UNIVERSITY ADMIT ELIGIBLITY PREDICTOR

A PROJECT REPORT

Submitted by

PATHAN MUNA	**210419104118**
SHEEBA SANDHYA	**210419104156**
BOMMANA BOINA MEGHA PRIYA	**210419104036**
ANDENA HARICHANDANA	**210419104013**

TEAM ID: PNT2022TMID24674

CHENNAI INSTITUTE OF TECHNOLOGY

TABLE OF CONTENTS:

1 . INTRODUCTION

1.1 : Project Overview

✔ A Web page is designed for the students where they can check their Admission Eligibility for Universities based on their ranks.

✔ This Web page will get various parameters from students and will calculate the student's eligibility for that university.

1.2 : Purpose

The Purpose of this project is

✔ To reduce the work load of the user and also the use of paper.

✔ To enable the online Eligibility Checking for the Universities.

✔ To reduce the work load of the students.

✔ It will Automatically calculate the chance of the students.

2 . LITERATURE SURVEY

2.1 : Existing Problem

✔ Students who need to check their chances of getting admission in the Universities.

✔ Students can visit the web site and check their chances.

✔ And the Existing Problem can provide the high probability chances for the students who wish to get Admission in the University.

✔ This Existing Problem is user friendly for the students who can

visit this site.

2.2 : Problem Statement

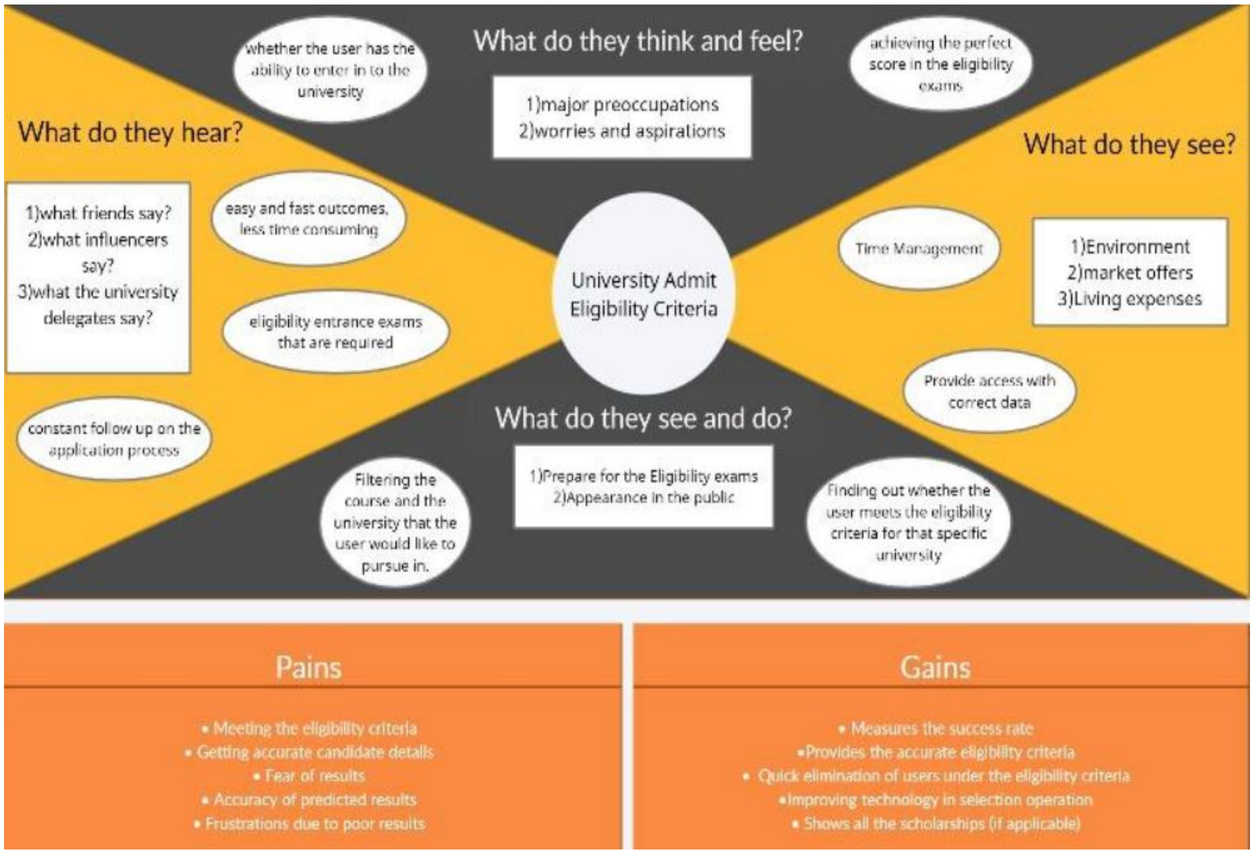
✔ Students need to check their Admission Eligibility in the respective Universities.

✔ Students can put their mark details that will calculate and provide the probable chances.

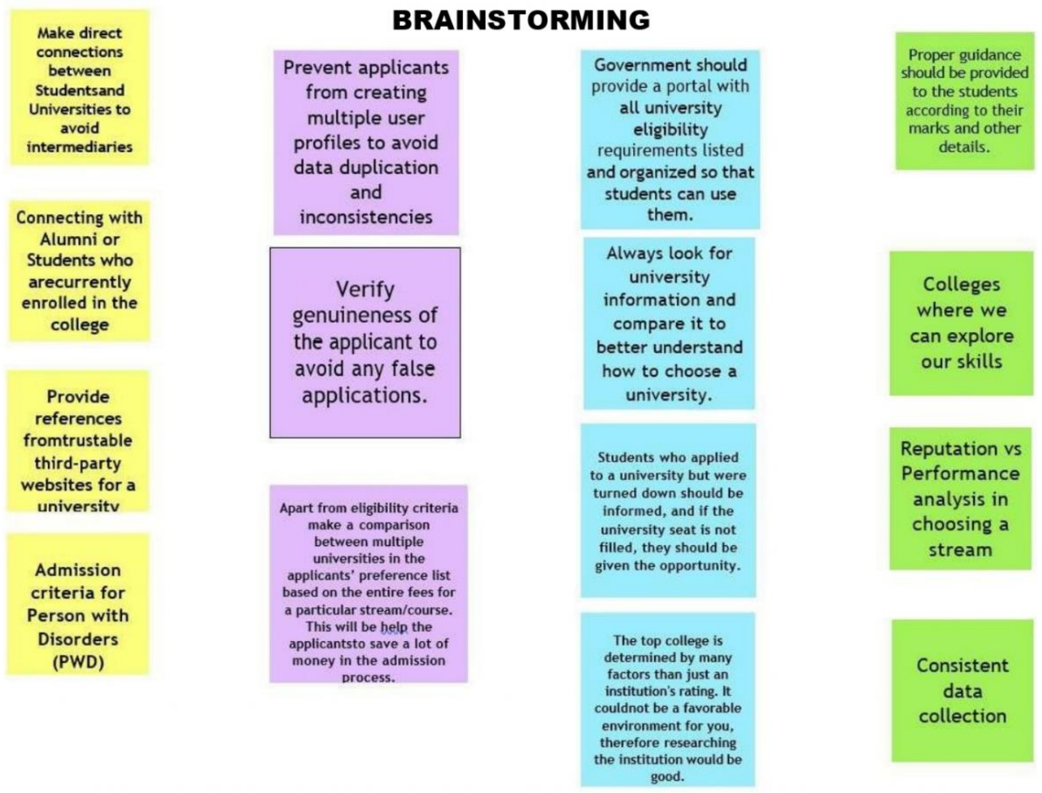
✔ Students need this platform to get the idea about the Admissions for the Universities.

3. IDEATION & PROPOSED SOLUTION

3.1 : Empathy Map Canvas


What do they think and feel?
whether the user has the ability to enter in to the university
1)major preoccupations
2)worries and aspirations
achieving the perfect score in the eligibility exams
What do they hear?
1)what friends say?
2)what influencers say?
3)what the university delegates say?
easy and fast outcomes, less time consuming
eligibility entrance exams that are required
constant follow up on the application process
University Admit Eligibility Criteria
What do they see?
Time Management
1)Environment
2)market offers
3)Living expenses
Provide access with correct data
What do they see and do?
1)Prepare for the Eligibility exams
2)Appearance in the public
Filtering the course and the university that the user would like to pursue in.
Finding out whether the user meets the eligibility criteria for that specific university
Pains
• Meeting the eligibility criteria
• Getting accurate candidate details
• Fear of results
• Accuracy of predicted results
• Frustrations due to poor results
Gains
• Measures the success rate
•Provides the accurate eligibility criteria
• Quick elimination of users under the eligibility criteria
•Improving technology in selection operation
• Shows all the scholarships (if applicable)

3.2 : Ideation and Brainstorming


BRAINSTORMING

Make direct connections between Studentsand Universities to avoid intermediaries

Connecting with Alumni or Students who arecurrently enrolled in the college

Provide references fromtrustable third-party websites for a university

Admission criteria for Person with Disorders (PWD)

Prevent applicants from creating multiple user profiles to avoid data duplication and inconsistencies

Verify genuineness of the applicant to avoid any false applications.

Apart from eligibility criteria make a comparison between multiple universities in the applicants' preference list based on the entire fees for a particular stream/course. This will be help the applicantsto save a lot of money in the admission process.

Government should provide a portal with all university eligibility requirements listed and organized so that students can use them.

Always look for university information and compare it to better understand how to choose a university.

Students who applied to a university but were turned down should be informed, and if the university seat is not filled, they should be given the opportunity.

The top college is determined by many factors than just an institution's rating. It couldnot be a favorable environment for you, therefore researching the institution would be good.

Proper guidance should be provided to the students according to their marks and other details.

Colleges where we can explore our skills

Reputation vs Performance analysis in choosing a stream

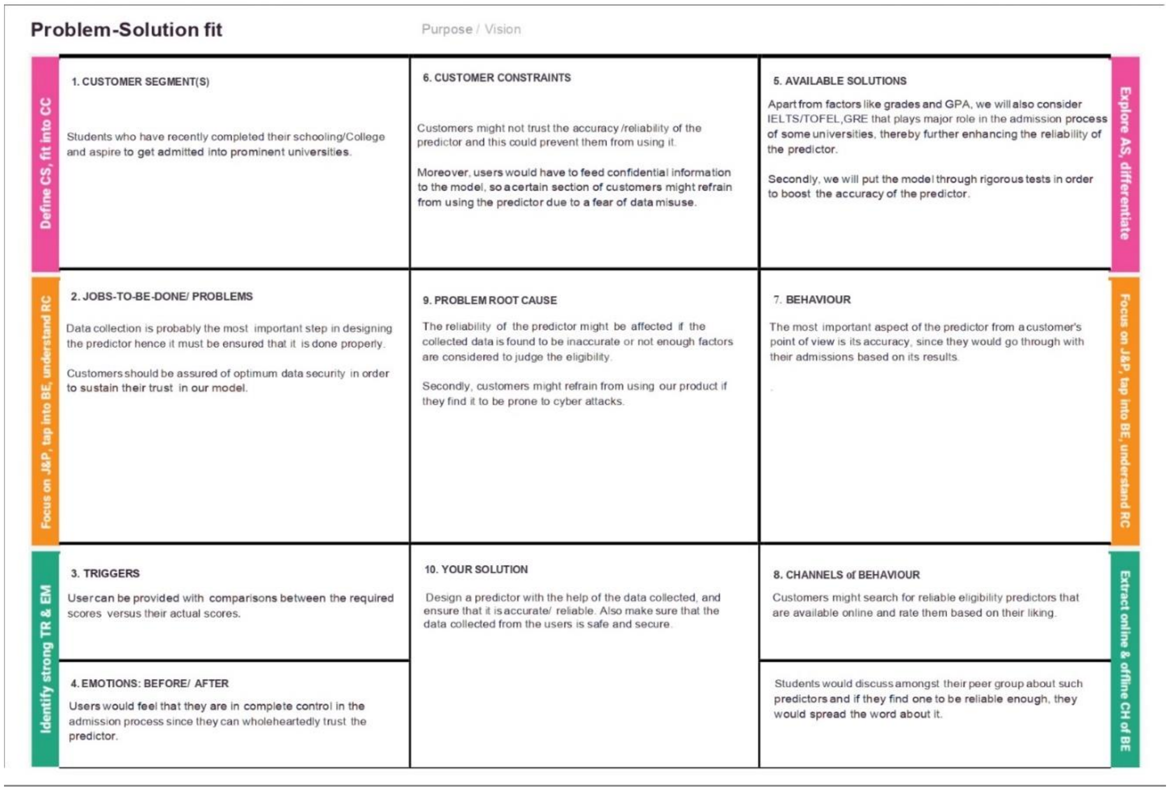
Consistent data collection

3.3 : Proposed Solution

S.No.	Parameter	Description
1.	**Problem Statement**	Students seek assistance from various educational consultancies to help them secure admission in the universities based on their profile because they are largely unaware of the procedures, requirements, and specifics of the institutions they wish to attend. In exchange, the students are expected to pay a significant amount as a consultancy fee.
2.	**Solution**	Depending on several factors, such as IELTS, the GRE, academic performance, etc., making an accurate projection of the student's admittance to the university of their choice.

3.	**Uniqueness**	It appears that there are no web tools that can forecast a student's eligibility requirements for admission to their ideal university and also offer tailored insights on particular areas where they might improve.
4.	**Social Impact**	It assists students in selecting the appropriate universities. The direct linkage between students and universities lowers the cost of consulting services.
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



Problem-Solution fit Purpose / Vision

Define CS, fit into CC	**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.	Explore AS, differentiate
Focus on J&P, tap into BE, understand RC	**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.	Focus on J&P, tap into BE, understand RC
Identify strong TR & EM	**3. TRIGGERS** User can be provided with comparisons between the required scores versus their actual scores.	**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.	Extract online & offline CH of BE
	4. EMOTIONS: BEFORE/ AFTER Users would feel that they are in complete control in the admission process since they can wholeheartedly trust the predictor.		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.	

4.REQUIREMENT ANALYSIS

4.1 : Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Administration work	Check qualified candidate detail Make allotment
FR-2	Admission Details	Check seat availability Check college infrastructure Check fees details
FR-3	Local counsellor	Issue the final allotment order

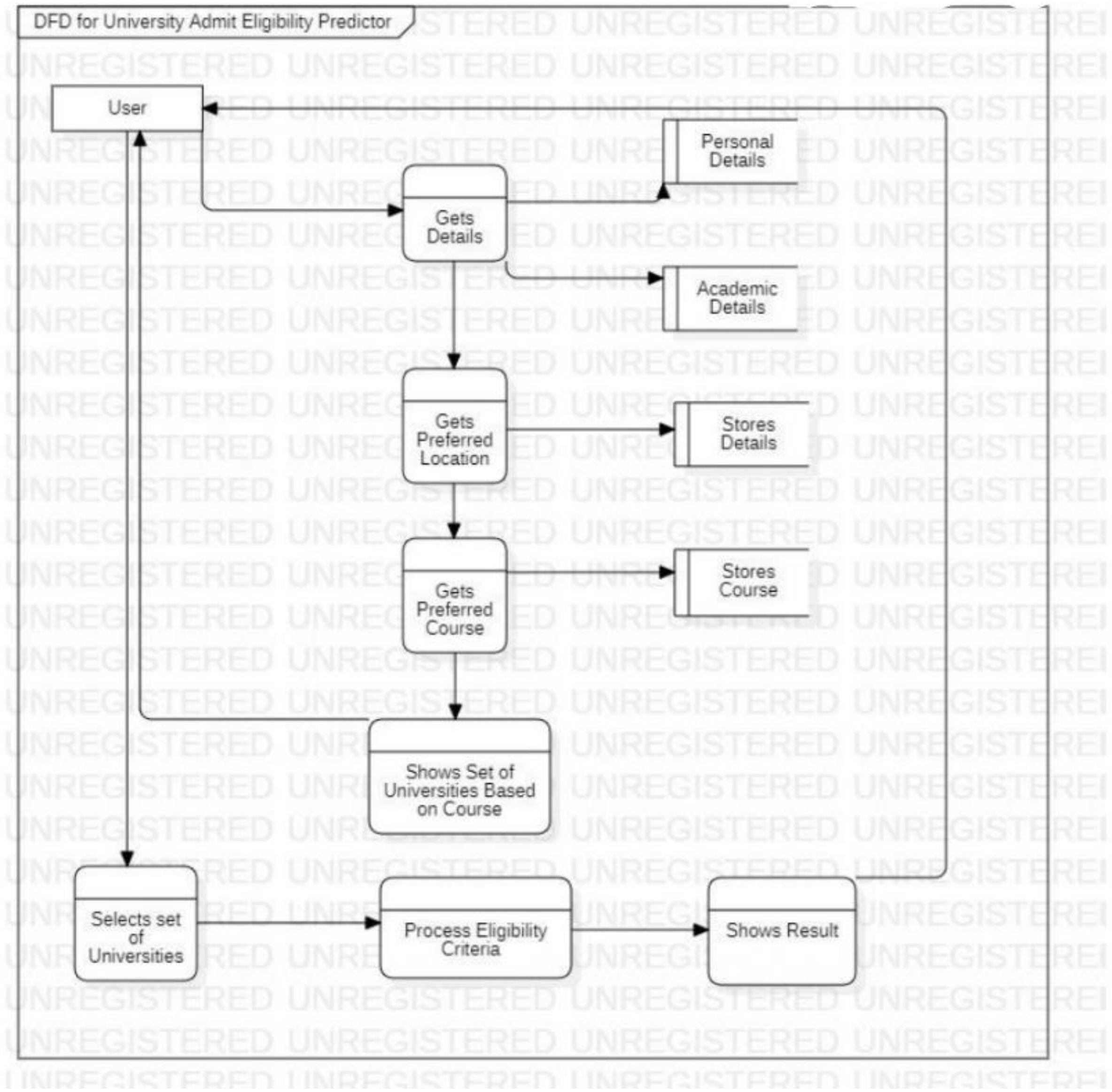
4.2: Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	i. A logical interface is essential to make easy use of system, speeding up common tasks. ii. The product could be used by two categories of people mainly administrator category and other users.
NFR-2	**Security**	Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below: i. Keep specific log or history data sets. ii. Utilize certain cryptographic techniques. iii. Restrict the no of systems that can access the online admission system site. This could be done only by registering the systems physical addresses before using them for online admission process. iv. Check data integrity for critical variables. v. Every user should be licensed to use the system under any of the four categories provided i.e. either verifier or advisor or local counsellor or administrator. vi. Communication needs to be restricted when the application is validating the user or license.
NFR-3	**Reliability**	i. All data storage for user variables will be committed to the database at the time of entry. ii. Data corruption is prevented by applying the possible backup procedures and techniques.

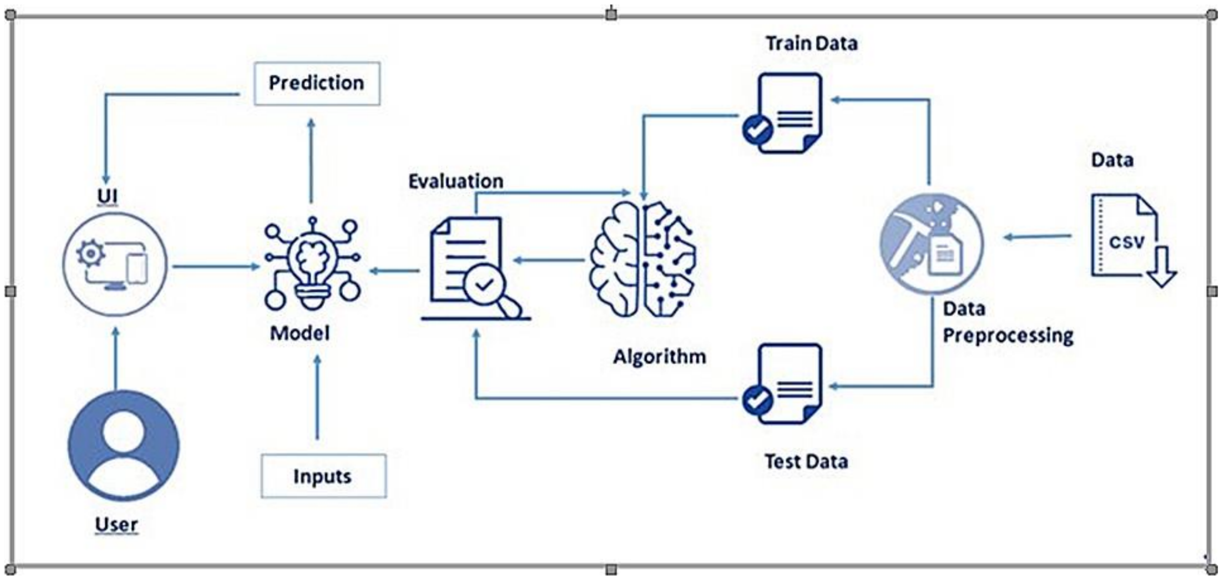
NFR-4	**Performance**	i. The database should be able to accommodate a minimum of 10,000 records of students. ii. At any instant the system should support use of multiple users at a time. iii. Availability results of the requested college should be presented to the student in max of two seconds, so retrieving of data should be reliable. iv. As each student will be given a maximum time of 10min, accessing from the database should be done at relevant speed.
NFR-5	**Availability**	The system should available at all the time meaning that the user can access easily. Increase of the hardware and data base failure a replacement page will be show and for database back should be retrieved from data folder.
NFR-6	**Scalability**	Assesses the highest workloads under which the system will still meet the performance Deals with the measure of the system's response time under different load conditions requirements. Example: The system must be scalable enough to support 1,000,000 visits at the same time while maintaining optimal performance.

5 . PROJECT DESIGN

5.1 : Data Flow Diagrams


DFD for University Admit Eligibility Predictor
User
Personal Details
Gets Details
Academic Details
Gets Preferred Location
Stores Details
Gets Preferred Course
Stores Course
Shows Set of Universities Based on Course
Selects set of Universities
Process Eligibility Criteria
Shows Result

5.2 : Solution &Technical Architecture



5.3: USER STORIES

User Stories

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

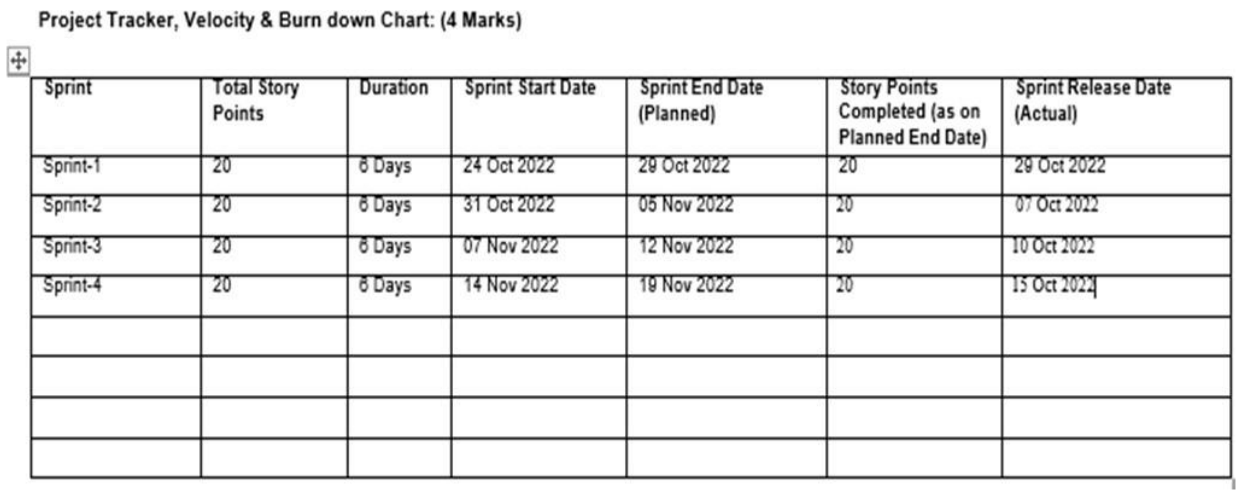
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Personal Details	USN-1	As a user, I can Give my academic information in the profile section	I can access my dashboard	High	Sprint-1
		USN-2	As a user, I will be able to select a location that I prefer	I can receive the list of location in the dropdown to select	High	Sprint-1
	Search	USN-3	As a user I can search for my preferred university	I can use the search bar	Medium	Sprint-2
	User Preference	USN-4	As a user, I can select my preferred university from the list to check my eligibility for the particular university	I can use the dropdown list provided to select the university	Medium	Sprint-2
		USN-5	As a user, I can select my preferred location	I can select my preferred location	High	Sprint-1
		USN-6	As a user, I will be able to select my preferred Course	I can select a course from the dropdown list	Medium	Sprint-1
	Result	USN-7	As a user, I can view the list of universities that I am eligible in accordance to my preferred location	I can view the list of universities filtered by the model	High	Sprint-3
		USN-8	As a user, I can access the link to the university that I am eligible from the list	I can access the university link	Medium	Sprint-3
		USN-9	As a user, I can access the location link of the university that I am eligible from the list	I can access the university location link	Low	Sprint-3
		USN-10	From the list of universities, I can select and view the eligibility for the particular university	I can view the eligible university	Medium	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1: Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priorit
Sprint-1	User Details	USN-1	User would be able to check and provide the information needed for the eligibility prediction .	2	High
Sprint-1		USN-2	User would receive their predictions based on the accuracy of the input given.	1	High
Sprint-2		USN-3	As a user, I can check the eligibility criteria for various universities by uploading the necessary documents	2	Low
Sprint-3		USN-4	As a user, I would be able to view the exact percentage value of my eligibility criteria.	2	Mediun
Sprint-4	Final Prediction	USN-5	Once I enter the accurate information into the responsive website, I as the user may see the outcomes of my prediction presented.	1	High
	Dashboard		Check dashboard for further updates and upload the details according to the desired and eligible universities based on the eligibility criteria.		

6.2 : Sprint Delivery Schedule



Project Tracker, Velocity & Burn down Chart: (4 Marks)

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	20	6 Days	31 Oct 2022	05 Nov 2022	20	07 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	10 Oct 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	15 Oct 2022

7 . CODING & SOLUTIONING

<u>UNIVERSITY ADMIT ELIGIBILITY PREDICTOR</u>

Description:

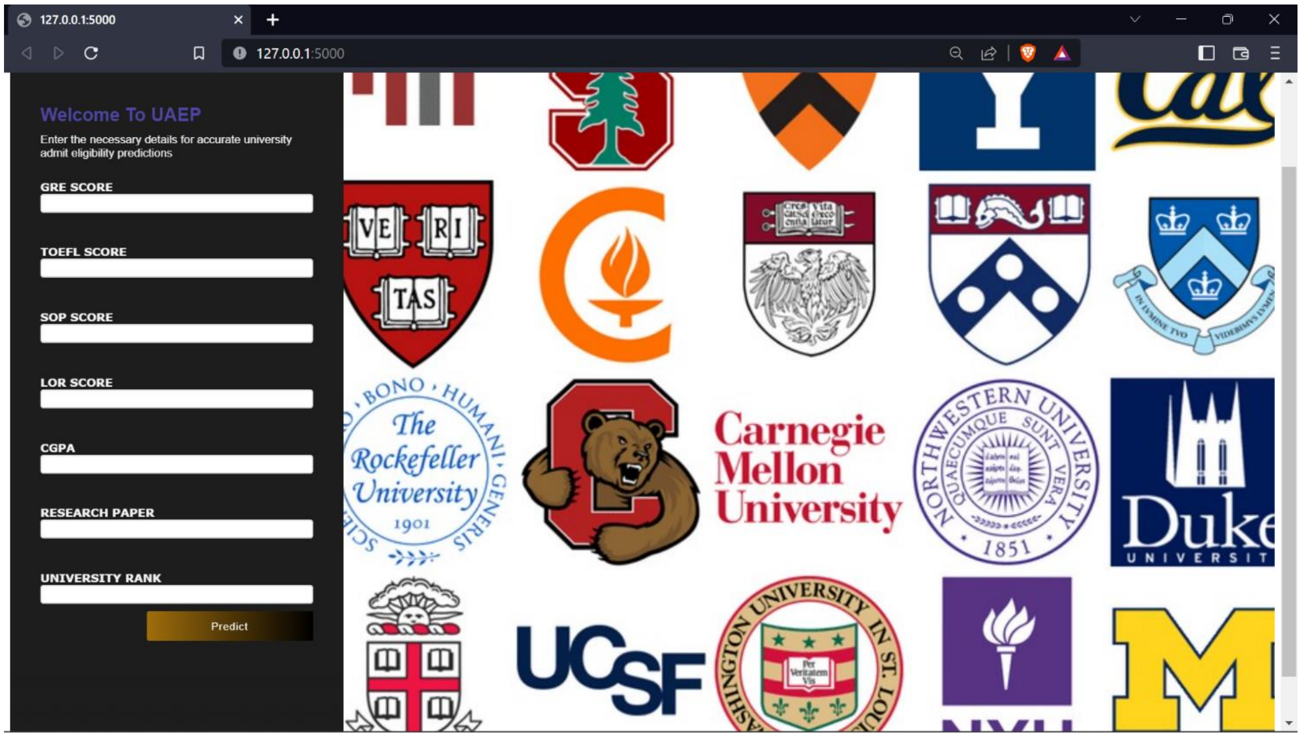
An individual can check their eligibility for the university using this project. In order to do so, the user must first provide his or her information, including the GRE, LOR, TOEFL, SOP, CGPA, details of his research paper, and the university's ranking. The user can therefore determine whether they are eligible or not depending on the evaluation. We are using Linear Regression Model to evaluate the details given by the user for accurate predictions.

The user can now view his findings in terms of percentage, which is the final step.

The user does not need to log in for this. The user can simply enter their scores in the form that is provided. If a person is qualified, they will be forwarded to a success page where they can view their results. If not, the user will be taken to a failure page where he can see his results.

In order to receive accurate predictions, the user can also choose to provide the scores in decimal format.

INDEX PAGE:



127.0.0.1:5000

Welcome To UAEP

Enter the necessary details for accurate university admit eligibility predictions

GRE SCORE

TOEFL SCORE

SOP SCORE

LOR SCORE

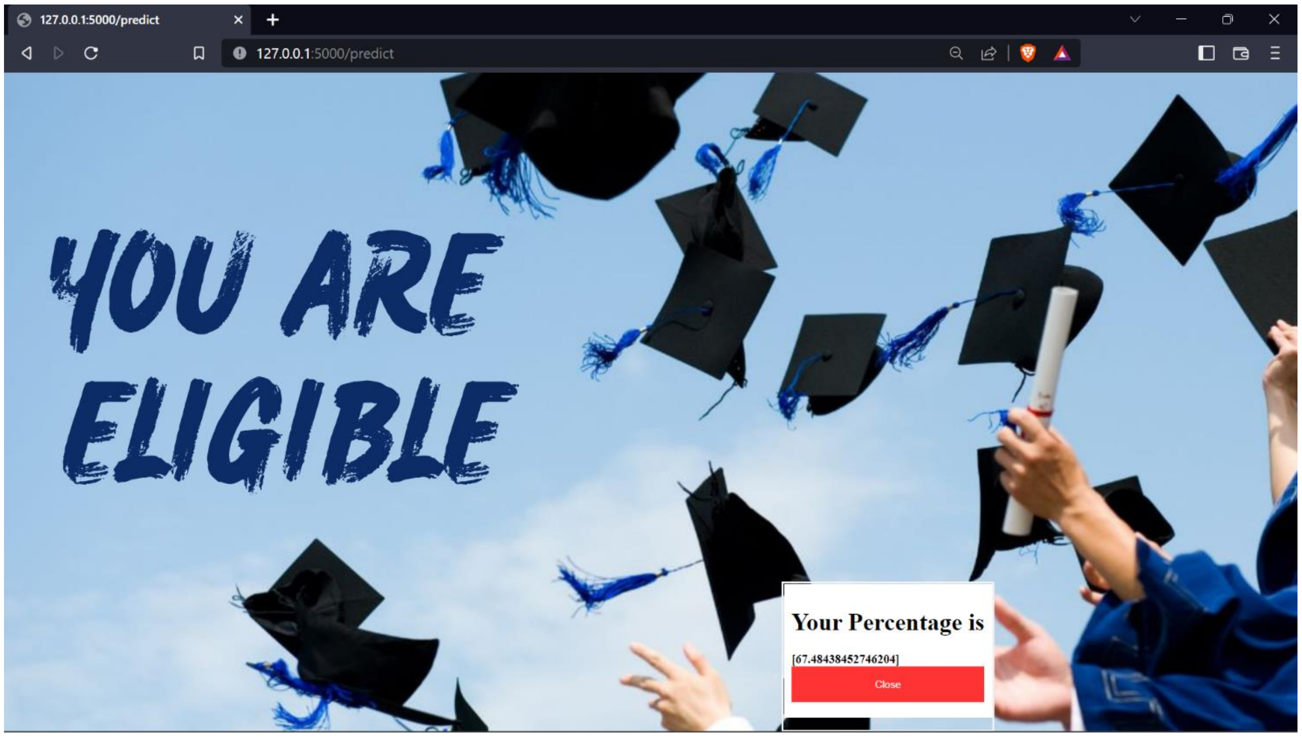
CGPA

RESEARCH PAPER

UNIVERSITY RANK

Predict

CHANCE PAGE:



127.0.0.1:5000/predict

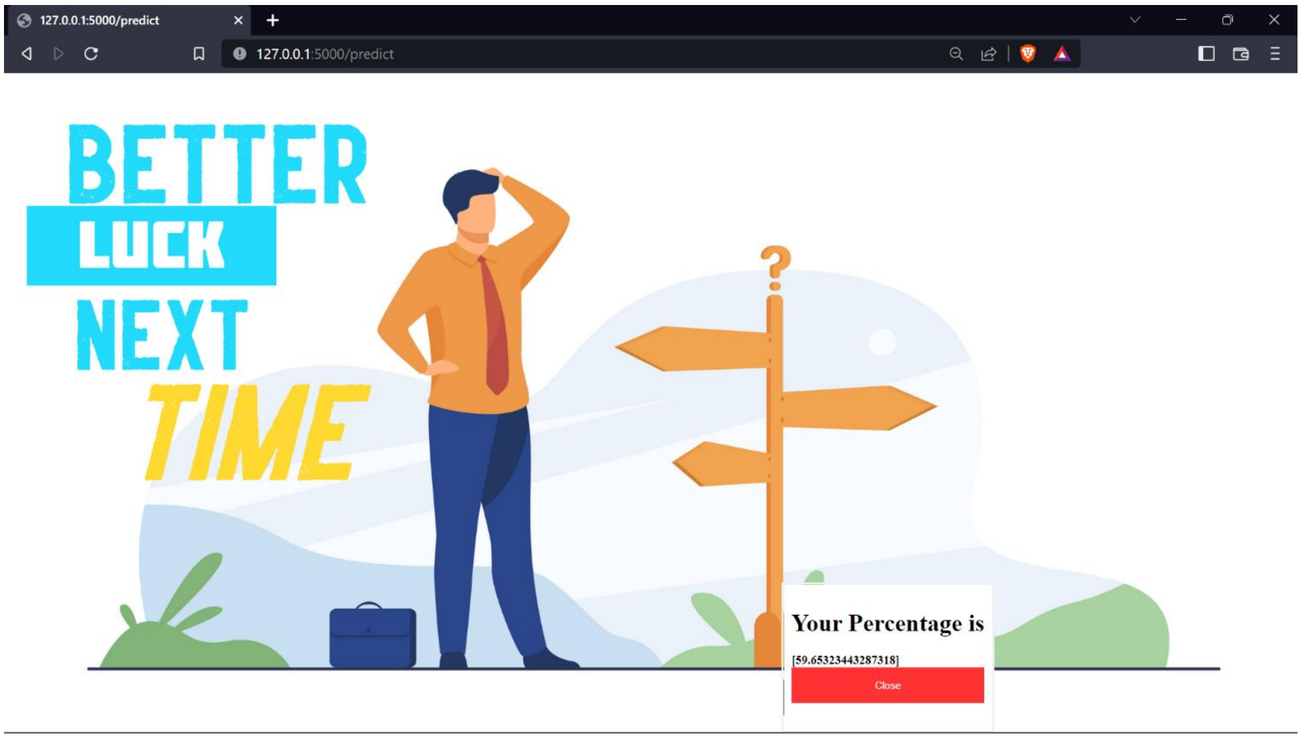
YOU ARE ELIGIBLE

Your Percentage is

[67.48438452746204]

Close

NO_CHANCE PAGE:



Your Percentage is

[59.65323443287318]

APP.PY

```
from flask import Flask, render_template, request
import numpy as np
import pickle
import pandas
import os
app=Flask (__name__)
model = pickle.load(open ('UAEP MAIN\model\Linear_Regression.pkl', 'rb'))
@app.route('/') # rendering the html template
def form1():
    return render_template('form1.html')
@app.route('/prediction',methods=["POST","GET"])


@app.route('/predict', methods = [ "POST","GET"])# route to show the predictions in
a web UI def
def predict():
   input_feature=[float(x) for x in request.form.values() ]
    #input_feature = np.transpose(input_feature)
   input_feature=[np.array(input_feature)]
   print(input_feature)
   names = ['GRE SCORE', 'TOFEL SCORE', 'SOP SCORE', 'LOR SCORE', 'CGPA', 'RESEARCH
PAPER', 'UNIVERSITY RANK']
   data = pandas.DataFrame(input_feature, columns=names)
   print(data)
   #data_scaled = scale.fit_transform(data) #data = pandas.DataFrame(,
columns=names)
   # predictions using the loaded model file prediction=model.predict(data)
   prediction=model.predict(data)
   print (prediction)
   prediction = float(prediction)
   print(type(prediction))
   if (prediction >= 60):
      return render_template("sucess.html", prediction_text = prediction)
   else:
    #showing the prediction results in a UI
      return render_template("failure.html",  prediction_text = prediction)




if __name__=="__main__":
# app.run(host='0.0.0.0', port=8000, debug=True)
  port=int(os.environ.get('PORT',5000))
  app.run(debug=False)
```

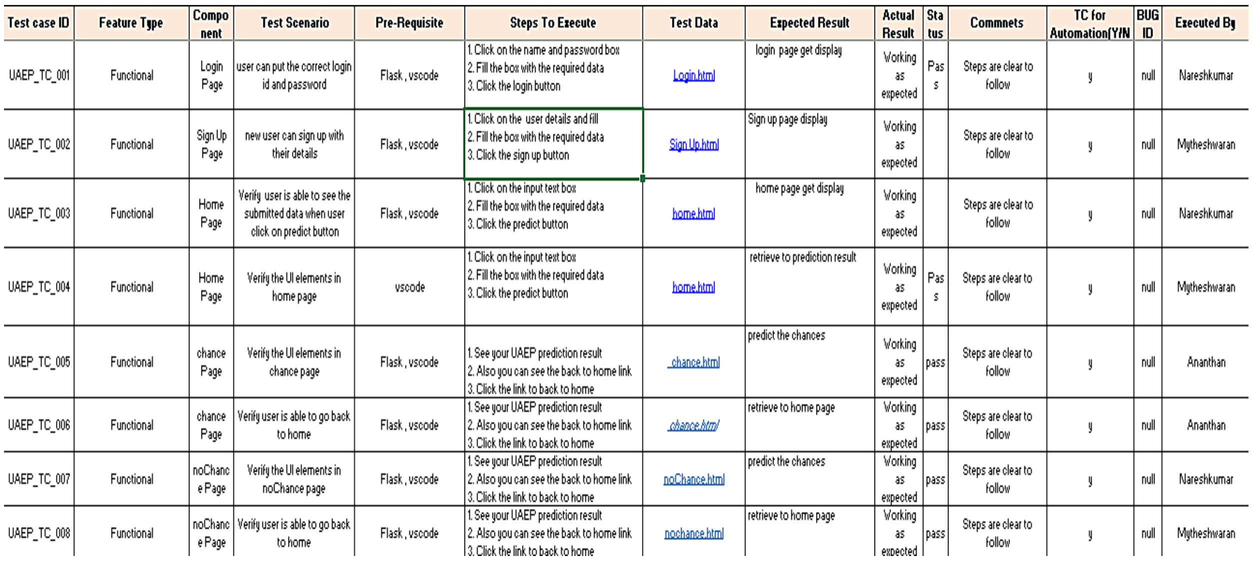
APP_IBM.PY

```
from flask import Flask, render_template, request
from flask_cors import CORS
import numpy as np
import pickle
import pandas
import os
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your
IBM Cloud account.
API_KEY = "5krRUFWdHgYbO4G9dm85TYoeeTlXRfJtTPcF5wo6ZrPF"
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}
```

```
app=Flask (__name__)
CORS(app)
@app.route('/', methods =["GET"]) # rendering the html template
def form1():
    return render_template('form1.html')


@app.route('/predict', methods = [ "POST"])# route to show the predictions in a web
UI def
def predict():
  GRE_SCORE = int(request.form['gre'])
  TOFEL_SCORE = int(request.form['ielts'])
  SOP_SCORE = float(request.form['sop'])
  LOR_SCORE = float(request.form['lor'])
  CGPA = float(request.form['cgpa'])
  RESEARCH_PAPER = int(request.form['research paper'])
  UNIVERSITY_RANK = int(request.form['university rank'])

   #data_scaled = scale.fit_transform(data) #data = pandas.DataFrame(,
columns=names)
   # predictions using the loaded model file prediction=model.predict(data)
  payload_scoring = {"input_data": [{"fields": ['gre', 'ielts', 'sop', 'lor',
'cgpa', 'research paper', 'university rank'], "values": [[GRE_SCORE, TOFEL_SCORE,
SOP_SCORE, LOR_SCORE,CGPA,RESEARCH_PAPER,UNIVERSITY_RANK]]}]}
  response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/776e4e48-19ec-4dc8-81b8-
333f4fc115d6/predictions?version=2022-11-18',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})
  print("Scoring Responce")
  print(response_scoring.json())
  predictions = response_scoring.json()
  prediction = predictions['predictions'][0]['values'][0][0]
  if (int(prediction[0]) >= 60):
    return render_template("sucess.html", prediction_text = prediction )
  else:
    return render_template("failure.html",  prediction_text =prediction)




if __name__=="__main__":
# app.run(host='0.0.0.0', port=8000, debug=True)
#  port=int(os.environ.get('PORT',5000))
  app.run(debug=False)
```

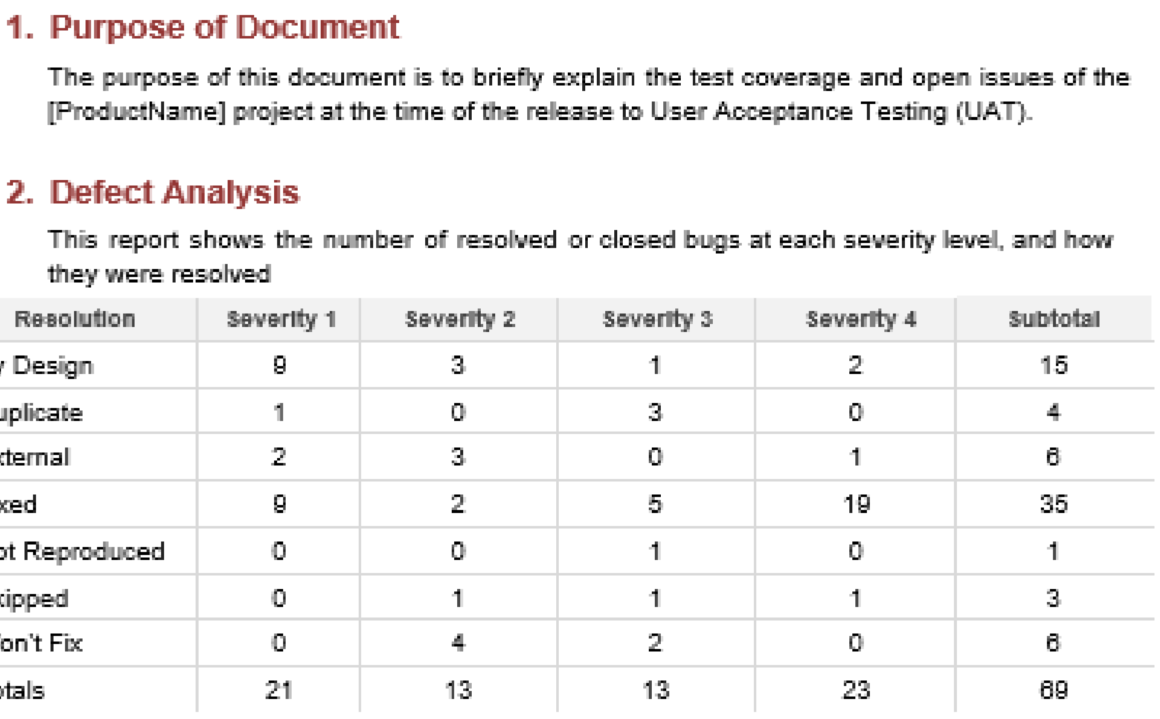
8 . TESTING

8.1 : Test Cases



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N	BUG ID	Executed By
UAEP_TC_001	Functional	Login Page	user can put the correct login id and password	Flask , vscode	1. Click on the name and password box 2. Fill the box with the required data 3. Click the login button	Login.html	login page get display	Working as expected	Pass	Steps are clear to follow	y	null	Nareshkumar
UAEP_TC_002	Functional	Sign Up Page	new user can sign up with their details	Flask , vscode	1. Click on the user details and fill 2. Fill the box with the required data 3. Click the sign up button	Sign Up.html	Sign up page display	Working as expected		Steps are clear to follow	y	null	Mytheshwaran
UAEP_TC_003	Functional	Home Page	Verify user is able to see the submitted data when user click on predict button	Flask , vscode	1. Click on the input text box 2. Fill the box with the required data 3. Click the predict button	home.html	home page get display	Working as expected		Steps are clear to follow	y	null	Nareshkumar
UAEP_TC_004	Functional	Home Page	Verify the UI elements in home page	vscode	1. Click on the input text box 2. Fill the box with the required data 3. Click the predict button	home.html	retrieve to prediction result	Working as expected	Pass	Steps are clear to follow	y	null	Mytheshwaran
UAEP_TC_005	Functional	chance Page	Verify the UI elements in chance page	Flask , vscode	1. See your UAEP prediction result 2. Also you can see the back to home link 3. Click the link to back to home	chance.html	predict the chances	Working as expected	pass	Steps are clear to follow	y	null	Ananthan
UAEP_TC_006	Functional	chance Page	Verify user is able to go back to home	Flask , vscode	1. See your UAEP prediction result 2. Also you can see the back to home link 3. Click the link to back to home	chance.html	retrieve to home page	Working as expected	pass	Steps are clear to follow	y	null	Ananthan
UAEP_TC_007	Functional	noChance Page	Verify the UI elements in noChance page	Flask , vscode	1. See your UAEP prediction result 2. Also you can see the back to home link 3. Click the link to back to home	noChance.html	predict the chances	Working as expected	pass	Steps are clear to follow	y	null	Nareshkumar
UAEP_TC_008	Functional	noChance Page	Verify user is able to go back to home	Flask , vscode	1. See your UAEP prediction result 2. Also you can see the back to home link 3. Click the link to back to home	nochance.html	retrieve to home page	Working as expected	pass	Steps are clear to follow	y	null	Mytheshwaran

8.2 : User Acceptance Testing



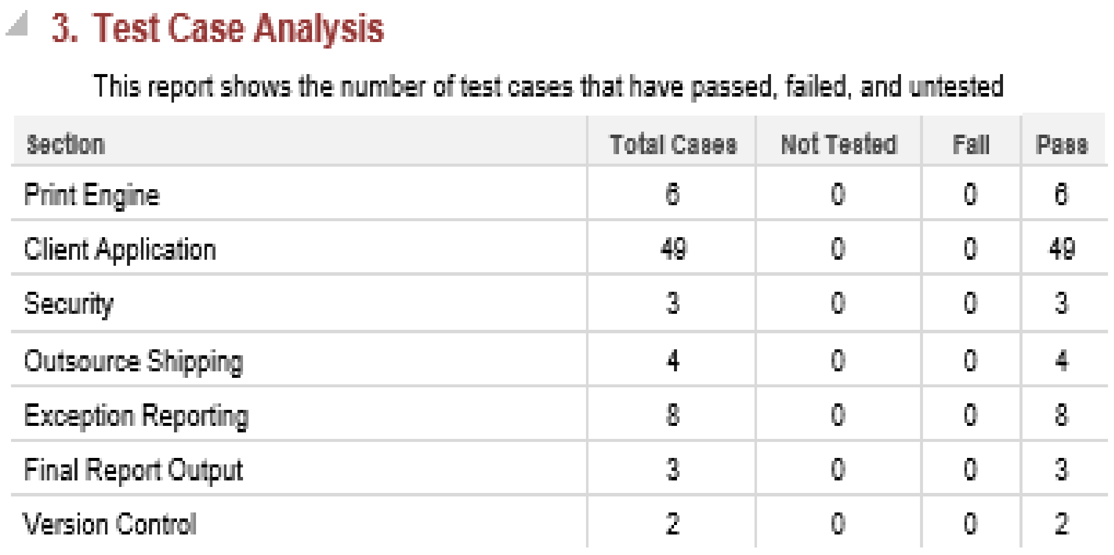
1. Purpose of Document

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	9	3	1	2	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	5	19	35
Not Reproduced	0	0	1	0	1
Skipped	0	1	1	1	3
Won't Fix	0	4	2	0	6
Totals	21	13	13	23	69



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
Print Engine	6	0	0	6
Client Application	49	0	0	49
Security	3	0	0	3
Outsource Shipping	4	0	0	4
Exception Reporting	8	0	0	8
Final Report Output	3	0	0	3
Version Control	2	0	0	2

9 . RESULTS

9.1 : Performance Metrices

Software quality is a measurement of something intangible, “how good” a software product really is. Some of the aspects of software quality taken are

i. Scalability

ii. Speed

iii. Stability

iv. Reliability

v. Security

vi. Maintainability and code quality

LOAD TEST

Scenario Name	Load Test – University Admit Eligibility Predictor
Scenario Type	Load Test – Duration 1 hour
Scenario Objective	To Simulate the peak load and to monitor the performance of the Website
Steps	The online load will be maintained at steady state
Entry Criteria	All the monitors are in ready state
Exit Criteria	Response met the criteria and test completion report is agreed

STRESS TEST

Scenario Name	Stress Test - University Admit Eligibility Predictor
Scenario Type	Stress Test
Scenario Objective	Objective is to verify that the application can handle the projected growth and to discover the breaking point
Steps	Ramp up to 150% of peak volume and continuously increase load until breaking point
Entry Criteria	All the monitors are in place Test Data is set up Peak load test completed successfully
Exit Criteria	Test completion report is agreed upon as per expectation

ENDURANCE / SOAK TEST

Scenario Name	Soak Test – University Admit Eligibility Predictor
Scenario Type	Endurance – Duration 8 hours
Scenario Objective	To discover memory issues and bottlenecks that might occur under daily usage of the application
Steps	Steady state is maintained for 8 hours with half of the peak load
Entry Criteria	All the monitors are in place Test Data is set up Peak load test completed successfully
Exit Criteria	Test completion report is agreed upon as per expectation

10 . ADVANTAGES

✔ 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.

✔ Very user-friendly and Easy accessibility of data.

DISADVANTAGES

✔ Academic Risks.

✔ Compliance Risks.

✔ Financial and Operational Risks.

✔ Reputational and Strategic Risks.

11 . CONCLUSION

✔ University Admit Eligibility Predictor is designed to reduce the work load of the user and also the use of paper.

✔ A Web page is designed for the users where they can check their admission prediction by providing the necessary details.

✔ Users can learn more about university admissions after entering the essential information on the prediction page.

✔ This University Admit Eligibility Predictor plays a significant role in outlining the precise eligibility requirements of current universities and helps the user shape to their future.

12 . FUTURE SCOPE

✔ Future indeed holds for big data science as there is a huge amount of data in the age of the internet, waiting to be utilized to make predictions, decisions and inventions.

✔ One of the most significant benefits of the University Admit Eligibility Predictor is that they are always open for business and can accept the predictions.

✔ Universities can use this project to get their work done quickly and efficiently in order to provide admissions to students.

13 . APPENDIX

Source Code :

form.html :

```
<html>
<head>
<link rel="stylesheet" href='static/style.css'>
</head>
<body>
  <form action="/predict" method="post">
<div class="container">
  <div class="left">
    <div class="header">
      <h2 class="animation a1">Welcome To UAEP</h2>
      <h4 class="animation a2">Enter the necessary details for accurate university
admit eligibility predictions</h4>
       </div>
        <label>GRE SCORE </label>
        <input type="number" class="form-field" max="340" min="260" name="gre"
required><br><br>
        <label>TOEFL SCORE </label>
        <input type="number" class="form-field" max="120" min="0" name="ielts"
required><br><br>
        <label>SOP SCORE </label>
        <input type="number" step="any" class="form-field"   max="5" min="0"
name="sop" required><br><br>
        <label>LOR SCORE </label>
        <input type="number" step="any"  class="form-field"  max="5" min="0"
name="lor" required><br><br>
        <label>CGPA </label>
        <input type="number" step="any" class="form-field"  max="10" min="0"
name="cgpa" required><br><br>
        <label>RESEARCH PAPER </label>
        <input type="number" class="form-field" min="0" name="research paper"
required><br><br>
        <label>UNIVERSITY RANK </label>
        <input type="number" class="form-field" max="50" min="1" name="university
rank" required maxlength='5'>
    <button class="animation a6">Predict</button>
  </div>
  <div class="right"></div>
</div>
</form>
</body>
```

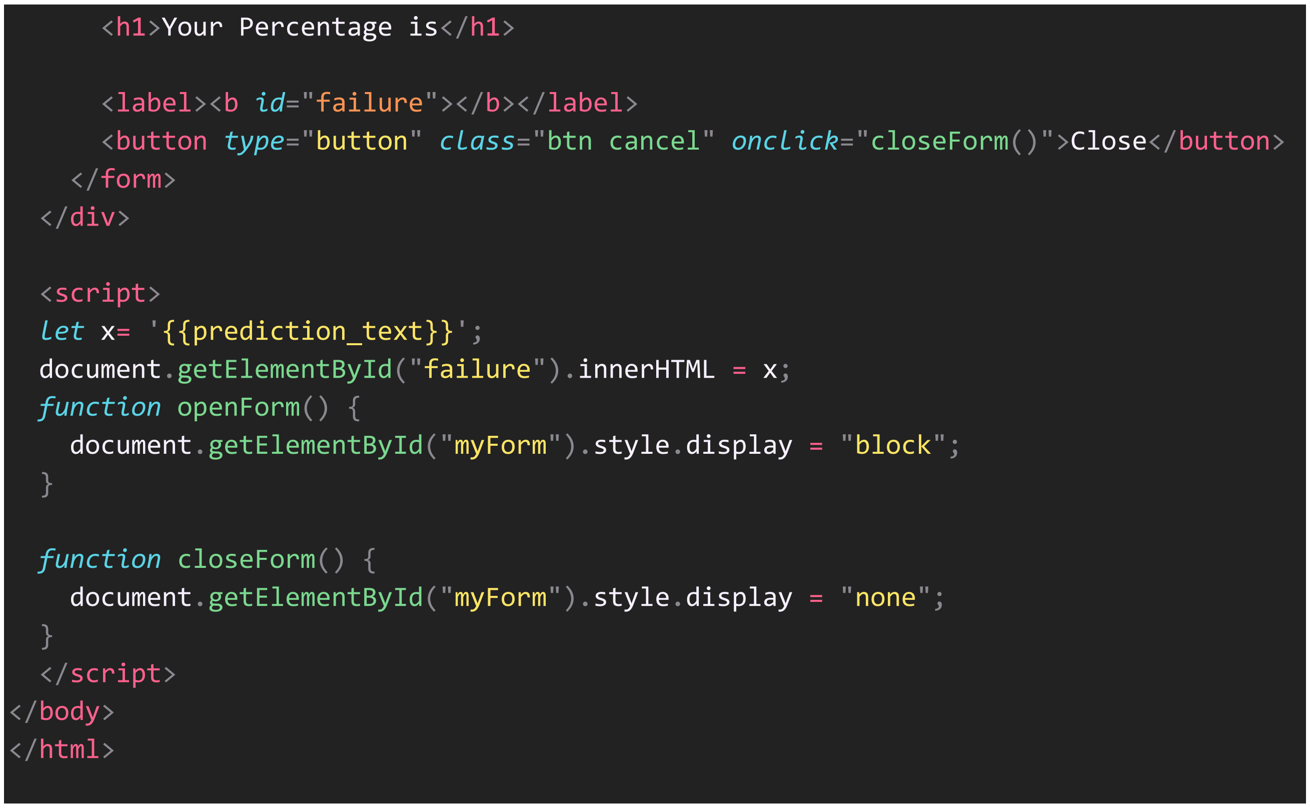
```
</html>
```

success.html :

```
<html>
  <head>
<style>
body {
  background-image: url('static/success.png');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
pre{

position: absolute;
bottom: 210px;

left: 30px;
font-size: 90px;
}
p{

  position: absolute;
  bottom: 200px;
  color:#154ba8;
  left: 300px;
font-size: 40px;
}
.open-button {
  background-color: #555;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  opacity: 0.8;
  position: fixed;
  bottom: 23px;
  right: 400px;
  width: 280px;
}

/* The popup form - hidden by default */
.form-popup {
  display: none;
  position: fixed;
  bottom: 0;
  right: 400px;
  border: 3px solid #f1f1f1;
  z-index: 9;
}

.form-container {
  max-width: 300px;
  padding: 10px;
  background-color: white;
}



.form-container .btn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  width: 100%;
  margin-bottom:10px;
  opacity: 0.8;
}

.form-container .cancel {
  background-color: red;
}

.form-container .btn:hover, .open-button:hover {
  opacity: 1;}
</style>
</head>
<body>
  <button class="open-button" onclick="openForm()">VIEW RESULTS</button>

  <div class="form-popup" id="myForm">
    <form class="form-container">
      <h1>Your Percentage is</h1>

      <label><b id="failure"></b></label>
      <button type="button" class="btn cancel" onclick="closeForm()">Close</button>
    </form>
  </div>

  <script>
  let x= '{{prediction_text}}';
  document.getElementById("failure").innerHTML = x;
  function openForm() {
    document.getElementById("myForm").style.display = "block";
  }

  function closeForm() {
    document.getElementById("myForm").style.display = "none";
  }
  </script>
</body>
</html>
```

failure.html :

```
<html>
  <head>
<style>
body {
  background-image: url('static/failure.png');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
pre{

position: absolute;
bottom: 210px;

left: 30px;
font-size: 90px;
}
p{

  position: absolute;
  bottom: 200px;
  color:#154ba8;
  left: 300px;
font-size: 40px;
}
.open-button {
  background-color: #555;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  opacity: 0.8;
  position: fixed;
  bottom: 23px;
  right: 400px;
  width: 280px;
}

/* The popup form - hidden by default */
.form-popup {
  display: none;
  position: fixed;
  bottom: 0;
  right: 400px;
  border: 3px solid #f1f1f1;
  z-index: 9;
}

.form-container {
  max-width: 300px;
  padding: 10px;
  background-color: white;
}



.form-container .btn {
  background-color: #04AA6D;
  color: white;
  padding: 16px 20px;
  border: none;
  cursor: pointer;
  width: 100%;
  margin-bottom:10px;
  opacity: 0.8;
}

.form-container .cancel {
  background-color: red;
}

.form-container .btn:hover, .open-button:hover {
  opacity: 1;}
</style>
</head>
<body>
  <button class="open-button" onclick="openForm()">VIEW RESULTS</button>

  <div class="form-popup" id="myForm">
    <form class="form-container">
```



```html
      <h1>Your Percentage is</h1>

      <label><b id="failure"></b></label>
      <button type="button" class="btn cancel" onclick="closeForm()">Close</button>
    </form>
  </div>

  <script>
  let x= '{{prediction_text}}';
  document.getElementById("failure").innerHTML = x;
  function openForm() {
    document.getElementById("myForm").style.display = "block";
  }

  function closeForm() {
    document.getElementById("myForm").style.display = "none";
  }
  </script>
</body>
</html>
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-50287-1660901957