PROJECT REPORT

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

submitted by

PNT2022TMID52695

Harshavardhan P R	-	CITC1907014
Roopa V	-	CITC1907036
Shirin K	-	CITC1907045
Thenmozhi S	-	CITC1907055

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Machine learning and deep learning play an important role in computer technology and artificial intelligence. With the use of deep learning and machine learning, human effort can be reduced in recognizing, learning, predictions and in many more areas.

This University Admit Eligibility Predictor is web based application in which students can register with their personal as well as marks details for prediction the admission in colleges and the administrator can allot the seats for the students. Administrator can add the college details and the batch details. Using this software, the entrance seat allotment became easier and can be implemented using system. The main advantage of the project is the computerization of the entrance seat allotment process. Administrator has the power for the allotment. He can add the allotted seats into a file and the details are saved into the system. The total time for the entrance allotment became lesser and the allotment process became faster.

1.2 PURPOSE

Students are often worried about their chances of admission to University. The aim of this project is to help students in short listing universities with their profiles. The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

Students are sometimes concerned about their chances of being admitted to university. This project's goal is to assist students in shortlisting institutions based on their profiles. The predicted outcome gives them a good indication of their prospects of admission to specific universities. Students who are preparing or will be preparing should benefit from this analysis in regards to having a better understanding.

2.2 REFERENCES

Title: Predicting Undergraduate Admission(2020)

Author: Md. Protikuzzaman, Mrinal Kanti Baowaly, Maloy Kumar Devnath, Bikash Chandra Singh

The university admission tests find the applicant's ability to admit to the desired university. Nowadays, there is a huge competition in the university admission tests. The failure in the admission tests makes an examinee depressed. This paper proposes a method that predicts undergraduate admission in universities. It can help students to improve their preparation to get a chance at their desired university. Many factors are responsible for the failure or success in an admission test. Educational data mining helps us to analyze and extract information from these factors. Here, the authors apply three machine learning algorithms XGBoost, LightGBM, and GBM on a collected dataset to estimate the probability of getting admission to the university after attending or before attending the admission test. They also evaluate and compare the performance levels of these three algorithms based on two different evaluation metrics – accuracy and F1 score. Furthermore, the authors explore the important factors which influence predicting undergraduate admission.

Title: University Admissions Predictor (2021)

Author: Aanchal Thakur

This paper is a Requirements Specification Document for a new web-based University Admissions Predictor – UNIPREDICT. Unipredict is an 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.

Title: Predicting Student University Admission Using Logistic Regression (2020)
Author: Sharan Kumar Paratala Rajagopal

The primary purpose is to discuss the prediction of student admission to university based on numerous factors and using logistic regression. Many prospective students apply for Master's programs. The admission decision depends on criteria within the particular college or degree program. The independent variables in this study will be measured statistically to predict graduate school admission. Exploration and data analysis, if successful, would allow predictive models to allow better prioritization of the applicants screening process to Master's degree programme which in turn provides the admission to the right candidates.

Title: College Admission Prediction using Ensemble Machine Learning Models (2021)
Author: Vandit Manish Jain, Rihaan Satia

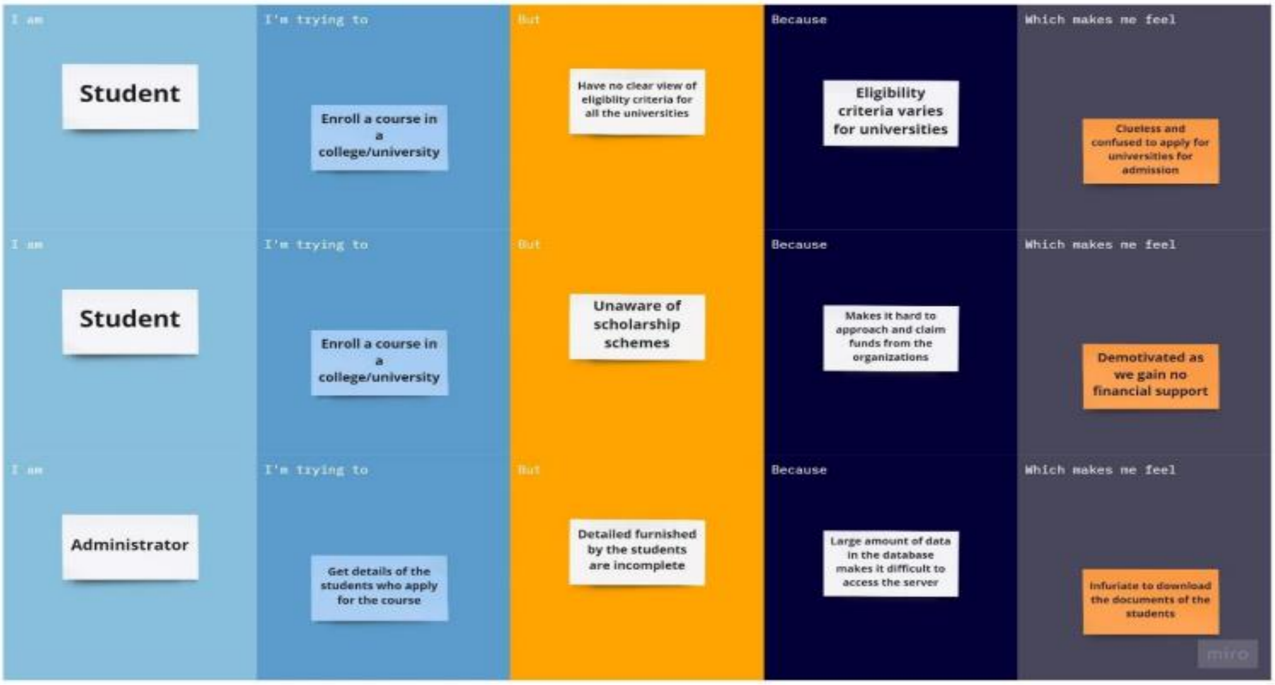
This paper aims to build a model that can help students to pick the right universities based on their profiles. We can judge across a wide variety of domainsthatincludeMS (international), M.Tech (India) and MBA (India and International). For the accurate predictions we plan on training a machine learning model in order to provide results. The dataset contains information on the student profile and the university details with a field detailing if the admission was positive or not. Various algorithms have been used i.e. Ensemble Machine Learning and the predictions have been compared using key performance indicators(KPIs). Themodel performing the best is then used to evaluate the dependent variable i.e. The chances of admit to a university. The chances of admit variable is a variable ranging from 0 to 1 which equates to the predicted probability of successful acceptance to a university. We also aim to create a portal which filters and then provides a list of universities that fall into the profile's acceptance range

Title: Prediction of Admission Process for Gradational Studies using Al Algorithm(2020)
Author: Saurabh Singhal, Ashish Sharma

The purpose of this work is to compare different Machine Learning algorithms and find out which algorithm is giving an accurate result for the available dataset. The algorithms we are going to use are Multi Linear Regression, Polynomial Regression and Random Forest. The input for these algorithms is GRE score, TOFEL score and CGPA of candidate. By using dataset we are going to train the representation and finally the output we are obtaining is the percentage of chance to get seat in reputed university.

2.3 PROBLEM STATEMENT DEFINITION

A problem statement allows understanding your customer's point of view. The Customer Problem Statement helps you focus on what matters to create experiences people will love.

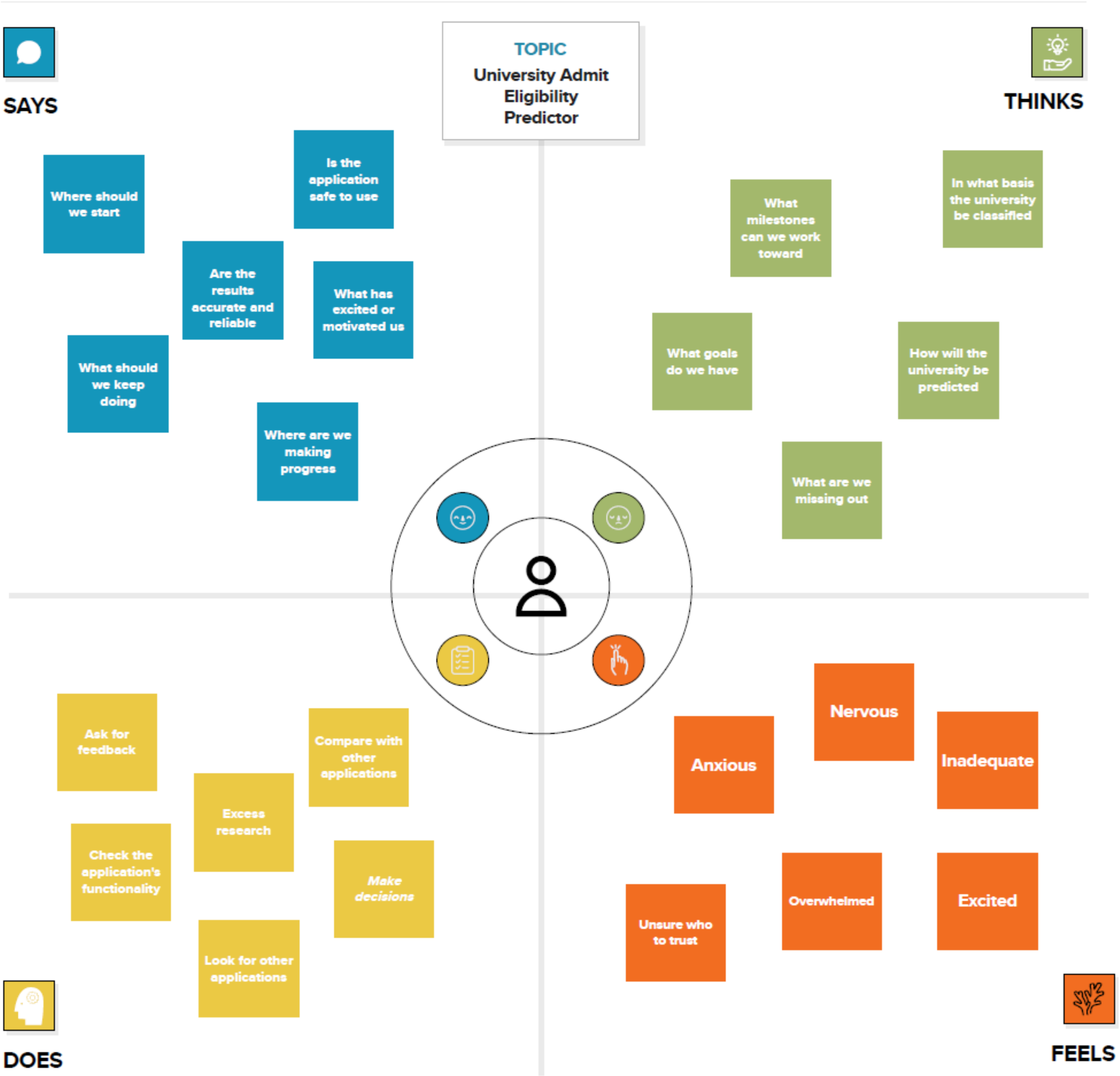


I am	I'm trying to	But	Because	Which makes me feel
Student	Enroll a course in a college/university	Have no clear view of eligiblity criteria for all the universities	Eligibility criteria varies for universities	Clueless and confused to apply for universities for admission
Student	Enroll a course in a college/university	Unaware of scholarship schemes	Makes it hard to approach and claim funds from the organizations	Demotivated as we gain no financial support
Administrator	Get details of the students who apply for the course	Detailed furnished by the students are incomplete	Large amount of data in the database makes it difficult to access the server	Infuriate to download the documents of the students

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



TOPIC

University Admit Eligibility Predictor

SAYS

Where should we start

Is the application safe to use

Are the results accurate and reliable

What has excited or motivated us

What should we keep doing

Where are we making progress

THINKS

What milestones can we work toward

In what basis the university be classified

What goals do we have

How will the university be predicted

What are we missing out

DOES

Ask for feedback

Compare with other applications

Excess research

Check the application's functionality

Make decisions

Look for other applications

FEELS

Nervous

Anxious

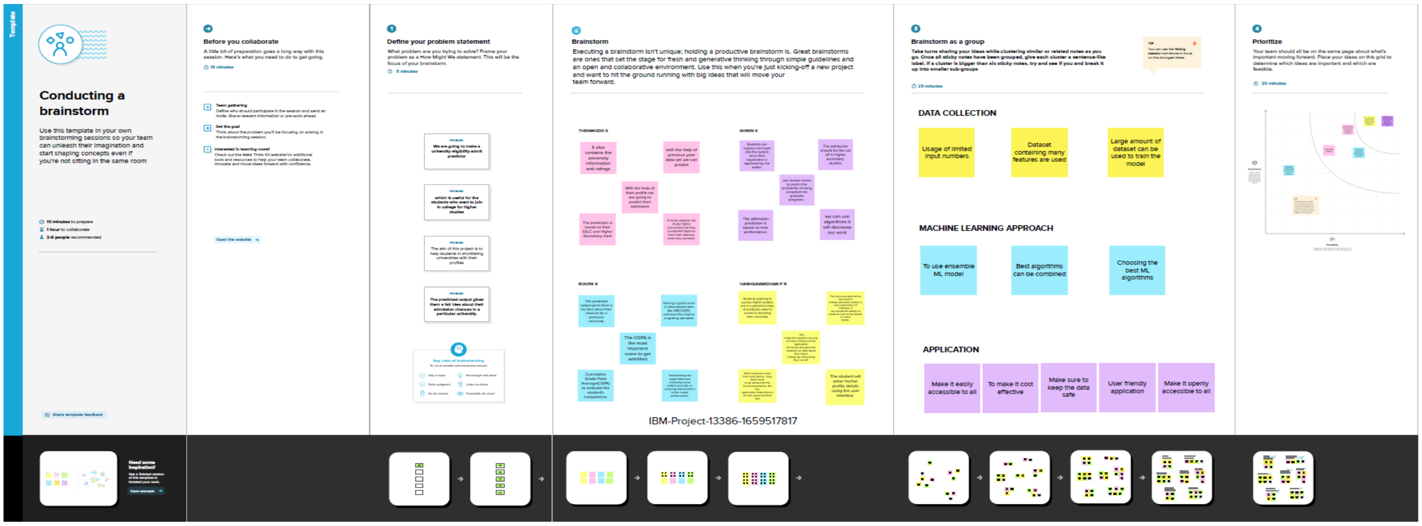
Inadequate

Unsure who to trust

Overwhelmed

Excited

3.2 IDEATION & BRAINSTORMING



Conducting a brainstorm

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Need some inspiration?
See a finished version of this template to kickstart your work.

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
15 minutes

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 Interested in learning more?
Check out the Meta Think Kit website for additional tools and resources to help your team collaborate, innovate and move ideas forward with confidence.

1 Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes

PROBLEM
We are going to make a university eligibility admit predictor

PROBLEM
which is useful for the students who want to join in college for higher studies

PROBLEM
The aim of this project is to help students in shortlisting universities with their profiles

PROBLEM
The predicted output gives them a fair idea about their admission chances in a particular university.

Key rules of brainstorming
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2 Brainstorm
Executing a brainstorm isn't unique; holding a productive brainstorm is. Great brainstorms are ones that set the stage for fresh and generative thinking through simple guidelines and an open and collaborative environment. Use this when you're just kicking-off a new project and want to hit the ground running with big ideas that will move your team forward.

THENMOZHI S

- It also contains the university information and ratings
- with the help of previous year data set we can predict
- With the help of their profile we are going to predict their admission
- The prediction is based on their SSLC and Higher Secondary mark
- If some student not study higher secondary but they completed diploma then their diploma mark also consider.

SHIRIN K

- Students can register and login into the system once their registration is approved by the admin.
- The admission should be like cut off in higher secondary studies.
- use random forest to predict the probability of being accepted into graduate programs.
- The admission prediction is based on their performance.
- we can use algorithms it will decrease our work

ROOPA V

- The predicted output gives them a fair idea about their chances for a particular university.
- Having a good score in international test like GRE,TOEFL enhance the chance of getting admitted.
- The CGPA is the most important score to get admitted
- Cumulative Grade Point Average(CGPA) to evaluate the student's competence.
- Transforming the target data and removing some outliers provide an amazing improvement in the model performance.

HARHSAVARDHAN P R

- Students aspiring to pursue higher studies are in a perennial state of confusion when it comes to deciding their university.
- The main principle behind the need of college admission system is easy supervision of institutes. It can handle the details of students such as [illegible] details or marks details.
- This helps the student not only to help in filling out the application forms but also give the students an idea about their future college by calculating their cut off.
- When students come from rural places , they find it hard to go along with the formal procedures. So, this application helps them a lot and eases out their fear.
- The student will enter his/her profile details using the user interface

IBM-Project-13386-1659517817

3 Brainstorm as a group
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups
25 minutes

TIP
You can use the Voting session tool above to focus on the strongest ideas.

DATA COLLECTION

- Usage of limited input numbers
- Dataset containing many features are used
- Large amount of dataset can be used to train the model

MACHINE LEARNING APPROACH

- To use ensemble ML model
- Best algorithms can be combined
- Choosing the best ML algorithms

APPLICATION

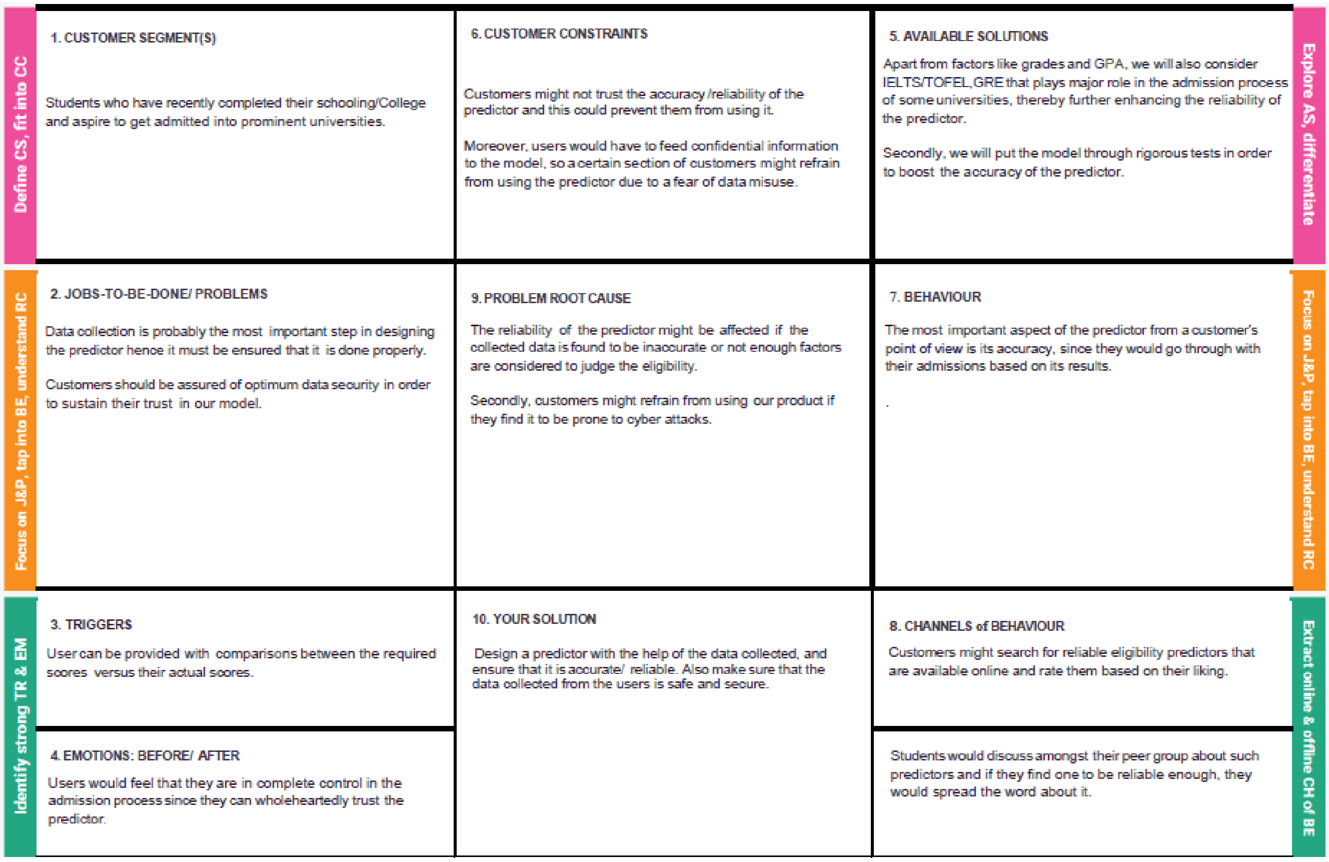
- Make it easily accessible to all
- To make it cost effective
- Make sure to keep the data safe
- User friendly application
- Make it openly accessible to all

4 Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement	Students do not have much idea about the procedures, requirement and details of the universities they want to join, so they seek help from various educational consultancies to help them secure admission in the universities based on their profile, for which the students are supposed pay a hefty amount as consultancy fee.
2.	Solution	Providing an accurate prediction for the student’s admission into the university of their choice based on various parameters like IELTS, GRE, Academic Performance, etc.
3.	Uniqueness	It seems there are no web applications for predicting the eligibility criteria of a student for getting in to their dream university and also provide personalized insights on specific areas where they can improve.
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	Further to reduce the immense pressure faced by the students to get admitted in a university, the model can also be evolved to consider university specific examinations and to maintain the latest eligibility criteria.

3.4 PROBLEM SOLUTION FIT


Define CS, fit into CC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Extract online & offline CH of BE

1. CUSTOMER SEGMENT(S)

Students who have recently completed their schooling/College and aspire to get admitted into prominent universities.

6. CUSTOMER CONSTRAINTS

Customers might not trust the accuracy /reliability of the predictor and this could prevent them from using it.

Moreover, users would have to feed confidential information to the model, so a certain section of customers might refrain from using the predictor due to a fear of data misuse.

5. AVAILABLE SOLUTIONS

Apart from factors like grades and GPA, we will also consider IELTS/TOFEL,GRE that plays major role in the admission process of some universities, thereby further enhancing the reliability of the predictor.

Secondly, we will put the model through rigorous tests in order to boost the accuracy of the predictor.

2. JOBS-TO-BE-DONE/ PROBLEMS

Data collection is probably the most important step in designing the predictor hence it must be ensured that it is done properly.

Customers should be assured of optimum data security in order to sustain their trust in our model.

9. PROBLEM ROOT CAUSE

The reliability of the predictor might be affected if the collected data is found to be inaccurate or not enough factors are considered to judge the eligibility.

Secondly, customers might refrain from using our product if they find it to be prone to cyber attacks.

7. BEHAVIOUR

The most important aspect of the predictor from a customer's point of view is its accuracy, since they would go through with their admissions based on its results.

.

3. TRIGGERS

User can be provided with comparisons between the required scores versus their actual scores.

4. EMOTIONS: BEFORE/ AFTER

Users would feel that they are in complete control in the admission process since they can wholeheartedly trust the predictor.

10. YOUR SOLUTION

Design a predictor with the help of the data collected, and ensure that it is accurate/ reliable. Also make sure that the data collected from the users is safe and secure.

8. CHANNELS of BEHAVIOUR

Customers might search for reliable eligibility predictors that are available online and rate them based on their liking.

Students would discuss amongst their peer group about such predictors and if they find one to be reliable enough, they would spread the word about it.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Requirements	All the needed files are been asked to feed in the website. By having the file, it will do all the pre- processing and shows all the required information to the student(user). The information includes the list of all the possible universities and streams.

FR-4	User Details	Has to feed some documents 1. GRE Score 2. TOEFL Score 3. Letter of Recommendation (LOR) 4. Statement of Purpose (SOP) 5. CGPA

4.2 NON FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• Our website is very user friendly so even the layman can able to access our website. • There is no need for any pre requisite technical skill in order to access our website. • Each and every content of the page will be in synchronous way. Thus, it will not take much time to refresh or reload.
NFR-2	**Security**	• The user who is having the valid credentials can able to access our site. • Data they are feeding into our website will not be accessed by any one of them.
NFR-3	**Reliability**	• Our website is more reliable. Since nobody can able to see the data fed by the user. • The user can get the result with higher percent of accuracy.
NFR-4	**Performance**	• User can able to handle the process in our website even by having internet connection with normal speed. There is no need of high-speed internet connection. • Traffics can be handled effectively.
NFR-5	**Availability**	• Students can avail our website from any of the browser in faster and efficient way.

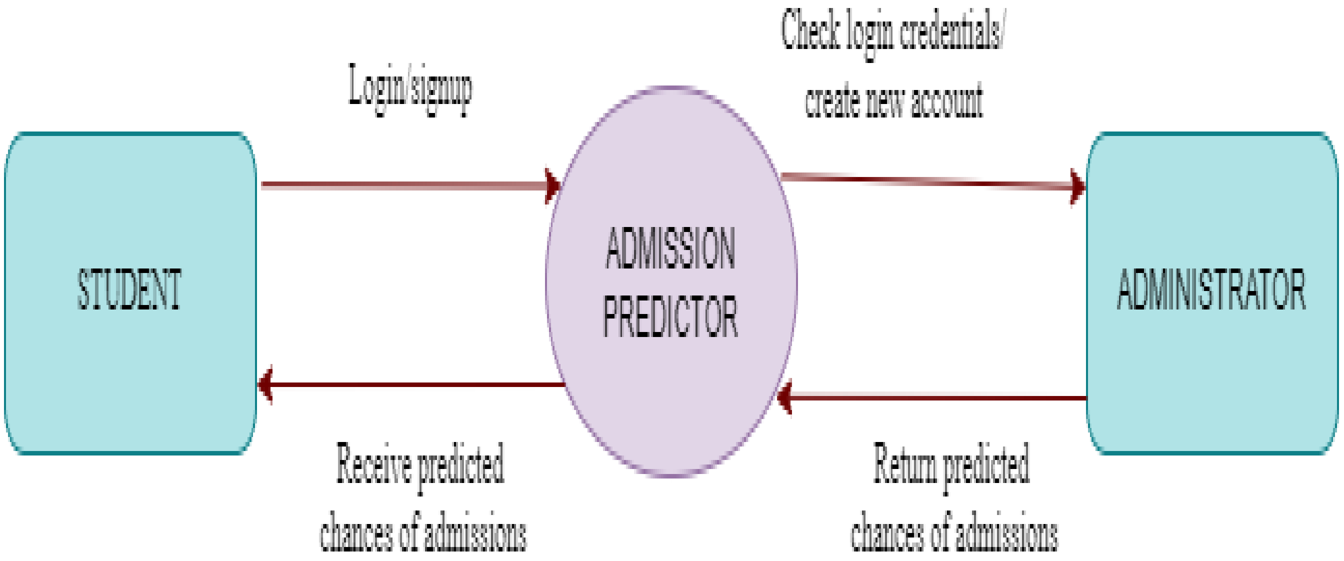
NFR-6	**Scalability**	• Our website will be easily scalable in the case of getting increasing number of users data from our website. • If needed we do scale up the CPU or Processer in order to speed up the processing capability of our website. • There by it reduces the downtime of our website.

CHAPTER 5

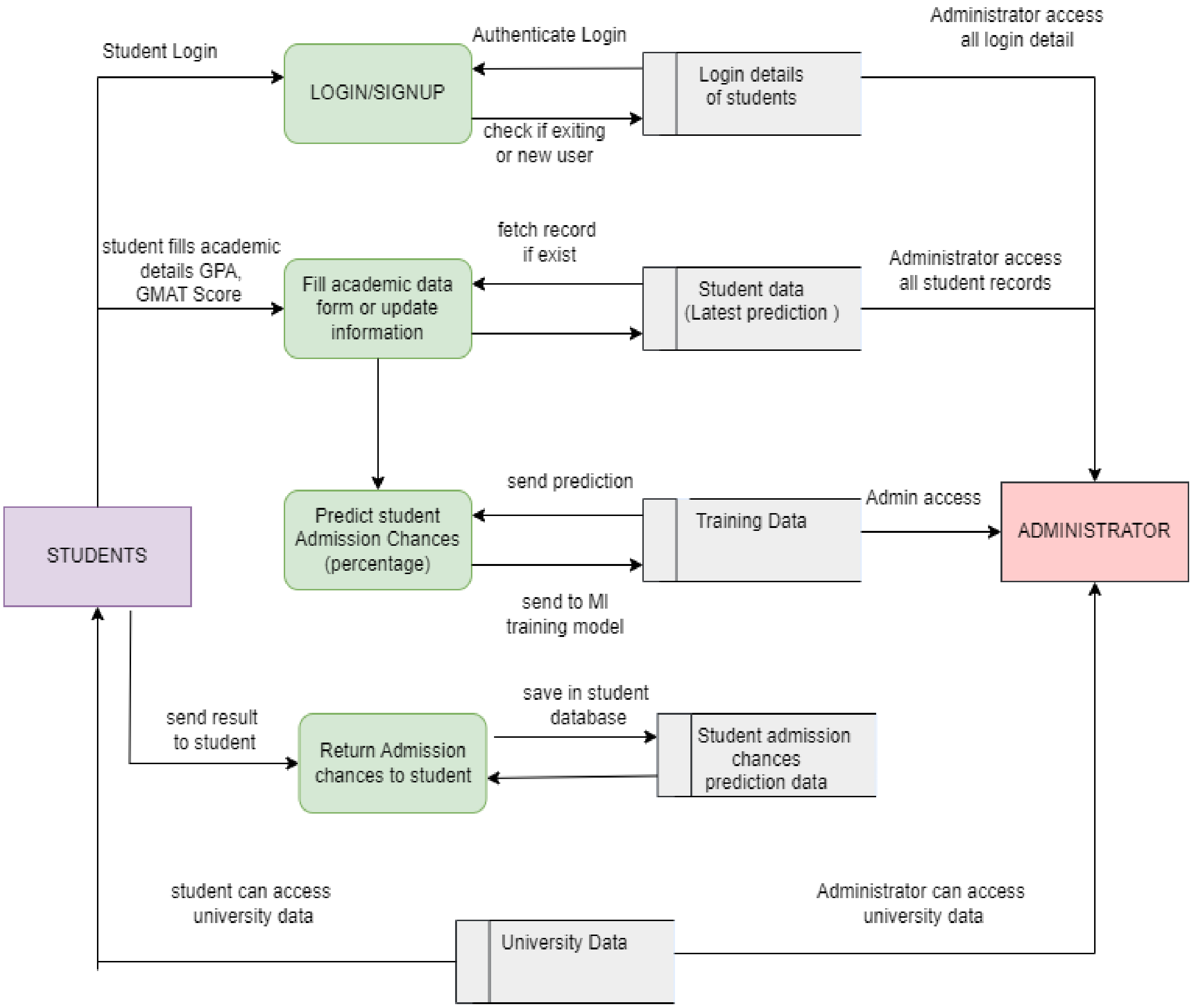
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

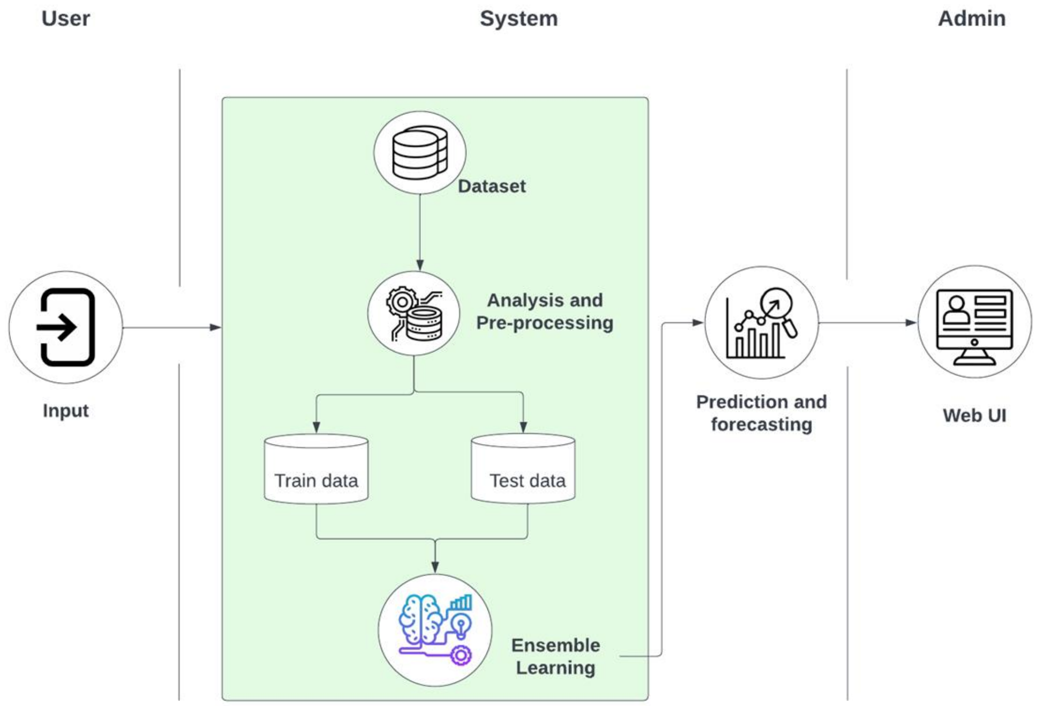
Level 0:



Level 1:


Student Login
LOGIN/SIGNUP
Authenticate Login
Login details of students
check if exiting or new user
Administrator access all login detail
student fills academic details GPA, GMAT Score
Fill academic data form or update information
fetch record if exist
Student data (Latest prediction)
Administrator access all student records
STUDENTS
Predict student Admission Chances (percentage)
send prediction
Training Data
Admin access
ADMINISTRATOR
send to Ml training model
send result to student
Return Admission chances to student
save in student database
Student admission chances prediction data
student can access university data
University Data
Administrator can access university data

5.1 SOLUTION & TECHNICAL ARCHITECTURE



S.No	Component	Description	Technology
1.	User Interface	The user get easy interaction through web based water quality analysis (UI).	HTML, CSS, Python
2.	Application Logic-1	Dealing with Dataset	Python

3.	Application Logic-2	Training and Building Ensemble Machine Learning Model	Python
4.	Application Logic-3	Deployment	Python Flask
5.	Database	Data format for processing	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	File Storage	To store the dataset for the process	Local Filesystem, IBM cloud
8.	Machine Learning Model	Ensemble Machine Learning Model can be used to increase the accuracy of the prediction.	Ensemble Machine Learning Model.
9.	Infrastructure (Server / Cloud)	On cloud server we will be deploying the web interface using flask in the web page	Python Flask

5.2 USER STORIES

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering the Name, email, password, and mobile number.	To store data in database for your identity	High	Sprint-1
Customer (Web user)	Access to the web page	USN-2	As a user, I can enter using email, name and mobile number.	I can enter the website	Medium	Sprint-4
Administrator	Evaluation page	USN-3	Enter the TOEFL score, GCPA score for validation.	To predict your eligibility	High	Sprint-2
Administrator	Manage the webpage	USN-4	Based on the accuracy produced it will show the result accordingly.	To produce result	High	Sprint-3
Administrator	Predicted result	USN-5	You will receive a mail regarding whether you are eligible or not.	To sent a notification based on eligibility criteria	Medium	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priorit y	Team Membe rs
Sprint-1	User Registration	USN-1	As a user, I can register for the application by entering my email address, password, and password confirmation.	2	High	2
Sprint-1		USN-2	Once I have registered for the application, I will receive a confirmation email.	1	High	1
Sprint-2		USN-3	I can check the eligibility criteria for various universities as a user by uploading the necessary documents.	2	Low	2
Sprint-3		USN-4	I can check the eligibility criteria for various universities as a user by uploading the necessary documents.	2	Mediu m	2

Sprint-4	User Login	USN-5	As a user, I can access the application by entering my email address and password.	1	High	2
	Dashboard		Check the dashboard for updates and upload the details to the desired and eligible universities based on the eligibility criteria.			4

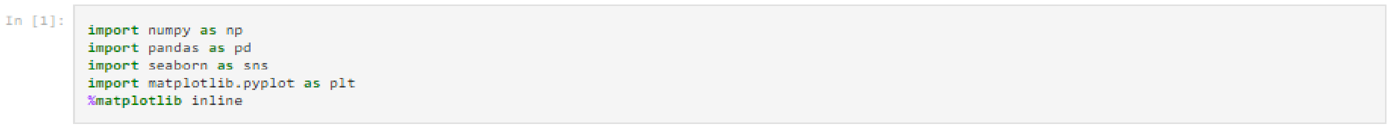
6.2 SPRINT DELIVERY SCHEDULE

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	30 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	25	20 Nov 2022

CHAPTER 7

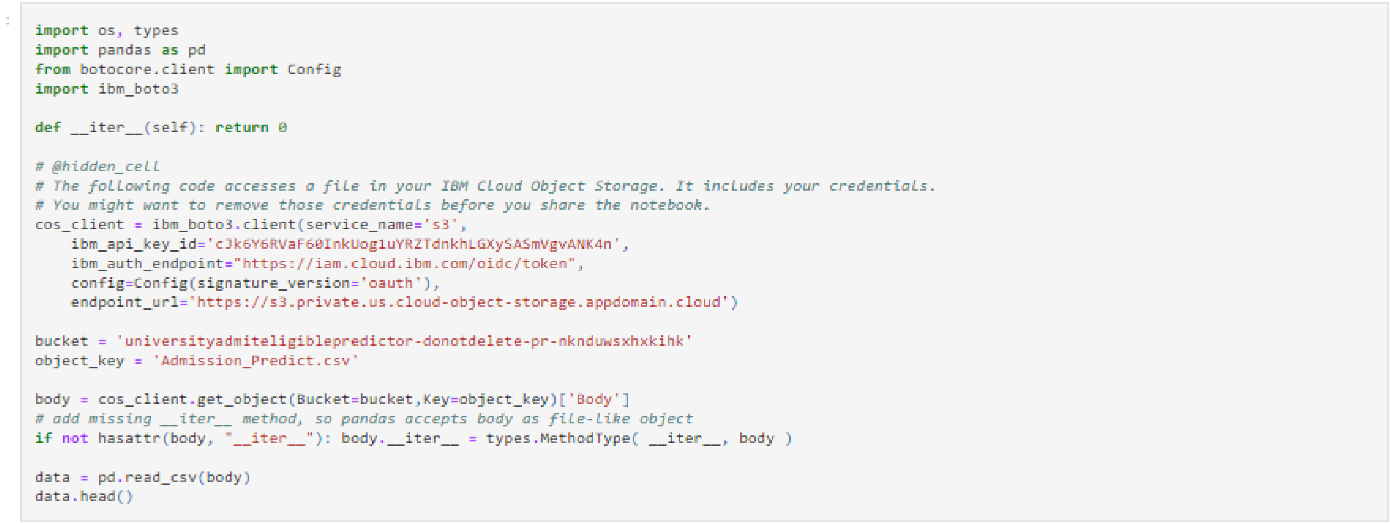
CODING & SOLUTIONING

IMPORT STATEMENTS



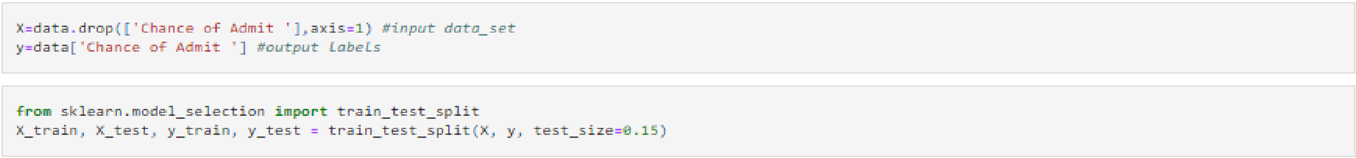
```
In [1]:  import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         %matplotlib inline
```

DATA READING AND ANALYSIS



```
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='cJk6Y6RVaF60InkUog1uYRZTdnkhLGXySASmVgvANK4n',
    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 = 'universityadmiteligiblepredictor-donotdelete-pr-nknduwsxhxkihk'
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()
```

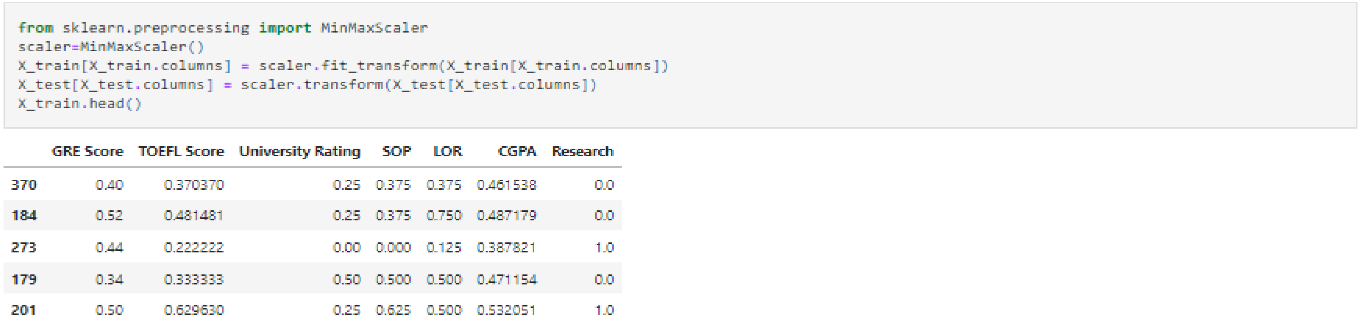
TRAIN-TEST SPLIT



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

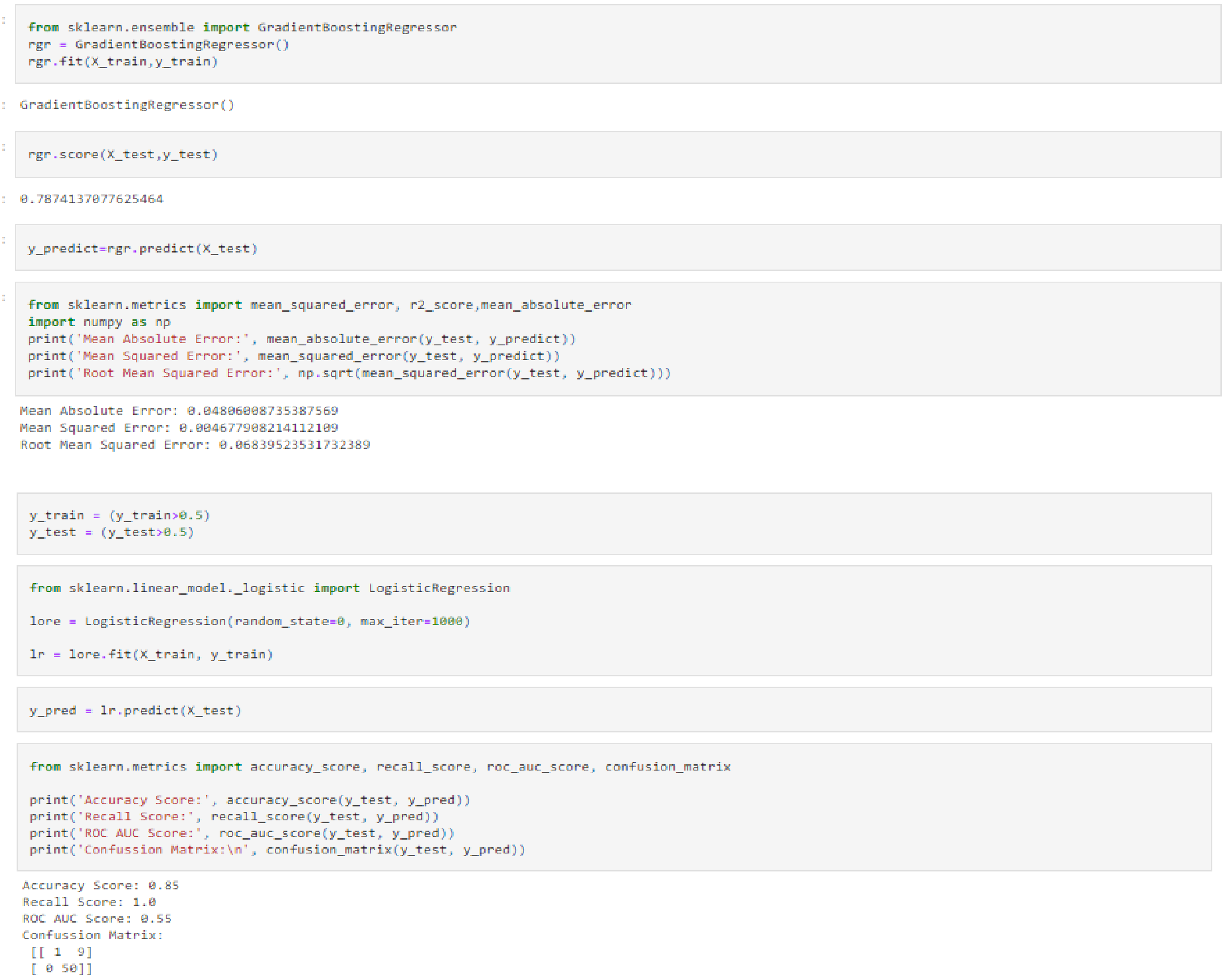
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15)
```

MODELING AND TRAINING



```
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()
```

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
370	0.40	0.370370	0.25	0.375	0.375	0.461538	0.0
184	0.52	0.481481	0.25	0.375	0.750	0.487179	0.0
273	0.44	0.222222	0.00	0.000	0.125	0.387821	1.0
179	0.34	0.333333	0.50	0.500	0.500	0.471154	0.0
201	0.50	0.629630	0.25	0.625	0.500	0.532051	1.0



```
from sklearn.ensemble import GradientBoostingRegressor
rgr = GradientBoostingRegressor()
rgr.fit(X_train,y_train)
```

```
GradientBoostingRegressor()
```

```
rgr.score(X_test,y_test)
```

```
0.7874137077625464
```

```
y_predict=rgr.predict(X_test)
```

```
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: 0.04806008735387569
Mean Squared Error: 0.004677908214112109
Root Mean Squared Error: 0.06839523531732389
```

```
y_train = (y_train>0.5)
y_test = (y_test>0.5)
```

```
from sklearn.linear_model._logistic import LogisticRegression

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

lr = lore.fit(X_train, y_train)
```

```
y_pred = lr.predict(X_test)
```

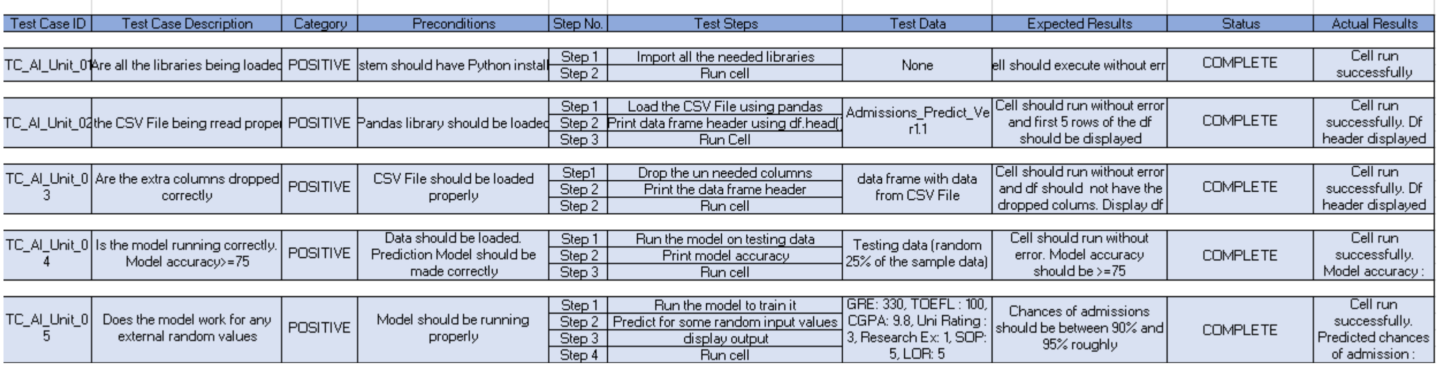
```
from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix

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: 0.85
Recall Score: 1.0
ROC AUC Score: 0.55
Confussion Matrix:
 [[ 1  9]
 [ 0 50]]
```

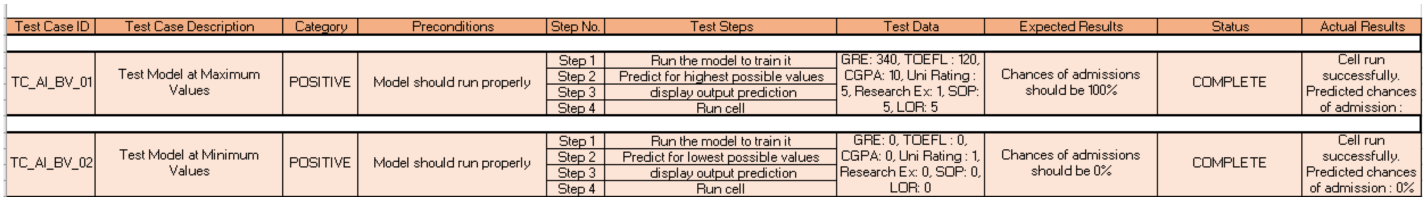
CHAPTER 8

TESTING

8.1 TEST CASES



Test Case ID	Test Case Description	Category	Preconditions	Step No.	Test Steps	Test Data	Expected Results	Status	Actual Results
TC_AI_Unit_01	Are all the libraries being loaded	POSITIVE	stem should have Python install	Step 1 Step 2	Import all the needed libraries Run cell	None	ell should execute without err	COMPLETE	Cell run successfully
TC_AI_Unit_02	the CSV File being rread proper	POSITIVE	Pandas library should be loaded	Step 1 Step 2 Step 3	Load the CSV File using pandas Print data frame header using df.head() Run Cell	Admissions_Predict_Ver1.1	Cell should run without error and first 5 rows of the df should be displayed	COMPLETE	Cell run successfully. Df header displayed
TC_AI_Unit_03	Are the extra columns dropped correctly	POSITIVE	CSV File should be loaded properly	Step1 Step 2 Step 2	Drop the un needed columns Print the data frame header Run cell	data frame with data from CSV File	Cell should run without error and df should not have the dropped colums. Display df	COMPLETE	Cell run successfully. Df header displayed
TC_AI_Unit_04	Is the model running correctly. Model accuracy>=75	POSITIVE	Data should be loaded. Prediction Model should be made correctly	Step 1 Step 2 Step 3	Run the model on testing data Print model accuracy Run cell	Testing data (random 25% of the sample data)	Cell should run without error. Model accuracy should be >=75	COMPLETE	Cell run successfully. Model accuracy :
TC_AI_Unit_05	Does the model work for any external random values	POSITIVE	Model should be running properly	Step 1 Step 2 Step 3 Step 4	Run the model to train it Predict for some random input values display output Run cell	GRE: 330, TOEFL : 100, CGPA: 9.8, Uni Rating : 3, Research Ex: 1, SOP: 5, LOR: 5	Chances of admissions should be between 90% and 95% roughly	COMPLETE	Cell run successfully. Predicted chances of admission :



Test Case ID	Test Case Description	Category	Preconditions	Step No.	Test Steps	Test Data	Expected Results	Status	Actual Results
TC_AI_BV_01	Test Model at Maximum Values	POSITIVE	Model should run properly	Step 1 Step 2 Step 3 Step 4	Run the model to train it Predict for highest possible values display output prediction Run cell	GRE: 340, TOEFL : 120, CGPA: 10, Uni Rating : 5, Research Ex: 1, SOP: 5, LOR: 5	Chances of admissions should be 100%	COMPLETE	Cell run successfully. Predicted chances of admission :
TC_AI_BV_02	Test Model at Minimum Values	POSITIVE	Model should run properly	Step 1 Step 2 Step 3 Step 4	Run the model to train it Predict for lowest possible values display output prediction Run cell	GRE: 0, TOEFL : 0, CGPA: 0, Uni Rating : 1, Research Ex: 0, SOP: 0, LOR: 0	Chances of admissions should be 0%	COMPLETE	Cell run successfully. Predicted chances of admission : 0%

8.2 USER ACCEPTANCE TESTING

8.2.1DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

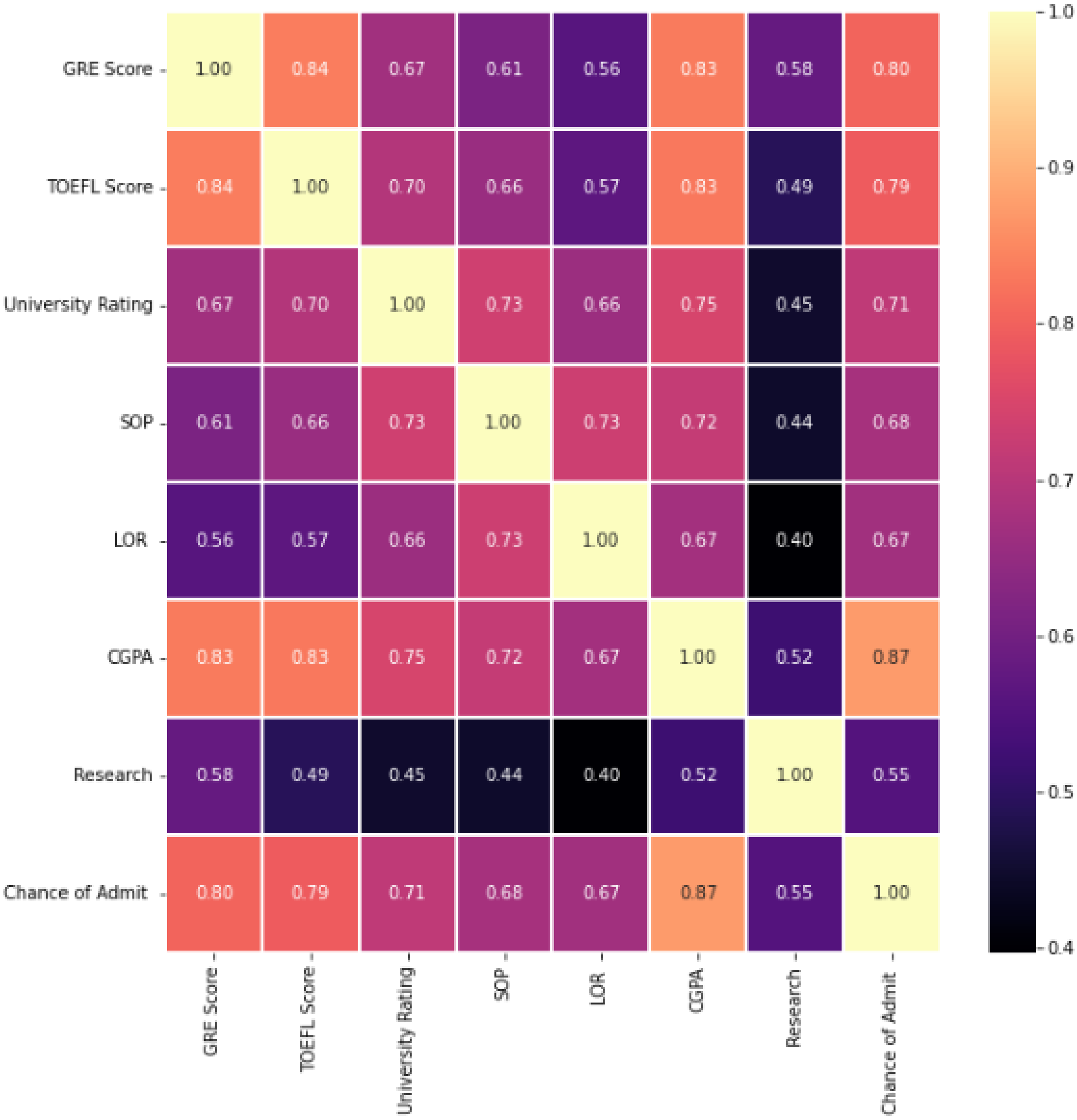
8.2.2TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS



	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
GRE Score	1.00	0.84	0.67	0.61	0.56	0.83	0.58	0.80
TOEFL Score	0.84	1.00	0.70	0.66	0.57	0.83	0.49	0.79
University Rating	0.67	0.70	1.00	0.73	0.66	0.75	0.45	0.71
SOP	0.61	0.66	0.73	1.00	0.73	0.72	0.44	0.68
LOR	0.56	0.57	0.66	0.73	1.00	0.67	0.40	0.67
CGPA	0.83	0.83	0.75	0.72	0.67	1.00	0.52	0.87
Research	0.58	0.49	0.45	0.44	0.40	0.52	1.00	0.55
Chance of Admit	0.80	0.79	0.71	0.68	0.67	0.87	0.55	1.00

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- The people who will benefit the most from using this system are Indian students.
- Especially students looking to pursue their higher education from foreign universities, particularly in the United States.
- It helps student for making decision for choosing a right college.
- Here the chance of occurrence of error is less when compared with the existing system.
- It is fast, efficient and reliable.
- Avoids data redundancy and inconsistency.
- Very user-friendly.
- Easy accessibility of data.

DISADVANTAGES

- The system is built on a limited data set, this could affect the accuracy of the predictions as a whole.
- The system cannot guarantee that our predictions will be a 100% guarantee of admissions because a lot other factors such as the Personal Interview also plays a major role in the admissions procedure.
- 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.
- Admissions also depend on the individual university’s policy regarding the intake of foreign students and is not modelled by our system.
- System will provide inaccurate results if data entered incorrectly.

CHAPTER 11

CONCLUSION

The subject of this examination was to determine if the below variables contribute to the admission of student to Master’s degree program.

- GRE Score
- TOEFL Score
- University Rating
- SOP
- LOR
- CGPA

The results of this examination appear to indicate that it greatly contributes to the response variable ‘Chance of Admit’. Higher the GRE, TOEFL score then higher the admit chances. The model predicts 87.5% accuracy and can be used for predicting the admit chances based on the above factors. This model will be helpful for the universities to predict the admission and ease their process of selection and timelines. The model proved that admission to Master’s degree program is dependent on GRE, TOEFL and other scores. This model would likely be greatly improved by the gathering of additional data of students from different universities which has similar selection criteria to choose the candidates for Master’s program.

CHAPTER 12

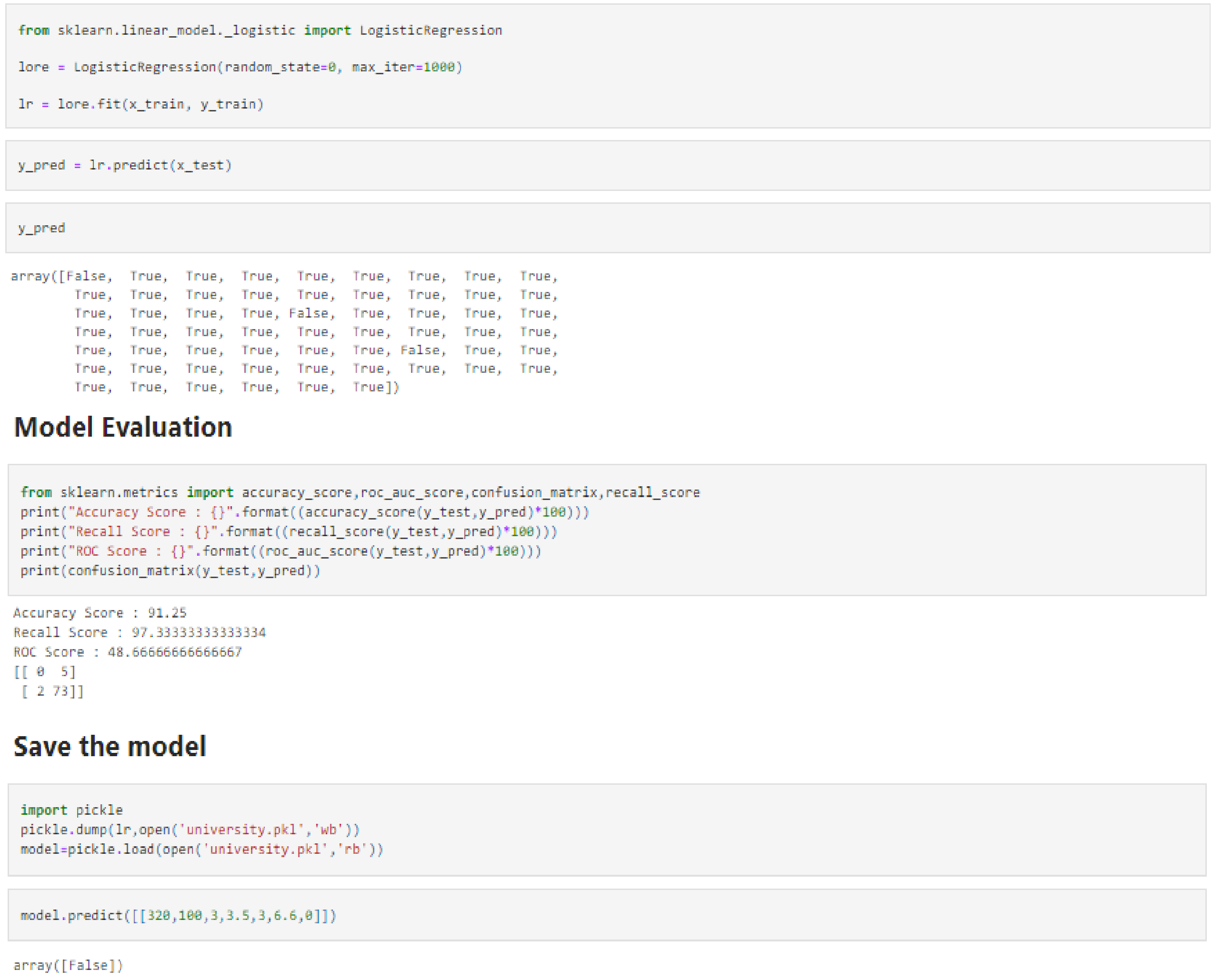
FUTURE SCOPE

Future work in the project could include weighing in the features that have been ignored as of yet like percentage seats for Foreign Students. Other criterions like Co-curricular achievements, Leadership positions held, job experience etc can also be included as metrics for the model.

APPENDIX

SOURCE CODE

MODEL CREATION



```
from sklearn.linear_model._logistic import LogisticRegression

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

lr = lore.fit(x_train, y_train)
```

```
y_pred = lr.predict(x_test)
```

```
y_pred
```

```
array([False,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True, False,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True, False,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True])
```

Model Evaluation

```
from sklearn.metrics import accuracy_score,roc_auc_score,confusion_matrix,recall_score
print("Accuracy Score : {}".format((accuracy_score(y_test,y_pred)*100)))
print("Recall Score : {}".format((recall_score(y_test,y_pred)*100)))
print("ROC Score : {}".format((roc_auc_score(y_test,y_pred)*100)))
print(confusion_matrix(y_test,y_pred))
```

```
Accuracy Score : 91.25
Recall Score : 97.33333333333334
ROC Score : 48.66666666666667
[[ 0  5]
 [ 2 73]]
```

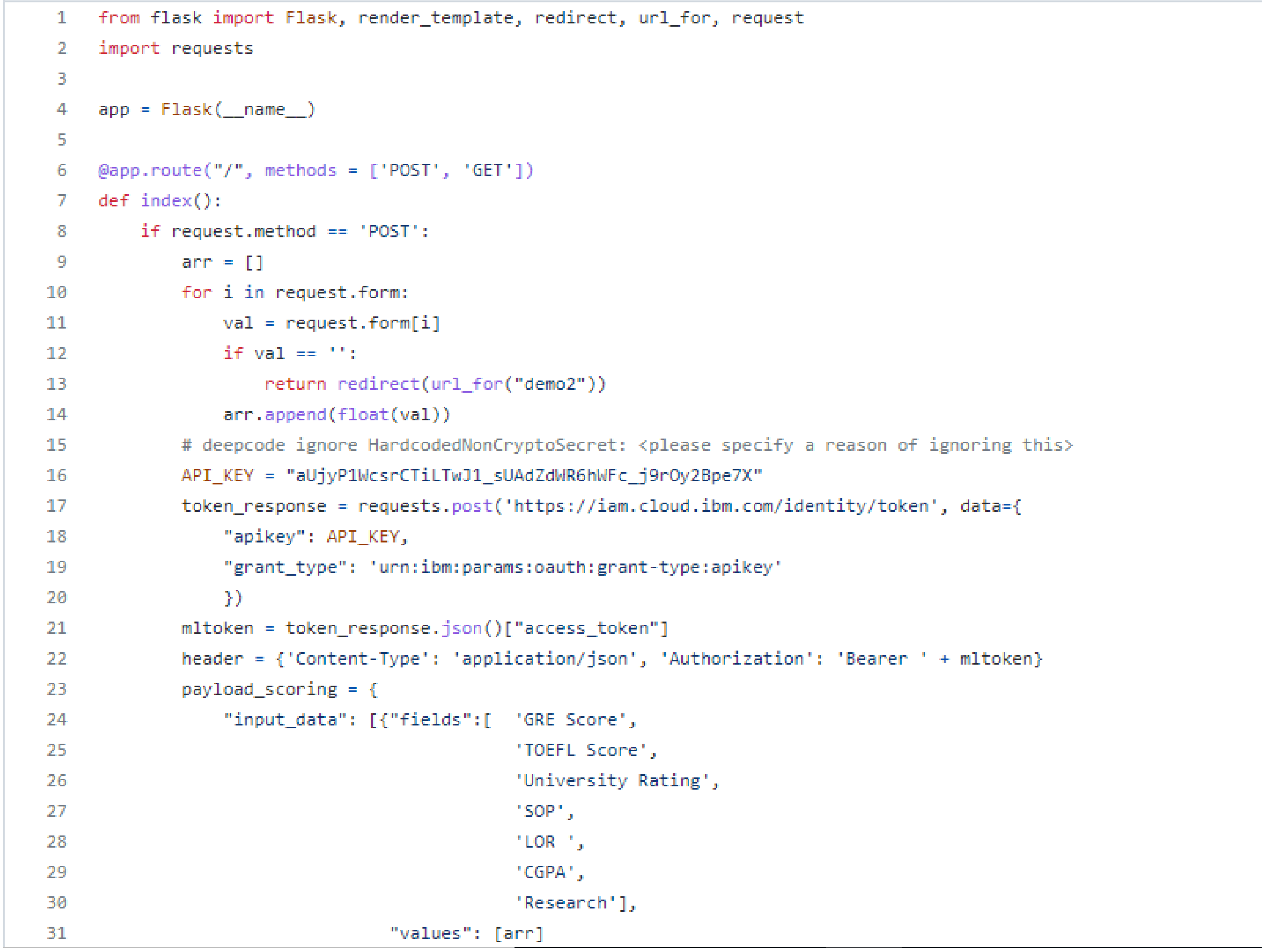
Save the model

```
import pickle
pickle.dump(lr,open('university.pkl','wb'))
model=pickle.load(open('university.pkl','rb'))
```

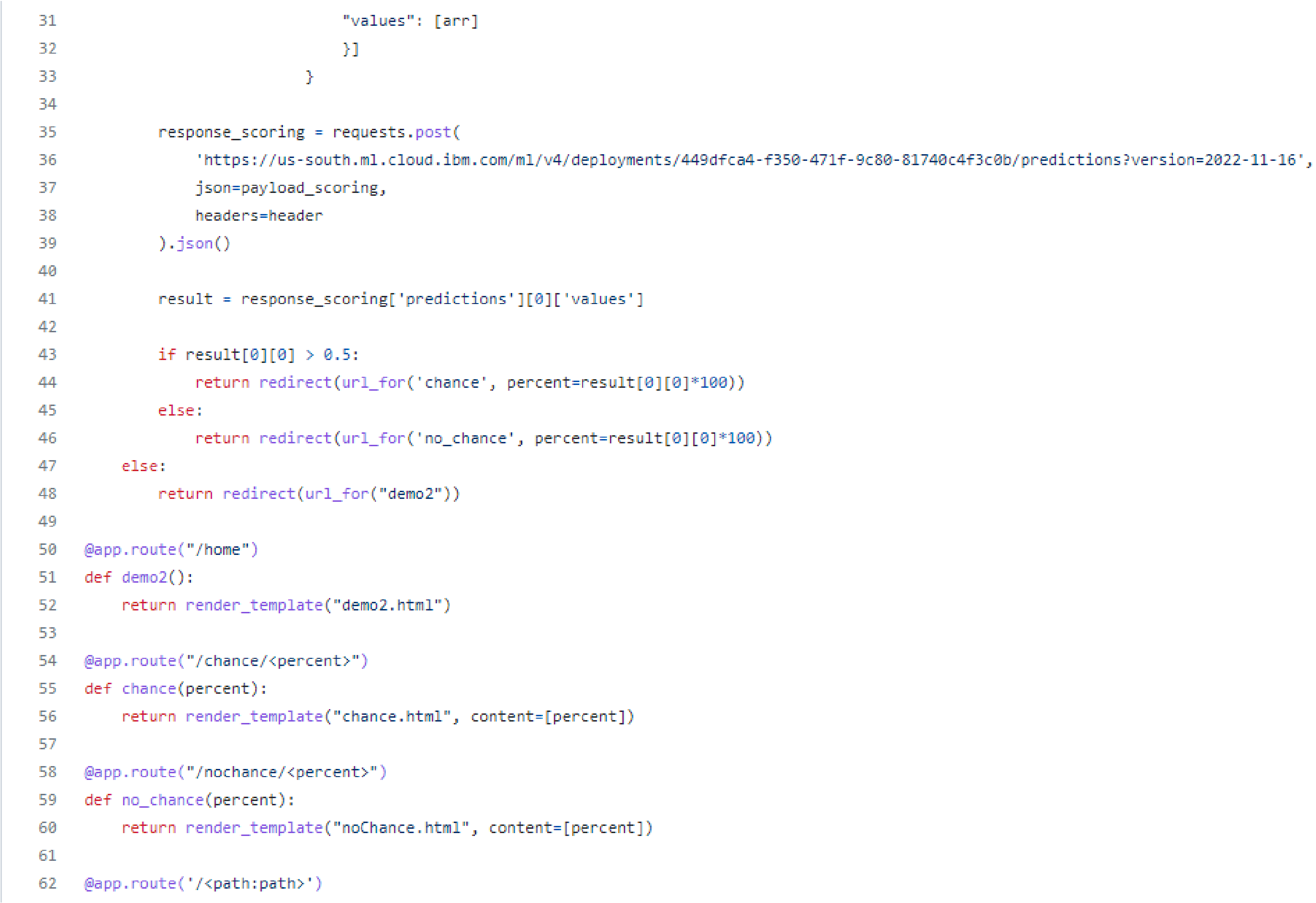
```
model.predict([[320,100,3,3.5,3,6.6,0]])
```

```
array([False])
```

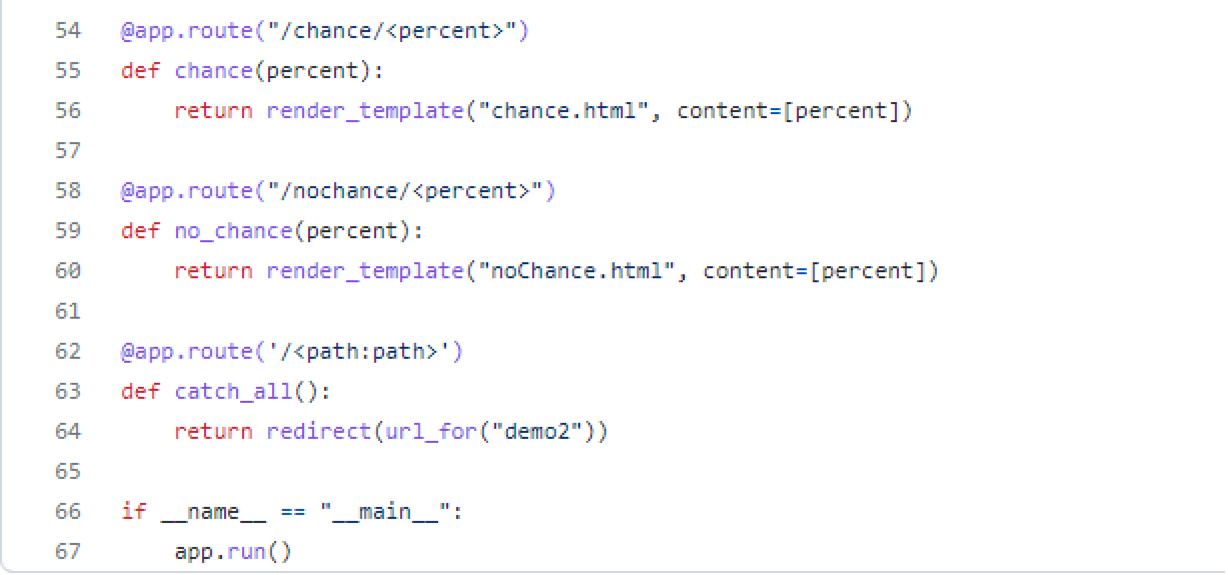
FLASK APP



```
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))
        # deepcode ignore HardcodedNonCryptoSecret: <please specify a reason of ignoring this>
        API_KEY = "aUjyP1WcsrCTiLTwJ1_sUAdZdWR6hWFc_j9rOy2Bpe7X"
        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]
```



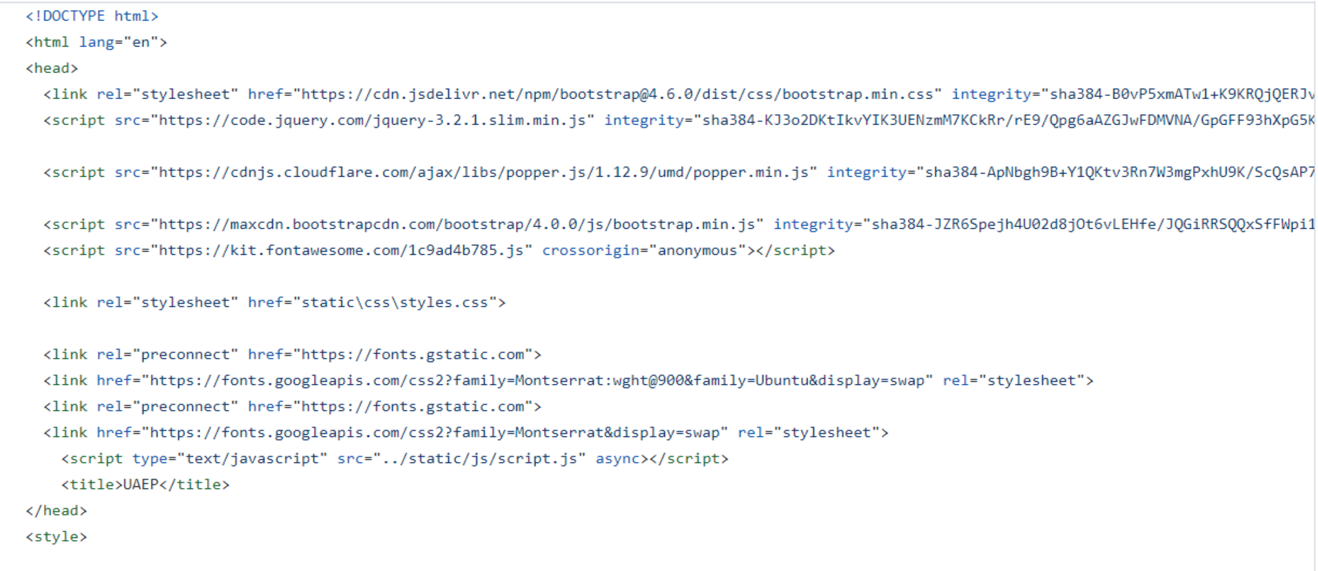
```
                            "values": [arr]
                            }]
                        }

        response_scoring = requests.post(
            'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/449dfca4-f350-471f-9c80-81740c4f3c0b/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>')
```



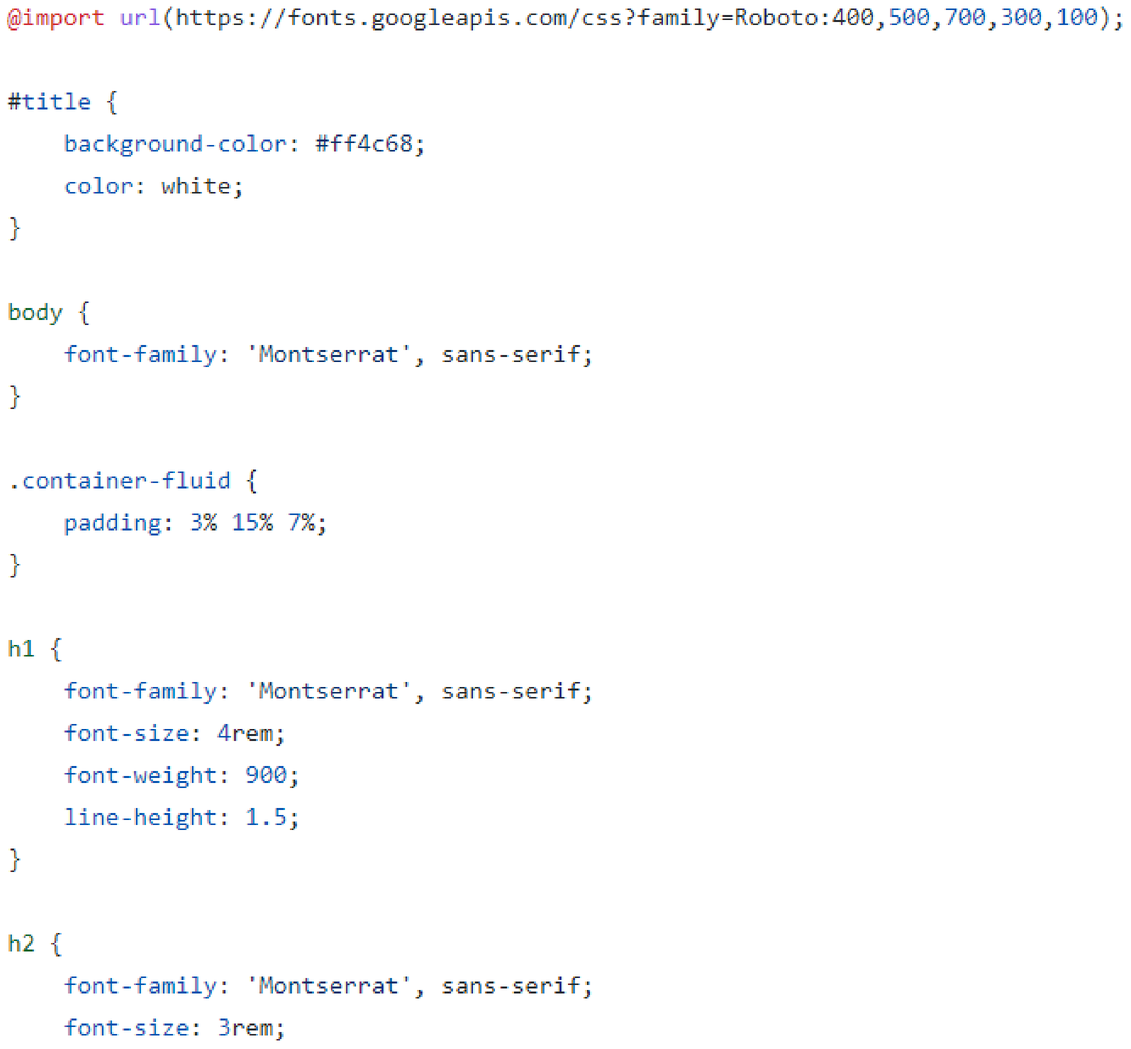
```
@app.route("/chance/<percent>")
def chance(percent):
    return render_template("chance.html", content=[percent])

@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()
```

INDEX (HTML)



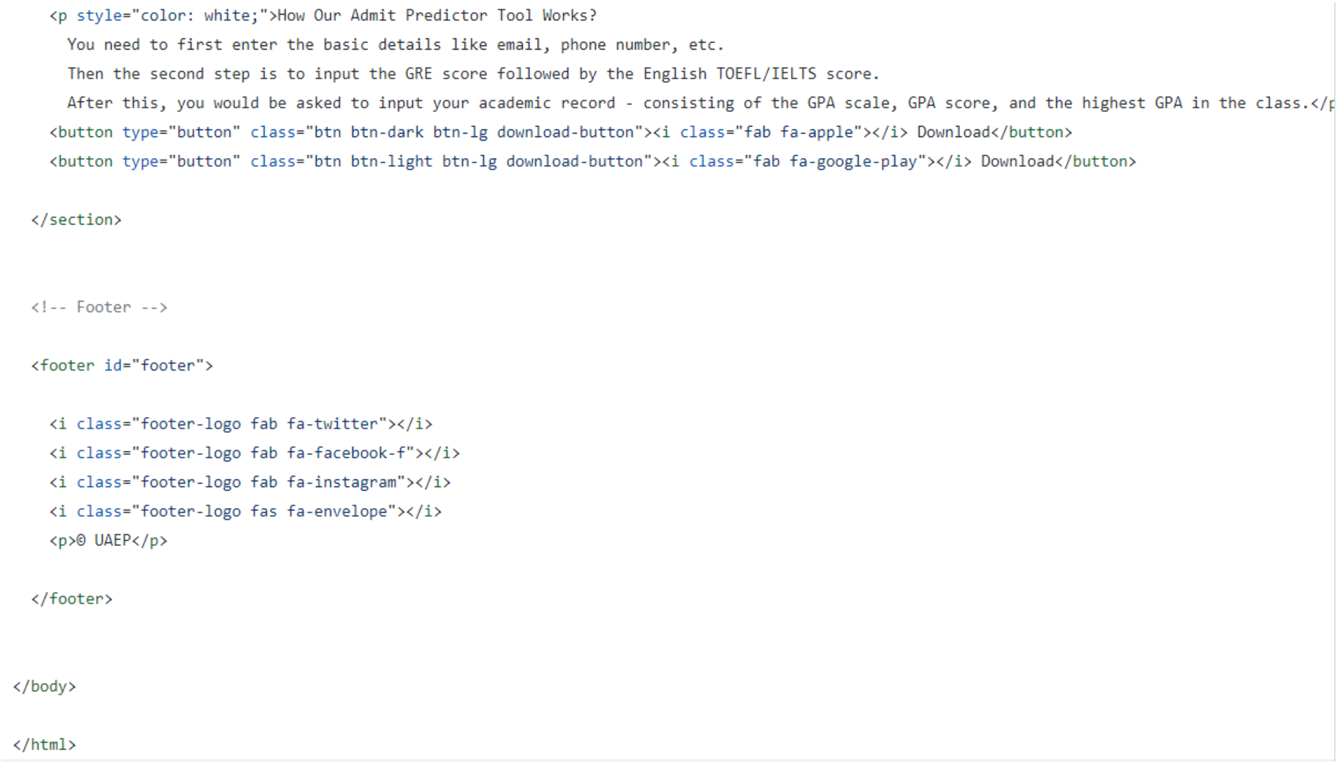
```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css" integrity="sha384-B0vP5xmATw1+K9KRQjQERJv
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5K

  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1
  <script src="https://kit.fontawesome.com/1c9ad4b785.js" crossorigin="anonymous"></script>

  <link rel="stylesheet" href="static\css\styles.css">

  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@900&family=Ubuntu&display=swap" rel="stylesheet">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Montserrat&display=swap" rel="stylesheet">
    <script type="text/javascript" src="../static/js/script.js" async></script>
    <title>UAEP</title>
</head>
<style>
```



```
@import url(https://fonts.googleapis.com/css?family=Roboto:400,500,700,300,100);

#title {
    background-color: #ff4c68;
    color: white;
}

body {
    font-family: 'Montserrat', sans-serif;
}

.container-fluid {
    padding: 3% 15% 7%;
}

h1 {
    font-family: 'Montserrat', sans-serif;
    font-size: 4rem;
    font-weight: 900;
    line-height: 1.5;
}

h2 {
    font-family: 'Montserrat', sans-serif;
    font-size: 3rem;
```

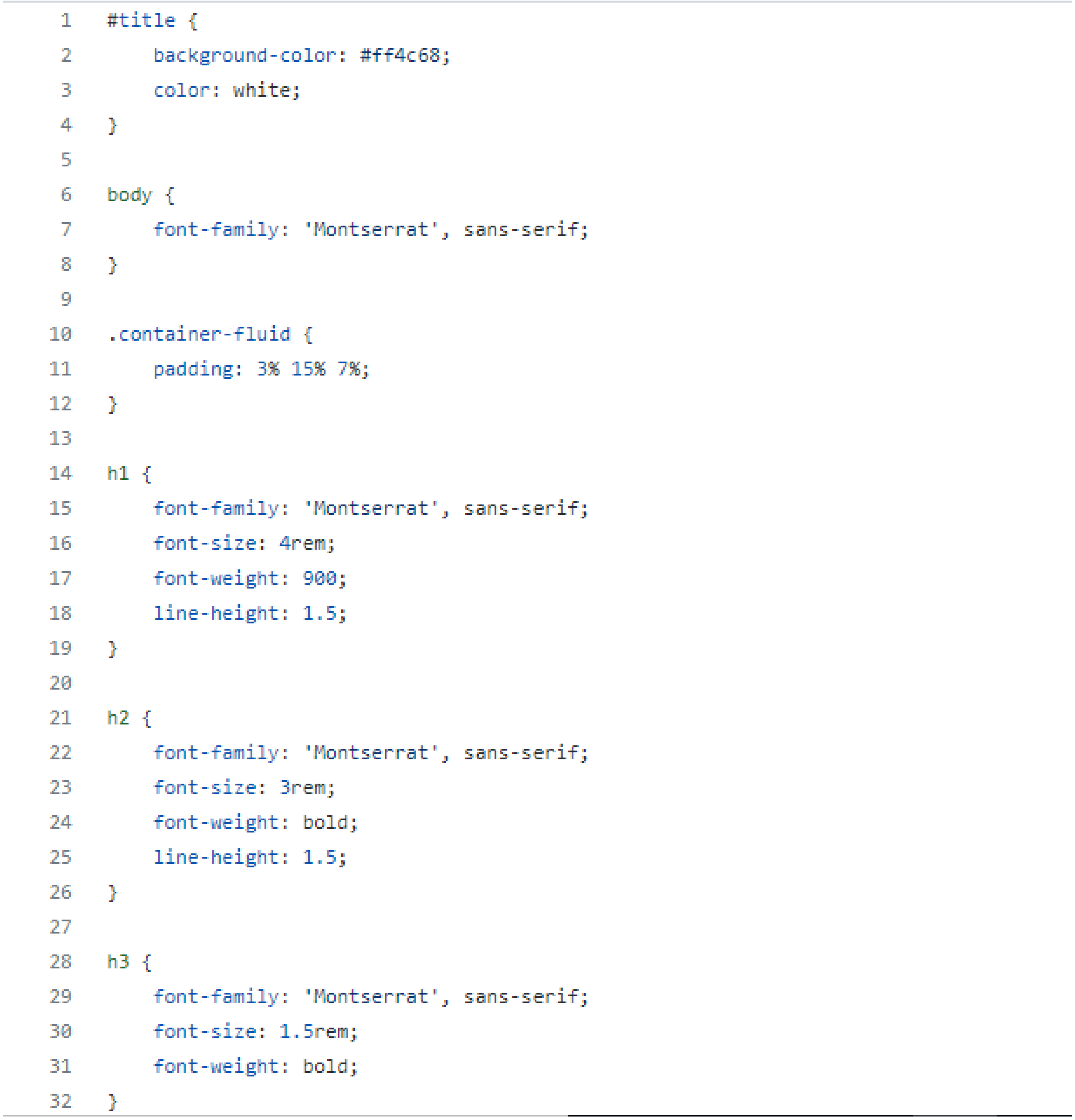
```css
    font-size: 3rem;
    font-weight: bold;
    line-height: 1.5;
}

h3 {
    font-family: 'Montserrat', sans-serif;
    font-size: 1.5rem;
    font-weight: bold;
}

/* Navigation bar */

.navbar {
    padding: 0 0 4.5rem;
}

.nav-item {
    padding: 0 18px;
}

.navbar-brand {
    font-family: 'Ubuntu', sans-serif;
    font-size: 2.5rem;
    font-weight: bold;
}
```

```
-

.nav-link {
    font-size: 1.2rem;
}

/* Download button */

.download-button {
    margin: 5% 3% 5% 0;
}

/* Title-image */

.title-image {
    transform: rotate(25deg);
    width: 125%;
    position: absolute;
    left: 3%;
}

/* Features section */

#features {
    background-color: white;
    padding: 7% 15%;
```

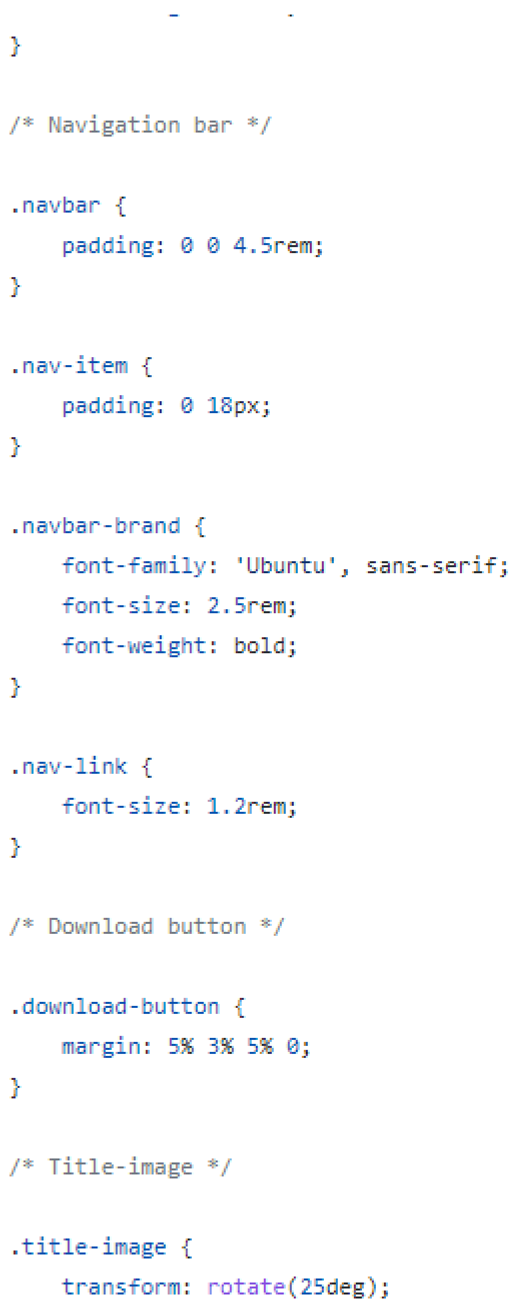


```
    <p style="color: white;">How Our Admit Predictor Tool Works?
      You need to first enter the basic details like email, phone number, etc.
      Then the second step is to input the GRE score followed by the English TOEFL/IELTS score.
      After this, you would be asked to input your academic record - consisting of the GPA scale, GPA score, and the highest GPA in the class.</p
    <button type="button" class="btn btn-dark btn-lg download-button"><i class="fab fa-apple"></i> Download</button>
    <button type="button" class="btn btn-light btn-lg download-button"><i class="fab fa-google-play"></i> Download</button>

  </section>


  <!-- Footer -->

  <footer id="footer">

    <i class="footer-logo fab fa-twitter"></i>
    <i class="footer-logo fab fa-facebook-f"></i>
    <i class="footer-logo fab fa-instagram"></i>
    <i class="footer-logo fas fa-envelope"></i>
    <p>© UAEP</p>

  </footer>


</body>

</html>
```

STYLES (CSS)

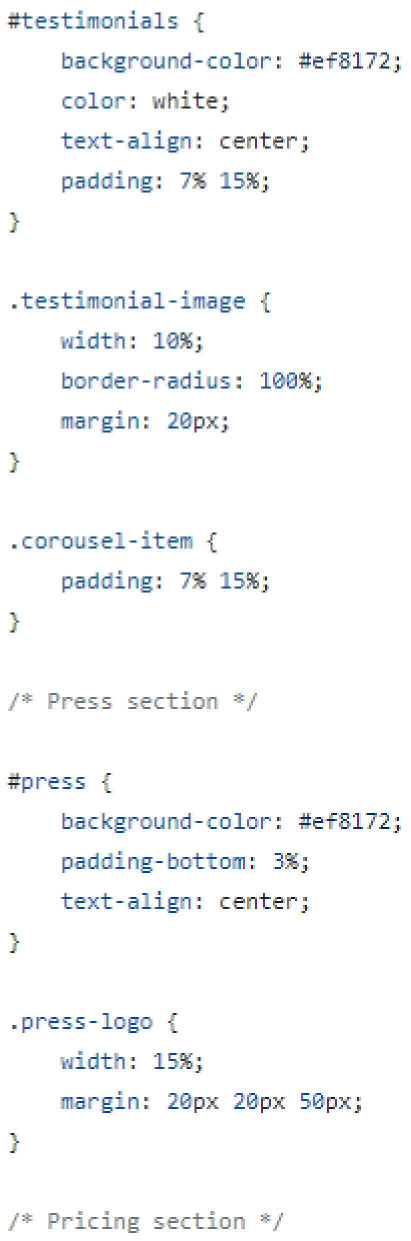


```css
#title {
    background-color: #ff4c68;
    color: white;
}

body {
    font-family: 'Montserrat', sans-serif;
}

.container-fluid {
    padding: 3% 15% 7%;
}

h1 {
    font-family: 'Montserrat', sans-serif;
    font-size: 4rem;
    font-weight: 900;
    line-height: 1.5;
}

h2 {
    font-family: 'Montserrat', sans-serif;
    font-size: 3rem;
    font-weight: bold;
    line-height: 1.5;
}

h3 {
    font-family: 'Montserrat', sans-serif;
    font-size: 1.5rem;
    font-weight: bold;
}
```

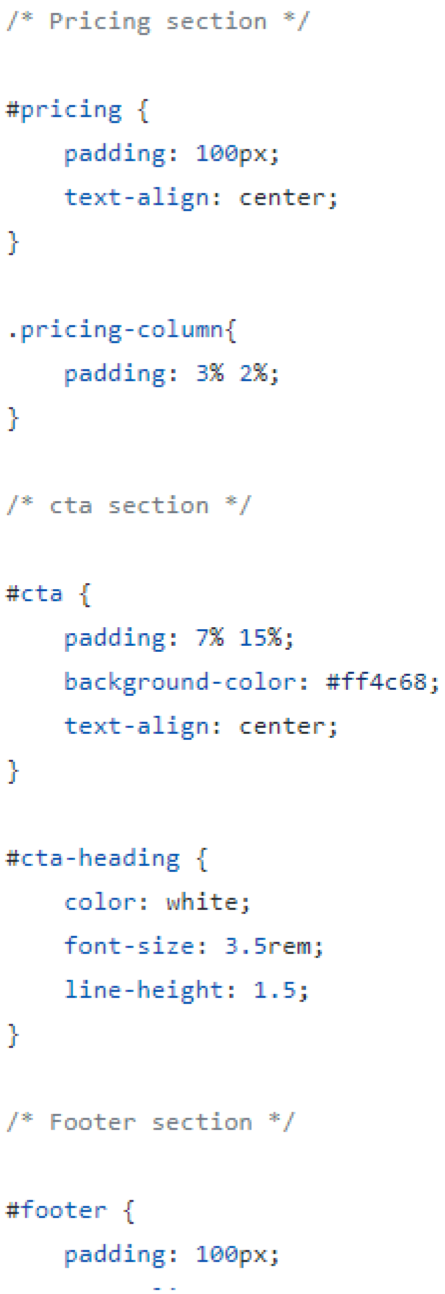


```css
}

/* Navigation bar */

.navbar {
    padding: 0 0 4.5rem;
}

.nav-item {
    padding: 0 18px;
}

.navbar-brand {
    font-family: 'Ubuntu', sans-serif;
    font-size: 2.5rem;
    font-weight: bold;
}

.nav-link {
    font-size: 1.2rem;
}

/* Download button */

.download-button {
    margin: 5% 3% 5% 0;
}

/* Title-image */

.title-image {
    transform: rotate(25deg);
```

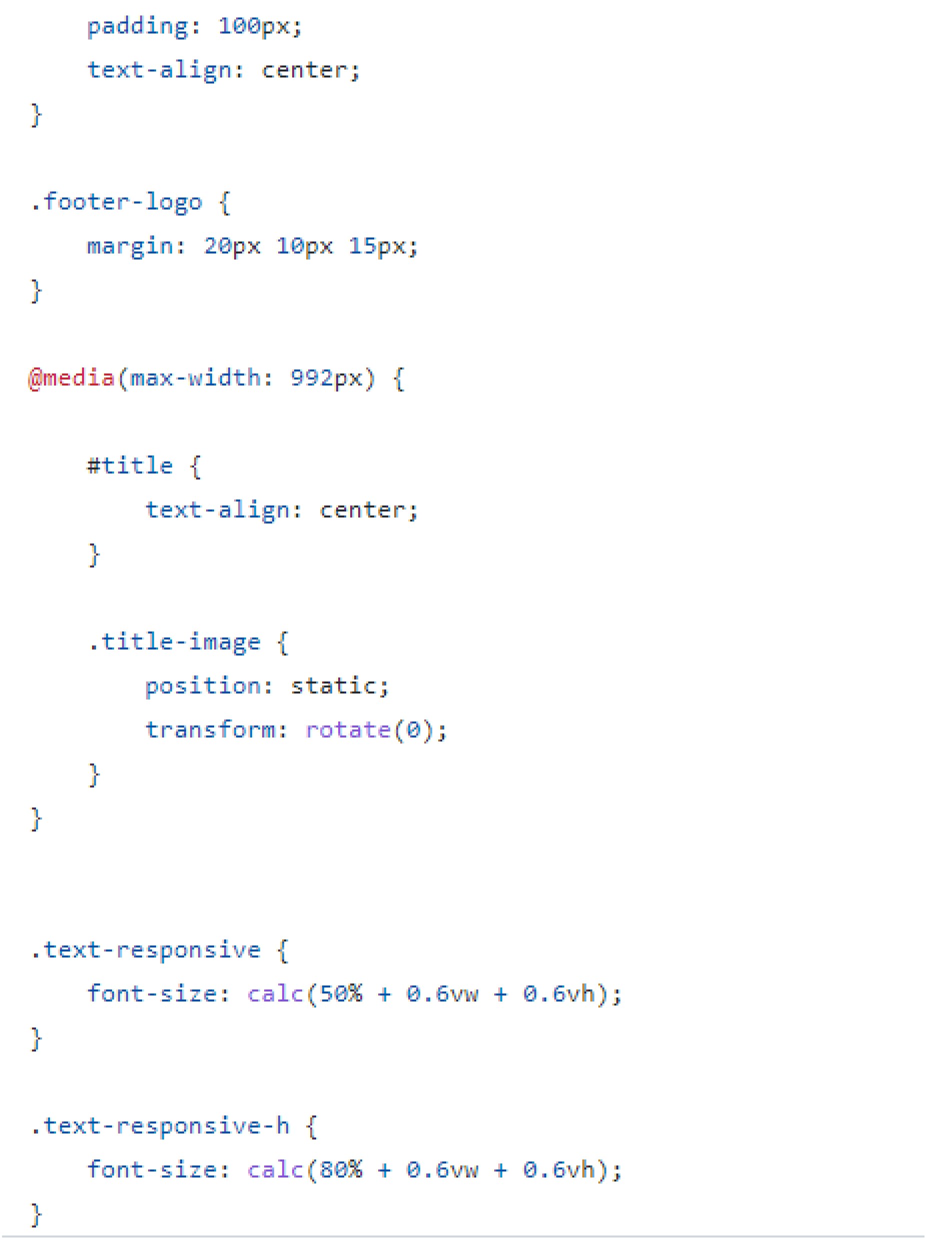
```css
.title-image {
    transform: rotate(25deg);
    width: 125%;
    position: absolute;
    left: 3%;
}

/* Features section */

#features {
    background-color: white;
    padding: 7% 15%;
    text-align: center;
    position: relative;
}

p {
    color: #8f8f8f;
}


.feature-image {
    color: #ef8172;
    margin: 40px 20px 20px 20px;
}

.feature-image:hover {
    color: #ff4c68;
}

/* Testimonial section */
```



```
#testimonials {
    background-color: #ef8172;
    color: white;
    text-align: center;
    padding: 7% 15%;
}

.testimonial-image {
    width: 10%;
    border-radius: 100%;
    margin: 20px;
}

.corousel-item {
    padding: 7% 15%;
}

/* Press section */

#press {
    background-color: #ef8172;
    padding-bottom: 3%;
    text-align: center;
}

.press-logo {
    width: 15%;
    margin: 20px 20px 50px;
}

/* Pricing section */
```

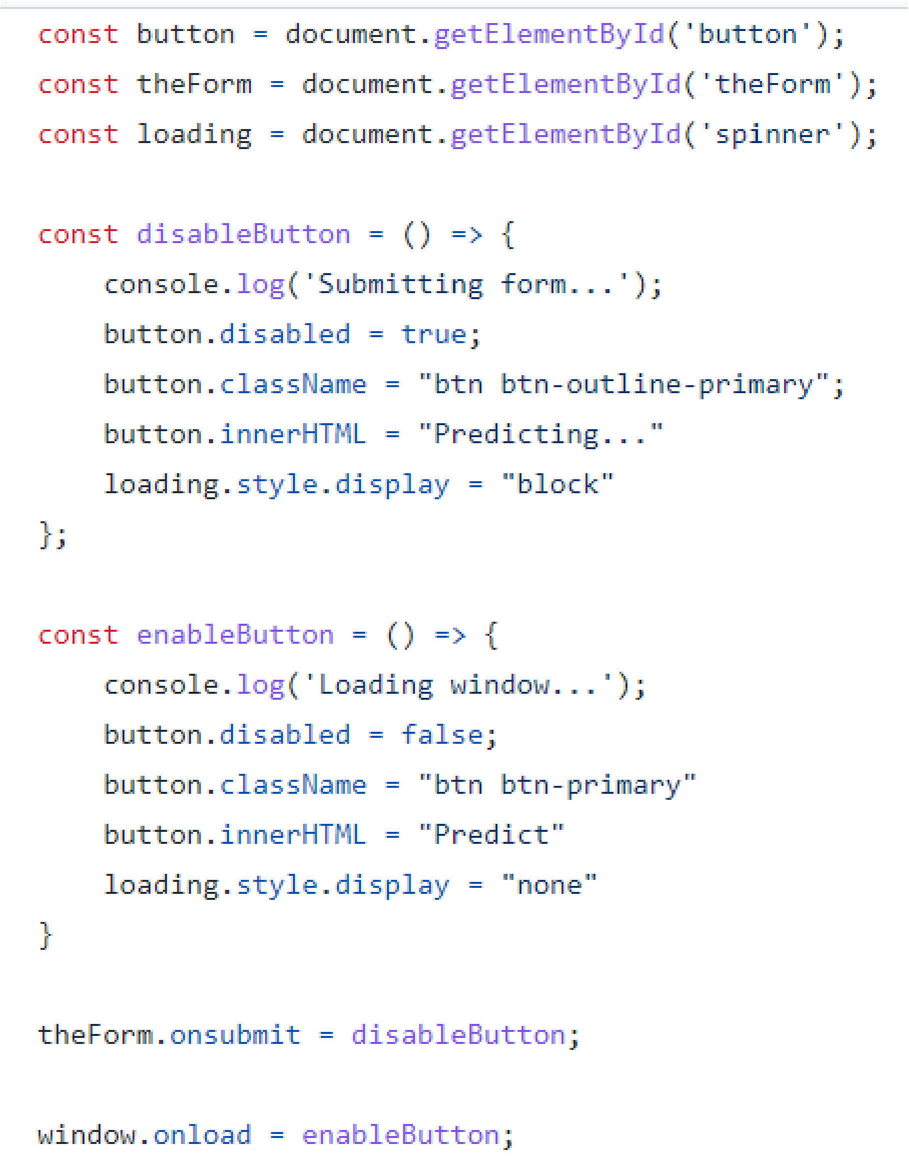


```
/* Pricing section */

#pricing {
    padding: 100px;
    text-align: center;
}

.pricing-column{
    padding: 3% 2%;
}

/* cta section */

#cta {
    padding: 7% 15%;
    background-color: #ff4c68;
    text-align: center;
}

#cta-heading {
    color: white;
    font-size: 3.5rem;
    line-height: 1.5;
}

/* Footer section */

#footer {
    padding: 100px;
```



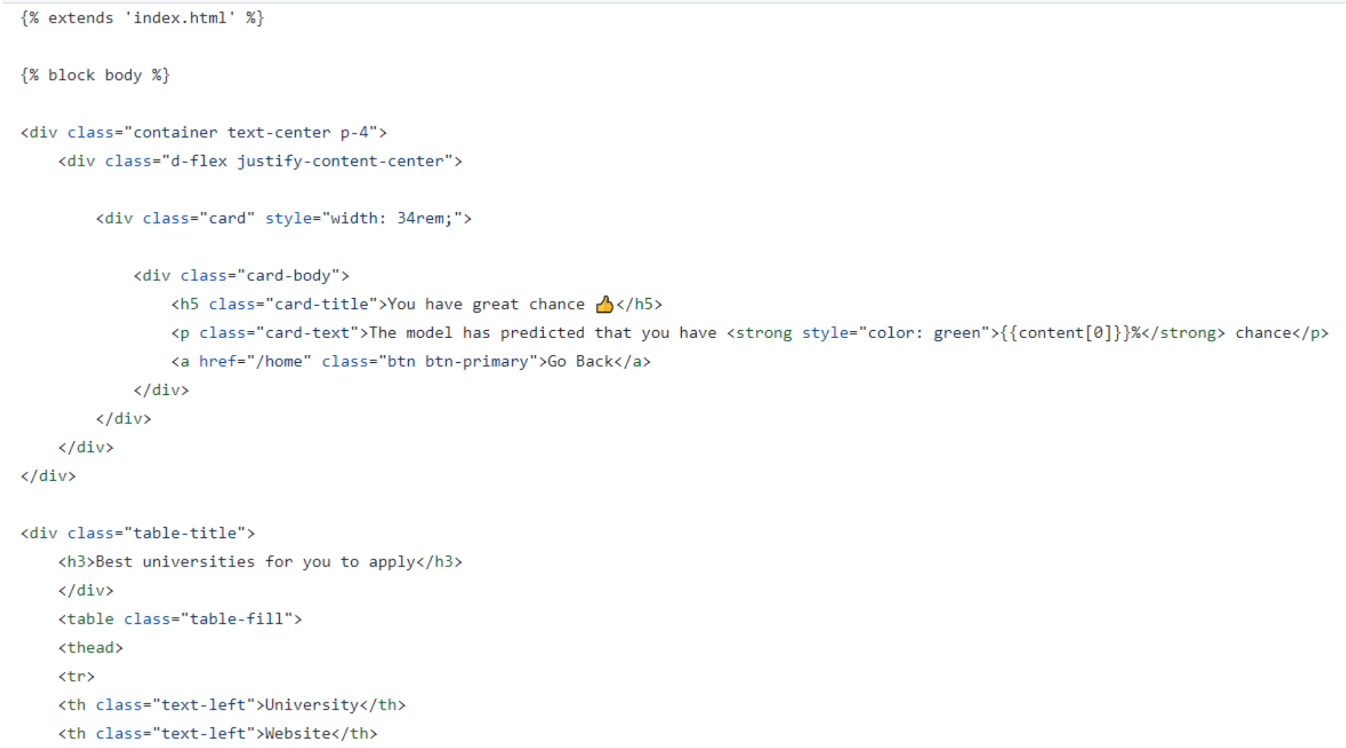
```
    padding: 100px;
    text-align: center;
}

.footer-logo {
    margin: 20px 10px 15px;
}

@media(max-width: 992px) {

    #title {
        text-align: center;
    }

    .title-image {
        position: static;
        transform: rotate(0);
    }
}


.text-responsive {
    font-size: calc(50% + 0.6vw + 0.6vh);
}

.text-responsive-h {
    font-size: calc(80% + 0.6vw + 0.6vh);
}
```

INDEX (JS)

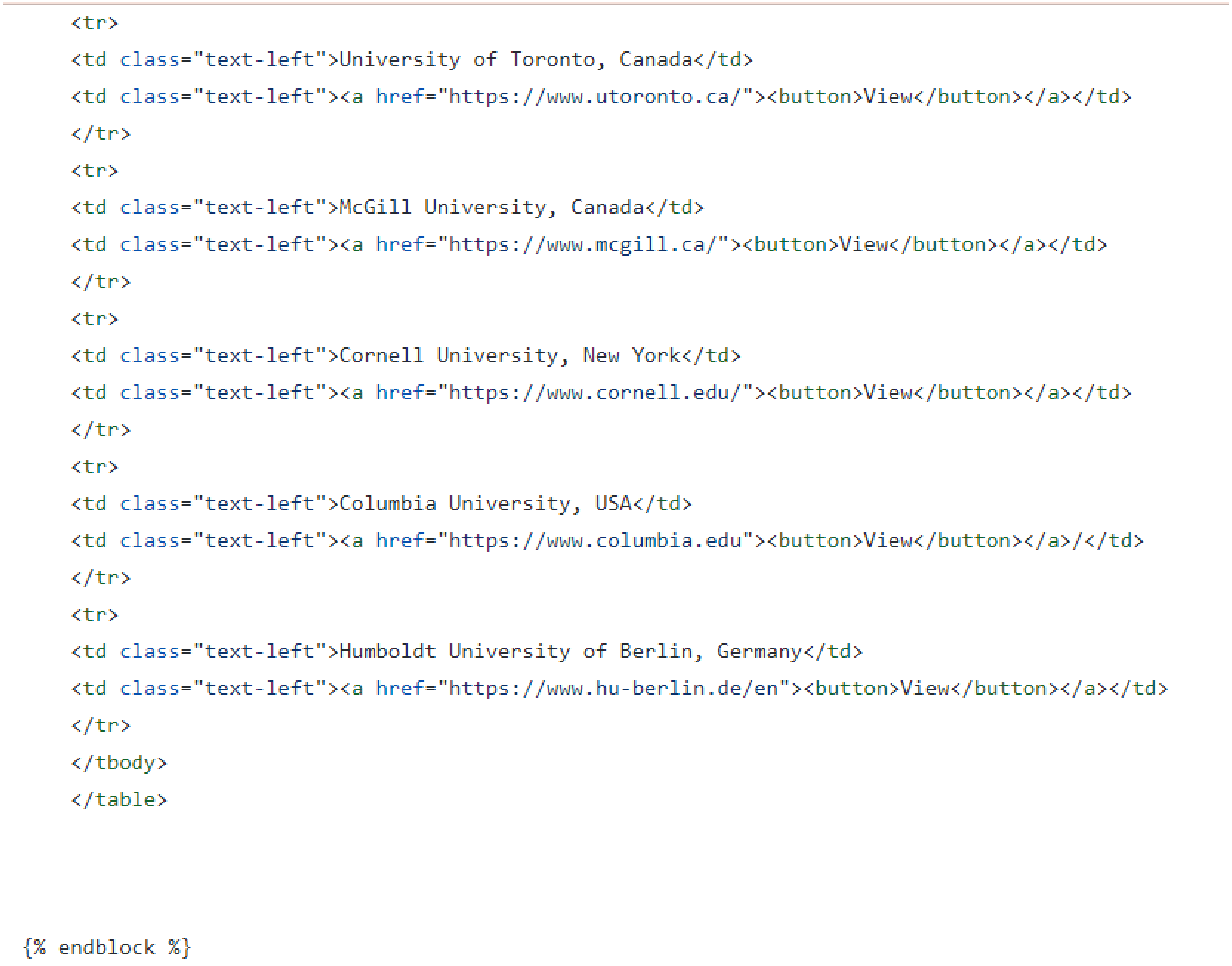


```
const button = document.getElementById('button');
const theForm = document.getElementById('theForm');
const loading = document.getElementById('spinner');

const disableButton = () => {
    console.log('Submitting form...');
    button.disabled = true;
    button.className = "btn btn-outline-primary";
    button.innerHTML = "Predicting..."
    loading.style.display = "block"
};

const enableButton = () => {
    console.log('Loading window...');
    button.disabled = false;
    button.className = "btn btn-primary"
    button.innerHTML = "Predict"
    loading.style.display = "none"
}

theForm.onsubmit = disableButton;

window.onload = enableButton;
```

CHANCE(HTML)

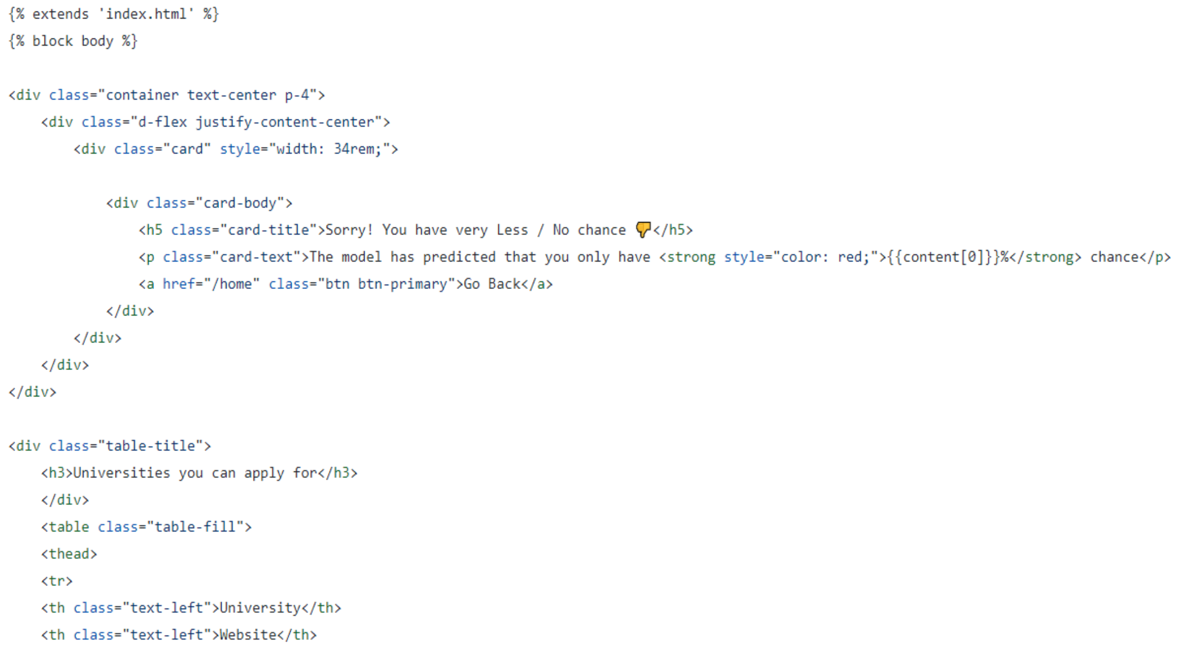


```
{% extends 'index.html' %}

{% block body %}

<div class="container text-center p-4">
    <div class="d-flex justify-content-center">

        <div class="card" style="width: 34rem;">

            <div class="card-body">
                <h5 class="card-title">You have great chance 👍</h5>
                <p class="card-text">The model has predicted that you have <strong style="color: green">{{content[0]}}%</strong> chance</p>
                <a href="/home" class="btn btn-primary">Go Back</a>
            </div>
        </div>
    </div>
</div>

<div class="table-title">
    <h3>Best universities for you to apply</h3>
    </div>
    <table class="table-fill">
    <thead>
    <tr>
    <th class="text-left">University</th>
    <th class="text-left">Website</th>
```

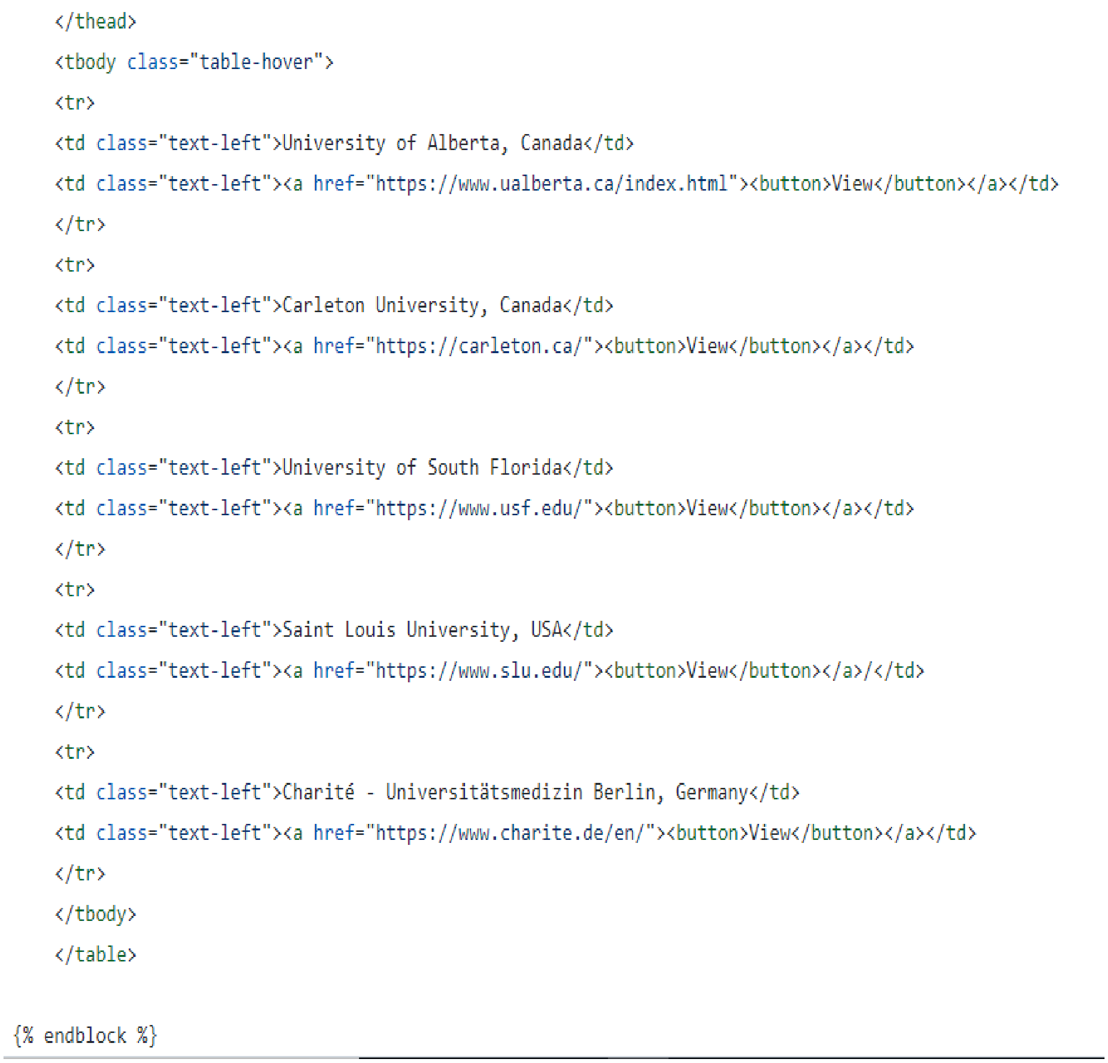


```
    <tr>
    <td class="text-left">University of Toronto, Canada</td>
    <td class="text-left"><a href="https://www.utoronto.ca/"><button>View</button></a></td>
    </tr>
    <tr>
    <td class="text-left">McGill University, Canada</td>
    <td class="text-left"><a href="https://www.mcgill.ca/"><button>View</button></a></td>
    </tr>
    <tr>
    <td class="text-left">Cornell University, New York</td>
    <td class="text-left"><a href="https://www.cornell.edu/"><button>View</button></a></td>
    </tr>
    <tr>
    <td class="text-left">Columbia University, USA</td>
    <td class="text-left"><a href="https://www.columbia.edu"><button>View</button></a>/</td>
    </tr>
    <tr>
    <td class="text-left">Humboldt University of Berlin, Germany</td>
    <td class="text-left"><a href="https://www.hu-berlin.de/en"><button>View</button></a></td>
    </tr>
    </tbody>
    </table>



{% endblock %}
```

NOCHANCE(HTML)

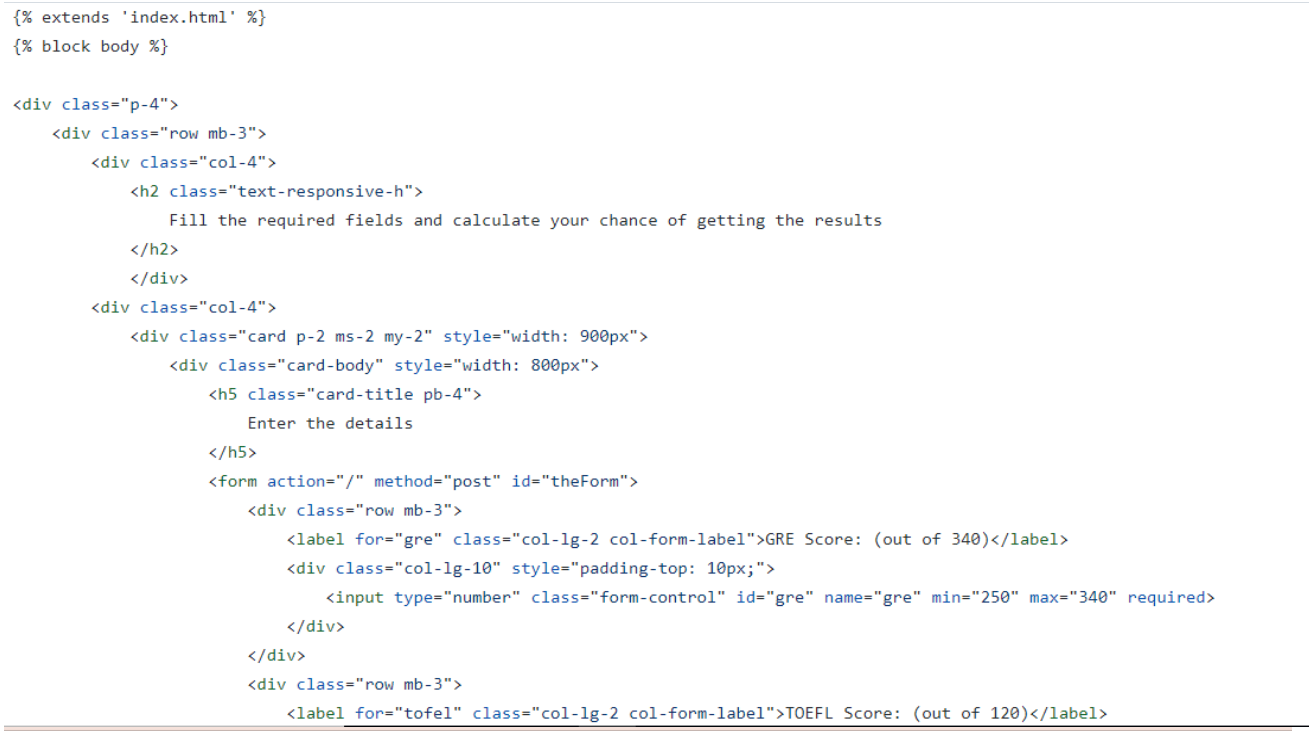


```
{% extends 'index.html' %}
{% block body %}

<div class="container text-center p-4">
    <div class="d-flex justify-content-center">
        <div class="card" style="width: 34rem;">

            <div class="card-body">
                <h5 class="card-title">Sorry! You have very Less / No chance 👎</h5>
                <p class="card-text">The model has predicted that you only have <strong style="color: red;">{{content[0]}}%</strong> chance</p>
                <a href="/home" class="btn btn-primary">Go Back</a>
            </div>
        </div>
    </div>
</div>

<div class="table-title">
    <h3>Universities you can apply for</h3>
    </div>
    <table class="table-fill">
    <thead>
    <tr>
    <th class="text-left">University</th>
    <th class="text-left">Website</th>
```

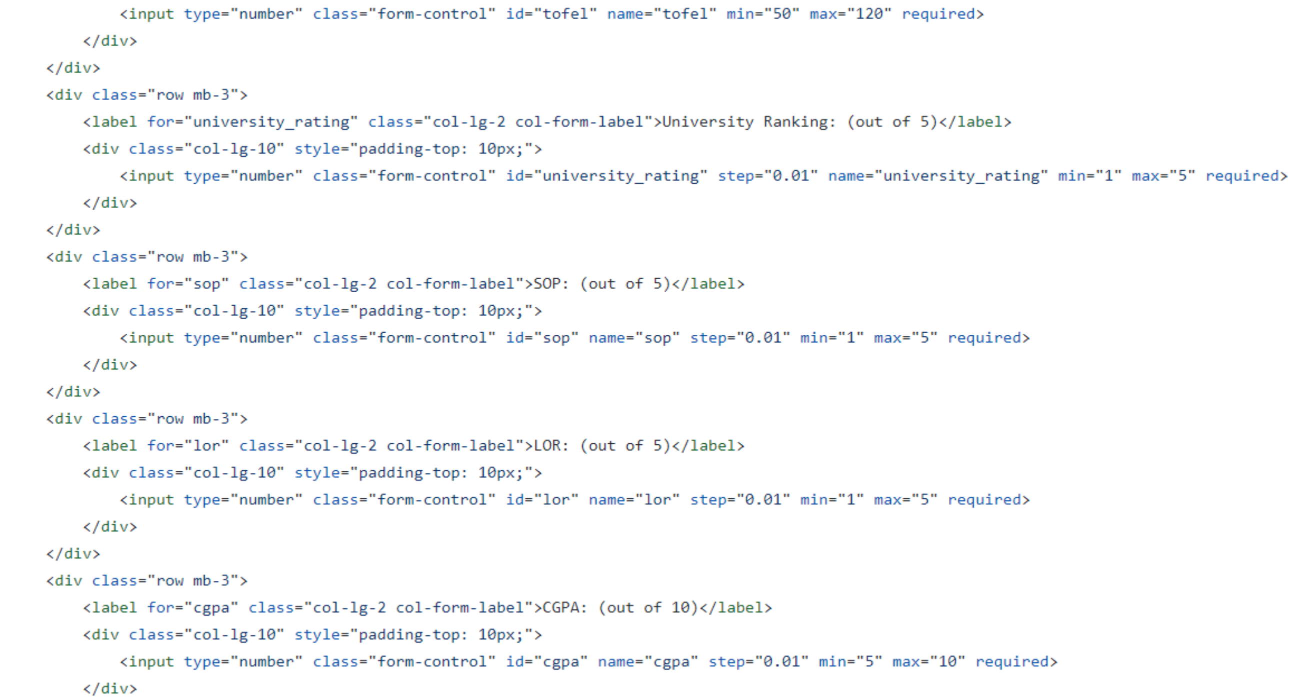


```
    </thead>
    <tbody class="table-hover">
    <tr>
    <td class="text-left">University of Alberta, Canada</td>
    <td class="text-left"><a href="https://www.ualberta.ca/index.html"><button>View</button></a></td>
    </tr>
    <tr>
    <td class="text-left">Carleton University, Canada</td>
    <td class="text-left"><a href="https://carleton.ca/"><button>View</button></a></td>
    </tr>
    <tr>
    <td class="text-left">University of South Florida</td>
    <td class="text-left"><a href="https://www.usf.edu/"><button>View</button></a></td>
    </tr>
    <tr>
    <td class="text-left">Saint Louis University, USA</td>
    <td class="text-left"><a href="https://www.slu.edu/"><button>View</button></a>/</td>
    </tr>
    <tr>
    <td class="text-left">Charité - Universitätsmedizin Berlin, Germany</td>
    <td class="text-left"><a href="https://www.charite.de/en/"><button>View</button></a></td>
    </tr>
    </tbody>
    </table>

{% endblock %}
```

RESULT(HTML)



```
{% extends 'index.html' %}
{% block body %}

<div class="p-4">
    <div class="row mb-3">
        <div class="col-4">
            <h2 class="text-responsive-h">
                Fill the required fields and calculate your chance of getting the results
            </h2>
            </div>
        <div class="col-4">
            <div class="card p-2 ms-2 my-2" style="width: 900px">
                <div class="card-body" style="width: 800px">
                    <h5 class="card-title pb-4">
                        Enter the details
                    </h5>
                    <form action="/" method="post" id="theForm">
                        <div class="row mb-3">
                            <label for="gre" class="col-lg-2 col-form-label">GRE Score: (out of 340)</label>
                            <div class="col-lg-10" style="padding-top: 10px;">
                                <input type="number" class="form-control" id="gre" name="gre" min="250" max="340" required>
                            </div>
                        </div>
                        <div class="row mb-3">
                            <label for="tofel" class="col-lg-2 col-form-label">TOEFL Score: (out of 120)</label>
```



```
        <input type="number" class="form-control" id="tofel" name="tofel" min="50" max="120" required>
    </div>
</div>
<div class="row mb-3">
    <label for="university_rating" class="col-lg-2 col-form-label">University Ranking: (out of 5)</label>
    <div class="col-lg-10" style="padding-top: 10px;">
        <input type="number" class="form-control" id="university_rating" step="0.01" name="university_rating" min="1" max="5" required>
    </div>
</div>
<div class="row mb-3">
    <label for="sop" class="col-lg-2 col-form-label">SOP: (out of 5)</label>
    <div class="col-lg-10" style="padding-top: 10px;">
        <input type="number" class="form-control" id="sop" name="sop" step="0.01" min="1" max="5" required>
    </div>
</div>
<div class="row mb-3">
    <label for="lor" class="col-lg-2 col-form-label">LOR: (out of 5)</label>
    <div class="col-lg-10" style="padding-top: 10px;">
        <input type="number" class="form-control" id="lor" name="lor" step="0.01" min="1" max="5" required>
    </div>
</div>
<div class="row mb-3">
    <label for="cgpa" class="col-lg-2 col-form-label">CGPA: (out of 10)</label>
    <div class="col-lg-10" style="padding-top: 10px;">
        <input type="number" class="form-control" id="cgpa" name="cgpa" step="0.01" min="5" max="10" required>
    </div>
```

```
                                    <label class="form-check-label" for="yes_no_radio">
                                    Yes
                                    </label>
                                </div>
                                <div class="form-check" style="padding-left: 150px;">
                                    <input class="form-check-input" type="radio" name="yes_no_radio" id="gridRadios2" value="0" checked>
                                    <label class="form-check-label" for="yes_no_radio">
                                    No
                                    </label>
                                </div>
                            </div>
                        </fieldset>
                        <div class="row lg-3">
                            <div class="col-lg-2 mb-2 me-3" style="padding-left: 355px; padding-right: 0px">
                                <button type="submit" class="btn btn-primary" id="button">Predict</button>
                            </div>

                    </form>
                </div>
            </div>
        </div>
    </div>
</div>


{% endblock %}
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-13386-1659517817

PROJECT DEMO

https://drive.google.com/file/d/1OOWfa9hXLedSOwGsWyfjHkoY1Itfnffo/view?usp=sharing