UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

Project Report Format

Submitted by

C.Kamali 730919106038

A.Devashree 730919106017

S.praveen Kumar 730919106065

S.Udhayakumar 730919106110

Team ID: PNT2022TMID09817

1. INTRODUCTION

1.1 Project overview

Today all the work at the time of admission of the students is done manually by ink and paper, which is very slow and consuming much efforts and time. In the modern world of technology, computer is affecting our lives in more ways than we probably are aware of computerized management maintaining information of an educational institute, colleges, other the list is endless. The main principle behind the need of college admission system is easy supervision of institutes. It can handle the details of students such as fee details or marks details. This Student Database has been designed taking into account the practical needs to manage a student's data. Moreover, it provides security at product level as well as user level. Its design concentrates on 3 types of users:

1. Admin

2. Student

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

Today in college's student details are entered manually. The student details in separate records are tedious task. Referring to all these records updating is needed. There is a chance for more manual errors.

1. When the student comes in college.

2. First of all, he/she takes admission form from reception.

3. Fills it and submits it into office.

4. Filled form is first checked with documents like merit list a detail came from university and verified by an official person, if there is any mistake then it is corrected.

5. Atthetimeofsubmissionofitthefeesisdepositedbythecandidate.

6. At the time of submission of admission form admission number is assigned to the candidate by the institute.

7. Candidate gets the receipt of fees deposition

2.2 References:

☞ Borah M.D., Application of knowledge-based decision technique to predict student enrolment decision, Recent Trends in Information Systems (Re TIS), 21-23 Dec. 2011,180-184.

☞ Ragab, A.H.M., Hybrid recommender system for predicting college admission, Intelligent Systems Design and Applications (ISDA), 29 Nov. 2012, 107-113.

☞ Rensong Dong, The module of prediction of College Entrance Examination aspiration, Fuzzy Systems and Knowledge Discovery (FSKD), 31 May 2012 ,1559-1562.

☞ E. Gottlieb, "Using integer programming to guide college admissions decisions: a preliminary report", Journal of Computing Sciences in Colleges, Volume 17, Issue 2, Pages: 271 – 279, 2001.

☞ J.S. Moore, "An expert system approach to graduate school admission decisions and academic performance prediction", ScienceDirect, Volume 26, Issue 5, October 1998, Pages 659–670.

☞ L. Chang, Applying Data Mining to Predict College Admissions Yield, Chapter 4 in J. Luan and C. Zhao (Eds.), Data mining in action: Case studies, Spring 2008 - College of Education. [7] S. Nadi, M.H. Saraee, and A. Bagheri," Hybrid Recommender System for Dynamic Web Users", International Journal Multimedia

and Image Processing (IJMIP), Vol. 1, Issue 1, March 2011.

☞ J. A. Freeman, and D. M. Skapura, "Neural Networks: Algorithms. Applications. And Programming", Addison Wesley Pub (Sd), June 1991.

☞ S. Vinnik, and H. Marc, "Decision Support System for Managing Educational Capacity Utilization in Universities", Int. Conf. on Engineering and Computer Education, ICECE05, 2005.

☞ W. C. Lou, "A Hybrid Model of Tree Classifier and Neural Network for University Admission Recommender System," Master of Science Thesis, University of Macau, Faculty of Science and Technology, 2008

2.3.Problem Statement Definition :

➤ I am a student, I am trying to find an choice of options.But, we have lot of conflicts in the past.because, have lot of paper work.Which makes me feel tired.
➤ I am faculty/staff, I am trying to counselling.But,we have lot of work.Because,hard to manage crowd,hich makes me feel stress.
➤ Students need to check their Admission Eligiblities in the respectUniversities.
➤ Students can put their mark details that will calculate and provide the
  probable chances.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2. Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement

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 →

Define your problem statement

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

5 minutes

PROBLEM

How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

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

Step-2: Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization

3.3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The main objective of the model is to predict university admission accurately and efficiently in order to help students in selecting college
2.	Idea / Solution description	Using a machine learning model, university admission for college is done. The input to the algorithm is rows of feature vector like marks or gpa, cut off, category etc. Then a decision tree and random forest is used to predict the eligibility of the student for that college / university.
3.	Novelty / Uniqueness	Unlike other models here, comparison of decision tree classifier with logistic regression and random forest classifier for various figures of merit is performed for better efficiency of prediction.
4.	Social Impact / Customer Satisfaction	➢ Ease for students ➢ Cost effective ➢ Safe and efficient
5.	Business Model (Revenue Model)	The models used is where students will be able to use features of the app for free. If the application used with more students, it is planned to enhance for subscription for some features
6.	Scalability of the Solution	As the dataset size is huge, the noise associated with the data is also huge and the pre-processing to be done is also high in this case. The output depends on the input given to the model. The response of the data is purely dependent on the data which is collected from the previous records

3.4. Problem Solution fit

1. CUSTOMER SEGMENT(S) `CS`

Students who have recently completed their schooling and ready to get admitted into top universities.

6. CUSTOMER CONSTRAINTS `CC`

Customer should receive right information about the universities.
Customer's personal data should be kept confidential.
Time of prediction should be less.
Prediction should be accurate.

5. AVAILABLE SOLUTIONS `AS`

Scores of JEE mains, advanced can also be used for predicting admissions along with cut offs.

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

Need to predict the university where they can be admitted.
The information they provide should be kept confidential.
Right and accurate information about universities should be shared with them.

9. PROBLEM ROOT CAUSE `RC`

There are more competition and admission criteria change every year.

7. BEHAVIOUR `BE`

Customer spends more money to get admission in universities.
Customer also spend time in searching in the details about universities.

3. TRIGGERS `TR`

Seeing students who use this predictor and get admissions on desired university.
The accuracy of past predictions.

10. YOUR SOLUTION `SL`

To create a solution to predict university admission accurately and efficiently in order to help students in selecting college using

8. CHANNELS of BEHAVIOUR `CH`

8.1 ONLINE
Customer search about the colleges in internet and gather the eligibility criteria and other information.

4. EMOTIONS: BEFORE / AFTER `EM`
Before: Anxious, confused ,Worried
After: more confident ,satisfied, complete

student's cutoff ,category ,marks etc.

8.2 OFFLINE
Customer visit the universities to gather information and gather information from students studying there.

4. REQUIREMENT ANALYSIS

4.1.Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	➢ Registration through Form ➢ Registration through Gmail ➢ Registration through LinkedIN
FR-2	User Confirmation	➢ Confirmation via Email ➢ Confirmation via OTP
FR-3	User Details	Submit the documents ➢ GRE or/and TOEFL Score Sheet ➢ 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

Non-Functional requirements

NFR 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. ➢ 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)
NFR-2	**Security**	➢ Only the authenticated user would be able to utilize the services of the site. ➢ Database should be backed up every hour
NFR-3	**Reliability**	➢ The system would always strive for maximum reliability due to the importance of data and damages that could be cause by incomplete and incorrect data
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
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

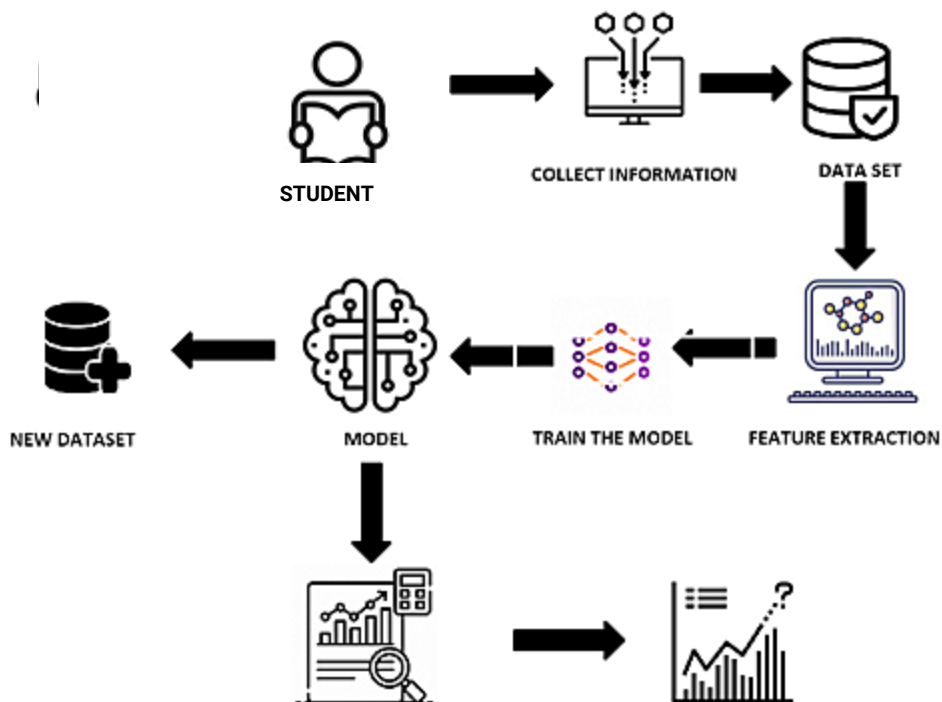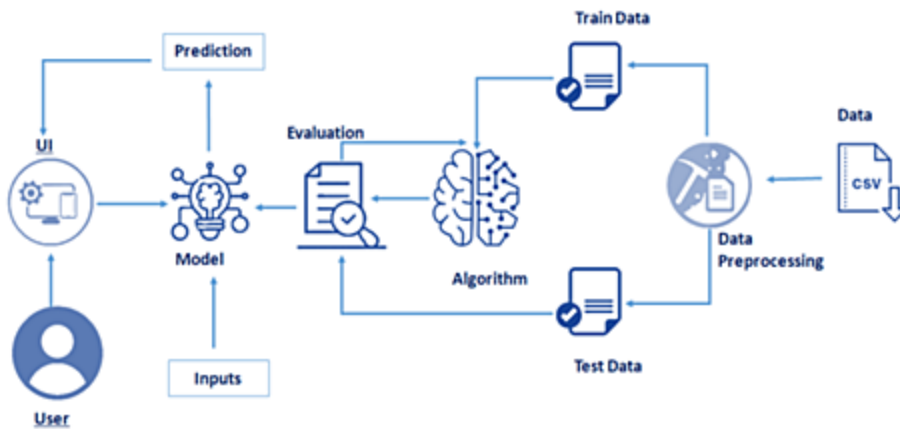Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.
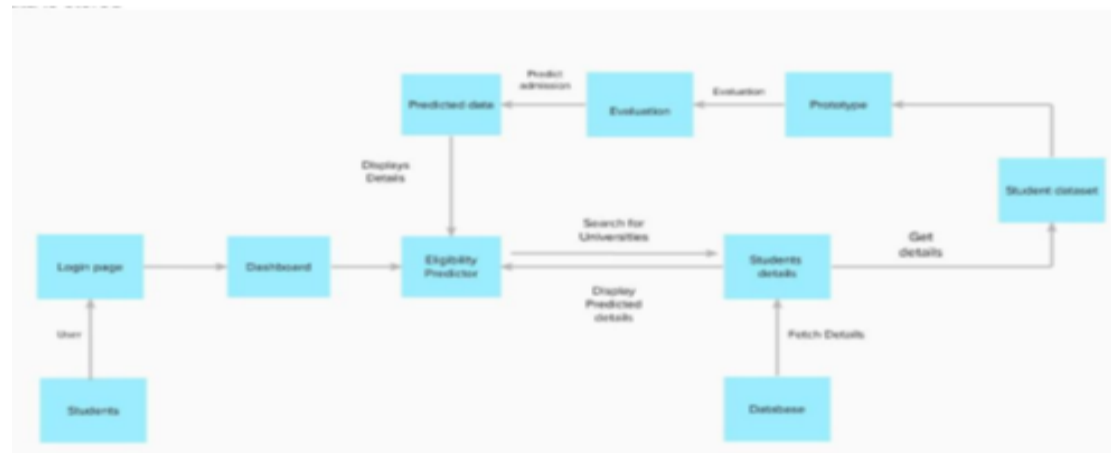
5.2.Solution & Technical Architecture

Technical Architecture

5.3.User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register and access the dashboard	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access various pages	High	Sprint-1
	Dashboard	USN-6	As a user, I can search for various universities	I can access several pages	High	Sprint-1
Customer (Web user)	Search	USN-7	As a user, I can search for universities with different field	I can receive information related to universities on various locations	High	Sprint-2
Customer Care Executive	View	USN-8	As a user, I can view the University details	I will get the information on seat availability, eligibility criteria.	High	Sprint-2
Administrator	Receive notification	USN-9	As a user, I will receive notifications about the Suggested universities based on student marks	I will get frequent updates of the preferred universities	High	Sprint-2
	Chat with expert	USN-10	As a user, I can chat with the expert for clarifications	I can clear my doubts through chat with expert option	Medium	Sprint-2
Admin	Analysis	USN-11	As an admin, I will analyse the given dataset	I can analyse the dataset	High	Sprint-2
	Predict	USN-12	As an admin, I will predict the admission	I can predict eligibility for admission	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING
6.1 Sprint Planning & Estimation

Title	Description	Date
Literature Survey and Information Gathering	Gathering Information by referring the technical papers,research publications etc	2 SEPTEMBER 2022
Prepare Empathy Map	To capture user pain and gains Prepare List of Problem Statement	10 SEPTEMBER 2022
Ideation	Prioritise a top 3 ideas based on feasibility and Importance	17 SEPTEMBER 2022
Proposed Solution	Solution include novelty, feasibility, business model, social impact and scalability of solution	24 SEPTEMBER 2022
Problem Solution Fit	Solution fit document	29 SEPTEMBER 2022
Solution Architecture	Solution Architecture	1 October 2022
Customer Journey	To understand user interactions and experiences with application.	8 October 2022
Functional Requirement	Prepare functional Requirement	14 October 2022
Data flow Diagrams	Data flow diagram	15 October 2022
Technology Architecture	Technology Architecture diagram	16 October 2022
Milestone & sprint delivery plan	Activity what we done &further plans	21 October 2022
ProjectDevelopment Delivery of sprint 1,2,3 & 4	Develop and submit the developed code by testing it	24 October 2022 – 19 November 2022

6.2.Sprint Delivery Schedule

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

6.3.Reports from JIRA

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

7.1.Feature 1

Index page:

7.2.Feature 2
Chance figure:
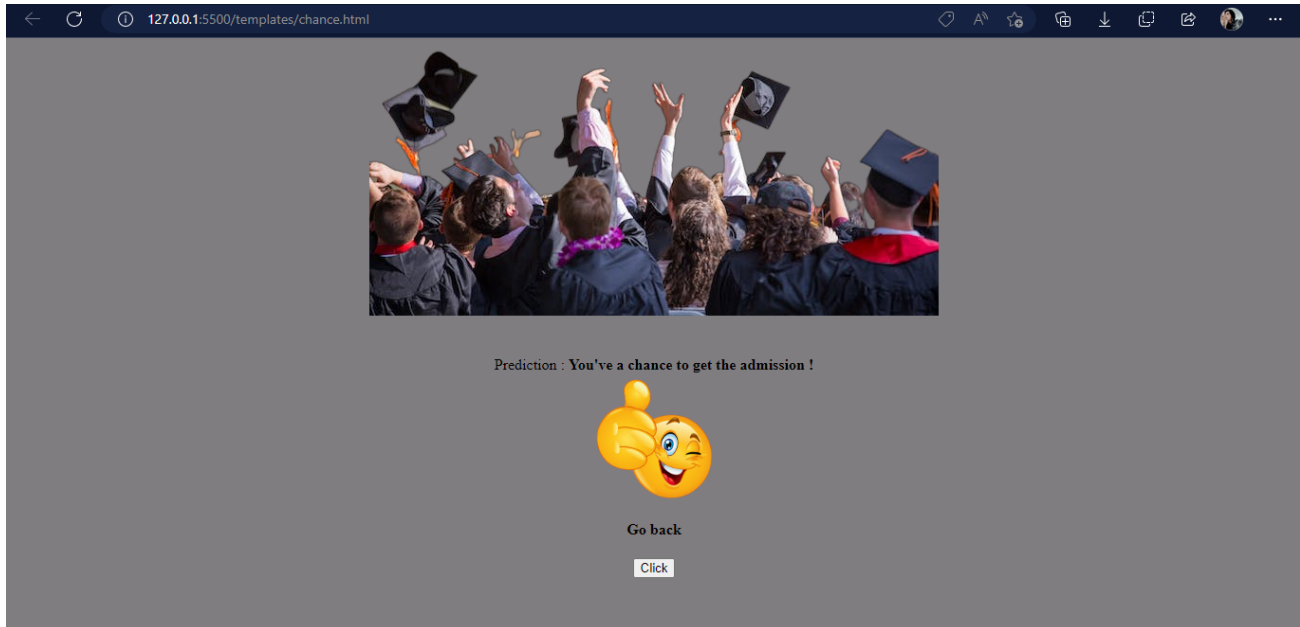
No chance Image:
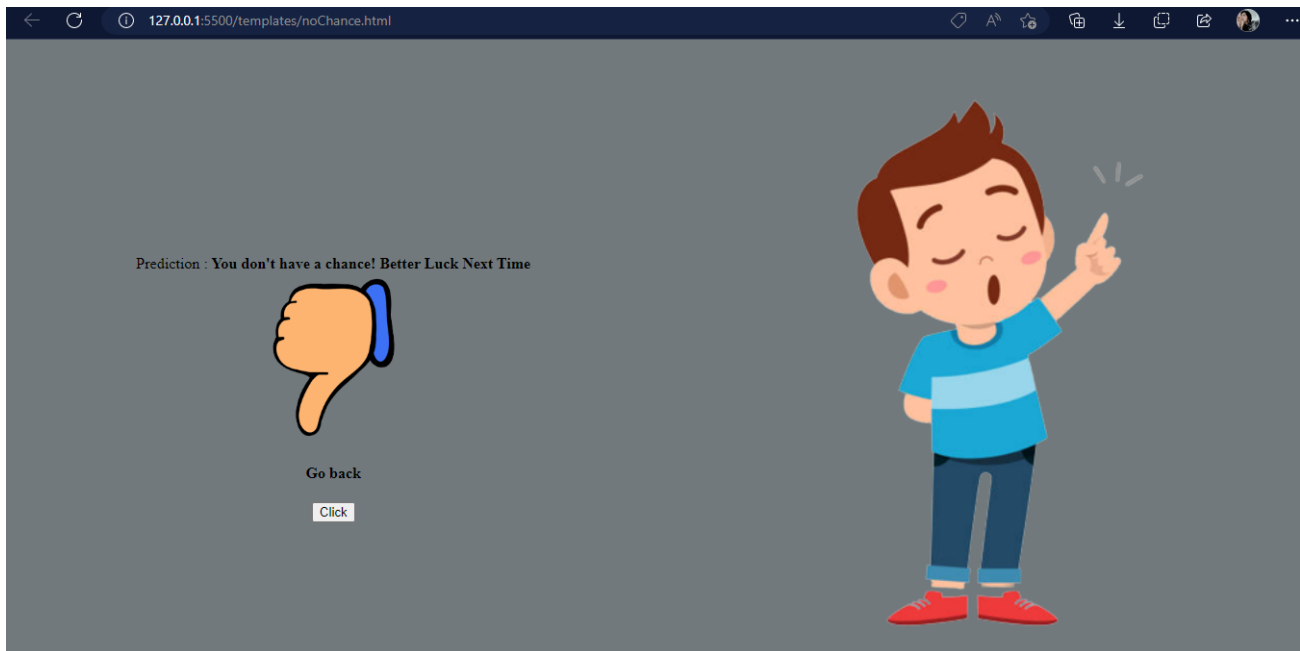
8.TESTING

8.1.Test Cases

Test case ID	Test case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/fail
LI-TC01	Checking Eligibility with valid data	1.Go to site 2.Enter the values of GRE, TOEFEL, CGPA, RATING, SOP, LRA, RESEARCH 3.Click Predict	GRE=323 TOEFEL=108 RATING=3 SOP=3.5 LOR=3 CGPA=8.6 RESEARCH=1	User is visible eligibity of the page	As Expected	Pass
LI-TC02	Checking Eligibility with valid data	1.Go to site 2.Enter the values of GRE, TOEFEL, CGPA, RATING, SOP, LRA, RESEARCH 3.Click Predict	GRE=250 TOEFEL=90 RATING=1 SOP=1 LOR=1 CGPA=7 RESEARCH=0	User is visible, non-eligibity of the page	As Expected	pass

8.2.User Acceptance Testing

Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

Test case Analysis

Section	Total cases	Not Tested	Fail	pass
print Engine	7	0	0	7
client Application	51	0	0	51
Security	2	0	0	2

9.RESULTS

9.1.Performance Metrics

 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.

➤ 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.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.

12.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.

13.APPENDIX

Source Code

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>University Admit Eligibility Predictor</title>
    </head>
    <link rel="stylesheet" href="/static/css/style.css">

  <header>
    <h1>UNIVERSITY ADMIT ELIGIBILITY PREDICTOR</h1>
    </header>
  <body>
    <h2>Enter your details and get probability of your admission</h2>
    <div class="container">
       <p><b>Enter GRE Score : <input type="number" value="score range 0-340" style="border-color:blue;"></b></p>
       <p><b>Enter TOEFL Score : <input type="number" value="0-120" style="border-color: blue;"></input></b></p>
    <form>
       <label  class="font_edits" for="university">University Rating:</label>
       <select id="University Rating" style="border-color: blue">
         <option value="University Rating">1</option>
         <option value="University Rating">2</option>
         <option value="University Rating">3</option>
         <option value="University Rating">4</option>
```

```html
        <option value="University Rating">5</option>
      </select>
      </option>
    </form>
    <p><b>Enter SOP Score : <input type="number" value="score range 0-5" style="border-color:blue;"></b></p>
    <p><b>Enter LOR Score : <input type="number" value="score range 0-5" style="border-color:blue;"></b></p>
    <p><b>Enter CGPA Score : <input type="number" value="score range 0-10" style="border-color:blue;"></b></p>
    <form>
      <label class="font_edits" for="university">Research:</label>
      <select id="Research" style="border-color: blue">
                    <option value="Research">Yes</option>

        onclick="window.location.href='http://127.0.0.1:5501/templates/index.html'" class="btn btn-warning btn-lg" value="Click"/>
  </div>
</div>

</body>
</html>
```

Chance.html

```html
<!DOCTYPE html>
<html>
<head>
      <title>eligibility</title>
</head>
<body style="background-color:rgba(102, 99, 102, 0.829);">
      <img src="/static/img/chance.png" class="center"/>
      <div style="padding-top: 2%">
              <style>
                      img {
                        display: block;
                        margin-left: auto;
                        margin-right: auto;
```

```
                    }
                </style>
                <style>
                    h1 {text-align: center;}
                    p {text-align: center;}
                    div {text-align: center;}
                    </style>
        <p>Prediction : <b>You've a chance to get the admission !</b><img
src="/static/img/thumbsup.png" height="10%" width="10%" ></p>
        <div class="container">
      <h4 style="text-align:center;">Go back</h4>
      <input type="button"
onclick="window.location.href='http://127.0.0.1:5501/templates/index.html'" class="btn btn-warning
btn-lg" value="Click"/>
    </div>
</div>


</body>
</html>
```

no chance.html

```
  <!DOCTYPE html>
<html>
<head>
        <title>eligibility</title>
</head>
<body style="background-color:rgb(113, 121, 124);"></body>
<body>
        <img src="/static/img/no chance.png" style="float: right; " height="50%" width="50%"/>
        <div style="padding-top: 15%">
                <style>
                        img {
                          display: block;
                          margin-left: auto;
                          margin-right: auto;
                        }
                </style>
```

```html
        <style>
                h1 {text-align: center;}
                p {text-align: center;}
                div {text-align: center;}
                p.small {
    line-height: 0.7;
     }
                </style>
    <p>Prediction : <b>You don't have a chance! Better Luck Next Time </b><img
src="/static/img/thumbsdown.png" height="10%" width="10%"></p>
    </div>
    <div class="container">
   <h4 style="text-align:center;">Go back</h4>
   <input type="button"
onclick="window.location.href='http://127.0.0.1:5501/templates/index.html';" class="btn btn-
warning btn-lg" value="Click"/>
  </div>
</body>
</html>
```

style.css

```css
h1 {
   text-align: center;
}
h2{
   margin-top: 30px;
   text-align: left;
 }
 .font_edits{
    font-weight: bold;
 }
 .btn_edits{
    font-size: large;
 }
 ::after{
   box-sizing: border-box;
 }
```

```css
picture{
  max-width: 100%;
  display: block;
}
body {
  background-image: url("/static/img/university\ background.jpeg");
  background-repeat: no-repeat;
  background-size: 100% 100%;
  background-attachment: fixed;
  color: white;
}

.container {
  border: 5px solid blue;
  margin-right: 800px;
  border-radius: 30px;
  background-color: grey;
  font-size: medium;
}

.footer {
  text-align: center;
  justify-content: center;
  padding-top: 10%;
  font-weight: bolder;

}
```

app.py
```python
import pickle
from flask import Flask , request, render_template
from math import ceil
app = Flask(__name__, template_folder='template')
model = pickle.load(open("model.pkl","rb"))

@app.route('/')
def index():
```

```python
        return render_template('/demo2.html')

@app.route('/predict',methods = ['GET','POST'])
def admin():
    gre=(eval(request.form["gre"])-290)/(340-290)
    tofl=(eval(request.form["tofl"])-92)/(120-92)
    rating=(eval(request.form["rating"])-1.0)/4.0
    sop=(eval(request.form["sop"])-1.0)/4.0
    lor=(eval(request.form["lor"])-1.0)/4.0
    cgpa=(eval(request.form["cgpa"])-290.0)/(340.0-290.0)
    research=request.form["research"]
    if (research=="Yes"):
        research=1
    else:
        research=0
    preds=[[gre,tofl,rating,sop,lor,cgpa,research]]
    xx=model.predict(preds)
    if (xx>0.5):
        return render_template("chance.html",p=str(ceil(xx[0]*100))+"%")
    return render_template("nochance.html")
if __name__ == '__main__':
    app.run(debug = False, port=4000)
```

GitHub Link

Project Demo Link:

[Link](#)