UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID: PNT2022TMID13322

TEAM MEMBERS :

GOWSALYA M	(621319104015)
PUVIYASREE M	(621319104042)
RUBADHARSHINI A K	(621319104046)
SRIVIDYA R	(621319104055)

TEAM MENTOR:

Mr. K. ANGURAJU AP/CSE

TEAM EVALUATOR:

Mr. C. RADHAKRISHNAN AP/CSE

INDUSTRY MENTOR:

Mr. P. SANDESH

INDEX

CHAPTER 1

INTRODUCTION

1.1. PROJECT OVERVIEW

In today's world, many students pursue their education outside of their home countries. Generally, because students have little knowledge of the procedures, requirements, and details of the universities, they seek assistance from education consultancy firms to help them successfully secure admission to the universities that are best suited to their profile; however, they must pay a significant amount of money as consultancy fees. Aside from these education consulting firms, there are a few websites and blogs that provide information to students about admission procedures. The disadvantage of the currently available resources is that they are very limited and, in terms of accuracy and reliability, are not truly dependable. The goal of this project is to create a system using machine learning algorithms that we will call University Admission Eligibility Predictor (UAE). It will assist students in determining the likelihood of their university application being accepted. It will also assist them in identifying the universities that are best suited to their profile and providing them with information on those universities. For users to access the university eligibility predictor mechanism, a simple user interface will be created and then we created a web application that allows users to enter their academic information and receive predictions of their chances of admission to the university tier of their choice. It also provides an analysis based on the data set used that shows how the various parameters affect admissions chances. Other than password protection on the website, this project does not address web security issues.

Keywords: University Admission Eligibility Predictor (UAE), Machine Learning, Lin Regression.

1.2. PURPOSE OF THE PROJECT

This is a Requirements Specification Document for a new web-based University Admissions Predictor. This project would be AI-based application that asks for the users to input their academic transcripts data and calculates their chances of admission into the University Tier that they selected. It also provides an analysis of the data and shows how chances of admissions can depend on various factors. This document describes the scope, objectives and goals of the system. In addition to describing the non-functional requirements, this document models the functional requirements with use cases, interaction diagrams and class models. This document is intended to direct the design and implementation of the target system in an object-oriented language.

1.3. MOTIVATION

In the current world scenario, it is not enough for a student to just have an Under Graduate degree. Most employers now look for higher qualifications in their new recruits. As a result, the demands for a good higher education are at time high. A lot of students from India prefer to continue their higher education with foreign universities, especially in the United States. In order to get admitted to these foreign universities, a set of academic requirements are needed. However, because of the sheer number of universities of different levels, students are often stuck in a dilemma till the very last minute as to whether or not their applications will be accepted or not as no concrete documentation is available which lists the requirements.

1.4. PROJECT SCOPE

The scope of this project is a web application that allows users to enter their academic data and get predictions of their chances of admission in the university tier of their choosing. It also provides an analysis based on the data set used that shows how the different parameters affect the chances of admissions. Issues of web security other than password protection within the website are not part of this project.

CHAPTER 2

LITERATURE SURVEY

[1] TITLE : Graduate admission prediction using machine learning techniques (2021).

Authors: K.Jeevan Ratnakar, G. Koteswara Rao, B. DurgaPrasanth Kumar, G.prithvi, D.Venkata Sai Eswar.

Reference Link:

https://www.researchgate.net/publication/348433004_Graduate_Admission_Prediction_Using_Machine_Learning

Proposed Problem:

The world markets are developing rapidly and continuously looking for the best knowledge and experience among people. Young workers who want to stand out in their jobs are always looking for higher degrees that can help them in improving their skills and knowledge. As a result, the number of students applying for graduate studies has increased in the last decade. This fact has motivated us to study the grades of students and the possibility of admission for master's programs that can help universities in predicting the possibility of accepting master's students submitting each year and provide the needed resources.

Proposed solution:

The decision tree algorithm based on the test attributes like GRE, TOEFL, CGPA are have a possibilities of a chance of admission is calculated.

Pros and cons:

Based on the different parameters that are provided in the dataset have been trained and tested with testing data whether the student has a chance or not, if parameter missing, it reflects on the result.

[2] TITLE: College admission prediction using Ensemble Machine Learning models. (2021)

Authors: Vandit Manish Jain, Rihaan Satia.

Reference Link: https://www.irjet.net/archives/V8/i12/IRJET-V8I1266.pdf

Proposed Problem:

Educational organizations have always played an important and vital role in society for development and growth of any individual. There are different college prediction apps and websites being maintained contemporarily, but using themis tedious tosome extent,due to the lack of articulate information regarding colleges, and the time consumed in searching the best deserving college. The problem statement, hence being tackled, is to design a college prediction/prediction system and to provide a probabilistic insight into college administration for overall rating, cut-offs of the colleges, admission intake and preferences of students. Also, it helps students avoid spending time and money on counsellor and stressful research related to finding a suitable college.

It has always been a troublesome process for students in finding the perfect university and course for their further studies. At times they do know which stream they want to get into, but it is not easy for them to find colleges based on their academic marks and other performances. We aim to develop and provide a place which would give a probabilistic output as to how likely it is to get into a university given upon their details.

Proposed solution:

Linear regression model is a method used as response for only a single feature, it is based on supervised learning.

Pros and cons:

Every year millions of students apply to universities to begin their educational life. So, every year we have to Change.

[3] TITLE: Prediction for university admission to students using Machine Learning (2020)

Authors: Chithra Apoorva D A, Malepati ChanduNath, Peta Rohith, Bindu Shree.S, Swaroop.S

Reference Link:

https://www.ijrte.org/wpcontent/uploads/papers/v8i6/F9043038620.pdf

Proposed Problem:

In today's education world there are many number of students who want to pursue higher education after engineering or any graduate degree course. Higher education in the sense, some people want to do M. Tech through GATE or through any educational institute entrance examination and some people want to do MBA through CAT or through any respective educational institute entrance examination and some people want to do Masters in abroad universities. We are focusing on only the students who want to pursue their higher education in abroad universities. Students who want to do masters in America have to write GRE (Graduate Records Examination) and TOEFL/IELTS (Test of English as a Foreign Language/International English Language Testing System). Once they have attended the exams they have to prepare their SOP (statement of purpose) and LOR (letter of recommendation) which are one of the crucial factors they have to consider.

Proposed solution:

Bayesian Networks Algorithm have been used to create a decision support network for evaluating the application submitted by foreign students of the university.

Pros and cons:

Previous research done in this area used Naive Bayes algorithm but the main drawback is they didn’t consider all the factors which will contribute in the Student admission process like TOEFL/ SOP, LOR and UG.

[4] TITLE: The Predictive validity of admission criteria for college assignment in Saudi Universities (2020)

Authors: Abdulaziz Althewini.

Reference Link: https://files.eric.ed.gov/fulltext/EJ1249542.pdf

Proposed Problem:

Admission criteria can be used to predict Saudi student performance in college, but significant differences across several studies exists. This study explores the predictive power of admission criteria for college assignment using King Saud bin Abdul aziz University for Health Sciences as a model. Scores from high school and standardized tests were collected for 1,595 students. Data were analysed with multinomial logistic and multivariate linear regression. A formula was generated to determine student college assignment based on their admission criteria profile.

The results showed that all admission criteria were significant predictors of college assignment but accounted for only 21.1% of the variance. Based on the results of this study, admission criteria may not be reliable predictors of college assignment on their own, and additional criteria for measuring student success are needed.

The current study is unique because existing literature only uses student GPA as an outcome, while this study uses college assignment.

Proposed solution:

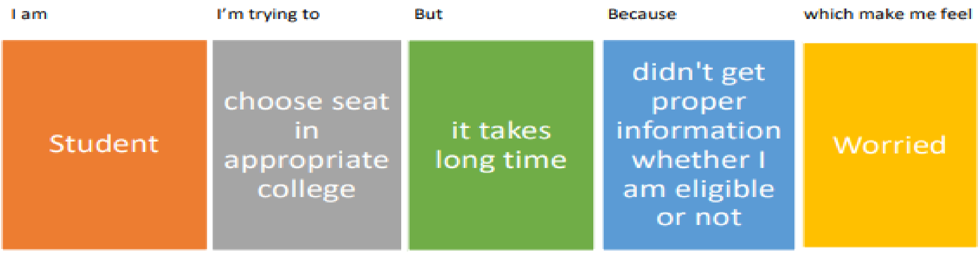
The college assignment using King Saud bin Abdul aziz University. Data were analyzed with multinomial logistic and multivariate linear regression.

Pros and cons:

Scores from high school and standardized tests were collected for 1,595 students. Based on the results of this study, admission criteria may not be reliable for prediction for the admission in their dream university.

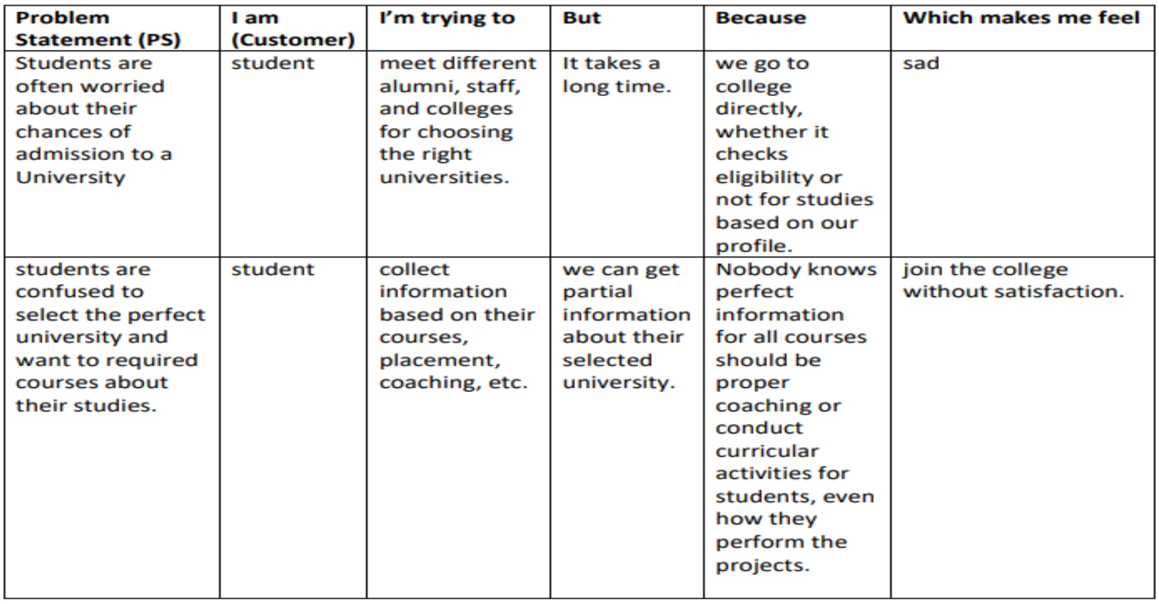
2.2. PROBLEM STATEMENT

- The problem statement is to develop a college prediction system that will offer a probabilistic perspective on college administration for factors including overall rating, college cut-offs, admission intake, and student preferences.
- Finding the ideal college and course for continuing their education has always been difficult for students. Sometimes students are certain of the stream they want to enter, but finding universities that match their academic standing and other achievements is difficult for them.
- Our goal is to create and offer a location that would provide a probabilistic output of how likely it is for someone to be admitted to a university given their information.



I am	I'm trying to	But	Because	which make me feel
Student	choose seat in appropriate college	it takes long time	didn't get proper information whether I am eligible or not	Worried

Fig. 2.2(a). Problem Statement



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
Students are often worried about their chances of admission to a University	student	meet different alumni, staff, and colleges for choosing the right universities.	It takes a long time.	we go to college directly, whether it checks eligibility or not for studies based on our profile.	sad
students are confused to select the perfect university and want to required courses about their studies.	student	collect information based on their courses, placement, coaching, etc.	we can get partial information about their selected university.	Nobody knows perfect information for all courses should be proper coaching or conduct curricular activities for students, even how they perform the projects.	join the college without satisfaction.

Fig. 2.2(b). Defining Problem Statement

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS

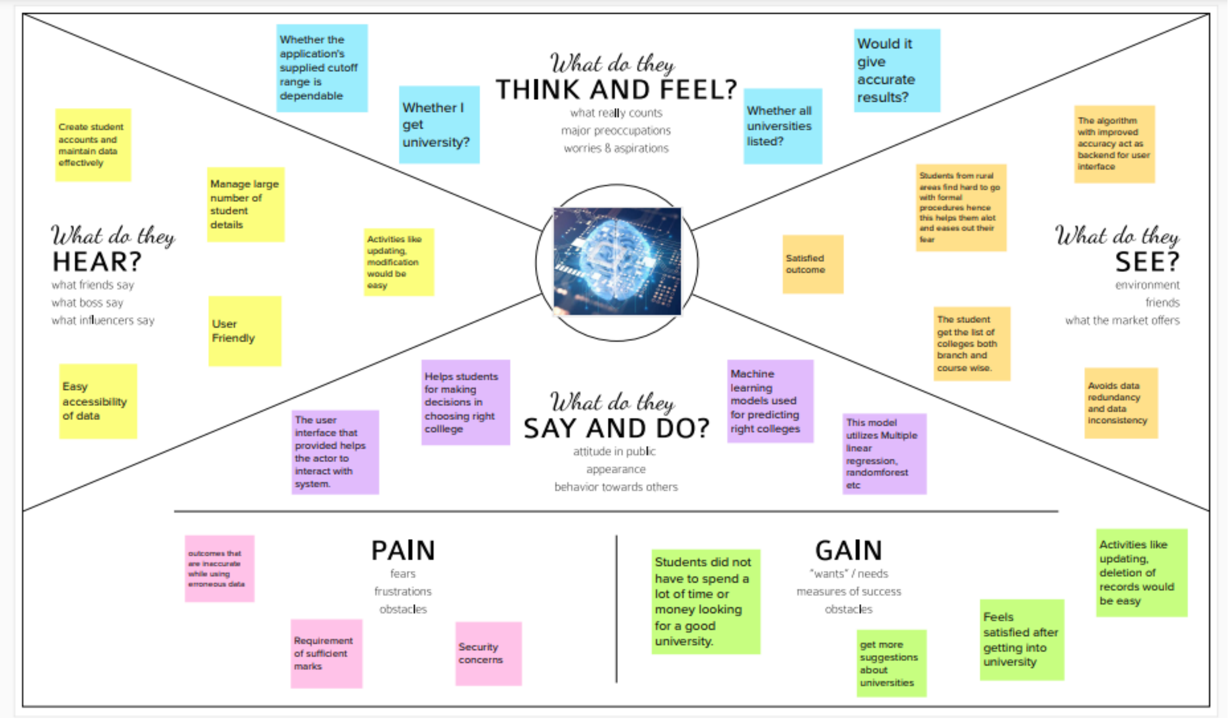
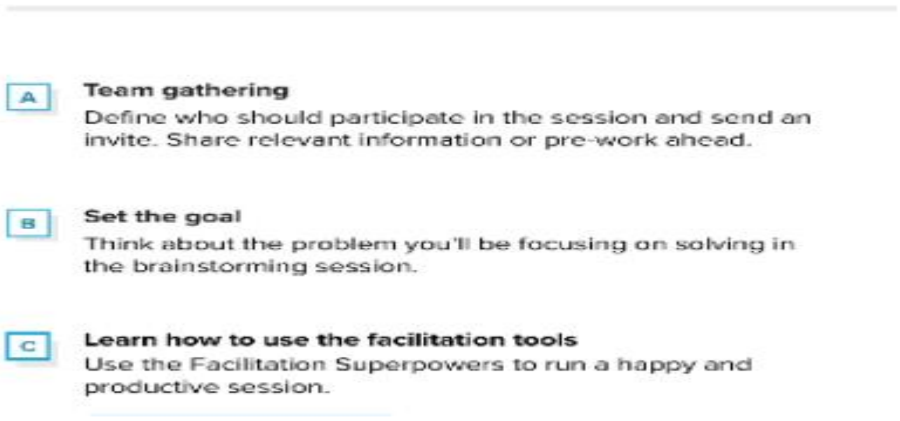

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Whether the application's supplied cutoff range is dependable
Whether I get university?
Whether all universities listed?
Would it give accurate results?
What do they
HEAR?
what friends say
what boss say
what influencers say
Create student accounts and maintain data effectively
Manage large number of student details
Activities like updating, modification would be easy
User Friendly
Easy accessibility of data
What do they
SEE?
environment
friends
what the market offers
The algorithm with improved accuracy act as backend for user interface
Students from rural areas find hard to go with formal procedures hence this helps them alot and eases out their fear
Satisfied outcome
The student get the list of colleges both branch and course wise.
Avoids data redundancy and data inconsistency
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
The user interface that provided helps the actor to interact with system.
Helps students for making decisions in choosing right colllege
Machine learning models used for predicting right colleges
This model utilizes Multiple linear regression, randomforest etc
PAIN
fears
frustrations
obstacles
outcomes that are inaccurate while using erroneous data
Requirement of sufficient marks
Security concerns
GAIN
"wants" / needs
measures of success
obstacles
Students did not have to spend a lot of time or money looking for a good university.
get more suggestions about universities
Feels satisfied after getting into university
Activities like updating, deletion of records would be easy

Fig. 3.1. Emphathy Map

3.2. IDEATION AND BRAIN STORMING

STEP 1: TEAM COLLABORATION & SELECTING THE PROBLEM STATEMENT



A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Fig. 3.2(a) Team Collaboration

PROBLEM

How might the students get the university?

PROBLEM

What are the parameters involved in getting the university?

PROBLEM

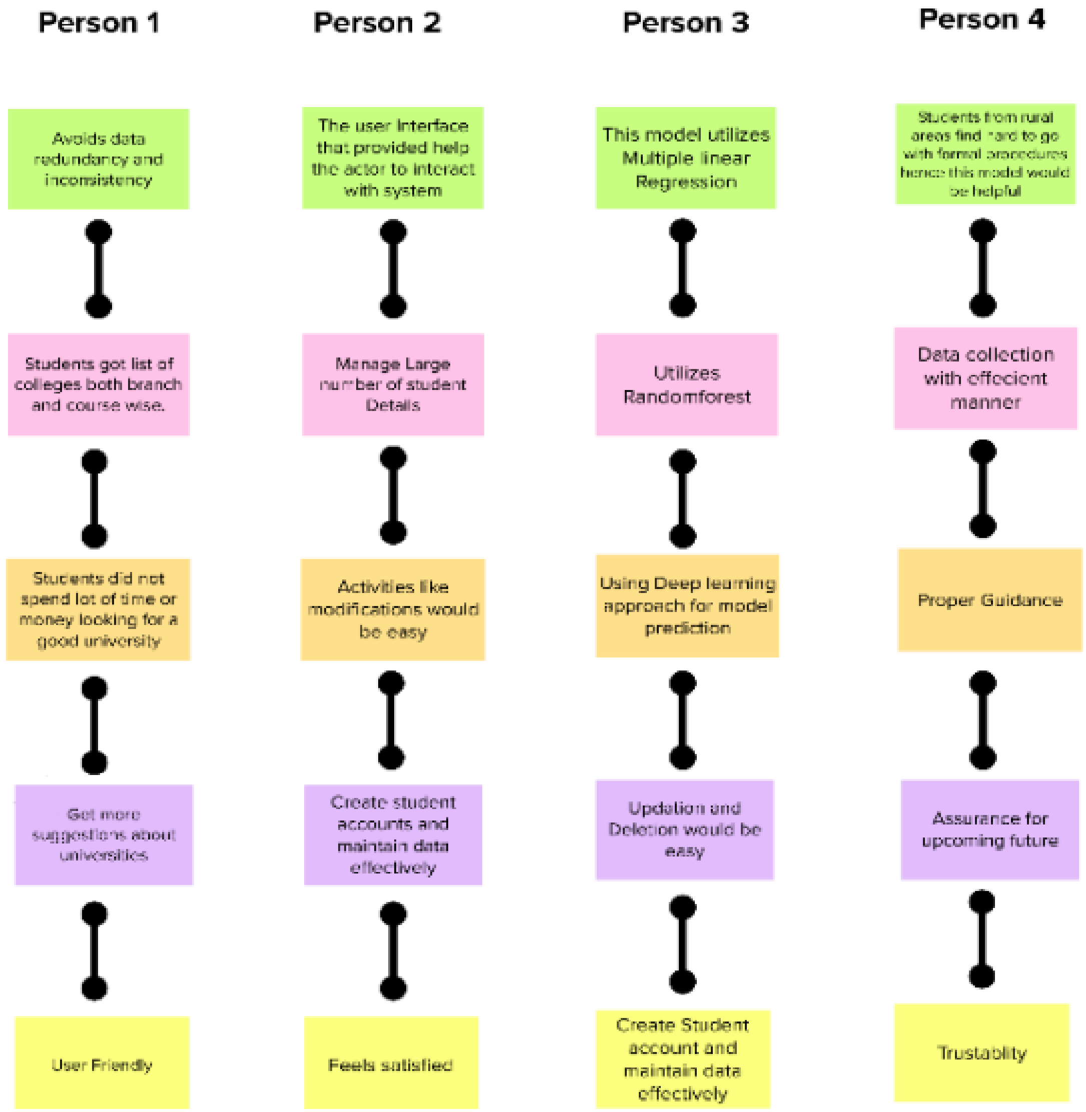
What are the parameters involved in getting the university?

PROBLEM

Wheather the application's supplied cuttoff range is dependable?

Fig. 3.2(b) Selecting the problem statement

STEP 2: Brainstorm, Idea Listing, and Grouping



Person 1

Avoids data redundancy and inconsistency

Students got list of colleges both branch and course wise.

Students did not spend lot of time or money looking for a good university

Get more suggestions about universities

User Friendly

Person 2

The user Interface that provided help the actor to interact with system

Manage Large number of student Details

Activities like modifications would be easy

Create student accounts and maintain data effectively

Feels satisfied

Person 3

This model utilizes Multiple linear Regression

Utilizes Randomforest

Using Deep learning approach for model prediction

Updation and Deletion would be easy

Create Student account and maintain data effectively

Person 4

Students from rural areas find hard to go with formal procedures hence this model would be helpful

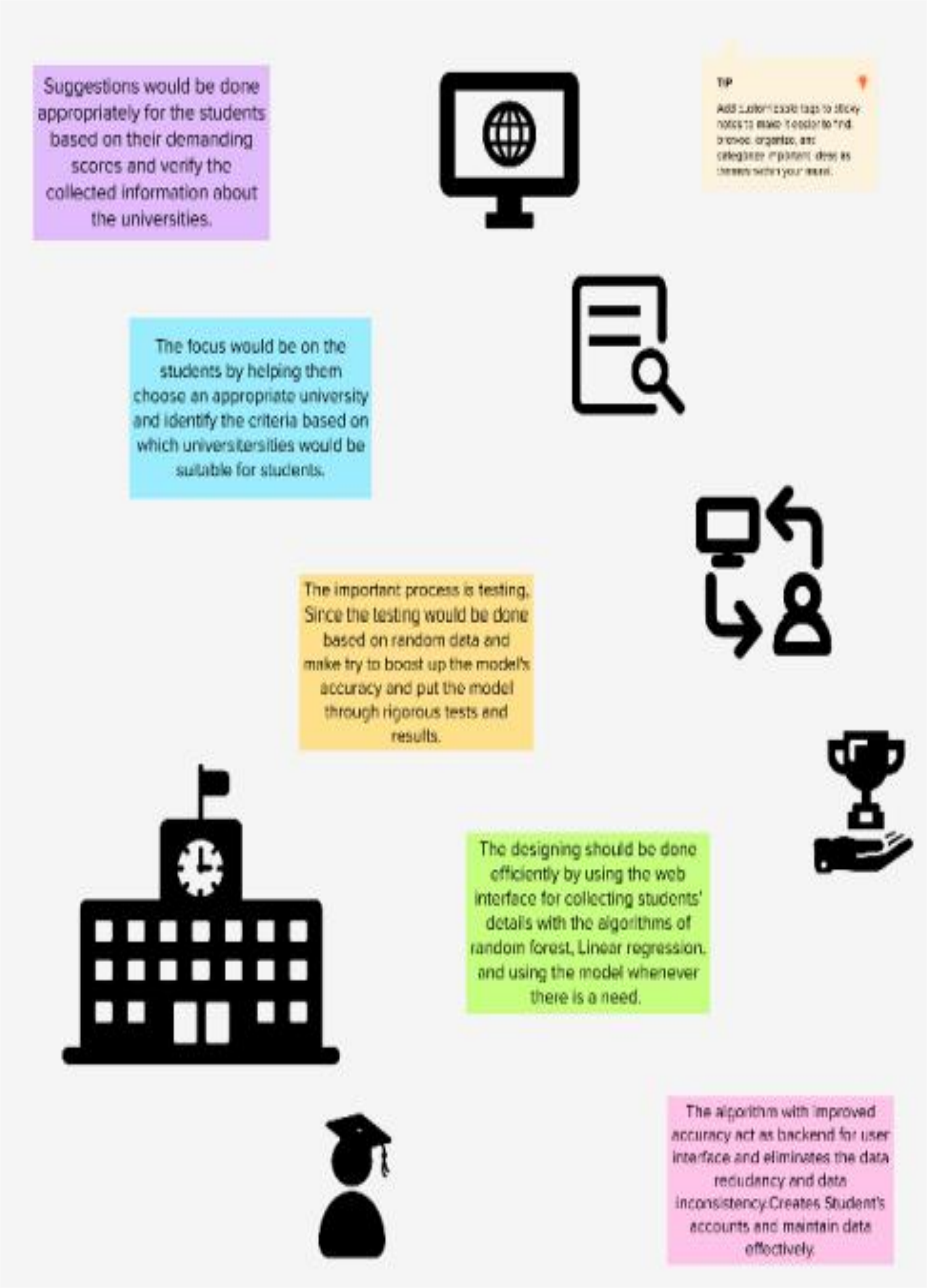
Data collection with effecient manner

Proper Guidance

Assurance for upcoming future

Trustablity

Fig. 3.2(c) Brain storm



Suggestions would be done appropriately for the students based on their demanding scores and verify the collected information about the universities.

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your board.

The focus would be on the students by helping them choose an appropriate university and identify the criteria based on which universitersities would be suitable for students.

The important process is testing. Since the testing would be done based on random data and make try to boost up the model's accuracy and put the model through rigorous tests and results.

The designing should be done efficiently by using the web interface for collecting students' details with the algorithms of random forest, Linear regression, and using the model whenever there is a need.

The algorithm with improved accuracy act as backend for user interface and eliminates the data redudancy and data inconsistency.Creates Student's accounts and maintain data effectively.

Fig. 3.2(d) Idea Listing and Grouping

STEP 3: IDEA PRIORITIZATION

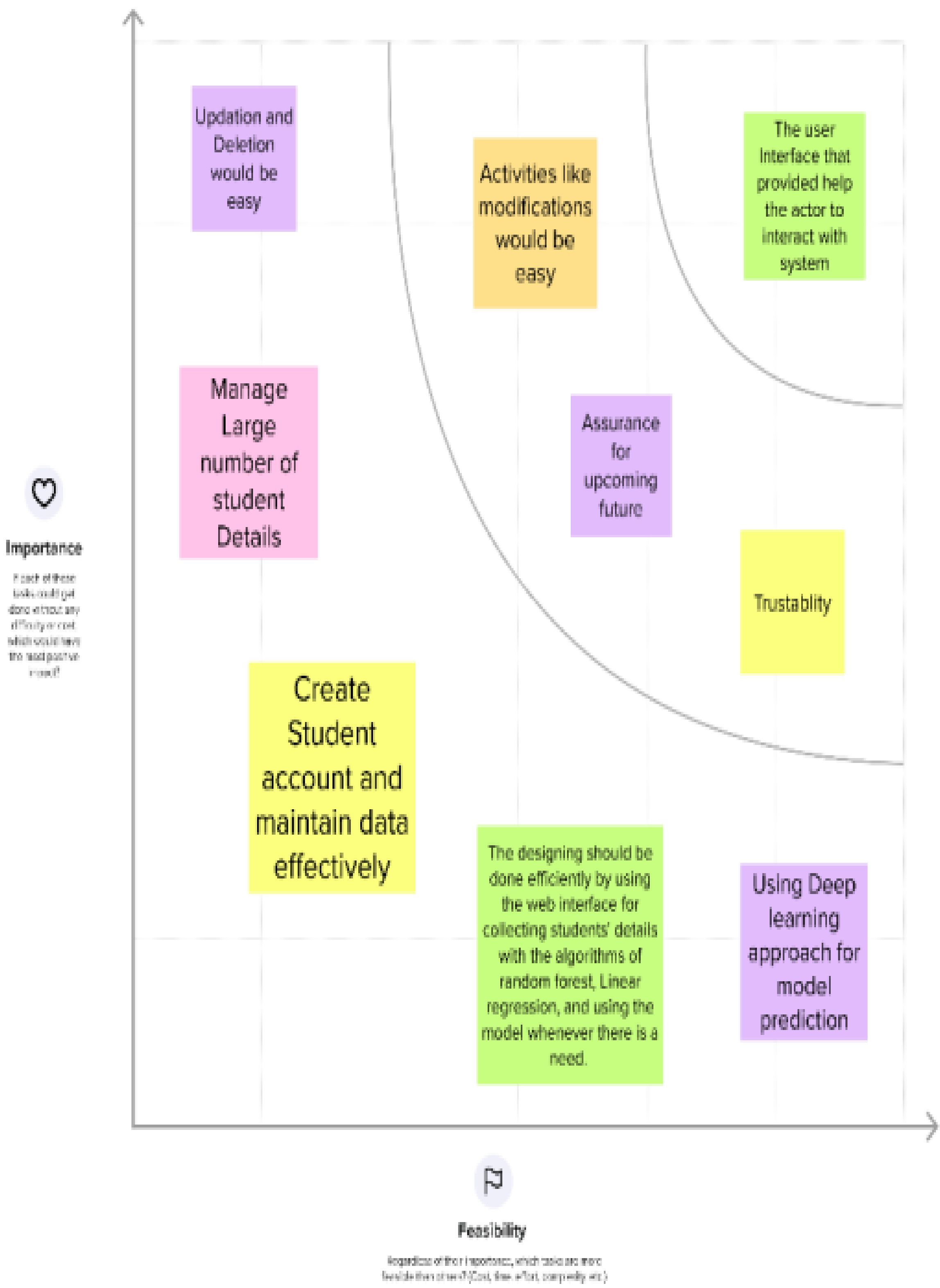
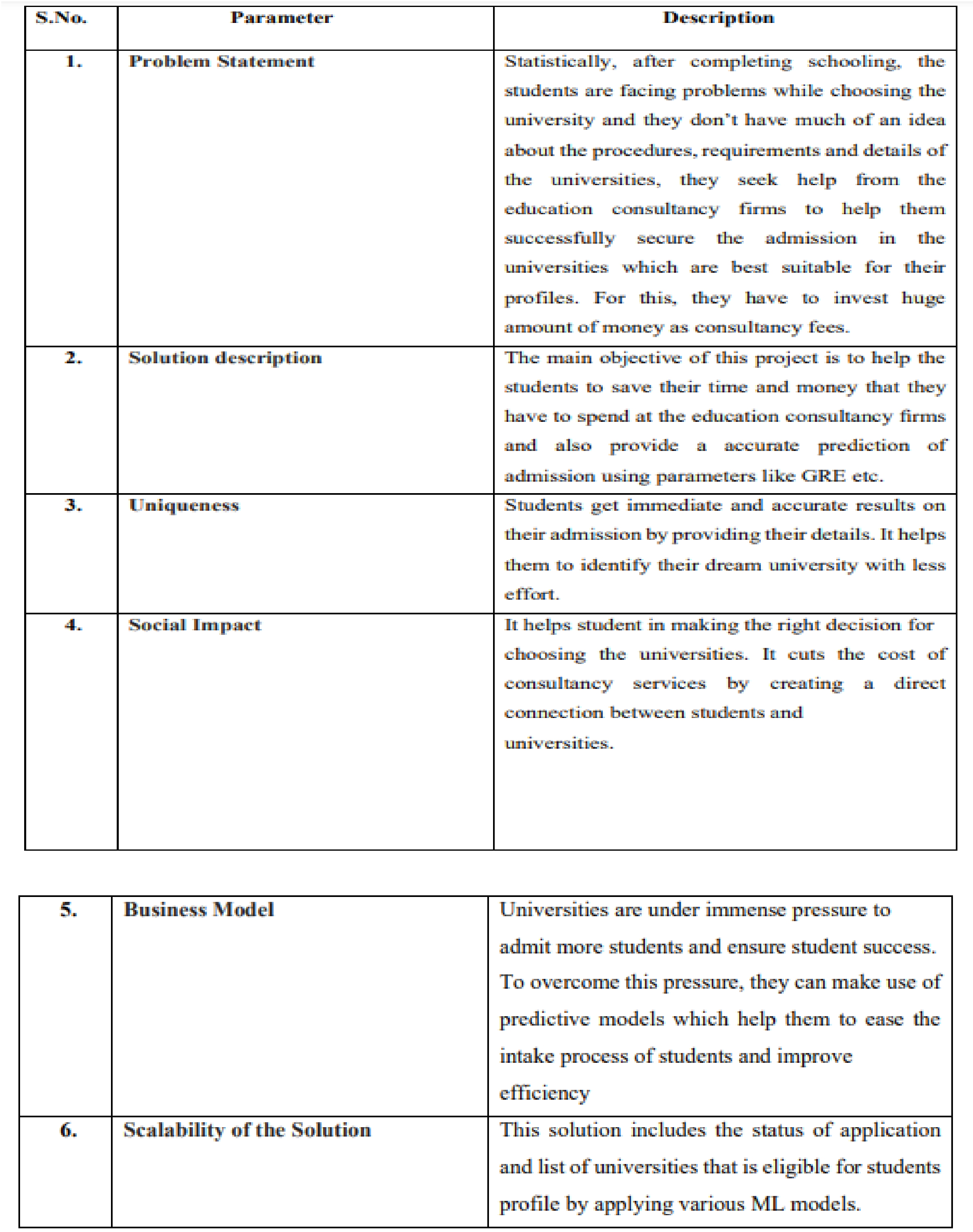

Updation and Deletion would be easy
Activities like modifications would be easy
The user Interface that provided help the actor to interact with system
Manage Large number of student Details
Assurance for upcoming future
Trustablity
Create Student account and maintain data effectively
The designing should be done efficiently by using the web interface for collecting students' details with the algorithms of random forest, Linear regression, and using the model whenever there is a need.
Using Deep learning approach for model prediction
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Fig. 3.2(e) Idea Prioritization

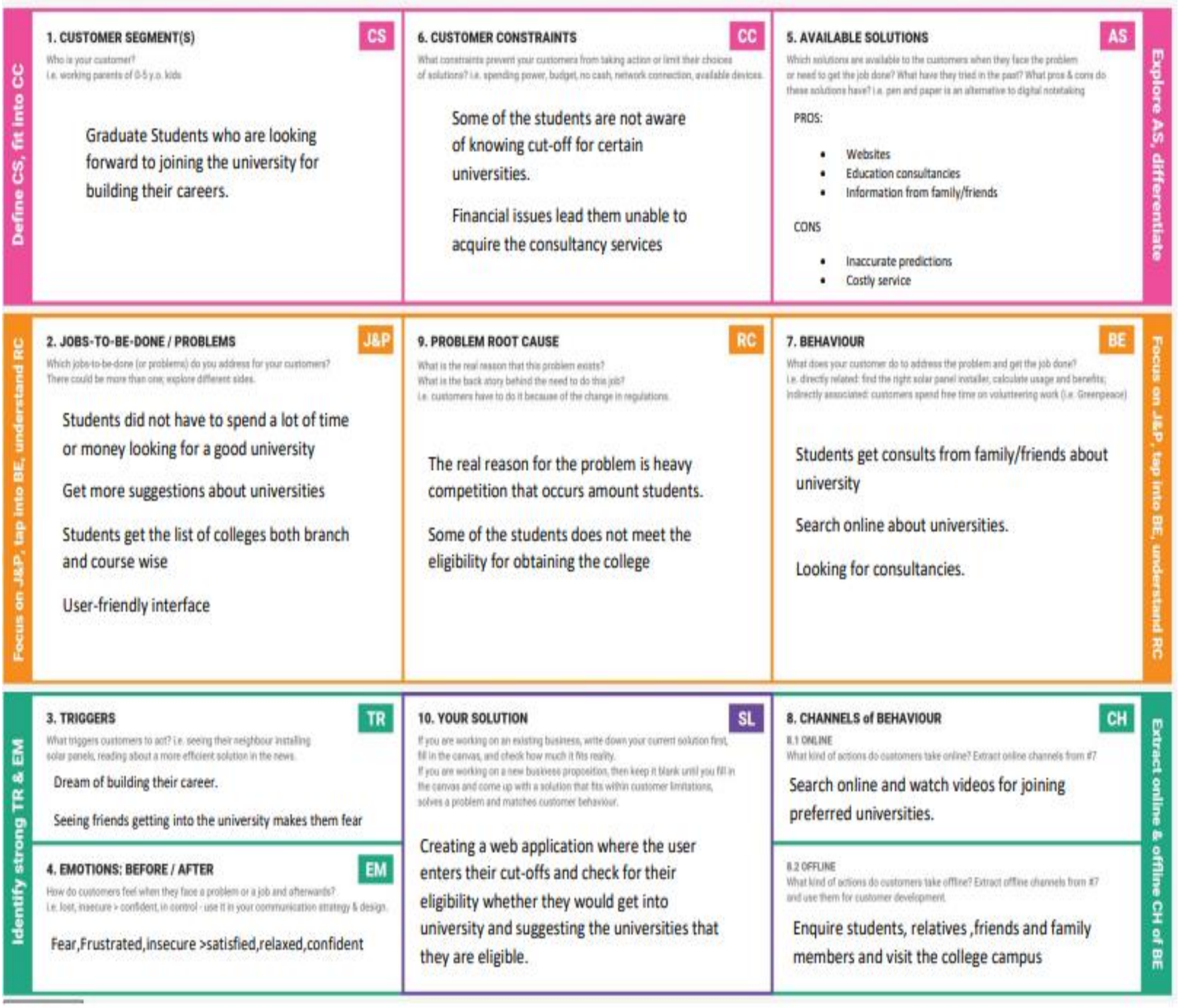
3.3. PROPOSED SOLUTION



S.No.	Parameter	Description
1.	Problem Statement	Statistically, after completing schooling, the students are facing problems while choosing the university and they don't have much of an idea about the procedures, requirements and details of the universities, they seek help from the education consultancy firms to help them successfully secure the admission in the universities which are best suitable for their profiles. For this, they have to invest huge amount of money as consultancy fees.
2.	Solution description	The main objective of this project is to help the students to save their time and money that they have to spend at the education consultancy firms and also provide a accurate prediction of admission using parameters like GRE etc.
3.	Uniqueness	Students get immediate and accurate results on their admission by providing their details. It helps them to identify their dream university with less effort.
4.	Social Impact	It helps student in making the right decision for choosing the universities. It cuts the cost of consultancy services by creating a direct connection between students and universities.
5.	Business Model	Universities are under immense pressure to admit more students and ensure student success. To overcome this pressure, they can make use of predictive models which help them to ease the intake process of students and improve efficiency
6.	Scalability of the Solution	This solution includes the status of application and list of universities that is eligible for students profile by applying various ML models.

Fig. 3.3 Proposed Solution

3.4. PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Graduate Students who are looking forward to joining the university for building their careers.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Some of the students are not aware of knowing cut-off for certain universities.

Financial issues lead them unable to acquire the consultancy services

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

PROS:

- Websites
- Education consultancies
- Information from family/friends

CONS

- Inaccurate predictions
- Costly service

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Students did not have to spend a lot of time or money looking for a good university

Get more suggestions about universities

Students get the list of colleges both branch and course wise

User-friendly interface

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

The real reason for the problem is heavy competition that occurs amount students.

Some of the students does not meet the eligibility for obtaining the college

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Students get consults from family/friends about university

Search online about universities.

Looking for consultancies.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Dream of building their career.

Seeing friends getting into the university makes them fear

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Fear,Frustrated,insecure >satisfied,relaxed,confident

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Creating a web application where the user enters their cut-offs and check for their eligibility whether they would get into university and suggesting the universities that they are eligible.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

Search online and watch videos for joining preferred universities.

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Enquire students, relatives ,friends and family members and visit the college campus

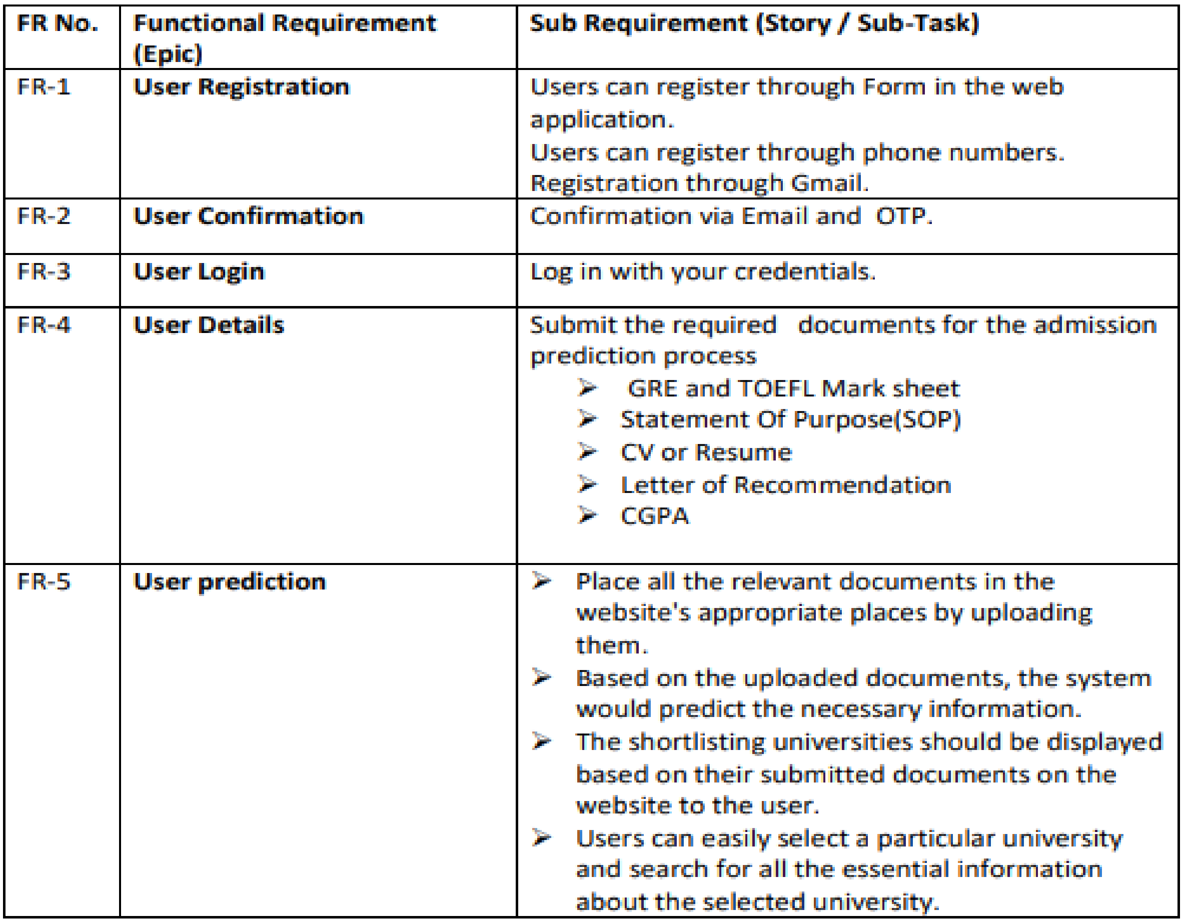
Extract online & offline CH of BE

Fig. 3.4 Problem Solution Fit

CHAPTER 4

REQUIREMENT ANALYSIS

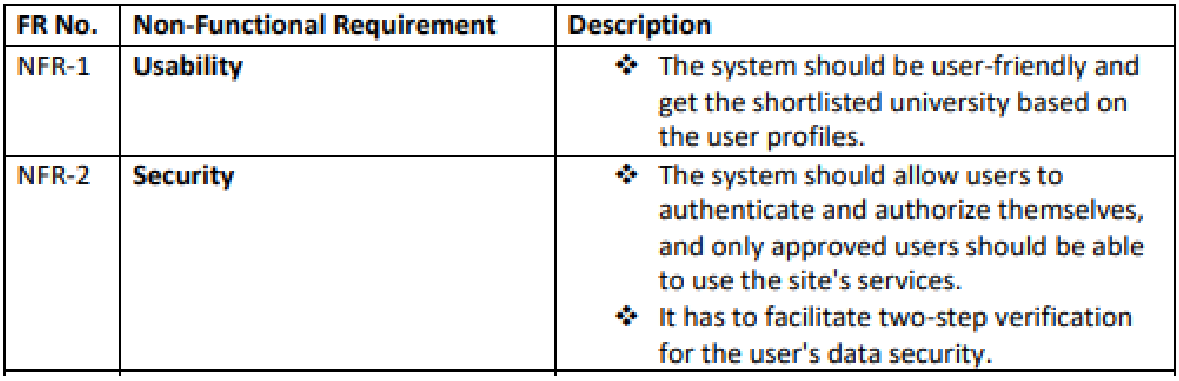
4.1. FUNCTIONAL REQUIREMENT



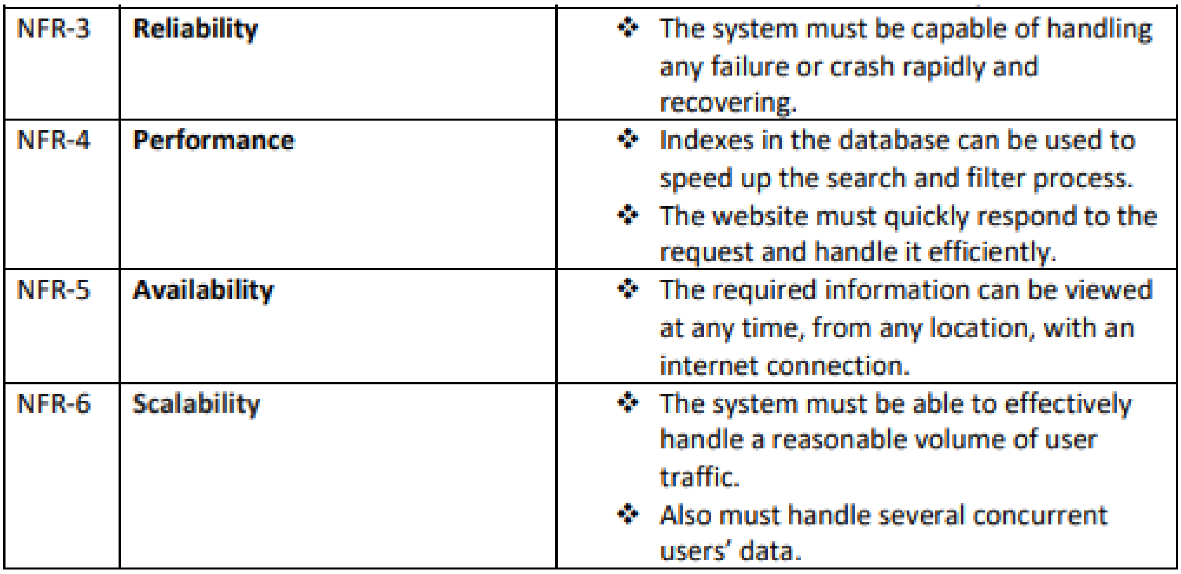
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Registration**	Users can register through Form in the web application. Users can register through phone numbers. Registration through Gmail.
FR-2	**User Confirmation**	Confirmation via Email and OTP.
FR-3	**User Login**	Log in with your credentials.
FR-4	**User Details**	Submit the required documents for the admission prediction process ➢ GRE and TOEFL Mark sheet ➢ Statement Of Purpose(SOP) ➢ CV or Resume ➢ Letter of Recommendation ➢ CGPA
FR-5	**User prediction**	➢ Place all the relevant documents in the website's appropriate places by uploading them. ➢ Based on the uploaded documents, the system would predict the necessary information. ➢ The shortlisting universities should be displayed based on their submitted documents on the website to the user. ➢ Users can easily select a particular university and search for all the essential information about the selected university.

Fig. 4.1.Functional Requirements

4.2. NON-FUNCTIONAL REQUIREMENT



FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	❖ The system should be user-friendly and get the shortlisted university based on the user profiles.
NFR-2	**Security**	❖ The system should allow users to authenticate and authorize themselves, and only approved users should be able to use the site's services. ❖ It has to facilitate two-step verification for the user's data security.



NFR-3	**Reliability**	❖ The system must be capable of handling any failure or crash rapidly and recovering.
NFR-4	**Performance**	❖ Indexes in the database can be used to speed up the search and filter process. ❖ The website must quickly respond to the request and handle it efficiently.
NFR-5	**Availability**	❖ The required information can be viewed at any time, from any location, with an internet connection.
NFR-6	**Scalability**	❖ The system must be able to effectively handle a reasonable volume of user traffic. ❖ Also must handle several concurrent users' data.

Fig. 4.2.Non-Functional Requirements

CHAPTER 5

PROJECT DESIGN

5.1. DATAFLOW DIAGRAM

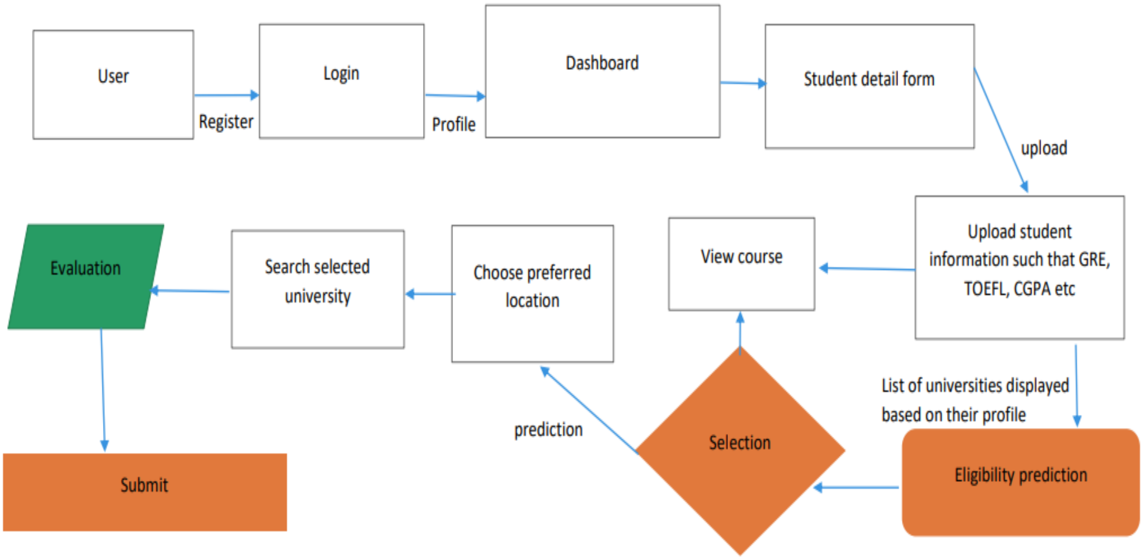

User
Register
Login
Profile
Dashboard
Student detail form
upload
Upload student information such that GRE, TOEFL, CGPA etc
View course
Choose preferred location
Search selected university
Evaluation
Submit
prediction
Selection
List of universities displayed based on their profile
Eligibility prediction

Fig. 5.1. Dataflow Diagram

5.2. TECHNICAL ARCHITECTURE

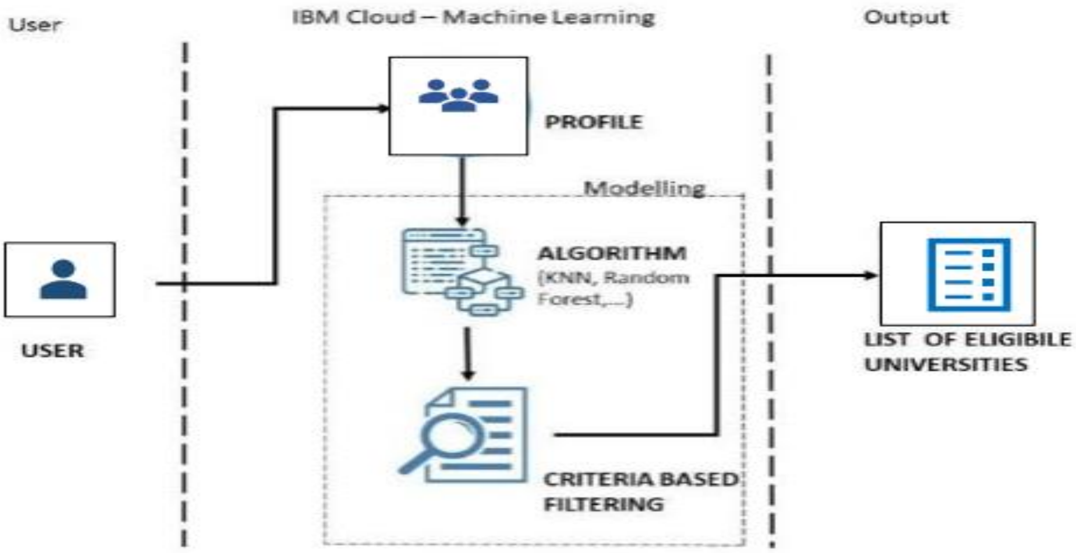


Fig. 5.2. Technical Architecture

5.3. SOLUTION ARCHITECTURE

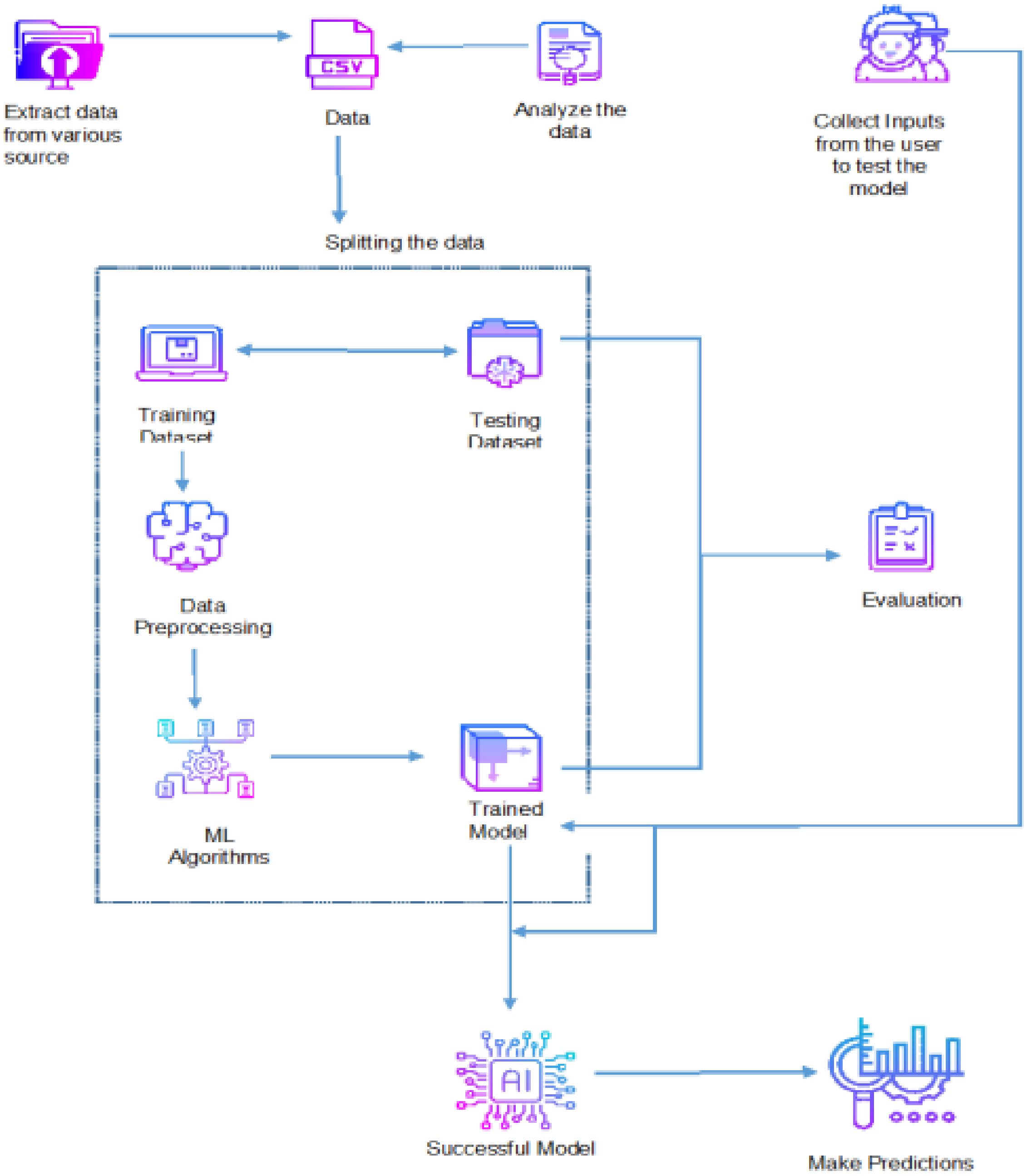
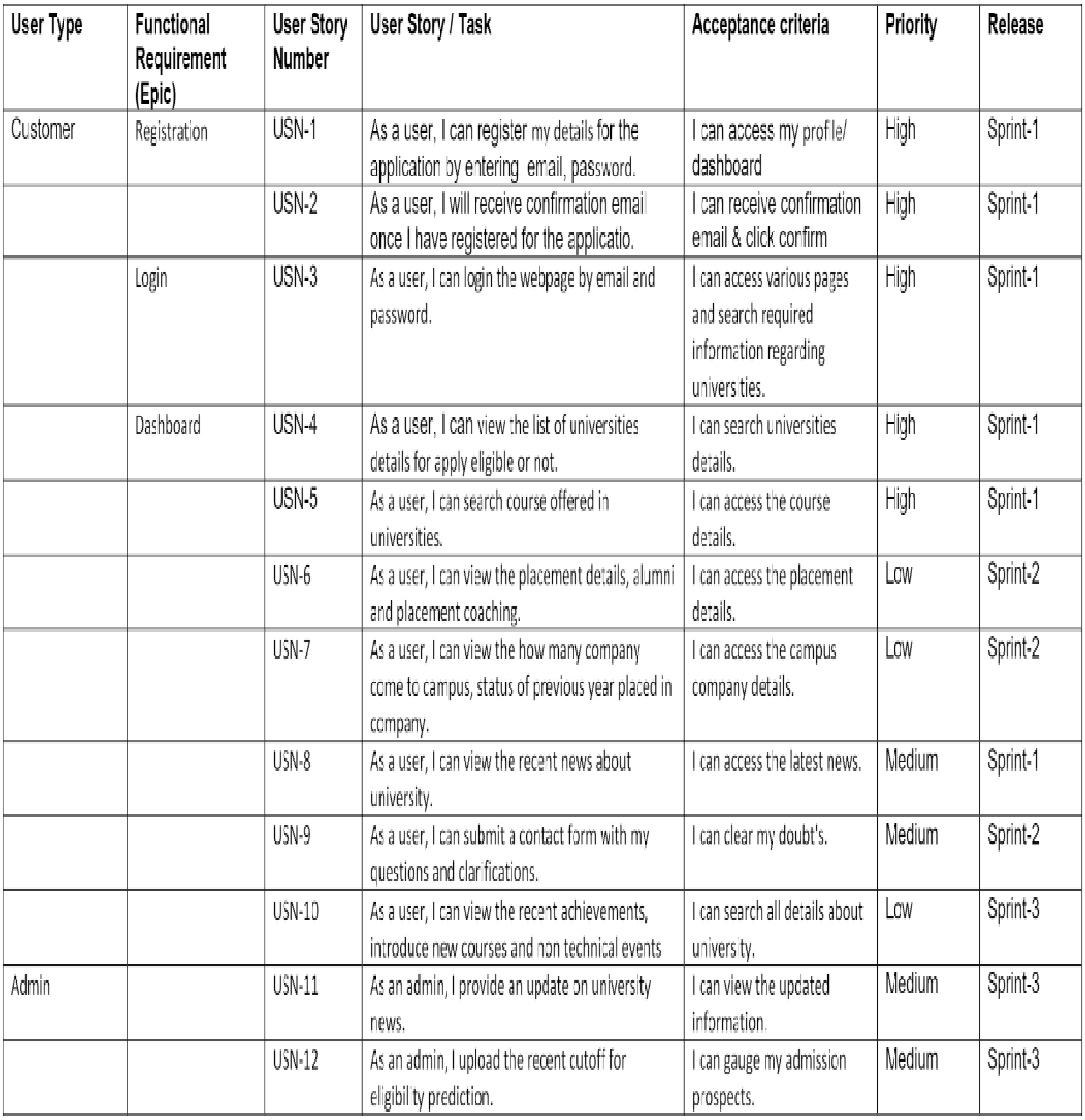


Fig. 5.3. Solution Architecture

5.4. USER STORIES



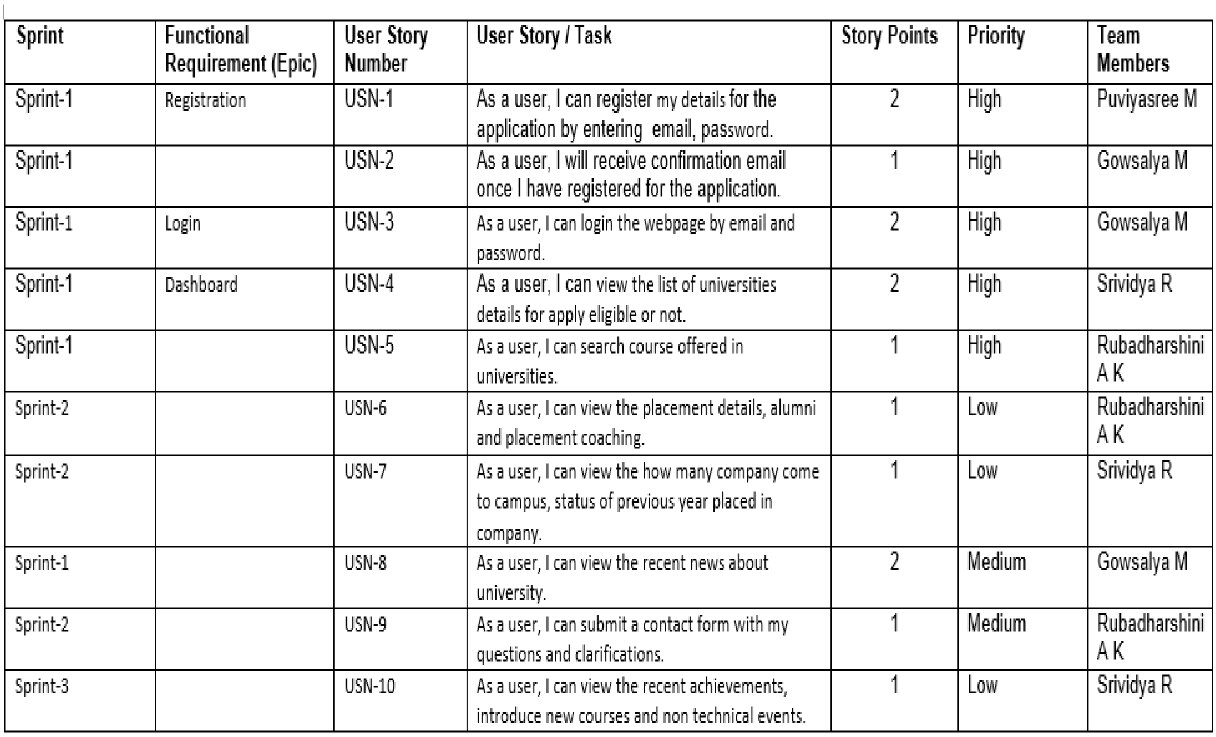
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register my details for the application by entering email, password.	I can access my profile/ dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the applicatio.	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user, I can login the webpage by email and password.	I can access various pages and search required information regarding universities.	High	Sprint-1
	Dashboard	USN-4	As a user, I can view the list of universities details for apply eligible or not.	I can search universities details.	High	Sprint-1
		USN-5	As a user, I can search course offered in universities.	I can access the course details.	High	Sprint-1
		USN-6	As a user, I can view the placement details, alumni and placement coaching.	I can access the placement details.	Low	Sprint-2
		USN-7	As a user, I can view the how many company come to campus, status of previous year placed in company.	I can access the campus company details.	Low	Sprint-2
		USN-8	As a user, I can view the recent news about university.	I can access the latest news.	Medium	Sprint-1
		USN-9	As a user, I can submit a contact form with my questions and clarifications.	I can clear my doubt's.	Medium	Sprint-2
		USN-10	As a user, I can view the recent achievements, introduce new courses and non technical events	I can search all details about university.	Low	Sprint-3
Admin		USN-11	As an admin, I provide an update on university news.	I can view the updated information.	Medium	Sprint-3
		USN-12	As an admin, I upload the recent cutoff for eligibility prediction.	I can gauge my admission prospects.	Medium	Sprint-3

Fig. 5.4. User stories

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

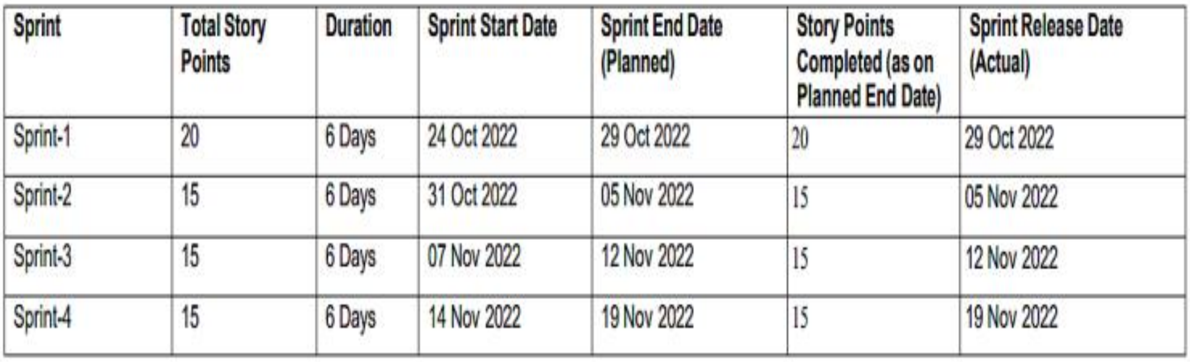
6.1. SPRINT PLANNING & ESTIMATION



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register my details for the application by entering email, password.	2	High	Puviyasree M
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application.	1	High	Gowsalya M
Sprint-1	Login	USN-3	As a user, I can login the webpage by email and password.	2	High	Gowsalya M
Sprint-1	Dashboard	USN-4	As a user, I can view the list of universities details for apply eligible or not.	2	High	Srividya R
Sprint-1		USN-5	As a user, I can search course offered in universities.	1	High	Rubadharshini A K
Sprint-2		USN-6	As a user, I can view the placement details, alumni and placement coaching.	1	Low	Rubadharshini A K
Sprint-2		USN-7	As a user, I can view the how many company come to campus, status of previous year placed in company.	1	Low	Srividya R
Sprint-1		USN-8	As a user, I can view the recent news about university.	2	Medium	Gowsalya M
Sprint-2		USN-9	As a user, I can submit a contact form with my questions and clarifications.	1	Medium	Rubadharshini A K
Sprint-3		USN-10	As a user, I can view the recent achievements, introduce new courses and non technical events.	1	Low	Srividya R

Fig. 6.1. Sprint planning & Estimation

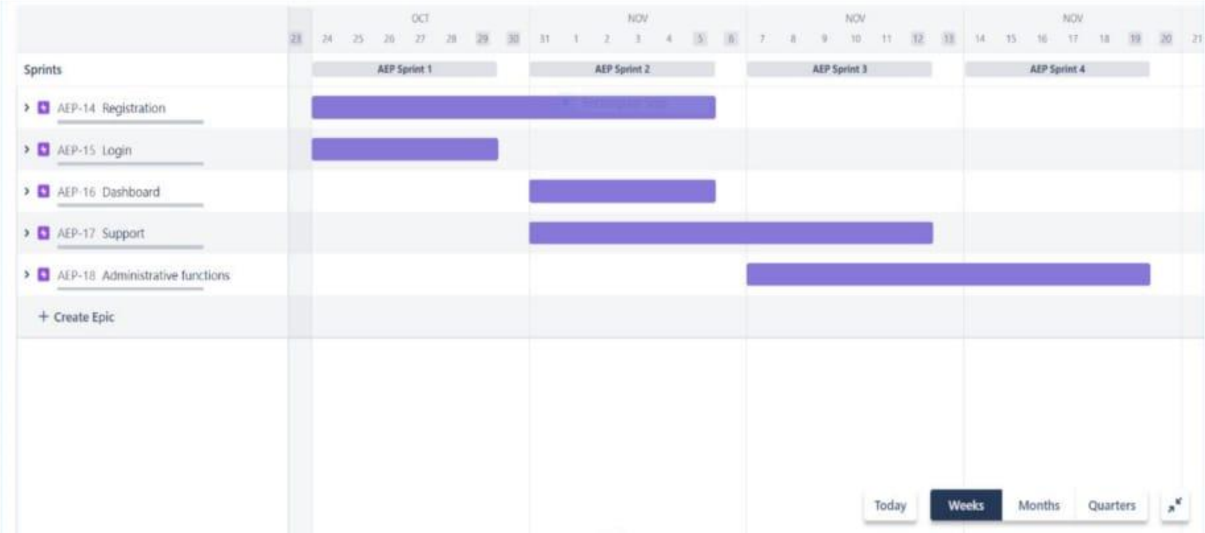
6.2. SPRINT DELIVERY SCHEDULE



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	15	05 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022	15	12 Nov 2022
Sprint-4	15	6 Days	14 Nov 2022	19 Nov 2022	15	19 Nov 2022

Fig. 6.2. Sprint Delivery Schedule

6.3. REPORTS FROM JIRA



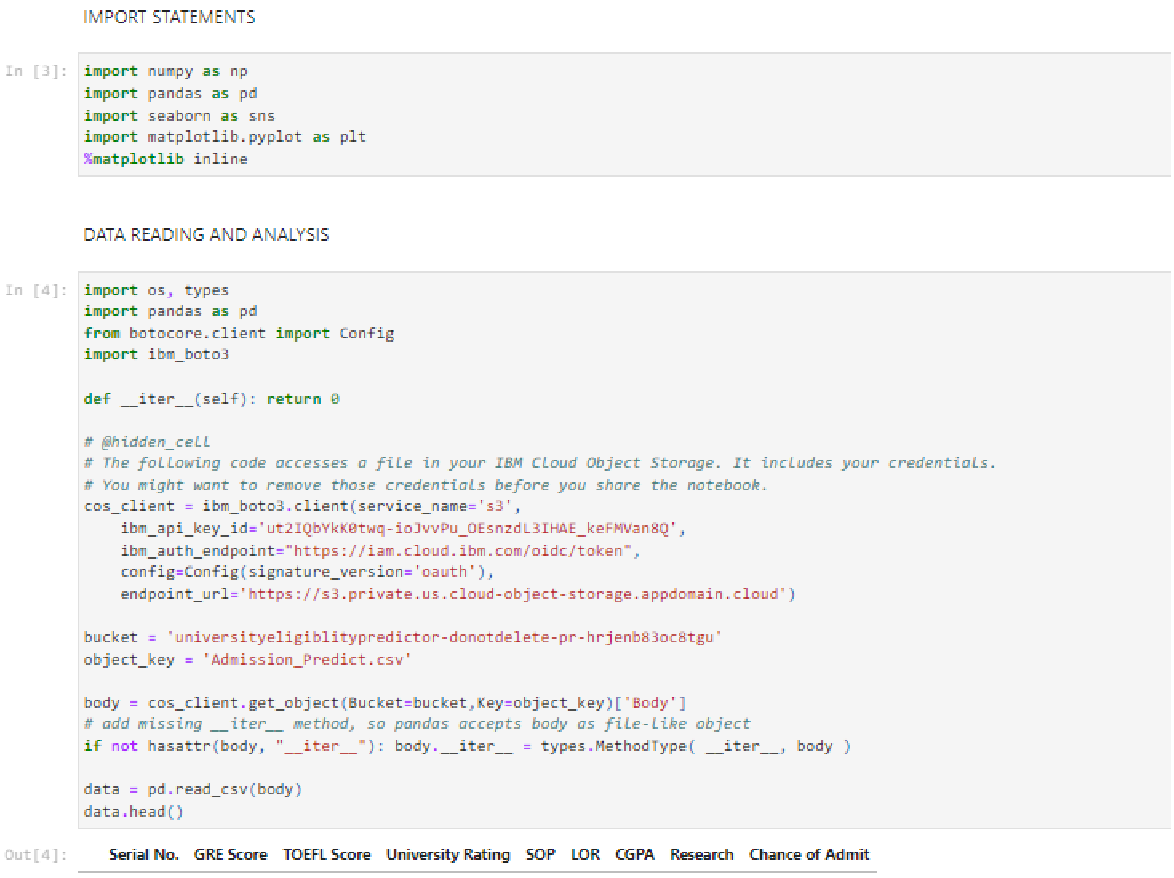
CHAPTER 7

CODING & SOLUTION

7.1. FEATURE 1

Analysed university admission statistics

- Developed tools for matching university (in percentile) using CGPA, GRE (Verbal, Quantitative, Analytical Writing) scores.
- Languages : Python
- Tools/IDE : Anaconda
- Libraries : Numpy ,Pandas, Scikit learn



IMPORT STATEMENTS

In [3]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

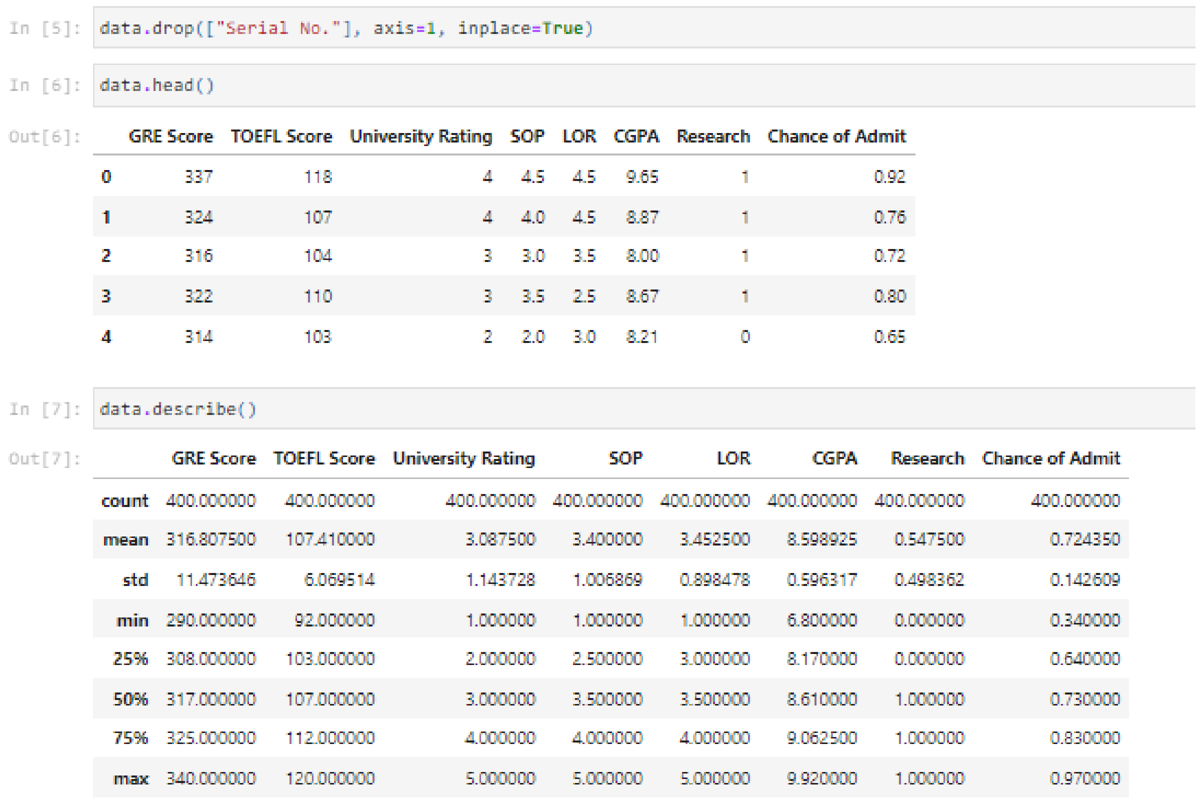
DATA READING AND ANALYSIS

In [4]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='ut2IQbYkK0twq-ioJvvPu_OEsnzdL3IHAE_keFMVan8Q',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'universityeligiblitypredictor-donotdelete-pr-hrjenb83oc8tgu'
object_key = 'Admission_Predict.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

data = pd.read_csv(body)
data.head()
```

Out[4]:

Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit



In [5]:
```
data.drop(["Serial No."], axis=1, inplace=True)
```

In [6]:
```
data.head()
```

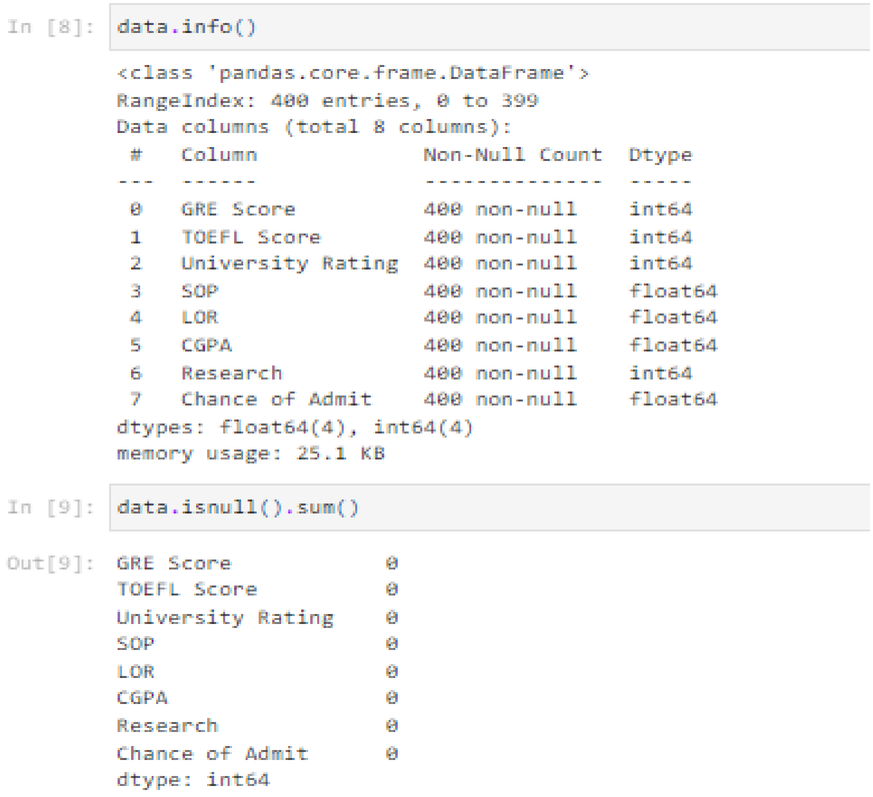
Out[6]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	337	118	4	4.5	4.5	9.65	1	0.92
1	324	107	4	4.0	4.5	8.87	1	0.76
2	316	104	3	3.0	3.5	8.00	1	0.72
3	322	110	3	3.5	2.5	8.67	1	0.80
4	314	103	2	2.0	3.0	8.21	0	0.65

In [7]:
```
data.describe()
```

Out[7]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

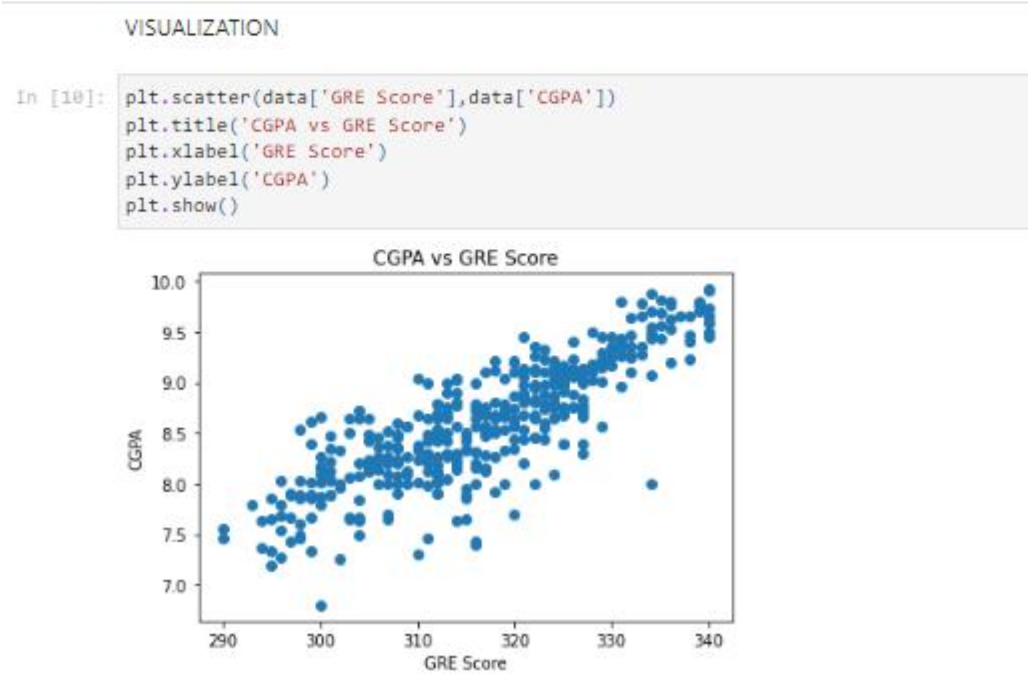


In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   GRE Score          400 non-null    int64  
 1   TOEFL Score        400 non-null    int64  
 2   University Rating  400 non-null    int64  
 3   SOP                400 non-null    float64
 4   LOR                400 non-null    float64
 5   CGPA               400 non-null    float64
 6   Research           400 non-null    int64  
 7   Chance of Admit    400 non-null    float64
dtypes: float64(4), int64(4)
memory usage: 25.1 KB
```

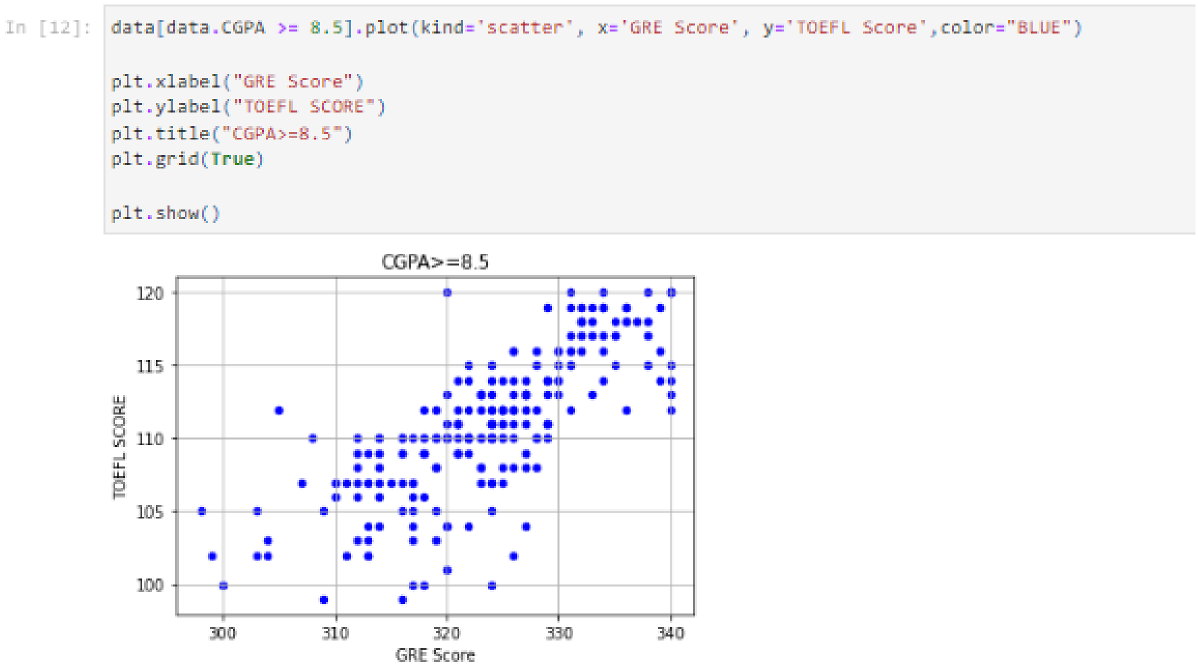
In [9]:
```
data.isnull().sum()
```

Out[9]:
```
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

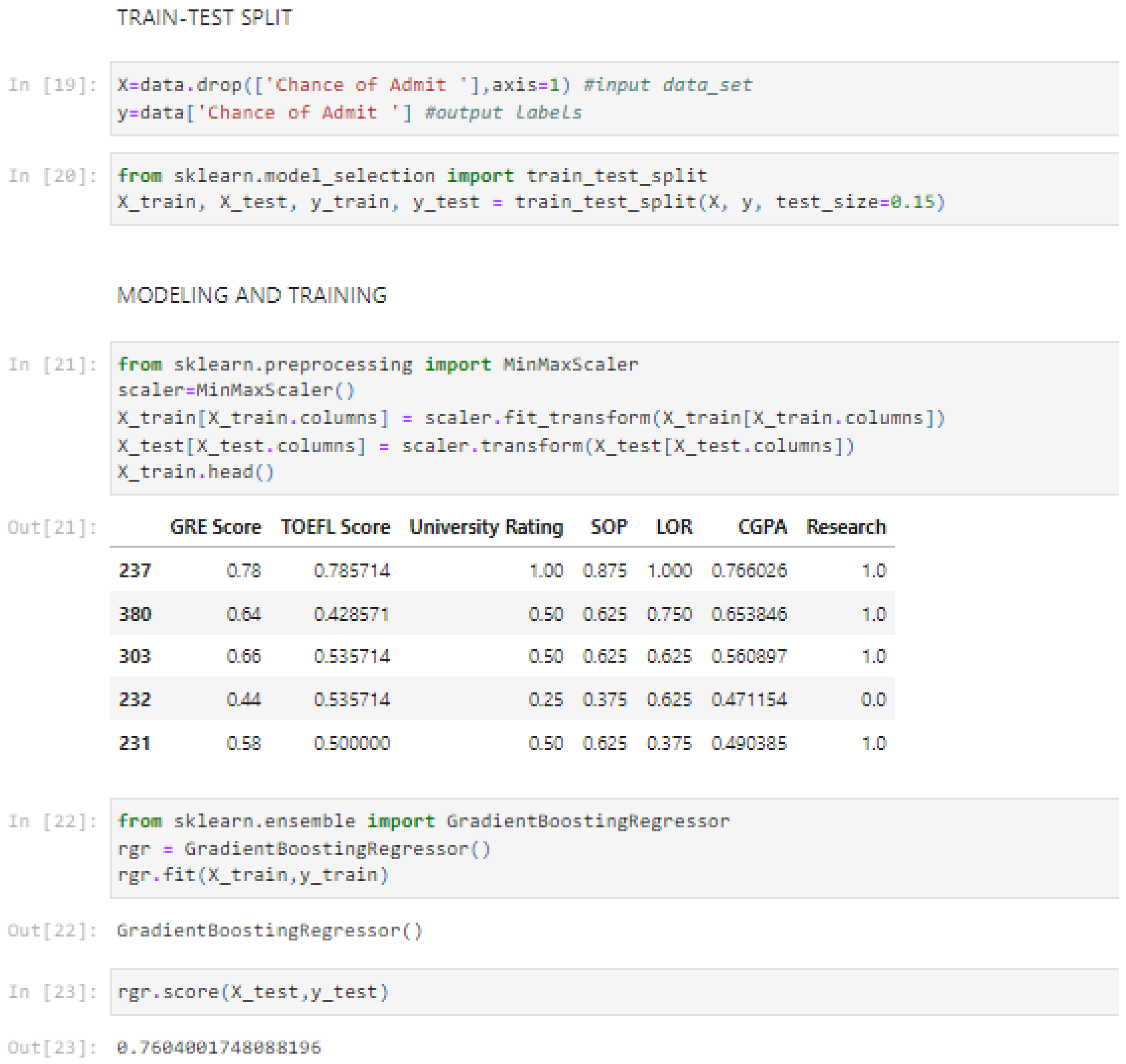


VISUALIZATION

```
In [10]: plt.scatter(data['GRE Score'],data['CGPA'])
         plt.title('CGPA vs GRE Score')
         plt.xlabel('GRE Score')
         plt.ylabel('CGPA')
         plt.show()
```



```
In [12]: data[data.CGPA >= 8.5].plot(kind='scatter', x='GRE Score', y='TOEFL Score',color="BLUE")

         plt.xlabel("GRE Score")
         plt.ylabel("TOEFL SCORE")
         plt.title("CGPA>=8.5")
         plt.grid(True)

         plt.show()
```



TRAIN-TEST SPLIT

```
In [19]: X=data.drop(['Chance of Admit '],axis=1) #input data_set
         y=data['Chance of Admit '] #output labels
```

```
In [20]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15)
```

MODELING AND TRAINING

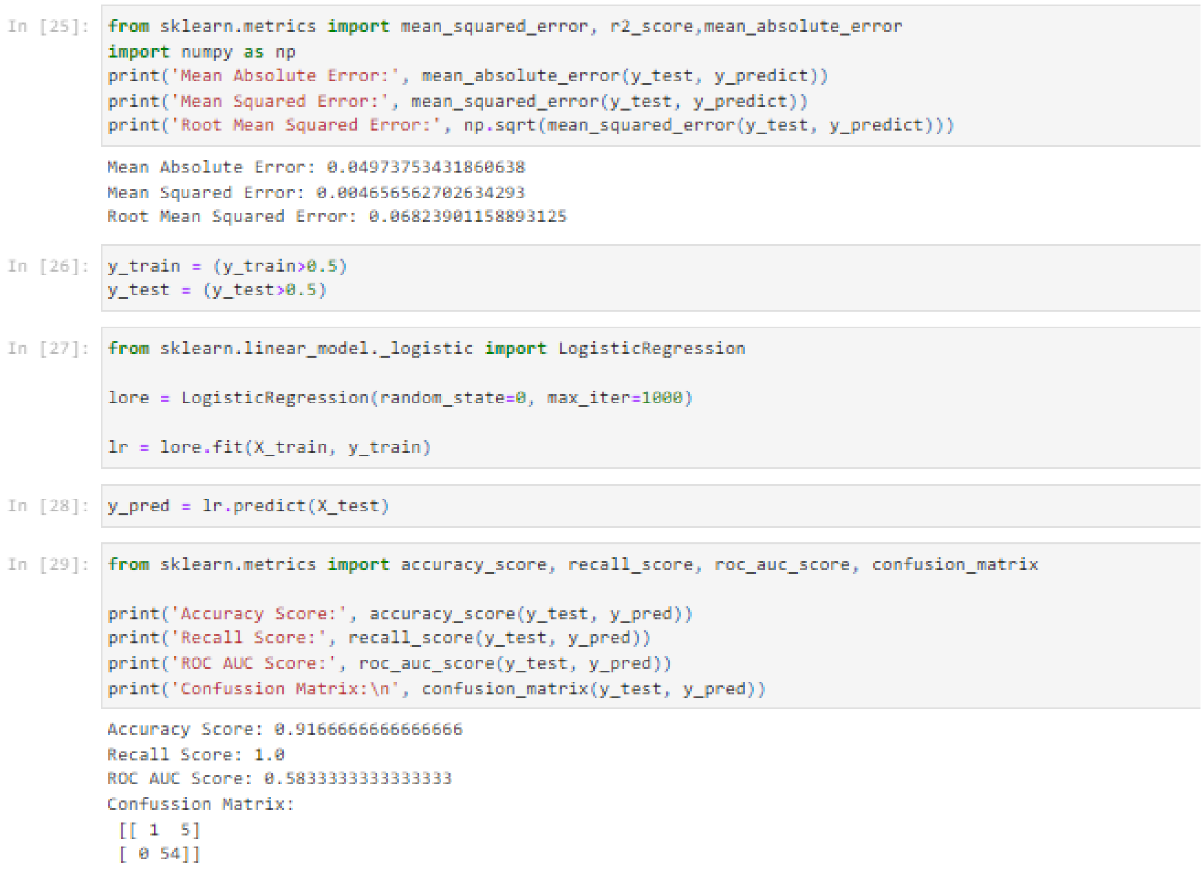
```
In [21]: from sklearn.preprocessing import MinMaxScaler
         scaler=MinMaxScaler()
         X_train[X_train.columns] = scaler.fit_transform(X_train[X_train.columns])
         X_test[X_test.columns] = scaler.transform(X_test[X_test.columns])
         X_train.head()
```

Out[21]:

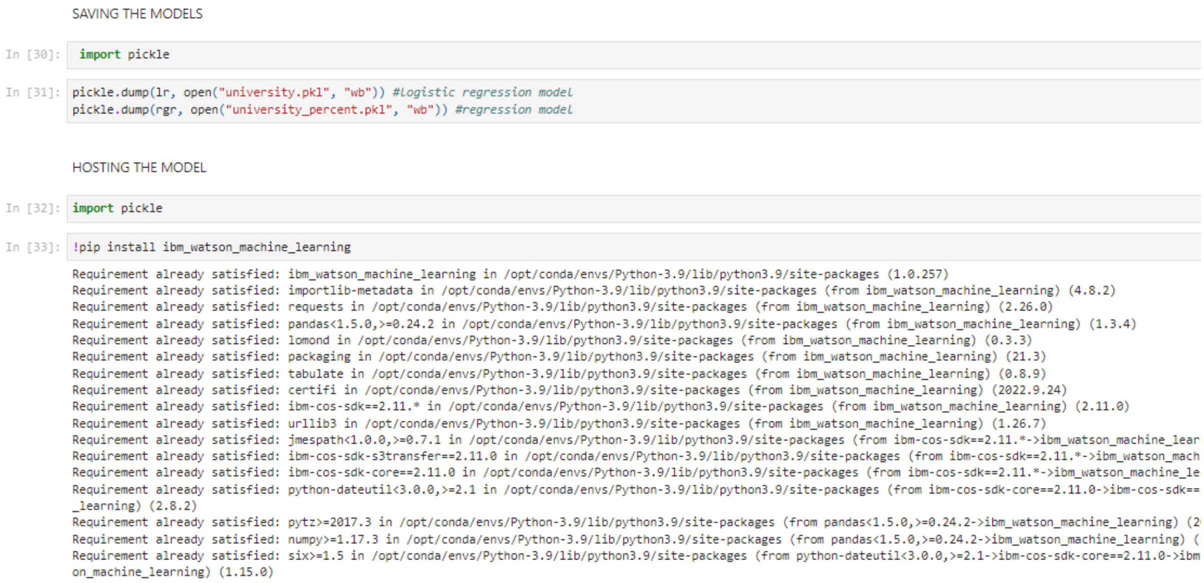
	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
237	0.78	0.785714	1.00	0.875	1.000	0.766026	1.0
380	0.64	0.428571	0.50	0.625	0.750	0.653846	1.0
303	0.66	0.535714	0.50	0.625	0.625	0.560897	1.0
232	0.44	0.535714	0.25	0.375	0.625	0.471154	0.0
231	0.58	0.500000	0.50	0.625	0.375	0.490385	1.0

```
In [22]: from sklearn.ensemble import GradientBoostingRegressor
         rgr = GradientBoostingRegressor()
         rgr.fit(X_train,y_train)

Out[22]: GradientBoostingRegressor()
```

```
In [23]: rgr.score(X_test,y_test)

Out[23]: 0.7604001748088196
```



```
In [25]: from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error
         import numpy as np
         print('Mean Absolute Error:', mean_absolute_error(y_test, y_predict))
         print('Mean Squared Error:', mean_squared_error(y_test, y_predict))
         print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_predict)))

         Mean Absolute Error: 0.04973753431860638
         Mean Squared Error: 0.004656562702634293
         Root Mean Squared Error: 0.06823901158893125

In [26]: y_train = (y_train>0.5)
         y_test = (y_test>0.5)

In [27]: from sklearn.linear_model._logistic import LogisticRegression

         lore = LogisticRegression(random_state=0, max_iter=1000)

         lr = lore.fit(X_train, y_train)

In [28]: y_pred = lr.predict(X_test)

In [29]: from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix

         print('Accuracy Score:', accuracy_score(y_test, y_pred))
         print('Recall Score:', recall_score(y_test, y_pred))
         print('ROC AUC Score:', roc_auc_score(y_test, y_pred))
         print('Confussion Matrix:\n', confusion_matrix(y_test, y_pred))

         Accuracy Score: 0.9166666666666666
         Recall Score: 1.0
         ROC AUC Score: 0.5833333333333333
         Confussion Matrix:
          [[ 1  5]
          [ 0 54]]
```

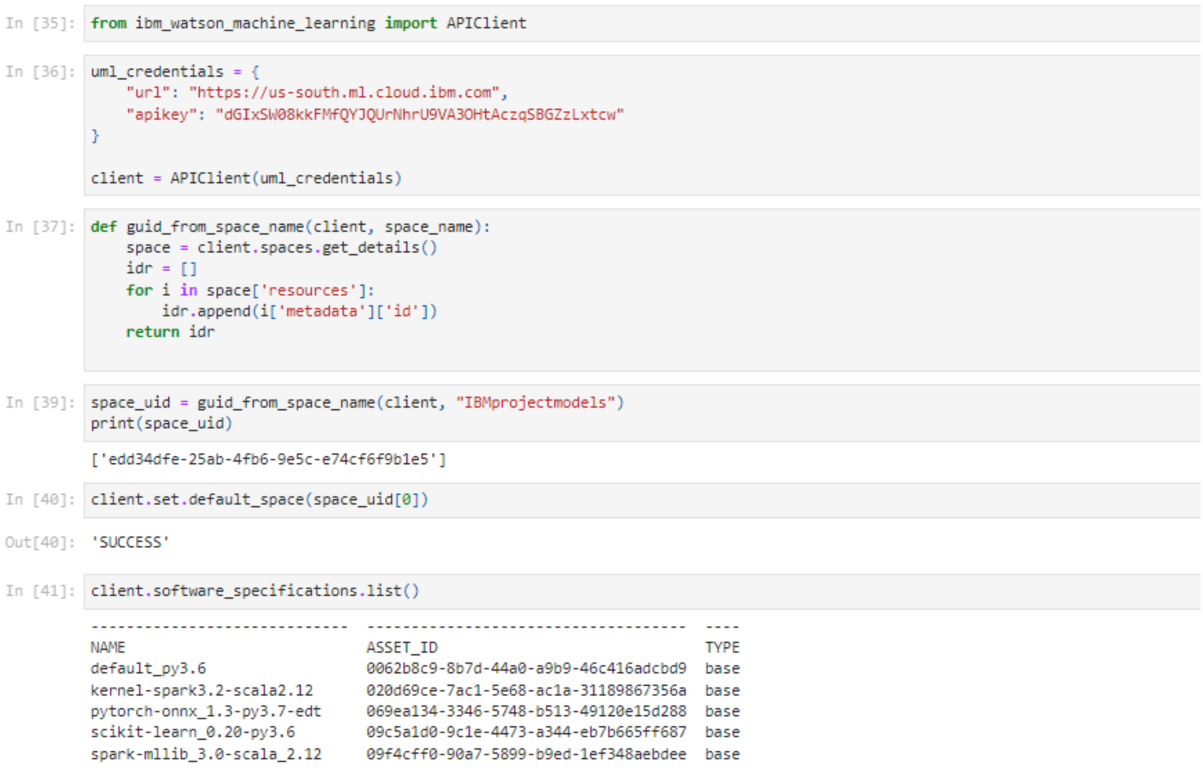


SAVING THE MODELS

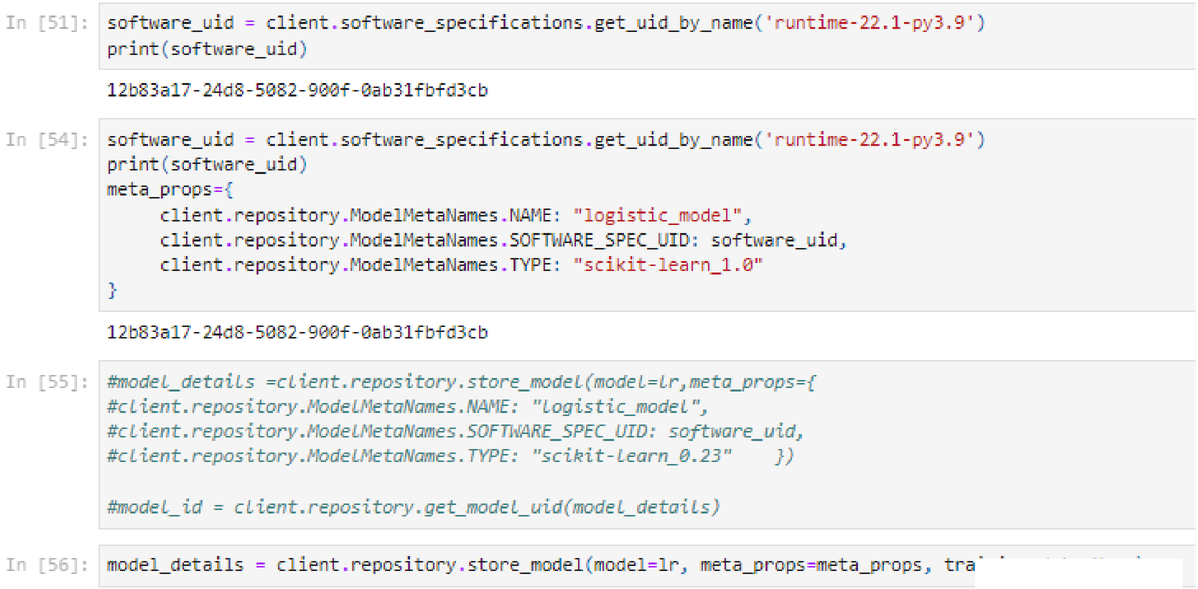
```
In [30]: import pickle

In [31]: pickle.dump(lr, open("university.pkl", "wb")) #logistic regression model
         pickle.dump(rgr, open("university_percent.pkl", "wb")) #regression model
```

HOSTING THE MODEL

```
In [32]: import pickle

In [33]: !pip install ibm_watson_machine_learning

Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_lear
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_mach
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_le
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==
_learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm
on_machine_learning) (1.15.0)
```



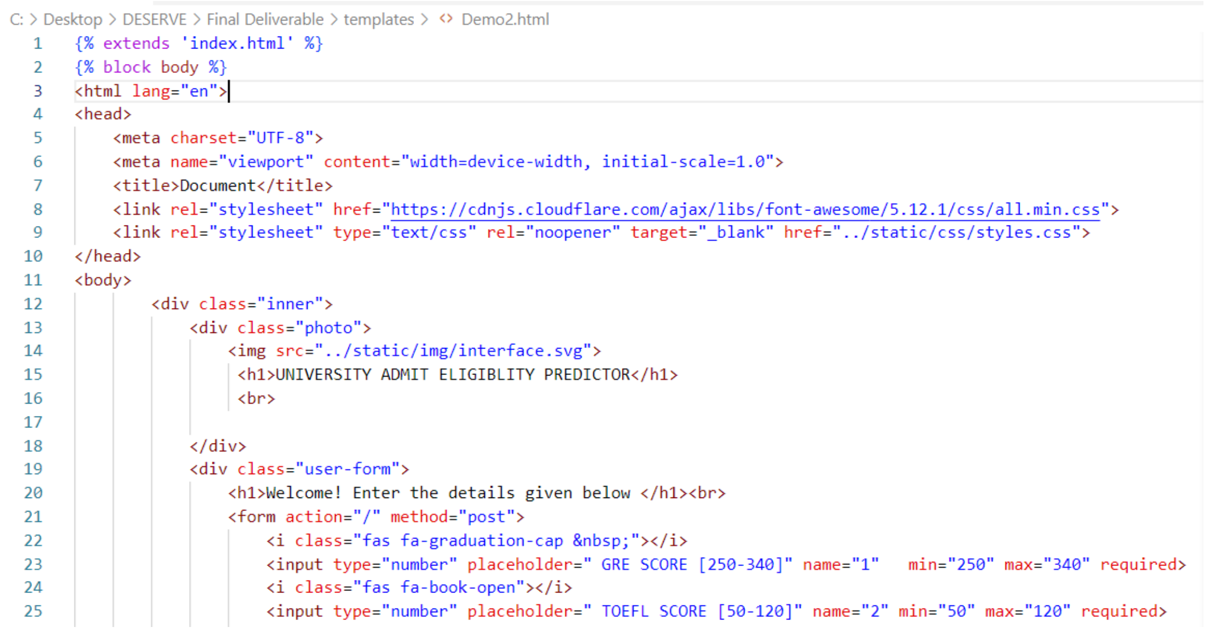
```
In [35]: from ibm_watson_machine_learning import APIClient

In [36]: uml_credentials = {
             "url": "https://us-south.ml.cloud.ibm.com",
             "apikey": "dGIxSW08kkFMfQYJQUrNhrU9VA3OHtAczqSBGZzLxtcw"
         }

         client = APIClient(uml_credentials)

In [37]: def guid_from_space_name(client, space_name):
             space = client.spaces.get_details()
             idr = []
             for i in space['resources']:
                 idr.append(i['metadata']['id'])
             return idr

In [39]: space_uid = guid_from_space_name(client, "IBMprojectmodels")
         print(space_uid)

         ['edd34dfe-25ab-4fb6-9e5c-e74cf6f9b1e5']

In [40]: client.set.default_space(space_uid[0])

Out[40]: 'SUCCESS'

In [41]: client.software_specifications.list()

         -----------------------------  ------------------------------------  ----
         NAME                           ASSET_ID                              TYPE
         default_py3.6                  0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
         kernel-spark3.2-scala2.12      020d69ce-7ac1-5e68-ac1a-31189867356a  base
         pytorch-onnx_1.3-py3.7-edt     069ea134-3346-5748-b513-49120e15d288  base
         scikit-learn_0.20-py3.6        09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
         spark-mllib_3.0-scala_2.12     09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
```



```
In [51]: software_uid = client.software_specifications.get_uid_by_name('runtime-22.1-py3.9')
         print(software_uid)

         12b83a17-24d8-5082-900f-0ab31fbfd3cb

In [54]: software_uid = client.software_specifications.get_uid_by_name('runtime-22.1-py3.9')
         print(software_uid)
         meta_props={
               client.repository.ModelMetaNames.NAME: "logistic_model",
               client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_uid,
               client.repository.ModelMetaNames.TYPE: "scikit-learn_1.0"
         }

         12b83a17-24d8-5082-900f-0ab31fbfd3cb

In [55]: #model_details =client.repository.store_model(model=lr,meta_props={
         #client.repository.ModelMetaNames.NAME: "logistic_model",
         #client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_uid,
         #client.repository.ModelMetaNames.TYPE: "scikit-learn_0.23"    })

         #model_id = client.repository.get_model_uid(model_details)

In [56]: model_details = client.repository.store_model(model=lr, meta_props=meta_props, tra
```

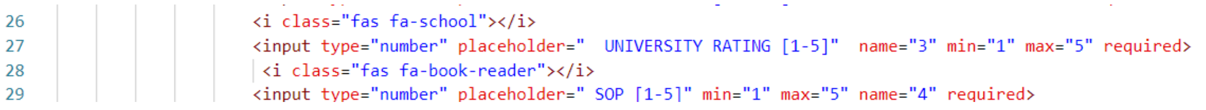
7.2. FEATURE 2

Index.html

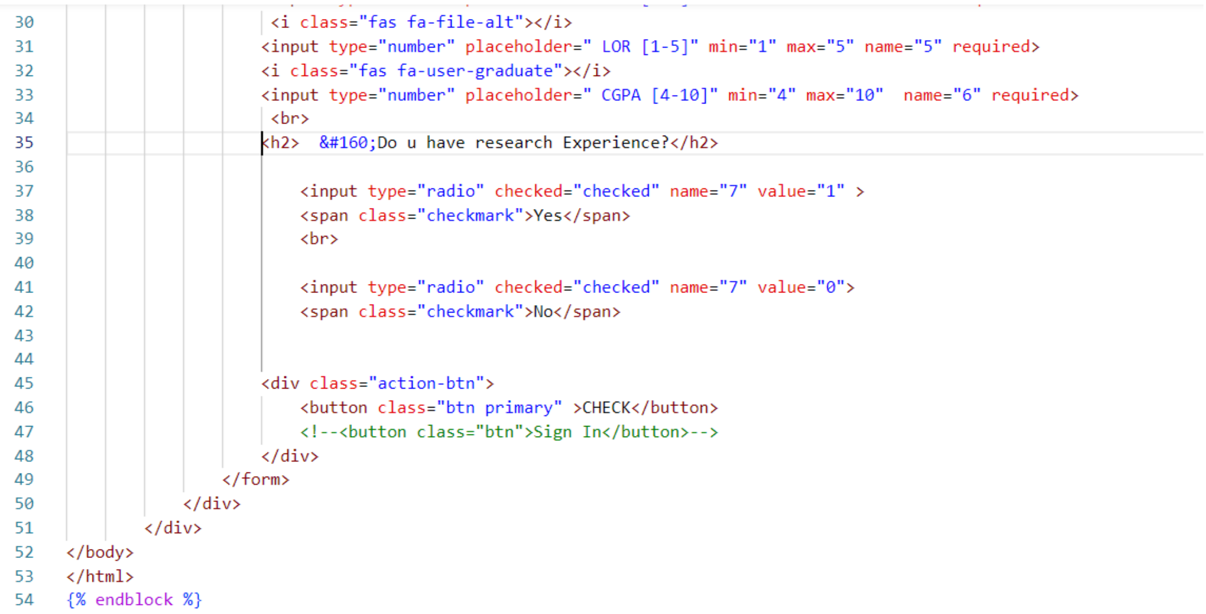


C: > Desktop > DESERVE > Final Deliverable > templates > <> Demo2.html

```
{% extends 'index.html' %}
{% block body %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
</head>
<body>
        <div class="inner">
            <div class="photo">
                <img src="../static/img/interface.svg">
                 <h1>UNIVERSITY ADMIT ELIGIBLITY PREDICTOR</h1>
                 <br>

            </div>
            <div class="user-form">
                <h1>Welcome! Enter the details given below </h1><br>
                <form action="/" method="post">
                    <i class="fas fa-graduation-cap &nbsp;"></i>
                    <input type="number" placeholder=" GRE SCORE [250-340]" name="1"   min="250" max="340" required>
                    <i class="fas fa-book-open"></i>
                    <input type="number" placeholder=" TOEFL SCORE [50-120]" name="2" min="50" max="120" required>
```

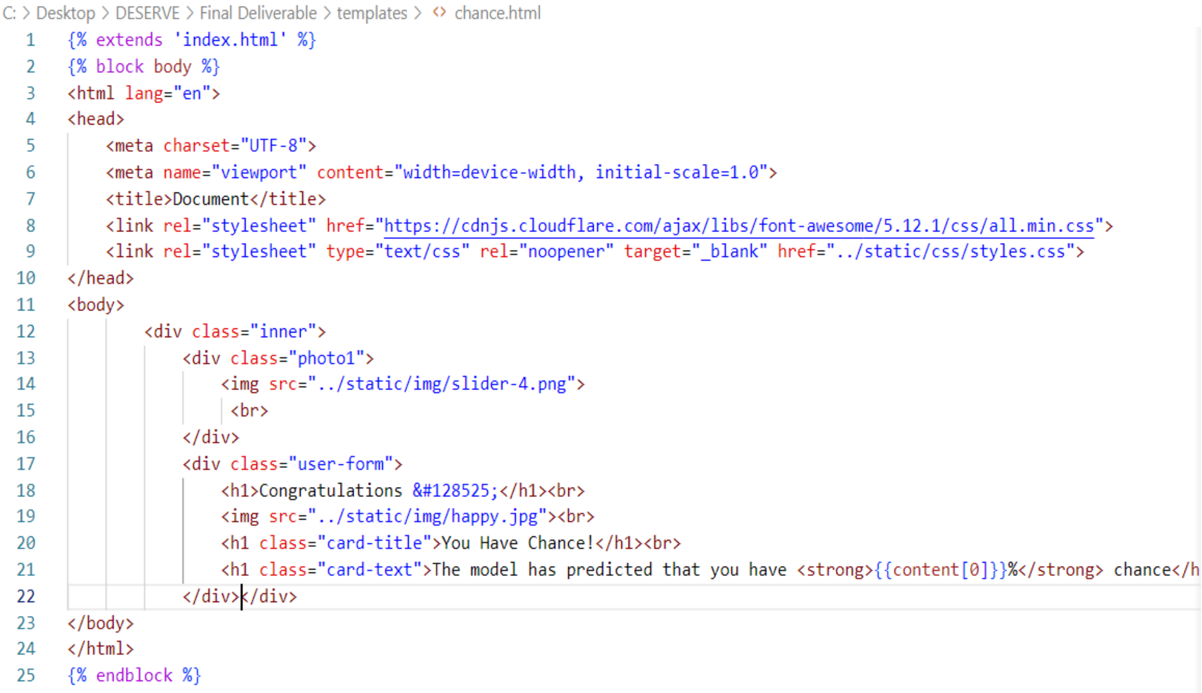


```
                    <i class="fas fa-school"></i>
                    <input type="number" placeholder="  UNIVERSITY RATING [1-5]"  name="3" min="1" max="5" required>
                     <i class="fas fa-book-reader"></i>
                    <input type="number" placeholder=" SOP [1-5]" min="1" max="5" name="4" required>
```



```
                     <i class="fas fa-file-alt"></i>
                    <input type="number" placeholder=" LOR [1-5]" min="1" max="5" name="5" required>
                    <i class="fas fa-user-graduate"></i>
                    <input type="number" placeholder=" CGPA [4-10]" min="4" max="10"  name="6" required>
                     <br>
                    <h2>  &#160;Do u have research Experience?</h2>

                        <input type="radio" checked="checked" name="7" value="1" >
                        <span class="checkmark">Yes</span>
                        <br>

                        <input type="radio" checked="checked" name="7" value="0">
                        <span class="checkmark">No</span>


                    <div class="action-btn">
                        <button class="btn primary" >CHECK</button>
                        <!--<button class="btn">Sign In</button>-->
                    </div>
                </form>
            </div>
        </div>
</body>
</html>
{% endblock %}
```

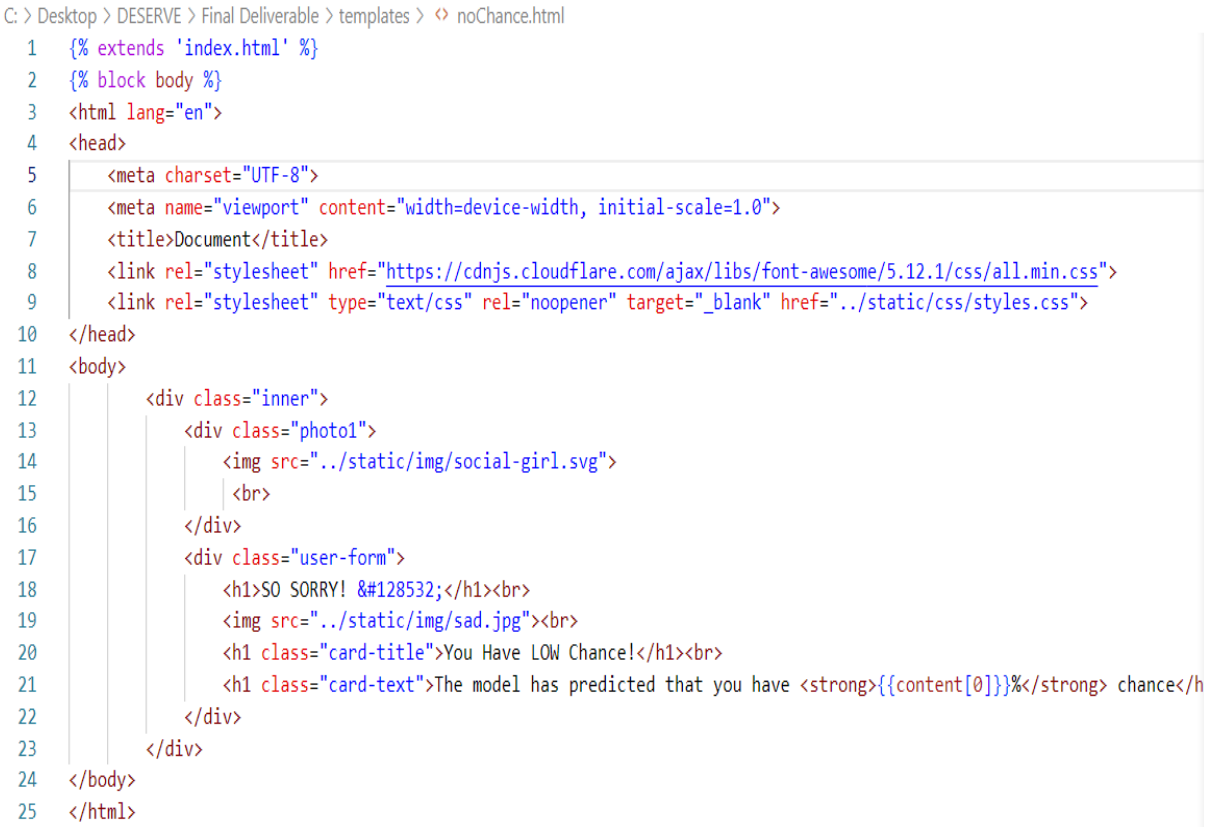
Chance.html



C: > Desktop > DESERVE > Final Deliverable > templates > <> chance.html

```
{% extends 'index.html' %}
{% block body %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
</head>
<body>
        <div class="inner">
            <div class="photo1">
                <img src="../static/img/slider-4.png">
                 <br>
            </div>
            <div class="user-form">
                <h1>Congratulations &#128525;</h1><br>
                <img src="../static/img/happy.jpg"><br>
                <h1 class="card-title">You Have Chance!</h1><br>
                <h1 class="card-text">The model has predicted that you have <strong>{{content[0]}}%</strong> chance</h
            </div></div>
</body>
</html>
{% endblock %}
```

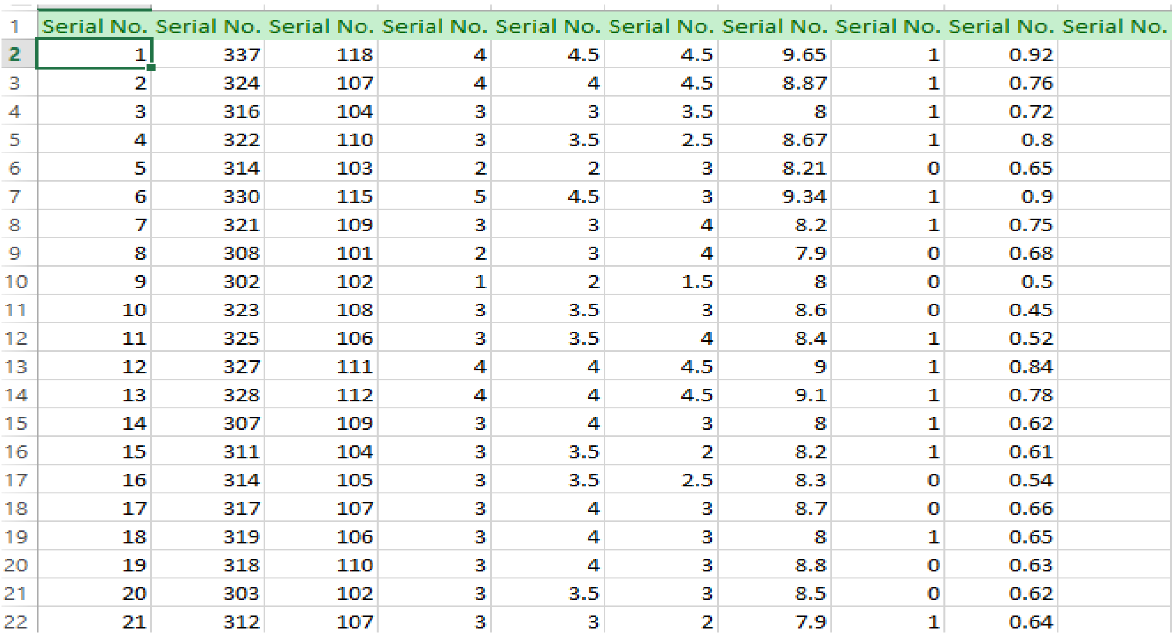
NoChance.html



C: > Desktop > DESERVE > Final Deliverable > templates > <> noChance.html

```
{% extends 'index.html' %}
{% block body %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
</head>
<body>
        <div class="inner">
            <div class="photo1">
                <img src="../static/img/social-girl.svg">
                 <br>
            </div>
            <div class="user-form">
                <h1>SO SORRY! &#128532;</h1><br>
                <img src="../static/img/sad.jpg"><br>
                <h1 class="card-title">You Have LOW Chance!</h1><br>
                <h1 class="card-text">The model has predicted that you have <strong>{{content[0]}}%</strong> chance</h
            </div>
        </div>
</body>
</html>
```

7.3. DATASET

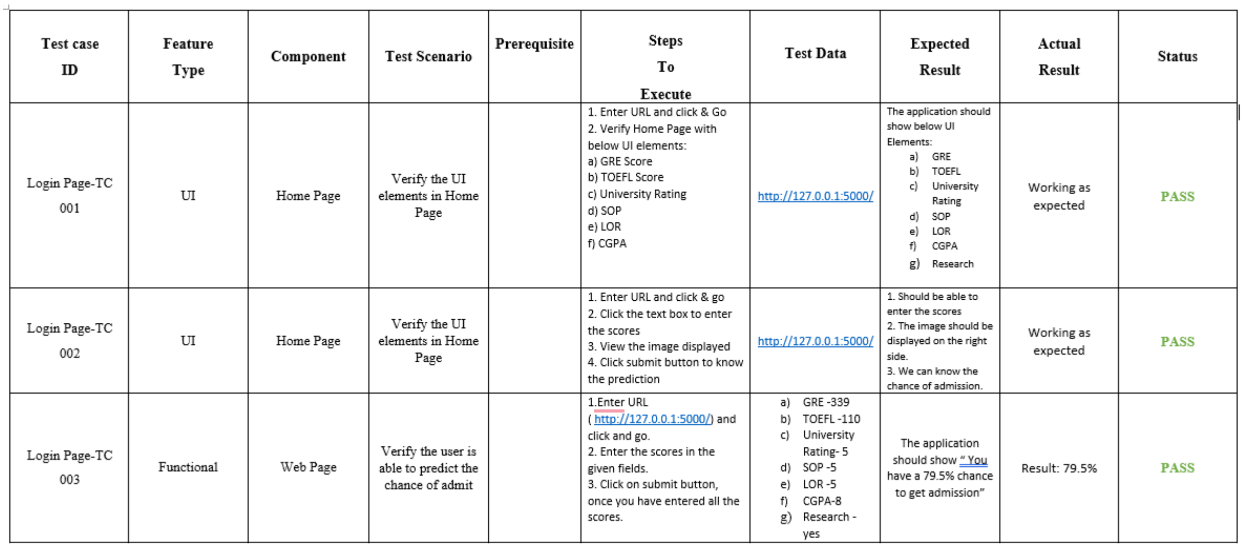


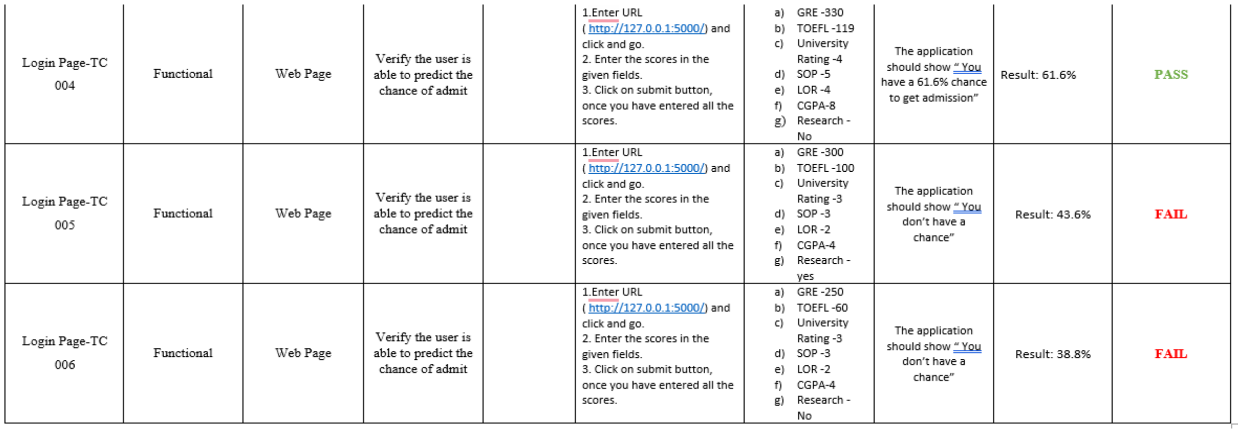
1	Serial No.	Serial No.	Serial No.	Serial No.	Serial No.	Serial No.	Serial No.	Serial No.	Serial No.	Serial No.
2	1	337	118	4	4.5	4.5	9.65	1	0.92	
3	2	324	107	4	4	4.5	8.87	1	0.76	
4	3	316	104	3	3	3.5	8	1	0.72	
5	4	322	110	3	3.5	2.5	8.67	1	0.8	
6	5	314	103	2	2	3	8.21	0	0.65	
7	6	330	115	5	4.5	3	9.34	1	0.9	
8	7	321	109	3	3	4	8.2	1	0.75	
9	8	308	101	2	3	4	7.9	0	0.68	
10	9	302	102	1	2	1.5	8	0	0.5	
11	10	323	108	3	3.5	3	8.6	0	0.45	
12	11	325	106	3	3.5	4	8.4	1	0.52	
13	12	327	111	4	4	4.5	9	1	0.84	
14	13	328	112	4	4	4.5	9.1	1	0.78	
15	14	307	109	3	4	3	8	1	0.62	
16	15	311	104	3	3.5	2	8.2	1	0.61	
17	16	314	105	3	3.5	2.5	8.3	0	0.54	
18	17	317	107	3	4	3	8.7	0	0.66	
19	18	319	106	3	4	3	8	1	0.65	
20	19	318	110	3	4	3	8.8	0	0.63	
21	20	303	102	3	3.5	3	8.5	0	0.62	
22	21	312	107	3	3	2	7.9	1	0.64	

CHAPTER 8

TESTING

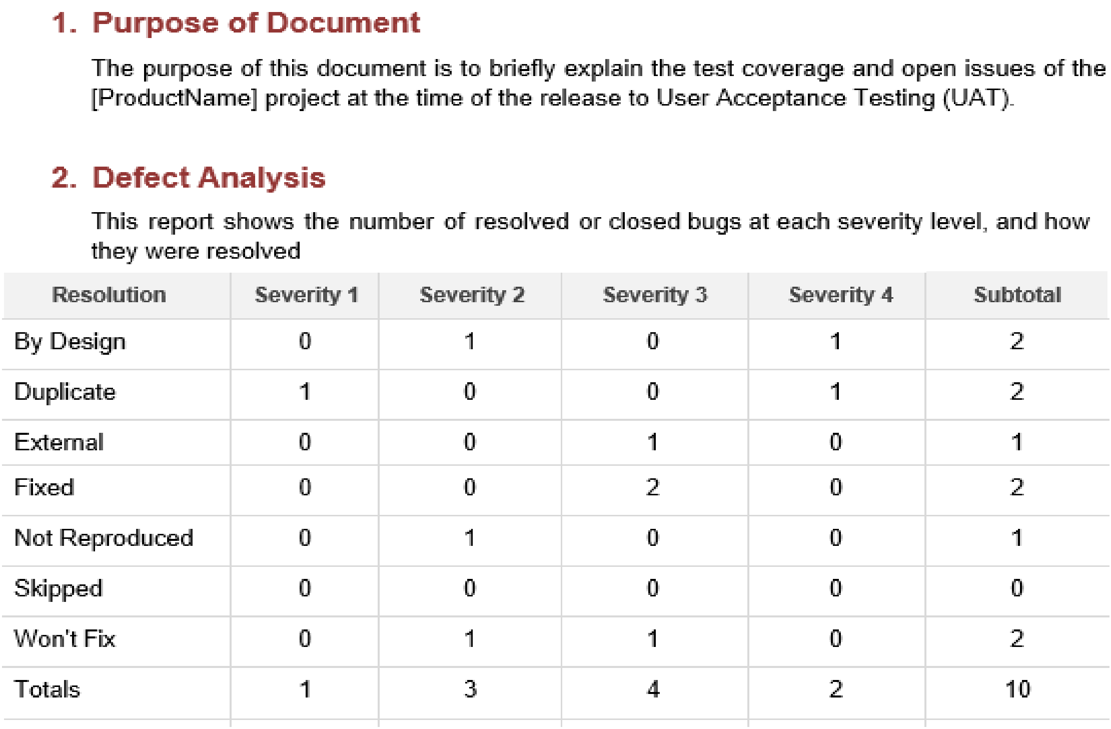
8.1. TESTCASES





Test case ID	Feature Type	Component	Test Scenario	Prerequisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
Login Page-TC 001	UI	Home Page	Verify the UI elements in Home Page		1. Enter URL and click & Go 2. Verify Home Page with below UI elements: a) GRE Score b) TOEFL Score c) University Rating d) SOP e) LOR f) CGPA	http://127.0.0.1:5000/	The application should show below UI Elements: a) GRE b) TOEFL c) University Rating d) SOP e) LOR f) CGPA g) Research	Working as expected	PASS
Login Page-TC 002	UI	Home Page	Verify the UI elements in Home Page		1. Enter URL and click & go 2. Click the text box to enter the scores 3. View the image displayed 4. Click submit button to know the prediction	http://127.0.0.1:5000/	1. Should be able to enter the scores 2. The image should be displayed on the right side. 3. We can know the chance of admission.	Working as expected	PASS
Login Page-TC 003	Functional	Web Page	Verify the user is able to predict the chance of admit		1.Enter URL (http://127.0.0.1:5000/) and click and go. 2. Enter the scores in the given fields. 3. Click on submit button, once you have entered all the scores.	a) GRE -339 b) TOEFL -110 c) University Rating- 5 d) SOP -5 e) LOR -5 f) CGPA-8 g) Research - yes	The application should show " You have a 79.5% chance to get admission"	Result: 79.5%	PASS
Login Page-TC 004	Functional	Web Page	Verify the user is able to predict the chance of admit		1.Enter URL (http://127.0.0.1:5000/) and click and go. 2. Enter the scores in the given fields. 3. Click on submit button, once you have entered all the scores.	a) GRE -330 b) TOEFL -119 c) University Rating -4 d) SOP -5 e) LOR -4 f) CGPA-8 g) Research - No	The application should show " You have a 61.6% chance to get admission"	Result: 61.6%	PASS
Login Page-TC 005	Functional	Web Page	Verify the user is able to predict the chance of admit		1.Enter URL (http://127.0.0.1:5000/) and click and go. 2. Enter the scores in the given fields. 3. Click on submit button, once you have entered all the scores.	a) GRE -300 b) TOEFL -100 c) University Rating -3 d) SOP -3 e) LOR -2 f) CGPA-4 g) Research - yes	The application should show " You don't have a chance"	Result: 43.6%	FAIL
Login Page-TC 006	Functional	Web Page	Verify the user is able to predict the chance of admit		1.Enter URL (http://127.0.0.1:5000/) and click and go. 2. Enter the scores in the given fields. 3. Click on submit button, once you have entered all the scores.	a) GRE -250 b) TOEFL -60 c) University Rating -3 d) SOP -3 e) LOR -2 f) CGPA-4 g) Research - No	The application should show " You don't have a chance"	Result: 38.8%	FAIL

8.2. USER ACCEPTANCE TESTING



1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	1	0	1	2
Duplicate	1	0	0	1	2
External	0	0	1	0	1
Fixed	0	0	2	0	2
Not Reproduced	0	1	0	0	1
Skipped	0	0	0	0	0
Won't Fix	0	1	1	0	2
Totals	1	3	4	2	10

3. Test Case Analysis

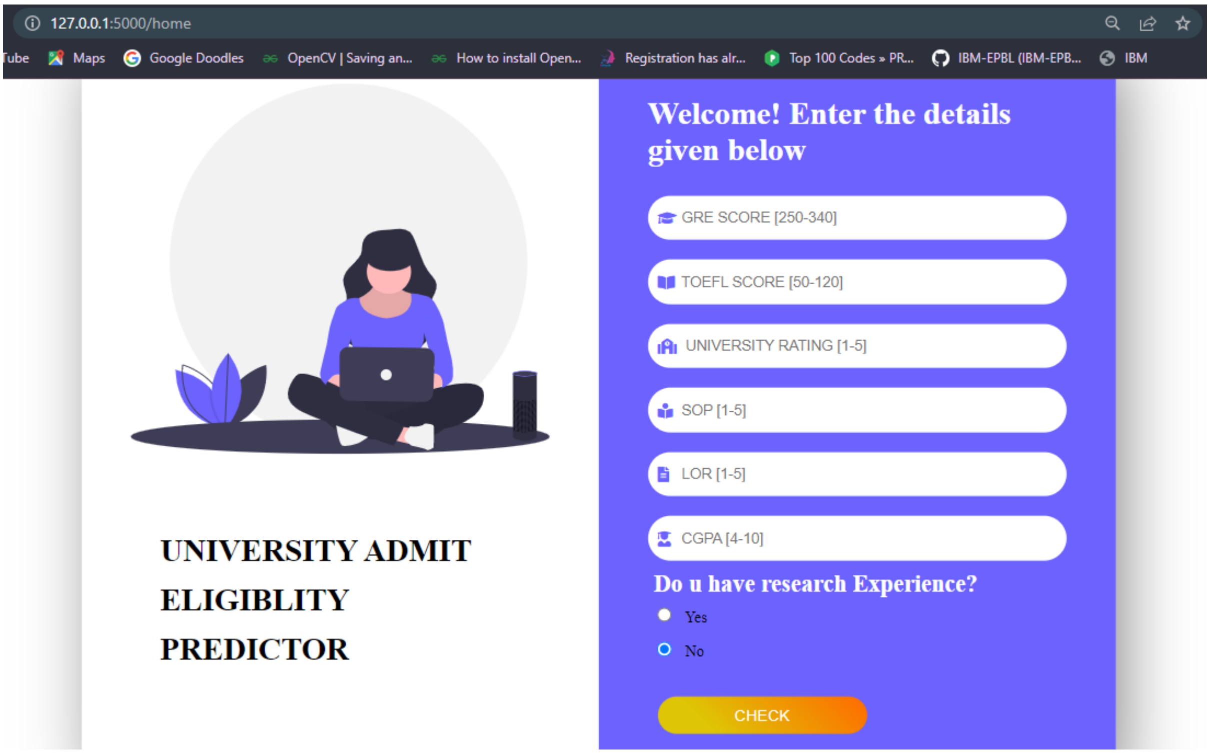
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
View homepage	12	0	3	9
Enter the scores	20	0	5	15
Click submit button	3	0	0	3
Image displayed	10	0	3	7
Selecting from drop down	5	0	0	5
Final report output	35	0	10	25
Version control	4	0	2	2

CHAPTER 9

RESULTS

9.1. PERFORMANCE METRICS



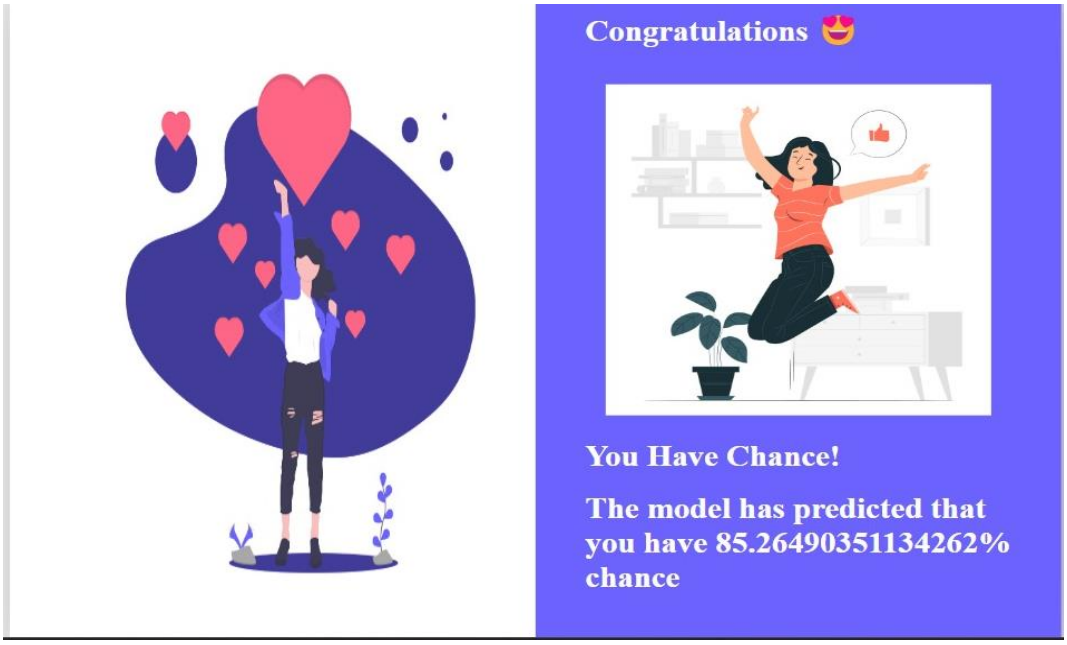
UNIVERSITY ADMIT ELIGIBLITY PREDICTOR

Welcome! Enter the details given below

Do u have research Experience?

○ Yes

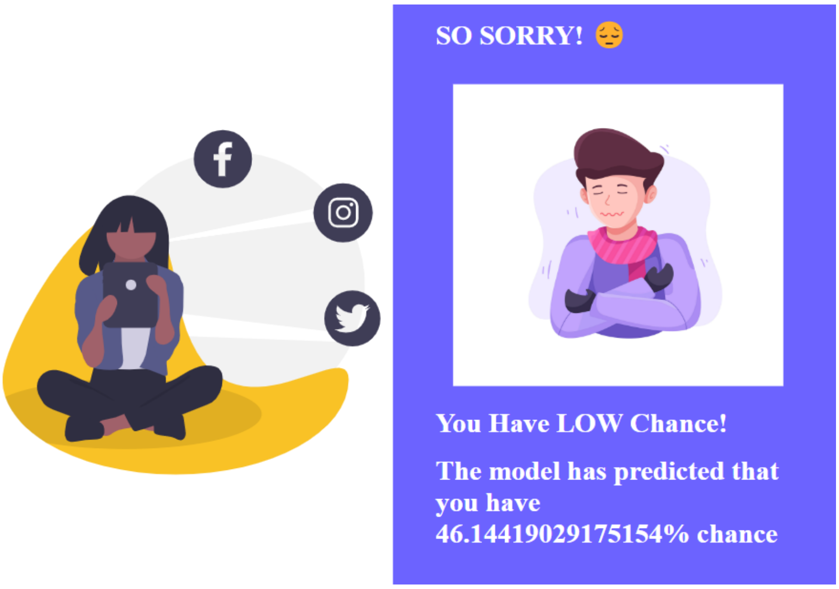
● No



Congratulations 😍

You Have Chance!

The model has predicted that you have 85.26490351134262% chance



SO SORRY! 😔

You Have LOW Chance!

The model has predicted that you have 46.14419029175154% chance

CHAPTER 10

10.1. ADVANTAGES

1. **Reach to geographically scattered student:** One of the important objectives of the admission system is to communicate with all the students scattered geographically.

2. **Reducing time in activities:** Reduce the time taken to process the applications of students, admit a student, conduct the online examination, verify student marks, and send call letters to selected students.

3**. Centralized data handling:** Transfer the data smoothly to all the departments involved and handle the data centralized way.

4. **Paperless admission with reduced manpower:** Reduce the manpower needed to perform all the admission and administration tasks by reducing the paper works Cost cutting. Reduce the cost involved in the admission process.

5**. Operational efficiency:** Improve operational efficiency by improving the quality of the process.

10.2. DISADVANTAGES

1. Required active internet connection.

2. System will provide inaccurate results if data entered incorrectly.

3. Other factors such as changes in policies by the university or by the country can also affect chances of admissions in a way that is beyond the scope of this project.

4. Admissions also depend on the individual university’s policy regarding the intake of foreign students and is not modelled by our system.

11. CONCLUSION

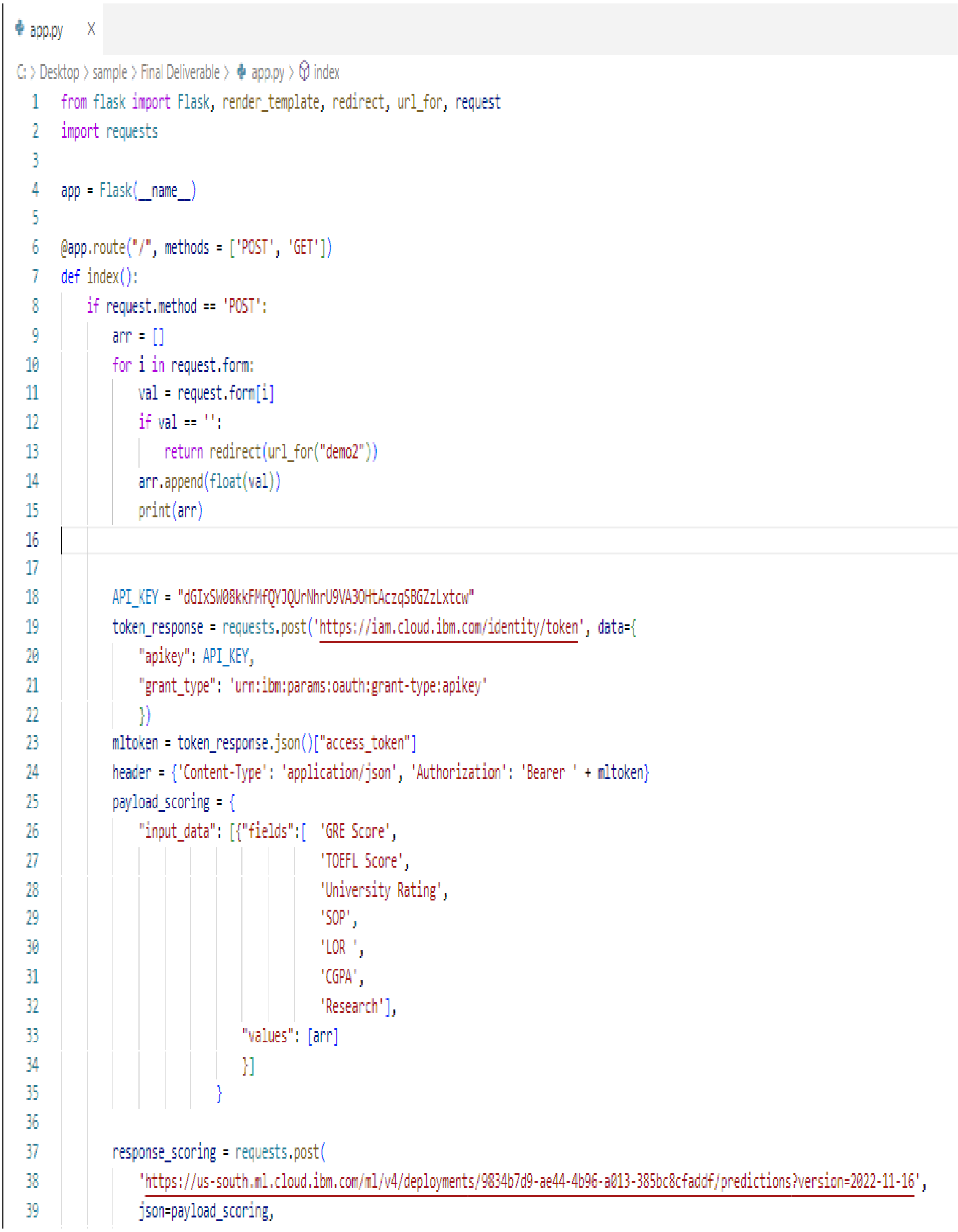
Every year millions of students apply to universities to begin their educational life. Most of them don’t have proper resources, or prior knowledge and are not cautious, which in turn creates a lot of problems as applying to the wrong university/college, which further wastes their time, money, and energy. With the help of our project, we have tried to help out such students who are finding difficulty finding the right university for them. It is very important that a candidate should apply to colleges that he/she has a good chance of getting into, instead of applying to colleges that they may never get into. This will help in the reduction of costs as students will be applying to only those universities that they are highly likely to get into. Our prepared models work to a satisfactory level of accuracy and may be of great assistance to such people. This is a project with good future scope, especially for students of our age group who want to pursue their higher education in their dream college/University.

13. FUTURE WORK

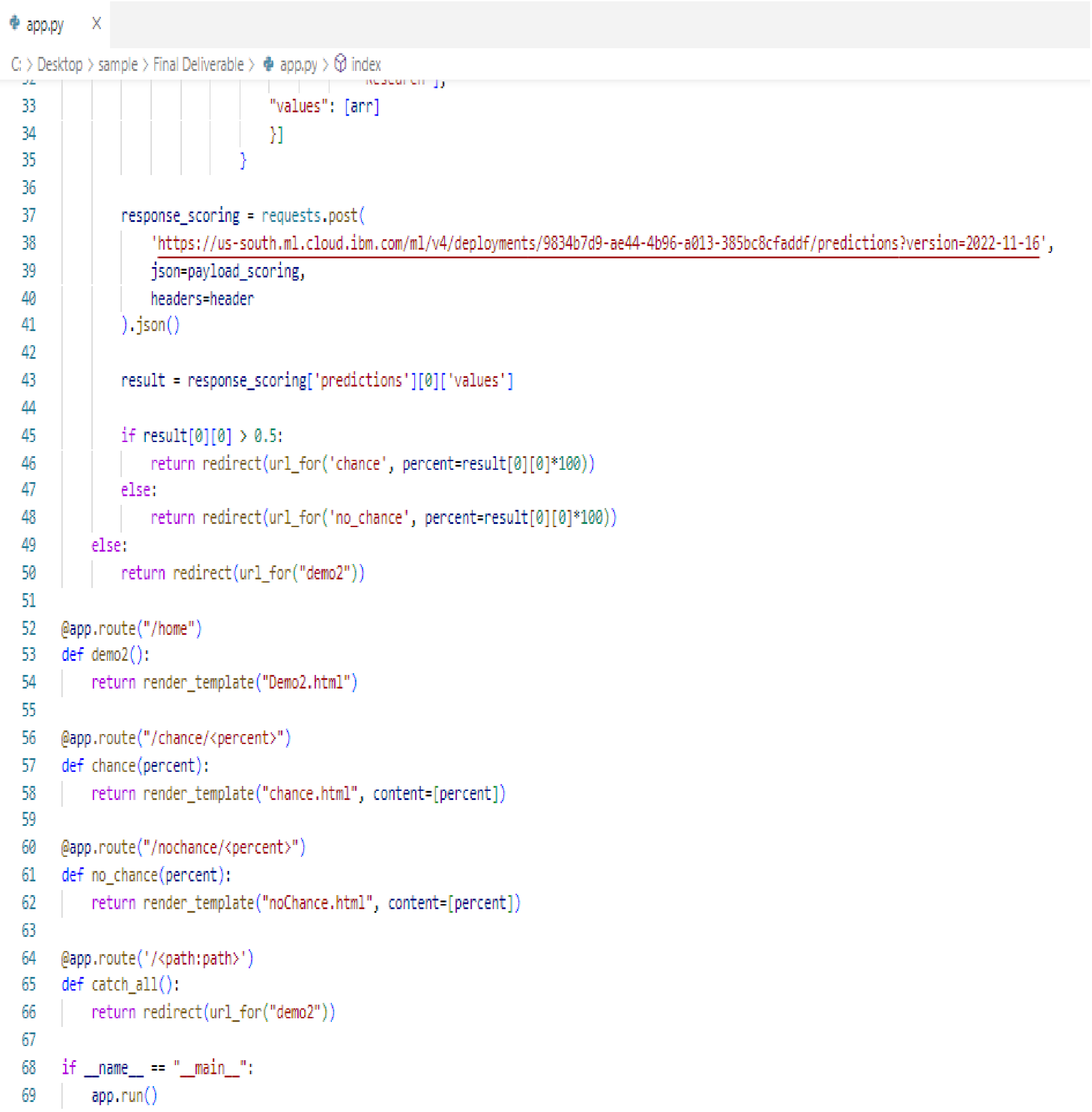
From the proposed work we are able to identify only the chance to get a seat and we are not able to identify which university we are obtaining. So, in the future, we can develop a representation, which gives us a list of universities to which we can obtain admission.

14. APPENDIX

14.1. Source Code



```python
from flask import Flask, render_template, redirect, url_for, request
import requests

app = Flask(__name__)

@app.route("/", methods = ['POST', 'GET'])
def index():
    if request.method == 'POST':
        arr = []
        for i in request.form:
            val = request.form[i]
            if val == '':
                return redirect(url_for("demo2"))
            arr.append(float(val))
            print(arr)


        API_KEY = "dGIxSW08kkFMfQYJQUrNhrU9VA3OHtAczqSBGZzLxtcw"
        token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={
            "apikey": API_KEY,
            "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
            })
        mltoken = token_response.json()["access_token"]
        header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
        payload_scoring = {
            "input_data": [{"fields":[  'GRE Score',
                                        'TOEFL Score',
                                        'University Rating',
                                        'SOP',
                                        'LOR ',
                                        'CGPA',
                                        'Research'],
                            "values": [arr]
                            }]
                        }

        response_scoring = requests.post(
            'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9834b7d9-ae44-4b96-a013-385bc8cfaddf/predictions?version=2022-11-16',
            json=payload_scoring,
```



```
                            "values": [arr]
                            }]
                        }

        response_scoring = requests.post(
            'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9834b7d9-ae44-4b96-a013-385bc8cfaddf/predictions?version=2022-11-16',
            json=payload_scoring,
            headers=header
        ).json()

        result = response_scoring['predictions'][0]['values']

        if result[0][0] > 0.5:
            return redirect(url_for('chance', percent=result[0][0]*100))
        else:
            return redirect(url_for('no_chance', percent=result[0][0]*100))
    else:
        return redirect(url_for("demo2"))

@app.route("/home")
def demo2():
    return render_template("Demo2.html")

@app.route("/chance/<percent>")
def chance(percent):
    return render_template("chance.html", content=[percent])

@app.route("/nochance/<percent>")
def no_chance(percent):
    return render_template("noChance.html", content=[percent])

@app.route('/<path:path>')
def catch_all():
    return redirect(url_for("demo2"))

if __name__ == "__main__":
    app.run()
```

14.2. GitHub and Project Demo Link

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-2733-1658481948

Project Demo link: https://www.youtube.com/watch?v=Nb4oQv5Or-k