UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID:PNT2022TMID23380

RAGHUL SAYEE K A D [113219031115]

KIRTHICK RAJ S [113219031074]

AMEER BATCHA S [113219031009]

DEEPAK S [113219031033]

CONTENTS

1. INTRODUCTION

1.1 OVERVIEW

The project is implemented using a Machine-Learning model that predicts whether the user is eligible for an admission in the selected rated universities with provided details such as marks and others. The algorithm works in such a way that when the user provides the details such as (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research) the percentage of chance of admit is displayed. The user is provided with a UI (Web based application) in which the user can enter the details mentioned above for prediction. The main advantage of this is that the user can avoid long process of having to check the eligibility of a university admission by himself and make use of this application to predict the eligibility / chance of admit.

1.2 PURPOSE

The purpose of this project is to make the prediction of eligibility of an admission to a rated university with ease using a UI with the provided user details (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research). This also eliminates the possibility of human errors.

2. LITERATURE SURVEY

2.1 Existing problem

Previous research done in this area used Naive Bayes algorithm which will evaluate the success probability of student application into a respective university but the main drawback is they didn't consider all the factors which will contribute in the student admission process like TOEFL/IELTS, SOP, LOR and under graduate score. Bayesian Networks Algorithm have been used to create a decision support network for evaluating the application submitted by foreign students of the university. This model was developed to forecast the progress of prospective students by comparing the score of students currently studying at university. The model thus predicted whether the aspiring student should be admitted to university on the basis of various scores of students. Since the comparisons are made only with students who got admission into the universities but not with students who got their admission rejected so this method will not be that much accurate.

2.2 References

● M. S. Acharya, A. Armaan, and A. S. Antony, “A Comparison of Regression Models for WrĞĔŝcΘŽn of Graduate Admissions,” Kaggle, 2018.

● M. N. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, “^yƐłĞmĂΘc ensemble model ƐĞůĞcΘŽn approach for ĞĔƵcĂΘŽnĂů data mining,” Knowledge-Based Syst., vol. 200, p. 105992, Jul. 2020.

● M. S. Acharya, A. Armaan, and A. S. Antony, “A comparison of regression models for ĐrĞĔŝcΘŽn of graduate admissions,” ICCIDS 2019 - 2nd Int. Conf. Comput. Intell. Data Sci. Proc., pp. 1–5, 2019.

● N. Chakrabarty, S. Chowdhury, and S. Rana, “A ^łĂΘƐΘcĂů Approach to Graduate Admissions’ Chance WrĞĔŝcΘŽn̰ ” no. March, pp. 145– 154, 2020.

● S. Sujay, “Supervised Machine Learning Modelling & Analysis for Graduate Admission WrĞĔŝcΘŽn̰ ” vol. 7, no. 4, pp. 5–7, 2020.

2.3 Problem Statement Definition

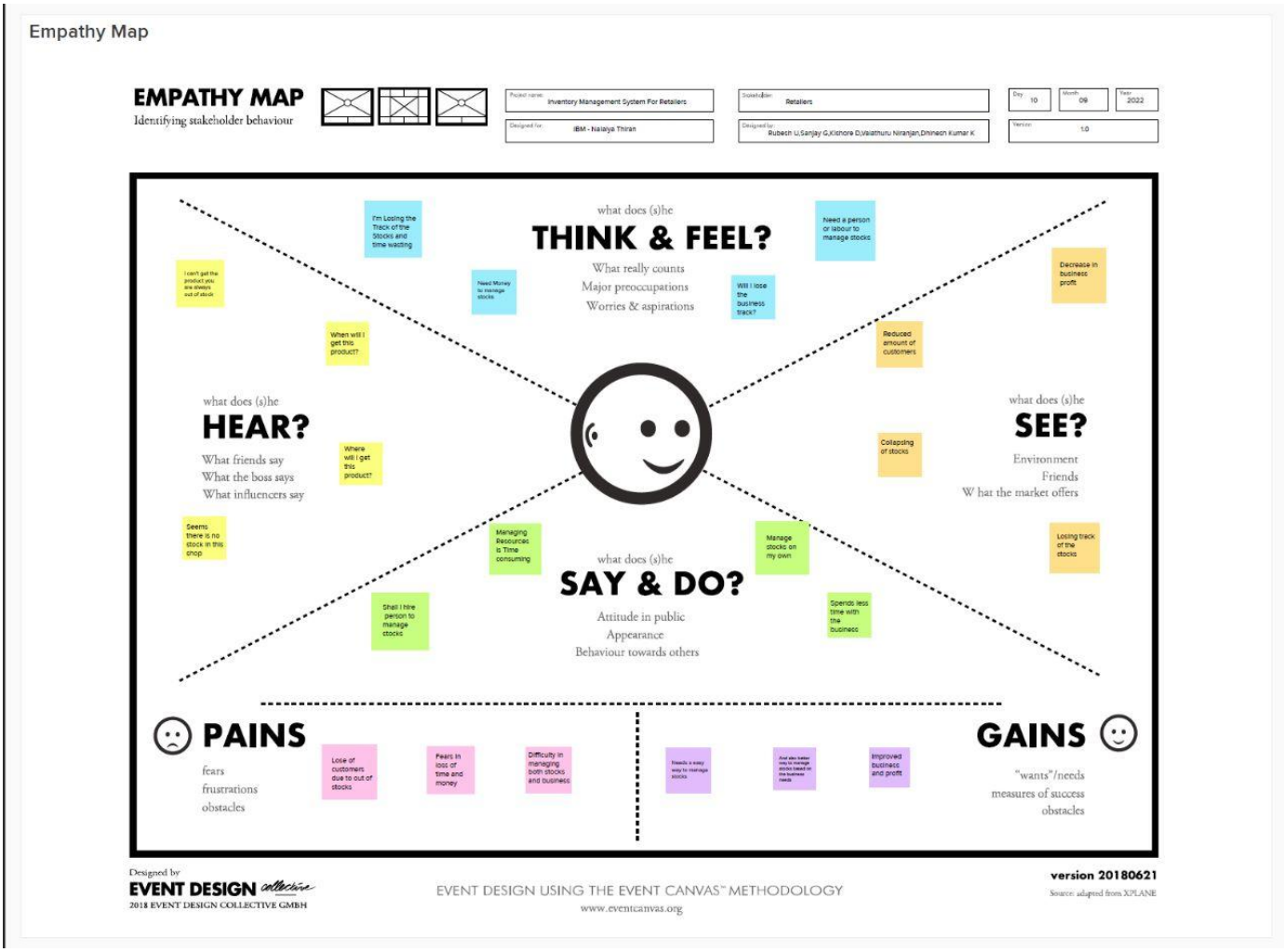
➔ In the current world scenario, it is not enough for a student to just have an Undergraduate degree. Most employers now look for higher qualifications in their new recruits. As a result, the demands for a good higher education are at an all time high. A lot of students from India prefer to continue their higher education with foreign universities.
➔ In order to get admitted to these foreign universities, a set of academic requirements are needed. However, because of the sheer number of universities of different levels, students are often stuck in a dilemma till the very last minute as to whether or not their applications will be accepted or not as no concrete documentation is available which lists the requirements.
➔ Generally the students don't have much idea about the procedures, requirements and details of the universities. They seek help from the education consultancy firms to help them successfully secure admission in the universities which are best suitable for their profile, for this they have to invest huge amount of money as consultancy fees. Apart from these the education consultancy firms there are few websites and blogs that guide the students on the admission procedures. The drawback of the currently available resources is that they are very limited and also they are not truly dependable taking into consideration of their accuracy and reliability.
➔ The aim of this project is to develop a system using machine learning algorithms. It will help the students to identify the chances of their application to a university being accepted. A simple user interface will

be developed for the users to access the system.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to 1) create a shared understanding of user needs, and 2) aid in decision making. Traditional empathy maps are split into 4 quadrants (Says, Thinks, Does, and Feels), with the user or persona in the middle. Empathy maps provide a glance into who a user is as a whole and are not chronological or sequential.



3.2 Ideation & Brainstorming

Brainstorming is a method design teams use to generate ideas to solve clearly defined design problems. In controlled conditions and a free-thinking environment, teams approach a problem by such means as “How Might We” questions. They produce a vast array of ideas and draw links between them to find potential solutions.

3.3 Proposed Solution

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools. Following are the objectives of the proposed system:

- Reach to geographically scattered student.
- Reducing time in activities
- Paperless admission with reduced man power
- Operational efficiency

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	While planning for education students often have several questions regarding the courses, universities, job opportunities, expenses involved, etc. Securing admission in their dream university is one of their main concerns. It is seen that often students prefer to pursue their education from universities which have global recognition. Students are often worried about their chances of admission to University.

2.	Idea / Solution description	The primary objective of this research is to develop a system to solve the problems the students are facing while applying for universities. We will be developing a University Admit Eligibility Predictor (UAEP) system which will help the students to predict the chances of their application being selected for a particular university for which they wish to apply based on their profile. Also, the system will provide a recommendation of universities to the student to which the student has a high possibility of getting admission.
3.	Novelty / Uniqueness	we will be creating a simple user interface which will help the users to input the data related to the student profile and get the predicted result for the application based on the profile as output. The aim of this project is to help students in shortlisting 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.
4.	Social Impact / Customer Satisfaction	This research will eventually help students save the extra amount of time and money they have to spend at the education consultancy firms. And also it will help them to limit their number of applications to a small number by providing them with the suggestion of the universities where they have the best chance of securing admission thus saving more money on the application fees.
5.	Business Model (Revenue Model)	Advertisements of different universities can be placed on this app from which we can earn money.
6.	Scalability of the Solution	In the future we will add one faculty member of each college which will help the student to ask question and clarify regarding the university, course, etc. we can also add a chat option in this app.

3.4 Problem Solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.

Problem-Solution fit

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.

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.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Details	Submit the documents • GRE or/and TOEFL scoresheet • Curriculum Vitae (CV) • Statement of Purpose (SoP) • Letter of Recommendation
FR-4	User Requirements	• Upload all the relevant documents in the appropriate location in the website • Based on the uploads, the system would scrape all the necessary information • The list of all possible university for the candidate would be displayed based on the scraped information

4.2 Non-Functional requirements

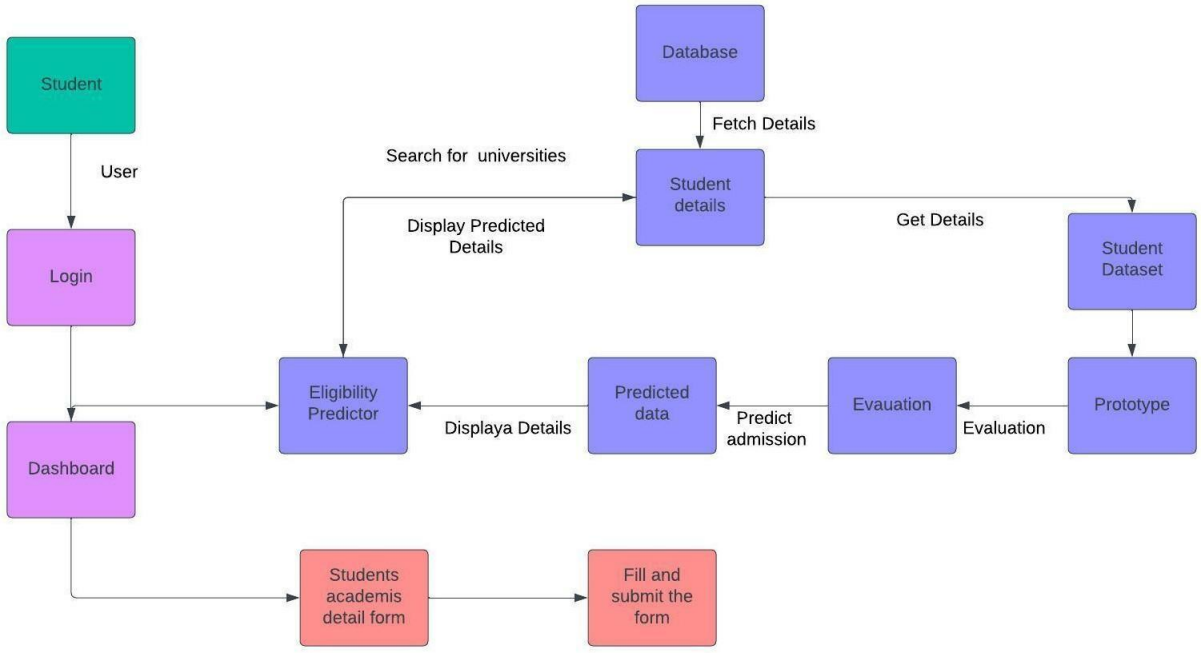
In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• The system doesn't expect any technical pre-requisite from the user i.e.; even the naïve user can access it • The UI would focus on recognize over recall • User friendly • Reduced focus on Short Term memory load Focus on Internal Locus of Control • The page would not take a lot of time to load the content and display them (< 30 seconds) • The fields in the site would be self-explanatory
NFR-2	**Security**	• Only the authenticated user would be able to utilize the services of the site. • Database should be backed up every hour
		• Under any error, the system should be able to come back to normal operation in under an hour.
NFR-3	**Reliability**	• The system would always strive for maximum reliability due to the importance of data and damages thar could be cause by incomplete and incorrect data • The system will run 7 days a week, 24 hours a day
NFR-4	**Performance**	• The website can efficiently handle the traffic by service the request as soon as possible • Viewing this webpage using a 56-kbps modem connection would not exceed 30 seconds (quantitatively, the mean time)
NFR-5	**Availability**	• Minimal data redundancy • Less prone to errors • Fast and efficient • The system will run 7 days a week, 24 hours a day
NFR-6	**Scalability**	• Since an academic portal is crucial to the courses that use it, it is crucial that a sizable number of users be able to access the system at the same time. • The admission season is probably when the system will be under the most strain. • It must therefore be able to manage numerous concurrent users.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

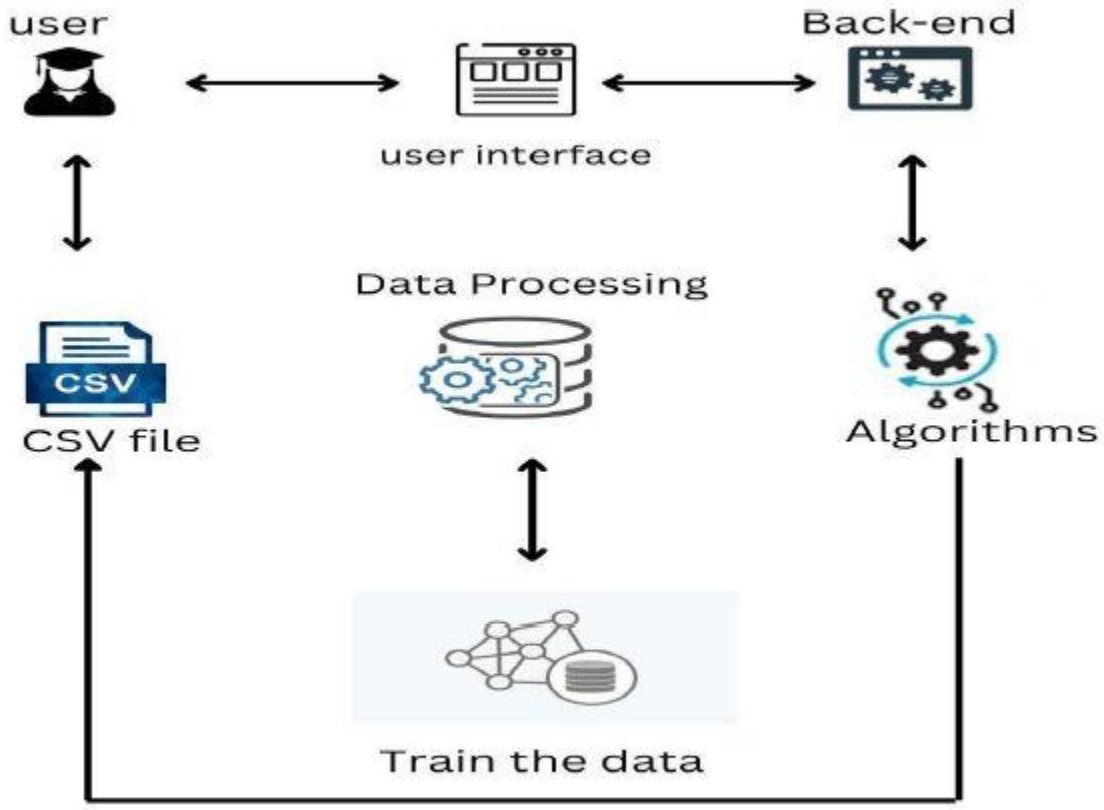


5.2 Solution & Technical Architecture

Solution Architecture:

A solution architecture (SA) is an architectural description of a specific solution. SAs combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).

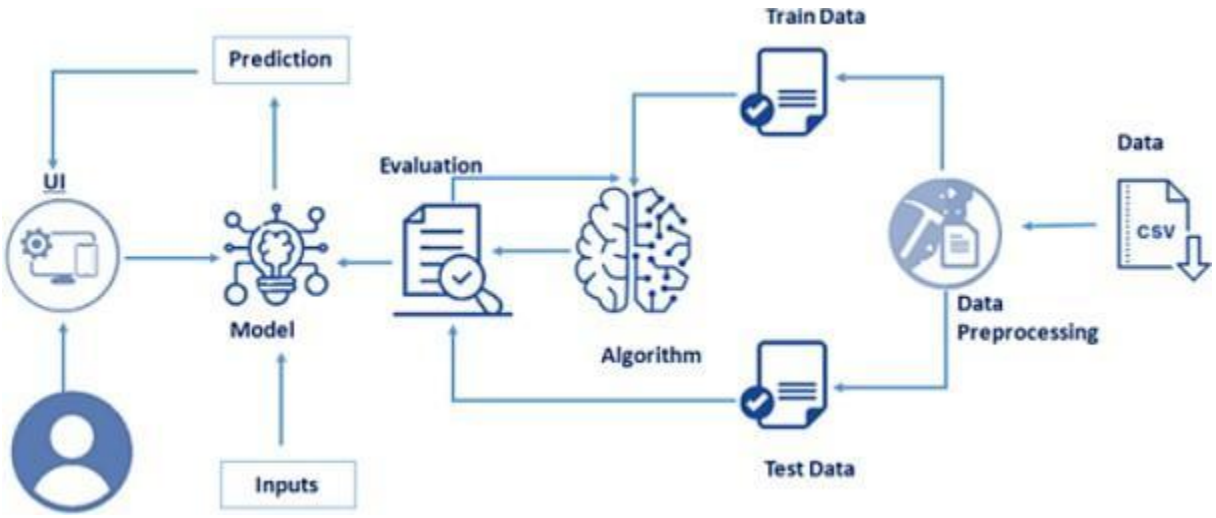
SOLUTION ARCHITECTURE DIAGRAM:



University admit eligibility predictor

Technical Architecture:

Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that systemrelevant requirements are met.



5.3 User Stories

A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Landing page	USN-1	As a user, I can view the details about the university	I can access the university landing page	Medium	Sprint-1
		USN-2	As a user, I can view the latest news about the university	I can access the latest news	Medium	Sprint-1
		USN-3	As a user, I can fill the contact form forqueries	I can fill and submit the contact form	Low	Sprint-2
		USN-4	As a user, I can see the social media profiles of the university	I can reach out to them via social media	Medium	Sprint-1
		USN-5	As a user, I can see testimonials of students who graduated from the university	I can access the testimonials	Medium	Sprint-1
	Admissions	USN-6	As a user, I can see the previous year cut-off marks	I can download the previous year cut-off details	High	Sprint-2
		USN-7	As a user, I can read about proud alumni of the university	I can access the details of alumni of the university	Medium	Sprint-2
		USN-8	As a user, I can predict my eligibility foradmission at the university	I can get result as either eligible/not eligible	High	Sprint-2
	Courses offered	USN-9	As a user, I can see the courses offered by the university for PG students	I can access the course details	Medium	Sprint-3
	Events	USN-10	As a user, I can check various technical events about to happen in the university	I can register for the events	Low	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

In Scrum Projects, Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

Product Backlog, Sprint Schedule, and Estimation:

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	2
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	1
Sprint-2		USN-3	As a user, I can check the eligibility criteria for various universities by uploading the necessary documents	2	Low	2
Sprint-3		USN-4	As a user, I can register for the desired university through Gmail and can also upload further course completion documents if necessary.	2	Medium	2
Sprint-4	User Login	USN-5	As a user, I can log into the application by entering email & password	1	High	2
	Dashboard		Check dashboard for further updates and upload the details according to the desired and eligible universities based on the eligibility criteria.			4

6.2 Sprint Delivery Schedule

A sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

Project Tracker, Velocity & Burndown Chart:

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

7. CODING & SOLUTIONING

7.1 Data Dictionary

Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1	337	118	4	4.5	4.5	9.65	1	0.92
2	324	107	4	4	4.5	8.87	1	0.76
3	316	104	3	3	3.5	8	1	0.72
4	322	110	3	3.5	2.5	8.67	1	0.8
5	314	103	2	2	3	8.21	0	0.65
6	330	115	5	4.5	3	9.34	1	0.9
7	321	109	3	3	4	8.2	1	0.75
8	308	101	2	3	4	7.9	0	0.68
9	302	102	1	2	1.5	8	0	0.5
10	323	108	3	3.5	3	8.6	0	0.45

7.2 Libraries Used

Pandas, numpy, scikit learn, matplotlib, seaborn.

7.3 Technologies

Used Software

·Python
·Anaconda
·Jupyter Notebook
·Windows 7 or higher
·XAMP Server

Hardware

Processor – Dual Core or higher
Hard Disk – 50 GB
Memory – 1GB RAM

7.4 Evaluation Metric

The evaluation metric for this competition is 100*RMSLE where RMSLE is Root of Mean Squared Logarithmic Error across all entries in the test set.

7.5 Initial Approach

- Simple Linear Regression model without any feature engineering and data transformation which gave a RMSE : 196.402

- Without feature engineering and data transformation, the model did not perform well and could'nt give a good score.

- Post applying feature engineering and data transformation (log and log1p transformation), Linear Regression model gave a RMSLE score of 0.734.

7.6 Advanced Models

- With improvised feature engineering, built advanced models using Ensemble techniques and other Regressor algorithms.

- Decision Tree Regressors performed well on the model which gave much reduced RMSLE.

- With proper hyper-parameter tuning, Decision Tree Regressor performed well on the model and gave the lease RMSLE of 0.5237.

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario
Main_Page_1	Functional	Main page	Verify user can able to access the page
Main_Page_2	UI	Main page	Verify user can able to access the page
Main_Page_3	UI	Main page	Verify user can able to access the page
Chance_Page_1	Functional	Chance Page	Verify user can able to access the page
Nochance_Page_1	Functional	Nochance Page	Verify user can able to access the page

Pre-Requisite	Steps To Execute	Test Data	Expected Result
	click the URL	http://127.0.0.1:5000/	Execute Successfully
	Give input in the input field	http://127.0.0.1:5000/	Execute Successfully
	click the predict button	http://127.0.0.1:5000/	Execute Successfully
	on clicking the predict button	http://127.0.0.1:5000/chance/92.	Execute Successfully
	on clicking the predict button	http://127.0.0.1:5000/nochance/49.	Execute Successfully

Actual Result	Status
Working as Expected	Pass
Working as Expected	Pass
Working as Expected	Pass
Working as Expected	Pass
Working as Expected	Pass

8.2 User Acceptance

Testing Defect Analysis:

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	7	3	2	3	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	7	3	2	17	29
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won'tFix	0	3	2	1	6
Totals	17	12	11	23	63

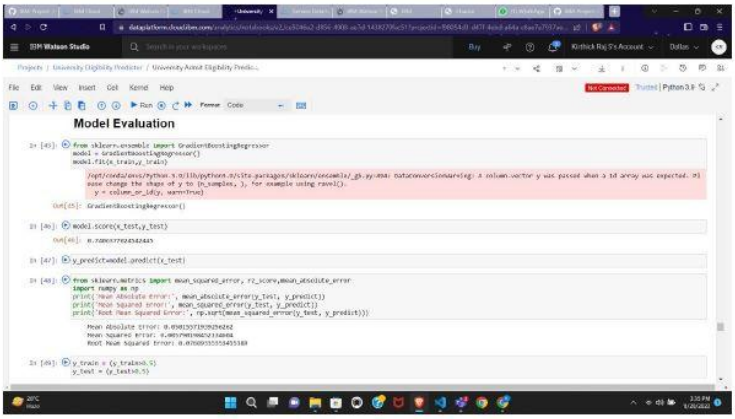
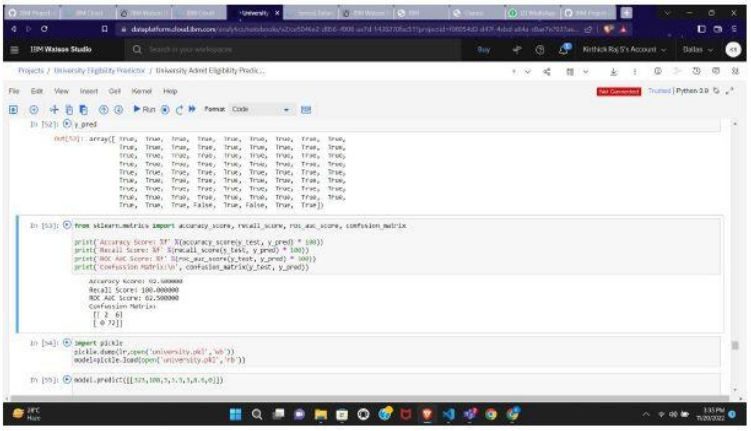
TestCase Analysis:

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

Section	TotalCases	Not Tested	Fail	Pass
PrintEngine	7	0	0	7
ClientApplication	51	0	0	51
Security	2	0	0	2
OutsourceShipping	3	0	0	3
ExceptionReporting	9	0	0	9
FinalReportOutput	4	0	0	4
VersionControl	2	0	0	2

9. RESULTS

9.1 Performance Metrics

S.No	Parameters	Value	Screenshot
1.	Metrics	**Regression Model:** Mean Absolute Error: 0.05015571939256262 Mean Squared Error: 0.00579019845213460 Root Mean Squared Error: 0.07609335353455389	
2.	Metrics	**Classification Model:** Accuracy Score: 92.500000 Recall Score: 100.000000 ROC AUC Score: 62.500000 Confusion Matrix: [[2 6] [0 72]]	

10. ADVANTAGES & DISADVANTAGES

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.
- Avoids data redundancy and inconsistency.
- It is fast, efficient and reliable.

Disadvantages:

- Machine errors are unavoidable when occurred. (Hardware failure, network failure, others).
- The predictions made are not 100% accurate but accurate to an acceptable value.

11. APPLICATIONS

- Reach to geographically scattered student.
- Reducing time in activities
- Paperless admission with reduced man power
- Operational efficiency

12. CONCLUSION

The project uses a Random forest regressor to predict the output and a web application is built to make the UI more accessible and easy using various technologies such as python, HTML5, CSS, Flask, Scikit, Matplot, Numpy, Pandas, Seaborn and other libraries. After the deployment of the web application, it can be accessed from anywhere with internet connection. This project reduces the long hours of analysis to predict the eligibility of the admission to a rated university.

13. FUTURE SCOPE

The future scope of this project is very broad. Few of them are:

- This can be implemented in less time for proper admission process.
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user had not need to travel a long distance for the admission and his/her time is also saved as a result of this automated system.

14. APPENDIX

SOURCE CODE:

main.html

```
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
<title>University Admission Eligibility Predictor</title>
<style>
   body{
      background-image:  linear-gradient(to bottom right, white, #333);
      margin: auto;
      width: fit-content;
   }
   h1{
      padding: 30px 0;
   }
   .gg {
      margin:auto;
      width: fit-content;
   }
</style>
</head>
<body>
<h1 style="text-align:center">UNIVERSITY ADMISSION ELIGIBILITY PREDICTOR</h1>
<div class="gg">
   <h3>Enter your details and get probability of your admission</h3>
```

```
    <form action="/predict" method="post">
        <p><b>Enter GRE Score : <input type="text" name="GRE Score"></input></p>
        <p>Enter TOEFL Score : <input type="text" name="TOEFL Score"></input></p>
        <p>Select University Rating :<br>
            <input type="radio" id="u1" name="University Rating" value="1"><label
for="u1">1</label></input><br>
            <input type="radio" id="u2" name="University Rating" value="2"><label
for="u2">2</label></input><br>
            <input type="radio" id="u3" name="University Rating" value="3"><label
for="u3">3</label></input><br>
            <input type="radio" id="u4" name="University Rating" value="4"><label
for="u4">4</label></input><br>
            <input type="radio" id="u5" name="University Rating" value="5"><label
for="u5">5</label></input><br>
        </p>
        <p>Enter SOP : <input type="text" name="SOP"></input></p>
        <p>Enter LOR : <input type="text" name="LOR"></input></p>
        <p>Enter CGPA : <input type="text" name="CGPA"></input></p>
        <p>Research<br>
        <input type="radio" id="r1" name="Research" value="1"><label
for="r1">Research</label></input><br>
        <input type="radio" id="r0" name="Research" value="0"><label for="r0">No
Research</label></input><br>
        </p></b>
        <button type="submit">Predict</button>
    </form>
</div>
</body>
</html>
```

chance.html

```
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
<title>Chance</title>
<style>
    body{
        background-image: linear-gradient(to bottom right, white, rgb(237, 161, 21));
    }

.sty{
margin-top:100px;
```

```
font-family:Verdana, Geneva, Tahoma, sans-serif;
margin-left:450px;
margin-right:450px;
padding:5px;
text-align:center
}

</style>
</head>
<body>
<form action="/chance" method="post">
<div class="sty">
<p><b>Predicting Chance of Admission</b>
<br>
<br>
Prediction :<u><b>You have a chance <b></u></p>
<p>The model has predicted that you have <strong>{{predict}}%</strong> chance</p>
</div><br>
</form>
</body>
</head>
</html>
```

nochance.html

```
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
<title>NoChance</title>
<style>
    body{
       background-image: linear-gradient(to bottom right, white, rgb(32, 189, 87));
   }

.sty{
margin-top:100px;
font-family:Verdana, Geneva, Tahoma, sans-serif;
margin-left:450px;
margin-right:450px;
padding:5px;
text-align:center;

}
```

```
</style>
</head>
<body>
<form action="/nochance" method="post">
<div class="sty">
<p><b>Predicting Chance of Admission</b>
<br>
<br>
Prediction :<u><b>You don't have a chance </b></u></p>
<p>The model has predicted that you have <strong>{{predict}}%</strong> chance</p>
</div><br>
</form>
</body>
</head>
</html>
```

app.py

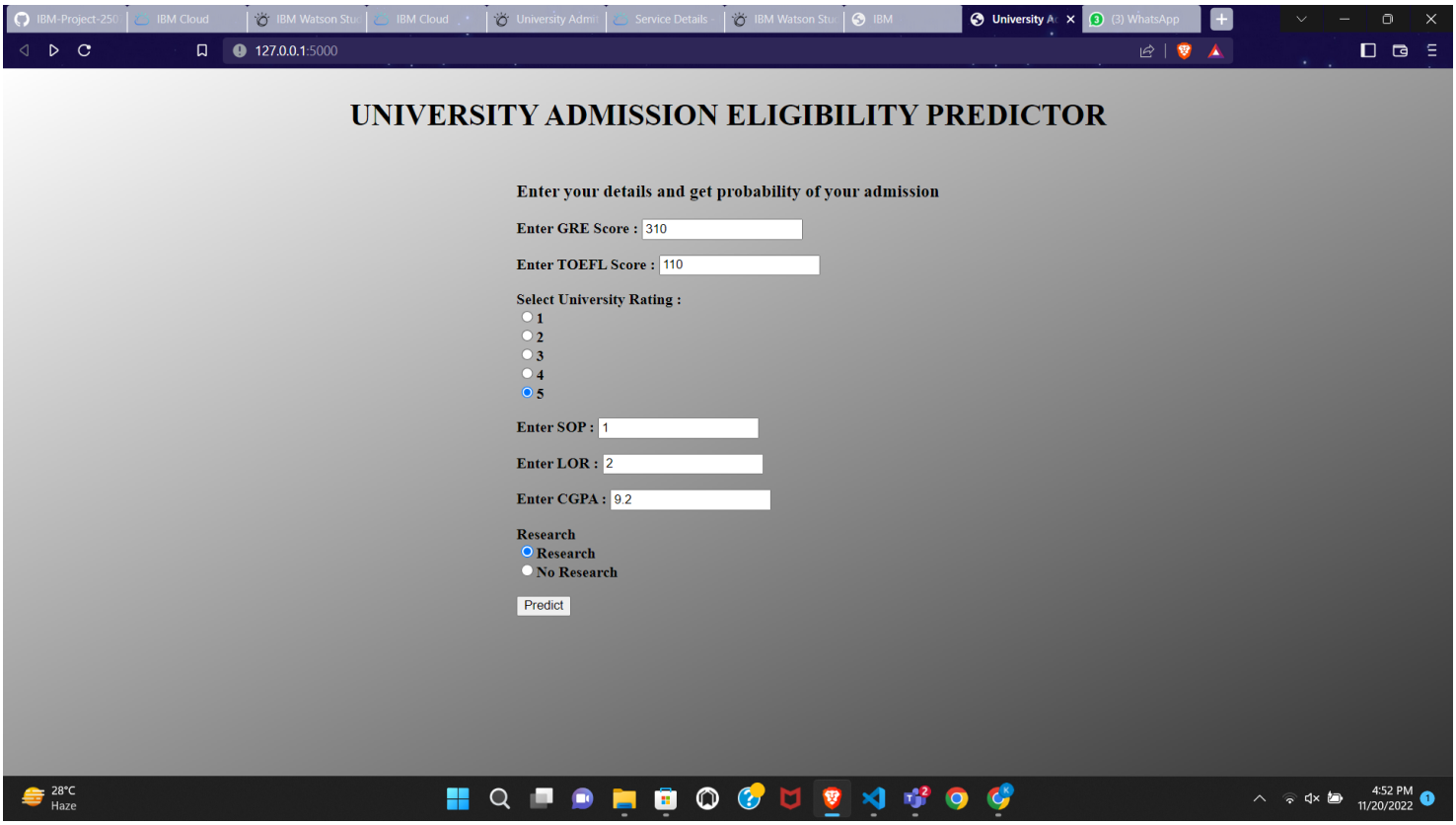
```
from flask import Flask,
    render_template,
    redirect, url_for, request
import requests

# NOTE: you must
    manually set API_KEY
    below using information
    retrieved from your IBM
    Cloud account.
API_KEY =
    "6fJrHdujsYQK3EvCJJ
    AkXCylrXC2bAA8oyE3r
    kKRmkdh"
token_response =
    requests.post('https://ia
    m.cloud.ibm.com/identit
    y/token',
    data={"apikey":
 API_KEY, "grant_type":
    'urn:ibm:params:oauth:
    grant-type:apikey'})
mltoken =
    token_response.json()["
    access_token"]
```

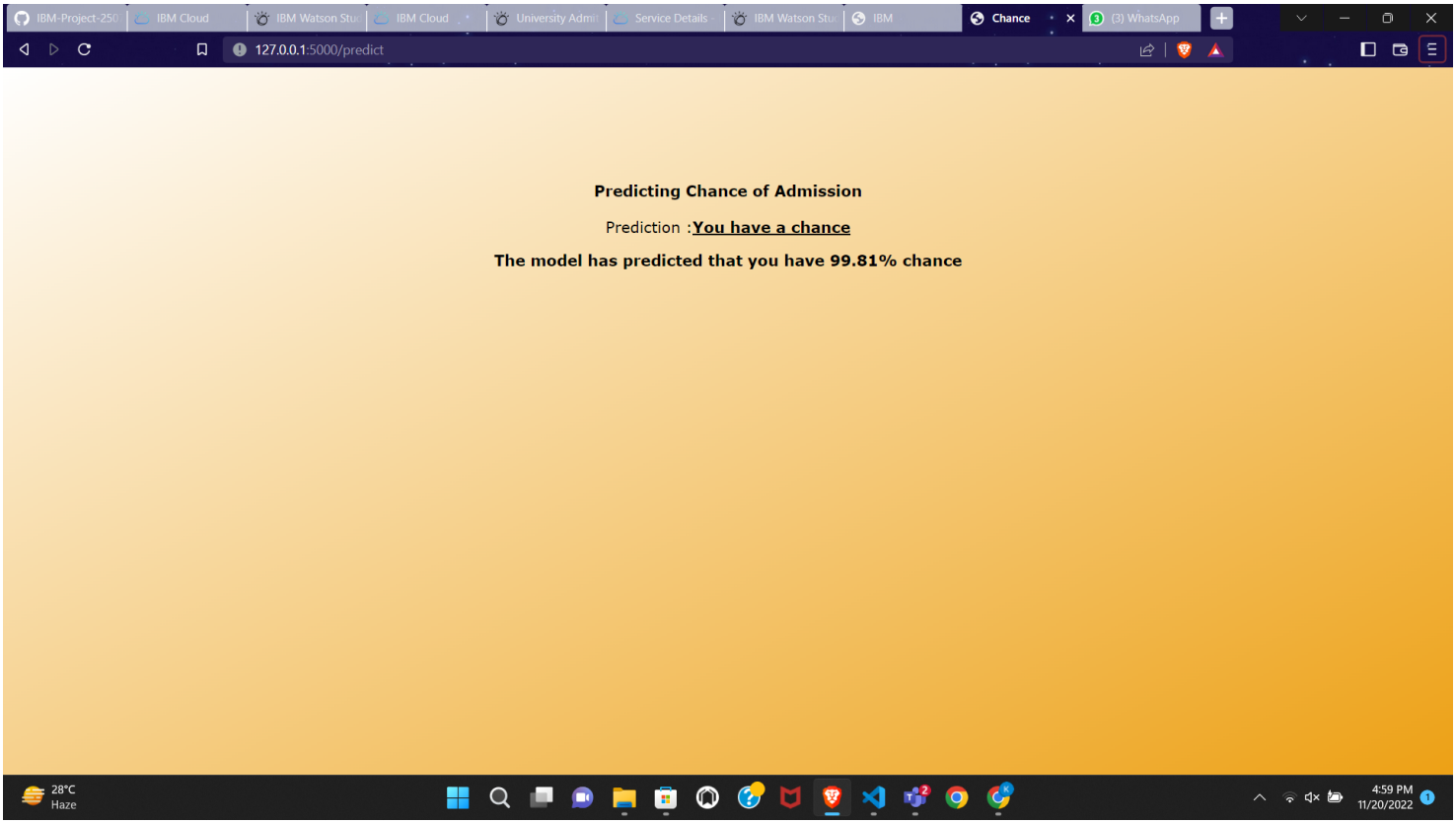
```
header = {'Content-Type':
    'application/json',
    'Authorization': 'Bearer '
    + mltoken}

app = Flask(_name_)
@app.route('/')
def home():
    return
    render_template("main.
    html")

@app.route("/predict",
    methods = ['POST'])
def predict():
    gre =
    request.form["GRE
    Score"]
    toefl=
    request.form["TOEFL
    Score"]
    u_rate=
    request.form["Universit
    y Rating"]
    sop=
    request.form["SOP"]
    lor=
    request.form["LOR"]
    cgpa=
    request.form["CGPA"]
    research=
    request.form["Research
    "]


    pre=[float(gre),float(toef
    l),float(u_rate),float(sop
    ),float(lor),float(cgpa),flo
    at(research)]

    # NOTE: manually
    define and pass the
    array(s) of values to be
    scored in the next line
    payload_scoring =
    {"input_data": [{"fields":
```

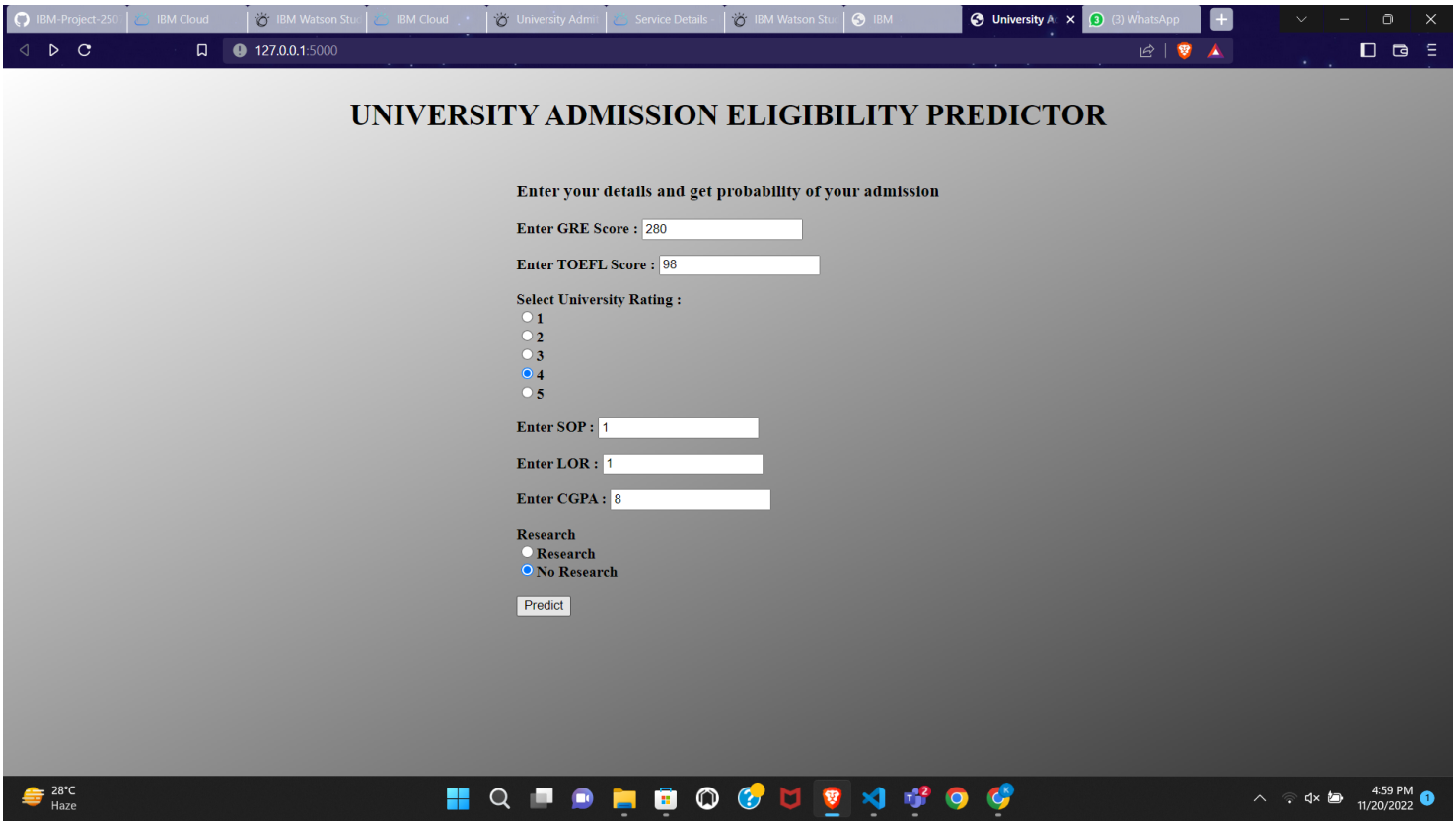
```
[ 'GRE Score',

'TOEFL Score',

'University Rating',

'SOP',

'LOR ',

'CGPA',

'Research'], "values":
[pre]}]}

response_scoring =
requests.post('https://us
-south.ml.cloud.ibm.co
m/ml/v4/deployments/f8
a4a158-7bfb-4930-869
4-305ff1860d66/predicti
ons?version=2022-11-2
0',
json=payload_scoring,

headers={'Authorization
': 'Bearer ' + mltoken})
print("Scoring
response")

pred=response_scoring
.json()
print(pred)
result =
pred['predictions'][0]['va
lues'][0][1][1]

if
pred['predictions'][0]['va
lues'][0][1][1]*100>50:
    return
render_template("chan
ce.html",
predict=round(pred['pre
dictions'][0]['values'][0][
1][1]*100,2))
```

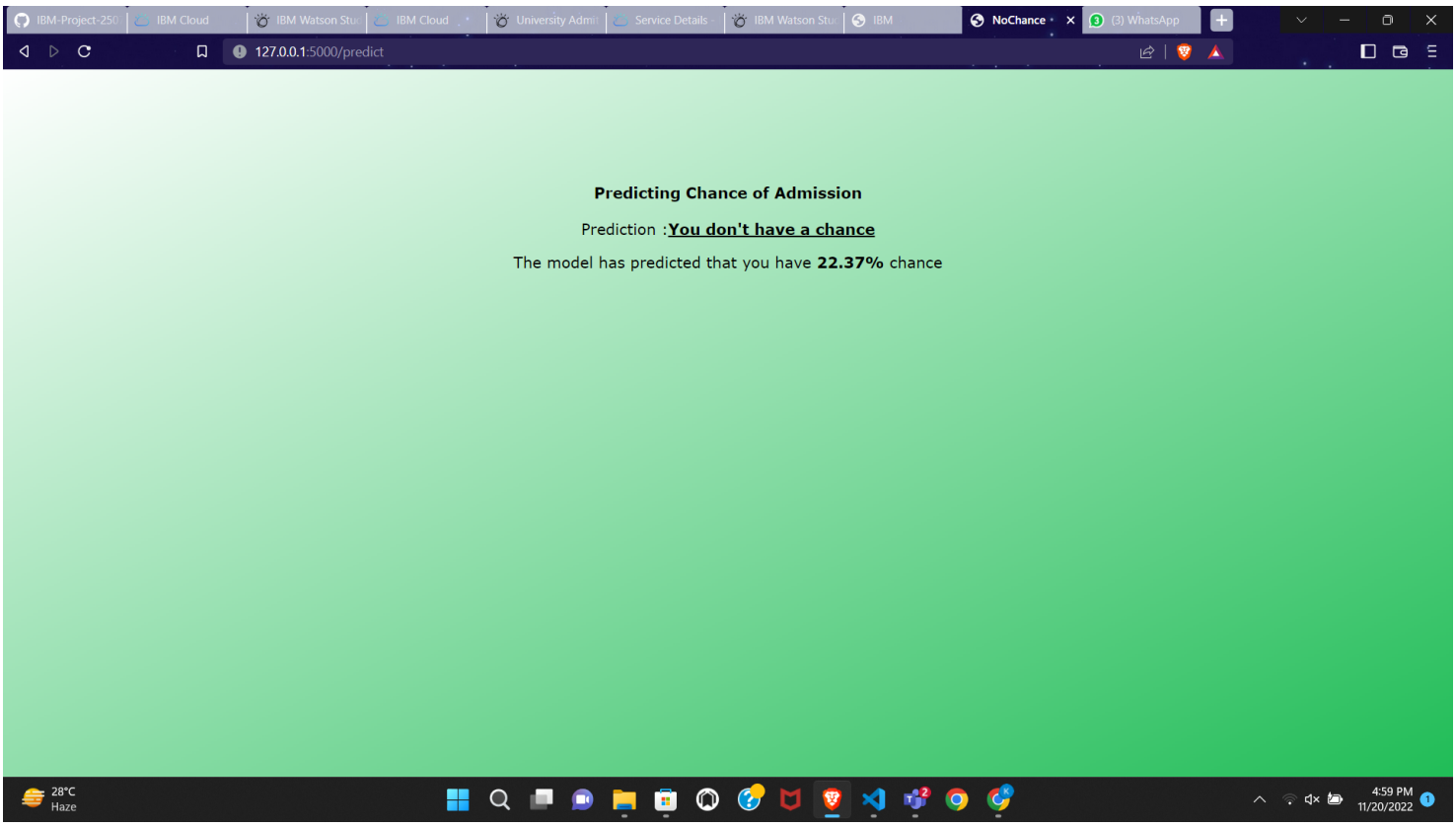
```
    else:
        return
    render_template("noch
    ance.html",
    predict=round(pred['pre
    dictions'][0]['values'][0][
    1][1]*100,2))


if _name_ == "_main_":
    app.run(debug=True)
```

Output Screenshots:






127.0.0.1:5000
UNIVERSITY ADMISSION ELIGIBILITY PREDICTOR
Enter your details and get probability of your admission
Enter GRE Score : 280
Enter TOEFL Score : 98
Select University Rating :
1
2
3
4
5
Enter SOP : 1
Enter LOR : 1
Enter CGPA : 8
Research
Research
No Research
Predict


127.0.0.1:5000/predict
Predicting Chance of Admission
Prediction :You don't have a chance
The model has predicted that you have 22.37% chance

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-22894-1659860293