

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID: PNT2022TMID32574

A PROJECT REPORT

SUBMITTED BY

**T Priya (Reg No. : 810019205078)
S Rashmi (Reg No. : 810019205083)
E Yahavarshini (Reg No. 810019205110)
S Lavanya (Reg No. : 810019205303)**

**UNIVERSITY COLLEGE OF ENGINEERING BIT CAMPUS
ANNA UNIVERSITY,TIRUCHIRAPALLI**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECEMBER 2022

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose1

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

7. CODING & SOLUTIONING

7.1 Feature 1

7.2 Feature 2

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

Students are often worried about their chances of admission to University. 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.

1.2 Purpose

A persons education plays a vital role in their life. 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.

2. LITERATURE SURVEY

PAPER 1: Graduate Admission Prediction Using Machine Learning[Sara Aljasmi, Ali Bou Nassif, Ismail Shahin, Ashraf Elnagar.]

Student admission problem is very important in educational institutions. This paper addresses machine learning models to predict the chance of a student to be admitted to a master's program. This will assist students to know in advance if they have a chance to get accepted. The machine learning models are multiple linear regression, k-nearest neighbor, random forest, and Multilayer Perceptron. Experiments show that the Multilayer Perceptron model surpasses other models.

PAPER 2 : HRSPCA: Hybrid recommender system for predicting college admission, [Abdul Hamid M Ragab, Abdul Fatah S. Mashat, Ahmed M Khedra]

This paper presents a new college admission system using hybrid recommender based on data mining techniques and knowledge discovery rules, for tackling college admissions prediction problems. This is due to the huge numbers of students required to attend university colleges every year. The proposed HRSPCA system consists of two cascaded hybrid recommenders working together with the help of college predictor, for achieving high performance.

PAPER 3: University Admissions Predictor Using Logistic Regression,[Haseeba Fathiya and Lipsa Sadath]

This is a novel study on a predictor for university admissions that allows students to assess their chances of being admitted to an institution. Real student data is gathered in order to construct this. The information is kept in the form of a training set that may be used by the logistic regression classifier that was designed to predict admissions.

PAPER 4: A Machine Learning Approach for Graduate Admission Prediction, [Amal AlGhamdi, Amal Barsheed, Hanadi AlMshjary and Hanan AlGhamdi]

This paper evaluates three learning strategies of regression to predict the university rate given the students' profile; namely, linear regression, decision tree, and logistic regression model. This paper evaluates, these models to select the best model in terms of the highest accuracy rate and the least error. It was determined that Logistic Regression model shows the most accurate prediction and hence this model was employed to predict the future applicant's university chance of admission.

PAPER 5 : Introduction to Modelling Tabular Data: Predicting a student's chance of gaining admission using ML [Jia Qing]

This article uses the Graduate Admissions dataset (UCLA Admissions Dataset) and predicts a student's chances of getting an admit into a US university using ML algorithms. It was concluded that Multiple Linear Regression was the best model for predicting the admission chances of a student.

2.1 Existing problem

Universities take into consideration different factors like score on aptitude based examination like the General Record Examination (GRE), command over the English language is judged based on their score in English competency test like Test Of English as a Foreign Language (TOEFL) OR International English Language Testing System (IELTS), their work experience in same or other fields, the quality of the Letters Of Recommendation (LOR) and the Statement Of Purpose documents provided by the student etc. Based on the overall profile of the student decision is taken by the universities admission team to admit or reject a particular candidate

2.2 References

- 1.Geiser, Saul, and with Roger Studley. "UC and the SAT: Predictive validity and differential impact of the SAT I and SAT II at the University of California." Educational Assessment 8.1 (2002): 1-26
2. Rothstein, Jesse M. "College performance predictions and the SAT." Journal of Econometrics 121.1-2 (2004): 297-317.
3. Leonard, David K., and Jiming Jiang. "Gender bias and the college predictions of the SATs: A cry of despair." Research in Higher education 40.4 (1999): 375-407

4. Aljasmi, S., Nassif, A.B., Shahin, I. and Elnagar, A., 2020. Graduate admission prediction using machine learning. *Int. J. Comput. Commun.*, 14, pp.79-83.
5. A. H. M. Ragab, A. F. S. Mashat and A. M. Kshedra, "HRSPCA: Hybrid recommender system for predicting college admission," 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA), 2012, pp. 107-113, doi: 10.1109/ISDA.2012.6416521.
6. H. Fathiya and L. Sadath, "University Admissions Predictor Using Logistic Regression," 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2021, pp. 46-51, doi: 10.1109/ICCIKE51210.2021.9410717.
7. AlGhamdi, A., Barsheed, A., AlMshjary, H. and AlGhamdi, H., 2020, March. A machine learning approach for graduate admission prediction. In *Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing* (pp. 155-158).

2.3 Problem statement definition

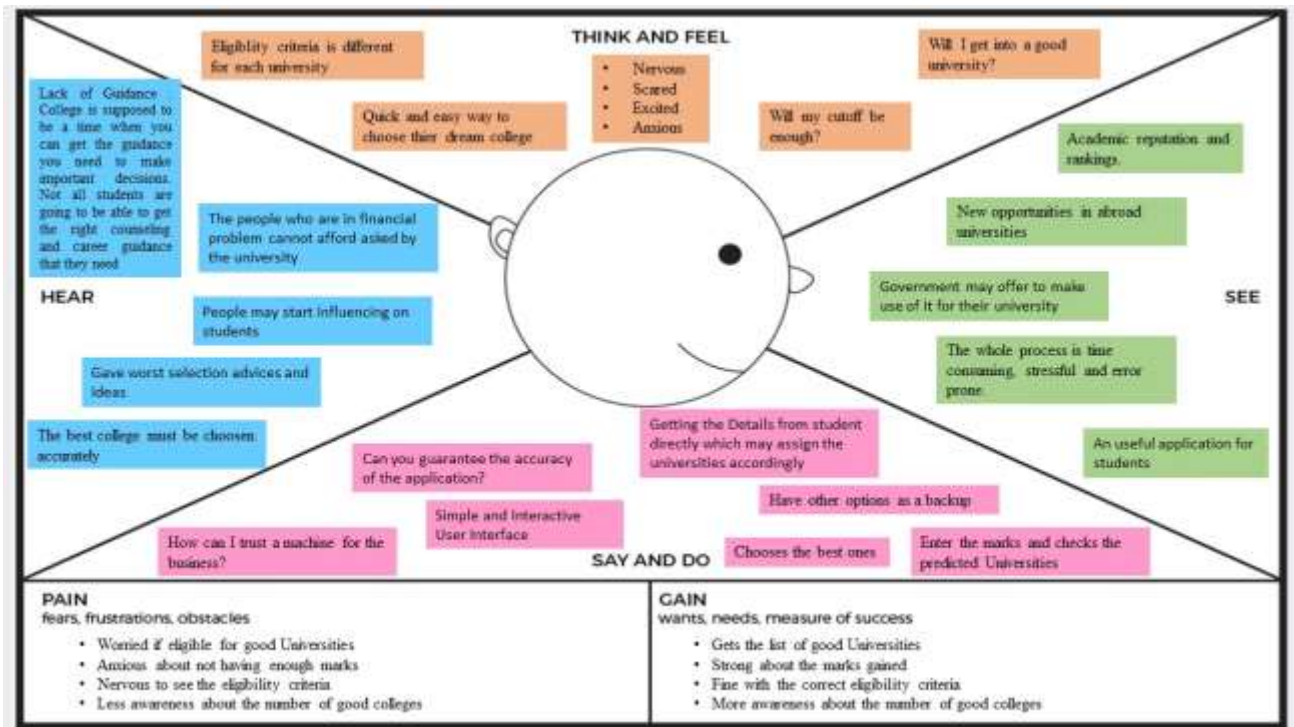
To design a college prediction/ prediction system and to provide a probabilistic insight into college administration for overall rating, cut-offs of the colleges, admission intake and preferences of students. It has always been a troublesome process for students in finding the perfect university and course for their further studies. At times they do know which stream they want to get into, but it is not easy for them to find colleges based on their academic marks and other performances. We aim to develop and provide a place which would give a probabilistic output of how likely it is to get into a university given their details.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

EMPATHY MAP FOR UNIVERSITY ADMIT ELIGIBILITY PREDICTOR:



3.2 Ideation & Brainstorming

Brainstorm & Idea Prioritization :

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. 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.



Brainstorm & idea prioritization

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



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.

🕒 10 minutes



Team gathering

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



Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.



Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#)



Brainstorm

Write down any ideas that come to mind that address your problem statement.

 10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

ESWARAMOORTHY K

Make direct connections between Students and Universities to avoid intermediaries

Connecting with Alumni or Students who are currently enrolled in the college

Provide references from trustable third party websites for a University

Admission criteria for Person with Disorders (PwD)

KRISHNAN S

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 applicant's preference list based on the entire fees for a particular stream/course. This will be help the applicants to save a lot of money in the admission process.

Collect and store all universities admission criteria to access from one place.

ROHITH S

Government should provide a portal with all university eligibility requirements listed and organised 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 could not be a favourable environment for you, therefore researching the institution would be good.

R VISHNU VASAN

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

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

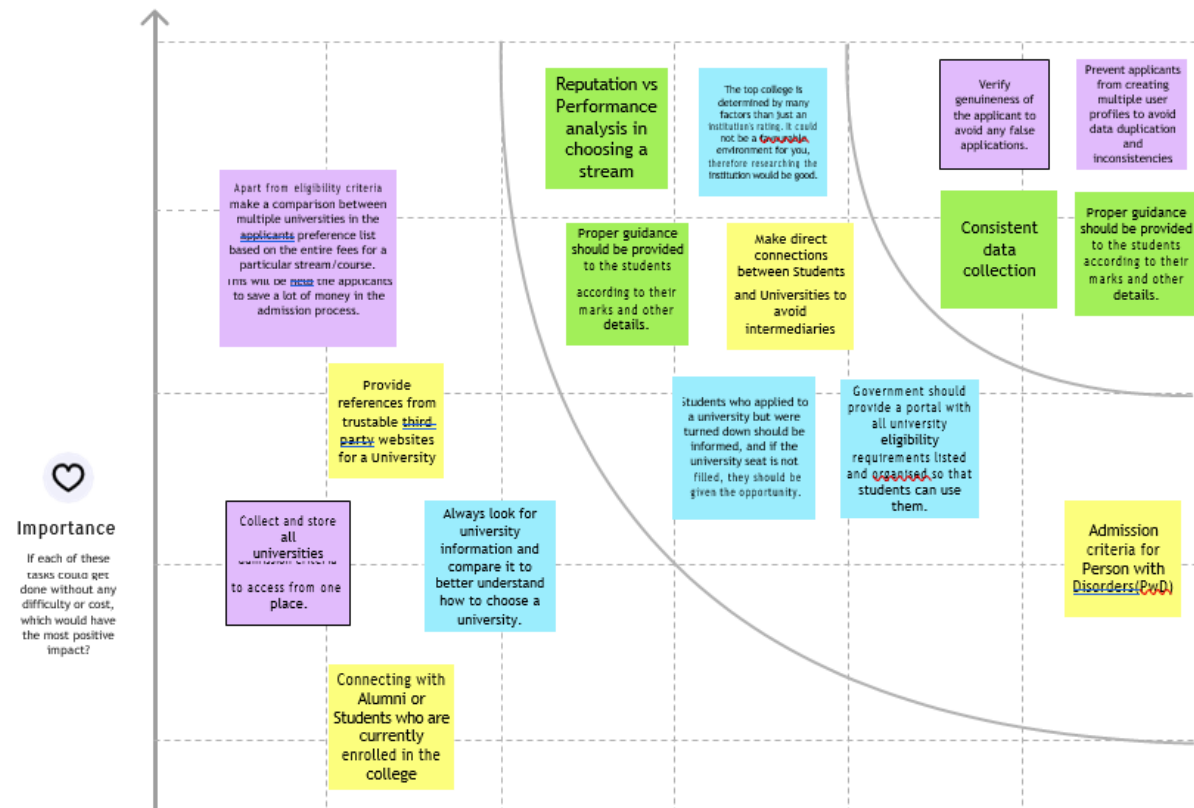


4

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



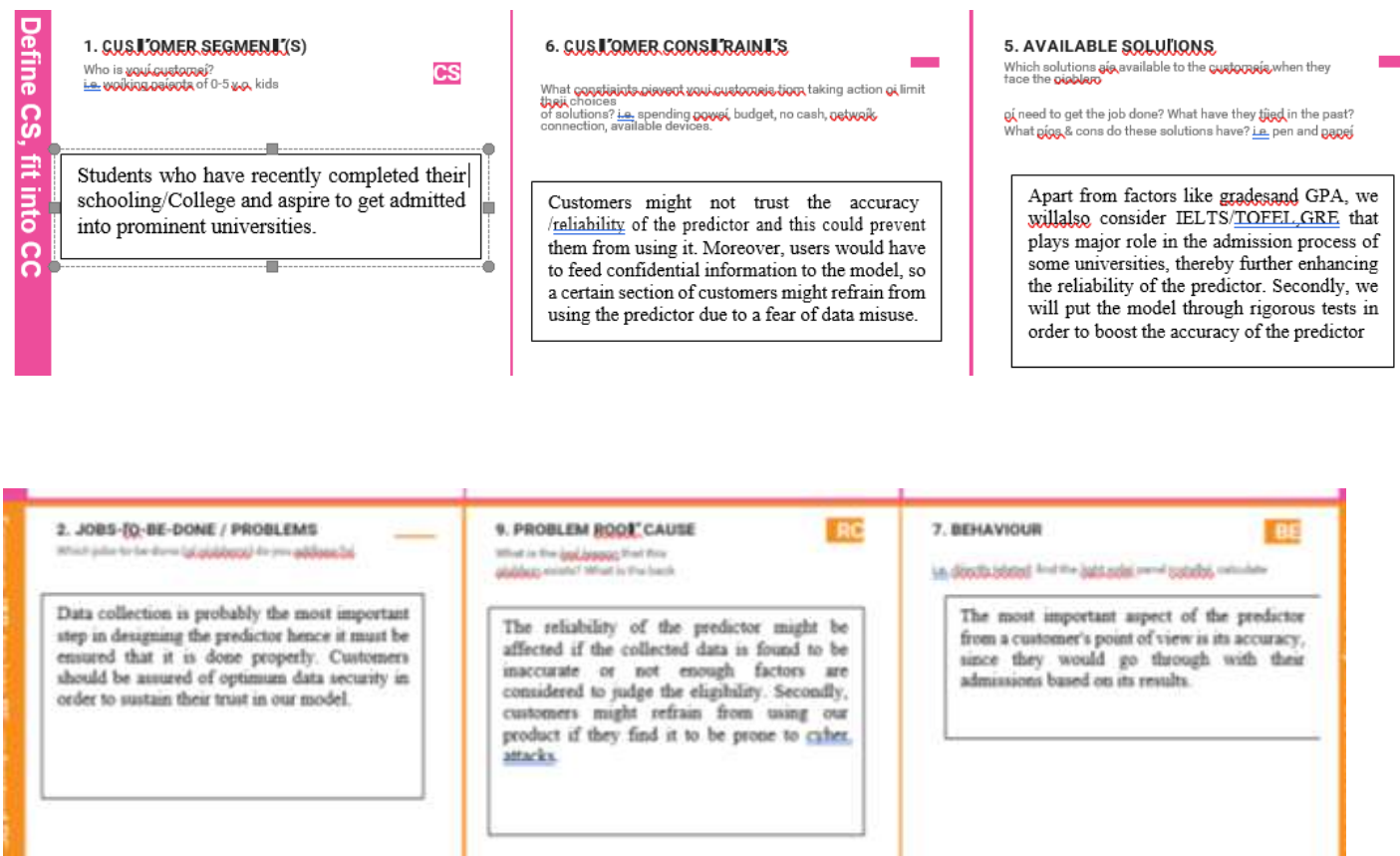
3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement(Problem to be solved)	Students are often worried about their chances of admission to university. 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. It also aims to make a direct connection between the students and the universities and avoid any intermediaries.

2.	Idea / Solution description	<p>This project intends to calculate the probability of acceptance in a particular grad-school after assessing the candidate's profile. The key attributes that will be considered for making the decisions are:</p> <ol style="list-style-type: none"> GRE & TOEFL Scores Undergrad CGPA SOP & LOR Corporate Work Experience / Research Experience Extracurriculars. <p>For determining the % of acceptance, we will be using various ML models such as Logistic Regression, Multiple Linear Regression, Decision Tree & Random Forest and assess which model gives the highest accuracy with the help of performance metrics like accuracy score, precision and recall</p>
3.	Novelty / Uniqueness	<p>We intend to develop a novel deep learning based hybrid model that has a better accuracy than the existing traditional ML models.</p> <p>The web-app will also provide feedback on the parameters where the candidate is lacking so that he can improve on those areas.</p>
4.	Social Impact / Customer Satisfaction	<p>Students often feel difficult in shortlisting the universities to apply which they tend to wonder if their profile matches the requirement of a certain university. Moreover, the cost of applying to a university is extremely high making it critical that students shortlist universities based on their profile. A university admission prediction system is quite useful for students to determine their chances of acceptance to a specific university. This system reduces dependence on educational consultancies, who charge loads of money to analyse a candidate's profile and determine the universities he/she should apply to.</p>
5.	Business Model (Revenue Model)	<p>Advertisements of different universities could be placed in the web-app to generate revenue through ads.</p> <p>In the future, a separate premium plan could be created where the students can directly interact with the professors and alumni of the university through video calls</p>

6.	Scalability of the Solution	<p>A future update could have chat space where candidates, faculties, current students of the university and alumni can interact and candidates can get their doubts resolved instantly.</p> <p>To deal with huge volumes of data in the future (Both - applicants and university details), cloud-based storages (IBM cloud, AWS, GCP, AZURE) and NoSQL databases (MongoDB, Redis, etc.) could be used instead of the traditional RDBMS storage.</p> <p>Alternatively, distributed big-data processing techniques could be explored if the no. of users using the website increase exponentially during the course of time</p>
----	-----------------------------	--

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	In order to prevent unauthorised access to the system, users must be able to log into their accounts using the system by providing their email and password.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Data Management	This application enables the user to CRUD (Change, Read, Update, and Delete) data.
FR-4	Web Service Management Process	Web Service Management is the process of registering a web client to provide SSO (Single sign-on) or member data transmission.
FR-5	Data retention	The proposed application system handles historical data archiving, retrieval, and retention.
FR-6	User Deliverables	Submission of relevant documents - Required Entrance Exam Marksheet, Curriculum vitae(CV), Personal Information, Letter of Recommendation
FR-7	User Profile	Applicant's dashboards - Personal information, wishlist, skills and course, percentage

4.2 Non-Functional requirements

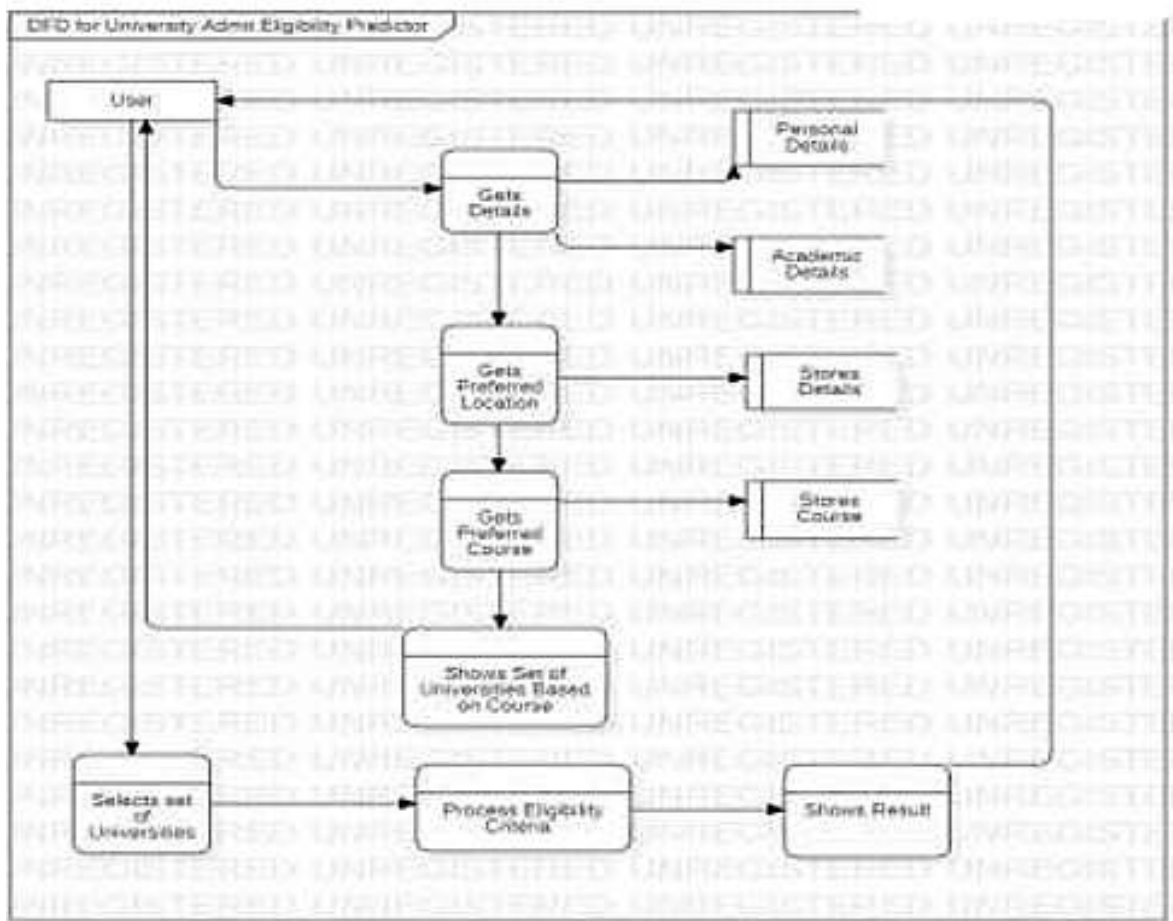
FR No	Non-Functional Requirement	Description
NFR-1	Usability	A logical interface to use and to speed up typical processes. The mistake rate of users providing their information on the checkout page must not exceed 10%ace is required to make the system easy.
NFR-2	Security	Authorization access scenarios and definitions, as well as student record handover processes between universities. Utilize certain cryptographic techniques. When the application is validating the user or licence, communication must be limited.
NFR-3	Reliability	Data corruption is avoided by employing backup methods and strategies. At the moment of input, all data

		stored for user variables will be committed to the database.
NFR-4	Performance	The availability results of the requested college should be supplied to the student in little more than two seconds, and data retrieval should be trustworthy because each student will be granted a maximum of 10 minutes, accessing the database should be done at a reasonable speed.
NFR-5	Availability	The system should be available at all times, allowing the user easy access. If the hardware or database fails, a substitute page will be displayed, and the database should be obtained from the data folder.
NFR-6	Scalability	Determines the highest workloads under which the system will still run satisfactorily. Deals with the measurement of the system's reaction time under varied load circumstances

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.



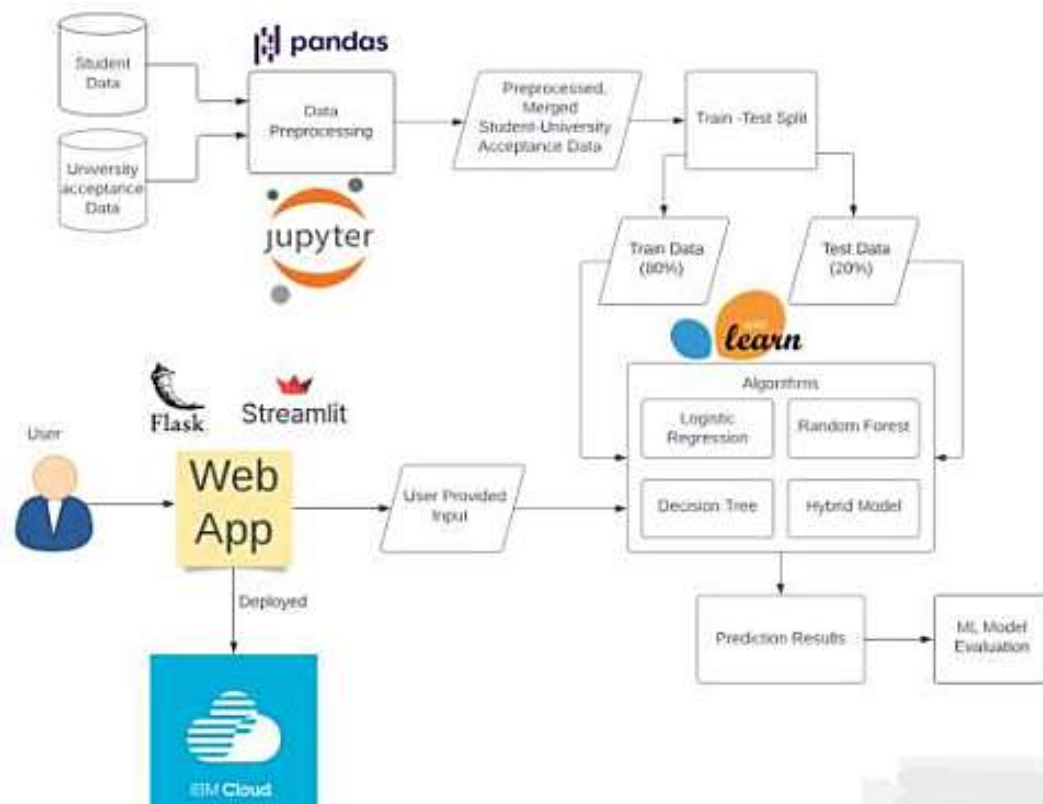
5.2 Solution & Technical Architecture

Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:



5.3 User Stories

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



6.2 Sprint Delivery 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	4

Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	4
Sprint-2		USN-3	As a user, I can check the eligibility criteria for various universities by uploading the necessary documents	2	Low	4
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	4
Sprint-4	User Login	USN-5	As a user, I can log into the application by entering email & password	1	High	4
	Dashboard		Check dashboard for further updates and upload the details according to the desired and eligible universities based on the eligibility criteria.			4

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

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Login.html:

```
<!DOCTYPE html>
<html>
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
  <link rel="stylesheet" href="{{url_for('static',filename='Lcssstyle.css')}}">

</head>
<body>
<h1 align="center" class="h1">University Admit Eligibility Predictor</h1>


<div class="login">
  <h2 class="h1">LOGIN</h2>
  <form action="{{ url_for('login')}}" method="post">

    Email: <input type="email" name="email" placeholder="Enter your mail" required="required" autocomplete />
    Password: <input type="password" name="pswd" id="pswd" placeholder="Enter your password"
required="required" maxlength=15/>
    {{error}}
    <span id="error" style="color:#F00;"> </span>
    <button type="submit" class="btn btn-primary btn-block" id="lbtn">login</button>
    <h5>not registered yet..?</h5><a href="{{url_for('reg')}}">click here for register</a>
  </form>
  <br>
  <br>

</div>
</body>
</html>
```

Register.html

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
  <link rel="stylesheet" href='{{url_for('static',filename='Lcssstyle.css')}}'>
</head>
<script>

function confirmPass() {
  var pass = document.getElementById("pswd").value
  var confPass = document.getElementById("cpswd").value
  if(pass != confPass) {
```

```

    //alert("Wrong confirm password !");
    document.getElementById('error').innerHTML='wrong confirm password';
    document.getElementById('rbtn').disabled=true;
}
else
{
    document.getElementById('error').innerHTML="";
    document.getElementById('rbtn').disabled=false;
}
}
</script>
<body><h1 align="center" class="h1">University Admit Eligibilty Predictor</H1>

<div class="register">
    <form action="{url_for('register')}" method="post">
        <h2 class="h1" align="center" >REGISTRATION</h2>
        Email: <input type="email" name="email" placeholder="Enter your mail" required="required" autocomplete />
        Password: <input type="password" name="pswd" id="pswd" placeholder="Enter your password"
required="required" maxlength=15/>
        Confirm Password:<input type="password" name="cpswd" id="cpswd" placeholder="re-enter same password"
required="required" oninput="confirmPass()" />
        <span id="error" style="color:#F00;"> </span>
        <button type="submit" class="btn btn-primary btn-block" id="rbtn">Register</button>
    </form><br>
<br>
</div>
</body>
</html>

```

Lcsstyle.css

```

@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom: 0; font-size: 13px; line-
height: 18px; color: #333333; text-align: center; text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75); vertical-align:
middle; background-color: #f5f5f5; background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6); background-image:
-ms-linear-gradient(top, #ffffff, #e6e6e6); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff),
to(#e6e6e6)); background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image: -o-linear-
gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top, #ffffff, #e6e6e6); background-repeat: repeat-x;
filter: progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6, GradientType=0);
border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border:
1px solid #e6e6e6; -webkit-border-radius: 4px; -moz-border-radius: 4px; border-radius: 4px; -webkit-box-shadow:
inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255,
0.2), 0 1px 2px rgba(0, 0, 0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn.active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px; -moz-border-radius:
5px; border-radius: 5px; }
.btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6; background-position: 0 -15px; -webkit-
transition: background-position 0.1s linear; -moz-transition: background-position 0.1s linear; -ms-transition:
background-position 0.1s linear; -o-transition: background-position 0.1s linear; transition: background-position 0.1s
linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #ffffff; }
.btn-primary.active { color: rgba(255, 255, 255, 0.75); }
.btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4); background-image: -webkit-gradient(linear, 0 0, 0
100%, from(#6eb6de), to(#4a77d4)); background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -o-linear-gradient(top, #6eb6de, #4a77d4); background-image: linear-gradient(top, #6eb6de,
#4a77d4); background-repeat: repeat-x; filter: progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0); border: 1px solid #3762bc; text-shadow: 1px 1px 1px rgba(0,0,0,0.4); box-
shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5); }

```

```

.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-primary[disabled] { filter: none; background-color: #4a77d4; }
.btn-block { width: 100%; display: block; }

* { -webkit-box-sizing: border-box; -moz-box-sizing: border-box; -ms-box-sizing: border-box; -o-box-sizing: border-box; box-sizing: border-box; }

html { width: 100%; height: 100%; }

body {
  width: 100%;
  height: 100%;
  font-family: 'sans-serif';
  background: #6d9ba3;
  color: #ffffff;
  font-size: 16px;
  text-align: center-left;
  letter-spacing: 1.2px;
}

.register {
  position: absolute;
  top: 40%;
  left: 50%;
  margin: -150px 0 0 -150px;
  width: 400px;
  height: 400px;
}

.login {
  position: absolute;
  top: 40%;
  left: 50%;
  margin: -150px 0 0 -150px;
  width: 400px;
  height: 400px;
}

.h1 { font-size: 38px; color: white; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing: 1px; text-align: center; margin-left: 0; }

body .origin { padding-bottom: 10px; }
body .dest { padding-bottom: 10px; }
body .btn { padding: 20px; }
input {
  width: 100%;
  margin-bottom: 10px;
  background: #ffff;
  border: none;
  outline: none;
  padding: 10px;
  font-size: 13px;
  color: #ffffff;
  text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
  border: 1px solid rgba(0,0,0,0.3);
  border-radius: 4px;
  box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
  -webkit-transition: box-shadow .5s ease;
  -moz-transition: box-shadow .5s ease;
  -o-transition: box-shadow .5s ease;
  -ms-transition: box-shadow .5s ease;
}

```

```

    transition: box-shadow .5s ease;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,255,0.2); }
#img{
    z-index: -1;
    width: 100%;
    height: 100%;
    opacity: 40%;
}
.h2{
    text-align:-30px;
    padding-right : 2rem;
}

```

7.2 Feature 2

mainpage.html

```

<!DOCTYPE html>
<html lang="en" dir="ltr" xmlns="http://www.w3.org/1999/html">
<head>
  <meta charset="UTF-8">
  <title> UAEP </title>
  <link rel="stylesheet" href="{{url_for('static',filename='mainstyle.css')}}">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<div class="container">
  <div class="title">Enter the details</div>
  <div class="content">
    <form action="{{ url_for('home')}}" method="post">
      <div class="user-details">
        <div class="input-box">
          <span class="details">GRE Score</span>
          <input type="text" name="gre" placeholder=" " required>
        </div>
        <div class="input-box">
          <span class="details">TOFEL Score</span>
          <input type="text" name="toefl" placeholder="" required>
        </div>
        <div class="input-box">
          <span class="details">University Ranking</span>
          <input type="text" name="urank" placeholder="" required>
        </div>
        <div class="input-box">
          <span class="details">SOP</span>
          <input type="text" name="sop" placeholder="" required>
        </div>
        <div class="input-box">
          <span class="details">LOR</span>
          <input type="text" name="lor" placeholder="" required>
        </div>
        <div class="input-box">
          <span class="details">CGPA</span>
          <input type="text" name="cgpa" placeholder="" required>
        </div>
      </div>
      <div class="gender-details">
        <input type="radio" name="research" value="1" id="dot-1">

```



```

<input type="radio" name="research" value="0" id="dot-2">
<span class="gender-title">Research</span>
<div class="category">
  <label for="dot-1">
    <span class="dot one"></span>
    <span class="gender">Yes</span>
  </label>
  <label for="dot-2">
    <span class="dot two"></span>
    <span class="gender">No</span>
  </label>
</div>
</div>
<div class="button">
  <input type="submit" value="Predict">
</div>
</form>
</div>
</div>

</body>
</html>

```

main.css:

```

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins',sans-serif;
}
body{
  height: 100vh;
  background-image: url('coast.jpg');
  background-repeat: no-repeat;
  background-size: cover;
  display: flex;
  justify-content: center;
  align-items: center;
  /*padding: 10px;
  background: linear-gradient(135deg, #71b7e6, #9b59b6);*/
}
.container{
  max-width: 700px;
  width: 100%;
  background-color: #fff;
  padding: 25px 30px;
  border-radius: 5px;
  box-shadow: 0 5px 10px rgba(0,0,0,0.15);
}
.container .title{
  font-size: 25px;
  font-weight: 500;
  position: relative;
}
.container .title::before{
  content: "";
  position: absolute;
  left: 0;

```

```
bottom: 0;
height: 3px;
width: 30px;
border-radius: 5px;
background: linear-gradient(135deg, #71b7e6, #9b59b6);
}
.content form .user-details{
display: flex;
flex-wrap: wrap;
justify-content: space-between;
margin: 20px 0 12px 0;
}
form .user-details .input-box{
margin-bottom: 15px;
width: calc(100% / 2 - 20px);
}
form .input-box span.details{
display: block;
font-weight: 500;
margin-bottom: 5px;
}
.user-details .input-box input{
height: 45px;
width: 100%;
outline: none;
font-size: 16px;
border-radius: 5px;
padding-left: 15px;
border: 1px solid #ccc;
border-bottom-width: 2px;
transition: all 0.3s ease;
}
.user-details .input-box input:focus,
.user-details .input-box input:valid{
border-color: #9b59b6;
}
form .gender-details .gender-title{
font-size: 20px;
font-weight: 500;
}
form .category{
display: flex;
width: 80%;
margin: 14px 0 ;
justify-content: space-evenly;
}
form .category label{
display: flex;
align-items: center;
cursor: pointer;
}
form .category label .dot{
height: 18px;
width: 18px;
border-radius: 50%;
margin-right: 10px;
background: #d9d9d9;
border: 5px solid transparent;
transition: all 0.3s ease;
}
```

```

#dot-1:checked ~ .category label .one,
#dot-2:checked ~ .category label .two,
#dot-3:checked ~ .category label .three{
  background: #9b59b6;
  border-color: #d9d9d9;
}
form input[type="radio"]{
  display: none;
}
form .button{
  height: 45px;
  margin: 35px 0
}
form .button input{
  height: 100%;
  width: 100%;
  border-radius: 5px;
  border: none;
  color: #fff;
  font-size: 18px;
  font-weight: 500;
  letter-spacing: 1px;
  cursor: pointer;
  transition: all 0.3s ease;
  background: linear-gradient(135deg, #71b7e6, #9b59b6);
}
form .button input:hover{
  /* transform: scale(0.99); */
  background: linear-gradient(-135deg, #71b7e6, #9b59b6);
}
@media(max-width: 584px){
  .container{
    max-width: 100%;
  }
}
form .user-details .input-box{
  margin-bottom: 15px;
  width: 100%;
}
form .category{
  width: 100%;
}
.content form .user-details{
  max-height: 300px;
  overflow-y: scroll;
}
.user-details::-webkit-scrollbar{
  width: 5px;
}
}
@media(max-width: 459px){
  .container .content .category{
    flex-direction: column;
  }
}

```

cssstyle.css:

```

@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom: 0; font-size: 13px; line-height: 18px; color: #333333; text-align: center; text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75); vertical-align:

```

```

middle; background-color: #f5f5f5; background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6); background-image:
-ms-linear-gradient(top, #ffffff, #e6e6e6); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff),
to(#e6e6e6)); background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image: -o-linear-
gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top, #ffffff, #e6e6e6); background-repeat: repeat-x;
filter: progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6, GradientType=0);
border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border:
1px solid #e6e6e6; -webkit-border-radius: 4px; -moz-border-radius: 4px; border-radius: 4px; -webkit-box-shadow:
inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05); -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255,
0.2), 0 1px 2px rgba(0, 0, 0, 0.05); box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
cursor: pointer; *margin-left: .3em; }
.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }
.btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radius: 5px; -moz-border-radius:
5px; border-radius: 5px; }
.btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6; background-position: 0 -15px; -webkit-
transition: background-position 0.1s linear; -moz-transition: background-position 0.1s linear; -ms-transition:
background-position 0.1s linear; -o-transition: background-position 0.1s linear; transition: background-position 0.1s
linear; }
.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #ffffff; }
.btn-primary.active { color: rgba(255, 255, 255, 0.75); }
.btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4); background-image: -webkit-gradient(linear, 0 0, 0
100%, from(#6eb6de), to(#4a77d4)); background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -o-linear-gradient(top, #6eb6de, #4a77d4); background-image: linear-gradient(top, #6eb6de,
#4a77d4); background-repeat: repeat-x; filter: progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0); border: 1px solid #3762bc; text-shadow: 1px 1px 1px rgba(0,0,0,0.4); box-
shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5); }
.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled, .btn-primary[disabled] { filter:
none; background-color: #4a77d4; }
.btn-block { width: 100%; display: block; }

* { -webkit-box-sizing: border-box; -moz-box-sizing: border-box; -ms-box-sizing: border-box; -o-box-sizing: border-box;
box-sizing: border-box; }

html { width: 100%; height: 150%; }

body {
width: 100%;
height: 100%;
font-family: 'sans-serif', sans-serif;
background: #6d9ba3;
color: #fff;
font-size: 16px;
text-align: center-left;
letter-spacing: 1.2px;
}

.login {
position: absolute;
top: 40%;
left: 50%;
margin: -150px 0 0 -150px;
width: 400px;
height: 400px;
}

h1 { font-size: 38px; color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing: 1px; text-align: center; margin-left:
90px; }
body .origin { padding-bottom: 10px; }

```

```
body .dest{padding-bottom:10px}
body .btn{padding:20px}
input {
  width: 100%;
  margin-bottom: 10px;
  background: rgba(0,0,0,0.3);
  border: none;
  outline: none;
  padding: 10px;
  font-size: 13px;
  color: #fff;
  text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
  border: 1px solid rgba(0,0,0,0.3);
  border-radius: 4px;
  box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
  -webkit-transition: box-shadow .5s ease;
  -moz-transition: box-shadow .5s ease;
  -o-transition: box-shadow .5s ease;
  -ms-transition: box-shadow .5s ease;
  transition: box-shadow .5s ease;
}
#img{
  z-index: -1;
  width: 100%;
  height: 100%;
  opacity: 40%;
}
#drop{
  background: #6d9ba3;
}
input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,255,0.2);
```

8. TESTING

8.1 Test Cases

	Serial No.	GRE Score	TOEFL Score	University Rating	SDP	SDR	CDRA	Research	Chance of Admit
1	1	327	116	4	4.5	4.5	9.55	1	0.82
2	2	324	107	4	4	4.5	9.07	1	0.79
3	3	316	104	3	3	3.5	8	1	0.72
4	4	329	110	3	3.5	3.5	9.03	1	0.8
5	5	314	103	2	2	3	8.21	0	0.65
6	6	310	115	5	4.5	3	9.34	1	0.8
7	7	321	100	3	3	4	8.2	1	0.75
8	8	308	101	2	3	4	7.9	0	0.68
9	9	302	102	1	2	3.5	8	0	0.5
10	10	323	106	3	3.5	3	8.6	0	0.45
11	11	325	106	3	3.5	4	8.4	0	0.52
12	12	327	111	4	4	4.5	9	1	0.84
13	13	328	112	4	4	4.5	9.1	1	0.79
14	14	307	109	3	4	3	8	1	0.62
15	15	311	104	3	3.5	2	8.2	1	0.87
16	16	314	103	3	3.5	2.5	8.3	0	0.54
17	17	317	107	3	4	3	8.7	0	0.66
18	18	319	106	3	4	3	8	1	0.63
19	19	318	110	3	4	3	8.8	0	0.63
20	20	305	102	3	3.5	3	8.5	0	0.62
21	21	312	107	3	3	2	7.9	1	0.64
22	22	323	114	4	3	2	8.4	0	0.7
23	23	326	116	5	5	5	9.5	1	0.94
24	24	334	119	5	5	4.5	9.7	1	0.95
25	25	336	119	5	4	3.5	9.0	1	0.97
26	26	340	120	5	4.5	4.5	9.6	1	0.99
27	27	322	109	5	4.5	3.5	9.0	0	0.76

8.2 User Acceptance Testing

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

The User Acceptance of this product is not surveyed enough to give a solid conclusion. The theoretical and hypothetical acceptance is calculated to be high enough to conclude that this product is usable and valuable.

9. RESULTS

Login:



Register



Mainpage:

Enter the details

GRE Score

TOFEL Score

University Ranking

SOP

LOR

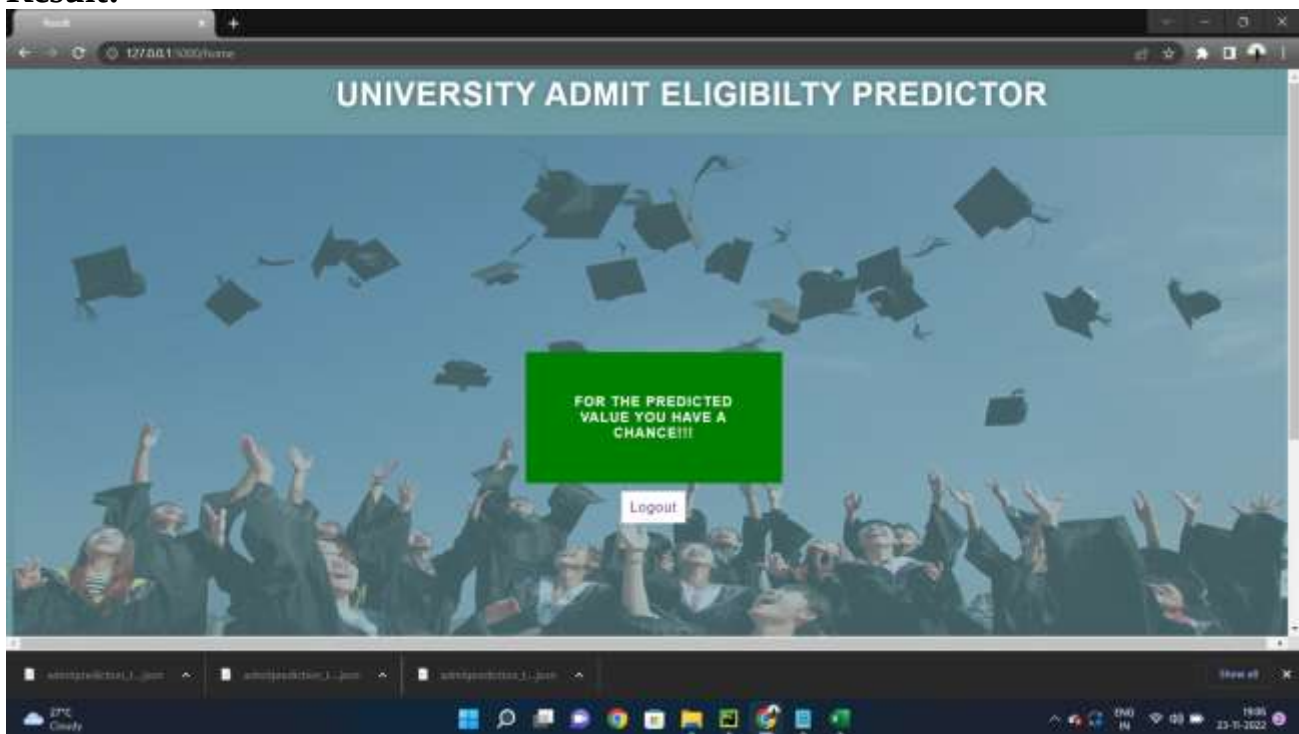
COPA

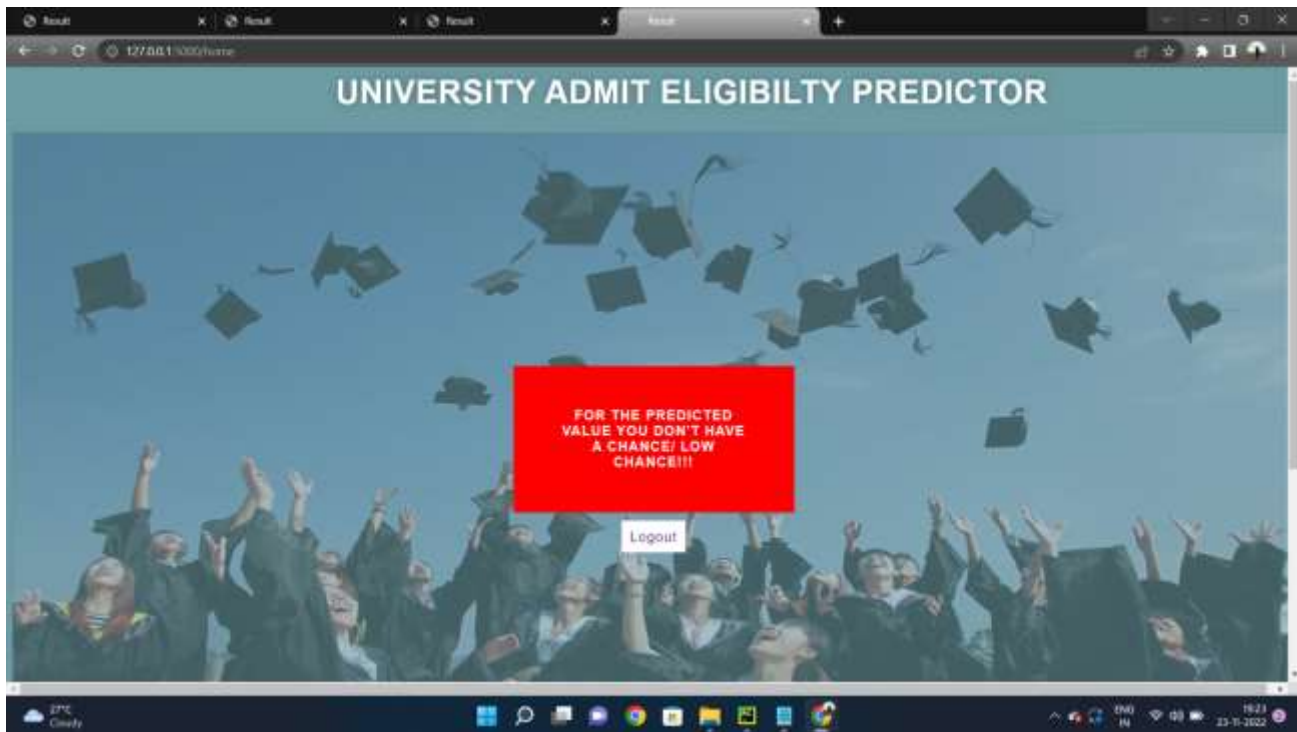
Research

☐ Yes ☐ No

[Predict](#)

Result:

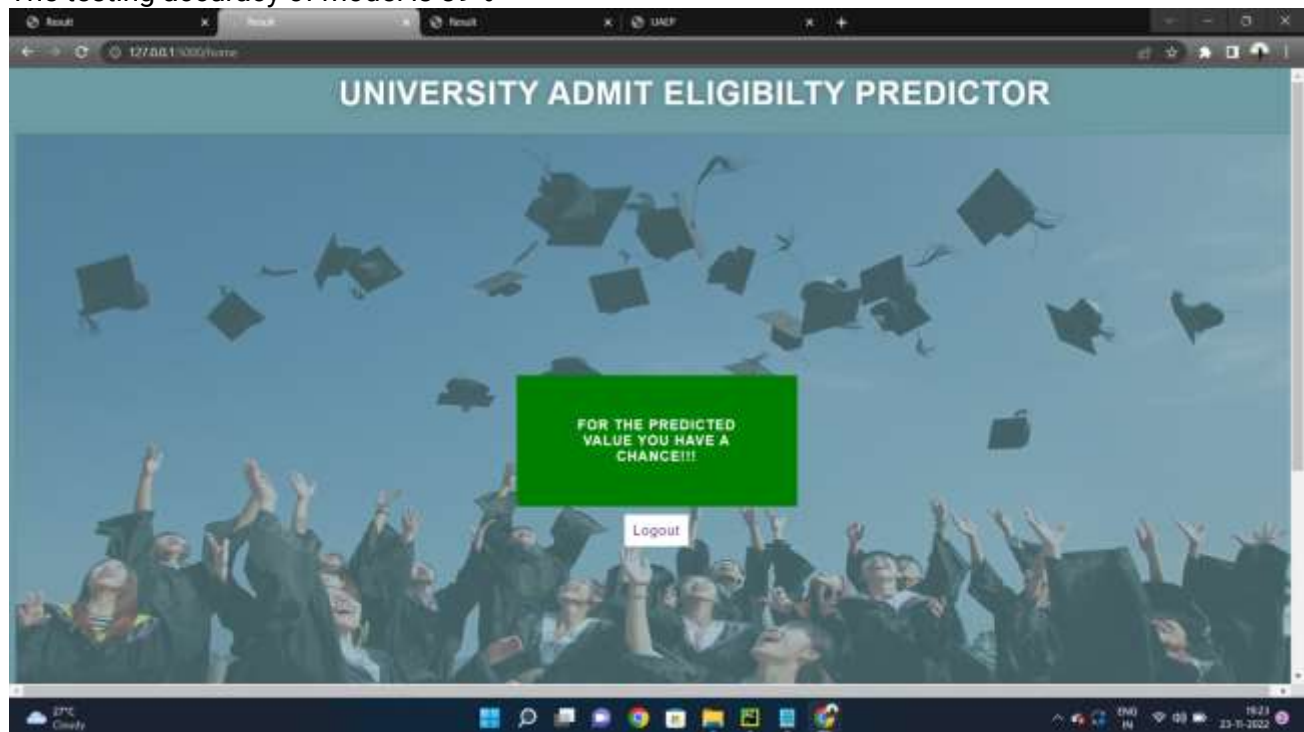




9.1 Performance Metrics

The Performance is the Accuracy of the model trained. The training accuracy of the model is 92%.

The testing accuracy of model is 89%.



10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. Reduces time consumption
2. Easy to predict
3. Good User Experience
4. Very User-friendly

DISADVANTAGES:

1. Prediction may not be more accurate
2. Privacy concerns

11. CONCLUSION

Thus, we have successfully developed an application using python flask. By using the application we can predict whether we can get admission in the desired University or not.

12. FUTURE SCOPE

- a. This can be accessed anytime anywhere, since it is a web application provided only on internet connection.
- b. The User had not need to travel a long distance for the admission and his/her time is also saved a result of this system.

13. APPENDIX

Source Code

app.py

```
import numpy as np
import os
from flask import Flask, request, jsonify, render_template, json, redirect, url_for, flash
import pickle
import requests
import sqlite3

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "yPIIj1lqnADJJNHseD12Oo_ytoKLPJupvD9O4bDjKwQf"
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__)
app.secret_key="21433253"
conn=sqlite3.connect("database1.db")
conn.execute("CREATE TABLE IF NOT EXISTS login(email TEXT PRIMARY KEY,password TEXT)")
conn.close()

@app.route('/')
def main():
    return render_template("login.html")

@app.route('/login',methods=['POST','GET'])
```

```

def login():
    if request.method=='POST':
        try:
            print("request1")
            fv=[x for x in request.form.values()]
            print(fv)
            print([x for x in request.form.values()])
            print(request.form["email"])
            email=request.form["email"]
            pswd=request.form["pswd"]
            print("request2")
            conn=sqlite3.connect("database1.db")
            cur=conn.cursor()
            print(email,pswd)
            cur.execute("SELECT password FROM login WHERE email=?",(str(email),))
            print("select")

            result=cur.fetchone()
            cur.execute("SELECT * FROM login")
            print(cur.fetchall())
            print("fetch")
            if result:
                print("login successfully success")
                print(result)
                if result[0]==pswd:
                    flash("login successfully",'success')
                    return redirect('/home')
                else:
                    return render_template("login.html", error="please enter correct password")

            else:
                print("register")
                flash("please Register",'danger')

                return redirect('/reg')

        except Exception as e:
            print(e)
            print('danger-----')
            return "hello error"
        else:
            return render_template("login.html")
# return render_template('login.html')
@app.route('/reg')
def reg():
    return render_template("register.html")

@app.route('/register',methods=['POST','GET'])
def register():
    if request.method=='POST':
        try:
            print("request1")
            fv=[x for x in request.form.values()]
            print(fv)
            print([x for x in request.form.values()])
            print(request.form["email"])
            email=request.form["email"]
            print(request.form["pswd"])
            pswd=request.form["pswd"]

```

```

conn=sqlite3.connect("database1.db")
print("database")
cur=conn.cursor()
print("cursor")
cur.execute("SELECT * FROM login WHERE email=?;",(str(email),))
print("fetch")
result=cur.fetchone()
if result:
    print("already")
    flash("user already exist,please login",'danger')
    return redirect('/')
else:
    print("insert")
    cur.execute("INSERT INTO login(email,password)values(?,?)",(str(email),str(pswd)))
    conn.commit()
    cur.execute("SELECT * FROM login")
    print(cur.fetchall())
    flash("Registered successfully",'success')
    return render_template('login.html')

except Exception as e:
    print(e)
    return "hello error1"

```

```
@app.route('/home', methods=['GET','POST'])
```

```
def home():
```

```

    if(request.method=='POST'):
        gre = request.form.get('gre')
        toefl = request.form.get('toefl')
        urank = request.form.get('urank')
        sop = request.form.get('sop')
        lor = request.form.get('lor')
        cgpa = request.form.get('cgpa')
        research = request.form.get('research')

```

```
t = [[gre, toefl, urank, sop, lor, cgpa, 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": t
        }
    ]
}

```

```

response_scoring = requests.post(
    'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/276935b3-7ad4-461a-a4ca-ad7220147914/predictions?version=2022-11-25',
    json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
pred = response_scoring.json()
output = pred['predictions'][0]['values'][0][0]
return render_template("result.html", y=float(output))

```

```
return render_template('mainpage.html')

if __name__ == "__main__":
    os.environ.setdefault('FLASK_ENV', 'development')
    app.run(debug=False)
```

GitHub & Project Demo Link

Project Link:

https://dataplatform.cloud.ibm.com/home2?context=cpdaas&apps=data_science_experience&nocache=true&quick_start_target=data_science_experience

https://drive.google.com/file/d/17ZJJpddbCW0H3Xb0uT9sv19nR43b1OTh/view?usp=share_link

Github Link:

<https://github.com/IBM-EPBL/IBM-Project-10397-1659177725>

Demo Link:

https://drive.google.com/file/d/10nbvPx_zjEtZmbTTsGgCWEIUs-nmOkTW/view?usp=share_link