UNIVERSITY ADMIT ELIGIBILTY PREDICTOR

PROJECT REPORT

Submitted By

A. NASHIAH JUSTIN

R. RUHI MISBAAH

J. ANUSHIYA

I. JOSHINA

TEAM ID: PNT2022TMID34570

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PONJESLY COLLEGE OF ENGINEERING, NAGERCOIL

ANNA UNIVERSITY::CHENNAI 600025

NOVEMBER 2022

TABLE OF CONTENT

ABSTRACT

This project University Admit Eligibility Predictor is created with the idea of making the complicated and tiring process of college admissions easier for students looking to apply for universities by checking what are the possibilities of them getting into their dream universities based on various factors. The students are required to fill in their personal details such as university marks, CGPA, and also additional exam scores such as GRE, IELTS, TOEFL, etc. The administrator of universities can view these details of the students enrolled. Using this software, the entrance seat allotment becomes a lot easier and can be implemented way faster than the normal procedure.

Based on the obtained data set, various models are trained and universities carrying similar properties are suggested so that it increases the chances of students getting into the universities they applied for. Classification algorithms also predict the rate of acceptance for any student to a university. This project also makes use of Machine-Learning model which predicts the eligibility of students for admission in the desired Universities.

The main advantage of this project is the computerization of the seat allotment process. The administrator has the power of allotment. They can add the allotted seats into a file and the details are saved into the system. The total time consumption for the allotment process is vastly reduced which saves time for all.

CHAPTER-1

INTRODUCTION

1.1 Purpose of Document

This is a Requirements Specification Document for our web-based project "UNIVERSITY ADMIT ELIGIBILITY PREDICTOR". It is an Technology 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.

1.2 Project Summary

Project Name: UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

Project type: Web Application

Developers: Nashiah Justin A
Ruhi Misbaah R
Anushiya J
Joshina I

Languages used: Python, HTML, Java script, CSS

Domain: Technology

Development Platform: Anaconda

Data Set Used: Admissions predictor dataset

1.3 Motivation

In the current world scenario, it is not enough for a student to just have an Under Graduate degree. Most employers now look for higher qualifications in their new recruits. As a result, the demands for a good higher education are at an all time high. A lot of students from India prefer to continue their higher education with foreign universities, especially in the United States, Canada, Germany.

In order to get admitted to these foreign universities, a set of academic requirements are needed. However, because of the sheer number of universities of different levels, students are often stuck in a dilemma till the very last minute as to whether or not their applications will be accepted or not as no concrete documentation is available which lists the requirements.

1.4 Significance

This project was developed to provide a solution to that problem. Not only do we provide a single platform that documents all the requirements as well as the different tiers of universities, but our website also incorporates an AI Model that was built after considering many leading Machine Learning Algorithms, to provide the most accurate prediction of how much of a chance of admissions does a student's current grades and other academic transcripts allow them in the tier of universities of their choice.

1.5 Project Scope

The scope of this project is a web application that allows users to enter their academic data and get predictions of their chances of admissions in the university tier of their choosing. It also provides them answers to the most common FAQ's that arise when thinking of admissions abroad for Post Graduate studies.

It also provides an analysis based on the data set used that shows how the different parameters affect chances of admissions. A Database will also be implemented for the system so that students can save their data and review and edit it as they progress with the most recent predictions being saved with their profile.

1.6 System Purpose

1.6.1 Users

Students- The people who will benefit the most from using this system are Indian students. Especially students looking to pursue their higher education from foreign universities, particularly in the United States.

Administrators- The administrator shall be able to access all the data stored in the application.

1.6.2 Locations

The system will be available to all users from any location as long as they have an Internet connection. The administrator can also access the website from any location as long as he has the correct login credentials and access to the Internet.

1.6.3 Responsibilities

The primary responsibilities of the system are:

- Provide customers access to the prediction model
- Provide answers to most common FAQs regarding PG Admissions abroad
- Provide administrator access to all records
- Provide analysis of how the various academic factors affect university admission

Other desired features of the system:

- Maintaining a profile for each user
- Password protection for each account

1.7 Limitations and Future Work

- Requires active internet connection.
- System will provide inaccurate result if data entered incorrectly.

- Future work in the project could include weighing in the features that have been ignored as of yet like percentage seats for Foreign Students. Other criterions like Co-circular achievements, Leadership positions held, job experience etc can also be included as metrics for the model.

1.8 Beneficiaries

- It helps student for making decision for choosing a right college.
- Here the chance of occurrence of error is less when compared with the existing system.
- It is fast efficient and reliable.
- Avoids data redundancy and inconsistency.
- Very user-friendly.
- Easy accessibility of data.

1.9 Overview of Document

Section 1: Functional Objectives

Each objective gives a desired behaviour for the system and a measure to determine if the final system has successfully met the objective. These objectives are organized by priority. In order for the new system to be considered successful, all high priority objectives must be met

Section 2: Non-Functional Objectives

This section is organized by category. Each objective specifies a technical requirement or constraint on the overall characteristics of the system. Each objective is measurable.

Section 3: Use Case Model

The specific behavioural requirements of the system are detailed in a series of use cases. Each use case accomplishes a business task and shows the interaction between the system and some outside actor. The system use case diagram depicts the interaction between all uses cases and system actors. A sequence diagram is also provided to show the flow of the application.

Section 4: Software and Hardware used

This section details the Hardware, Software and Online tools used for developing the UNIVERSITY ADMIT ELIGIBILITY PREDICTOR project.

Section 5: Code Snippets

This section provides some of the major code snippets used to develop the project.

Section 6: Screenshots

This section provides Screenshots of the UNIVRSITY ADMIT ELIGIBILITY PREDICTOR Application.

Section 7: Testing Report

This section details different test cases needed for the project and how they were implemented.

Section 8: Validation

This Section contains a final validation of the system where the system is checked to see if all the promised functional and non-functional requirements as mentioned in the SRS have been fulfilled or not.

CHAPTER-2

FUNCTIONAL OBJECTIVES

2.1 High Priority

- The system shall provide the user access to the AI predictor, wherein the user will be able to fill in a form with their academic transcripts data (GRE Score, TOFL Score, CGPA, SOP Score, LOR Score, Research experience), choose the tier of university they wish to apply to (1-5(top level)) and then get a prediction of their chances of admissions to that level university based on the mapping between their requirements and the student's results.

- The system shall provide the administrator access to all the records in the database on a "read-only" basis.

2.2 Medium Priority

- The system shall provide all the users with answers to the most common FAQ's like- "Distribution of University Tiers", "University Admissions Criteria"

- The system shall allow the user's details to be stored for the next time they return to the website. If the user chooses to take a new evaluation, the most recent inputs as well as prediction shall replace any previous data.

2.3 Low Priority

- The system shall provide users an analysis of how the various factors mentioned in the form affect their chances of admissions as well as what is the general trend of applications to the various tiers of universities.

2.4 Literature Survey

a) Existing Problem:

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

b) Proposed Problem:

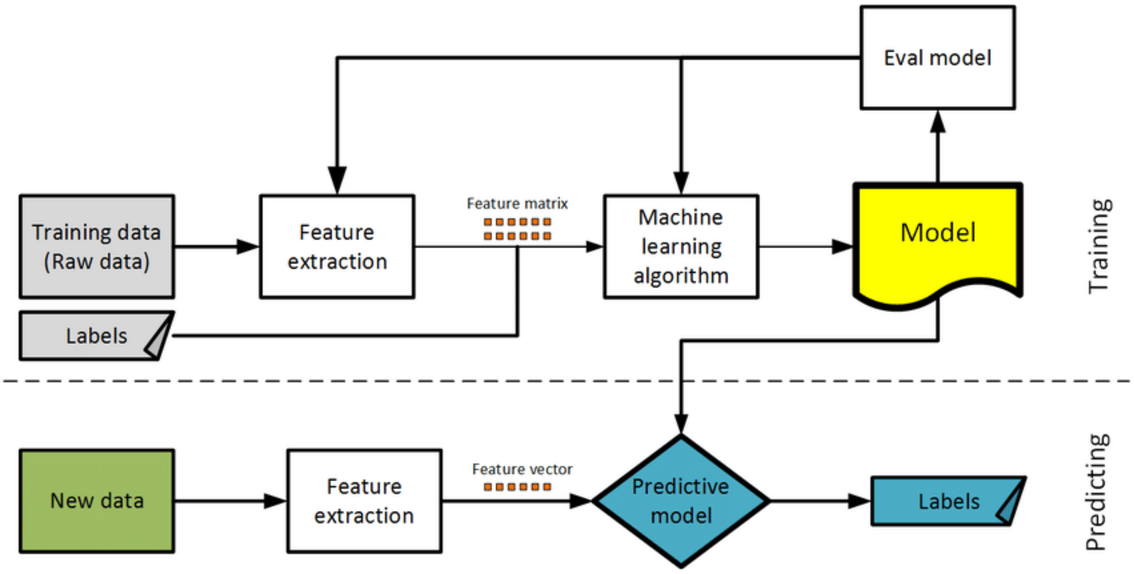
These problems can be resolved by using regression algorithms/classification algorithms as they can consider most of the features for prediction. Linear regression/KNN classification/Random can be used as the machine learning model for the model XG boost model can also be used which performs better on small to medium scale datasets but the model giving accurate and desired results only will be selected.

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

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

CHAPTER-3

ARCHITECTURE



CHAPTER-4

NON-FUNCTIONAL REQUIREMENTS

4.1 Reliability

- The system shall be completely operational all hours of the day unless system failure or up gradation work is to be performed.

- Down time after a failure shall not exceed 24 hours.

4.2 Usability

- No training is required to use the website.

- The form home, about, FAQ and analysis pages load up within 10 seconds.

- The results from the predictor should not take more than 30 seconds.

4.3 Performance

- The system can support any number of users at a time.
- The mean time to view a web page over a 56Kbps modern connection shall not exceed 5 seconds.

4.4 Security

- The system shall provide password protected access to the website to all users –students and administrations both.

4.5 Online user documentation and help

- The system shall provide a web page that explains how to navigate the site. This page should be customized based on what pages that user is allowed to access.

- This help page should be accessible from all other pages.

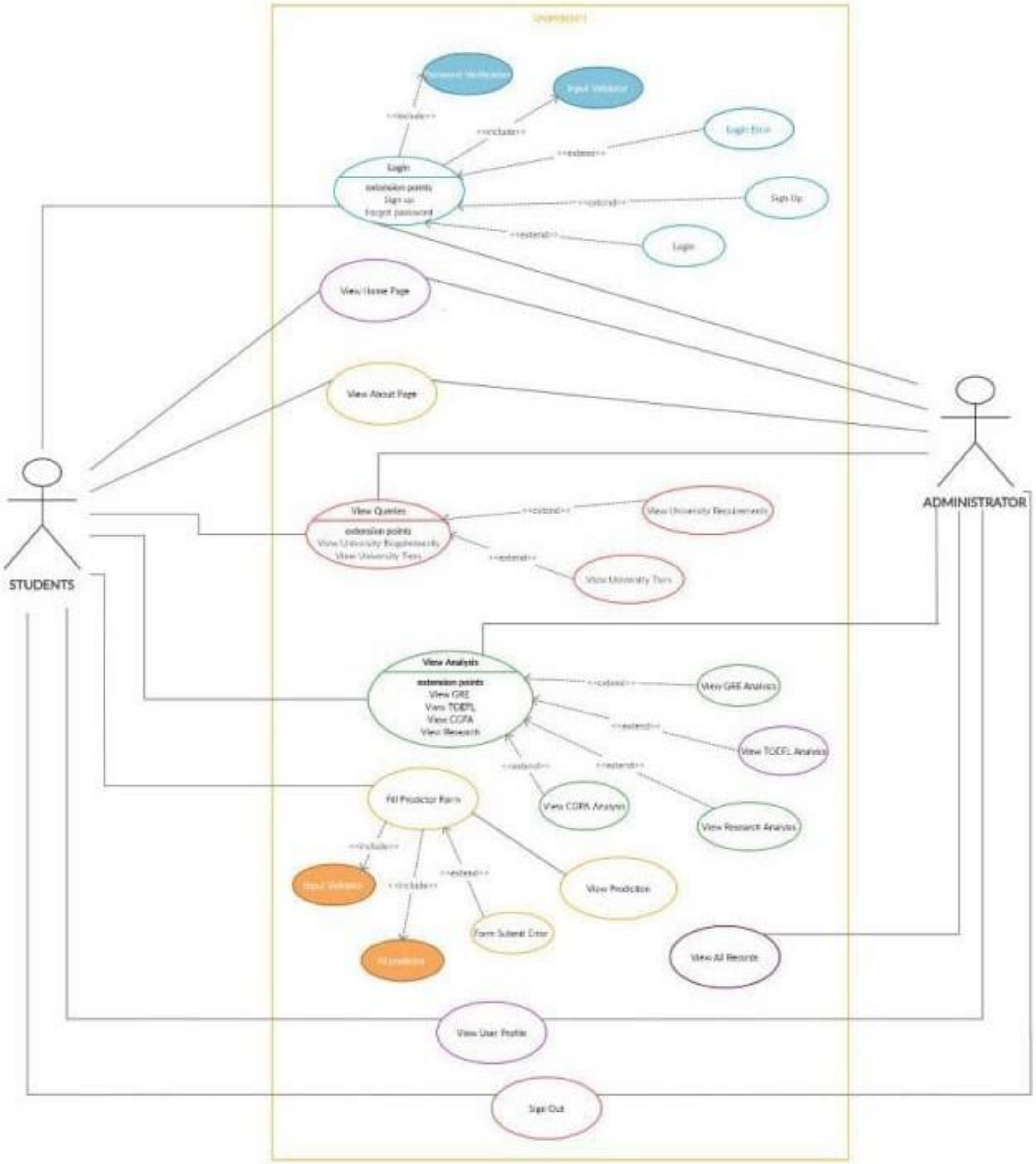
4.6 Supportability

- The system will be able to incorporate more features without major reengineering. The system web site shall be viewable from Internet Explorer 4.0 or later, Netscape Navigator/Communicator 3.0 later and the America Online web browser version 3.0 or later.

CHAPTER - V

USE CASE MODEL

5.1 Use Case Diagram



5.2 Use Case Descriptions

- ## Login User

Use Case Name:	**Login User**
Summary	In order to access the system and the prediction model, a user must sign in. It also determines whether a user is an Administrator, in which case, it offers different functionalities.
Basic Flow:	1. The use case starts when user indicate that he wants to login. 2. The system requests the username and password. 3. The user enters his username and password. 4. The system verifies the username and password against all registered users. 5. The system starts a login session and displays the home page.
Alternate Flows:	Step 1: If username is invalid, or field is left blank the use case goes back to step 2. Step 2: If the password is invalid or field is left blank the system requests that the user re-enter the password. When the user enters another password the use case continues with step 4 using the original username and new password.
Extensions points:	1. Sign Up- for new users 2. Forgot Password
Precondition:	User is registered.
Post Columns:	The user can now obtain data and perform functions according to his registered access level.

- **View Home Page**

Use Case Name:	**View Home Page**
Summary	The user can now view the home page of the app and get access to the other tabs.
Basic Flow:	1. The user case starts when the user gets logged in to the system. 2. The Home page is now visible to the user which holds all the information about how to work with the different elements of the application.
Alternate Flows:	None
Extensions Points:	Has tabs to access all other pages but no extension points as such.
Preconditions:	User must be logged in to the system.
Postconditions:	Other functionalities available to the user depend upon his/her clearance level.
Business Rules:	None

- **View About Page**

Use Case Name:	**View About Page**
Summary	The user can now view the about page of the app and get access to the other tabs.
Basic Flow:	1. The use case starts when the user clicks on the about tab on top of the Home page. 2. The About page is now visible to the user which holds all the information about how, why and by whom the application was developed.
Alternate Flows:	None
Extension Points:	Has tabs to access all other pages but no extension points as such.
Preconditions:	User must be logged in to the system.
Postconditions:	None
Business Rules:	None

- # View Queries

Use Case Name:	View Queries
Summary	The user can now view the queries the app provides answers to and get access to the other tabs.
Basic Flow:	1. The use case starts when the user clicks on the "Queries" tab. 2. The drop-down menu appears for the user to choose which query he/she wants the answer to.
Alternate Flows:	None
Extension Points:	Has tabs to access all other pages. Extensions Points are: 1. Universities Tiers page- A page that answers queries regarding how the university tiers are divided. 2. University Requirements page- It provides a table of the required mean Scores in order to get admitted to each tier of universities.
Preconditions:	User must be logged in to the system.
Postconditions:	None
Business Rules:	None

- **View Analysis**

Use Case Name:	**View Analysis**
Summary	The user can now view the analysis the app provides of how the various academic scores affect their chances of admissions.
Basic Flow:	1. The use case starts when the user clucks on the “Analysis” tab. 2. The drop-down menu appears for the user to choose which analysis he/she wants to view.
Extensions Points:	Extension point are: 1. GRE Score Analysis. 2. TOEFL Score Analysis. 3. CGPA Analysis. 4. Research Experience Analysis.
Alternate Flows:	None
Preconditions:	User must be logged in to the system.
Postconditions:	None
Business Rules:	None

• Full Predictor Form

Use Case Name:	**Full Predictor Form**
Summary	The Students can now access the admissions by filling up the form and getting their predicted chances of admissions.
Basic Flow:	1. The use case starts when the user clicks on the "predictor" tab 2. The user can access the form and fill up all the data required by the system. All data fields must be filled. 3. On clicking the submit button, the data is sent to the predictor and the results are generated. 4. The resultant prediction of chances of admissions are returned to the user.
Alternate Flows:	Step 3: If all the fields are not filled before clicking on enter, the user is prompted to fill out the remaining fields and process goes back to step 2 and continues from there.
Extension Points:	Has tabs to access all other pages No extension points as such.
Preconditions:	User must be logged to the system as a student.
Postconditions:	None
Business Rules:	This functionality is only available to a user who is registered as a student.

- ## View All Records

Use Case Name:	**View All Users**
Summary	The administrator can access all the records for all users that are registered in the system.
Basic Flows:	1. The use case starts when the users is verified to be an administrator. 2. Once the admin clicks on the view all tab, all the records currently stored in the system become available to him/her.
Alternate Flows:	None
Extension Points:	Has tabs to access all other pages. No Extensions Points as such.
Preconditions:	User must be logged in to the system and verified as an admin.
Postconditions:	None
Business Rules:	This functionality is only available to users who are verified Administrators of the system.

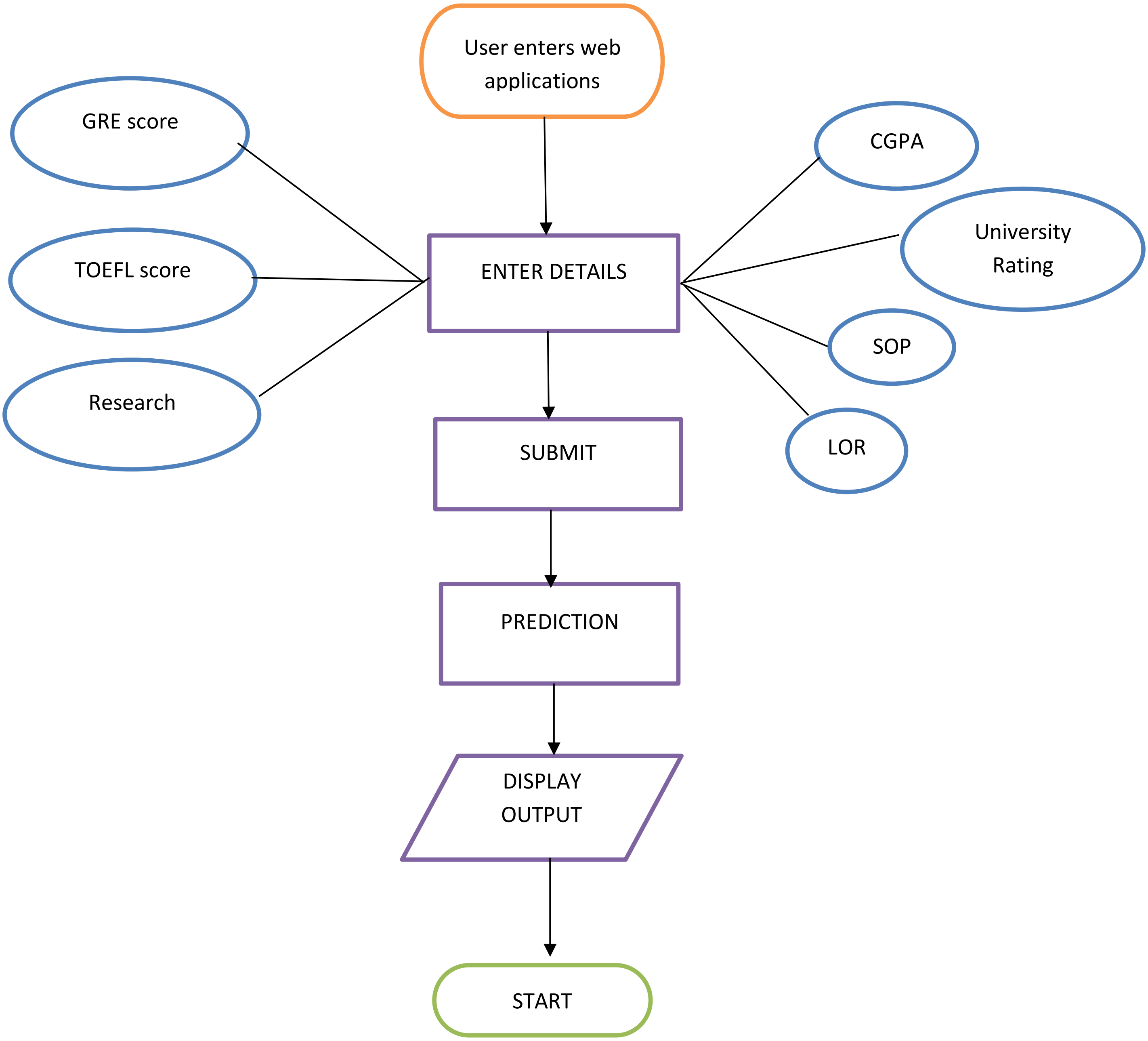
- ## View User Profile

Use Case Name	**View User Profile**
Summary	The user can view his/her own profile. Students can also see the most recent values of their scores and prediction as stored in the system.
Basic Flow:	1. The use case starts when the user clicks on the “Queries” tab. 2. The drop-down menu appears for the user to choose which query he/she wants the answer to.
Alternate Flows:	None
Extensions Points:	Has tabs to access all other pages. No Extension points as such.
Preconditions:	User must be logged in the system.
Postcondition:	None
Business Rules:	None

- **Sign Out**

Use Case Name:	**Sign Out**
Summary	The user can sign out of the system and be redirected to the login age.
Basic Flow:	1. The use case starts when the user clicks on the “Logout” tab. 2. The user session is ended, and he/she gets redirected to the login page.
Alternate Flows:	None
Extensions Points:	None
Precondition:	User must be logged in to the system.
Postconditions:	None
Business Rules:	Some data and functions are restricted to certain types of users or users with a particular access level.

5.3 FLOWCHART



CHAPTER - 6

SOFTWARE AND HARDWARE PLATFORM

6.1 Hardware

A home PC -capable of handling light ML processing.

Device Specifications:

1. I5 10th Gen processor
2. 8 GB RAM
3. 64bit Operation System

6.2 Software

- **Anaconda (Jupiter Notebook)**

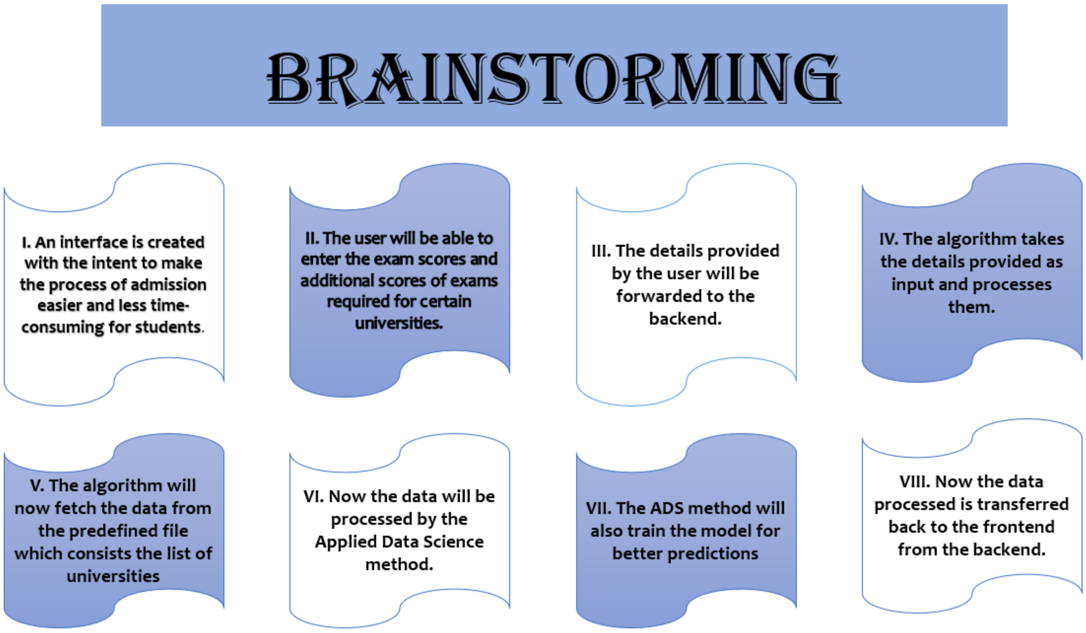
Project Jupiter is a non-profit organization created to “develop open-source software, open-standards, and services for interactive computing across dozens of programming languages”. [2] Spun off from I Python in 2014 by Fernando Perez, Project Jupiter supports execution environments in several dozen languages. Project Jupiter’s name is a reference to the three core programming languages supported by Jupiter, which are Julia, Python and R, and also a homage to Galileo’s notebooks recording the discovery of the moons of Jupiter.

CHAPTER-7

PROJECT DELIVERABLES

7.1 IDEATION PHASE

7.1.1 BRAINSTORMING



BRAINSTORMING

I. An interface is created with the intent to make the process of admission easier and less time-consuming for students.

II. The user will be able to enter the exam scores and additional scores of exams required for certain universities.

III. The details provided by the user will be forwarded to the backend.

IV. The algorithm takes the details provided as input and processes them.

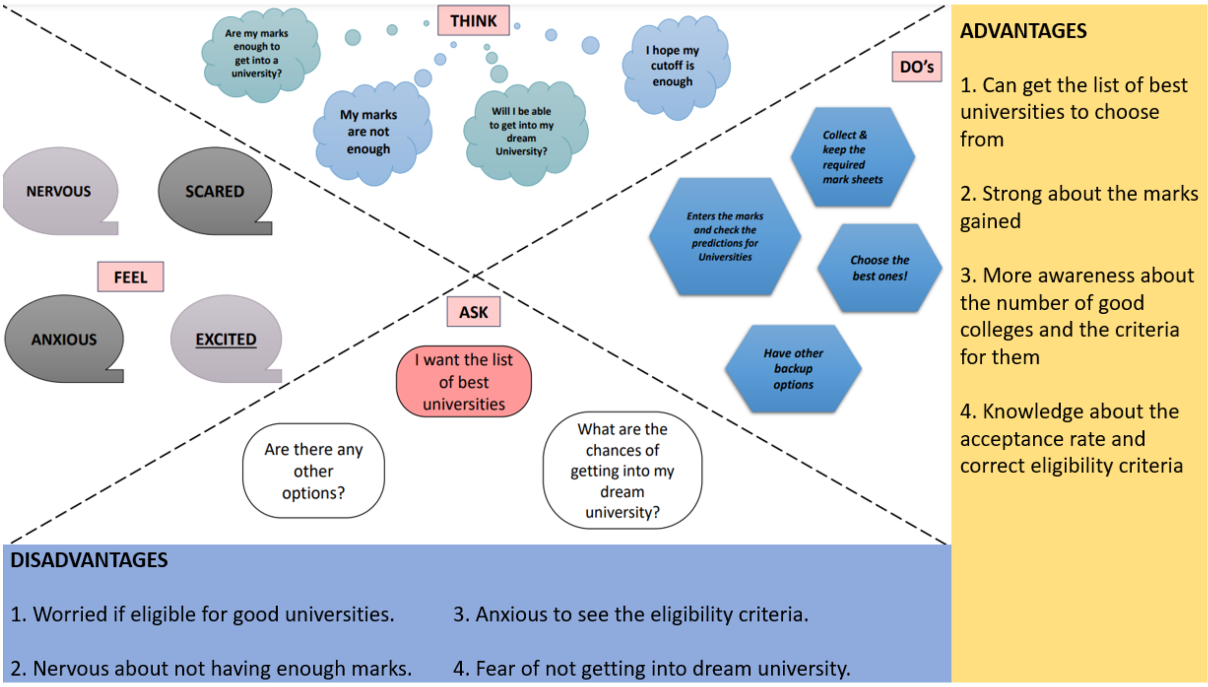
V. The algorithm will now fetch the data from the predefined file which consists the list of universities

VI. Now the data will be processed by the Applied Data Science method.

VII. The ADS method will also train the model for better predictions

VIII. Now the data processed is transferred back to the frontend from the backend.

7.1.2 EMPATHY MAP


THINK
Are my marks enough to get into a university?
My marks are not enough
Will I be able to get into my dream University?
I hope my cutoff is enough
DO's
Collect & keep the required mark sheets
Enters the marks and check the predictions for Universities
Choose the best ones!
Have other backup options
FEEL
NERVOUS
SCARED
ANXIOUS
EXCITED
ASK
I want the list of best universities
Are there any other options?
What are the chances of getting into my dream university?

ADVANTAGES

1. Can get the list of best universities to choose from

2. Strong about the marks gained

3. More awareness about the number of good colleges and the criteria for them

4. Knowledge about the acceptance rate and correct eligibility criteria

DISADVANTAGES

1. Worried if eligible for good universities.

2. Nervous about not having enough marks.

3. Anxious to see the eligibility criteria.

4. Fear of not getting into dream university.

7.1.3 LITERATURE SURVEY

EXISTING PROBLEM:

Previously, much research has been done in this field and they use Naïve-Bayes algorithm. Here, Naïve-Bayes algorithm is used for calculating the possibility of successfully getting into the university the user wishes. But here the major drawback of this is that they didn't consider the various other factors which will increase the chances of getting admission to a university like GRE scores, IELTS/TOEFL scores, etc. Bayesian Networks Algorithms are used to create a network for evaluating the eligibility criteria based on data submitted by foreign students. Thus, it helps aspiring students to have a better vision of eligibility criteria for specific colleges by comparing the data already submitted by various students who successfully got into the university. But this model only shows the comparisons of data from students who successfully got admission but not data from the students who got rejected. Hence, this method will is not very accurate and will not gove a clear vision to students.

PROPOSED SOLUTION:

These various drawbacks can be solved by using classification and regression algorithms as they have emerged as the best for prediction features. Linear Regression/ KNN Classification / Random Forest Regressor can be used as the Machine Learning model for this model. XG Boost model also can be used as it performs best for small to medium-sized datasets. This will address the problems and limitations of the existing model which is the aim of this proposed solution.

The defects from the past model have been gathered as the requirements for this system based on the user feedback.

Following are the objectives of Proposed System: ¬

- ✓ Less time-consuming.
- ✓ Efficiency in operating and administering.
- ✓ Eco- Friendly and paperless admission.
- ✓ Usage of less manpower.
- ✓ Access to students all around the world.

7.1.4 PROBLEM STATEMENT

- ❖ The problem statement for this project is to design a userfriendly college or university predictor and to provide probabilistic insight into the whole admission process, the cut-offs required for respective colleges, and eligibility criteria according to the preferences of the students

- ❖ It has always been a troublesome and tiring process for students to search for a good university/college that provides their preferred course or to choose a course from the options available options considering the futuristic career options.

- ❖ At times when the students are not sure about what field to get into, it is very hard and difficult for them to find universities based on their scores.

- ❖ The main idea of this project is to lessen their burden by providing a list of good colleges or universities with their marks scored and additional performances as input.

AIM:

- ➢ To help students in shortlisting comparing with their profile

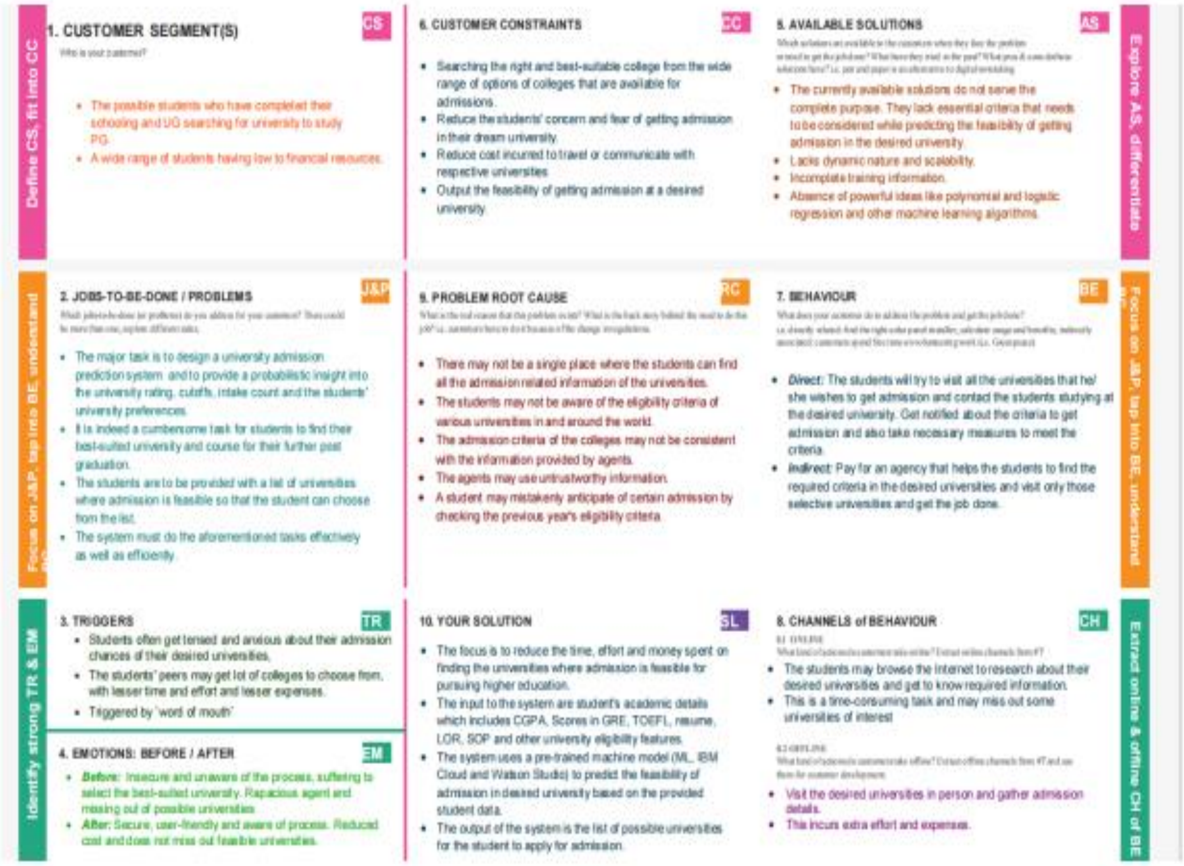
- ➢ The predicted output will give students a clear vision and guide them throughout their admission process

- ➢ This analysis also gives a clear idea about the eligibility criteria for the students who are or will be preparing.

7.2 PROJECT DESIGN PHASE-I

7.2.1 PROBLEM FIT

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR


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

1. CUSTOMER SEGMENT(S)

Who is your customer?

- The possible students who have completed their schooling and UG searching for university to study PG.
- A wide range of students having low to financial resources.

6. CUSTOMER CONSTRAINTS

- Searching the right and best-suitable college from the wide range of options of colleges that are available for admissions.
- Reduce the students' concern and fear of getting admission in their dream university.
- Reduce cost incurred to travel or communicate with respective universities
- Output the feasibility of getting admission at a desired university

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

- The currently available solutions do not serve the complete purpose. They lack essential criteria that needs to be considered while predicting the feasibility of getting admission in the desired university.
- Lacks dynamic nature and scalability.
- Incomplete training information.
- Absence of powerful ideas like polynomial and logistic regression and other machine learning algorithms.

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.

- The major task is to design a university admission prediction system and to provide a probabilistic insight into the university rating, cutoffs, intake count and the students' university preferences.
- It is indeed a cumbersome task for students to find their best-suited university and course for their further post graduation.
- The students are to be provided with a list of universities where admission is feasible so that the student can choose from the list.
- The system must do the aforementioned tasks effectively as well as efficiently.

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.

- There may not be a single place where the students can find all the admission related information of the universities.
- The students may not be aware of the eligibility criteria of various universities in and around the world.
- The admission criteria of the colleges may not be consistent with the information provided by agents.
- The agents may use untrustworthy information.
- A student may mistakenly anticipate of certain admission by checking the previous year's eligibility criteria.

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)

- *Direct:* The students will try to visit all the universities that he/ she wishes to get admission and contact the students studying at the desired university. Get notified about the criteria to get admission and also take necessary measures to meet the criteria.
- *Indirect:* Pay for an agency that helps the students to find the required criteria in the desired universities and visit only those selective universities and get the job done.

3. TRIGGERS

- Students often get tensed and anxious about their admission chances of their desired universities.
- The students' peers may get lot of colleges to choose from, with lesser time and effort and lesser expenses.
- Triggered by 'word of mouth'

4. EMOTIONS: BEFORE / AFTER

- ***Before:*** Insecure and unaware of the process, suffering to select the best-suited university. Rapacious agent and missing out of possible universities.
- ***After:*** Secure, user-friendly and aware of process. Reduced cost and does not miss out feasible universities.

10. YOUR SOLUTION

- The focus is to reduce the time, effort and money spent on finding the universities where admission is feasible for pursuing higher education.
- The input to the system are student's academic details which includes CGPA, Scores in GRE, TOEFL, resume, LOR, SOP and other university eligibility features.
- The system uses a pre-trained machine model (ML, IBM Cloud and Watson Studio) to predict the feasibility of admission in desired university based on the provided student data.
- The output of the system is the list of possible universities for the student to apply for admission.

8. CHANNELS of BEHAVIOUR

8.1 ONLINE

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

- The students may browse the internet to research about their desired universities and get to know required information.
- This is a time-consuming task and may miss out some universities of interest

8.2 OFFLINE

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

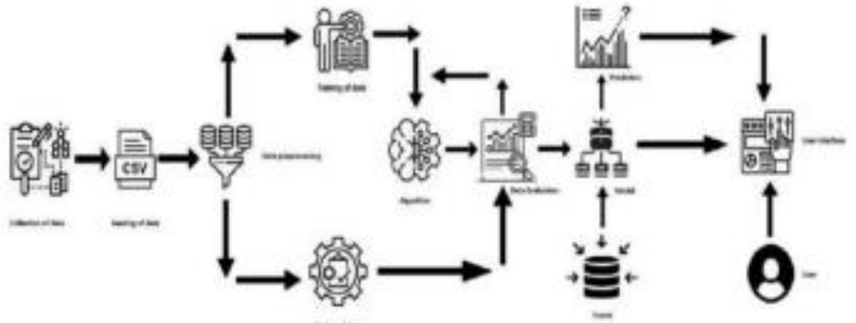
- Visit the desired universities in person and gather admission details.
- This incurs extra effort and expenses.

7.2.2 PROPOSED SLOUTION

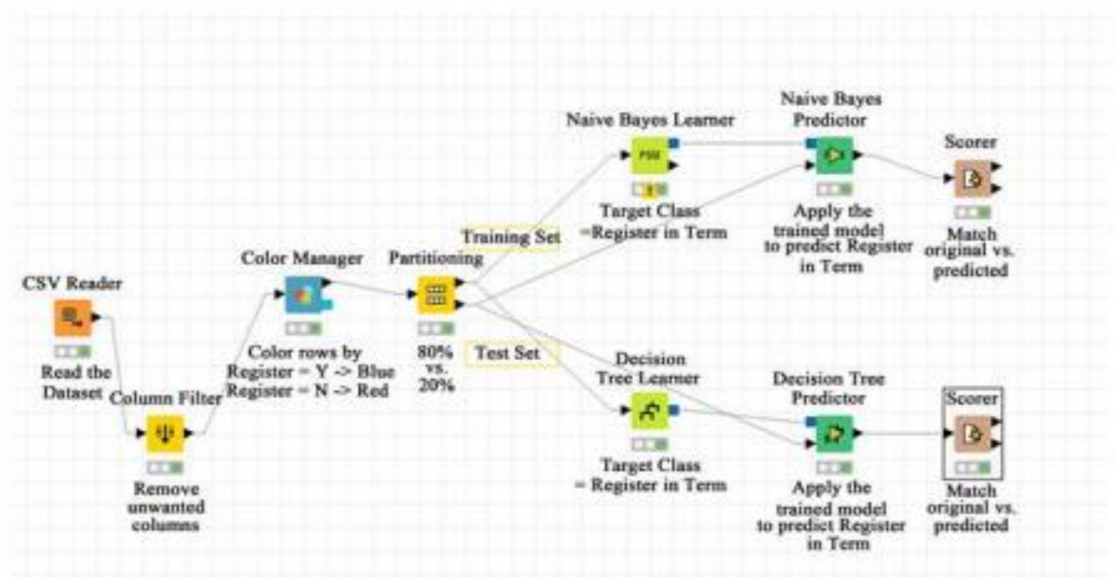
S. No	PARAMETER	DESCRIPTION
1.	Problem Statement	Choosing the right universities or colleges is definitely a Student has to face. Many students apply for universities in which they have little chance of acceptance. This leads students of poor economic backgrounds to frustration and anxiety as they only lose a surplus amount of money just for applying to those universities.
2.	Solution Description	University and College research being one part of the university application process is itself an arduous and lengthy task. This issue is a big problem for students have not been solved till now. There are recognized sites that filter the best universities and colleges based on location, tuition fees, major, and degree but none of them have used machine learning algorithms to solve the issue. Hence, we have done this research project to solve that issue to some extent with the use of data mining techniques.
3.	Uniqueness	The university application process itself is a tedious task Students need lots of endeavor and determination for completing the overall application process. It would definitely be easier for students if they get relief from the step of selecting the best-suited universities and colleges for application.
4.	Social Impact	Results of this project are not applicable to college graduates of each and every major. As there was limitation of information on dataset this system could not predict and recommend universities to students of every major. Nevertheless, the statistical data mining techniques used in this project can be applicable to all majors. If any universities have insufficient data on the major chosen by the student it will return insufficient data for prediction to the user.

5.	Revenue Model	From this project, financially can earn from the students admission fees but while they want to first select in their selected college in prediction. Although which is done by this project for prediction. In this project, this problem has been addressed by modeling a recommender system based on various classification algorithms. The required data was obtained from thegradcafe.com. Based on this data set, various models were trained and one best and some other similar properties carrying universities are suggested for the students such that it maximizes the chances of a student getting an admit from that university list.
6.	Scalability of the Solution	In this project, this problem has been addressed by modeling a recommender system based on various classification algorithms. To predict the best University for a particular student his/her GPA score, GRE (Verbal and Quant) Score, and TOEFL score has been used as attributes for classification. K nearest neighbor has been used to predict the best universities and K means clustering has been used to find more similar universities. Support Vector Machine and Random forest has been used to predict the admission chance of a particular student on a specific University

7.2.3 SOLUTION ARCHITECTURE



(a)


CSV Reader
Read the Dataset
Column Filter
Remove unwanted columns
Color Manager
Color rows by Register = Y -> Blue Register = N -> Red
Partitioning
80% vs. 20%
Training Set
Test Set
Naive Bayes Learner
PMM
Target Class =Register in Term
Naive Bayes Predictor
Apply the trained model to predict Register in Term
Scorer
Match original vs. predicted
Decision Tree Learner
Target Class = Register in Term
Decision Tree Predictor
Apply the trained model to predict Register in Term
Scorer
Match original vs. predicted

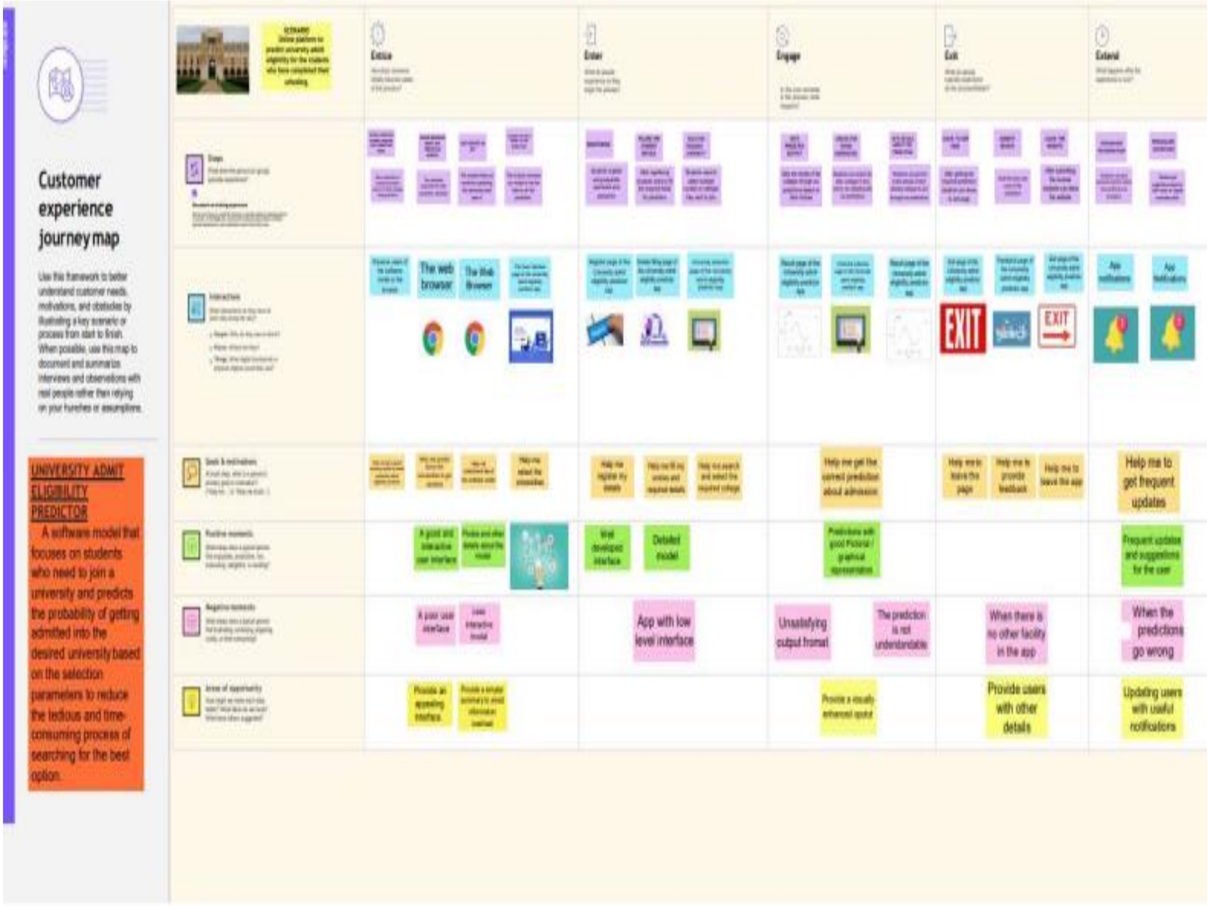
(b)

7.3 PROJECT DESIGN PHASE-II

7.3.1 CUSTOMER JOURNEY

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

CUSTOMER JOURNEY



Customer experience journey map

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.

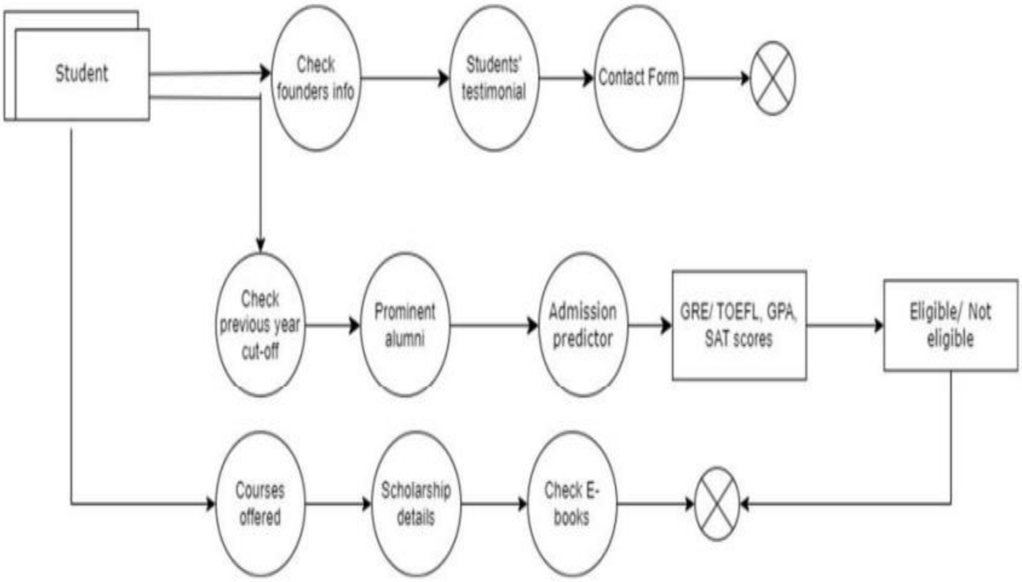
UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

A software model that focuses on students who need to join a university and predicts the probability of getting admitted into the desired university based on the selection parameters to reduce the tedious and time-consuming process of searching for the best option.

Entice
Enter
Engage
Exit
Extend
The web browser
The Web Browser
EXIT
EXIT
Help me to get frequent updates
Help me to leave the app
A good and interactive user interface
Well developed interface
Detailed model
Frequent updates and suggestions for the user
A poor user interface
App with low level interface
Unsatisfying output fromat
The prediction is not understandable
When there is no other facility in the app
When the predictions go wrong
Provide an appealing interface
Provide users with other details
Updating users with useful notifications

7.3.2 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.



USER STORIES

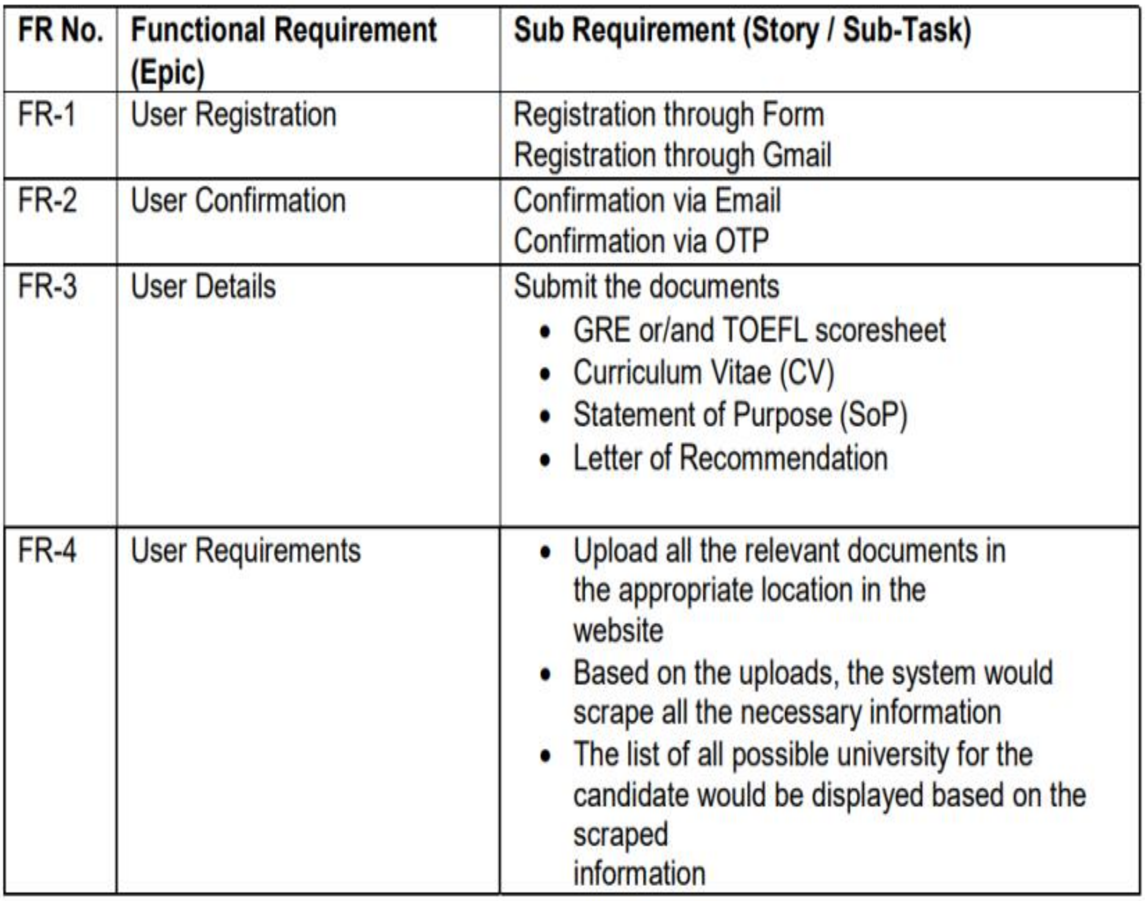
User Types	**Functional Requirement (Epic)**	**User Story Number**	**User Story/ Task**	**Acceptance Criteria**	**Priority**	**Release**
Customer	Landing page	USN-1	As a user, I can view the details about the university	I can access the university landing page	Medium	Sprint-1
		USN-2	As a user, I can view the latest news about the university	I can access the latest news	Medium	Sprint-1
		USN-3	As a user, I can fill the contact form for queries	I can fill and submit the contact form	Low	Sprint-2
		USN-4	As a user, I can see the social media profiles of the university	I can reach out to them via social media	Medium	Sprint-1
		USN-5	As a user, I can see testimonials of students who graduated from the university	I can access the testimonials	Medium	Sprint-1

	Admissions	USN-6	As a user, I can see the previous year cut-off marks	I can download the previous year cut-off details	High	Sprint-2
		USN-7	As a user, I can read about proud alumni of the university	I can access the details of alumni of the university	Medium	Sprint-2
		USN-8	As a user, I can predict my eligibility for admission at the university	I can get result as either eligible/not eligible	High	Sprint-2
	Courses offered	USN-9	As a user, I can see the courses offered by the university for PG students	I can access the courses details	Medium	Sprint-3
	Events	USN-10	As a user, I can check various technical events about to happen in the university	I can register for the events	Low	Sprint-3

7.3.3 SOLUTIONS REQUIREMENTS

7.3.3.1 FUNCTIONAL REQUIREMENTS:

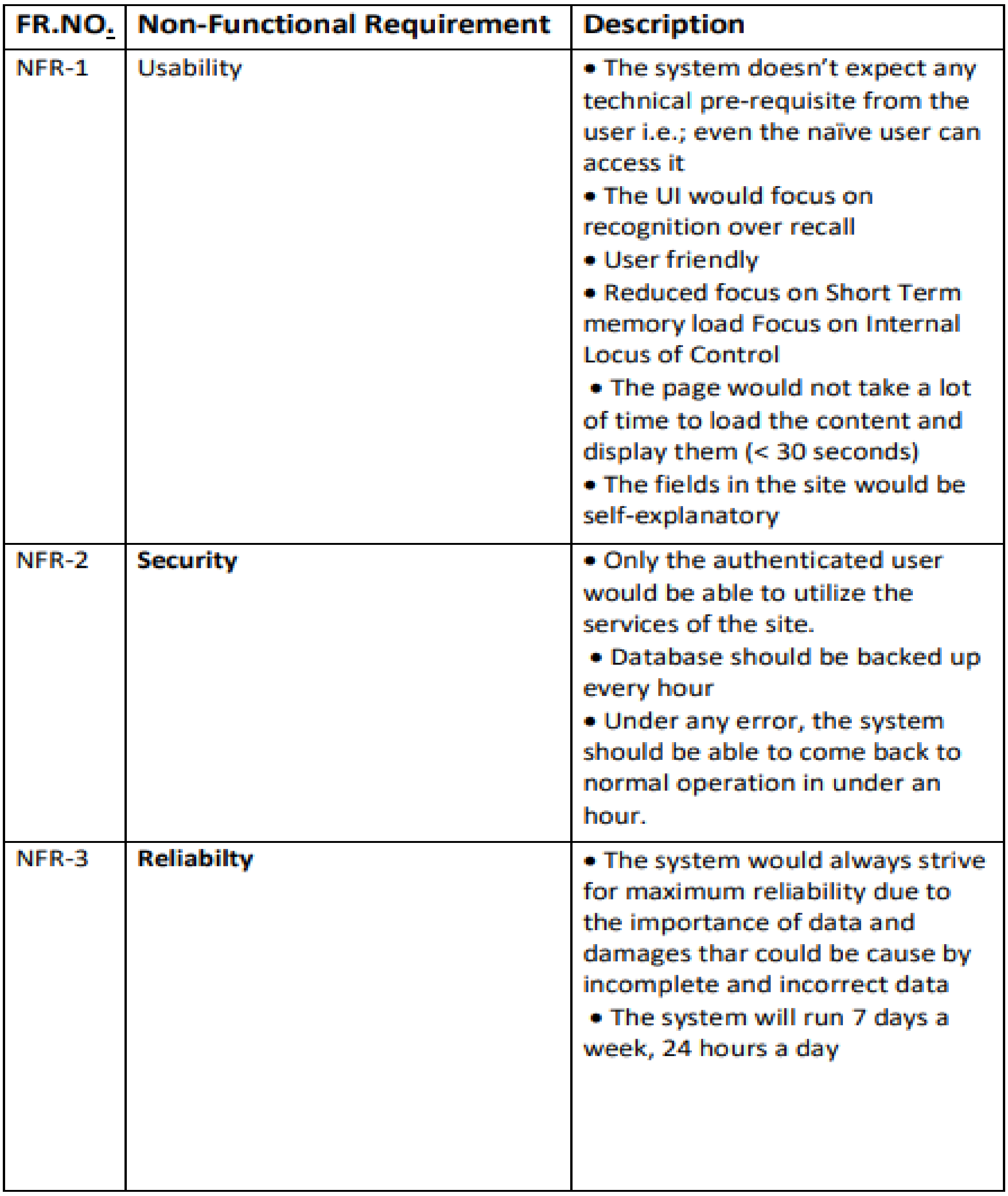
Following are the functional requirements of the proposed solution.



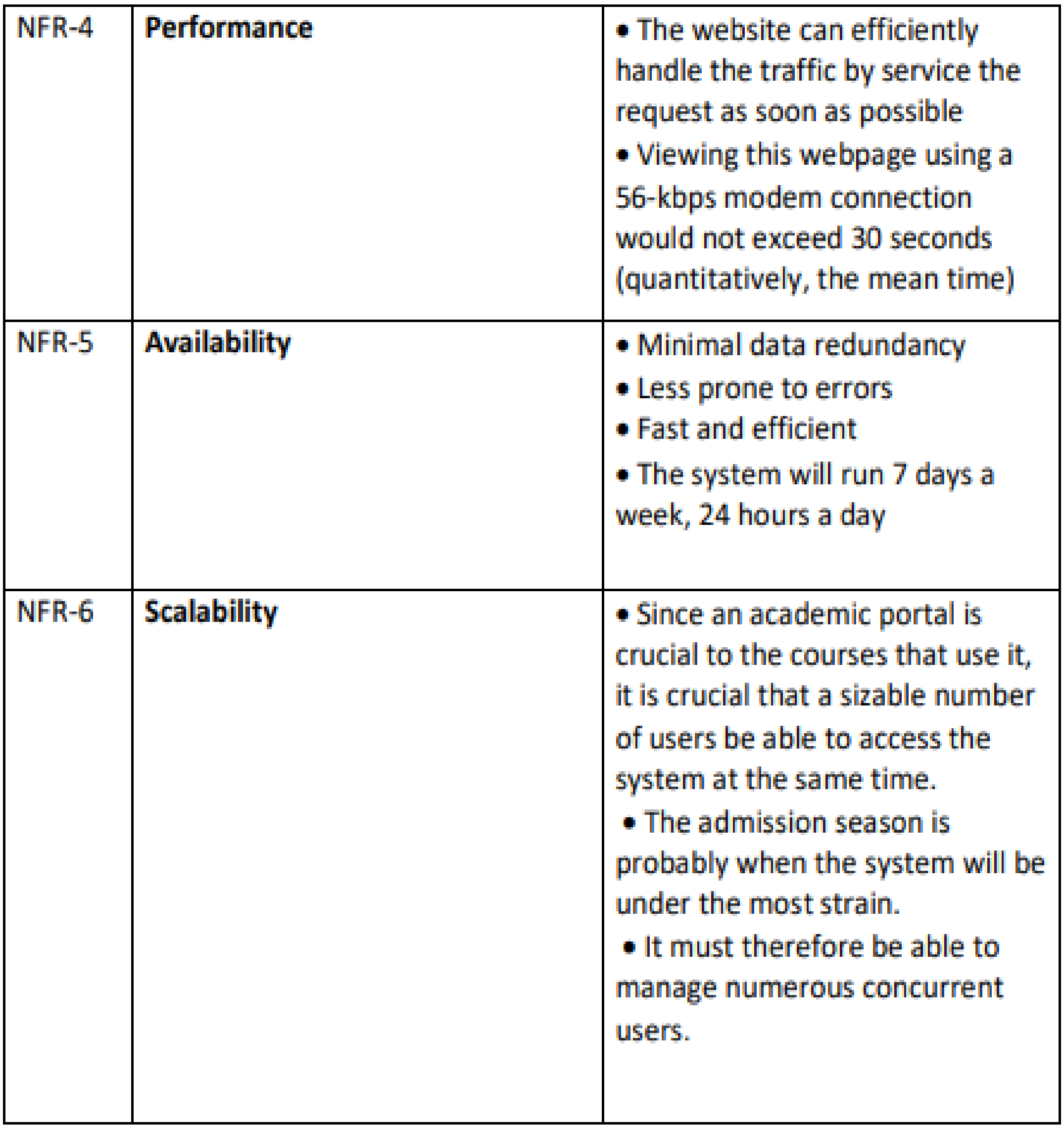
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Details	Submit the documents • GRE or/and TOEFL scoresheet • Curriculum Vitae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

7.3.3.2 NON-FUNCTIONAL REQUIREMENTS:

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



FR.NO.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	• The system doesn't expect any technical pre-requisite from the user i.e.; even the naïve user can access it • The UI would focus on recognition over recall • User friendly • Reduced focus on Short Term memory load Focus on Internal Locus of Control • The page would not take a lot of time to load the content and display them (< 30 seconds) • The fields in the site would be self-explanatory
NFR-2	**Security**	• Only the authenticated user would be able to utilize the services of the site. • Database should be backed up every hour • Under any error, the system should be able to come back to normal operation in under an hour.
NFR-3	**Reliabilty**	• The system would always strive for maximum reliability due to the importance of data and damages thar could be cause by incomplete and incorrect data • The system will run 7 days a week, 24 hours a day



NFR-4	**Performance**	• The website can efficiently handle the traffic by service the request as soon as possible • Viewing this webpage using a 56-kbps modem connection would not exceed 30 seconds (quantitatively, the mean time)
NFR-5	**Availability**	• Minimal data redundancy • Less prone to errors • Fast and efficient • The system will run 7 days a week, 24 hours a day
NFR-6	**Scalability**	• Since an academic portal is crucial to the courses that use it, it is crucial that a sizable number of users be able to access the system at the same time. • The admission season is probably when the system will be under the most strain. • It must therefore be able to manage numerous concurrent users.

7.3.4 TECHNOLOGY ARCHITECTURE

TECHNICAL ARCHITECTURE:

The Deliverable shall include the architectural diagram below and the information as per the table1 & table2.

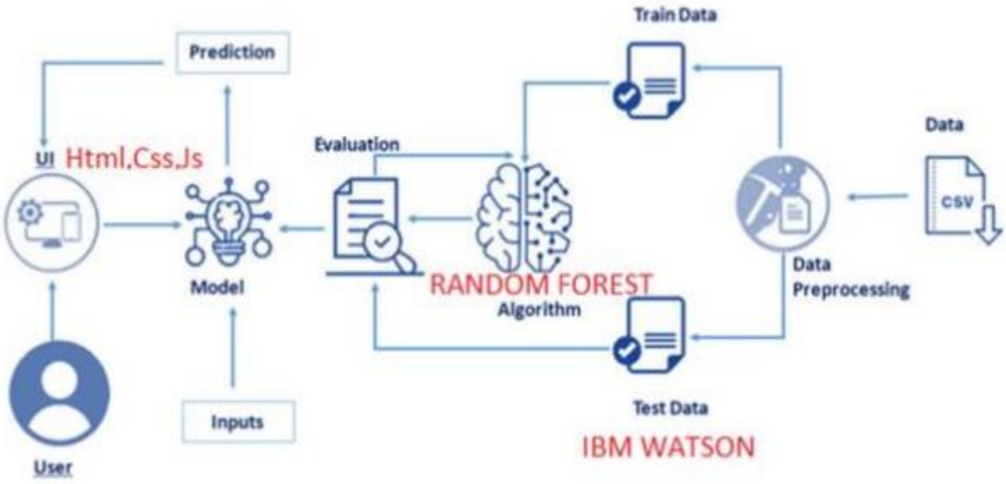


Table-1: Components & Technologies:

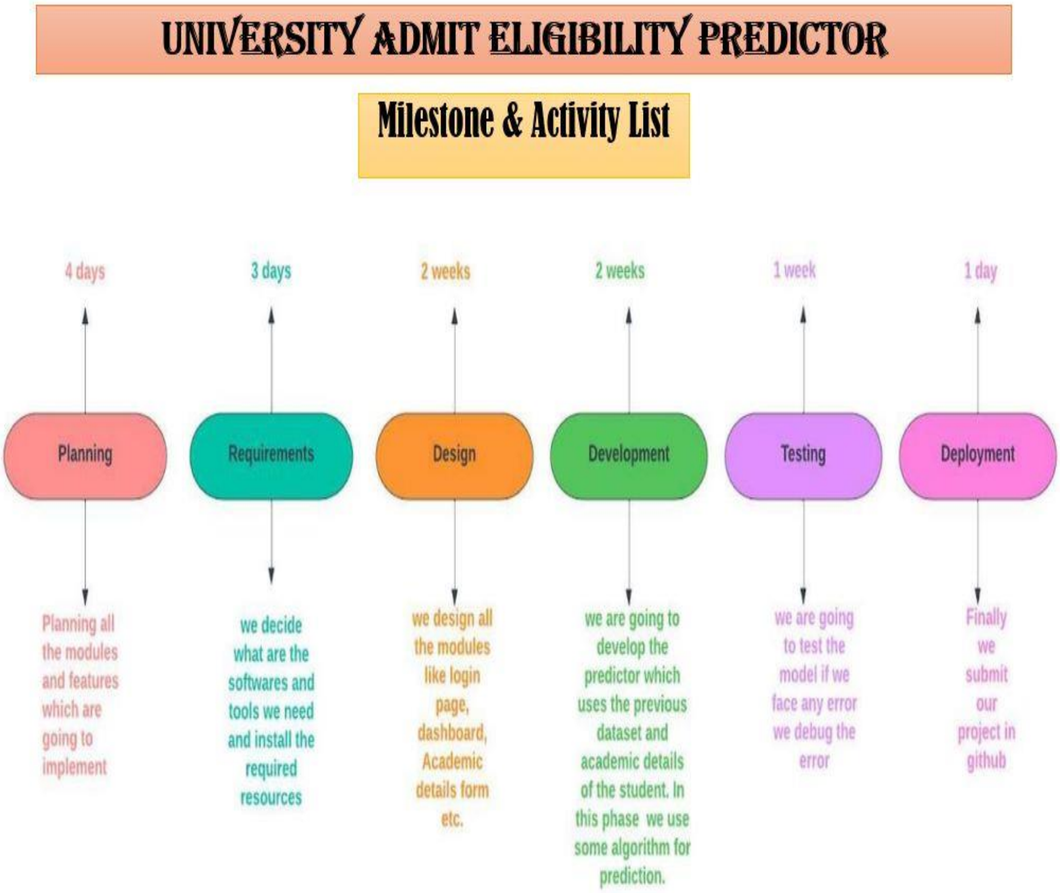
S. No	**Component**	**Technology**
1.	User Interfae	Html,css,js
2.	Application Logic-1	Python
3.	Application Logic-2	IBM Watson
4.	Machine Learning Model	Random forsest

Table-2: Application Characteristics:

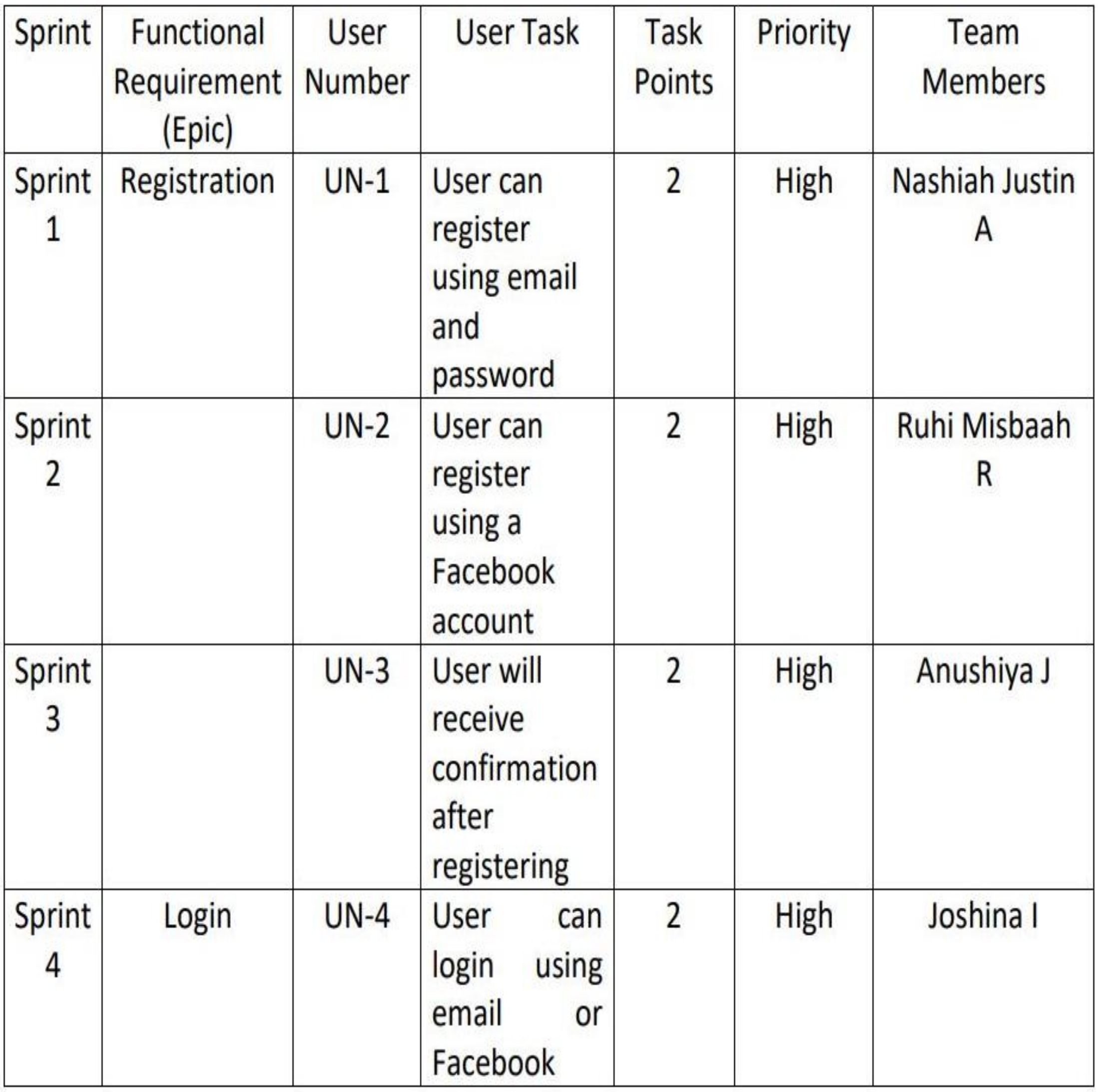
S. NO	**Component**	**Technology**
1.	Open-Source Frameworks	Flask
2.	Performance	It can handle about 100 requests per second

7.4 PROJECT PLANNING PHASE

7.4.1 MILESTONE & ACTIVITY LIST:

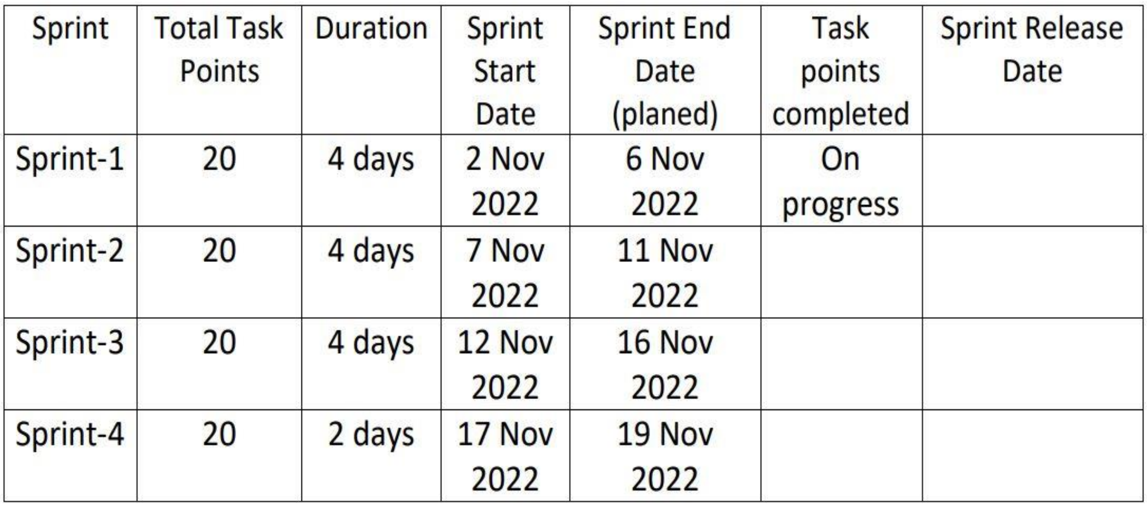

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR
Milestone & Activity List
4 days
3 days
2 weeks
2 weeks
1 week
1 day
Planning
Requirements
Design
Development
Testing
Deployment
Planning all the modules and features which are going to implement
we decide what are the softwares and tools we need and install the required resources
we design all the modules like login page, dashboard, Academic details form etc.
we are going to develop the predictor which uses the previous dataset and academic details of the student. In this phase we use some algorithm for prediction.
we are going to test the model if we face any error we debug the error
Finally we submit our project in github

7.4.2 SPRINT DELIVERY PLAN:



Sprint	Functional Requirement (Epic)	User Number	User Task	Task Points	Priority	Team Members
Sprint 1	Registration	UN-1	User can register using email and password	2	High	Nashiah Justin A
Sprint 2		UN-2	User can register using a Facebook account	2	High	Ruhi Misbaah R
Sprint 3		UN-3	User will receive confirmation after registering	2	High	Anushiya J
Sprint 4	Login	UN-4	User can login using email or Facebook	2	High	Joshina I

Project Tracker, Velocity & Burndown Chart



Sprint	Total Task Points	Duration	Sprint Start Date	Sprint End Date (planed)	Task points completed	Sprint Release Date
Sprint-1	20	4 days	2 Nov 2022	6 Nov 2022	On progress	
Sprint-2	20	4 days	7 Nov 2022	11 Nov 2022		
Sprint-3	20	4 days	12 Nov 2022	16 Nov 2022		
Sprint-4	20	2 days	17 Nov 2022	19 Nov 2022		

Velocity:

Imagine we have a 4-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 per day)

$$AV = \frac{\textbf{Sprint Duration}}{\textbf{velocity}} = \textbf{20 \% 4} = \textbf{5}$$

Burndown Chart:

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

CHAPTER-8

IMPLEMENTATION

```
import numpy as np
import pandas as pd
#import os
from matplotlib import pyplot as plt
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
import seaborn as sns
sns.set(style='white')
sns.set(style='whitegrid', color_codes=True)
df = pd.read_csv("Admission_Predict_Ver1.1.csv")
df.rename(columns = {'Chance of Admit ':'Chance of Admit', 'LOR ':'LOR'},
inplace=True)
df.drop(labels='Serial No.', axis=1, inplace=True)
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(df.corr(), annot=True, cmap='Blues')
plt.figure(figsize=(20,6)) plt.subplot(1,2,1)
sns.distplot(df['CGPA'])
plt.title('CGPA Distribution of Applicants')
plt.subplot(1,2,2)
```

```
sns.regplot(df['CGPA'], df['Chance of Admit'])
plt.title('CGPA vs Chance of Admit')
plt.figure(figsize=(20,6))
plt.subplot(1,2,1)
sns.distplot(df['GRE Score'])
plt.title('Distributed GRE Scores of Applicants')
plt.subplot(1,2,2)
sns.regplot(df['GRE Score'], df['Chance of Admit'])
plt.title('GRE Scores vs Chance of Admit')
plt.figure(figsize=(20,6))
plt.subplot(1,2,1)
sns.distplot(df['TOEFL Score'])
plt.title('Distributed TOEFL Scores of Applicants')
plt.subplot(1,2,2)
sns.regplot(df['TOEFL Score'], df['Chance of Admit'])
plt.title('TOEFL Scores vs Chance of Admit')
fig, ax = plt.subplots(figsize=(8,6))
sns.countplot(df['Research'])
plt.title('Research Experience')
plt.ylabel('Number of Applicants')
ax.set_xticklabels(['No Research Experience', 'Has Research Experience'])
fig, ax = plt.subplots(figsize=(8,6))
sns.countplot(df['University Rating'])
plt.title('University Rating')
plt.ylabel('Number of Applicants')
targets = df['Chance of Admit']
```

```
features = df.drop(columns = {'Chance of Admit'})
X_train, X_test, y_train, y_test = train_test_split(features, targets, test_size=0.2, random_state=42)
linreg = LinearRegression()
linreg.fit(X_train, y_train)
y_predict = linreg.predict(X_test)
linreg_score = (linreg.score(X_test, y_test))*100
linreg_score
@app.route('/predictor', methods =['GET','POST'])
def predictor():
form = PredictorForm()
if form.is_submitted():
#form inputs
Record_dictionary2=request.form.to_dict()
del Record_dictionary2['csrf_token']
del Record_dictionary2['submit']
gre=float(request.form['gre'])
toefl=float(request.form['toefl'])
#rating=float(request.form['uni'])
sop=float(request.form['sop'])
lor=float(request.form['lor'])
cgpa=float(request.form['cgpa'])
research=float(request.form['research'])
uni=float(request.form['uni'])
#global personId
#Record_dictionary2["person_id"]=personId['_id']
```

```
Record_dictionary2["username"]=session['user']
#--------------------------------------prediction model-----------------
import numpy as np
import pandas as pd
#from matplotlib import pyplot as plt
import sklearn
import sklearn.preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
#from sklearn.ensemble import RandomForestRegressor
#import seaborn as sns
import pickle
#sns.set(style='white')
#sns.set(style='whitegrid', color_codes=True)
df = pd.read_csv("static\Admission_Predict_Ver1.1.csv")
df.rename(columns = {'Chance of Admit ':'Chance of Admit', 'LOR ':'LOR'}, inplace=True)
df.drop(labels='Serial No.', axis=1, inplace=True)
targets = df['Chance of Admit']
features = df.drop(columns = {'Chance of Admit'})
X_train, X_test, y_train, y_test = train_test_split(features, targets, test_size=0.2, random_state=42)
test=[[gre,toefl,uni,lor,sop,cgpa,research]]
X_test=np.vstack((X_test,test))
```

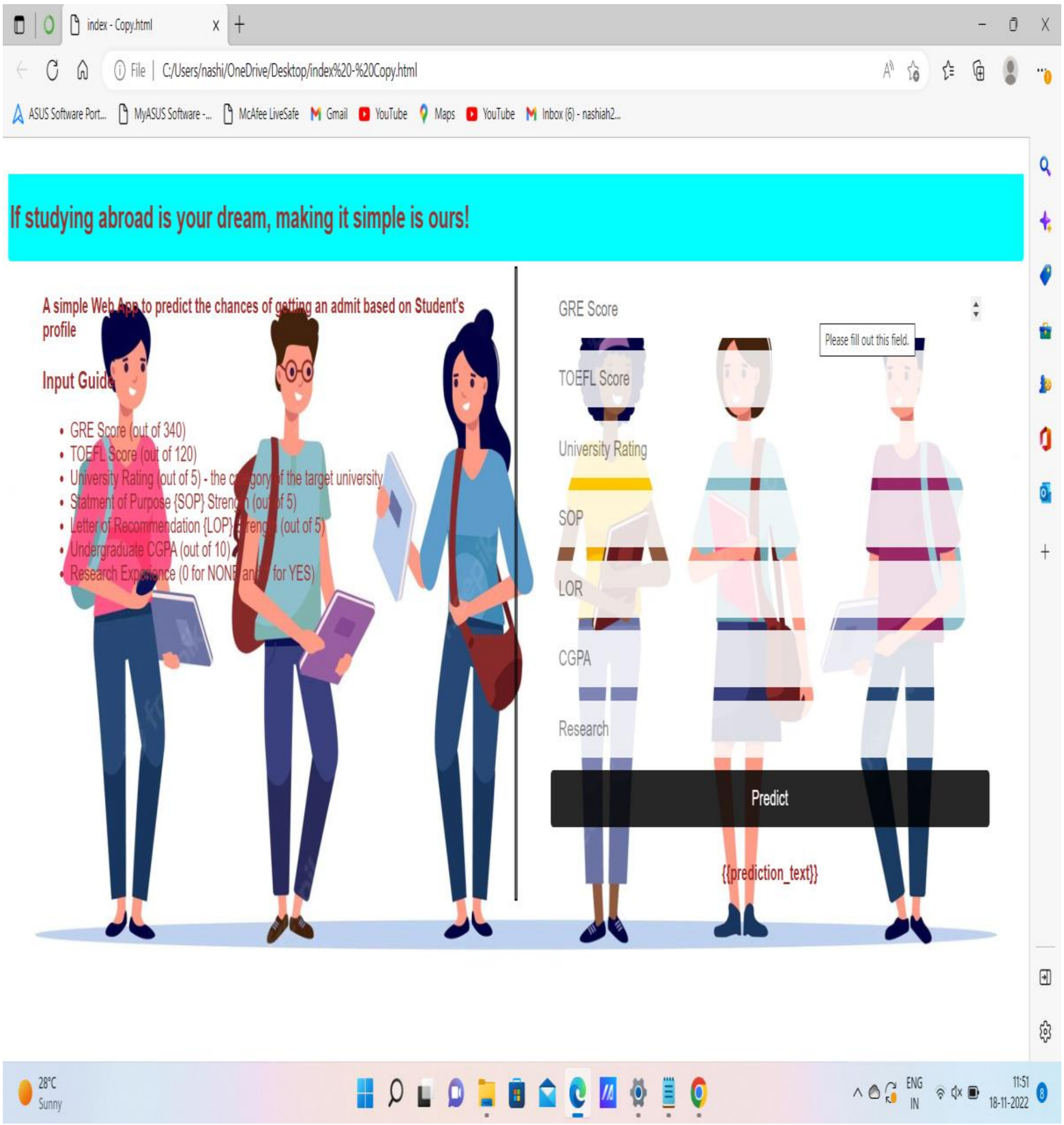
```
rec_num=X_test.shape[0]

#scaler = StandardScaler()

#X_train = scaler.fit_transform(X_train)

#X_test = scaler.fit_transform(X_test)

linreg = LinearRegression()

linreg.fit(X_train, y_train)

pickle.dump(linreg, open('model.pkl','wb'))

model=pickle.load(open('model.pkl','rb'))

#print(model.predict([[gre,toefl,rating,sop,lor,cgpa,research]]))

#-------------------------------------end model -----------------------------------

y_predict=linreg.predict(X_test)

prediction=round(y_predict[rec_num-1]*100,2)

#prediction=round(y_predict[0]*100,2)

if(prediction>=75):

message="Good Job! Your current scores show that you are well on the path to joining your dream college! Keep up the hardwork and dont forget about the other factors of your application"

elif(prediction>=50 and prediction<75):

message="Needs Improvement! Your current scores show that some more effort to get you to your dream college! If improvement in these areas is not possible, focus on the other factors of your application"

else:

message="Sorry! Your current scores show that the chances of you getting into this tier of universities are very slim! Might we suggest that you look at other options?"

#linreg_score = (linreg.score(X_test,y_test))

#print(linreg_score)

Record_dictionary2['prediction']=prediction
```

```
client1 = pymongo.MongoClient('localhost',27017)
db = client1['admin']
db2 =client1["UNIPREDICT"]
collection=db2["students"]
collection2=db2["data_table"]
temp=collection.find({'username':Record_dictionary2['username']})
print (temp)
for i in temp:
Record_dictionary2["First_Name"]=i['firstname']
Record_dictionary2["Last_Name"]=i['lastname']
#Record_dictionary3=Record_dictionary2
collection2.find_one_and_update(
{'username': session['user']},
{"$set":
{'gre' : gre,
'toefl': toefl,
'uni': uni,
'cgpa': cgpa,
'research': research,
'sop' : sop,
'lor': lor,
'prediction': prediction}
},upsert=True)
return render_template('pages/output.html', prediction=prediction, message=message)
return render_template('forms/predictor.html', form
```

CHAPTER-9

SCREENSHOTS

➢ **Index:**



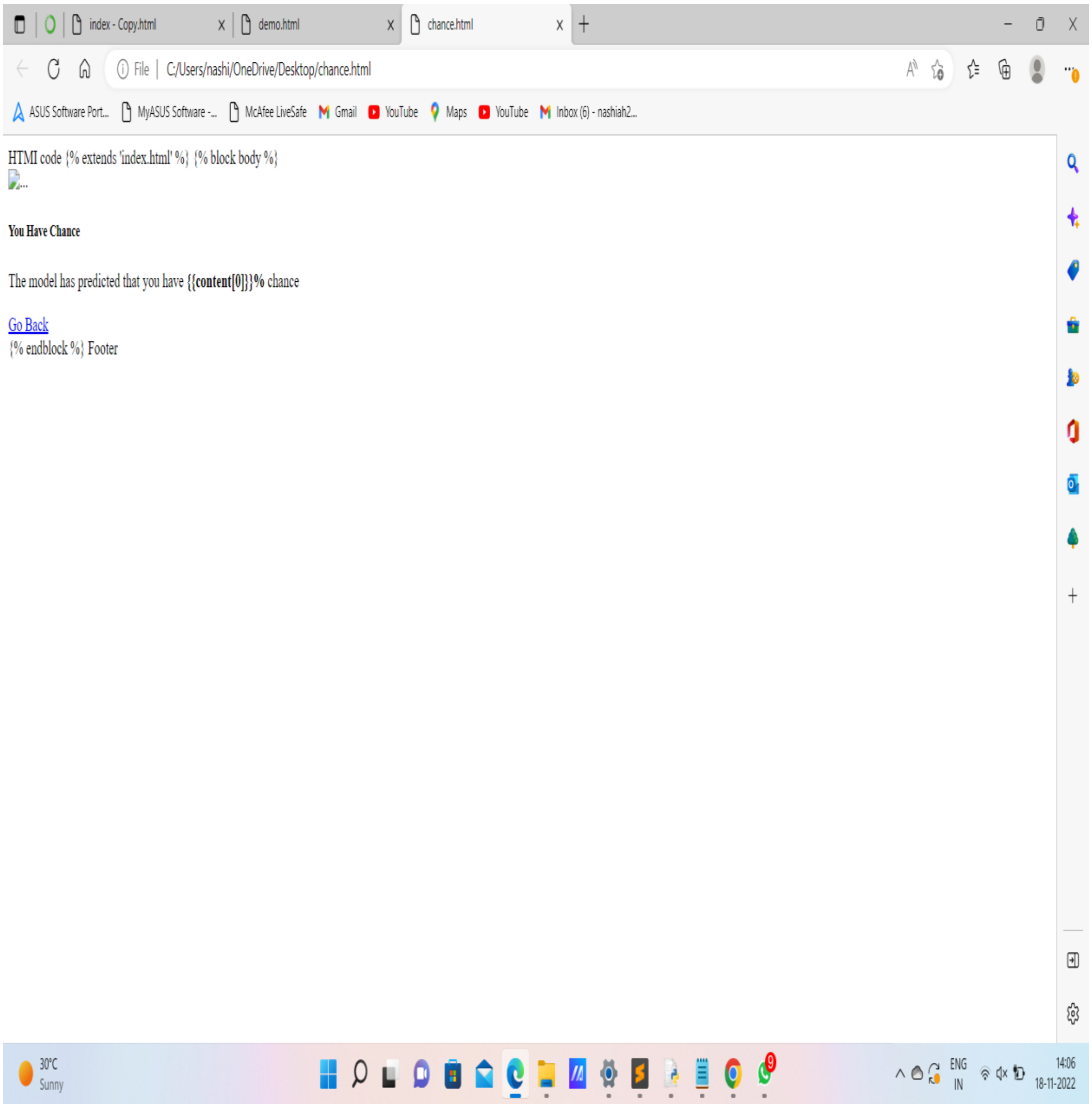
If studying abroad is your dream, making it simple is ours!

A simple Web App to predict the chances of getting an admit based on Student's profile

Input Guide

- GRE Score (out of 340)
- TOEFL Score (out of 120)
- University Rating (out of 5) - the category of the target university
- Statment of Purpose {SOP} Strength (out of 5)
- Letter of Recommendation {LOP} Strength (out of 5)
- Undergraduate CGPA (out of 10)
- Research Experience (0 for NONE and 1 for YES)

➢ Chance:



HTMl code {% extends 'index.html' %} {% block body %}

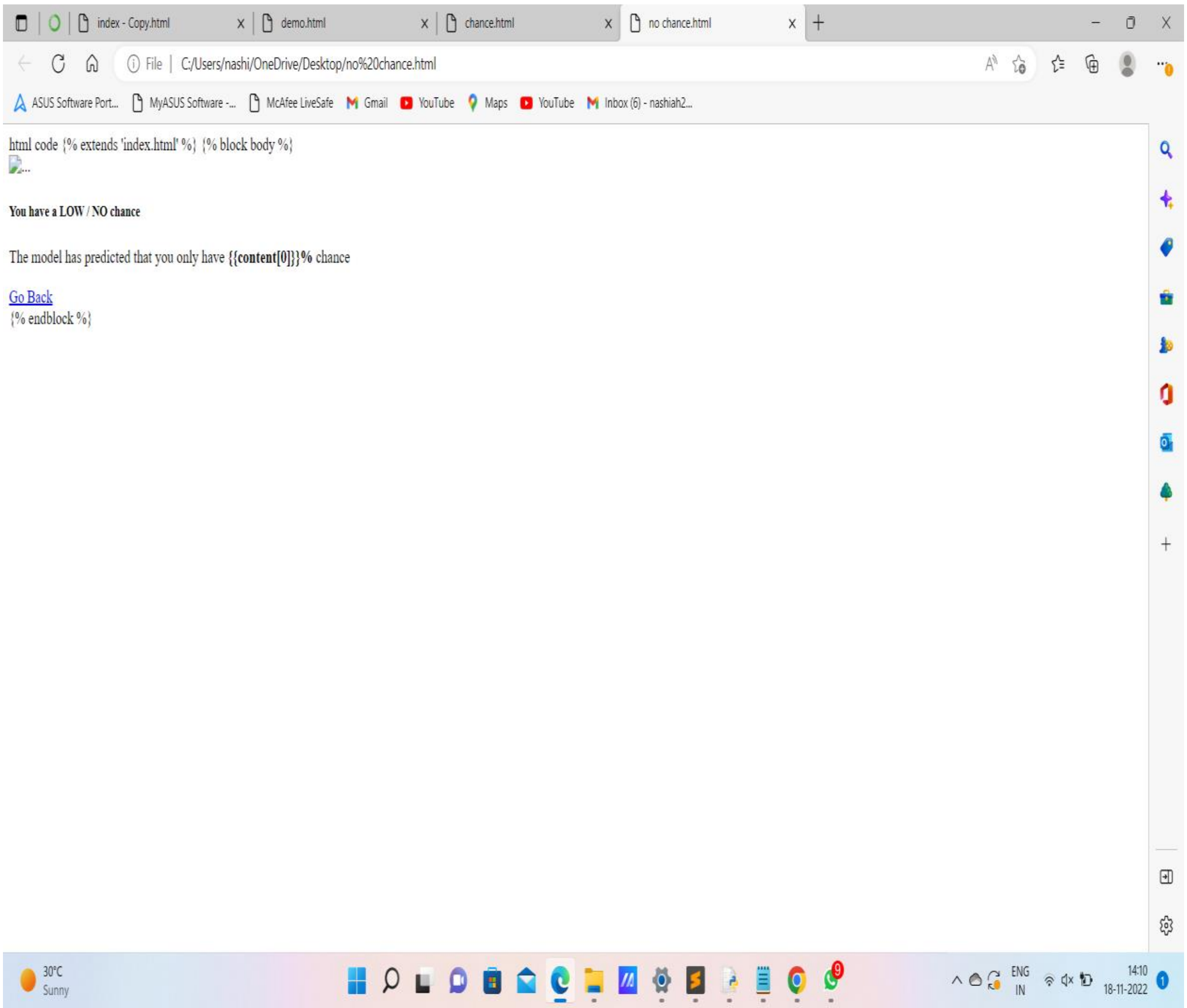
You Have Chance

The model has predicted that you have **{{content[0]}}%** chance

Go Back

{% endblock %} Footer

➢ No Chance:



html code {% extends 'index.html' %} {% block body %}

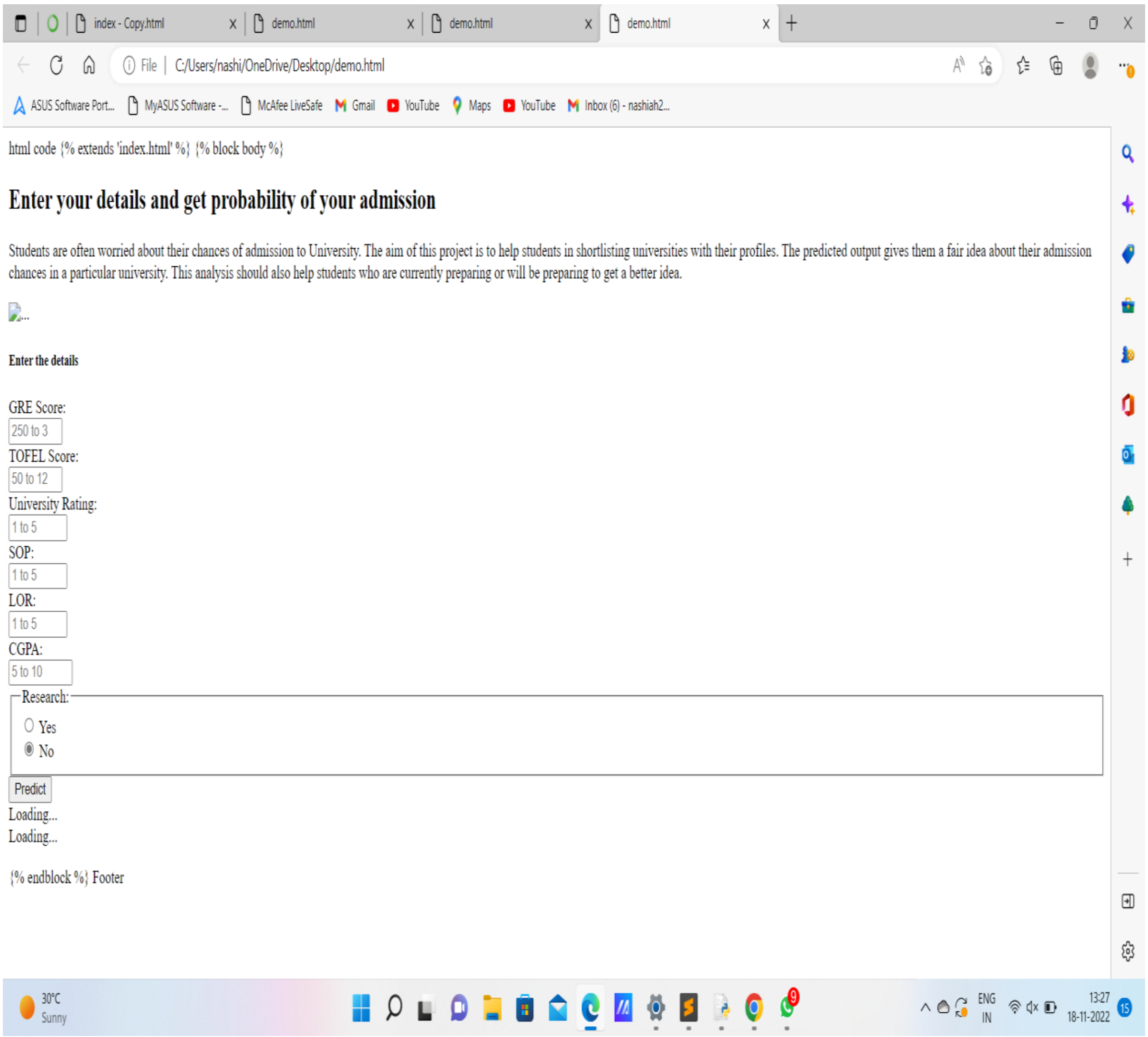
You have a LOW / NO chance

The model has predicted that you only have **{{content[0]}}%** chance

Go Back

{% endblock %}

➢ Enter Your Details:



html code {% extends 'index.html' %} {% block body %}

Enter your details and get probability of your admission

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.

Enter the details

GRE Score:
250 to 3
TOFEL Score:
50 to 12
University Rating:
1 to 5
SOP:
1 to 5
LOR:
1 to 5
CGPA:
5 to 10

Research:
○ Yes
● No

Predict

Loading...
Loading...

{% endblock %} Footer

CHAPTER-10

CONCLUSION & FUTURE ENHANCEMENT

10.1 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

10.2 Future Enhancement

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.