UNIVERSITY ADMIT ELIGIBILITY PREDICTOR A PROJECT REPORT

Submitted by

VIGNESH KUMAR K	**- 412519205155**
SNEHA JAISWAL	**- 412519205162**
HARIHARAN SV	**- 412519205043**
NIVETHA K	**-412519205097**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

INFORMATION TECHNOLOGY

SRI SAI RAM ENGINEERING COLLEGE

(An Autonomous Institution; Affiliated to Anna University, Chennai -600 025)ANNA UNIVERSITY:: CHENNAI 600 025

JULY 202

1.INTRODUCTION

1.1 PROJECT OVERVIEW

In the current world scenario, just having a bachelor's degree is not enough. Most employers now require new employees to be highly qualified. Correspondingly, the requirements for a good university education are also higher. Many students prefer to continue their higher education. Admission to these universities requires some academic requirements. However, with so many different levels of colleges and universities, students are often left with the last-minute dilemma of whether or not their application will be accepted due to the lack of specific documentation stating their requirements. Developed after examining many leading machine learning algorithms, our AI models are the most accurate predictors of a student's admissions likelihood based on their current college-level grades and other academic background of their choice. To do. The purpose of this project is for users to enter their academic data and receive predictions regarding their likelihood of admission to their desired college level. The system will also implement a database so students can store data, review and edit as they progress, and their latest predictions will be saved in their profile. Aspects of web security other than password protection within the site are not part of this project. Students - Indian students will benefit most from this scheme. Administrators - Administrators should have access to all data stored in the application. The system is available to all users from anywhere as long as they have an internet connection. The administrator can access her website from anywhere with the correct credentials and internet access.

The system should answer students' questions competently and succinctly, and provide the most accurate analysis possible of their chances of getting into college. The system is based on a limited data set, which can affect the overall accuracy of predictions. Many other factors such as personal interviews also play a large role in the admission process, so the system cannot guarantee that our predictions are 100% guaranteed admission. Other factors, such as changes in colleges and university policies, may also affect your chances of admission in ways beyond the scope of this project.

1.2 PURPOSE

This is the project for a new web-based University Admit Eligibility Predictor. Predictor is an ML based application that asks for the users to input their academic transcripts data and calculates their chances of admission into the University Tier that they selected. It also provides an analysis of the data and shows how chances of admissions can depend on various factors. This document describes the scope, objectives and goals of the system. In addition to describing the non-functional requirements, this document models the functional requirements with use cases, interaction diagrams and class models. This document is intended to direct the design and implementation of the target system in an object-oriented language.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

It’s almost admission season and I’ve couple of friends who are in panic mode waiting for a call from the universities they’ve applied at. This made me think — How can we predict whether a student will get an admit or not? What are the parameters for selection? Can it be mathematically expressed? All of these questions started popping up. This is the main existing problem.

2.2 REFERENCES

➤ https://ieeexplore.ieee.org/document/9418279

Abstract: Students regularly have difficulty finding a fitting institution to pursue higher studies based on their profile. There are some advisory administrations and online apps that recommend universities but they ask huge consultancy fees and online apps are not accurate. So, the aim of this research is to develop a model that predict the percentage of chances into the university accurately. References: MS Acharya, A Armaan and AS Antony, "A comparison of regression models for prediction of graduate admissions", 2019.

➤ https://ieeexplore.ieee.org/document/9410717

Abstract: Students applying for admissions to universities find it difficult to understand whether they have good chances of getting admission in a university or not. Keeping this in focus, we have used logistic regression techniques that have gained attention in software engineering field for its ability to be used for predictions. This is a novel work on a university admissions predictor using which students can evaluate their competitiveness for getting admission at a university.

➤ https://ieeexplore.ieee.org/document/6416521

Abstract: 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. References: G. Ganapathy, and K. Arunesh, "Models for Recommender Systems in Web Usage Mining Based on User Ratings" Proceedings of the World Congress on Engineering, Vol. I WCE 2011.

➤ https://dl.acm.org/doi/10.1145/3388818.3393716

Abstract: With the increase in the number of graduates who wish to pursue their education, it becomes more challenging to get admission to the students' dream university. Newly graduate students usually are not knowledgeable of the requirements and the procedures of the postgraduate admission and might spent a considerable amount of money to get advice from consultancy organizations to help them identify their admission chances. References: E. Roberts, "using machine learning and predictive modeling to assess admission policies and standards," 2013.

➤ https://github.com/satwik2663/Machine-Learning-Graduate-Studuent-AdmissionPredictor

Abstract: Today, there are many students who travel to USA to pursue higher education. It is necessary for the students to know what are their chances of getting an admit in the universities. Also, universities manually check and count the total number of applicants who could get an admit into university. These methods are slow and certainly not very consistent for students and universities to get an actual result. This method is also prone to human error and thus accounts for some inaccuracies. Since the frequency of students studying abroad has increased, there is a need to employ more efficient systems which handle the admission process accurately from both perspectives.

➤ https://github.com/anjanatiha/University-Admission-Match-Predictor
Abstract: 1. Analyzed university admission statistics. 2. Developed tools for matching university (in percentile) using CGPA, GRE (Verbal, Quantitative, Analytical Writing) scores.

➤ https://github.com/karanwadhwa/dd-admission-predictor
Abstract: This system was originally developed only for Engineering College Admissions in Maharashtra, India but can essentially be adapted for other streams too. The purpose of it is to build a system to predict the user's chances for getting into a certain college.

2.3 PROBLEM STATEMENT DEFINITION

This is a requirement specification for a new college admissions eligibility prediction based on data science. This is an AI-based application that prompts users to enter their transcript details and calculates their chances of admission to their chosen college level. It also provides an analysis of the data and shows how the likelihood of admission depends on various factors. This document describes the scope, goals, and objectives of the system. In addition to explaining non-functional requirements.

In the current world scenario, just having a bachelor's degree is not enough. Most employers now require new employees to be highly qualified. Correspondingly, the requirements for a good university education are also higher. Many students prefer to continue their higher education. Admission to these universities requires some academic requirements. However, with so many different levels of colleges and universities, students are often left with the last-minute dilemma of whether or not their application will be accepted due to the lack of specific documentation stating their requirements. Developed after examining many leading machine learning algorithms, our AI models are the most accurate predictors of a student's admissions likelihood based on their current college-level grades and other academic background of their choice. To do. The purpose of this project is for users to enter their academic data and receive predictions regarding their likelihood of admission to their desired college level. The system will also implement a database so students can store data, review and edit as they progress, and their latest predictions will be saved in their profile. Aspects of web security other than password protection within the site are not part of this project.

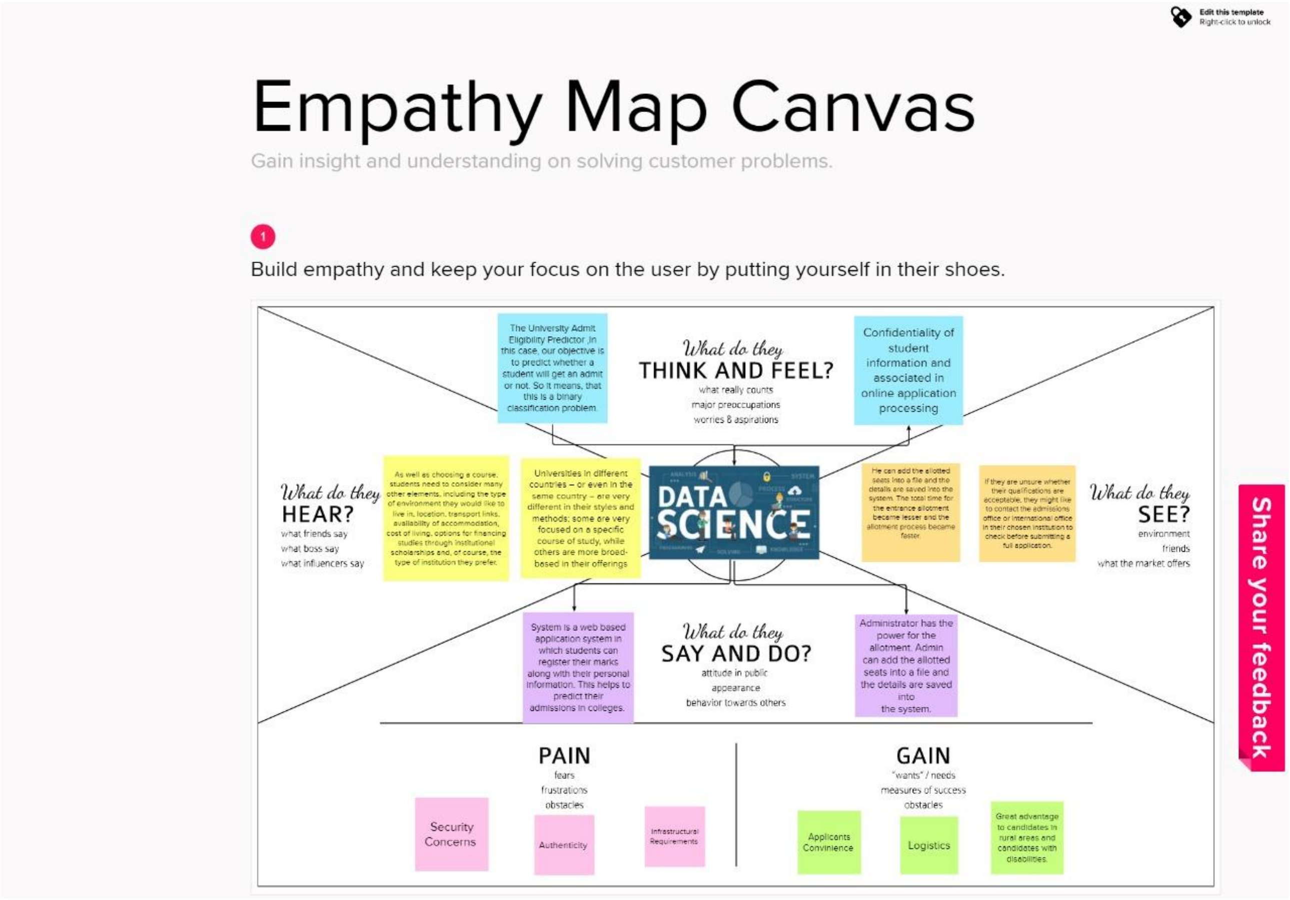
Students - Indian students will benefit most from this scheme.

Administrators - Administrators should have access to all data stored in the application. The system is available to all users from anywhere as long as they have an internet connection. The administrator can access her website from anywhere with the correct credentials and internet access.

Overview of Document: The system should answer students' questions competently and succinctly, and provide the most accurate analysis possible of their chances of getting into college. The system is based on a limited data set, which can affect the overall accuracy of predictions. Many other factors such as personal interviews also play a large role in the admission process, so the system cannot guarantee that our predictions are 100% guaranteed admission. Other factors, such as changes in colleges and university policies, may also affect your chances of admission in ways beyond the scope of this project.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

The University Admit Eligibility Predictor ,In this case, our objective is to predict whether a student will get an admit or not. So it means, that this is a binary classification problem.

Confidentiality of student information and associated in online application processing

What do they HEAR?
what friends say
what boss say
what influencers say

As well as choosing a course, students need to consider many other elements, including the type of environment they would like to live in, location, transport links, availability of accommodation, cost of living, options for financing studies through institutional scholarships and, of course, the type of institution they prefer.

Universities in different countries – or even in the same country – are very different in their styles and methods; some are very focused on a specific course of study, while others are more broad-based in their offerings

What do they SEE?
environment
friends
what the market offers

He can add the allotted seats into a file and the details are saved into the system. The total time for the entrance allotment became lesser and the allotment process became faster.

If they are unsure whether their qualifications are acceptable, they might like to contact the admissions office or international office in their chosen institution to check before submitting a full application.

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

System is a web based application system in which students can register their marks along with their personal information. This helps to predict their admissions in colleges.

Administrator has the power for the allotment. Admin can add the allotted seats into a file and the details are saved into
the system.

PAIN
fears
frustrations
obstacles

Security Concerns

Authenticity

Infrastructural Requirements

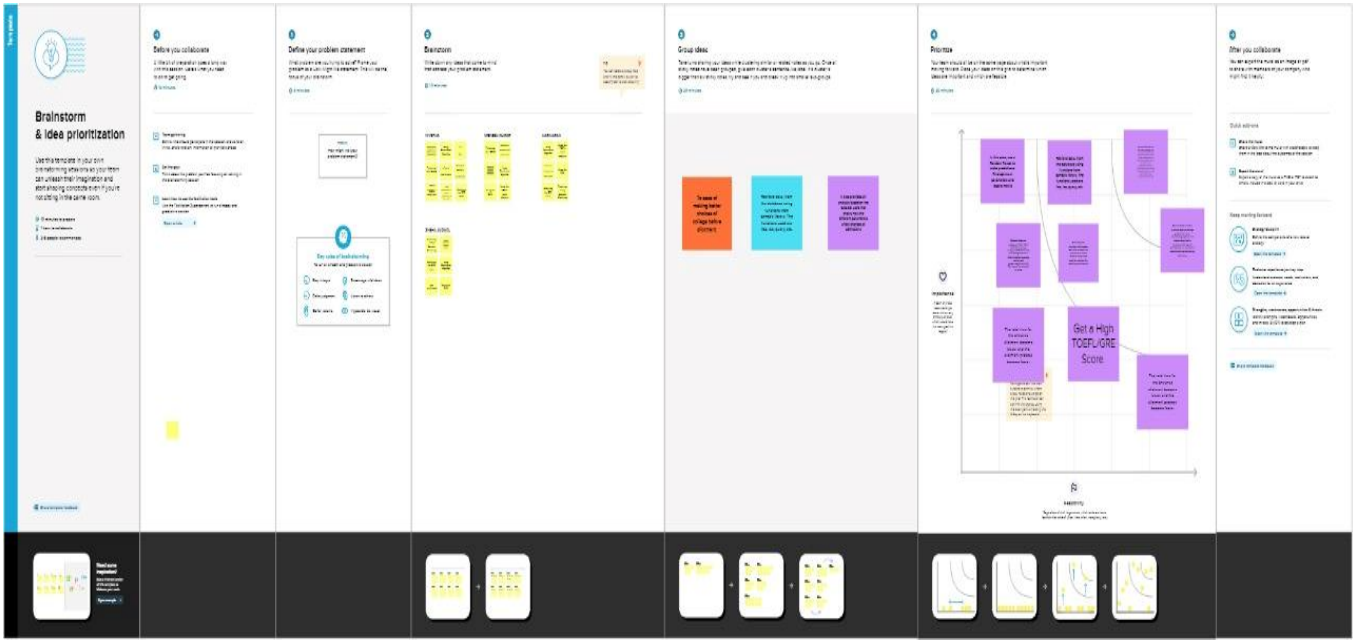
GAIN
"wants" / needs
measures of success
obstacles

Applicants Convinience

Logistics

Great advantage to candidates in rural areas and candidates with disabilities.

3.2 Ideation & Brainstorming



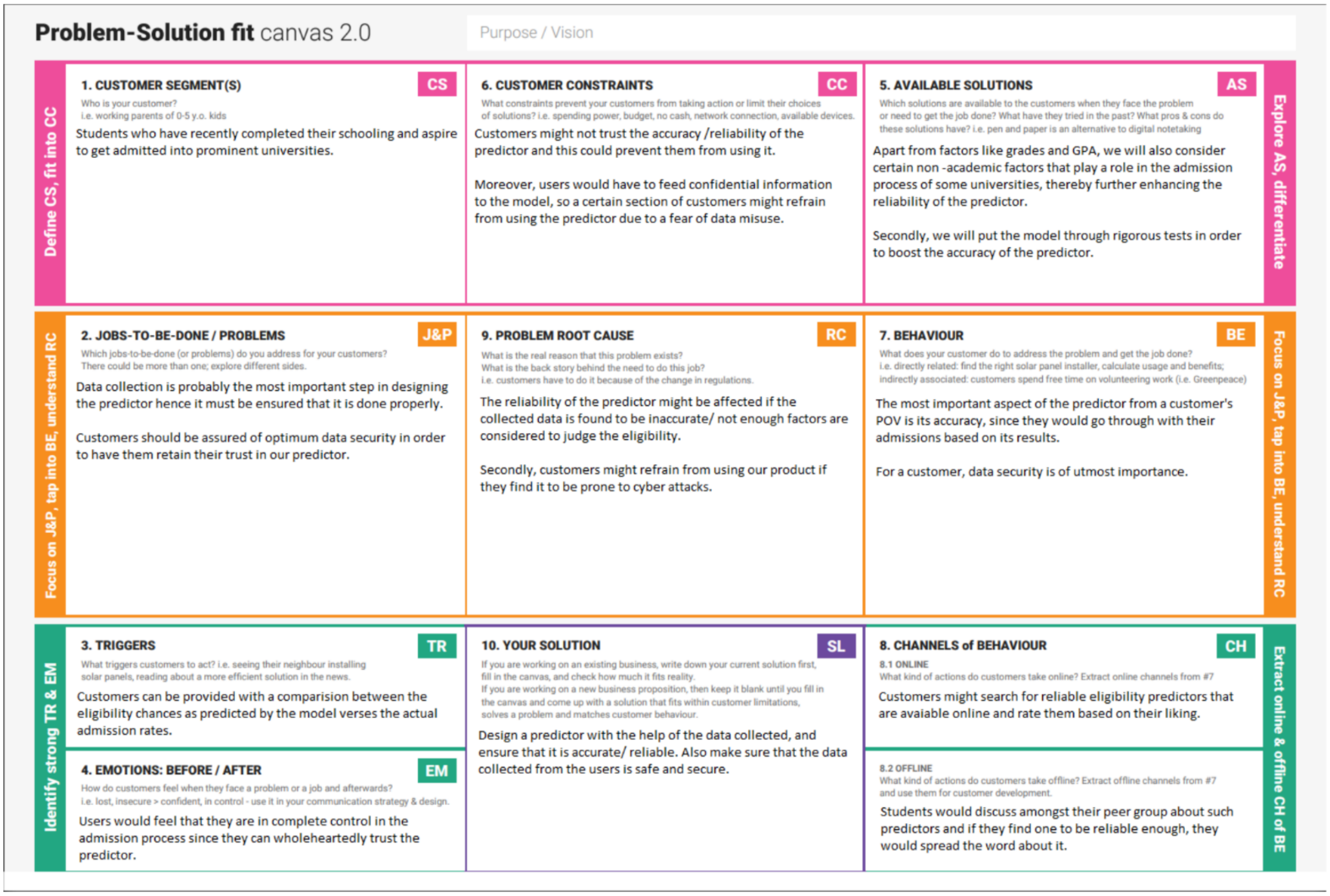
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

3.3 Proposed Solution :

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	How might we design an eligibility predictor for students which provides them with their chances of getting admitted into different universities based on their scoresand other important criteria.
2.	Idea / Solution description	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 regarding the admission process. A model is developed which analyses the data providedby the user and evaluates it in accordance with the algorithm developed to predict the eligibility of the user for the specified university.
3.	Novelty / Uniqueness	We aim to design the model in such a way that it takes certain non-academic factors which influence the admission process into consideration as well. This further enhances the accuracy of the predictor. This attribute is not considered in most predictors availablein the market.

4.	Social Impact / Customer Satisfaction	This predictor would provide a clarity to passed out students who might be confused regarding their future with respect to university admissions. The students can apply to universities based on their eligibility chances.
5.	Business Model (Revenue Model)	Such predictors have a huge demand in the market since students who complete their schooling are always in need of tools like this to plan out their universityadmissions.
6.	Scalability of the Solution	The scope of this predictor is very wide as a large number of universities could be brought within the range of this predictor depending on the requirementsof the user. Hence, this solution is largely scalable in nature.

3.4 Problem Solution fit:


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

Problem-Solution fit canvas 2.0

Purpose / Vision

1. CUSTOMER SEGMENT(S)

Who is your customer?
i.e. working parents of 0-5 y.o. kids

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

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

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

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Apart from factors like grades and GPA, we will also consider certain non -academic factors that play a role in the admission process of some universities, thereby further enhancing the reliability of the predictor.

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

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

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 have them retain their trust in our predictor.

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

The reliability of the predictor might be affected if the collected data is found to be inaccurate/ 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

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

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

For a customer, data security is of utmost importance.

3. TRIGGERS

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Customers can be provided with a comparision between the eligibility chances as predicted by the model verses the actual admission rates.

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

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

10. YOUR SOLUTION

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

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

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

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

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

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

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	• Registration through Form • Registration through Gmail • Registration through LinkedIn
FR-2	User Confirmation	• Confirmation via Email • Confirmation via OTP
FR-3	User Requirements	• All the needed files are been asked to feedin the website. • Based on the uploads, the system would collect all the necessary information. • The information includes the list of all the possible universities and streams.
FR-4	User Details	Has to feed some documents • Score Sheets • Letter of Recommendation (LOR) • Statement of Purpose (SOP) • Curriculum Vitae (CV)

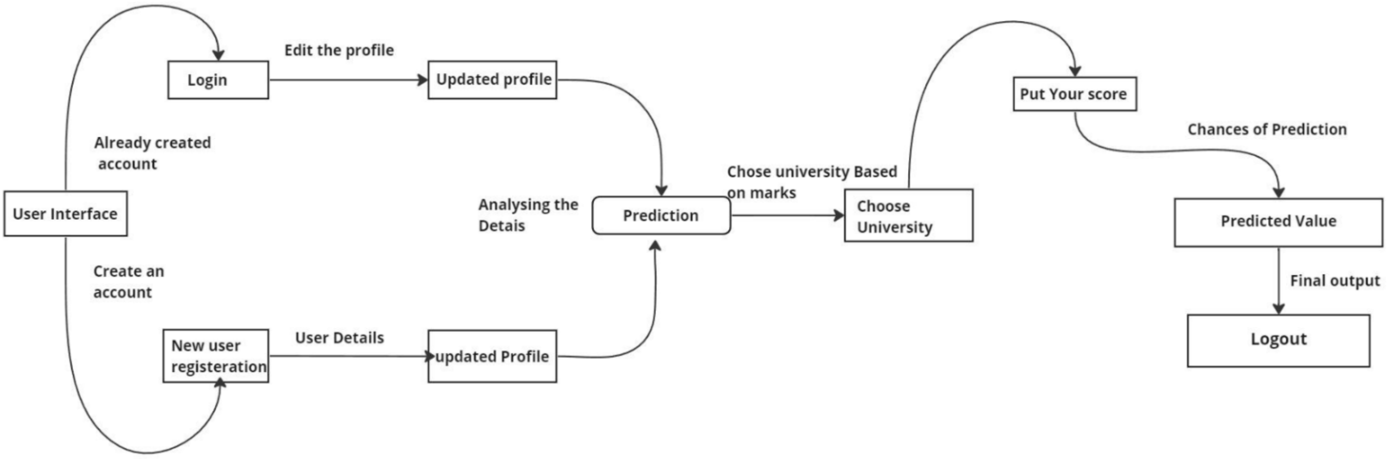
4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

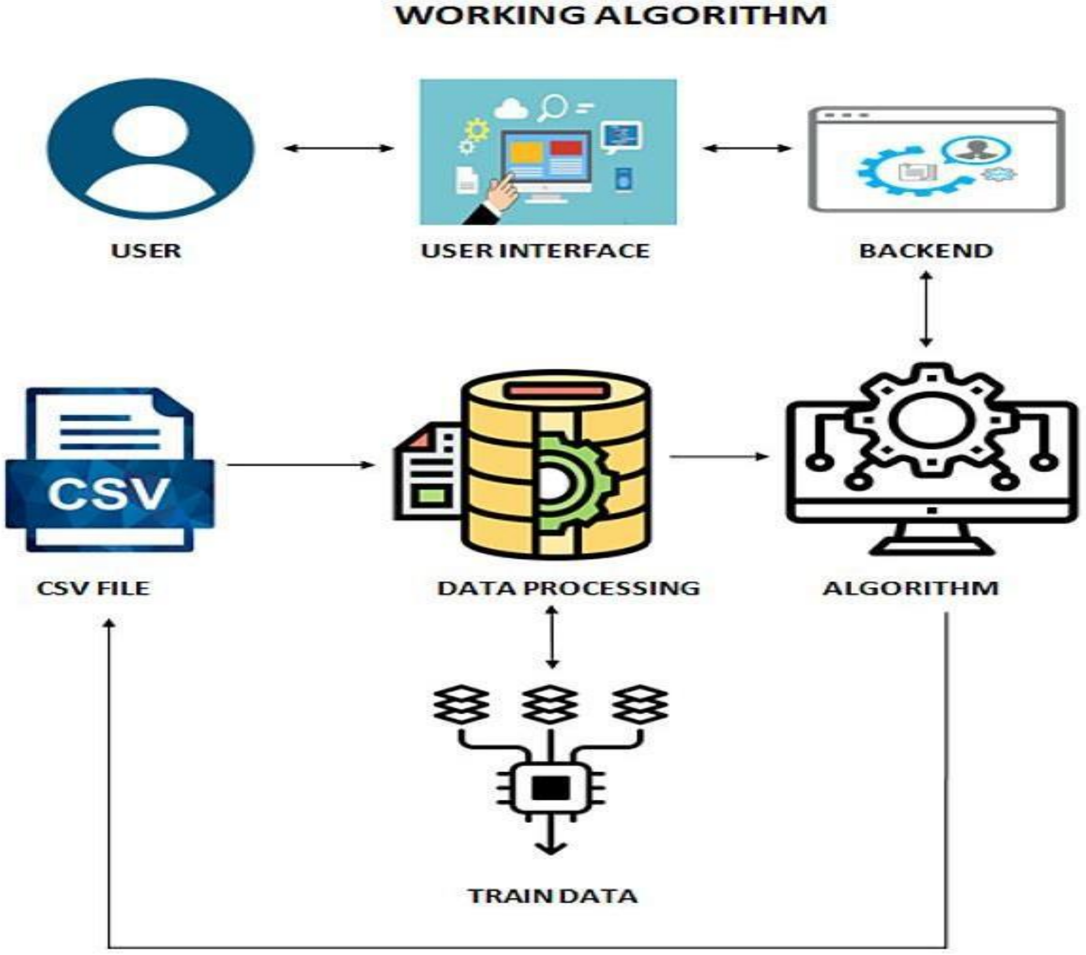
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• Our website is very user friendly. • There is no need for any technical skill in order to access our website. • The page would not take a lot of time to load the content.
NFR-2	**Security**	• The user who is having the valid credentials can able to access our site.
		• Under any error, the system should be ableto come back to regular operation in under an hour.
NFR-3	**Reliability**	• Our website is more reliable. • Since nobody can able to see the data fedby the user.
NFR-4	**Performance**	• User can able to access in our website with internet connection. • Traffics can be handled effectively.
NFR-5	**Availability**	• Fast and efficient. • Students can access our website from anyof • the available browser.

5. PROJECT DESIGN

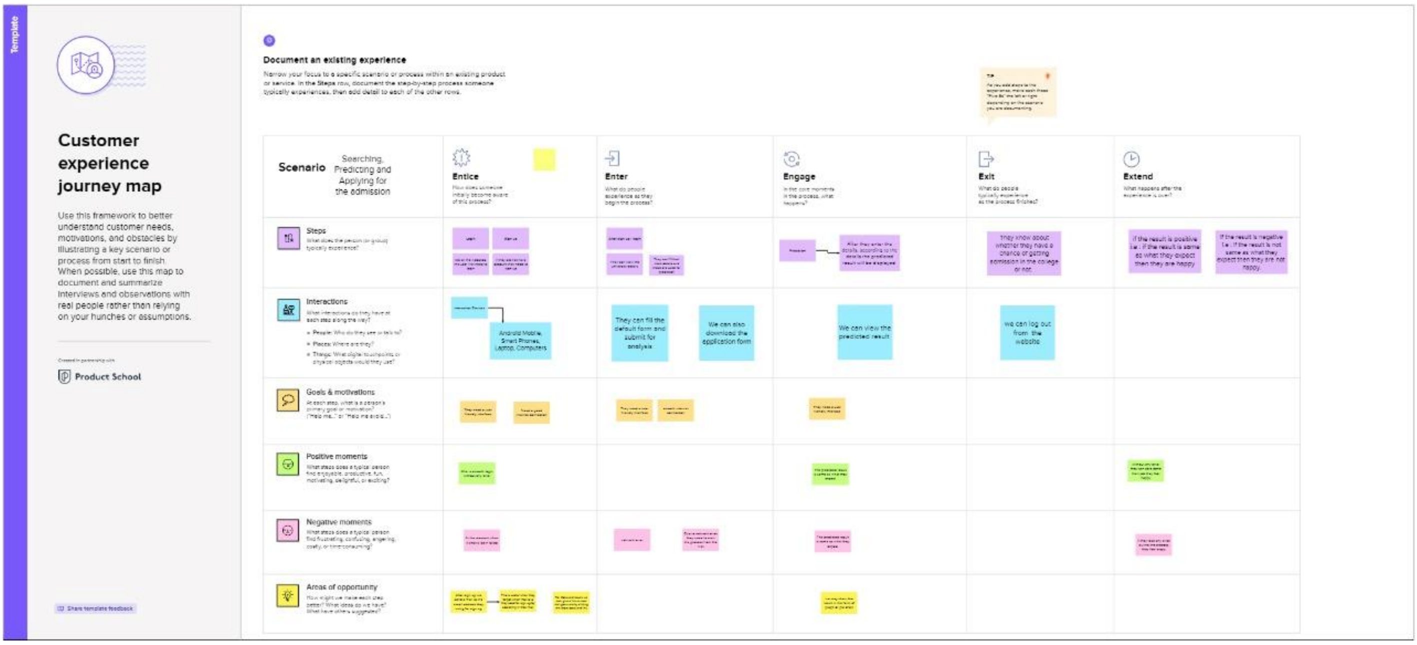
5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories



Use this framework to better understand customer needs, motivations, and obstacles by illustrating a key scenario or process from start to finish. When possible, use this map to document and summarize interviews and observations with real people rather than relying on your hunches or assumptions.

Document an existing experience

Narrow your focus to a specific scenario or process within an existing product or service. In the Steps row, document the step-by-step process someone typically experiences, then add detail to each of the other rows.

Scenario: Searching, Predicting and Applying for the admission

Android Mobile, Smart Phones, Laptop, Computers

They can fill the default form and submit for analysis

We can also download the application form

We can view the predicted result

we can log out from the website

They know about whether they have a chance of getting admission in the college or not.

If the result is positive i.e : if the result is same as what they expect then they are happy

If the result is negative i.e : if the result is not same as what they expect then they are not happy.

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

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	7 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-2	20	4 Days	06 Oct 2022	08 Nov 2022	20	09 Nov 2022
Sprint-3	20	4 Days	09 Nov 2022	11 Nov 2022	20	11 Nov 2022
Sprint-4	20	4 Days	12 Nov 2022	14 Nov 2022	20	15 Nov 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, you can register in the application by entering your email address, password, and confirming the password	2	High	Nivetha K

Sprint-1		USN-2	As a user, you will receive a confirmation email after registering in the application	1	High	Vignesh Kumar K
Sprint-2		USN-3	As a user, you can register in the application via Facebook	2	Low	Hariharan SV
Sprint-1		USN-4	As a user, you can register in the application via Gmail	2	Medium	Sneha Jaiswal
Sprint-1	Login	USN-5	As a user, you can login to the application by entering your email and password	1	High	Vignesh Kumar K

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points perday)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often usedin agile software development methodologies such as Scrum. However, burn down charts canbe applied to any project containing measurable progress over time.

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

7.1 Feature 1

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>
    UNIVERSITY ADMISSION PREDICTOR
    </title>
    <link rel="stylesheet" href="/static/styles/style.css">
</head>

<body>
    <h1 style="color: aliceblue; background-color: black;">UNIVERSITY ADMISSION PREDICTOR</h1>

    <form id="form" action="/prediction" method="post">

        <div class="form-control">
            <label for="name" id="label-name">ENTER GRE SCORE:</label>
            <input type="text" name="grescore" id="grescore" placeholder="90" />
        </div>

        <div class="form-control">
            <label for="email" id="label-email">ENTER TOEFL SCORE:</label>
            <input type="text" name="toeflscore" id="toeflscore" placeholder="90" />
        </div>

        <div class="form-control">
            <label for="university">SELECT UNIVERSITY RATING:</label>
            <select name="university" id="university">
                <option value="select university no">SELECT UNIVERSITY NO</option>
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
                <option value="4">4</option>
                <option value="5">5</option>
            </select>
        </div>

        <div class="form-control">
            <label for="name" id="label-name">ENTER SOP:</label>
            <input type="text" name="sop" id="sop" placeholder="1" />
        </div>

        <div class="form-control">
            <label for="name" id="label-name">ENTER LOR:</label>
            <input type="text" name="lor" id="lor" placeholder="3" />
        </div>

        <div class="form-control">
            <label for="name" id="label-name">ENTER CGPA:</label>
            <input type="text" name="cgpa" id="cgpa" placeholder="8.9" />
        </div>

        <div class="form-control">
            <label >RESEARCH:</label>
            <label for="recommed-1">
                <input type="radio" id="recommed-1" name="recommed-1">RESEARCH</input>
            </label>
            <label for="recommed-2">
            <input type="radio" id="recommed-2" name="recommed-1">NO RESEARCH</input>
          </label>
        </div>
        <button type="submit" value="submit" href="#"> PREDICT </button>
    </form>
</body>

</html>
```

7.2 Feature 2

Eligible

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>

</head>
<body>
    <img style="margin-top:100px;height:220px; width:220px" src="/static/bg3.jpg" class="img-responsive center-block d-block mx-auto" alt="Sample Image">
    <center><p style="font-size: 50px;">HURRAY!! YOU ARE ELIGIBLE FOR THIS UNIVERSITY</p></center>
</body>
</html>
```

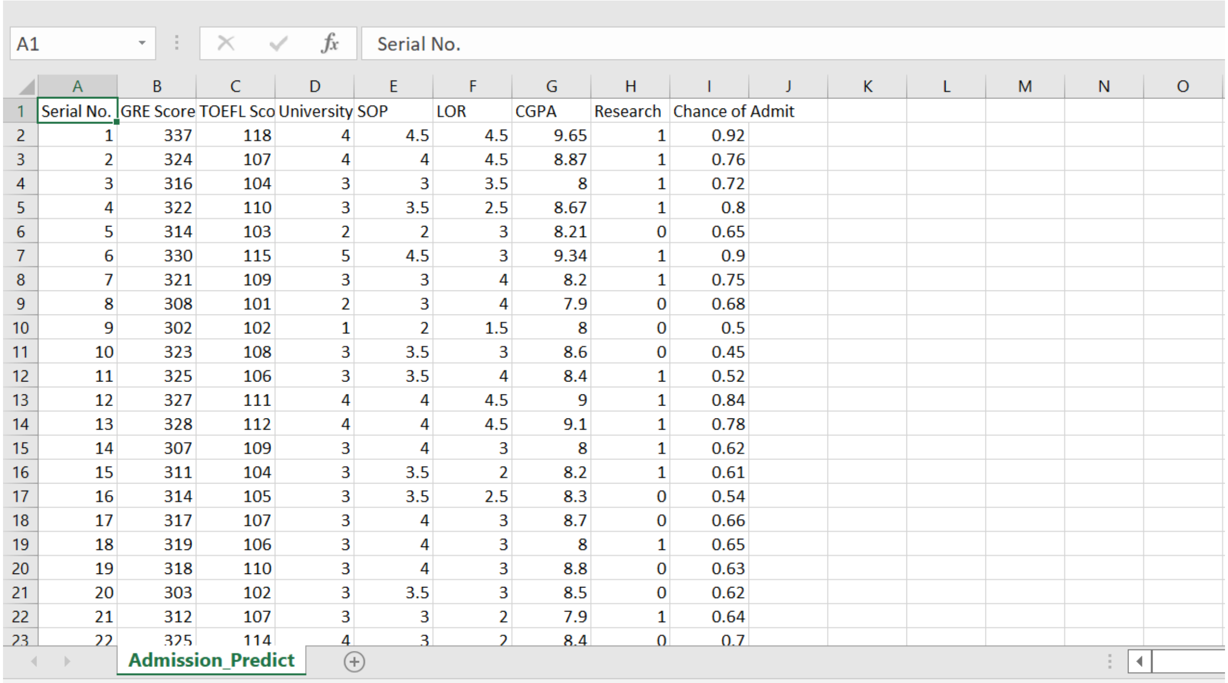
Not Eligible

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>

</head>
<body>
```

```
    <img style="margin-top:100px;height:220px; width:220px" src="/static/bg2.png" class="img-responsive center-
block d-block mx-auto" alt="Sample Image">
    <center><p style="font-size: 50px;">SORRY!! YOU ARE NOT ELIGIBLE FOR THIS
UNIVERSITY</p></center>
</body>
</html>
```

7.3 Database schema



Serial No.	GRE Score	TOEFL Sco	University	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
11	325	106	3	3.5	4	8.4	1	0.52
12	327	111	4	4	4.5	9	1	0.84
13	328	112	4	4	4.5	9.1	1	0.78
14	307	109	3	4	3	8	1	0.62
15	311	104	3	3.5	2	8.2	1	0.61
16	314	105	3	3.5	2.5	8.3	0	0.54
17	317	107	3	4	3	8.7	0	0.66
18	319	106	3	4	3	8	1	0.65
19	318	110	3	4	3	8.8	0	0.63
20	303	102	3	3.5	3	8.5	0	0.62
21	312	107	3	3	2	7.9	1	0.64
22	325	114	4	3	2	8.4	0	0.7

8. TESTING

8.1 Test Cases

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
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8.2 User Acceptance Testing

Purpose of Document

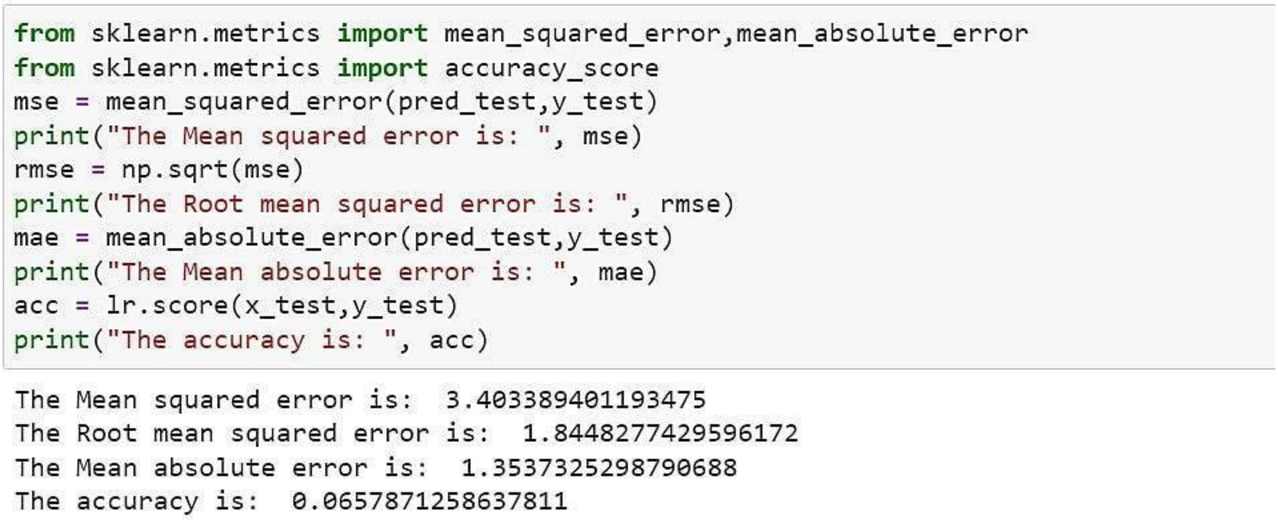
The purpose of this document is to briefly explain the test coverage and open issues of the University Admit Eligibility Predictor project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

Resolution	**Severity 1**	**Severity 2**	**Severity 3**	**Severity 4**	**Subtotal**
By Design	10	4	2	3	19
Duplicate	0	0	0	0	0
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	24	14	13	26	64

9. RESULTS

9.1 Performance Metrics



```
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.metrics import accuracy_score
mse = mean_squared_error(pred_test,y_test)
print("The Mean squared error is: ", mse)
rmse = np.sqrt(mse)
print("The Root mean squared error is: ", rmse)
mae = mean_absolute_error(pred_test,y_test)
print("The Mean absolute error is: ", mae)
acc = lr.score(x_test,y_test)
print("The accuracy is: ", acc)
```

```
The Mean squared error is:  3.403389401193475
The Root mean squared error is:  1.8448277429596172
The Mean absolute error is:  1.3537325298790688
The accuracy is:  0.0657871258637811
```

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

1. It helps student for making decision for choosing a right college.
2. Here the chance of occurrence of error is less when compared with the existing system.
3. It is fast, efficient and reliable.

10.2 Disadvantages:

1. Required active internet connection.
2. System will provide inaccurate results if data entered incorrectly.

11. CONCLUSION

This system ,being the first we have created in Python using ML algorithms and other front end languages such as html, css, java script , has proven more difficult than originally imagined. While it may sound simple to fill out a few forms and process the information, much more is involved in the selection of applicants than this. Every time progress was made and features were added, ideas for additional features or methods to improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself. Debugging can sometimes be a relatively straight forward process, or rather rather finding out what you must debug can be. Since so many parts of the admissions system are integrated into one another, if an error occurs on one page, it may be a display error, for example; it may be the information is not correctly read from the database; or even that the information is not correctly stored in the database initially, and all three must be checked on each occasion. This slows down the process and can be frustrating if the apparent cause of a problem is not obvious at first. Language used must be simple and easy to understand and compatibility is paramount. If this system were not designed as an entirely web based application, it would not have been possible to recreate its current state of portability. Overall, the system performs well,and while it does not include all of the features that may have been desired, it lives up to initial expectations. The majority of features that are included work flawlessly and the errors that do exist are minor or graphical.

12. FUTURE SCOPE

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

- 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

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

2. Reading the dataset.

In [2]:

```
admission = pd.read_csv("dataset/Admission_Predict.csv") # readin the dataset
```

In [3]:

```
admission.head() # to see the top five records of the data sets
```

Out[3]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

In [4]:

```
admission.shape # to  see what is the shape of data_set our data set has 400 records
and 9 fields
```

Out[4]:

```
(400, 9)
```

In [5]:

```
admission.columns #to see the name of the fields
```

Out[5]:

```
Index(['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating', 'SOP',
       'LOR ', 'CGPA', 'Research', 'Chance of Admit '],
      dtype='object')
```

In [6]:

```
admission.describe() # to see the mathematical values of the data sets i.e
mean,standar_deviation ,minimum_value,maximum_value,counts etc.
```

Out[6]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	200.500000	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	115.614301	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	1.000000	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	100.750000	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
50%	200.500000	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	300.250000	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	400.000000	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

In [7]:

```
admission.info() #to see the type of values in every fields i.e int ,float etc
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Serial No.         400 non-null    int64  
 1   GRE Score          400 non-null    int64  
 2   TOEFL Score        400 non-null    int64  
 3   University Rating  400 non-null    int64  
 4   SOP                400 non-null    float64
 5   LOR                400 non-null    float64
 6   CGPA               400 non-null    float64
 7   Research           400 non-null    int64  
 8   Chance of Admit    400 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
```

In [8]:

```
admission.isnull().sum() # to see that if dataset has any null_values or not
```

Out[8]:

```
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

3. Finding the dependent and independent variable.

In [9]:

```
X=admission.drop(['Serial No.','Chance of Admit '],axis=1) #input data_set
X.shape
```

Out[9]:

```
(400, 7)
```

In [10]:

```
y=admission['Chance of Admit '] #output labels
y.shape
```

Out[10]:

```
(400,)
```

In [11]:

```
admission.sample(5)
```

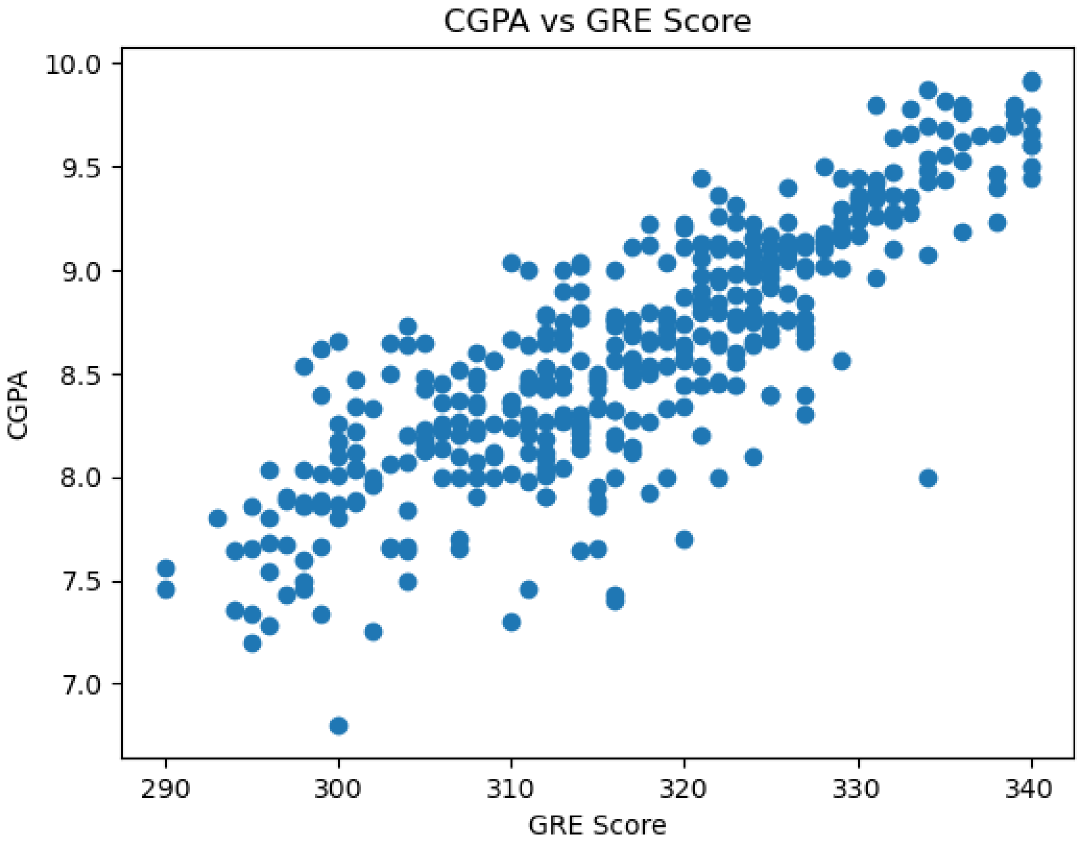
Out[11]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
347	348	299	94	1	1.0	1.0	7.34	0	0.42
397	398	330	116	4	5.0	4.5	9.45	1	0.91
207	208	310	102	3	3.5	4.0	8.02	1	0.66
327	328	295	101	2	2.5	2.0	7.86	0	0.69
224	225	305	105	2	3.0	2.0	8.23	0	0.67

4. Data Visualization

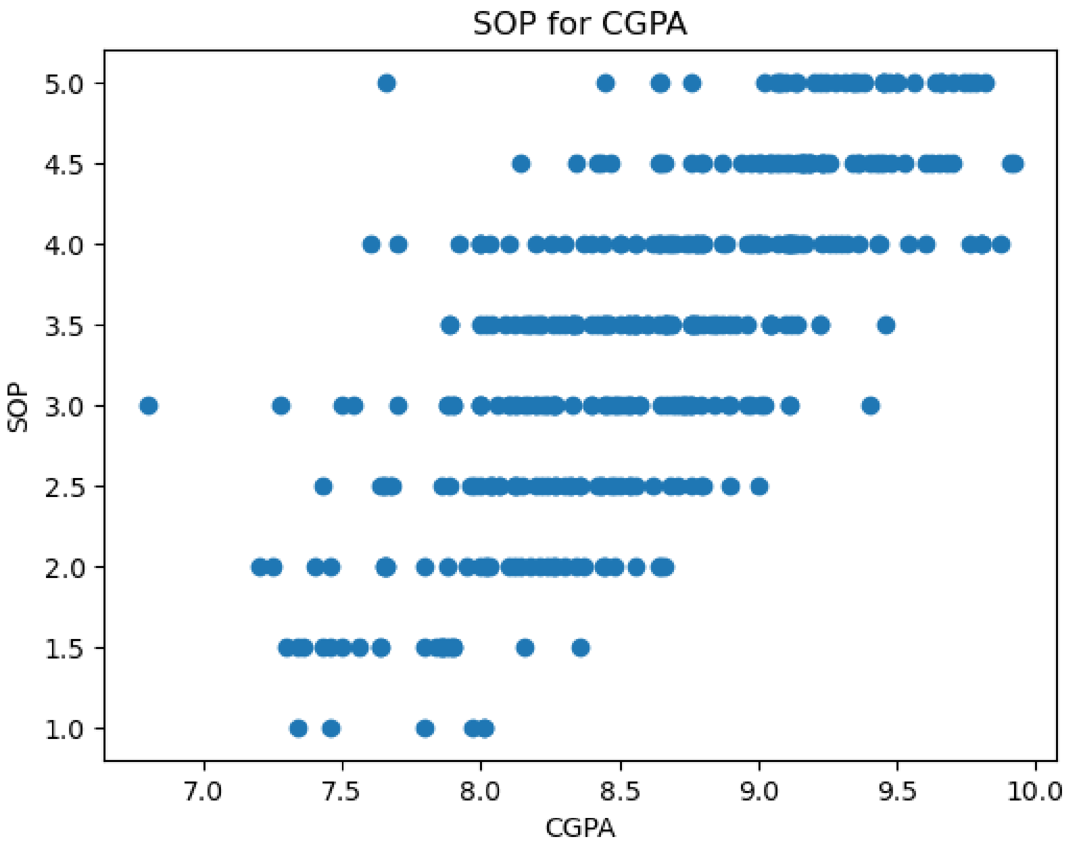
In [12]:

```
plt.scatter(admission['GRE Score'],admission['CGPA'])
plt.title('CGPA vs GRE Score')
plt.xlabel('GRE Score')
plt.ylabel('CGPA')
plt.show()
```



In [13]:

```
plt.scatter(admission['CGPA'],admission['SOP'])
plt.title('SOP for CGPA')
plt.xlabel('CGPA')
plt.ylabel('SOP')
plt.show()
```

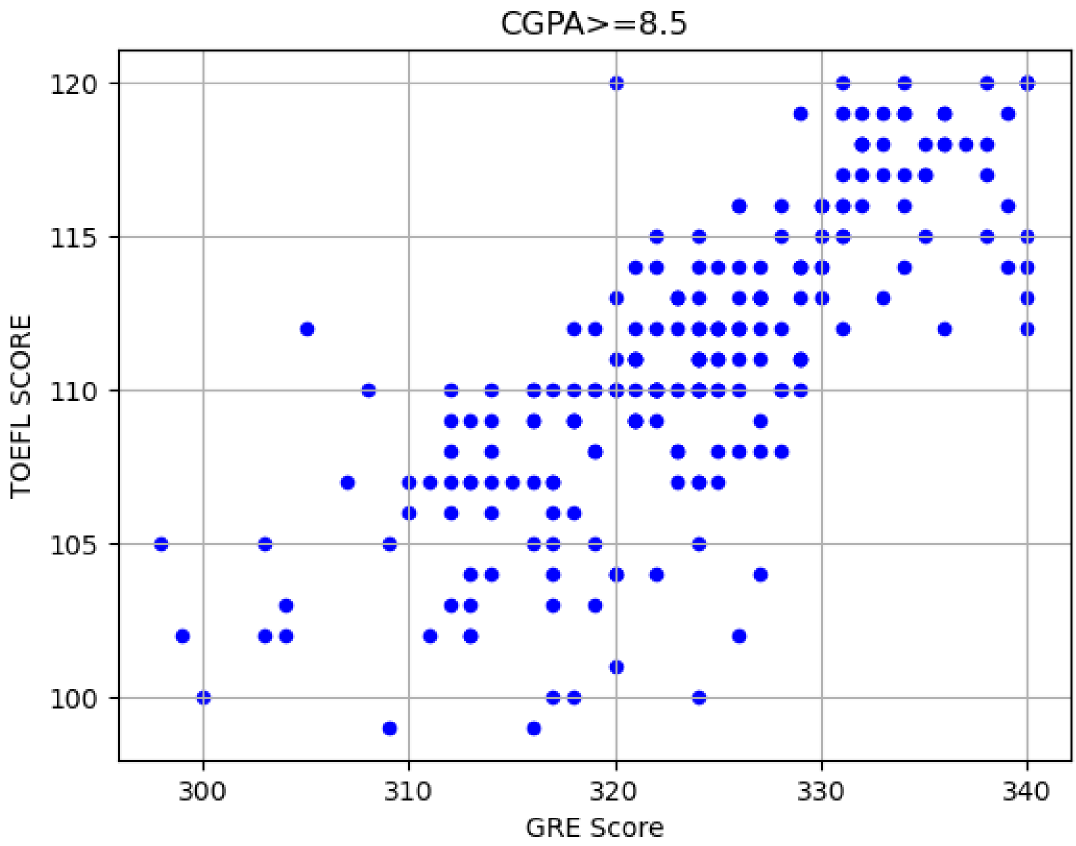


In [14]:

```
admission[admission.CGPA >= 8.5].plot(kind='scatter', x='GRE Score', y='TOEFL
Score',color="BLUE")

plt.xlabel("GRE Score")
plt.ylabel("TOEFL SCORE")
plt.title("CGPA>=8.5")
plt.grid(True)

plt.show()
```

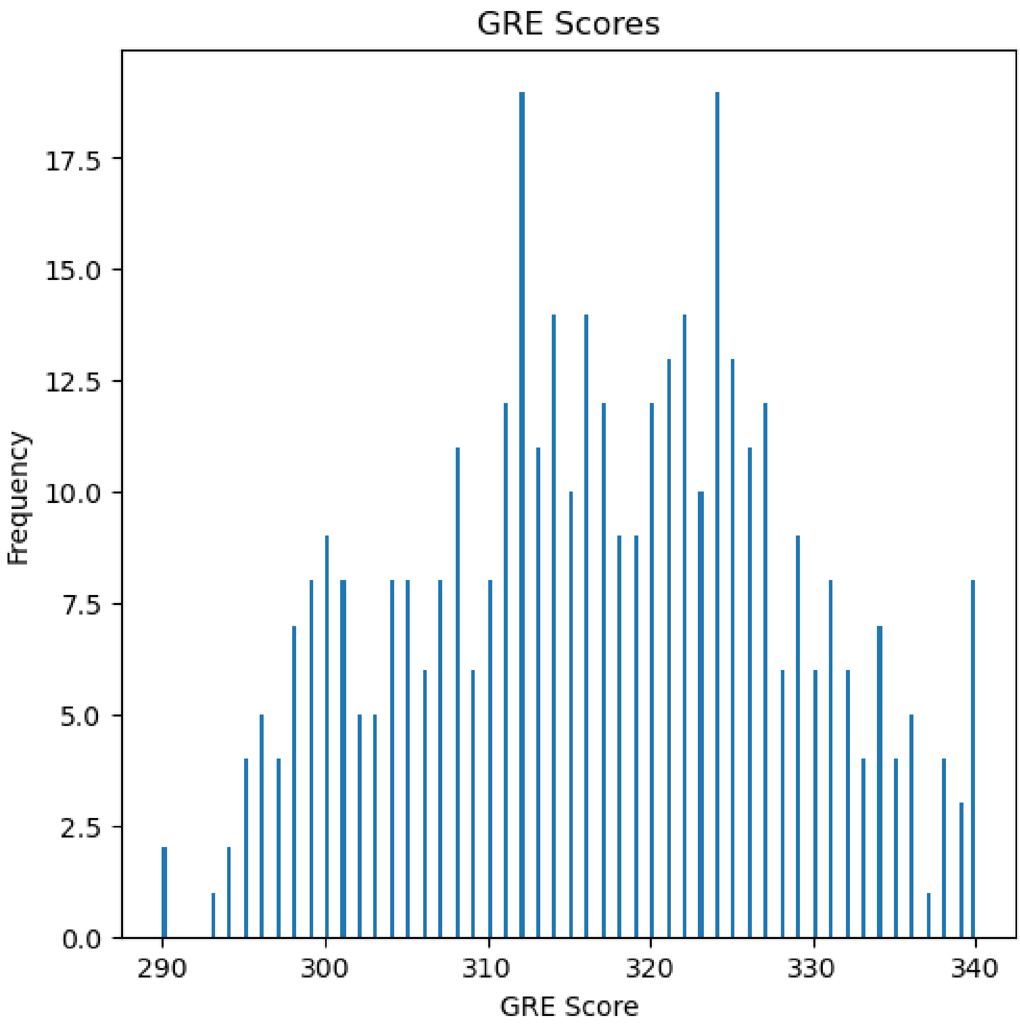


In [15]:

```
admission["GRE Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))

plt.title("GRE Scores")
plt.xlabel("GRE Score")
plt.ylabel("Frequency")

plt.show()
```

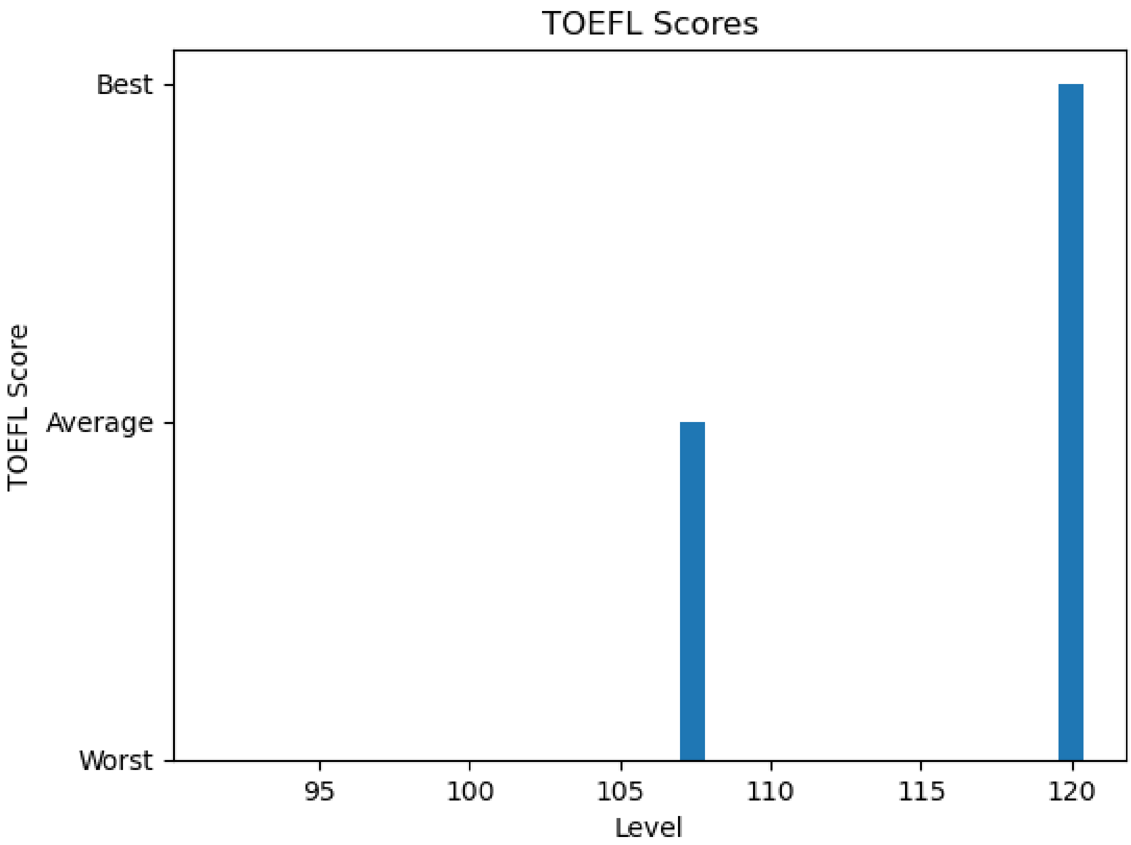


In [16]:

```
p = np.array([admission["TOEFL Score"].min(),admission["TOEFL
Score"].mean(),admission["TOEFL Score"].max()])
r = ["Worst","Average","Best"]
plt.bar(p,r)

plt.title("TOEFL Scores")
plt.xlabel("Level")
plt.ylabel("TOEFL Score")

plt.show()
```

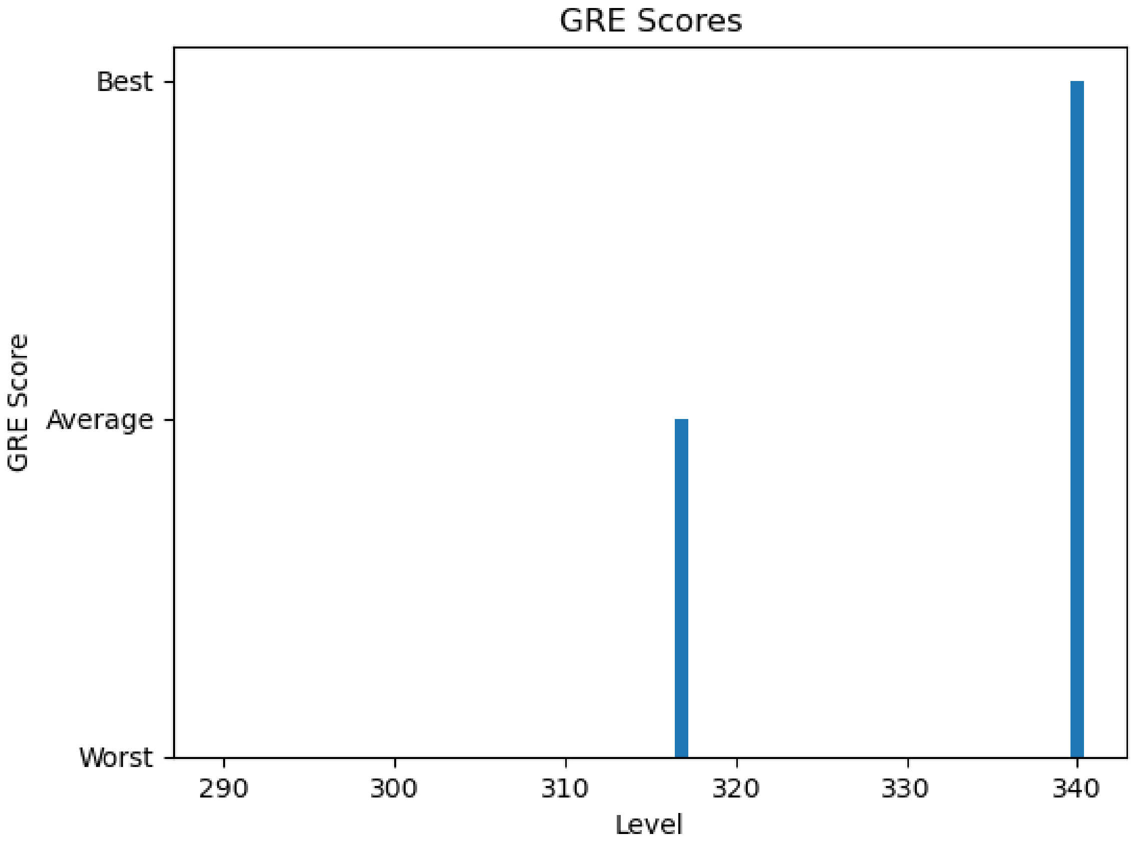


In [17]:

```
g = np.array([admission["GRE Score"].min(),admission["GRE
Score"].mean(),admission["GRE Score"].max()])
h = ["Worst","Average","Best"]
plt.bar(g,h)

plt.title("GRE Scores")
plt.xlabel("Level")
plt.ylabel("GRE Score")

plt.show()
```



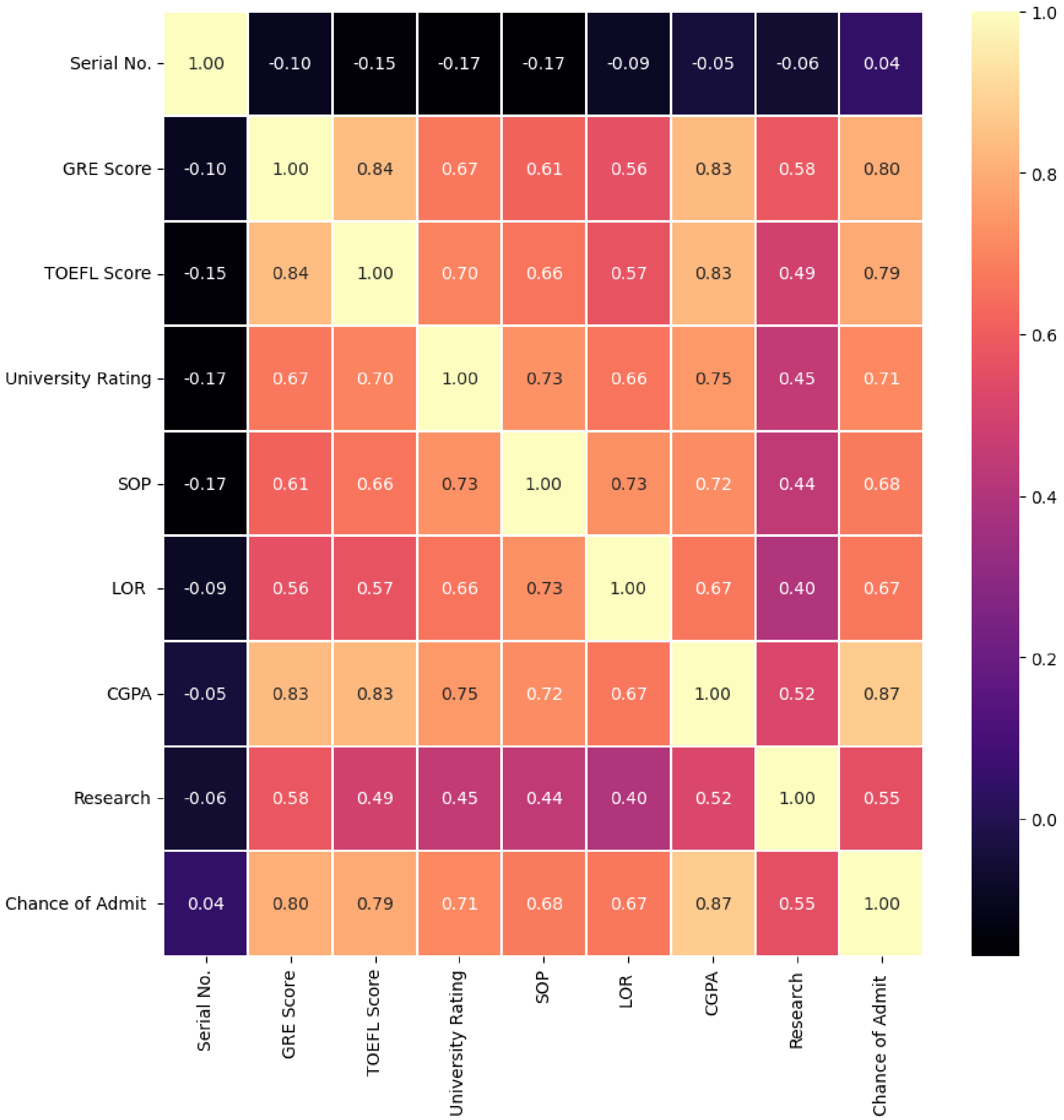
In [18]:

```
import seaborn as sns

plt.figure(figsize=(10, 10))

sns.heatmap(admission.corr(), annot=True, linewidths=0.05, fmt= '.2f',cmap="magma")

plt.show()
```

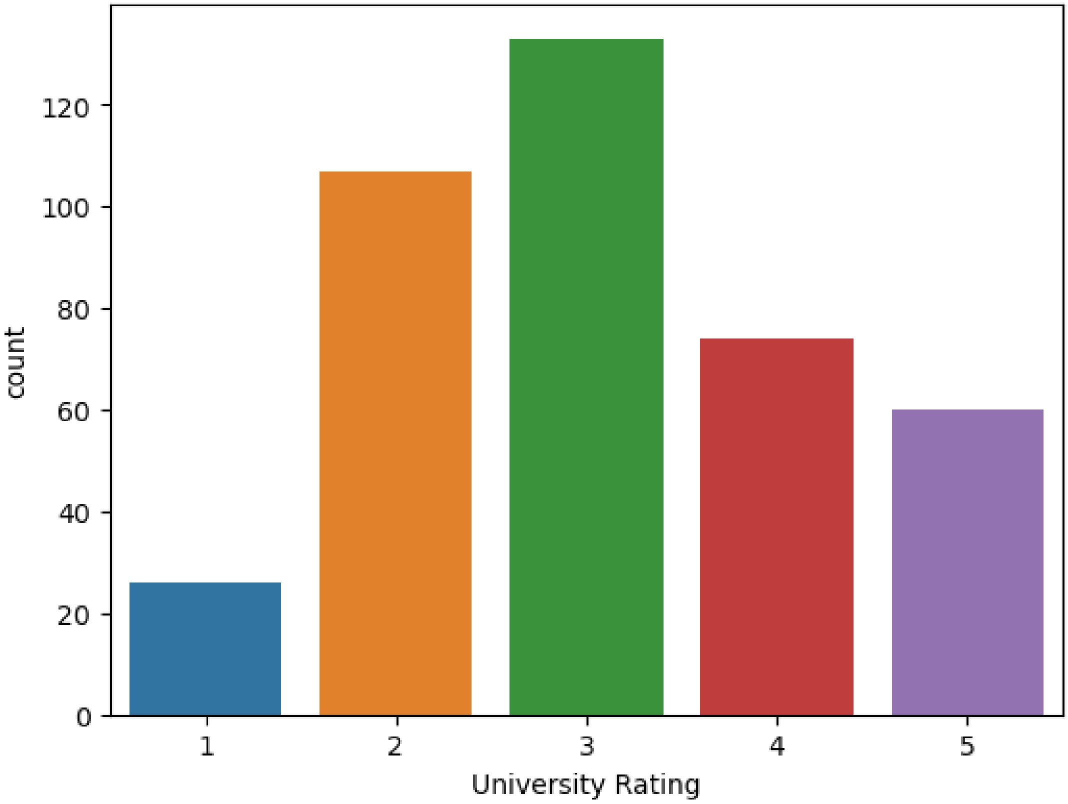


In [19]:

```
admission.Research.value_counts()

sns.countplot(x="University Rating",data=admission)
```

Out[19]:

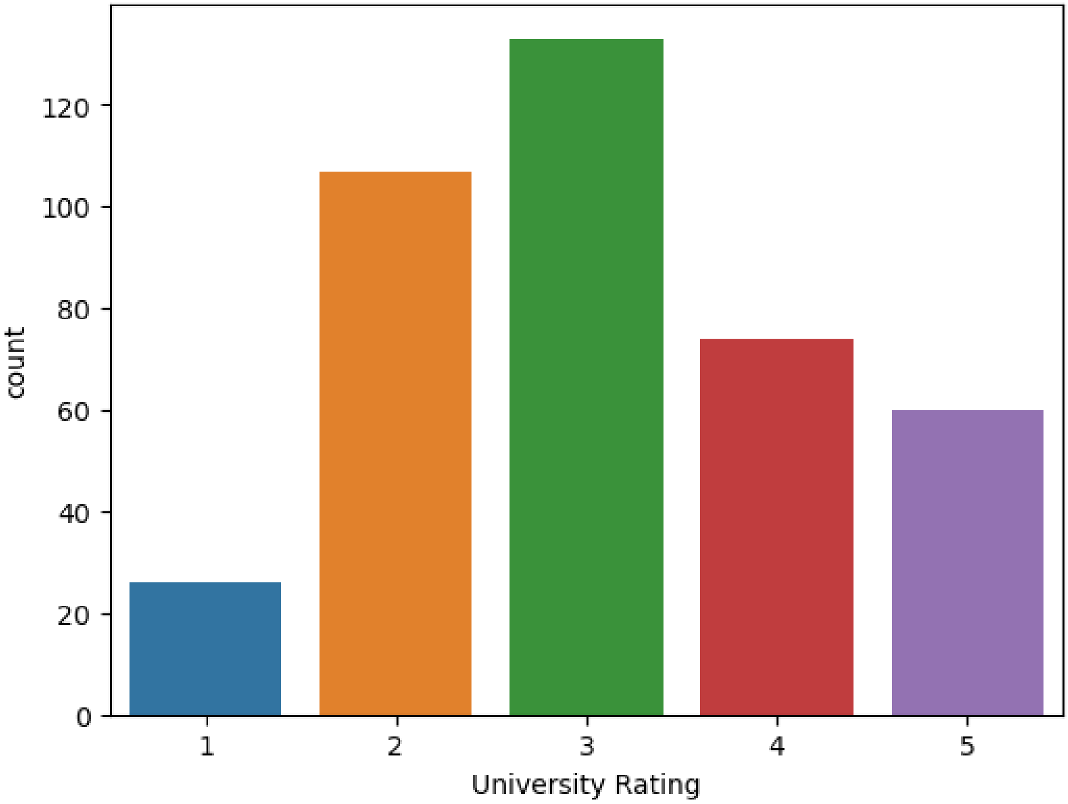


In [20]:

```
admission.Research.value_counts()

sns.countplot(x="University Rating",data=admission)
```

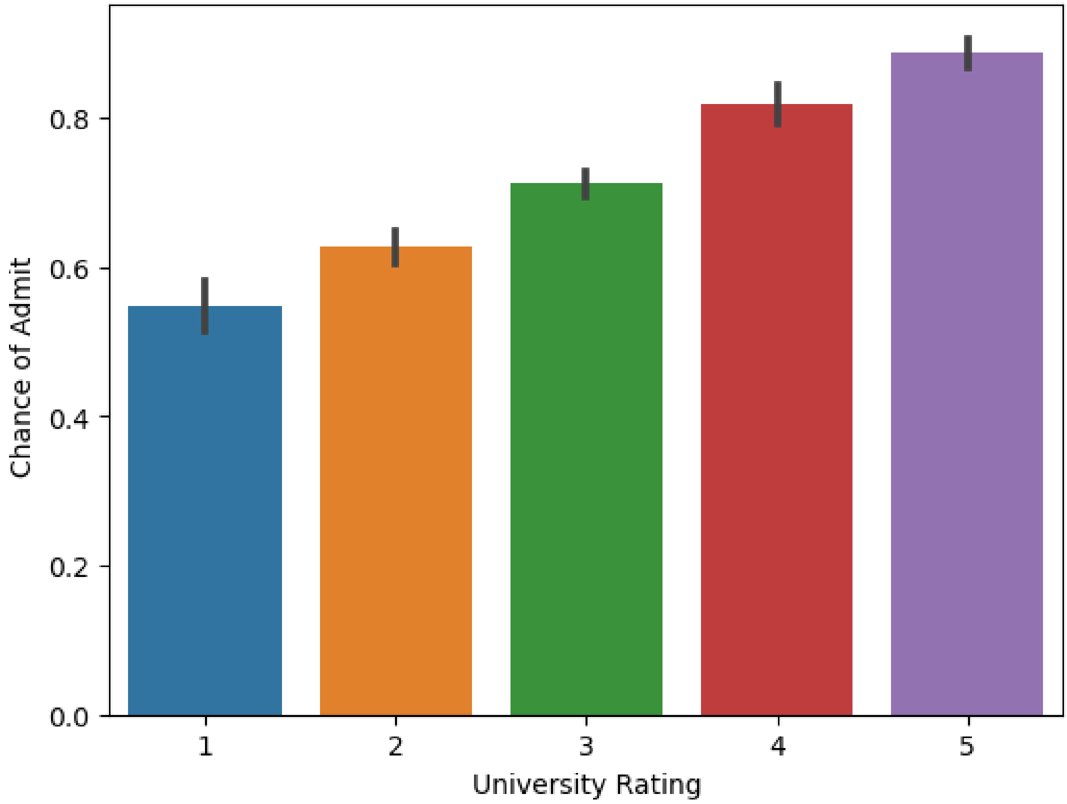
Out[20]:



In [21]:

```
sns.barplot(x="University Rating", y="Chance of Admit ", data=admission)
```

Out[21]:



5. Train Test splitting.

In [22]:

```
#splittin the input data(x) and output labels(y) into train data and test data
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2) # test_size defins
the volume of train data and test data here 0.2 means 20% of the data belongs to the
test data
```

In [23]:

```
X_train.shape
```

Out[23]:

```
(320, 7)
```

In [24]:

```
X_test.shape
```

Out[24]:

```
(80, 7)
```

6. Data Preprocessing.

In [25]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
X_train[X_train.columns] = scaler.fit_transform(X_train[X_train.columns].values)
X_test[X_test.columns] = scaler.transform(X_test[X_test.columns].values)
X_train.head()
```

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
373	0.586957	0.607143	0.50	0.500	0.500	0.492647	1.0

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
22	0.739130	0.857143	1.00	1.000	1.000	0.845588	1.0
387	0.282609	0.464286	0.25	0.250	0.625	0.330882	0.0
123	0.304348	0.571429	0.50	0.625	0.625	0.375000	0.0
110	0.239130	0.571429	1.00	0.500	0.500	0.470588	0.0

7. Selecting the ML model.

In [26]:

```
from sklearn.ensemble import RandomForestRegressor
rgr=RandomForestRegressor()
rgr.fit(X_train,y_train)
y_pred = rgr.predict(X_test)
print(y_pred)
[0.6517 0.6546 0.8602 0.9525 0.6206 0.7594 0.4461 0.8767 0.4899 0.9091
 0.432  0.6999 0.6504 0.8401 0.905  0.6321 0.8477 0.9277 0.6197 0.8839
 0.624  0.6407 0.6075 0.7921 0.8011 0.931  0.9394 0.7331 0.6757 0.5925
 0.6439 0.6573 0.6809 0.9414 0.7547 0.7148 0.655  0.6797 0.7562 0.73
 0.4789 0.9497 0.595  0.51   0.6656 0.9393 0.6864 0.8237 0.6649 0.742
 0.7438 0.7957 0.5085 0.6033 0.9226 0.8274 0.7199 0.9141 0.9172 0.5949
 0.705  0.8919 0.6536 0.6315 0.6823 0.7893 0.8459 0.513  0.7773 0.54
 0.6304 0.513  0.5502 0.9548 0.6373 0.6817 0.6574 0.8482 0.6992 0.4573]
```

In [27]:

```
y_test
```

Out[27]:

```
342    0.58
62     0.54
190    0.90
130    0.96
224    0.67
       ...
280    0.68
247    0.71
12     0.78
252    0.71
79     0.46
Name: Chance of Admit , Length: 80, dtype: float64
```

In [28]:

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

Out[28]:

```
0.820153259924319
```

8. Model Evaluation

In [29]:

```
from sklearn.metrics import mean_squared_error,
r2_score,mean_absolute_error,roc_auc_score,recall_score
print('model score:',rgr.score(X_test,y_test))
print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
print('roc score:',roc_auc_score(y_test>0.5, y_pred>0.5))
print('recall score:',recall_score(y_test>0.5, y_pred>0.5))
```

```
model score: 0.820153259924319
Mean Absolute Error: 0.04676374999999999
Mean Squared Error: 0.0038259696250000003
Root Mean Squared Error: 0.06185442284105479
roc score: 0.7430555555555556
recall score: 0.9861111111111112
```

9. Saving the Model.

In [30]:

```
import pickle
pickle.dump(rgr,open('model1.pkl','wb'))
```

GitHub & Project Demo Link

GITHUB: https://github.com/IBM-EPBL/IBM-Project-26103-1660014094

Project Demo Link: https://drive.google.com/drive/folders/16EJkLeBXtOqzBwmo6b38bwJKvHcO_AWw