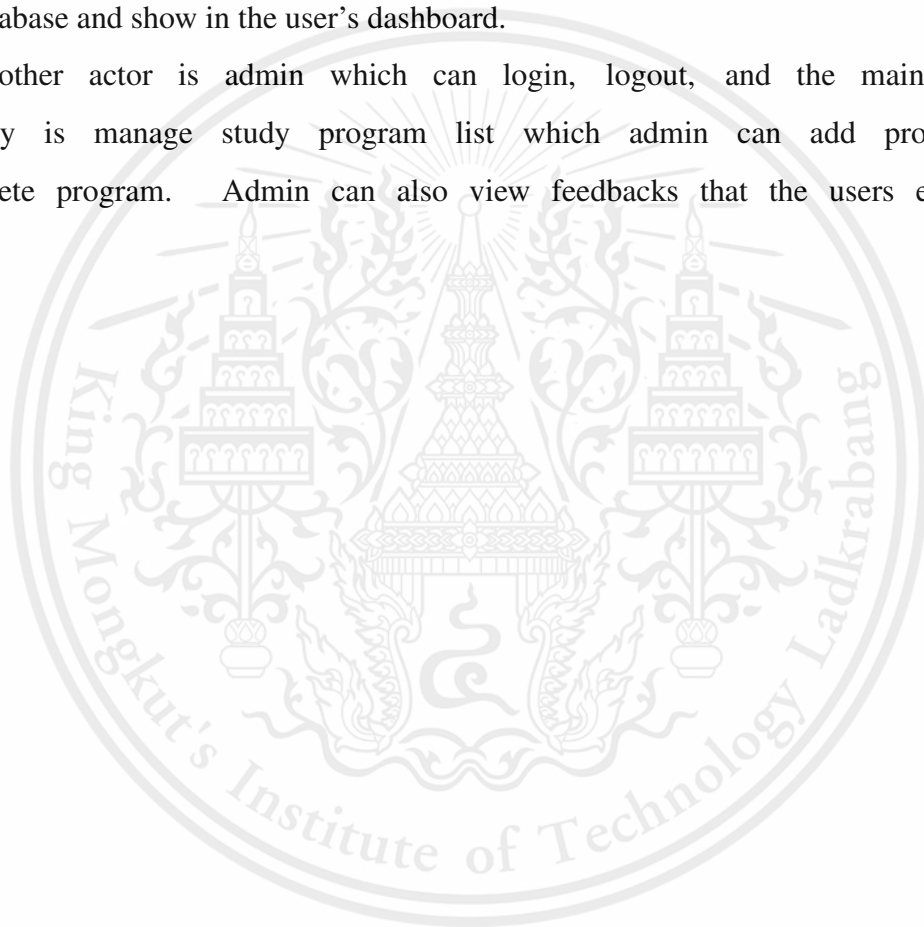


Figure 4.1: Use Case Diagram

The main actor of this diagram and system is user. User can register, login, logout, enter their educational information, take personality test to view their personality type, make feedback, Browse program using search box, get their most recent recommended program and previous result and edit their personal information such as email and password. User will have to login to the system in order to use some function on the system, User with no account can create new account through the web page. To get recommended programs, The system will first analyse the input data. Then, The system will start analyse the data. Lastly, The recommended program result will be store in the database and show in the user's dashboard.

Another actor is admin which can login, logout, and the main functionality is manage study program list which admin can add program and delete program. Admin can also view feedbacks that the users entered.



4.1.3 Activity Diagrams

The following diagram is the Activity diagram of our Study Program Recommendation System, which the diagram represent the flow of our system .

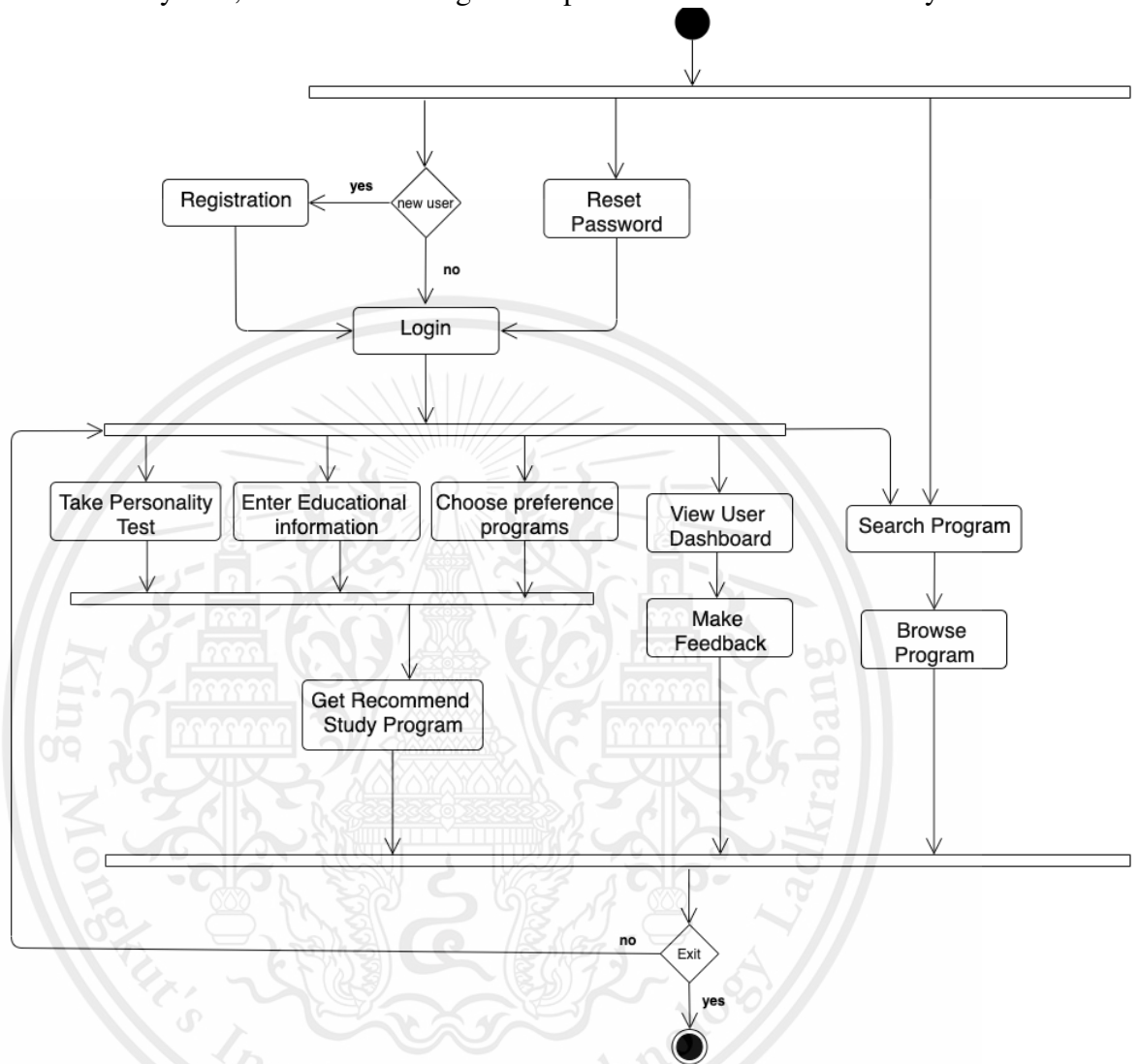


Figure 4.2: Activity Diagram

The following diagram is the Admin Activity diagram of our Study Program Recommendation System, which the diagram represent the flow of the admin's side .

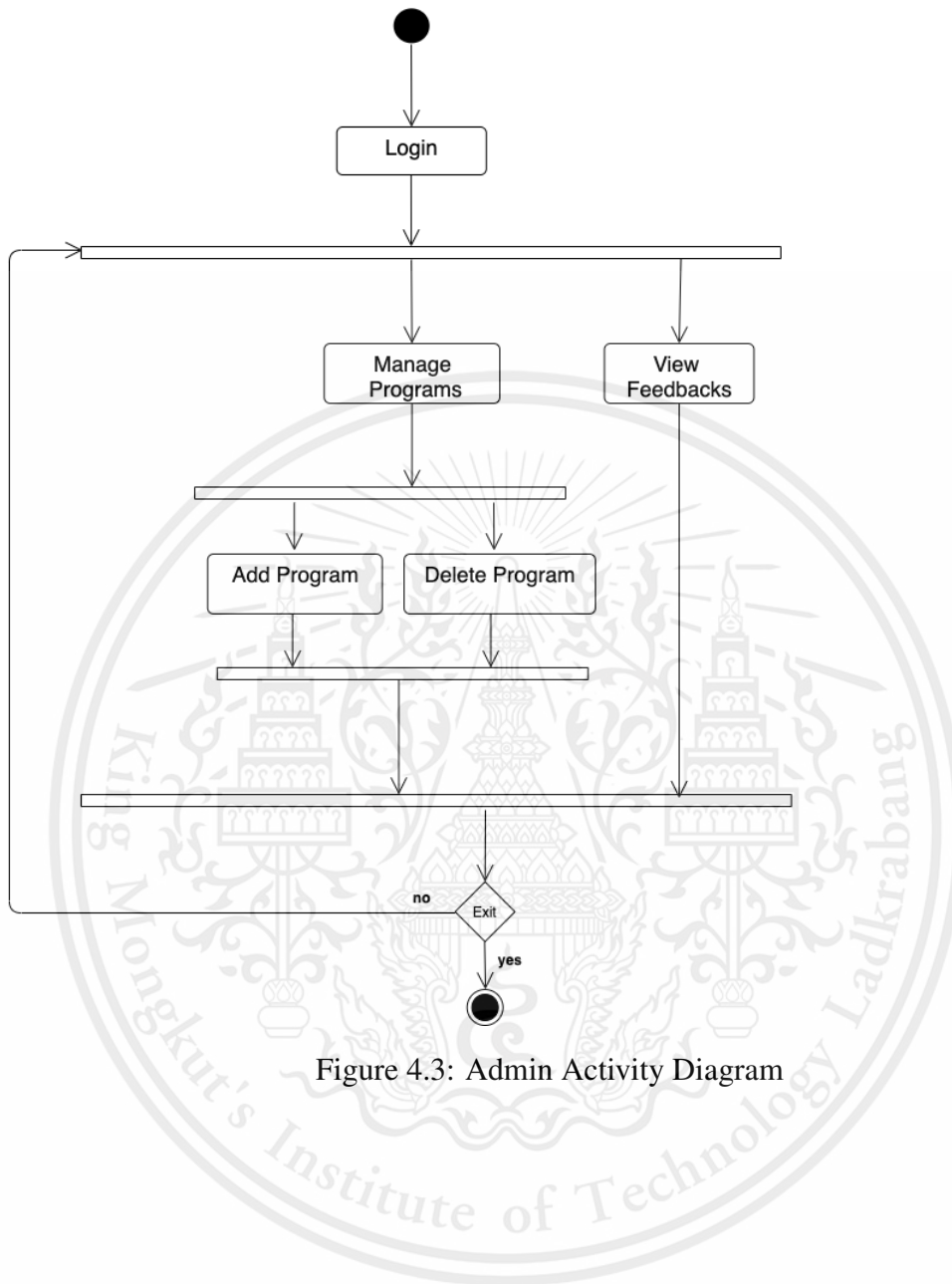


Figure 4.3: Admin Activity Diagram

4.2 Software Design and Architecture

4.2.1 System Architecture

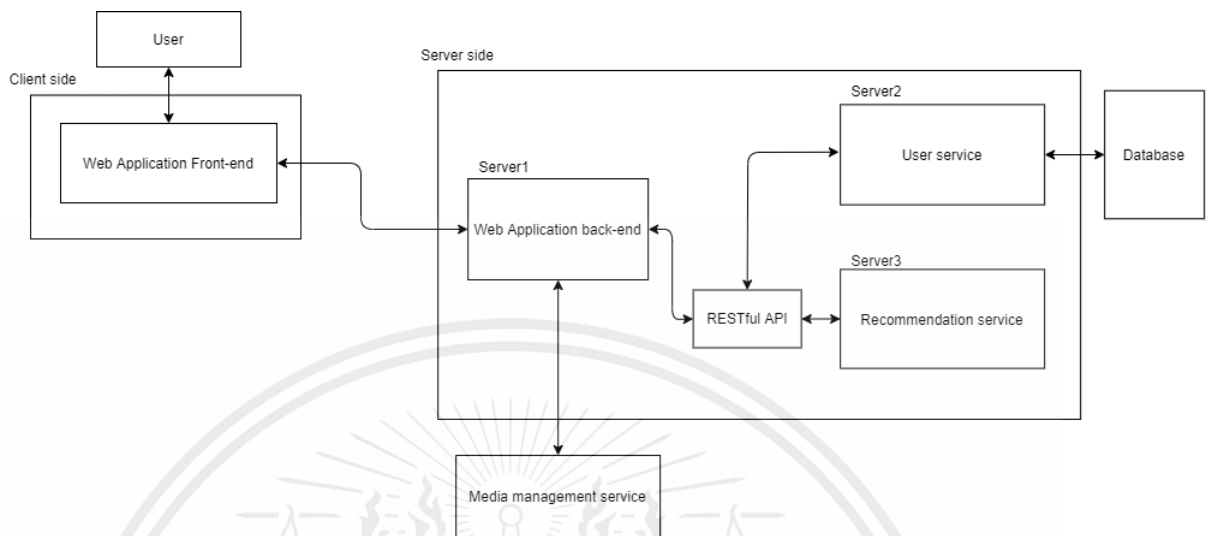


Figure 4.4: Architecture Diagram version 4.0

There are 6 parts combined to form a working system, User, Client, HTTP-Requests service, Server, Database and Media management services. Client side and Server side being the main part of the system. First, Client side. Web application front-end will consist of a front-end where users interact with the system developed using React as the web framework of the system which help support the development of our system and web application back-end which our front-end were build on, it is node.js, it will handle all the network send and receive and also handle the data from the user. Second, Server side. Server side in the version 4.0 there are a 2 services in each server, Main service and API framework service. For the first server, Recommendation service server, developed using Python language and in this service is where the input from the user being processed to return the recommendation result to the user. Second server, user service server is the service designed to complete common tasks that users can interact with the system in the user's dashboard for example: Change personal detail, View result history, login, logout, sign-up, etc. API framework, it is the part where it accepts the request from the client through HTTP-Requests protocol and handle the request according to the service client requested. Third, User part. User part is where users access the system, they can either access the system by desktop browser

or through mobile browser. Forth, HTTP-Request service. HTTP-Request service is the part that receive and sent the request from the user to the service that it's suppose to get the requests. Fifth, Database. For our system we choose PostgreSQL. The reason that we choose PostgreSQL is because of it's simplicity and efficiency of querying, storing data and the ease of integration between Django. The database part will communicate with the service in the server part using the queuing technique to avoid collision of process and query that can cause data lost or even system crash. Lastly, media management services. This service will be handling the media that's being uploaded to the system, such as profile picture or course picture. It will store the media on the cloud storage and return the URLs to the media back to the server for easy access.

Client-Side Components

- Front-end developed using React and JavaScript.
- Web Framework: React
- HTTP-Requests Services: Axios.

Server-Side Components

- Recommendation service to recommend the study program to user.
- User service to complete user's common task.
- Web application back-end: Node.Js
- API Framework: Django

External Components

- Database: PostgreSQL.
- Media management service: Cloudinary.

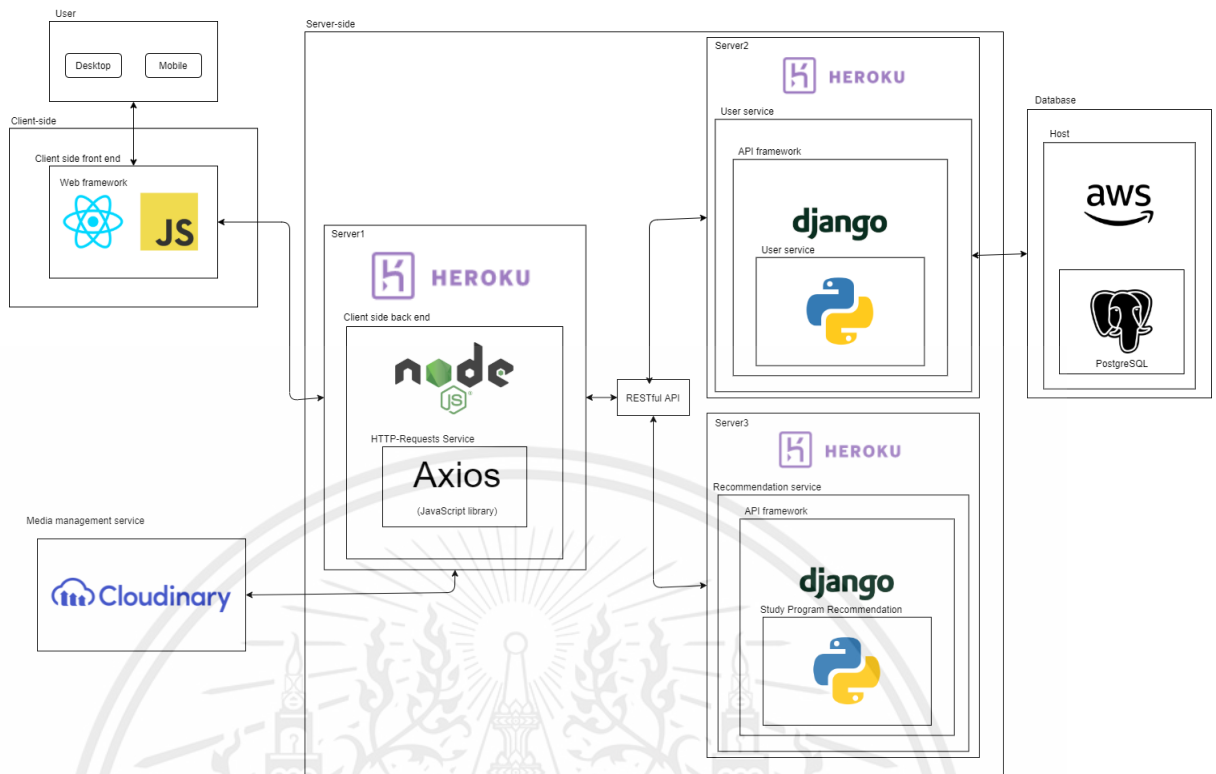


Figure 4.5: Tool diagram

4.2.2 Database design

In our system, we decided to have only 1 database for the system. The database that stored user's information along with their recommendation history and personality results. We also include the table where admin can store available course at KMITL, by adding it to the system via admin page. Tables consist of:

- Client personal information
- Client education information
- Client personality score
- Client recommendation history
- Feedback
- Course

Table design

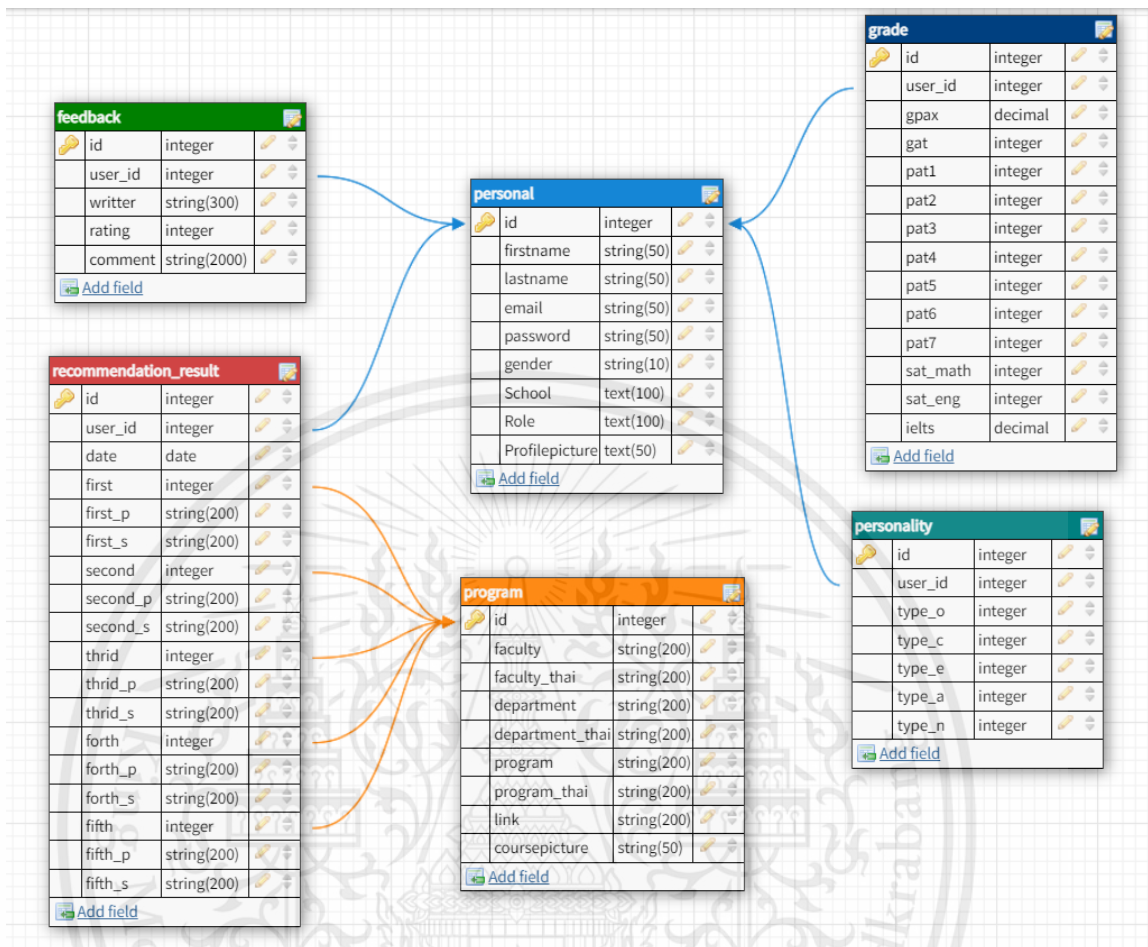


Figure 4.6: Table design and it's relations

- (a) Store global setting and other properties that used across all application.
- (b) **__init__.py** : It is an empty file that tells Python the directory that it have been stored should be considered a Python package.
- (c) **setting.py** : Store global setting and configuration for Django project
- (d) **urls.py** : Store the URL declarations for Django project
- (e) **wsgi.py** : It is an entry point for WSGI-compatible web servers to server the project

3. Apps folder

- (a) Store all application in the project, Can store multiple applications.
- (b) App_name folder
 - i. migration folder : Store migration files for every migration that's apply to the database.
 - ii. **admin.py** : It's where you register your models
 - iii. **apps.py** : Application configuration file
 - iv. **__init__.py** : It is an empty file that tells Python the directory that it have been stored should be considered a Python package.
 - v. **models.py** : It store a information about your data. Contains essential filed and type of data that you are storing. Each model maps to a single database table.
 - vi. **test.py** : File to store a test for an application.
 - vii. **views.py** : This file store python functions that take a web request and return a response. Response can be of any format or datatype.
 - viii. **serializers.py** : It store the set of code that allow data like queryset or model instances to be converted to python datatype like: JSON, XML, etc. It also provide deserialization allowing parsed data to be convert back to queryset or model instances.

4. Static, Template, Media, Virtual_env folders

- (a) **Static folder** : Store a static file such as icon, background picture.
- (b) **Template folder** : Store a HTML files that is the template of the application.
- (c) **Media folder** : Store media file likes video, GIF file.
- (d) **Virtual_env folder** : Store a virtual environment file for development process and can be used to test the application.

Our system is design to use Django only as an API framework, Because we already use React as our web-framework. So the part where Django provide support for front-end design and development are not been used. Static folder, Templates folder, media folder are left empty. For virtual environment, We used PIPENV, It is a python extension package. PIPENV automatically creates and manages a virtual environment for our project and automatically adds and removes packages from Pipfile when install or uninstall any packages. It also automatically generates the Pipfile.lock, which is used to produce deterministic builds when we deploy our system. Virtual environment provide isolated environment to isolate the project. This will endure that any change made to the on-going project won't affect the real production project.

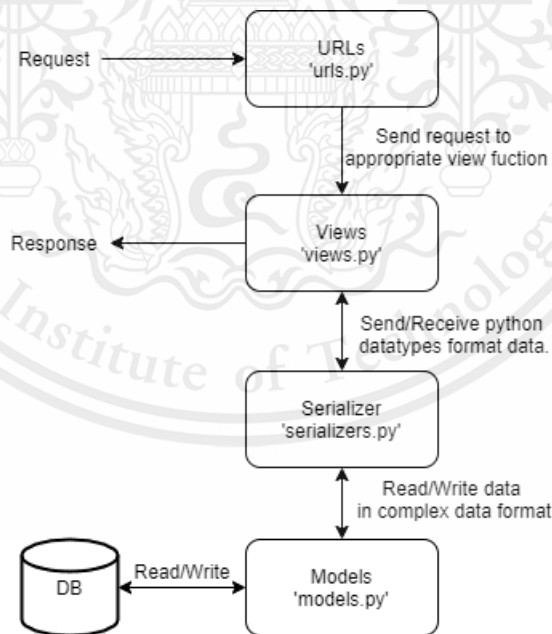


Figure 4.8: Software architecture diagram

Above showing the software architecture diagram for both client service and recommendation service and it's flow from start to finish.

Chapter 5

Implementation

5.1 Web Application

This chapter describes the development process of our project. Additionally, this chapter also describes the development tools required to implement the system components. The development tools are divided into 2 parts, client-side and server-side.

For the client side we used React and a JavaScript to build the user interface. Which is the web-site for user to interact with. For server side, we used python as our main programming language. With the help from Django , which is Python based web framework. Django has an ability to automatically create, delete, modify database properties from the code it self, that's eliminate the need of developer to manually edit the properties of the database.

5.1.1 System Deployment

Both client side and server side are deployed and running on Heroku. Also the databases that we use is an extension of Heroku where the databases are hosted by Amazon web service(AWS). We create a repositories on Github to store our code and with the ability of Heroku we can link those repository directly to Heroku, so every time we push new code to the repository it's triggered the CI/CD process where it automatically deploy and compile new code to Heroku.

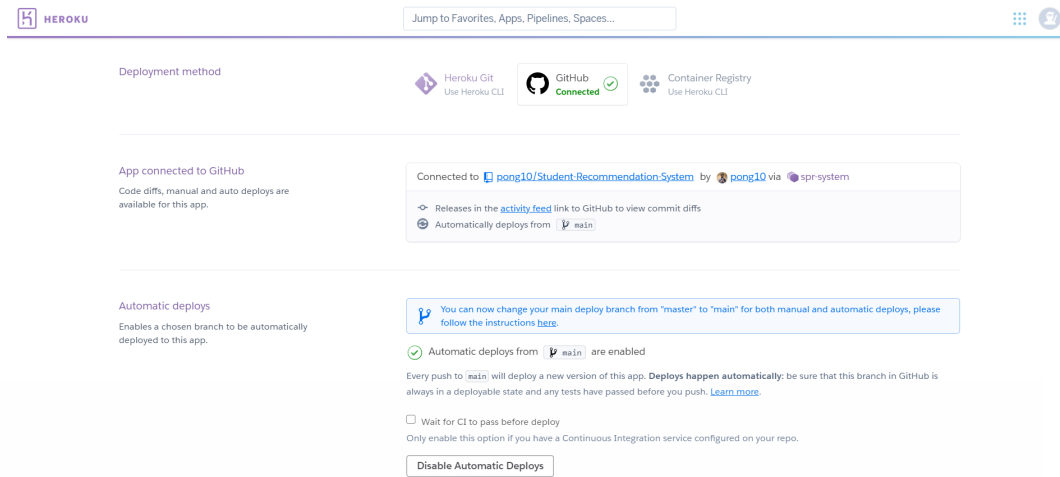


Figure 5.1: System deployment on Heroku

5.1.2 Client-Side Development

Our goal of the system is to develop a web-based system, So there is no specific requirements for any device to use the service as long as it has an access to the internet. We develop our web site to be responsive, which means both users from desktop and mobile will be able to use our services. There are 2 main important component that we develop in the client side.

- React web framework.
- Axios HTTP-Requests service

React web framework

React is an open-source, front-end, JavaScript library for building user interfaces. As React code is made of entities called components. Component will be rendered to a particular element in the DOM(Document Object Model) using React-DOM library. When rendering a component of the web page we can pass in values which known as "Props". "Props" is a special keyword in React, which stands for properties and is being used for passing data from one component to another.

```

ReactDOM.render(
  <App /> ,document.getElementById('root')
);

```

Figure 5.2: React-DOM render a props

In order to get a props. We need to export props component from '.JS' file that contain the UI elements. The file that contain the UI elements are also contain the function that written in JavaScript.

```

class SignIn extends React.Component {
  constructor() {
    super();
    this.state = {
      input: {},
      errors: {},
      response: {},
    };
    this.handleChange = this.handleChange.bind(this);
    this.handleSubmit = this.handleSubmit.bind(this);
  }
  handleChange(event) {
    let input = this.state.input;
    input[event.target.name] = event.target.value;

    this.setState({
      input
    });
  }
}

```

Figure 5.3: JavaScript in Sign-In Component

The actual UI component is written in HTML format enclosed by 'render()' function which will be called by React-DOM to render the page.

```

render() {
  return (
    <div>
      <img src={bg} alt="Logo" width="1440" className = "signup-bg" />
      <div className="signin-container">
        <h1 className="signin-h1">Sign In</h1>

        <form onSubmit={this.handleSubmit}>

          <div class="form-group ">
            <label for="email" className="signin-email-label">Email Address:</label>
            <input
              type="text"
              name="email"
              value={this.state.input.email}
              onChange={this.handleChange}
              class="form-control signin-email-input"
              placeholder="Enter email "
              id="email" />

```

Figure 5.4: HTML code in 'render()' function

Axios HTTP-Requests service

Axios is a JavaScript library used to make HTTP requests from node.js or XMLHttpRequests from the browser. Axios is similar to Fetch API, but Axios is more simple as Axios perform automatic transforms of JSON data, but Fetch is a 2-step process when handling JSON data, first to make the request, then call '.json()'

to transform the data. In our project we use Axios to handle the request from client to server. It can make HTTP requests using every method available like: GET, POST, PUT, PATCH, DELETE, etc.. to the server.

```

if (isValid) {
  let payload = {
    'email' : input["email"],
    'password' : input["password"]
  }
  this.response = await axios.post("http://127.0.0.1:8000/login/", payload)
}
console.log(this.response.data)
if (this.response.data['status'] === true) {
  console.log("Passed")
  //this.history.push("/");
  this.props.history.push("/")
  //return <Redirect to="/" />;
}

```

Figure 5.5: Axios request 'POST' method to the server and handle returned JSON formatted data.

List of React dependencies

Dependencies are an extension to the React, which give the developers more tools from third party developers to make the code more flexible as different dependencies give additional functions for developer to work with. Below is the list of dependencies in our system.

```
5   "dependencies": {
6     "@lottiefiles/lottie-player": "^1.0.3",
7     "@material-ui/core": "^4.11.0",
8     "@testing-library/jest-dom": "^4.2.4",
9     "@testing-library/react": "^9.5.0",
10    "@testing-library/user-event": "^7.2.1",
11    "axios": "^0.21.1",
12    "bootstrap": "^4.5.2",
13    "chart.js": "^2.9.4",
14    "cloudinary": "^1.25.1",
15    "express": "^4.17.1",
16    "framer-motion": "^4.1.6",
17    "material-design-icons": "^3.0.1",
18    "mdbreact": "^5.0.1",
19    "rc-scrollbars": "^1.1.2",
20    "react": "^16.14.0",
21    "react-bootstrap": "^1.4.0",
22    "react-chartjs-2": "^2.11.1",
23    "react-dom": "^16.13.1",
24    "react-fade-in": "^2.0.1",
25    "react-google-material-icons": "^1.0.4",
26    "react-icons": "^4.2.0",
27    "react-lottie": "^1.2.3",
28    "react-redux": "^7.2.2",
29    "react-router-dom": "^5.2.0",
30    "react-scripts": "3.4.3",
31    "react-select": "^4.1.0",
32    "redux": "^4.0.5"
33  },
```

Figure 5.6: List of dependencies used in Client-side.

Cloudinary media management service

Cloudinary is an online media management service that allow us to upload and store our user profile picture, program cover picture and other images present on our website. When the user upload their own image to the website, we will safely upload and store their image on our cloud storage and if they deleted from our website the picture will also be deleted from our cloud storage also. When uploaded, the service will return the URLs to the image uploaded, then when user press confirm button then we will store the name of the image to the user's object on our database for convenience.

```
function uploadFile(file) {
  const url = `https://api.cloudinary.com/v1_1/[redacted]/image/upload`;
  const xhr = new XMLHttpRequest();
  const fd = new FormData();
  xhr.open("POST", url, true);
  xhr.setRequestHeader("X-Requested-With", "XMLHttpRequest");

  // Update progress (can be used to show progress indicator)
  xhr.upload.addEventListener("progress", (e) => {
    setProgress(Math.round((e.loaded * 100.0) / e.total));
    // console.log(Math.round((e.loaded * 100.0) / e.total));
  });

  Complexity is 3 Everything is cool!
  xhr.onreadystatechange = (e) => {
    if (xhr.readyState === 4 && xhr.status === 200) {
      const response = JSON.parse(xhr.responseText);

      let a = response.secure_url.split('/');
      setprofilepic(a[7]);
      setImage(response.secure_url);
      setPublicid(response.public_id);
      // console.log(response);
    }
  };
};
```

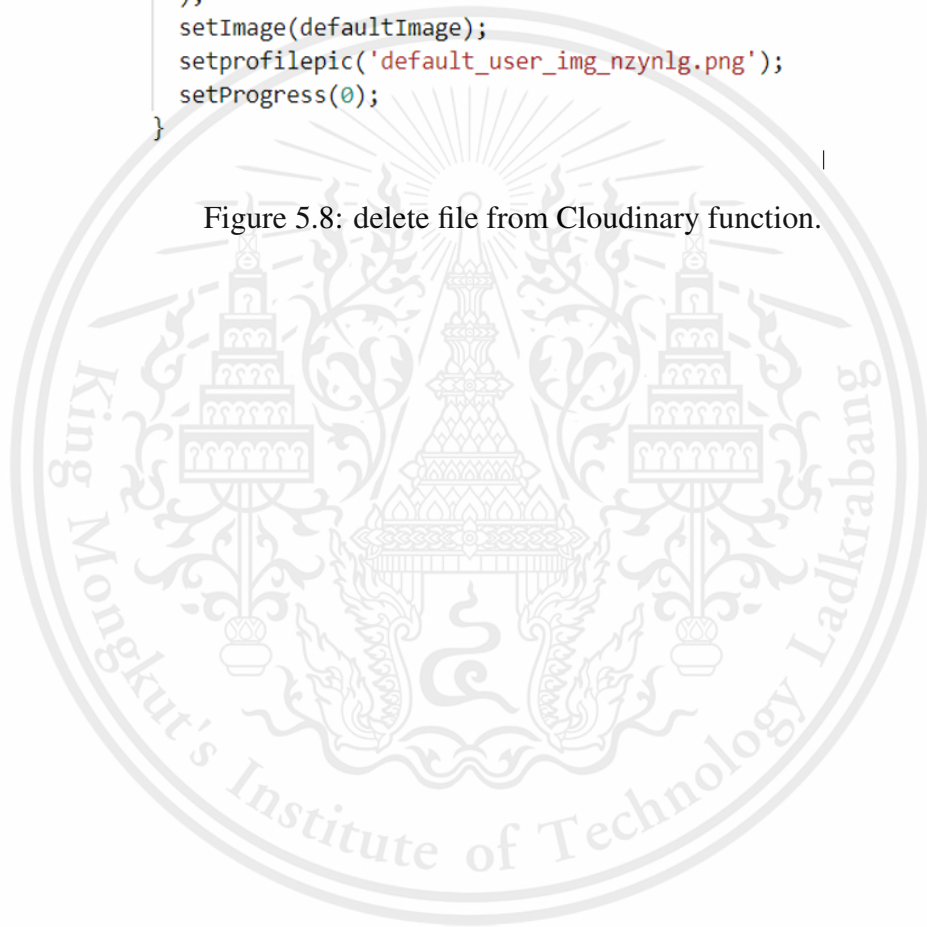
Figure 5.7: Upload file to Cloudinary function.

```
const deletepic = async () => {
  cloudinary.config({
    cloud_name: ██████████,
    api_key: ██████████,
    api_secret: ████████████████████
  });

  // console.log(Publicid);

  await cloudinary.uploader.destroy(Publicid,
    function(err, res) {
      console.log(res);
    }
  );
  setImage(defaultImage);
  setprofilepic('default_user_img_nzynlg.png');
  setProgress(0);
}
```

Figure 5.8: delete file from Cloudinary function.



5.1.3 Server-Side Development

For the development of server side we used Python as our main programming language and we use Django as our API framework which Django it self is a web framework, but Django can also serve as API framework. There are 3 main parts for our server side.

- Django as API framework
- Database
- User part and Recommendation part
- Email service

Django as API framework

In this project we use Django as an API framework because on the client-side, React already act as web framework. Django REST framework is an extension that we installed in order for Django to receive the RESTful API requests.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'corsheaders',  
    'rest_framework',  
    'StudyProgram_Recommendation_System',  
    'StudyProgram_Recommendation_System_Training',  
    'django_extensions'  
]
```

Figure 5.9: Django restful-framework extension in INSTALLED-APPS

Then, after we install Django RESTful-framework extension. We can now write Python code that can receive the request from client-side. In 'views.py' above the function is the label that tell Django which methods is for that particular function, One

function can have multiple methods.

```
@api_view(['GET'])
✓ def get_all_user(request):
✓   if request.method == 'GET':
       client = client_personal_info.objects.all()

       email = request.query_params.get('email', None)
       if email is not None:
           client = client.filter(email__icontains=email)

       client_serializer = StudyProgram_Recommendation_System_Serializer(client, many=True)
       return JsonResponse(client_serializer.data, safe=False)

@api_view(['POST'])
✓ def login(request):
    user = client_personal_info.objects.all()
    temp = StudyProgram_Recommendation_System_Serializer(user, many=True)
    user_list = temp.data
✓   if request.method == 'POST':
       client_data = JSONParser().parse(request)
       for i in user_list:
✓         if(i['email'] == client_data['email'] and i['password'] == client_data['password']):
           a = {}
           temp = {}
           a['status'] = True
           a['message'] = "Successfully login"
           temp['email'] = i['email']
           temp['age'] = i['age']
           temp['sex'] = i['sex']
           a['info'] = temp
           return JsonResponse(a)
    a = {}
```

Figure 5.10: POST and GET method in 'login' and 'get-all-user' functions

The API end-point is defined in another file, 'urls.py' file is where all the APIs end-point is stored. In each end-point it will refer to a function in the 'views.py' file.

```

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include(router.urls)),
    path('get_all_user/', views.get_all_user, name='get_all_user'),
    path('signup/', views.signup, name='signup'),
    path('login/', views.login, name='login'),
    path('logout/', views.logout, name='logout'),
    path('get_user/', views.get_user, name="get_user"),
    path('get/recommend/', views.recommend_result, name='recommend'),
    path('get/recommend_history/', views.recommend_history, name='recommend_history'),
    path('get_course/', views.get_course, name='get_course'),
    path('test_edit/', views.test_edit, name='test_edit'),
    path('edit/personal/info/', views.edit_personal_info, name='edit_personal_info'),
    path('edit/education/info/', views.edit_education_info, name='edit_education_info'),
    path('get/education/info/', views.get_education_info, name='get_education_info'),
    path('get/personality/info/', views.get_personality_info, name='get_personality_info'),
    path('personality_result/', views.personality_result, name='personality_result'),
    path('get_current_user/', views.get_current_user, name='get_current_user'),
    path('get_necessary/', views.get_necessary, name='get_necessary'),
    path('add_course/', views.add_course, name='add_course'),
    path('test_send_email/', views.test_send_email, name='test_send_email'),
    path('activate_account/', views.activate_account, name='activate_account'),
    path('reset_password/', views.reset_password, name='reset_password'),
    path('preset/', views.change_password_page, name='preset'),
    path('preset/change_password/', views.change_password, name='change_password')
]

```

Figure 5.11: APIs end-point in 'urls.py'

List of APIs

List of available APIs.

Path	HTTP Method	Action
get/all/user/	GET	Get all users in the database.
signup/	POST	Register new user.
login/	POST	Validate the email and password, then return the result.
logout/	POST	Clear user's current state from the system and save new data to the databases.
get_user/	GET	Get specific user from some conditions and return user's information.
get/recommend/	GET	Return user's newest recommended result.
get/recommend_history/	GET	Return all user's recommended result from the database.
get_course/	GET	Return all course from the database.
edit/personal/info/	POST	Edit user's personal information in the database.
edit/education/info	POST	Edit user's education record field in/from the database.
get/education/info	POST	Return user's education record from the database.
get/personality/info	POST	Return user's personality record from the database.
personality_result/	POST	Calculate user's personality trails and save the result in the database.
get_current_user/	GET	Get all currently online user.
get_necessary/	POST	Return the necessary information for first time login.
add_course/	POST	Add course in the database.
activate_account/	GET	Create a link for user to activate their account through email.
reset_password/	POST	Create a password reset request for user to reset their password.
preset/	GET	Get user new password then save it to the database
preset/change_password/	-	Return landing page when user completely reset their password.
get_course/	POST	Return the list of course from the database.
get_course_admin/	GET	Return the list of course along with it's necessary information.
add_admin/	POST	Add new user with the role of admin.
delete_course/	POST	Delete the selected course from the database.
delete_user/	POST	Delete user from the database
new_feedback/	POST	Create feedback and save the information as a new row inside the database.
get_feedback/	GET	Return the list of all feedbacks to admin.
predict/	POST	Predict the programs based on user's information.

Table 5.1: List of available APIs

Database

Django has an ability to automatically modify the database properties just from the python code. In Django, we needed to create database and tables in the file 'models.py'. In 'models.py' it store classes of table, one class for one table in the database. In the class we initialize the variables with their respective data types.

```
class client_personal_info(models.Model):  
  
    id = models.AutoField(primary_key=True)  
    firstname = models.CharField(max_length=50, default="Firstname")  
    lastname = models.CharField(max_length=50, default="Lastname")  
    email = models.CharField(max_length=50)  
    password = models.CharField(max_length=50)  
    age = models.IntegerField()  
    sex = models.CharField(max_length=10)  
  
    class Meta:  
        # app_label helps django to recognize your db  
        app_label = 'StudyProgram_Recommendation_System'
```

Figure 5.12: Client personal info class in 'models.py'

To make these code automatically create the table for us. Command 'python manage.py makemigration' and 'python manage.py migrate' is called respectively to make it do the rough work for us.

Our database of choice is PostgreSQL. It is a powerful, open source object-relational database system that uses and extends the SQL Language combined with many features that safely store and scale the most complicated data workload and it can be integrated with Django seamlessly.

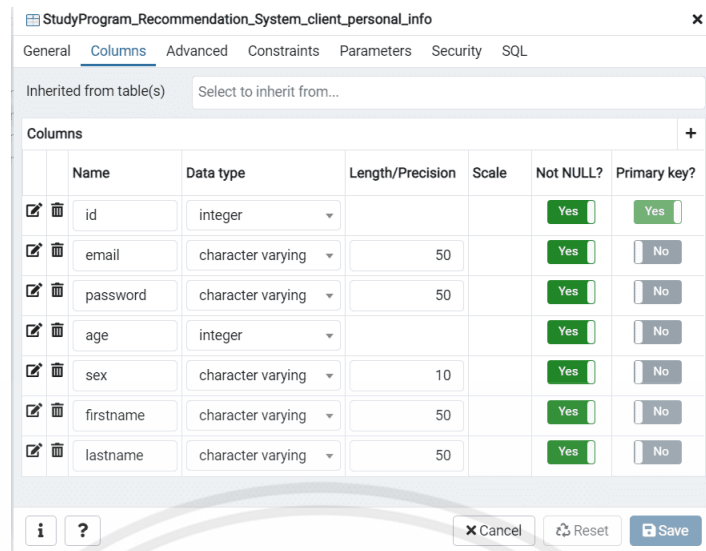


Figure 5.13: Client personal info table created using Django

User part and Recommendation part

In our system, we divided into 2 part. User part and Recommendation part. This 2 parts serve different purposes, User part will be responsible of handling all user action toward the system, but Recommendation part will be responsible for model training, recommendation, prediction, etc. So we decided to create 2 separated databases, One is for storing user information and another one is for storing training data, prediction model and many more information's that not related to client-side at all. To avoid confusion in query process and for backup if there is any accidents.


```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'  
EMAIL_HOST = 'smtp.gmail.com'  
EMAIL_USE_TLS = True  
EMAIL_PORT = 587  
EMAIL_HOST_USER = 'sprsysnoti@gmail.com'  
EMAIL_HOST_PASSWORD = [REDACTED]
```

Figure 5.15: Email setting in 'setting.py' (Blurred out for security reason)

We decide to use Gmail to be our host email. To use this email, first we need to declare the recipient email, then the subject of the email, lastly the content of the email. Content of the email can be either plain text or the HTML pages that have content inside. So we choose the HTML page as the template of the content so we can add the user's information and links in before sending.

```
recipient = client_data['email']  
subject = 'Confirm your account on SPR System'  
html_message = render_to_string('./email_template.html', {'link': link})  
plain_message = strip_tags(html_message)  
email_from = settings.EMAIL_HOST_USER  
recipient_list = [recipient,]  
send_mail(subject, plain_message, email_from, recipient_list,html_message=html_message)
```

Figure 5.16: Usage of email service

5.1.4 Big 5 Personality test scoring implementation

We've implement Big 5 personality test to our system to calculate the personality score of the user categorized by the types which consist of Conscientiousness, Agreeableness, Neuroticism, Openness to Experience and Extraversion. The question set that we used will divided the question to match the types of personality and arranged in random format. Then when the user answer the question they won't by bias toward curtain types. Then when the answer were sended to the server we will calculate the score of each traits and wight it out in to the score out of 100.

```
def personality_result(request):
    global current_user
    req = JSONParser().parse(request)
    tok = req['token']
    req_answer = req['answer']

    #e
    # 1, 6, 11, 16, 21, 26, 31, 36
    #a
    # 2, 7, 12, 17, 22, 27, 32, 37, 42
    #c
    # 3, 8, 13, 18, 23, 28, 33, 38, 43
    #n
    # 4, 9, 14, 19, 24, 29, 34, 39
    #o
    # 5, 10, 15, 20, 25, 30, 35, 40, 41, 44

    #Special : 2,6,12,18,21,23,24,27,31,34,35,37,41,43
    special = [2,6,12,18,21,23,24,27,31,34,35,37,41,43]
    o = [5, 10, 15, 20, 25, 30, 35, 40, 41, 44]
    c = [3, 8, 13, 18, 23, 28, 33, 38, 43]
    e = [1, 6, 11, 16, 21, 26, 31, 36]
    a = [2, 7, 12, 17, 22, 27, 32, 37, 42]
    n = [4, 9, 14, 19, 24, 29, 34, 39]
    type_o = []
    type_c = []
    type_e = []
    type_a = []
    type_n = []
```

Figure 5.17: Personality test scoring algorithm

5.2 Recommendation System

5.2.1 Methodology

In this section the methodology use for this project is being discussed here. The method is divided into two part, the first part talk about using the classification method and other and by using regression analysis. We perform the test and select the best method.

5.2.2 Classification method

Classification description

The problem is too classified the student into 4 group based on their grade in which the first group refers to the grade below 25 percentile second group refers to the grade between 25 to 50 percentile and third group is between 50 to 75 percentile and the last group is above the 75 percentile of the total grade based on specific faculty and department.

Data extraction

The data set is taken from KMITL register office with all the student being unknown. The student name and the information which could lead to the student is remove. The register office change the student number to hashedId which is using some encoder and was not known to us therefore, the privacy of the student is taken care of.

The data is given in two excel file with different sheet. The first excel, it tell about the student grade and some information before entering the college, the sheet is separated by the student entry year between 2558 to 2562. Similarly the excel is the student information about the grade during the college.

We combine and merge different sheet of first excel to one file having the student data from the year 2558 to 2562 to csv file, then the college GPA is taken from the second excel. We take the consideration of only last year in second excel as the excel tell about GPA not GPS. The GPA is further merge with the current csv to form one full

csv file consisting of all student information,

Data pre-processing

From the observation, we did not remove the outlier as the data was small but categories type variable is converted to numeric type ranging from 0 to the total categories. Feature scaling was not necessary.

We group the school together as there were about 1500 school in the data set so we divide school into 4 group with similar grade by taking the mean of the grade of all student in that school. Null value and unknown school are remove. Also the categories variable is converted to numerical value by using the decoder. than the data is further used for training and testing .

Algorithm selection

The label of the data set is given so this model can be classified using supervised learning algorithm where test and train was split in ration 70:20 for verification of the algorithm. To select the algorithm, we train the model using classification algorithm with complex learning such as Support Vector machine, Gradient Boosting , Random Forest, Ada boost and Decision tree and then the algorithm with the least mean square error is selected and the selected algorithm was Random forest.

Parameter tuning

The parameter tuning is done to give the highest accuracy. It is done by exhaustive search and select the hyperparameter that suit the best for the data. In this we randomly selected different group of parameter given by the sklearn library for Random foest model.

Testing

Testing is done to check the accuracy of data. It is done using both cross validation and matrix confusion.

5.2.3 Regression analysis

Regression description

Regression analysis is used to estimate the relationships between a dependent variable and independent variables. The dependent variable is the variable that depends on other factors and independent variables is the variable that stand alone and does not depend on other factor, for this project the dependent variable is grade of the student during the college regarding to theirs specific course and the dependent variables are the grade of the student at high school, the number of teacher,number of student,number of working computer,number of classroom and lastly is the location of the school.

Data extraction

The student data is taken from the same source as the above but we also include the data that contain the information of the school.The data was scrapped from the website called "<https://data.bopp-obec.info/emis/>" using selenium and python. The attribute that we scrap includes the total number of teacher in that school, the total number of student,the number of classroom and also the number of computer in that school. and we merge the school information to each student who studies in the specific school.

Data processing

As the information the school that we scrap was limited so we consider only the student that studies in the school in which information was available for this we have around 11000 row in the data set.Instead of taking the school name directly we take the school information such as number of student,number of teacher, number of classroom, number of working computer and also the location of the school.the dummy is created for the location of the separate by the 6 direction such as north, south, west, east, north east ,western.

5.2.4 Algorithm selection

The label of the data set is given so this model can be predict using supervised learning algorithm where test and train was split in ration 70:20 for verification of the algorithm. To select the algorithm we take a good look into a data and the GPA high school and GPA at college has a linear correlation so the multiple linear regression to predict the GPA of the student.

Testing

Testing is done to check the accuracy of data. It is done by checking the R square value and also the MAPE (mean Absolute Percentage Error).

5.2.5 Dataset information

The dataset is divided into two table the first table consist of 5 attributes include both contiuous and categories variable with no missing value.The 5 attributes consist of :-

- **department:** This include the department in the kmitl college for example: engineering department,architectural department,business administration department etc.The data consist of 15 differnt department.
- **faculty:** This include the faculty in the KMITL. It is the subset of department such as software engineering,finance etc.The data consist of 107 different faculty.
- **school:** This is the school where the student are graduated from.The data consist of 1449 different school.
- **GPA at school:** This is the GPA of the student after graduated from the school.
- **GPA:** This is the GPA of the student when studying in KMITL college.
- **GPA(group):** The GPA of the student is divided into different quantile based on the percentile value of 25,50,75 respectively.

The second table consist of the properties of the school having four attributes . The attributes consist of :-

- **school:** This is the school in Thailand.
- **total teacher:** This is the total number of teacher in the school.
- **total student:** This is the total number of student in the school.
- **total classroom:** This is the total number of classroom in the school.
- **total working computer:** This is the total number of working computer in the school.

5.2.6 Classification

By considering the problem as classification problem, the school is directly encoded to numerical value but the school is classify into 4 categories based on the average grade of the student in that school. The GPA is also group into 4 categories based on their grade with percentile 25,50,75. The data is then put into classification model for supervised learning using different classification algorithm and select the best model for prediction.

5.2.7 Regression

By considering the problem as regression problem, the school cannot be converted to numerical directly because regression take value as distance function making a worst performance model when trying predict continuous value. Therefore instead of taking the school directly into the model we take the properties of school such as number of teacher in the school number of student and also the number of class room.The above is taken into the consideration of the model and preform multiple regression to obtained predicted grade.

5.2.8 Train and test

Before entering the model We randomly select a row from the prepare dataset and separate them into train and test having the ratio 70:20 where 70 % of the data is used

to create a model and 20 % of the data is used for verification of the model

5.2.9 Classification model

Create model by using the attributes 'GPA_group' as the labels and split into train and test accordingly to the size of the predictor variable. The algorithm used here is random forest. We have used the library from Sklearn.ensemble to create model with Random decision Forest and the parameter is adjusted accordingly to the defined parameter.

5.2.10 Regression model

Create model by using the attributes 'GPA' as the labels and split into train and test accordingly to the size of the predictor variable. The algorithm used here is multiple regression. We have used the library from Sklearn.ensemble to create model with multiple regression and the parameter is adjusted accordingly to the defined parameter.

5.2.11 Evaluate model performance

The final step of the machine learning problem is to evaluate the performance of the algorithm. In this classification problems the confusion metrics and precision recall is used to evaluate an algorithm are accuracy, and for regression we used the r square value and the MAPE to evaluate the accuracy.

Chapter 6

Results

6.1 Web Application

6.1.1 Home Page

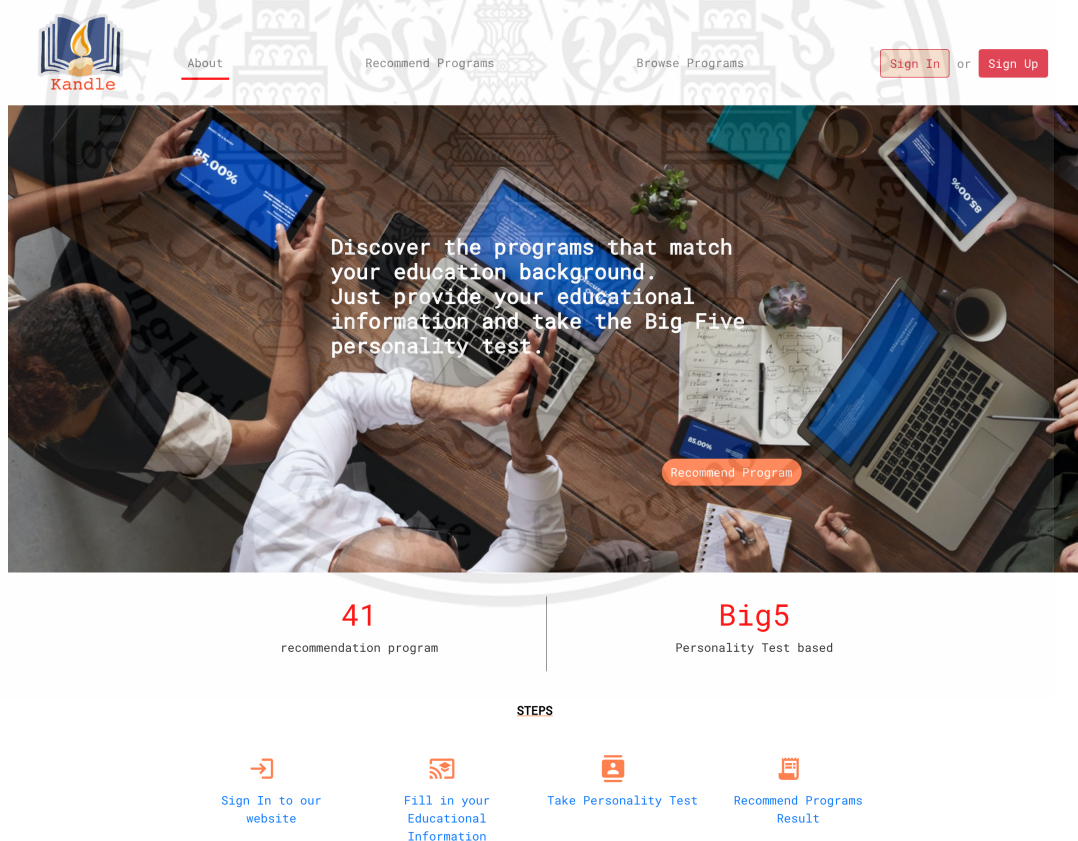


Figure 6.1: Home Page

Home page when the user has not sign in to the website.

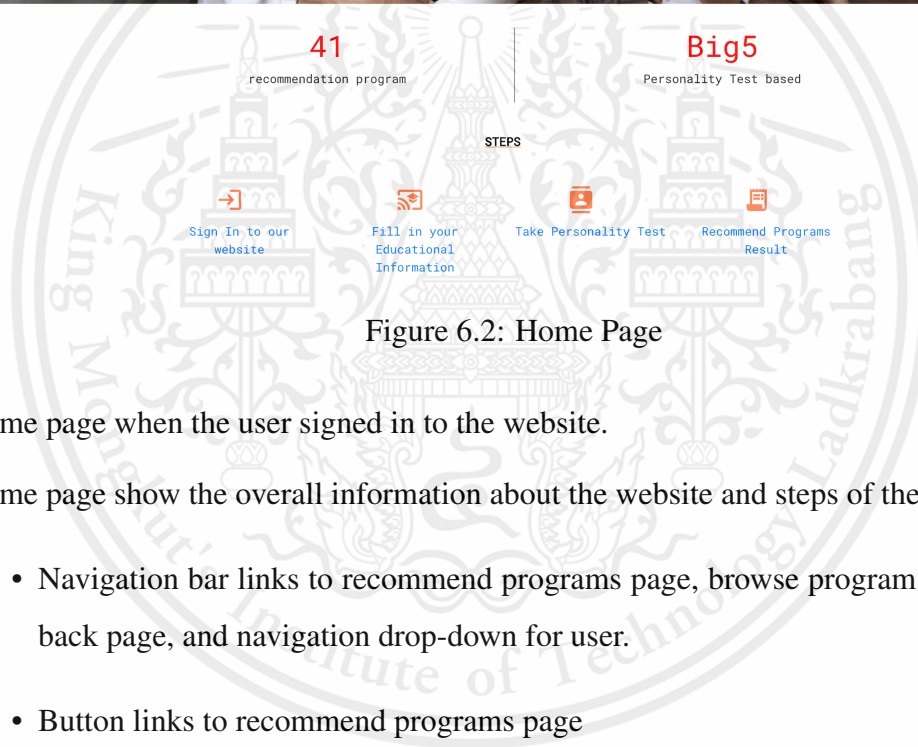
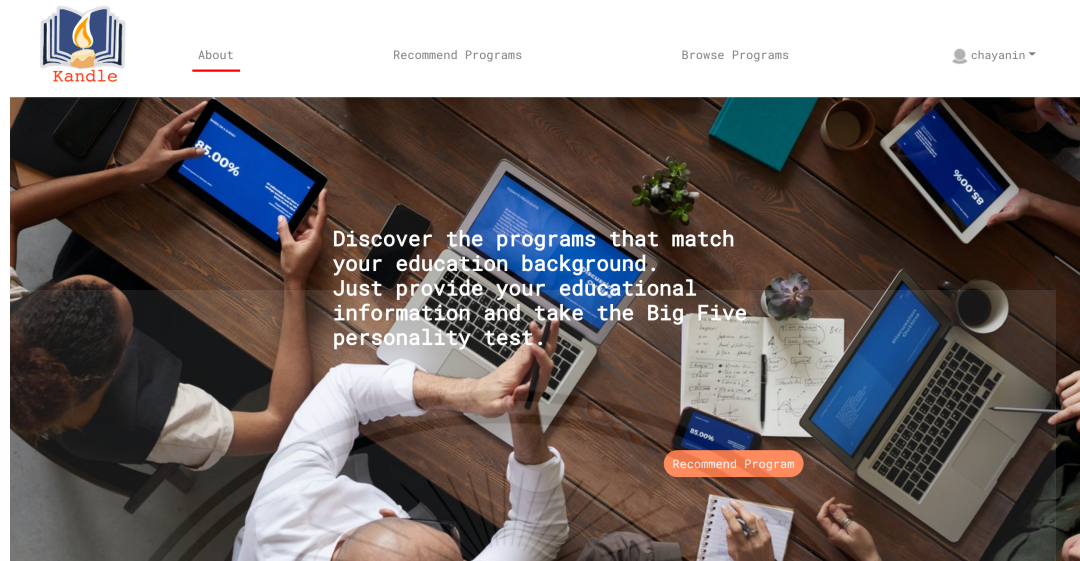


Figure 6.2: Home Page

Home page when the user signed in to the website.

Home page show the overall information about the website and steps of the website:

- Navigation bar links to recommend programs page, browse program page, feed-back page, and navigation drop-down for user.
- Button links to recommend programs page

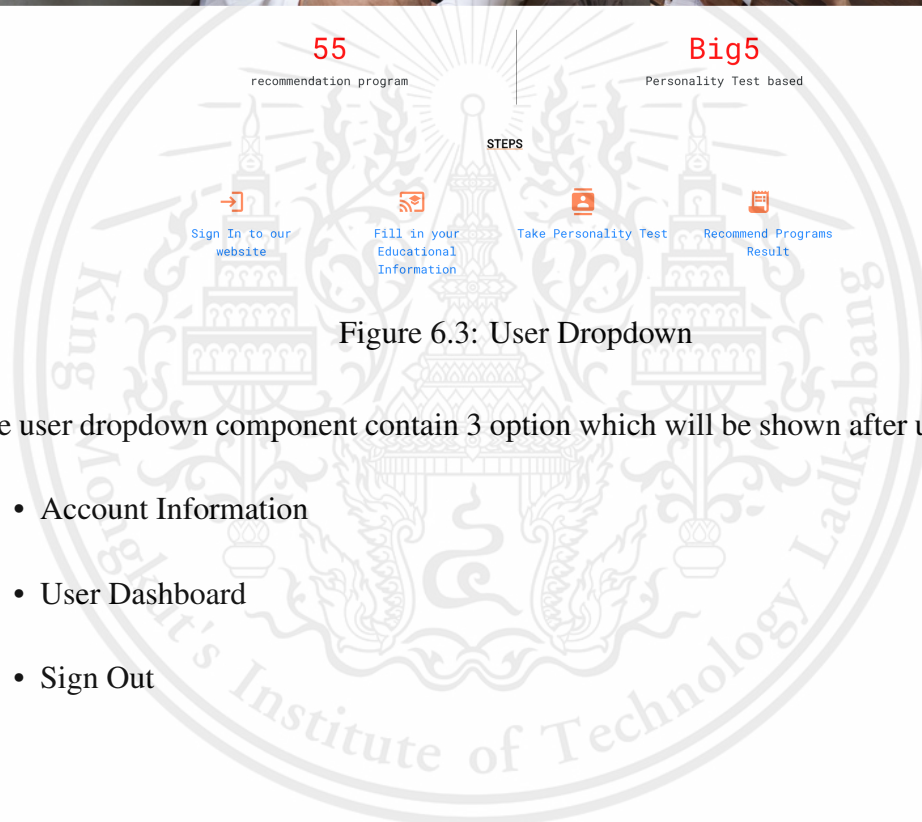
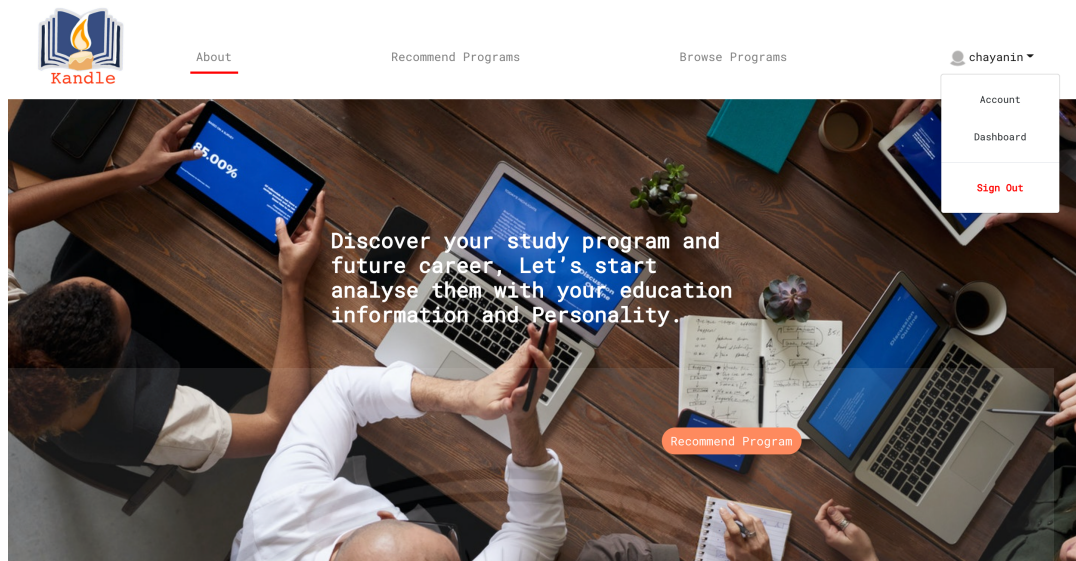
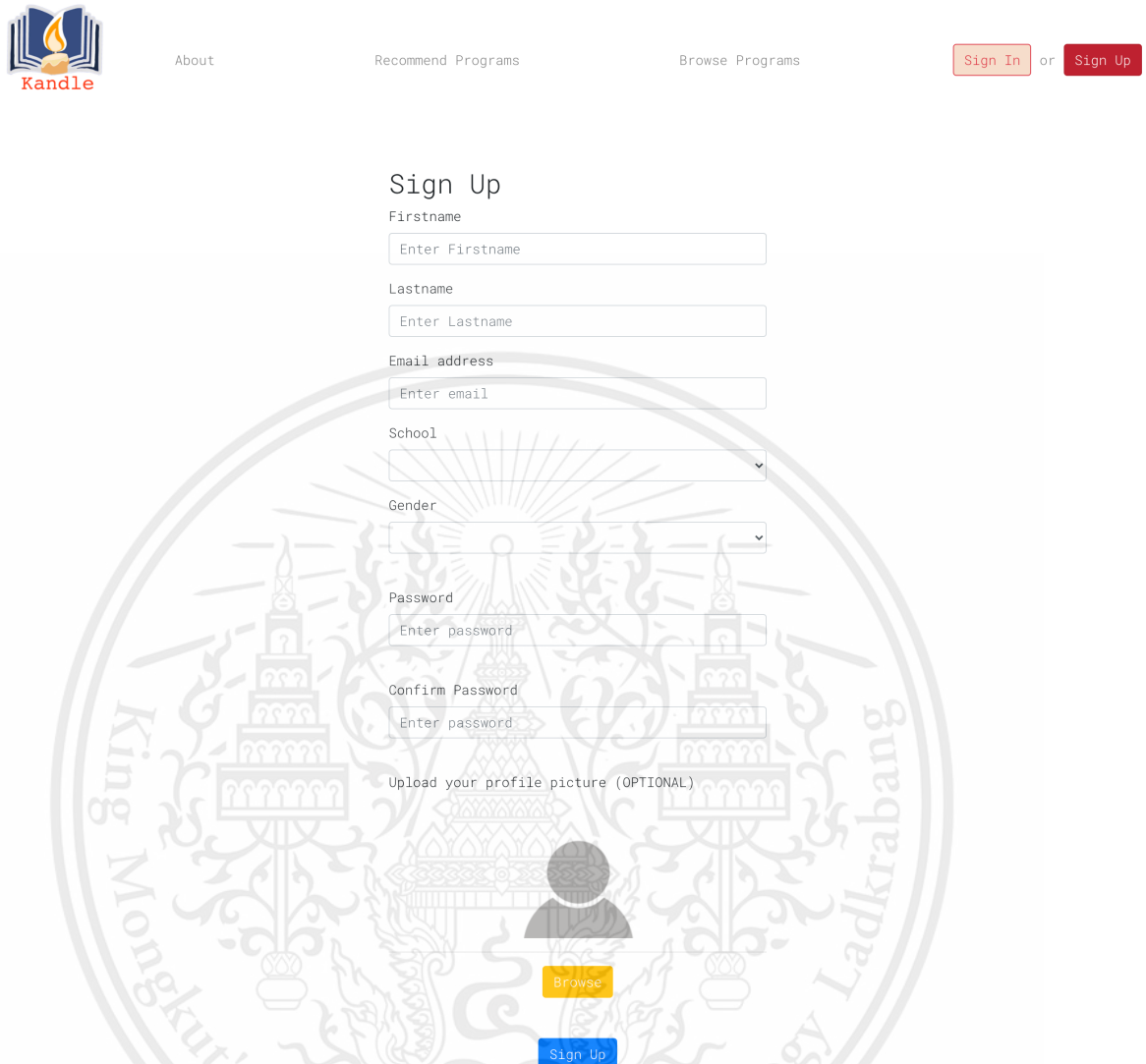


Figure 6.3: User Dropdown

The user dropdown component contain 3 option which will be shown after user sign in:

- Account Information
- User Dashboard
- Sign Out

6.1.2 Sign in/ Sign Up Page



Kandle

About Recommend Programs Browse Programs Sign In or Sign Up

Sign Up

Firstname
Enter Firstname

Lastname
Enter Lastname

Email address
Enter email

School
▼

Gender
▼

Password
Enter password

Confirm Password
Enter password

Upload your profile picture (OPTIONAL)

Browse

Sign up

Figure 6.4: Sign Up page

Sign Up page allows users to sign up to our website

- Users are required to enter their email address, gender, school, and password.
- The information that user enter will be use in our program recommendation system.



About

Recommend Programs

Browse Programs

Sign In

or Sign Up

Sign In

Email address

Password

Submit

Don't have an account? [Sign Up](#)

[Forgot password?](#)

Figure 6.5: Sign In page

Sign In page allows both user and admin to sign in:

Sign In page also links user to forgot password page and sign up page

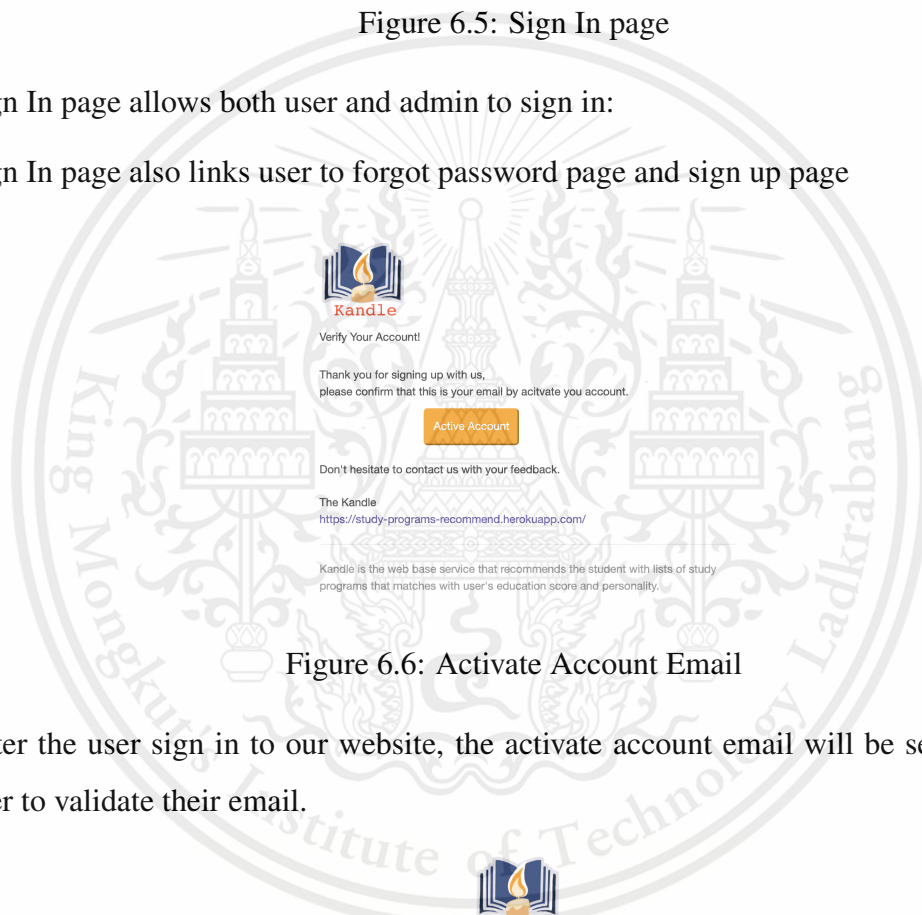


Figure 6.6: Activate Account Email

After the user sign in to our website, the activate account email will be send to each user to validate their email.



Welcome to Kandle

Your account has been activated please follow the following link to enter our website:
<https://study-programs-recommend.herokuapp.com/>

Thank you for using our service.

Copyright ©2021 | All Rights Reserved

Figure 6.7: Welcome

After the user click activate in the activate account email, they will be lead to this page to enter the website.



Forgot Your Password?

Please fill in your email address and we will send you the link to reset your password.

Email address

Reset Password

Back

Figure 6.8: Forgot Password page

Forgot password page allow user to receive their new password by entering their email address and the system will send users with links to change their password.

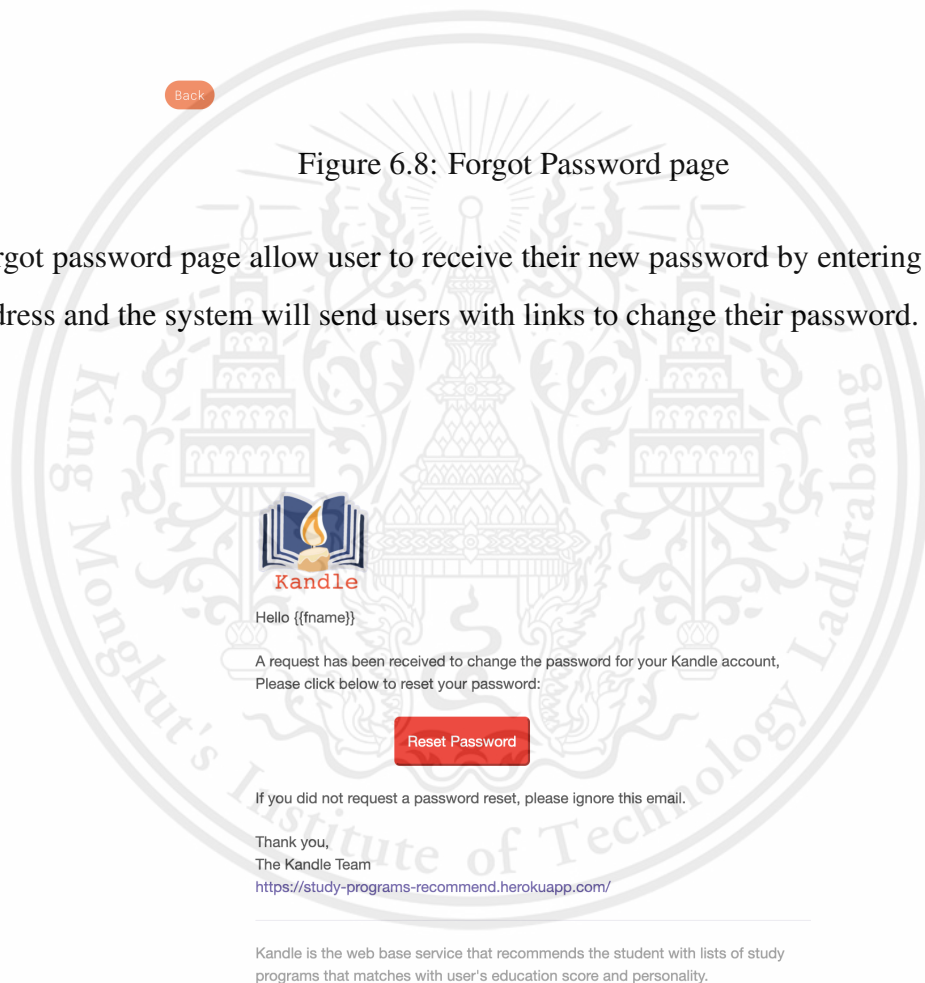
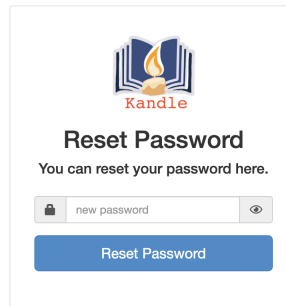


Figure 6.9: Email for reset password

If the user enter valid email, then they will receive an email to confirm that they want to reset password.



Kandle is the web base service that recommends the student with lists of study programs that matches with user's education score and personality.

Figure 6.10: Enter new password

The user will be ask to enter new password for their account.

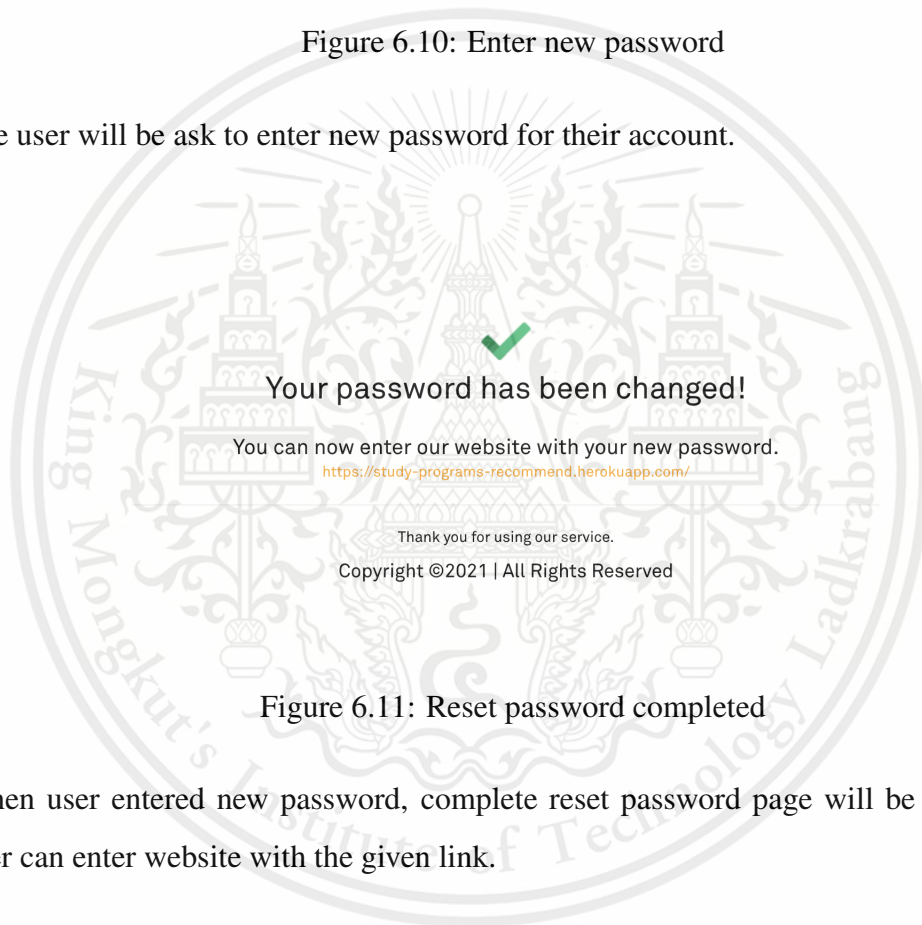


Figure 6.11: Reset password completed

When user entered new password, complete reset password page will be shown and user can enter website with the given link.

6.1.3 Browse Program Page

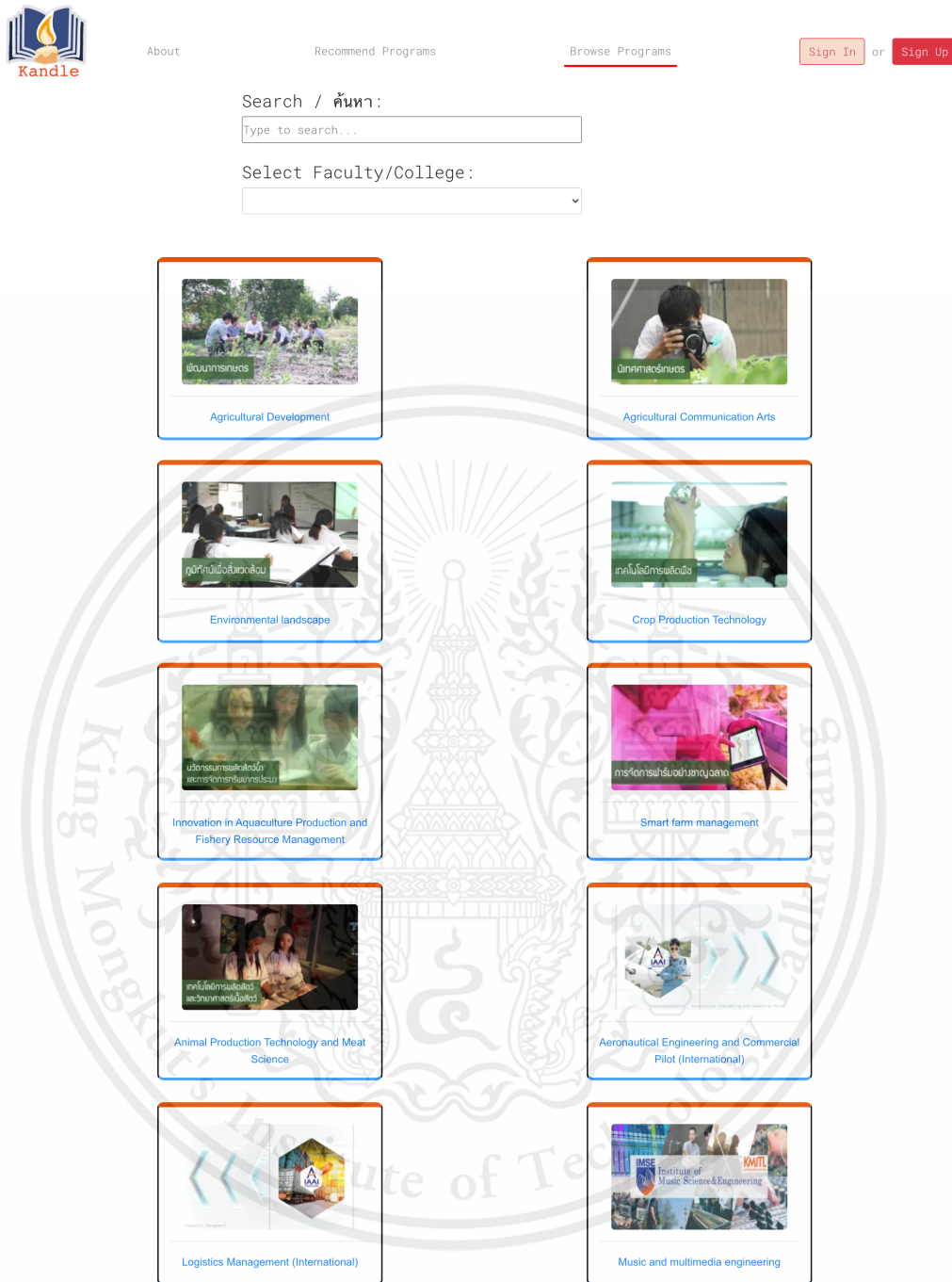


Figure 6.12: Browse Program Page

The list of the programs will be shown on this page.

User can browse their interested programs through search box and dropdown.

6.1.4 Account Information Page



About

Recommend Programs

Browse Programs

chayanin

User Information

Profile image

The form is overlaid on a large, semi-transparent watermark of the King Mongkut's Institute of Technology Ladkrabang logo. The form includes a profile picture placeholder, an upload status indicator, and several input fields for personal information.

Upload status: 0%

Browse

Save change Cancel

Firstname
chayanin

Lastname
sutjaritvanichpongse

Gender
Female

School
School1

Save change

CHANGE EMAIL
Current Email
Enter Current Email

New Email
Enter New Email

Save change

CHANGE PASSWORD
Current Password
Enter Current Email

New Password
Enter New Email

Confirm New Password
Enter New Email

Save change

Figure 6.13: Account Information Page

Account Information page allows user to edit their information such as their first name, last name, gender, and nationality.

Account Information page also allows user to edit:

- First Name
- Last Name
- Gender
- School
- Email Address
- Password

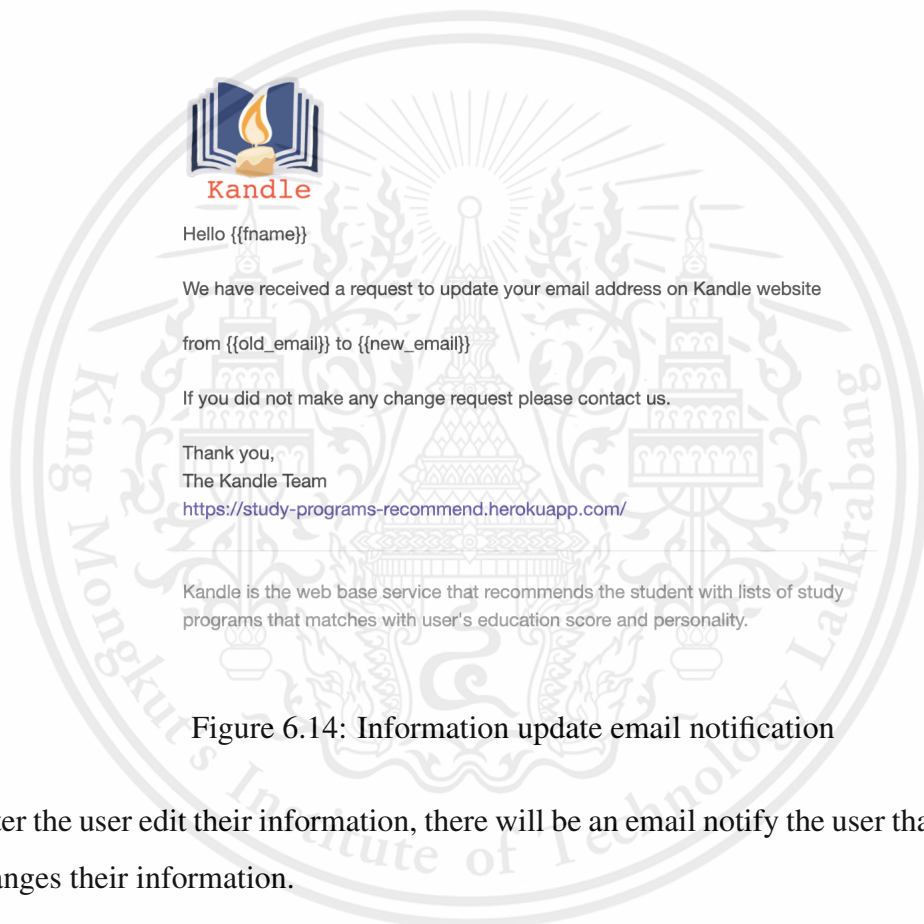


Figure 6.14: Information update email notification

After the user edit their information, there will be an email notify the user that they have changes their information.

6.1.5 User Dashboard Page

Result History

Date	Recommended Program	Predicted Grade
2021-04-19	วิศวกรรมระบบควบคุม	80.8
	วิศวกรรมคณิศร	70.8
	วิศวกรรมอาหาร	55.8

Date	Recommended Program	Predicted Grade
2021-05-05	วิศวกรรมระบบควบคุม	80.8
	วิศวกรรมคณิศร	70.8
	วิศวกรรมอาหาร	55.8

Date	Recommended Program	Predicted Grade
------	---------------------	-----------------

Educational Information

GPAX	3.75
GAT	200
PAT1	150
PAT2	200
PAT3	0
PAT4	0
PAT5	0
PAT6	0
PAT7	0
SAT Math	450
SAT English	500
IELTS	8.0

[Edit Information](#)

Big 5 Personality Label

[Take test](#)

Figure 6.15: User Dashboard Page

Consists of 4 components about user which are:

- Result history
- Feedback box

- program list dropdown
- feedback input box
- Educational Information
- Big 5 Personality Label



6.1.6 Recommendation Programs Page

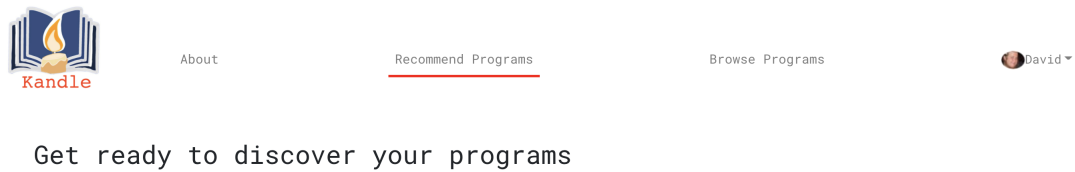


Figure 6.16: Incomplete Recommendation Page

When user have not complete their educational information and have not take personality test, "not complete" label will be shown below.

- First component allows user to enter their educational information by pressing "Enter Information" button.
- Second component allows user to take their personality test by pressing "Take Test" button.
- "Recommend Programs" button, will be activate only when user enter educational information, take personality test, and choose the preference program.

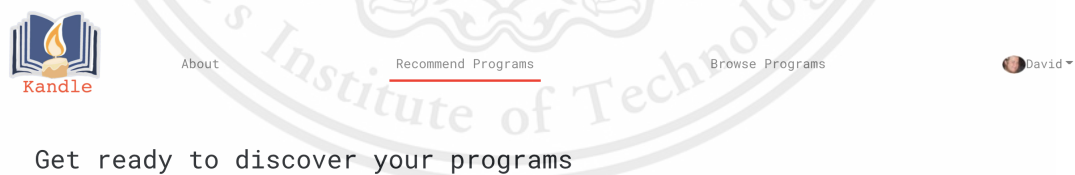
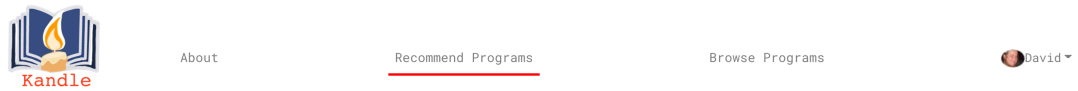


Figure 6.17: Complete label

When user entered educational information, "complete" label will be shown below.



Get ready to discover your programs

1
Enter your Educational Information
Enter Information
Complete

2
Take Big 5 Personality Label test
Take test
Complete

Available Programs
Please select 5 programs that you're interested in from the list below

Recommended programs based on your personality traits

- บริหารธุรกิจบัณฑิต
Bachelor of Business Administration
- วท.บ. เศรษฐศาสตร์ธุรกิจและการจัดการ
B.Sc. Business and Managerial Economics
- ศ.อ.บ. วิศวกรรมออกแบบสภาพแวดล้อมภายใน
B.S.Ind.Ed. Interior Environmental Design
- ศ.อ.บ. วิศวกรรมวิศวกรรม
B.S.Ind.Ed. Engineering Education
- สด.บ. สถาปัตยกรรมภายใน
B.Arch. Interior Architecture
- วศ.บ. วิศวกรรมอุตสาหกรรม
B.Eng. Industrial Engineering
- วศ.บ. วิศวกรรมระบบการผลิต
B.Eng. Manufacturing System Engineering
- ก.สด.บ. ภูมิสถาปัตยกรรม
Bachelor of Landscape Architecture
- ศศ.บ. นวัตกรรมการท่องเที่ยวและการบริการ
B.A. Innovation in Tourism and Hospitality

Selected Programs

Figure 6.18: Preference program

When user entered both educational information and done personality test, "complete" label will be shown below.

- User will be allow to choose 5 preference programs.
- Preference programs will be listed on the left hand side for user to choose.
- The preference programs based on big 5 personality label will be first shown followed by other alternative programs for user to choose.

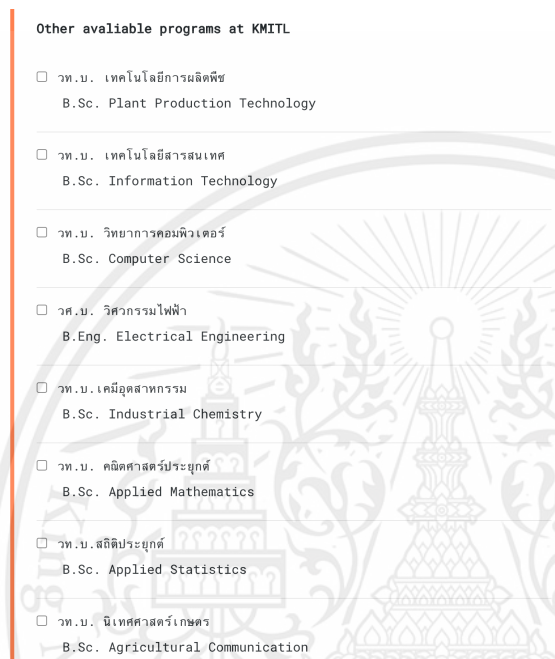


Figure 6.19: Preference program

When user entered both educational information and done personality test, "complete" label will be shown below.

- Other programs will also be available for user to choose.

Get ready to discover your programs

1

Enter your Educational Information

[Enter Information](#)

Complete

2

Take Big 5 Personality Label test

[Take test](#)

Complete

Available Programs	Selected Programs
<p>Please select 5 programs that you're interested in from the list below</p>	
<p>Recommended programs based on your personality traits</p>	
<input type="checkbox"/> วิทยาลัยการผลิต B.Sc. Plant Production Technology	
<input type="checkbox"/> วิทยาลัยสารสนเทศ B.Sc. Information Technology	
<input type="checkbox"/> วิทยาการคอมพิวเตอร์ B.Sc. Computer Science	
<input type="checkbox"/> เคมีอุตสาหกรรม B.Sc. Industrial Chemistry	
<input type="checkbox"/> คณิตศาสตร์ประยุกต์ B.Sc. Applied Mathematics	
<input type="checkbox"/> สถิติประยุกต์ B.Sc. Applied Statistics	
<input type="checkbox"/> นิเทศศาสตร์อักษร	

Figure 6.20: Preference program

The 5 selected programs will be shown on the right hand side.

Other available programs at KMITL
<input type="checkbox"/> บริหารงานทั่วไป Bachelor of Business Administration
<input type="checkbox"/> เศรษฐกิจบริหารธุรกิจ B.Sc. Business and Managerial Economics
<input type="checkbox"/> วิศวกรรมไฟฟ้า B.Eng. Electrical Engineering
<input type="checkbox"/> วิศวกรรมโทรคมนาคมและโครงข่าย B.Eng. Telecommunications and Network Engineering
<input type="checkbox"/> วิศวกรรมอิเล็กทรอนิกส์ B.Eng. Electronics Engineering
<input type="checkbox"/> วิศวกรรมคอมพิวเตอร์ B.Eng. Computer Engineering
<input type="checkbox"/> สาขาวิชาออกแบบภายใน B.S.Ind.Ed. Interior Environmental Design

Figure 6.21: Preference program

- The other alternative available program will be shown below the recommend program based on the user's big 5 personality label.

<input type="checkbox"/>	ค.อ.บ. วิศวกรรมเกษตร B.S.Ind.Ed. Agricultural Education
<input type="checkbox"/>	วท.บ. เทคโนโลยีสารสนเทศ B.Sc. Information Technology
<input type="checkbox"/>	วท.บ. พัฒนาการเกษตร B.Sc. Agricultural Development
<input type="checkbox"/>	วท.บ. วิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ B.Sc. Data Science and Business Analytics
<input checked="" type="checkbox"/>	วศ.บ. วิศวกรรมเคมี B.Eng. Chemical Engineering

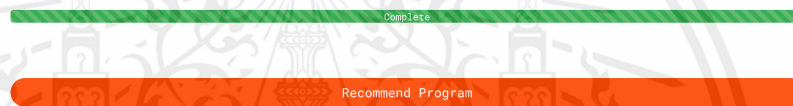


Figure 6.22: Preference program

- When user selected 5 program then the progress bar will be label as "complete"
- "Recommend Program" button will also be activate for user to press.

6.1.7 Educational Information Page



About

Recommend Programs

Browse Programs

chayanin

Education Information

GPAX

3.75

SAT Math

450

SAT English

500

IELTS

8.0

Edit Score

GAT/PAT Section

GAT - General Aptitude Test

200

PAT 1 - Aptitude in Mathematics

150

PAT 2 - Aptitude in Scientifics

200

PAT 3 - Aptitude in Engineering

0

PAT 4 - Aptitude in Architectural

0

PAT 5 - Aptitude in Teacher professional

0

PAT 6 - Aptitude in Arts

0

PAT 7 - Aptitude in Foreign language proficiency (French, German, Chinese, Japanese, Arabic, Pali)

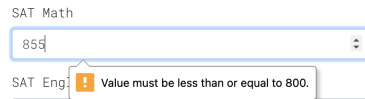
0

Edit GAT/PAT Score

Back

Figure 6.23: Education Information (All)

Educational Information page allow user to enter their educational information in the input box.

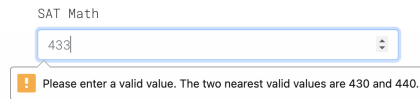


SAT Math
85

SAT Eng: Value must be less than or equal to 800.

Figure 6.24: Education Information (Range)

The input box only allow user to input in particular range of score based on which score the users are entering or else the alert will be shown.

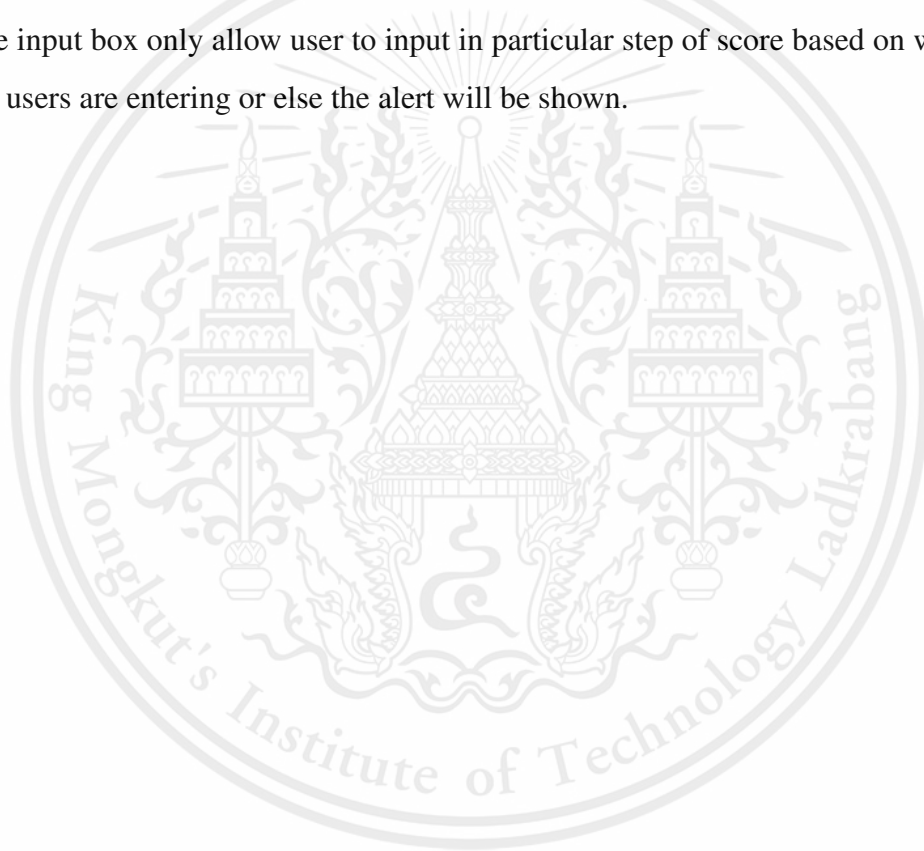


SAT Math
43

Please enter a valid value. The two nearest valid values are 430 and 440.

Figure 6.25: Education Information (Step)

The input box only allow user to input in particular step of score based on which score the users are entering or else the alert will be shown.



6.1.8 Personality Information Page

Answer all 44 questions to see your Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism score which labeled based on Big 5 Personality.
Rate each statement according to how it describes you, where 1 = Disagree, 3 = Neutral, and 5 = Agree.

Personality Test

Is talkative
1 2 3 4 5

Tends to find fault with others
1 2 3 4 5

Does a thorough job
1 2 3 4 5

Is depressed, blue
1 2 3 4 5

Is original, comes up with new ideas
1 2 3 4 5

Is reserved
1 2 3 4 5

Is helpful and unselfish with others
1 2 3 4 5

Can be somewhat careless
1 2 3 4 5

Is relaxed, handles stress well
1 2 3 4 5

Is curious about many different things
1 2 3 4 5

Is full of energy
1 2 3 4 5

Starts quarrels with others
1 2 3 4 5

Figure 6.26: Personality Test Page

Personality Test page allow user to take the personality test which consists of 44 question based on big 5 personality label.

Big 5 Personality Result

Personality Label	Score
Extraversion	40
Neuroticism	48
Agreeableness	56
Conscientiousness	57
Openness	68

Figure 6.27: Personality Test Result

Personality Test Result Page show the score based on the big 5 personality label

Recommend Programs Results

Possible Programs	
สอ.บ. สถาปัตยกรรมภายใน B.Arch. Interior Architecture	★★★★★★ Predicted grade: 3.56
ค.อ.บ. วิศวกรรมศาสตรบัณฑิต สถาปัตยกรรมภายใน B.S.Ind.Ed. Interior Environmental Design	★★★★★★ Predicted grade: 3.47
บริหารธุรกิจบัณฑิต Bachelor of Business Administration	★★★★★ Predicted grade: 3.1
ท.บ. วิทยาการคอมพิวเตอร์ B.Sc. Computer Science	★★★★★ Predicted grade: 3.08
ท.บ. วิทยาการข้อมูลและการ วิเคราะห์เชิงธุรกิจ B.Sc. Data Science and Business Analytics	★★★★ Predicted grade: 2.67

[Done](#)

Figure 6.28: Study Program Recommendation Result Page

Study Program Recommendation Result Page show the recommended program result after the user have entered their educational information and take the personality test. .

6.1.9 Admin Page

ID	Faculty : ภาควิชา	Department : ภาควิชา	Program : หลักสูตร
1	Faculty of Agricultural Technology	Science	Agricultural Development
2	Faculty of Agricultural Technology	Science	Agricultural Communication Arts
3	Faculty of Agricultural Technology	Science	Environmental landscape
4	Faculty of Agricultural Technology	Science	Crop Production Technology
5	Faculty of Agricultural Technology	Science	Innovation in Aquaculture Production and Fishery Resource Management
6	Faculty of Agricultural Technology	Science	Smart farm management
7	Faculty of Agricultural Technology	Science	Animal Production Technology and Meat Science

Figure 6.29: Admin Program List

Admin Program list page allows admin to see lists of programs which the page consists of: .

- Search box, allows admin to search for program id and name.
- Drop-down on the left, allows admin to choose number of entries they want to see.
- "Add Program" button, allows admin to add new program to the lists.
- "trash bin" button, allows admin to delete program from the lists.

Figure 6.30: Pop up input for new program

Figure 6.31: Pop up input for new program

Feedbacks from our users!

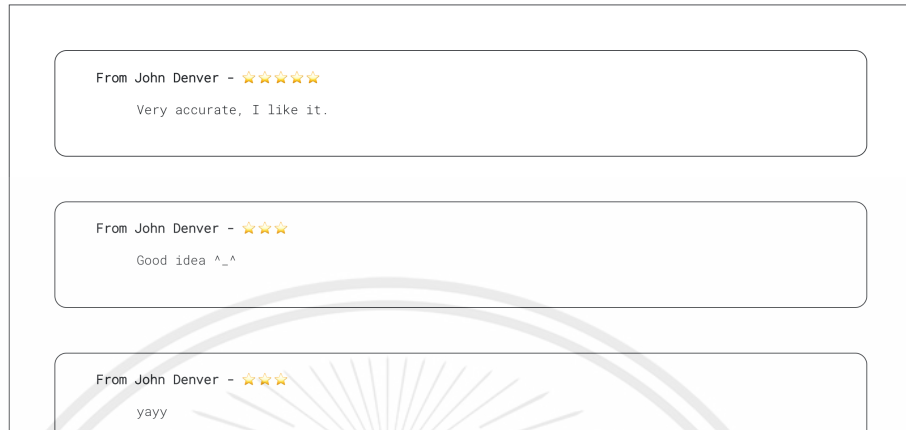


Figure 6.32: Feedback from user

Feedback page show lists of feedback from the user of our recommendation system with rating stars and comments.

6.2 Experimental results

6.2.1 classification algorithm testing

Using cross validation

The method that are taken here are logistic regression, LinearDiscriminantAnalysis, Random forest To check for the accuracy first we used cross validation to compare the accuracy

	Accuracy
RandomForest	34.34
logistic regression	32.5
LinearDiscriminantAnalysis	34.88

Table 6.1: Cross validation

Using confusion matrix

Similarly logistic regression, LinearDiscriminantAnalysis, Random forest is compare using confusion matrix:-

	1predict	2predict	3predict	4predict
1Acutal	323	219	137	94
2Actual	225	206	173	165
3Actual	165	219	216	227
4Actual	111	133	195	335

Table 6.2: Matrix confusion for Random Forest

	1predict	2predict	3predict	4predict
1Acutal	357	104	14	298
2Actual	234	118	12	407
3Actual	176	112	11	522
4Actual	145	86	11	532

Table 6.3: Matrix Logistic regression

From this you can see that Random forest out perform support vector machine and Ada boost But it is not enough as you can see that there are large interval between the 3group

	1predict	2predict	3predict	4predict
1Actual	430	122	17	204
2Actual	304	134	19	314
3Actual	231	150	24	422
4Actual	146	91	28	509

Table 6.4: LinearDiscriminantAnalysis

prediction and fourth group. As the prediction could be around the end of 2 group and starting of the 4 group that why the prediction rarely go the third group. This is the reason that we use regression to get the accurate value.

6.2.2 Regression testing

By using the regression, considering overview of all the program the R square value is around 0.18 meaning that our model can explain only 18 percent of total data, but the residual is a bell curve shape also the qq plot has value pass through most of the residual showing that even if all predictors are used, the r-square value also ranges in the same number and also our model uses predictors of high significance, so our model explains how changes in response values are associated with changes in predictor values.

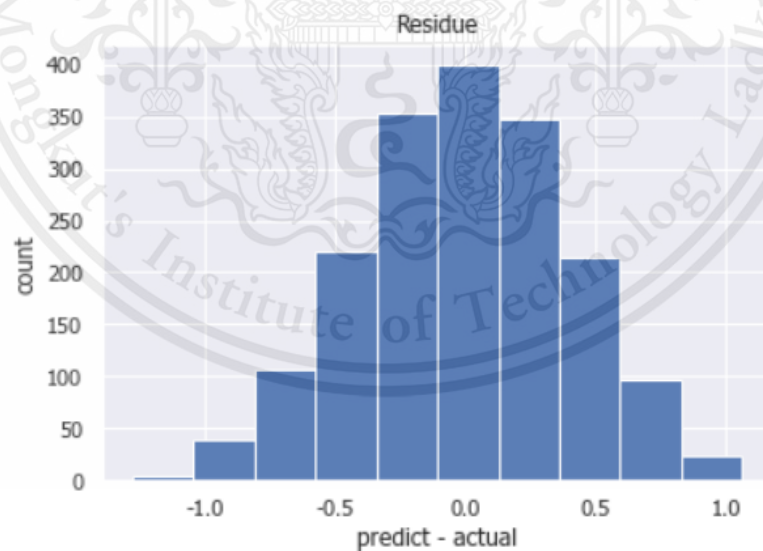


Figure 6.33: Residual

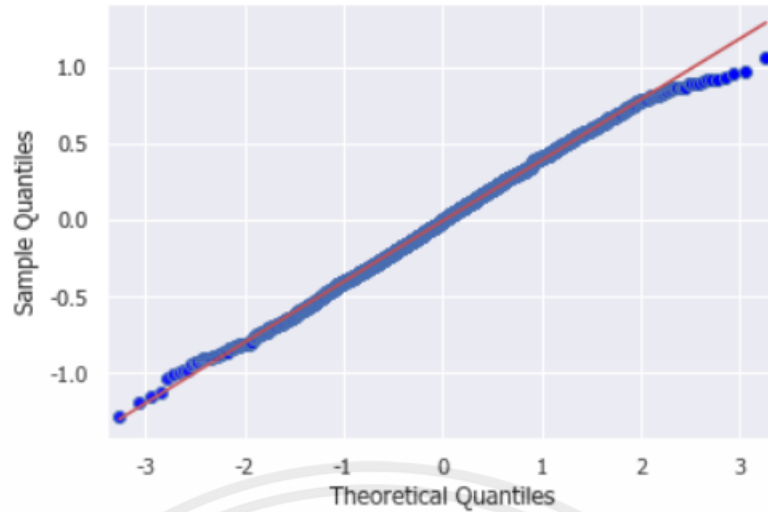
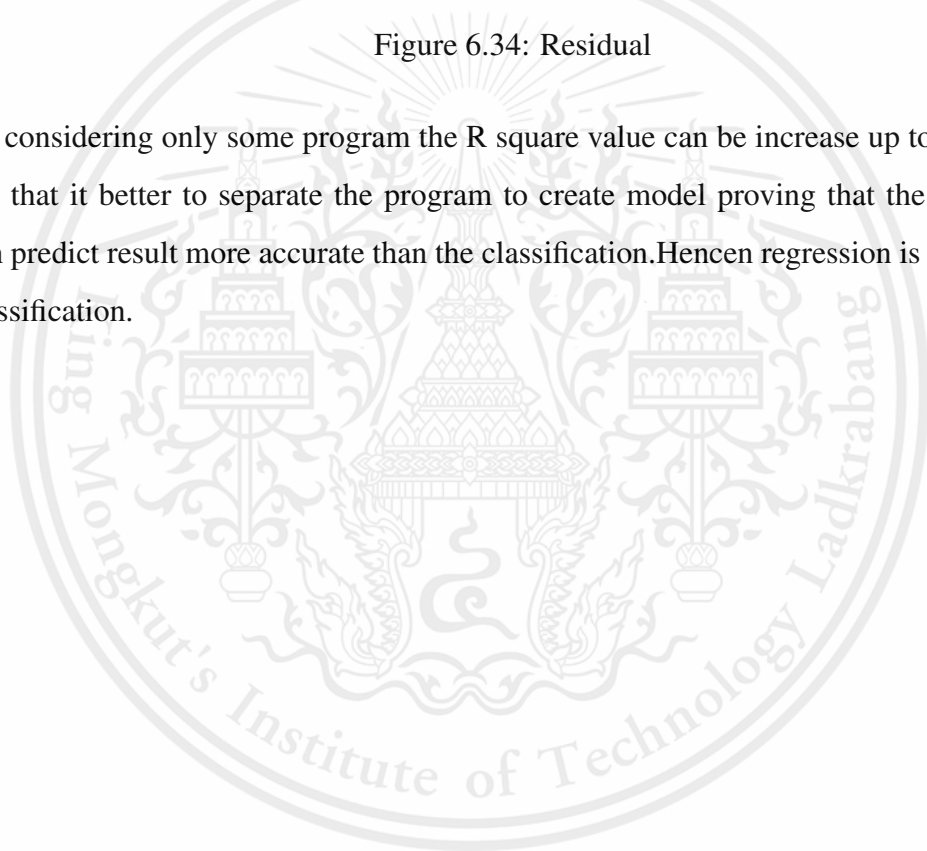


Figure 6.34: Residual

By considering only some program the R square value can be increase up to 0.4 showing that it better to separate the program to create model proving that the regression can predict result more accurate than the classification. Hencen regression is prefer than classification.



Chapter 7

Conclusion

7.1 What we have done

We have created a web based recommendation system that user can register and sign-in to find their matched programs based on their personality traits and education background.

For the server we have created a Django server which can handle multiple user requests at the same time. The continuous integration has also been implemented by using the feature of Heroku that can connect with GitHub repository to allow us to quickly deploy and run the server easily. For database and media management service, we have learned how to implement the outsource service into our system and learn how to properly handle and develop using new tools.

Lastly, we also learn how to work as a team under pressure of the time and resources limitation.

7.2 Problems that we faced

The problem that we faced is how to use the new tools to implement this system. Some of the tools are very new to us, so we have to spend lots of time to learn on how to implement it wisely. Some of the problems that we faced also consumed more times than what it actually need to solve it, but as a team we could help one another to solve

the problem.

7.3 Possible improvements

For possible improvements, more study programs could be used to recommend to the users.

The educational background and Big Five personality label of the past users could be use to increase the accuracy of our recommendation system. Data of KMILT current student are also being collect for varies study program recommendation.

For the user that's going to be studying in KMITL, we could keep track and record the students performance while studying through the course to create a new model and then compare it with our existing models to ensure our system accuracy.

The back-end server could be improved by reducing the steps needed to query the data from the database. Perhaps the design of the database could be improved by splitting a large table into smaller tables which will then improved the time of querying the data. The API's endpoints can also be improve by grouping some similar endpoints together which will reduce the confusion during development phase.

Bibliography

- [1] Stuart Russell, Peter Norvig, Artificial Intelligence — A Modern Approach, third. , Prentice Hall, 2013.
- [2] Maria Goga, Shade Kuyoro, Nicolae Goga, A recommender for improving the student academic performance Issues”, 7th - 9th November 2014
- [3] Recommender System for Predicting Student Performance, Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, Lars Schmidt-Thieme, 2010
- [4] Developing an Intelligent Recommendation System for a private university in Thailand, Kanokwan Kongsakun, Chun Che Fung, Tuchtawan chanakul, 2010