A Recommender System to help Grade 12 Students in the Kurdistan Region of Iraq in Applying for an Undergraduate Study

By

Mustafa Majeed Mohammed

Software Engineer

Feb 2020

Erbil, Kurdistan

Abstract

Every year thousands of high school graduates apply for admission at universities in the Kurdistan Region of Iraq. A large number of these applicants do not have enough information about the programs and how to select them. Therefore, some admitted applicants would find themselves in the program programs that do not interest them. This project studies the usability of a recommender system in assisting the applicants who apply to the universities to choose the programs that suit them the most. We argue that the evaluation of the current university students and alumni about the fitness of their studying program to their interests and abilities along with some particular questions about their educational background and desire help in the development of a recommender system that can assist the current applicants. We collect this information using an online questionnaire. We develop a program to asks a series of questions from the applicants, and we use the collected data to recommend to the applicant the most suitable programs according to the collected data and tested its user interface. We used 60% of the collected data as the training set and 40% as the testing set. The evaluation of the system shows it is 60% accurate in its recommendations.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

A lot of students in the Kurdistan region apply for universities every year. Many of these applicants do not know which universities are suitable for them to apply or which programs to choose. The majority of these applicants do not have enough information about university programs. The relation of these programs to the expected careers is also not clear to the applicants.

Therefore, finding the right university program is not easy, because there is a lot of parameters such as how much the salary will be after graduation, how good the reputation of the program is in society, the difficulty of the program. However, most of the students in the Kurdistan region will choose based on some parameters such as family, friends, community. For example, a student will look to his/her family members, if most of them were doctors for instance then the student most likely will go to medicine.

However, Income is a major part for a student to choose a program because most of the students will think about the salary after graduation even if they didn't like the program; they will sacrifice their passion and love of a program for the better income. Therefore, choosing the right program is a big issue in a region such as Kurdistan.

In this project we design and develop a recommender system to help the university applicants to recommend the most suitable programs to the university applicants. Ricci, Rokach, and Shapira (2010) Stated that “Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user”. Recommender systems are used in many areas such as e-commerce and e-learning. The system will use collected data from current students in the university, and the alumni students based on some designed and specific questions. Krosnick (2007) states that many of us know the best practice about questioner design, but also many of future researches is needed. A set of questions will be provided for current and alumni students to answer and based on that the data will be collected. The system will work on that big amount of data, and it will classify the data through some methods such as regression, classification and Nearest Neighbours.

The methods that are mentioned are part of machine learning that will be used by the system. Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed (Machine learning, 2017). The system will provide questions to the new applicant. The system will use supervised machine learning to classify the data. Kotsiantis (2007) stated, "The goal of supervised machine learning is to build a concise model of the distribution of class labels in terms of predictor features”. The system is expected to be accurate at least more than 50 or 60 percent so that the applicant will end up going to the program with a big rate of success of the recommendation of the system. A recommender system with a Machine learning concept will help many of students to choose the right program, and save them income, stress, and a lot of thinking.

1.1 Problem Statement

Most students when they finish high school they don’t know where to go for university, and most of them are confused. They don't have any background in any program, so they go based on people's opinions around them, or based on the salary of that program. After graduation or during the time of the study the student will realize that the program is wrong, but it is too late. Students also decide based on other members of their families. For example, if many members of the student's family are engineers then the student will mostly decide to be an engineer. I went to petroleum engineering based on my father's wish, but later I realized that it is not my area. However, later I changed my major to computer engineering, and after 2 years I found myself in software engineering. This indicates that most of the students in the Kurdistan region don't go to the right program at university.

1.2 Objectives

- To design a model that can recommend the most suitable program for university applicants.
- To develop a software system to implement the mentioned model.

1.3 Report Structure

- **Chapter 2_ Literature Review.**
  - This chapter will show the evidence and articles that support this research
- **Chapter 3_ Methodology.**
  - This chapter will show the methods that are used to make the recommender system and it will provide all the necessary information for building a recommender system.
- **Chapter 4_ Experiment.**
  - This chapter will show the step by step procedure to build a recommender system.
- **Chapter 5_ Results.**
  - This chapter will show the testing of the system and different results based on different inputs.
- **Chapter 6_ Conclusion and Future work.**
  - This chapter will show the conclusion and future work to be done for the system to develop it more.

Chapter 2 Literature Review

Technology has evolved in a very high rate and with it, the need of information become higher. Students in the Kurdistan region of Iraq apply to universities every year, and many of them choose their programs based on wrong concepts. Therefore, a recommender system will be helpful for them to recommend the best suitable program. This chapter will show the previous studies that have been done in this area, and it will show the methods that have been used for a recommender system, and a brief explanation about several studies that is related to the area of recommender system and machine learning.

Serval research explained the work of the recommender system. Lucas Drummond et al. (2010) explained that the recommender system as a big tool to help people make choices, and it is widely used in a lot of things such as e-learning. Recommender systems work to reduce unnecessary information and filter it so that the user can decide better. Lucas Drummond et al. (2010) used collaborative filtering to build the recommender system. Collaborative filtering works on the assumption that similar users like similar things, and it focuses on the past results to predict what happens next in order to increase the accuracy of prediction. Recommender system technique has been compared with traditional regression models, and educational data has been used to predict student performance.

Jie Lu (2004) explained the recommender system in e-learning for students to find courses based on their need. Jie Lu (2004) explains how the need of student to find the right courses is increasing. Therefore, e-learning changed the traditional learning methods. This research used a framework to recommend learning material to different student with different needs. The framework has four main components generating recommendation, getting student information, identifying student requirements, and learning material matching analysis. There are two main methods in recommender systems which is content based and collaborative filtering. The framework used in this research uses integration of both of them. The framework has some characteristic in helping student to choose the best suitable courses.

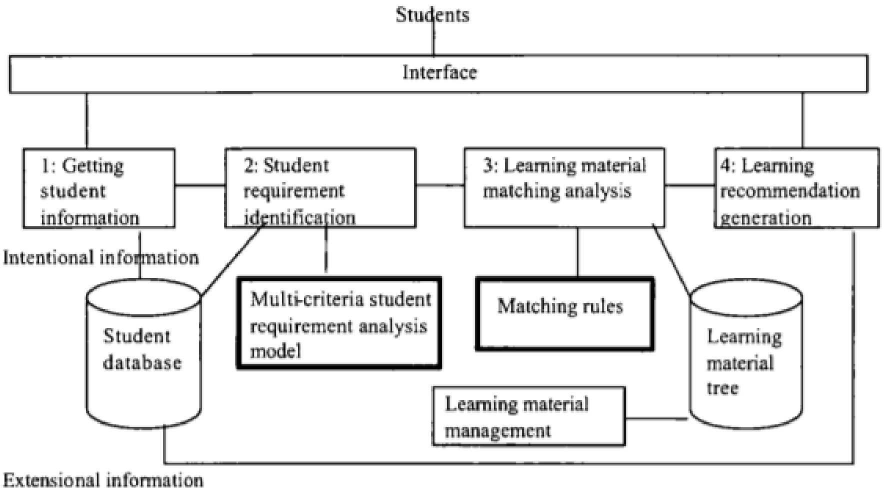

Students
Interface
1: Getting student information
2: Student requirement identification
3: Learning material matching analysis
4: Learning recommendation generation
Intentional information
Student database
Multi-criteria student requirement analysis model
Matching rules
Learning material tree
Learning material management
Extensional information

Figure-1: Framework diagram used by Jie Lu (2004).

Maria Goga et al. (2014) explains the use of a recommender system to improve academic student performance. Maria Goga et al. (2014) stated that student in academic facilities need to improve their academic skills and performance. Therefore, a recommender system is used to predict ways to improve academic performance. The research used large amount of data and store it in a large database. The large dataset included the academic policies. The research used major information to build the recommender system. They took information from university applicants when they enrolled to the university. The information included family background factors such as parent education status, gender, parent material status, and religion. Based on collected information a model has been built, and in-depth interview has been made to complete the recommender system to recommend the best way to improve student academic performance.

Zafar Iqbal et al. (2017) explains the use of machine learning based student grade prediction. The case study explains how some students in universities struggle to pass courses. Therefore, they need more attention in order for them to finish the required courses. This research used collaborative filtering and matrix factorization. Machine learning is used in the section of collaborative filtering which helps the recommender system to learn from previous datasets and predict the new coming dataset with higher accuracy. K nearest neighbor is used to find the most similar student and predict the grade. The research uses RBM machine learning technique to predict student performance in the specific course.

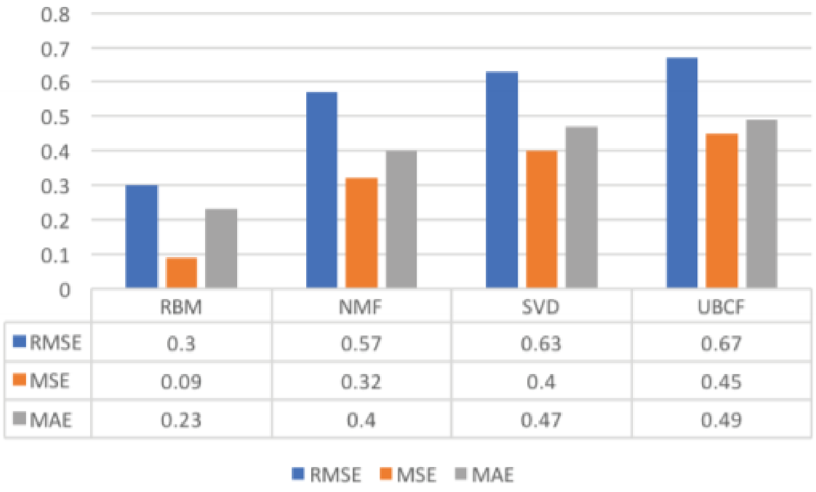


Figure-2: Grade prediction model. Zafar Iqbal et al. (2017).

Wenyi Huang et al. (2013) explains the use of supervised machine learning in a recommender system for context -aware citation. Wenyi Huang et al. (2013) stated that many authors search a lot amount of time to find the right book, or article. Sometimes many of them do not find what they seek to. Therefore, a recommender system will help them to easily find what they want. The research used supervised machine learning methods which include pre filtering, memory space saving, and rotation forest model. The model and the methods used to build the recommender system helped the authors to find the necessary articles, and the unnecessary information has been filtered.

Jason Smith et al. (2019) explains the use of a recommender system in sound libraries foe EarSketch browser. EarSketch browser helps user to find the needed sounds and music. Jason Smith et al. (2019) stated that the EarSketch browser has filtering mechanism, but it doesn’t have any mechanism to recommend the right music for the right user. Content-based filtering is used to do the task by comparing the result of what the user generates or like in the music libraries, and then recommend the similar music to the user. The research studied the users to analysis the information and produces the best recommendation. The recommender system will then be improved to reach high accuracy of recommendation.

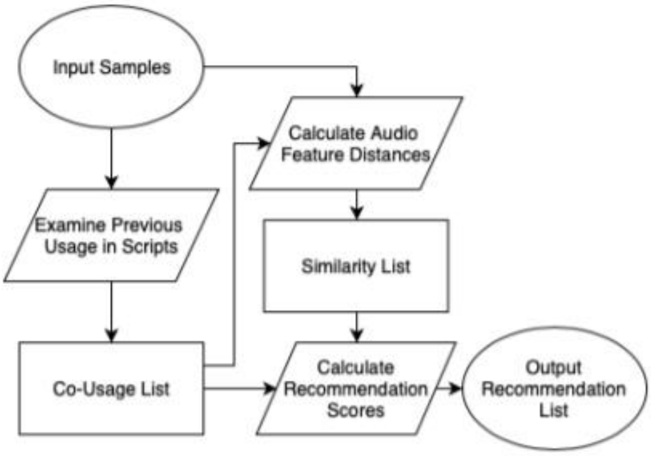

Input Samples
Calculate Audio Feature Distances
Examine Previous Usage in Scripts
Similarity List
Co-Usage List
Calculate Recommendation Scores
Output Recommendation List

Figure-3: Program flow of the recommendation system model used by Jason Smith et al. (2019).

Hana Bydžovská (2016) explains how to use recommender system for course enrolment. The recommender system will mainly use data mining techniques, and it works on the student's knowledge, interest, and skill. The recommender system will look at the time table of the student and based on the free time the course was recommended. Pre courses were designed for students to enroll, and then the information was collected. The collected information was helpful for the system to determine which the best recommendation for student course enrolment is. The result of this research was successful, and helped many students to enrol in the right and suitable courses.

Dileep Chaudhary et al. (2019) explains the use of machine learning to predict future for students. Now days there are a lot of options for students. Therefore, confusion happens for students who want to choose a career. A recommender system will help them to choose the right career, and it will filter, or remove unnecessary information and paths. This research created a web application for students who want help in choosing their career. A set of question has been asked and based on those questions the system will determine the best career for the student. Many methods are used such as correlation matrix, regression, and random forest algorithm. The methods help the system to have high accuracy for predicting the right career for students.

2.1 Summary

The research mentioned above are all working on the concept of machine learning and machine learning algorithms and methods. Many of them used collaborative filtering to achieve a valid result. Others used content-based, but they all share the same concept of recommender system. In our approach we use supervised machine learning which works on labelling the data that has been collected. The supervised machine learning has some methods. In our approach we will use classification method to classify the data into categories. More will be explained in the methodology chapter.

Chapter 3 Methodology

This chapter will show all the steps and methods necessary to build the recommender system. The methods will be clarified and explained in detail. The chapter will show all the steps needed to gather information and use it to build the whole system. Figures and tables will be provided to demonstrate all the steps for the recommender system.

Data collection section will show all the steps to collect the data in a proper way. A questioner is designed to be provided to the university of Kurdistan Hawler students and the alumni as well. After that a pilot test is done to show a part of the data that will be used to be fed to the system. The pilot test will help to collect data, and then the collected data will be analysed. Relationship between data will be found and used to create the algorithm that will then create the model.

Model section will provide the methods that are used to create the model, and how the model will work. The expected results will be measured for the model. the algorithm will be explained for creating the model and the functionality of the model will be clarified. Figures will be provided to better understand the concepts.

Output of the system section will show the expected output from the recommender system.

System interface design section will show all the steps needed to create the user interface including use case and entity relationship diagram. All the steps and explanation will be provided for both diagrams, and the sections and relationship between entities are explained for the entity relationship diagram.

3.1 Model

After the data collection creating the model will begin. Supervised machine learning will be used in our approach. Supervised machine learning is when you have an input and an output. An algorithm will be used to learn the mapping from input to output such as y=f(x) where x is the input and y is the output. The goal of supervised machine learning is to get the mapping function right enough as much as possible so that if a new input gets into the system the prediction of the input will be right enough, or accurate.

3.1.1 Method

Supervised machine learning has methods and algorithms. In our approach we will use classification method to label and categories the data that has been collected. Classification method is when you have a set of data and you classify them based on their category.

For example, an item could be drink or sweets if the item is kunafa then it is sweets, and if the item is coffee then it is drink. In our data set we mainly focus on the level of satisfaction and the relation between behaviour and satisfaction. The data will be categorized into satisfied and not satisfied. For example, a student may have high score in math and physic, but he may not be interested in engineering, or a student may have a high score in chemistry, but he may not be interested in any field of chemistry. The system will identify the data as satisfied or not satisfied. When new data is entered the system will know what category the data belongs to as accurate as possible. The figure-4 below will show a sample of the classification method.

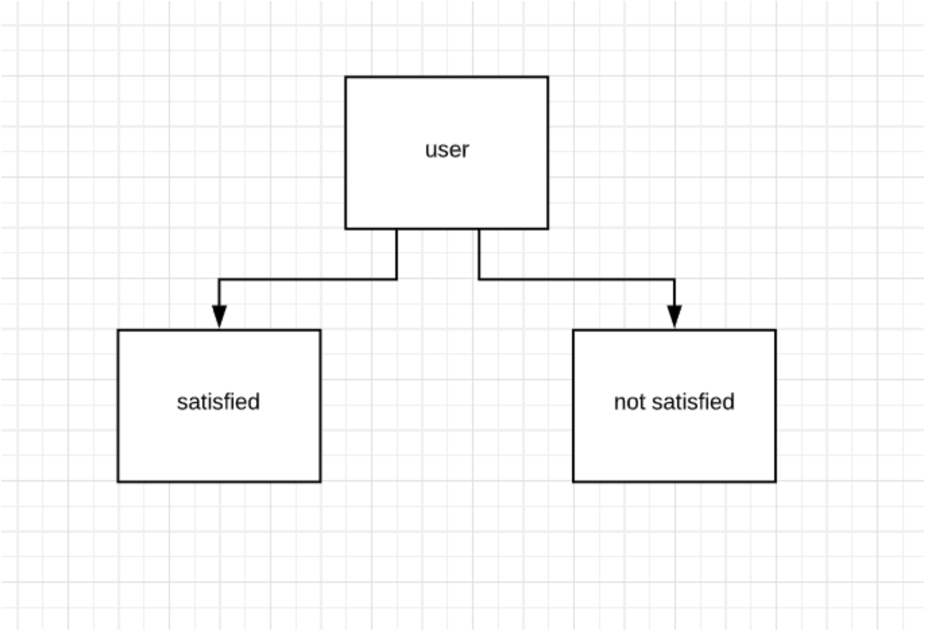


Figure-4: Sample of classification method.

3.1.2 Feature Extraction and Data Labelling

An algorithm is created to run the model and suggest, or predict the best suitable program for the new applicants. The algorithm consists of getting the data that has been collected, then the collected data will be labelled and a weight will be given to it. The last part of the algorithm is labelling data and giving weight to the data. Each question will have it is own weight.

For example, math, physic, chemistry will have 45% weight on them. The behaviour questions will have 40% weight on them. The happy or not happy questions will have 15% weight on them. The relationship between all the questions specifically behaviour and subject scores will be used in a classification method to determine that if the student or alumni are satisfied or not satisfied. This 2 stated will be used along with the collected data, or the weight of subject scores to predict, or suggest the best suitable program for the new applicant.

3.1.3 Training Data Set

The output of the system should be accurate more than 70%. To calculate the accuracy of the output in our approach we will use comparison method. A set of data will be entered to the system. The data are well labelled and classified into satisfied and not satisfied. Based on the classified data the accuracy of the system can be calculated.

For example, when a new data enters the system the system should be able to predict the answer with accuracy more than 70%. If not then we revise the data to see where is the missing link, or were did we miss a point. The data will be entered to the system again, and a new data will enter the system. the system now should predict the answer with higher accuracy. This procedure will continue until the desired accuracy level is obtained.

Therefore, the most important part in the output of the recommender system is accuracy, because the goal of recommender system is to obtain the right answer by 100% which is ideal and doesn’t exist until now. In our system we will try to obtain the highest level of accuracy to ensure that the recommender system predict the answer right enough.

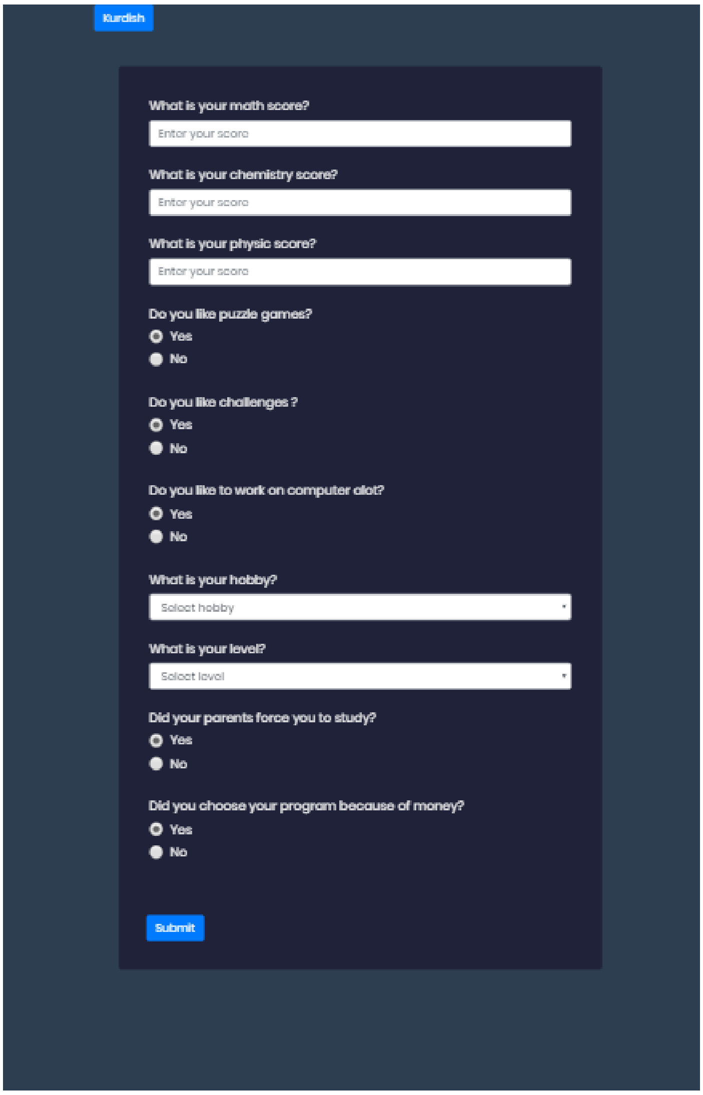
3.2 Data Collection

Data collection is an important part of any research. Therefore, there are many methods that are used to collect data such as questioner, interview, survey. Each research uses specific type of this method to collect data. In our approach we use questioner method to collect data by providing some specific questions to students in universities, and they will answer the questions. The alumni will also have the same set of questions to answer. Part of the questioner will include questions about math, chemistry, physic. This part is to collect data about the scientific score of the student. This part will be used to determine the relation between the scientific programs with this score for the students. Second part of the questioner is questions about the behaviour of the students such as what type of games they like, what kind of activity they like most, what hobbies they have.

Annu Rev Sociol (2018) stated that researchers in the interdisciplinary field of Judgment and Decision Making worked on the behaviour of human in felids of economic, politic, science to find the best solutions of many problems through behaviour research. The last part of the questioner is the level of satisfaction of the student about the program that they will choose, or in case of the alumni it will be the level of satisfaction of the program that they chose.

3.2.1 Pilot Test

Testing is a very important part of every project. Therefore, a pilot test is nearly a must to ensure that the data collection is right, and to see where it is wrong, or the missing link in the collected data. A pilot test has been done that included the 3 parts of questioner mentioned above. The pilot test was done by using HTML, CSS, JAVASCRIPT, PHP, MYSQL. The data has been taken through the test interface which is HTML and CSS. The interface is easy to understand and it is dynamic. For example, when the user choose option 1 based on option one another input will come up with several other options, and if the user chose option 2 then based on option 2 another input will show up. All the questions are multiple choices, because collecting and arranging data is very easier when your answers are fixed, and it is easier for the user to understand. The only part which the user will enter by him/her will be the math, physic, chemistry score inputs. The interface is shown in figure-5 below.



What is your math score?

Enter your score

What is your chemistry score?

Enter your score

What is your physic score?

Enter your score

Do you like puzzle games?

◉ Yes

○ No

Do you like challenges ?

◉ Yes

○ No

Do you like to work on computer alot?

◉ Yes

○ No

What is your hobby?

Select hobby ▾

What is your level?

Select level ▾

Did your parents force you to study?

◉ Yes

○ No

Did you choose your program because of money?

◉ Yes

○ No

Figure-5: The questioner interface design.

The collected data from the interface above has been processed through PHP and then passed to MYSQL database and has been stored there. An algorithm has been used to ensure that the data is not redundant which means that the same user can not fill the questioner more than one time. The algorithm takes IP address of each user that opens the interface and when the user is done filling the questioner the interface will close. If the user tries to enter the questioner again then the algorithm will compare the IP of that user with the stored IP in the database, if they match, then the interface will close itself. By this algorithm redundancy will be reduced nearly to zero.

3.2.2 Statistical Analysis for Pilot Test

After the data has been collected a statistical interface has been created to analyse the data. The interface includes the total number of users who has been filled the questioner, the total number of users who like puzzle games, the total users who like challenges, the average score of math, physic, chemistry of each user, etc. The statistic interface shows a lot of relations between the data. For example, most of the users who liked sport as their hobbies were interested in games, and challenge tasks, and most of the users who liked reading as their hobbies were interested more in challenge and puzzle than computer. The table below will show some of the relation between hobby and behaviour of the users. The total number of users who filled the questioner form is 136 users.

Table-1: The relation between hobby and behaviour.

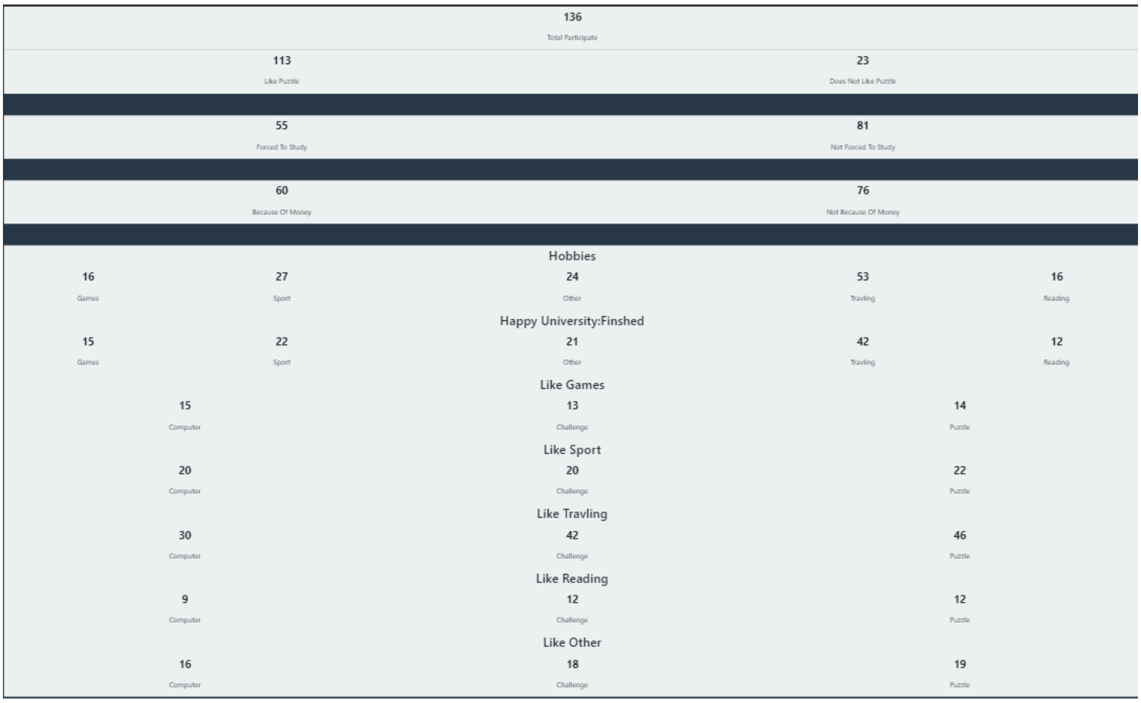
Hobbies	
Games	16
Sport	27
Traveling	53
Reading	16

The statistic part showed another important aspect which is the relation between the average score of the three subjects and the behaviour of the users. For example, most of the user whose score in math were above 70 they were interested in computer and challenge. The users whose scores are more than 70 in chemistry and physic were interested in puzzle. The table below will show the average score of the three subjects for the total 136 users.

Table-2: The relation between subject scores and behaviour.

Average score		
Math	Physic	Chemistry
75	75	78

The pilot test showed very important aspects of the student’s behaviour, score, and satisfaction. The test showed the relation between student behaviour and hobbies of the students such as the stuff that they like to do, the programs that they want to attend. Overall it was a helpful process to understand the behaviour of the students, and the data needed to build the recommender system. the figure-6 below will show the interface of the statistic page.



136
Total Participate

113
Like Puzzle

23
Does Not Like Puzzle

55
Forced To Study

81
Not Forced To Study

60
Because Of Money

76
Not Because Of Money

Hobbies

16	27	24	53	16
Games	Sport	Other	Travling	Reading

Happy University:Finshed

15	22	21	42	12
Games	Sport	Other	Travling	Reading

Like Games

15	13	14
Computer	Challenge	Puzzle

Like Sport

20	20	22
Computer	Challenge	Puzzle

Like Travling

30	42	46
Computer	Challenge	Puzzle

Like Reading

9	12	12
Computer	Challenge	Puzzle

Like Other

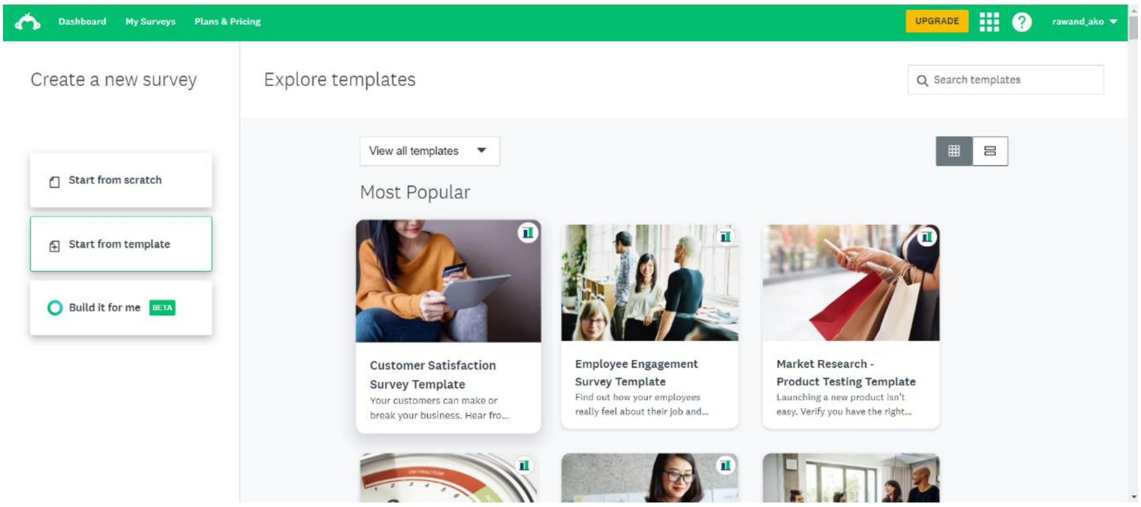
16	18	19
Computer	Challenge	Puzzle

Figure-6: The interface of the statistic page.

3.2.3 Survey Monkey

Survey monkey website is nearly the best tool to collect data, because of its features. Survey monkey provides best templates that best suit your survey need. It also provides all the possible statistics for your survey after people fill it. The interface is easy to use, and easy to understand. The most important part it uses a special method that makes the data none redundant, it means no user can fill the survey twice. The identity of the user remains hidden which gives the user confidence to fill the survey without fear.

Pilot test has been done using a website and a self-created algorithm, but for the real data collection we will use survey monkey. The self-created website, and algorithm had the future of none redundant data but survey monkey is trusted, tasted, and more popular, because of that survey monkey will be used. The figure-7 below will show the interface of survey monkey.



Customer Satisfaction Survey Template
Your customers can make or break your business. Hear fro...

Employee Engagement Survey Template
Find out how your employees really feel about their job and...

Market Research - Product Testing Template
Launching a new product isn't easy. Verify you have the right...

Figure-7: Survey monkey interface.

3.2.4 Evaluation

Accuracy of the recommender system is important. The accuracy of the recommender system can be measured using regression error evaluation metrics. Common error evolution metrics are mean absolute error (MAE), mean square value (MSE). For example, if the recommender system was working for rating items, then we would use these equations.

The equation of (MAE) is:

MAE= $\frac{\sum rating|P-R|}{rating}$ equation (3.1)

Where (R) is true rating and (P) is predicted rating.

The equation of (MSE) is:

MSE= $\frac{\sum rating(P-R)^2}{rating}$ equation (3.2)

We are measuring the average squared divergence from the predicted rating (P) and the true rating (R).

We can measure the accuracy of the system by these equations. In future we can improve the accuracy even more.

3.3 Software Component

Every software system needs an understandable interface design in order to make the system easy to use. Many interface designs are hard to understand and hard to use. They fail to deliver the best user experience which is an important part of the system interface design.

User experience is when the user is satisfied with the interface and he/she understand all the element of the interface clearly. In our approach we try to deliver the best user experience in order to help the user to understand the interface better and easier.

The system interface design will include use case to clarify who will interact with the system, Entity Relationship Diagram (ERD) to understand how the data is stored in the database, the programming languages that are used to make the system interface design. All this element will work together to deliver the best user experience and an understandable easy to use interface design.

3.3.1 Use Case Model

Use case diagram is a tool to show the people who interact with the system. in use case diagram they called actors. A use case diagram will show the boundary of the system, and the rules of the actors interacting with the system. The boundary of the use cases is called the boundary of the system. in many cases a sub system which is a system by itself can be an actor in use case diagram. The figure-8 below will show the use case diagram of the recommender software interface design.

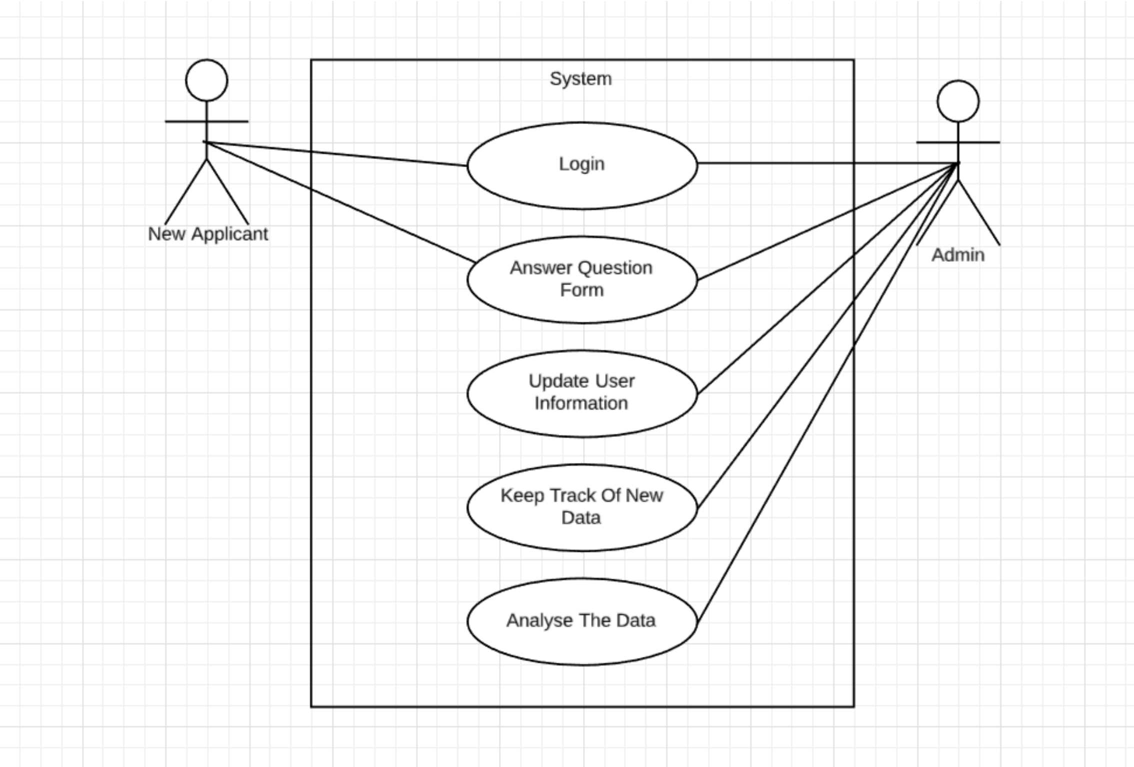


Figure-8: Use case diagram of the recommender system software.

The lines drawn from both actor to the use cases describe the rules of each stick man, or each actor.

- The new applicant can login and then fill the question form to insert the data to the system and do nothing more.

- The admin will have more ability in the system. the admin can login, access the question form interface, update login information of the student, keep track of the inserted data, update form questions and options if necessary, and monitor the statistic that has been calculated based on the inserted data.

- There will be no subsystems as the recommender system is not huge and require simple interface design to get the data from the new applicant and process it.

- Statistical management will show all measurement and data analysis to see the system better and understand it better.

3.3.2 Database Design

Entity Relationship Diagram (ERD) is the design boxes that are used before implementing any database. Entity Relationship Diagram shows the tables and the relation between those tables to produce a complete database design.

However, there are six types of relationship between the tables of a database such as one to many, one to one, many to many, one and only one, zero or many, one or many. The figure-9 below will show the types and how it looks like.

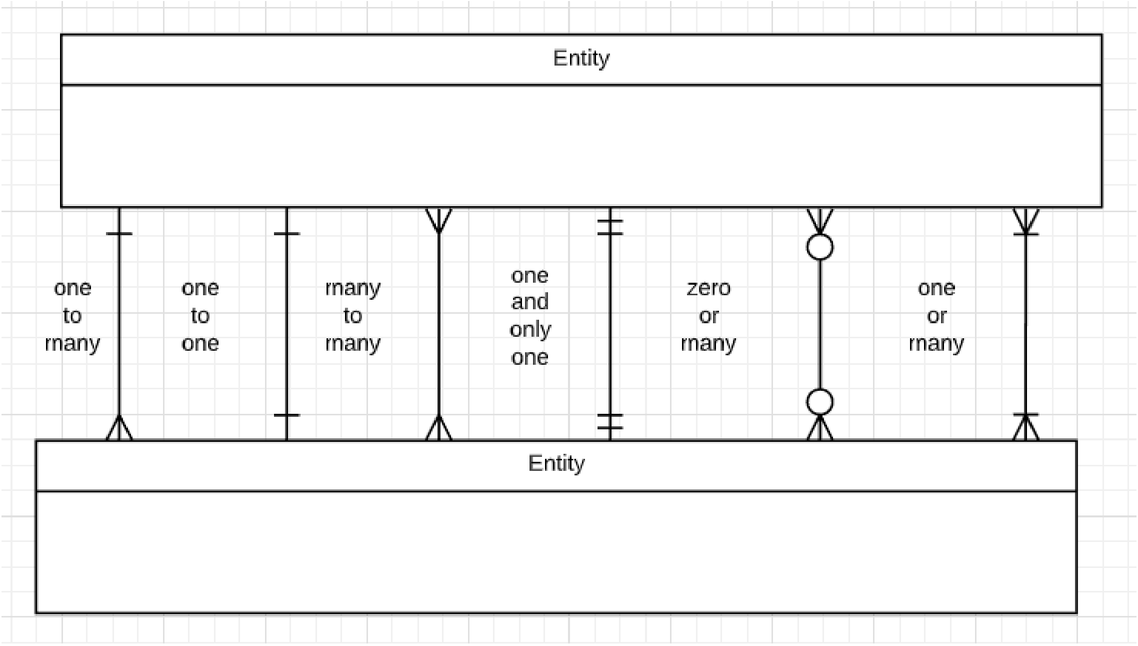


Figure-9: Entity Relationship Diagram notations.

Each one can be used to describe the relation of 2 tables in the database. There are other types as well but those six types are the most common one. There is a rule of thumb in database design with ERD, and the rule is that always escape many to many relationships between tables.

Therefore, the escape can be done by producing the third table between the two tables and putting the primary keys of both tables as foreign key in the table between them. The ERD must be clear and show all the aspect of the database along with the proper relationship between those tables, or entities. The figure-10 below will show the ERD of the recommender system to store the data inserted by new applicants.

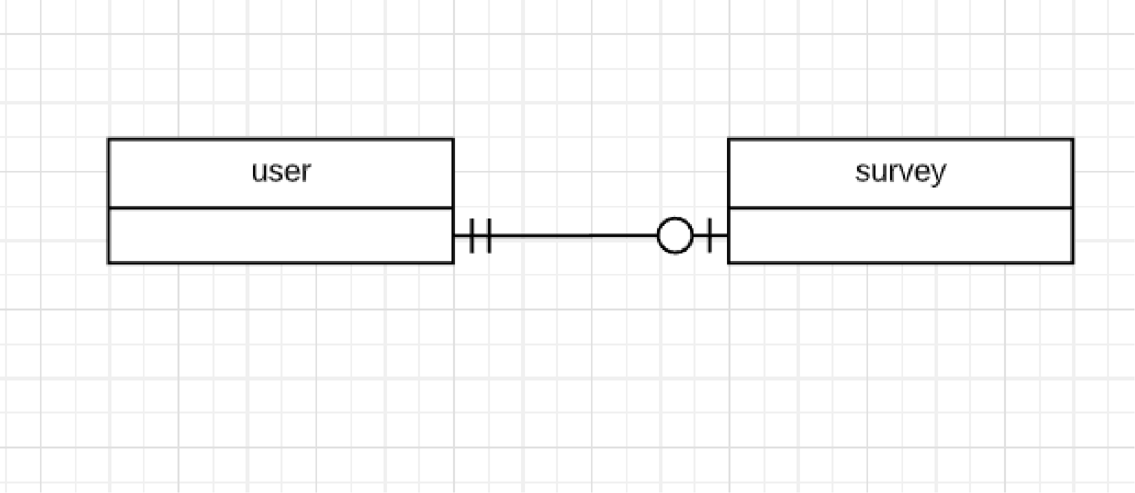


Figure-10: Entity Relationship Diagram (ERD) of the recommender system.

Figure-10 shows the elements to make the database design and store data in a way that redundancy is reduced nearly to zero.

- The table user is to store the information of the new students who wan to apply for the survey.

- Table survey is to store the information of survey questions and then using MYSQL queries the statistic data will be generated.

- The relationship between them will be one and only one to one or zero because we only allow one form submission for each user. In other words, it means that a survey must be filled with one and only one user but a user can exist without applying the survey.

3.3.3 Development Environment

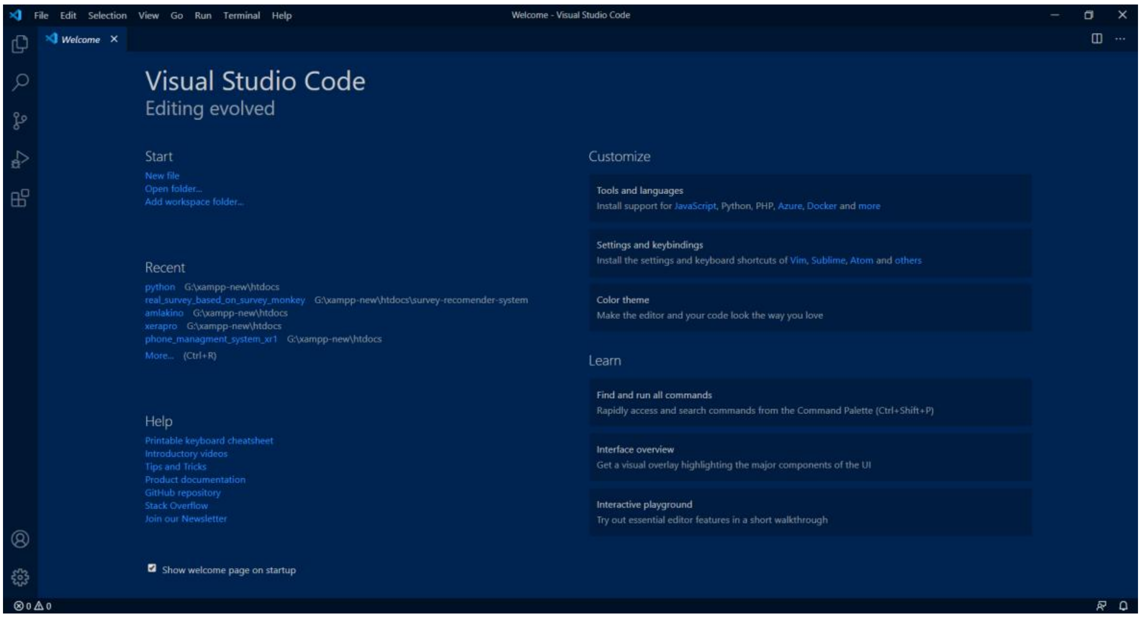
All softwares are built with a programming language or multiple programming languages. In our approach we will use multiple programming languages to build the recommender system such as Python, JavaScript. Many languages will be used with these two programming languages to help the interface, and make it easy to understand and deliver best user experience. Languages such as HTML, CSS, SQL will be used to manipulate data and search in the database to retrieve the required information, and also it is used to insert information into the database. Updating and deleting are other functions that SQL can do. The data base we use called MYSQL database and it is a relational database. Python will be used for the machine learning functions and methods such as classification method. HTML will be used to create the interface in other word the skeleton of the interface. CSS will be used to style the interface and give it a nice look that the user would like to look at it. JavaScript will give it a dynamic look and functions it will be used along with python. Python and SQL will work together to capture the information and insert it to the database using insert query. Python will do the calculation and use the function to do all the tasks for the recommender system, and then it will connect to HTML to represent the interface and give data to the interface to show it. All the languages will work together to produce the system interface design in its best image.

Chapter 4 Experiment

This chapter will cover the aspects of experiment and implementation of the system**.** It will show how the algorithm is developed and how the system will predict the most accurate program for the student. It will show how data is processed as well as the survey based on survey monkey and the database that stored the data.

4.1 Environment

The recommender system will be developed by python programming language. Therefore, visual studio code will be used as the environment for the system. Visual studio code is a text editor from Microsoft, what makes visual studio code better than any other editor is the user experience and the huge amount of extensions that can be easily downloaded to help the developing process. According to (software.com) visual studio code was the most used editor in 2019, and the editor supports multiple languages including python. Figure-11 below shows the interface of visual studio code.



python G:\xampp-new\htdocs
real_survey_based_on_survey_monkey G:\xampp-new\htdocs\survey-recomender-system
amlakino G:\xampp-new\htdocs
xerapro G:\xampp-new\htdocs
phone_managment_system_xr1 G:\xampp-new\htdocs
More... (Ctrl+R)

☑ Show welcome page on startup

Tools and languages
Install support for JavaScript, Python, PHP, Azure, Docker and more

Settings and keybindings
Install the settings and keyboard shortcuts of Vim, Sublime, Atom and others

Color theme
Make the editor and your code look the way you love

Find and run all commands
Rapidly access and search commands from the Command Palette (Ctrl+Shift+P)

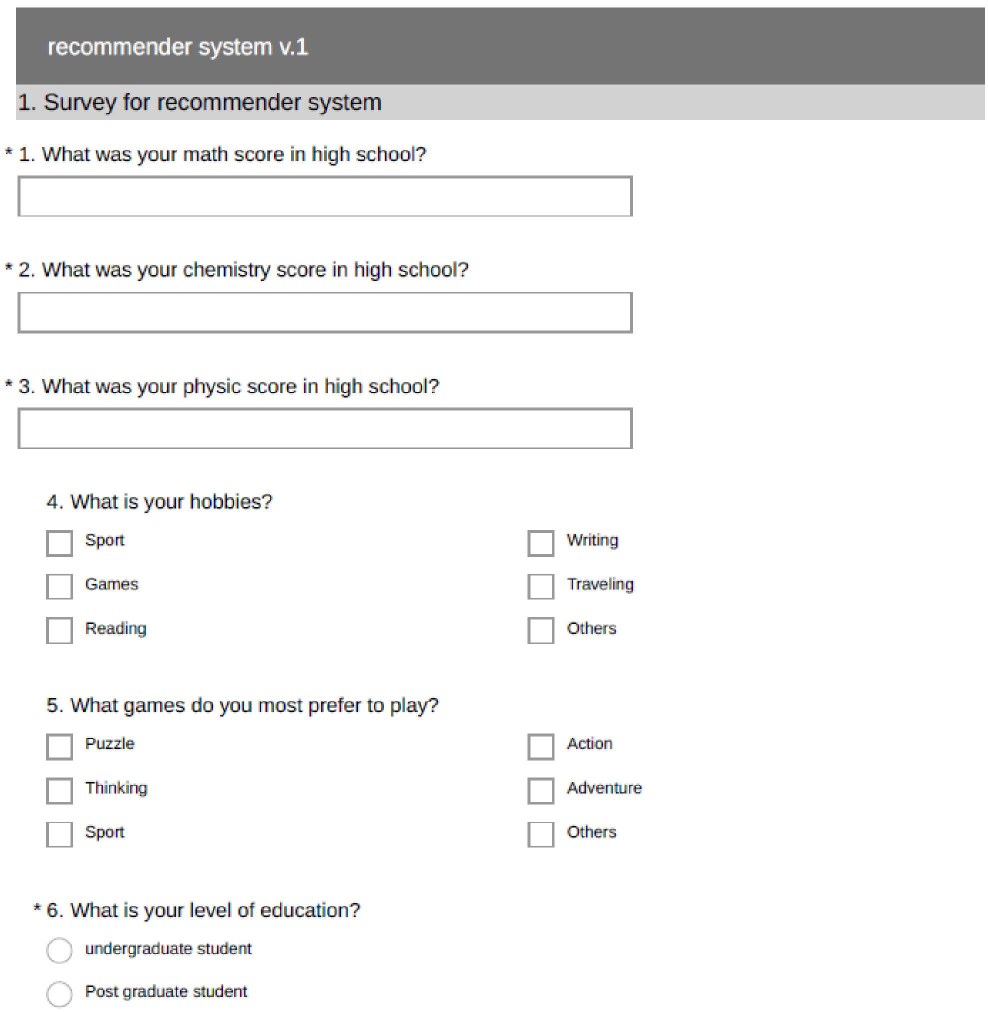
Interface overview
Get a visual overlay highlighting the major components of the UI

Interactive playground
Try out essential editor features in a short walkthrough

Figure-11: Visual studio code interface

4.2 Collected data

Survey monkey was supposed to be used to collect data, but because of some difficulty we could not collect data by survey monkey. However, the questions and logic were created by survey monkey and all was ready to implement. Survey monkey by default was not allowing any user to answer the survey twice, and validation was done om some inputs. Figure 12 below shows an example of the survey monkey questions.



recommender system v.1

1. Survey for recommender system

* 1. What was your math score in high school?

* 2. What was your chemistry score in high school?

* 3. What was your physic score in high school?

4. What is your hobbies?

□ Sport
□ Games
□ Reading
□ Writing
□ Traveling
□ Others

5. What games do you most prefer to play?

□ Puzzle
□ Thinking
□ Sport
□ Action
□ Adventure
□ Others

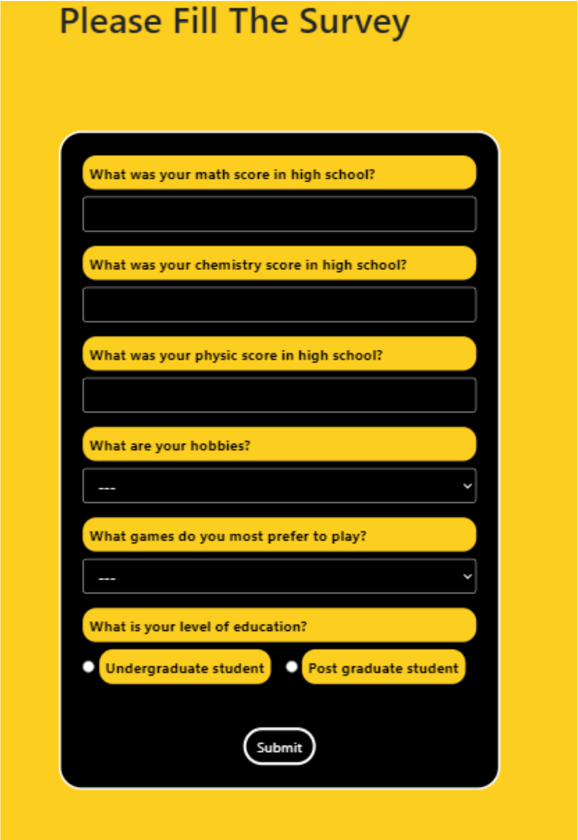
* 6. What is your level of education?

○ undergraduate student
○ Post graduate student

Figure-12:

Survey monkey questioner example.

Therefore, we created a website application to collect data as our plan B. The website application included full validation such as that the new student could not enter more than 100 or a negative number in mark question fields. There was no need to validate the selection questions because of that the new student would just select the answer, and for constancy we used IP address in order to get consistent data. When the new student clicks on submit the IP address will be stored in the database and the link will close on the browser. Figure 13 below will show the web interface.



Please Fill The Survey

What was your math score in high school?

What was your chemistry score in high school?

What was your physic score in high school?

What are your hobbies?

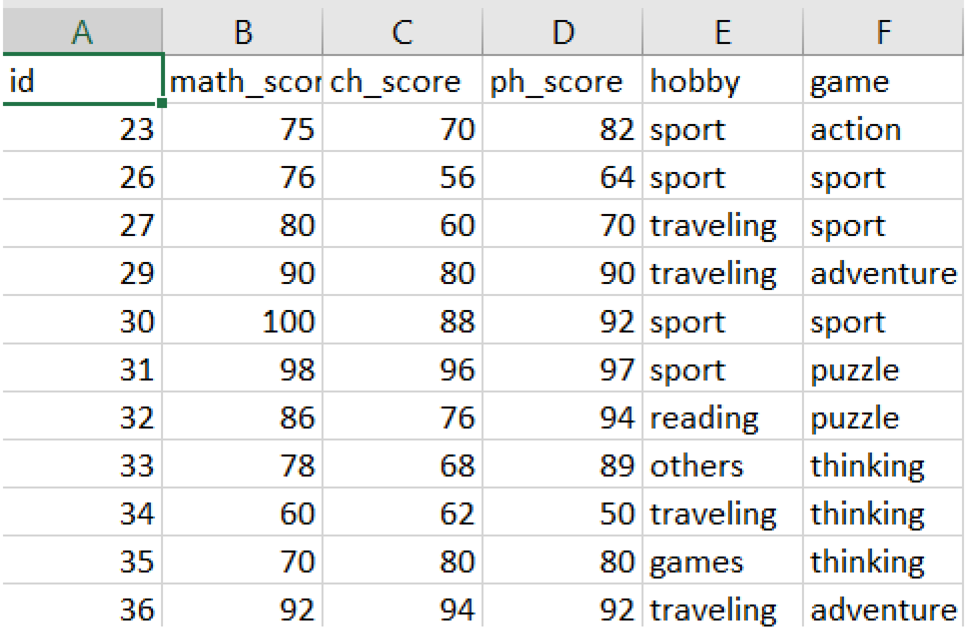
What games do you most prefer to play?

What is your level of education?

○ Undergraduate student ○ Post graduate student

Figure-13: Web application interface

However, the survey contained mixed questions including behaviour questions to understand the perspective of the participants and subject scores to evaluate the student performance. The stored data then was exported to a CSV file to be processed using pandas library available in python programming language. What makes pandas library used is the amount of built in function that makes processing data very easy and efficient. Figure 14 below shows the CSV file.



A	B	C	D	E	F
id	math_scor	ch_score	ph_score	hobby	game
23	75	70	82	sport	action
26	76	56	64	sport	sport
27	80	60	70	traveling	sport
29	90	80	90	traveling	adventure
30	100	88	92	sport	sport
31	98	96	97	sport	puzzle
32	86	76	94	reading	puzzle
33	78	68	89	others	thinking
34	60	62	50	traveling	thinking
35	70	80	80	games	thinking
36	92	94	92	traveling	adventure

Figure-14: CSV file of recorded data from survey

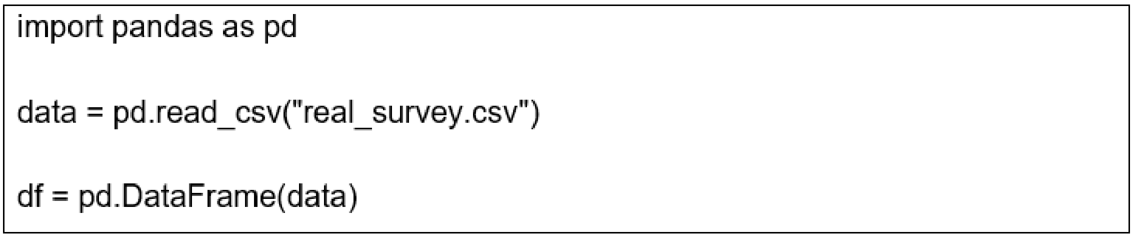
4.3 Algorithm

In chapter 3 we explained how the algorithm and the model should be developed, but because of some problems in collecting data as explained in 4.2 collected data. We could not develop the algorithm as planned. Therefore, we developed an algorithm based on the few data that we collected. The algorithm has some machine learning specification to show how machine learning is working.

However, the algorithm of the recommender system is built using python and it is based on the behaviour and average mark of the survey and then asking several questions to new students. The answers of the new student will be compared with collected data and then the algorithm will find the most repeated value and recommend it for the new student. For example, if the most chosen program is computer engineering for those who like games as their hobby then if the answer of the new student is game as their hobby the system will recommend computer engineering to the student. Which is done by taking into account the satisfaction of collected data from university of Kurdistan Hawler students and alumni. Below sub sections will explain the development of the algorithm and data processing in detail.

4.3.1 Reading the CSV file using python

Python programming language provide the complete use of pandas library. The first step is to read the collected data that is in CSV format in order to process it, bellow commands shown in figure 15 are used to import and read the CSV file.



```
import pandas as pd

data = pd.read_csv("real_survey.csv")

df = pd.DataFrame(data)
```

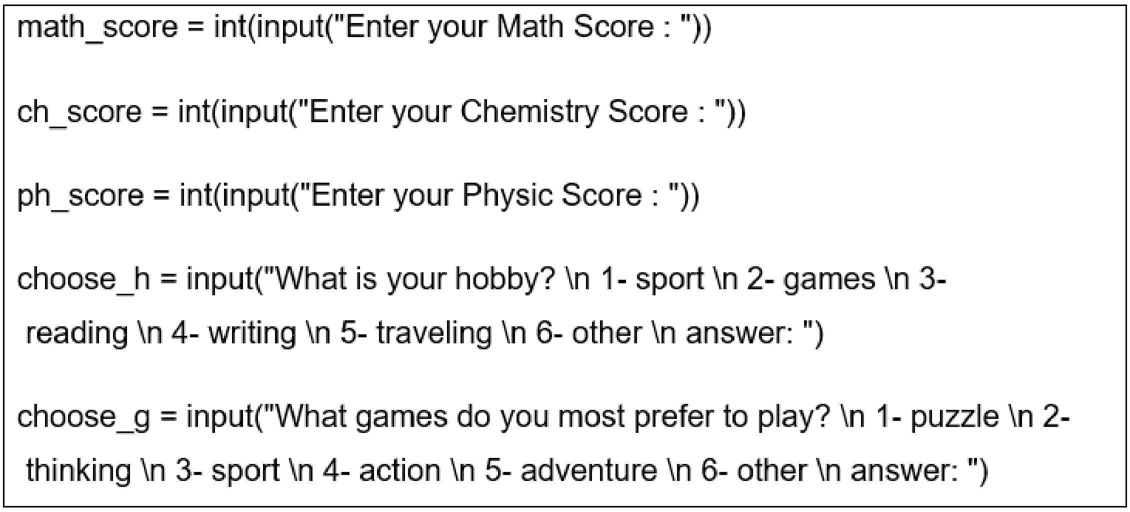
Figure-15: Reading CSV file.

The panda's library is imported by import command and it is imported as pd which is the variable name that we will use through the algorithm.

The (pd.read_csv) command used to read the CSV file and we assigned it to data variable. Finally the data frame is created by the (pd.DataFrame) command and assigned to df variable to be used in the algorithm.

4.3.2 Taking inputs

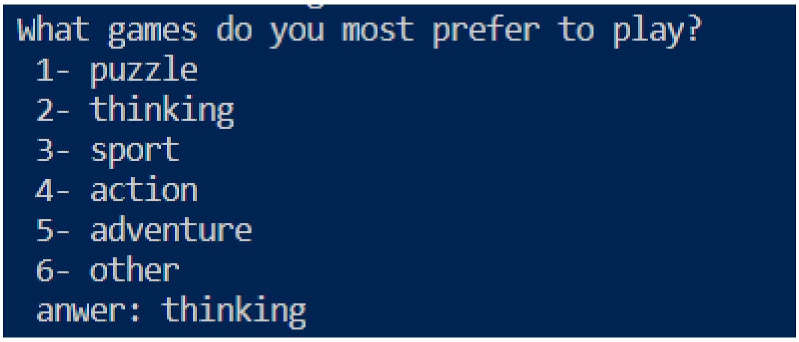
The next step in the algorithm is to take inputs from the students who want to use the system. In python programing language there is a function to take inputs from user and then process it based on the algorithm rules. Below is a sample code on how to take input from new student shown in figure 16.



```
math_score = int(input("Enter your Math Score : "))

ch_score = int(input("Enter your Chemistry Score : "))

ph_score = int(input("Enter your Physic Score : "))

choose_h = input("What is your hobby? \n 1- sport \n 2- games \n 3-
 reading \n 4- writing \n 5- traveling \n 6- other \n answer: ")

choose_g = input("What games do you most prefer to play? \n 1- puzzle \n 2-
 thinking \n 3- sport \n 4- action \n 5- adventure \n 6- other \n answer: ")
```

Figure-16: Taking input from new students.

The first line is the input function to take match score from new student who uses the system and it is inside int () function in order to convert the input into integer to be processed by the algorithm later. All the score inputs will be taken the same way. They are assigned to variables math_score, ch_score and ph_ score to indicate math score, chemistry score and physic score. Next the system will ask for two behavior questions by input function. The first one is hobby question in which it will ask what your hobby is and the student will write the hobby then the algorithm processes the data. The second question is what game the student likes to play. The same process will apply here the student must write the answer based on the given selections. Figure 15 shows the question in the console.

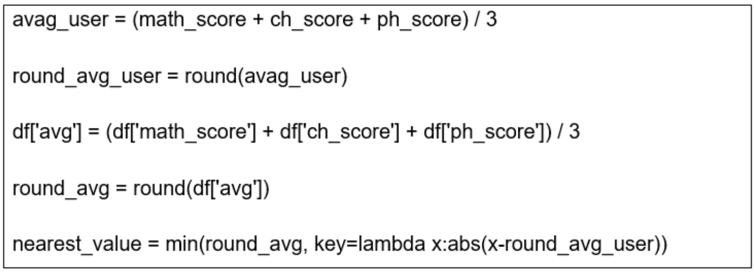


```
What games do you most prefer to play?
 1- puzzle
 2- thinking
 3- sport
 4- action
 5- adventure
 6- other
 anwer: thinking
```

Figure-17: game question in the console

4.3.3 Calculating mark questions

After taking the inputs the algorithm will start processing data through comparing the inputs with the collected data in the CSV file. Using panda's library, the average mark of the three subject from the collected data will be calculated and rounded to become an integer. Next step using panda's library the algorithm will take the three inputs of the student and calculate the average value. Then the algorithm will compare the average values in the collected data with the average value of input data and find the closest one. The last step is to look up in the CSV file to find what program falls under this value. Below is an example code of the first process.

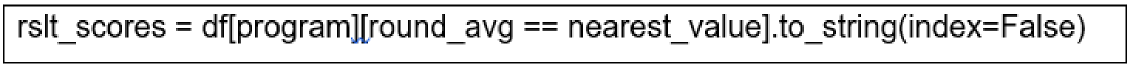


```
avag_user = (math_score + ch_score + ph_score) / 3

round_avg_user = round(avag_user)

df['avg'] = (df['math_score'] + df['ch_score'] + df['ph_score']) / 3

round_avg = round(df['avg'])

nearest_value = min(round_avg, key=lambda x:abs(x-round_avg_user))
```

Figure-18: First process of the algorithm

The first line is to calculate the average value from input data. The second line is to round the value using round function. The third line is to find the average values in the collected data. The fourth line is to round the average values to be integers and the last line is to find the nearest value in the average values to the average input data.

Using one command the algorithm will look up to the collected data and find what program falls into the range of calculated value. Shown below is the command used.



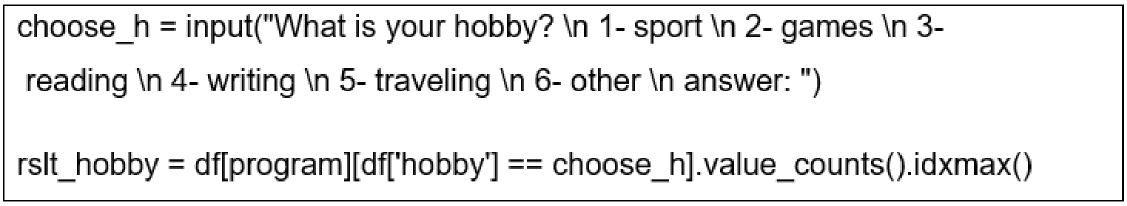
```
rslt_scores = df[program][round_avg == nearest_value].to_string(index=False)
```

Figure-19: Command used for nearest value

This line will choose the programs available and finds the range of value that falls into it.

4.3.4 Calculating behavior questions

Behavior questions are calculated in a different way. For example, marks and average value calculated using round function or range of value, but the behavior questions will count on frequency of the data. For example, if the student entered sport as the input data of hobby then the algorithm will look up of the collected data and find how many time survey participants selected sport as their hobby, then it calculates the frequency of that and shows the program. Below is the example code of calculating behavior questions.

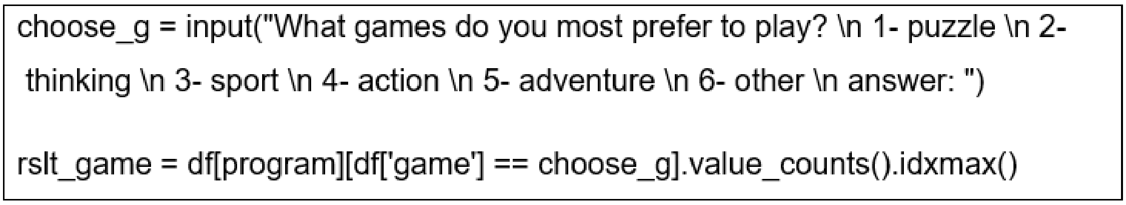


```
choose_h = input("What is your hobby? \n 1- sport \n 2- games \n 3-
 reading \n 4- writing \n 5- traveling \n 6- other \n answer: ")

rslt_hobby = df[program][df['hobby'] == choose_h].value_counts().idxmax()
```

Figure-20: Calculating behavior questions.

The first line is the input explained above in calculating mark questions to get the student input of what hobby they have. The question is the same as the question asked in the survey in order to calculate the frequency. The second line is based on pandas library it look up for program column and get all the values where the value of hobby column is what the input is, then it counts how many times the program is repeated for the chosen hobby and get the max number to calculate the most frequently repeated program.

However, the game column which is another behavior question will be calculated in the same way as the hobby question. Below is an example of the code in the python programing language.



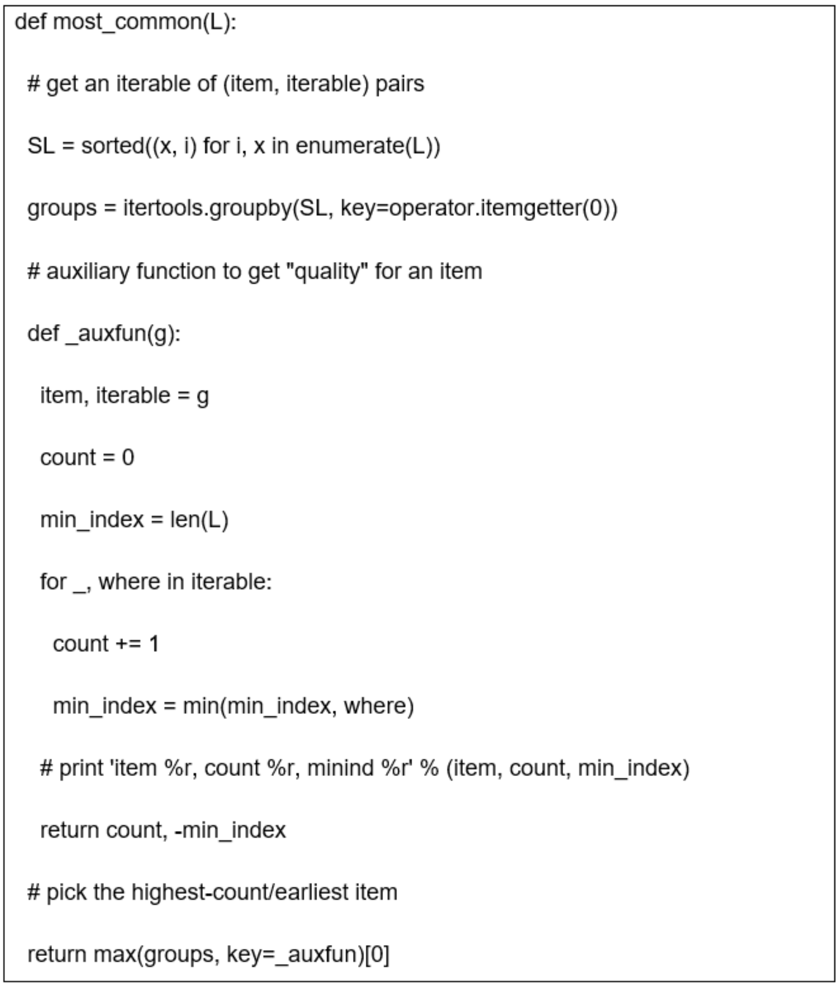
```
choose_g = input("What games do you most prefer to play? \n 1- puzzle \n 2-
 thinking \n 3- sport \n 4- action \n 5- adventure \n 6- other \n answer: ")

rslt_game = df[program][df['game'] == choose_g].value_counts().idxmax()
```

Figure-21: Calculating game question.

The first line is to get the input data from the student and assign it to variable choose_g. The second line is to find the most frequent value in column program where value of column game is what the student entered as input data.

4.3.5 Calculating recommended program

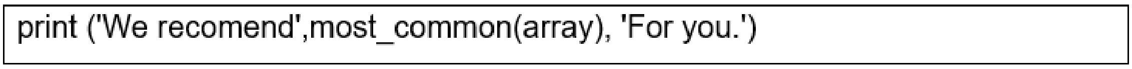
The last step of the algorithm is to recommend the program to the student. The recommendation will be based on the calculated mark questions and behavior questions including game and hobby. The algorithm will push the three results in an array and then using a function it will find the most frequent data to recommend on to the student. The function will go through the array and find the most frequent data in the array to recommend it to the student. Below is the example code of the function called most_common.



```
def most_common(L):
 # get an iterable of (item, iterable) pairs
 SL = sorted((x, i) for i, x in enumerate(L))
 groups = itertools.groupby(SL, key=operator.itemgetter(0))
 # auxiliary function to get "quality" for an item
 def _auxfun(g):
  item, iterable = g
  count = 0
  min_index = len(L)
  for _, where in iterable:
   count += 1
   min_index = min(min_index, where)
  # print 'item %r, count %r, minind %r' % (item, count, min_index)
  return count, -min_index
 # pick the highest-count/earliest item
 return max(groups, key=_auxfun)[0]
```

Figure-22: Most common function.

However, to complete the algorithm a print statement must be implemented with the recommended program. Below is the example code.



```
print ('We recomend',most_common(array), 'For you.')
```

Figure-23: Printing the array.

The line above will print the recommended program to the console with the message we recommend (recommended program) for you. The algorithm will go through all the steps to finally recommend the most suitable program.

Chapter 5 Results

This chapter will show illustration of the results of the recommender system. it will cover the aspects of what are the results based on input data and how the recommender system interacts with the student when entering data and processing it by the system. It will show how different input data will result in different outputs.

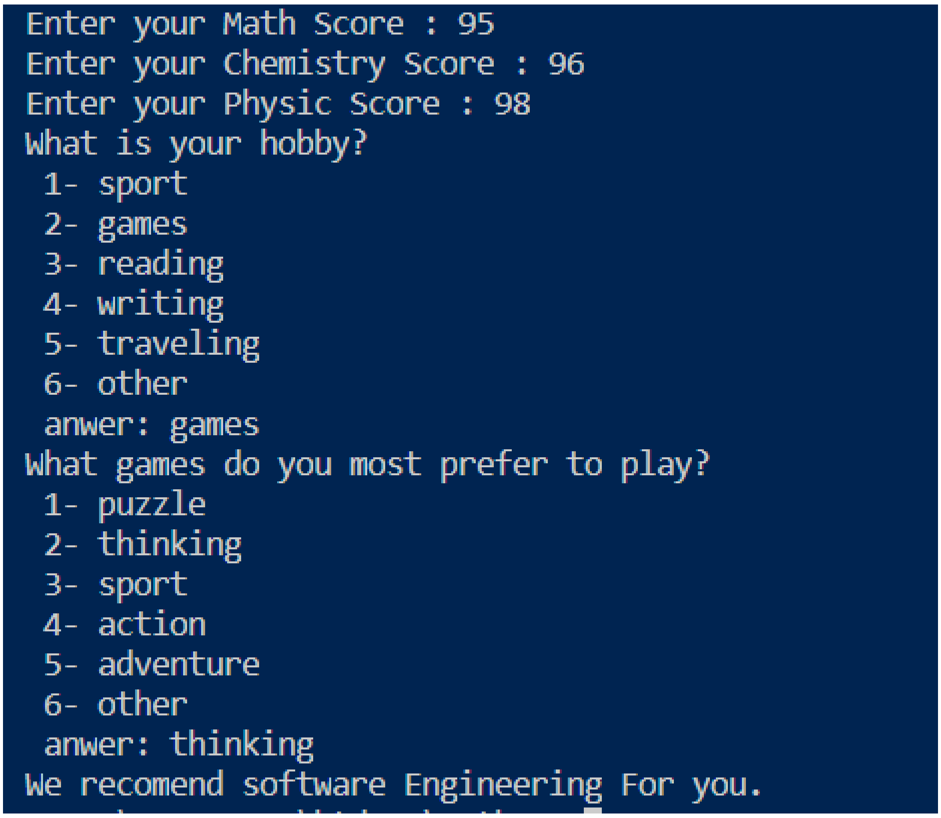
5.1 Testing input data

The data will be tasted and based on the input the output will be generated by the system. Table 3 below will show some different data entering the system and their outputs.

Table-3: Results of different input data.

Math score	Chemistry score	Physic score	Hobby	Game	Recommended output
90	95	96	Sport	puzzle	Computer engineering
60	70	69	Games	Action	business
90	90	98	Sport	Sport	Computer science
95	96	98	Games	Thinking	Software engineering

The algorithm will calculate the results based on the answers of the students and recommend the best suitable program for them. The nice thing about this algorithm is that it count the satisfaction and behavior and marks of the students who finished university or they are currently student, and that gives a new perspective for the new student to go to the best suitable program based on those three factors not just one factor which is the marks. Figure 16 below shows and example interface console for the process.



```
Enter your Math Score : 95
Enter your Chemistry Score : 96
Enter your Physic Score : 98
What is your hobby?
 1- sport
 2- games
 3- reading
 4- writing
 5- traveling
 6- other
 anwer: games
What games do you most prefer to play?
 1- puzzle
 2- thinking
 3- sport
 4- action
 5- adventure
 6- other
 anwer: thinking
We recomend software Engineering For you.
```

Figure-24: console interface of the program.

The figure shows how the questions are asked and how output is shown to the student who uses the system.

5.2 Validating input data

The input data of the recommender system is all validated in order to make the data readable and understandable, and that specifically includes mark questions fields. All the three fields are validated such as that the new student cannot enter more than 100 or a negative number. The behavior questions do not need validating because the new student will just select the answer.

Chapter 6 Conclusion and Future work

In conclusion the system was developed to recommend the best suitable program for new students who graduate from grade 12 in Kurdistan region of Iraq. The system works based on collected data from current students and alumni of universities, and then calculate the best suitable program through three main factors such as marks, behavior and satisfaction. The system takes input data from new students and compares it to the collected data and finds the most frequent program to recommend. The concept of the system is machine learning the more data we give to the system the more it learns about the range of marks and behavior of students with the level of satisfaction. The system is developed by python programing language because of its advanced functionality and library such as pandas to handle CSV files used in the system.

However, the system will always be hungry for data the more it gets data the more accurate the system will be. In our project we used 60% of the data to train the system and then we fed back 40% to test the system and calculate the accuracy of the system, through the testing we calculated the accuracy which was 60% accurate. In future if the system reached a huge set of data, then it will be able to recommend programs at very high rate of accuracy and it can change life for some students.

The future work to be done would be the following.

- Collecting more data for the system to perform better and have more accuracy.
- Modifying the algorithm to adapt full machine learning concepts.
- Providing a nice interface for the system.