

ADHIYAMAAN COLLEGE OF ENGINEERING (AUTONOMOUS)

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

**PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND
ENTREPRENEURSHIP**

TOPIC: UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

Team ID-PNT2022TMID08090

TEAM LEADER :

BHUVAN HR

TEAM MEMBERS :

**CHETHAN R
DINESH S
DINESHHKUMAR S**

CONTENTS

1. Introduction

Project Overview

Purpose

2. Literature Survey

Existing Problem

References

Problem Statement Definitions

3. Ideation & Proposed Solution

Empathy Map Canvas

Ideation & Brainstorming

Proposed Solution

Problem Solution Fit

4. Requirement Analysis

Functional Requirement

Non-Functional Requirement

5. Project Design

Data Flow Diagrams

Solution & Technical Architecture

User Stories

6. Project Planning & Scheduling

Sprint Planning & Estimation

Sprint Delivery Schedule

Reports From JIRA

7. Coding & Solutioning

Feature 1

Feature 2

8. Testing

Test Cases

User Acceptance Testing

9. Results

9.1 Performance Metrics

10. Advantages & Disadvantages

11. Conclusion

12. Future Scope

13. Appendix

Source Code

GitHuB

ACKNOWLEDGEMENT

On the submission of this report on “UNIVERSITY ADMIT ELIGIBILITY PREDICTOR”, we would like to extend our gratitude and sincere thanks to our Mentor K.Sathyamoorthy , Assistant Professor, Department of computer science and Engineering for his constant motivation and support during the course. We truly appreciate and value his good guidance and encouragement from the beginning to the end of this project. We are indebted to his help for having helped us shape the problem and providing insights towards the solution.

1. INTRODUCTION

The world's business sector is escalating and is constantly seeking information and experiences that are commonly beneficial to individuals. Young specialists who need to stay in their current positions are always looking for advanced degrees to help them address their skills and information. As such, the number of international sophomores applying for graduation exams has increased over the past decade. One of her main concerns is getting into fantasy her university. You can see that undergraduates are actually choosing to get their education at prestigious universities. Furthermore, when it comes to international alumni, the United States is the main trend for most of them. The most prestigious universities offer a wide range of courses accessible in any order, exceptionally accredited teaching and education programs, an international second Researchscholarships for degrees are available.

According to Gauges, more than 4,444 of her 10 million international sophomores are enrolled in her 4,200+ colleges and universities, both private and public. In general, the number of undergraduates concentrated in America comes from Asian countries such as India, Pakistan, Sri Lanka, Japan and China. Select the United Kingdom, Germany, Italy, Australia, Canada as well as the United States. These countries are witnessing a rapid increase in the number of individuals seeking more advanced investigations. The basic reason why sophomores go on to master's programs in foreign graduate schools is that the number of vacancies is low and the number of people in these positions in each country is huge. This has led many professional undergraduates to pursue postgraduate studies. You can see that there are quite a few bachelor's degrees and master's degrees in computer science at US universities. The focus of this study applies to these undergraduate degrees. Many schools in the US follow comparative requirements for undergraduate accreditation. Schools consider several variables, including placement in fitness assessments and school performance ratings. English rankings are determined by exposure in English

proficiency tests such as TOEFL and IELTS.

The University's Admissions Advisory Board makes decisions regarding the acceptance or rejection of specific young researchers based on the general profile of the applicant's application. Records recorded with this company are marked with informative areas. Acknowledgment is a 400-row data set containing seven different autonomic factors. ie

1. Graduate Record Examination 1 (GRE) score. The score consists of 340 foci.
2. English as a Foreign Language (TOEFL) test score. It consists of 120 priority areas.
3. University Rating. Shows the position of colleges offering bachelor's degrees among various colleges. Your score will be out of 5.
4. Statement of Purpose (SOP), a record written to reveal the life, motivations and inspirations of a selected degree/college applicant. The score consists of five focal points.
5. The strength of a letter of recommendation (LOR) verifies the applicant's professional experience, falsifies validity, supports certainty, and guarantees your competence. The score consists of five focal points.
6. Undergraduate GPA (CGPA) from 10.
7. Research experience (either 0 or 1) that could support the application, such as distributing research papers at conferences or filling out as a right-hand exam for university faculty. One valid variable can be anticipated which is possibility of affirmation, that is as per the input given will be going from 0 to 1.

PROJECT

OVERVIEW

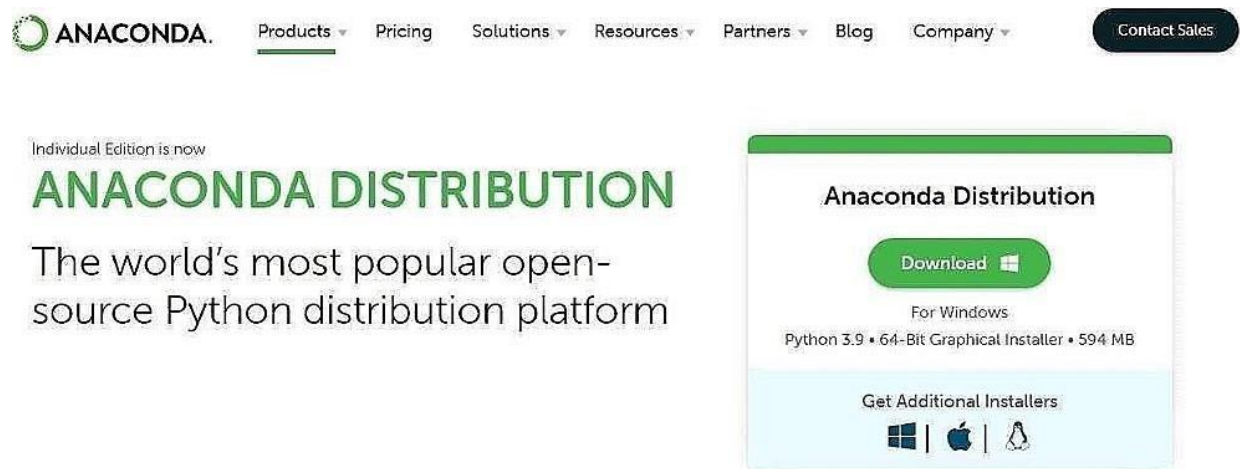
PRE REQUISITES

Anaconda Installation:

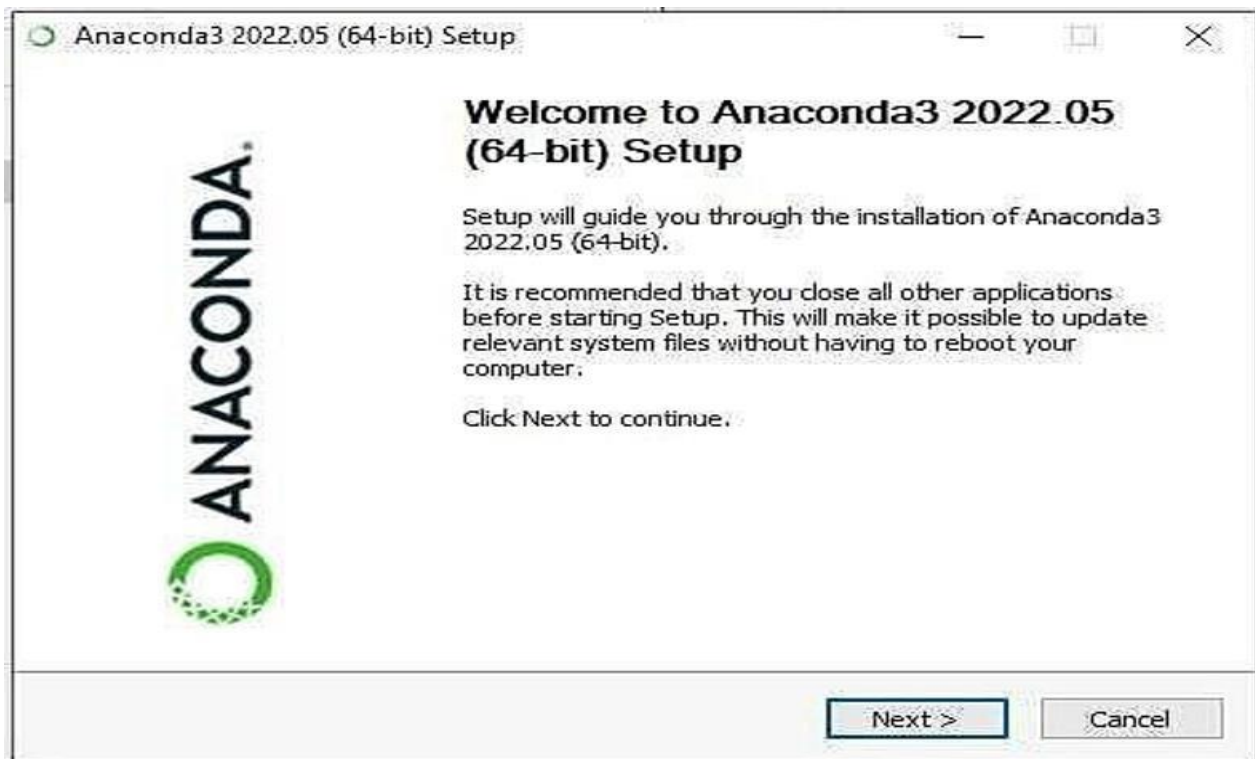
Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes data science packages suitable for Windows, Linux, and macOS. Developed and maintained by Anaconda. Founded in 2012 by Peter Wang and Travis Olyphant. As Anaconda, also known as Anaconda Distribution or Anaconda Individual Edition, the company's other products include his Anaconda Team Edition and Anaconda Enterprise Edition, neither of which are free.

WAY TO INSTALL ANACONDA:

STEP 1: Download and Anaconda



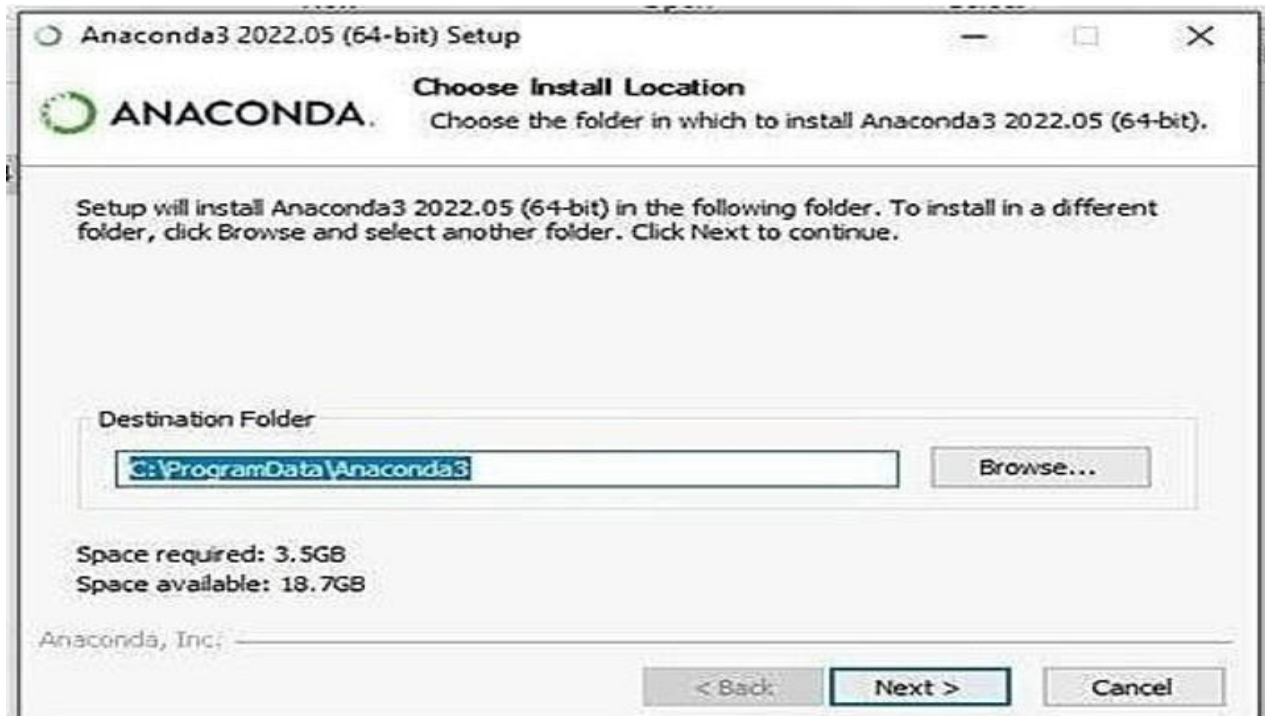
STEP 2: Install the Anaconda



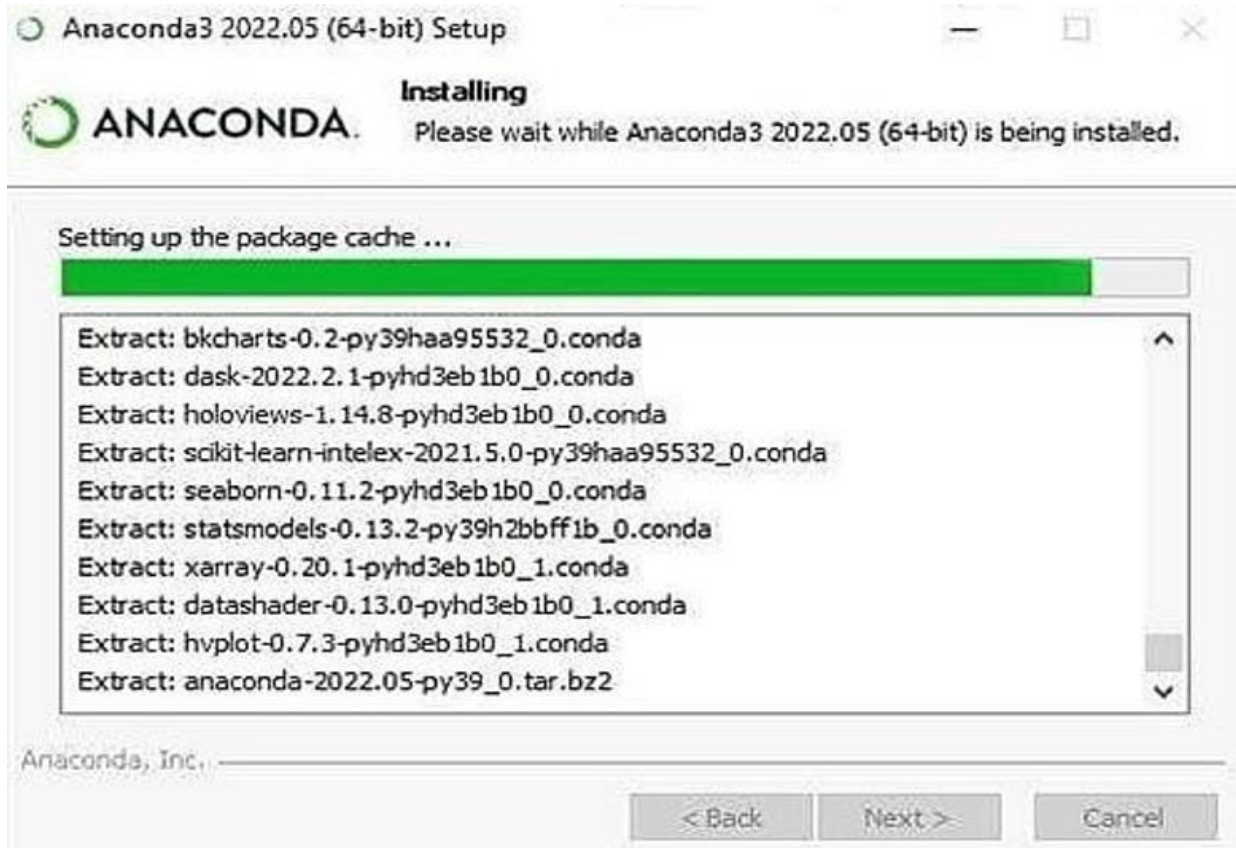
STEP 3: Click I Agree



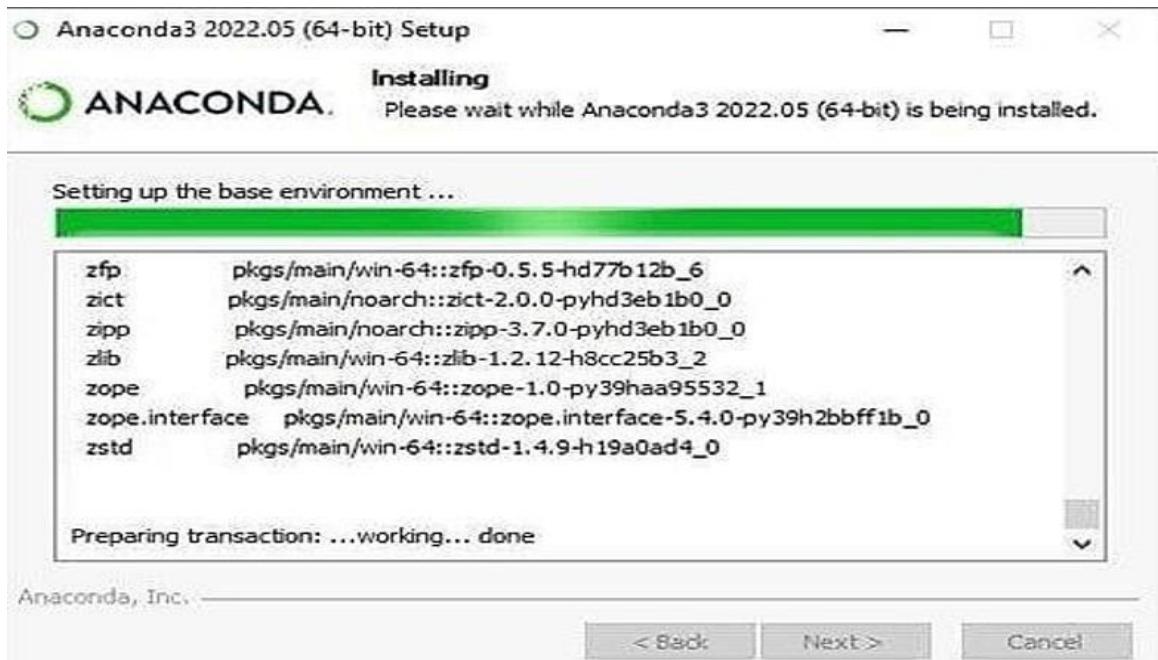
STEP 4: Choose the Installation Location



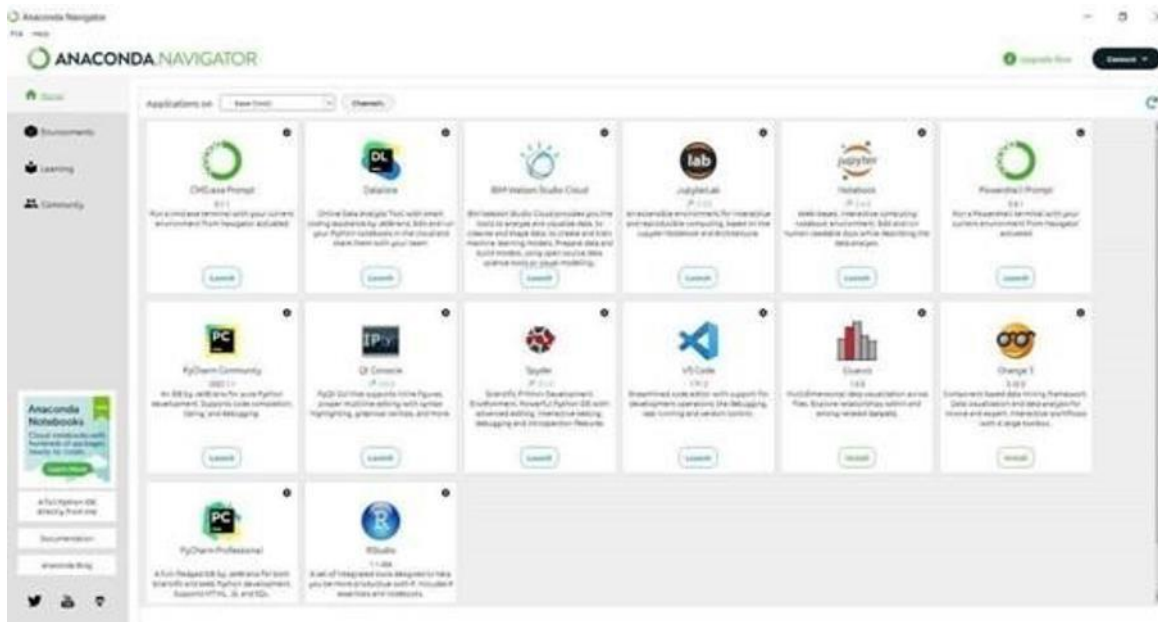
STEP 5: Installing the Requiring packages



STEP 6: Setting up the base environment

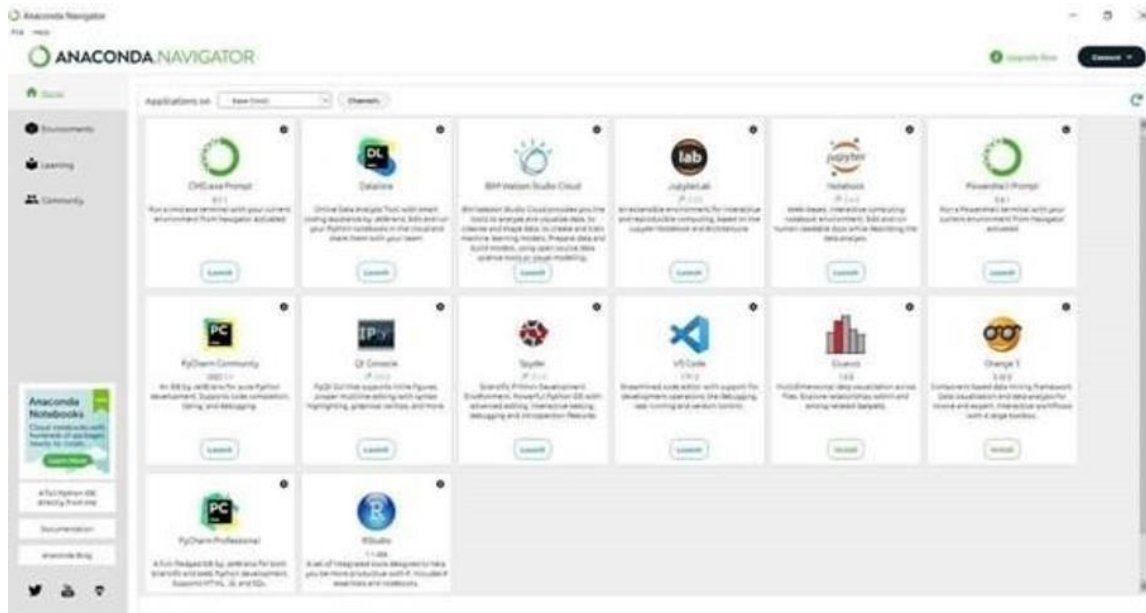


STEP 7: Successfully Installed and check the Anaconda Navigator working or not

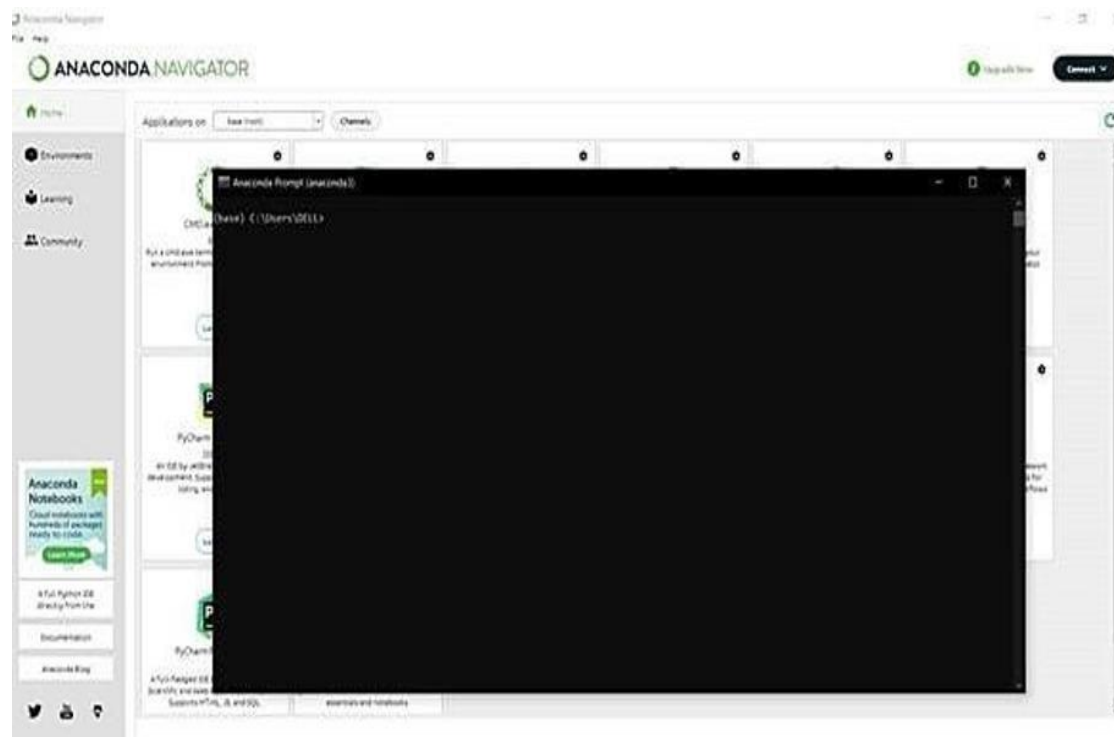


Python packages installation:

Step 1: Open the anaconda navigator in the start menu



Step 2: Open the CMD.exe prompt

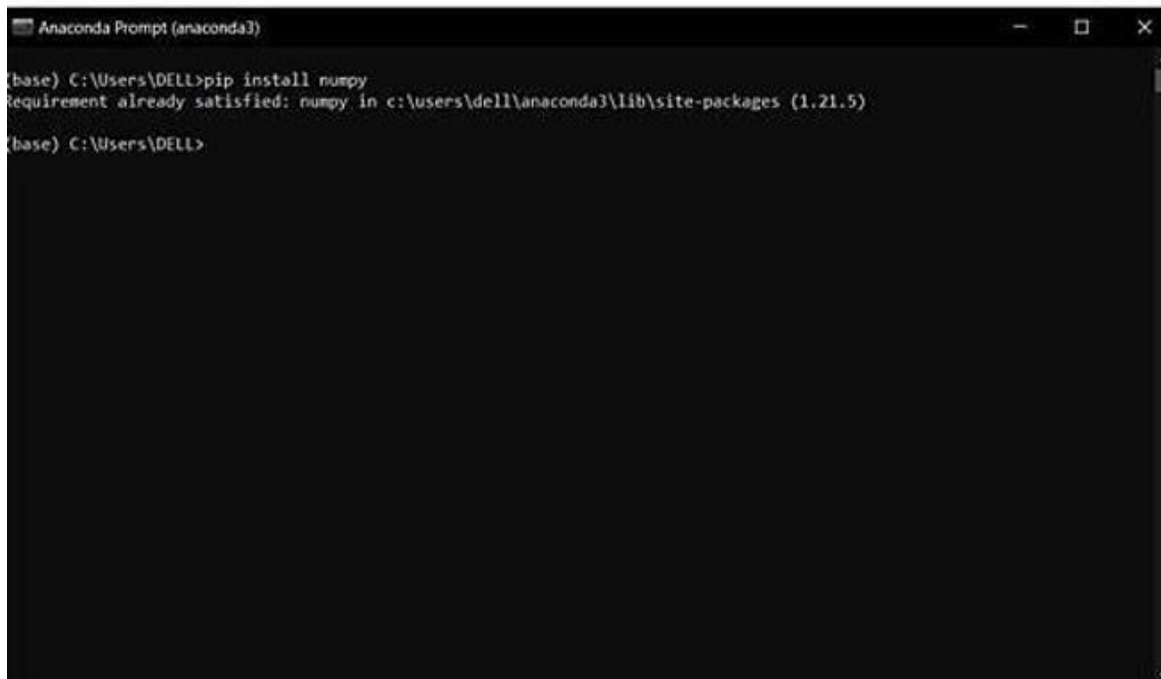


Step 3: Install the NUMPY package

To enter the numpy package enter the command in the CMD.exe Command: **Pip install numpy**

Numpy:

This package is used to perform numerical computations. This package comes pre-installed with Anaconda. NumPy is used for manipulating arrays. NumPy stands for Numerical Python.



```
Anaconda Prompt (anaconda3)
(base) C:\Users\DELL>pip install numpy
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (1.21.5)
(base) C:\Users\DELL>
```

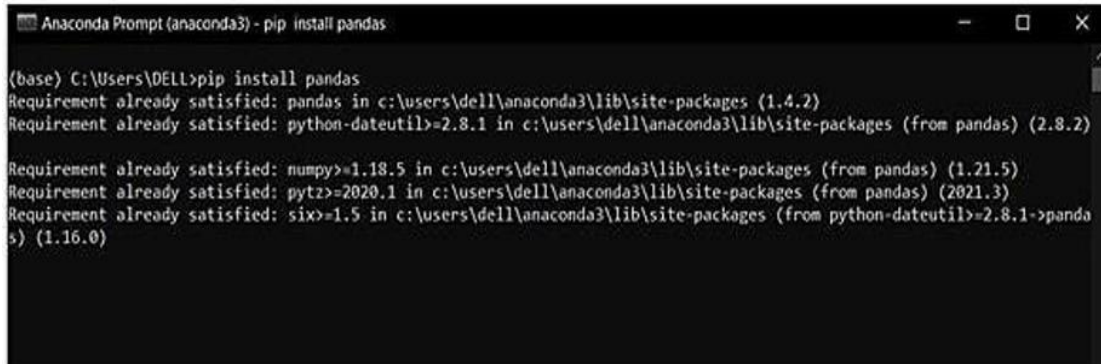
Step 4: Install the pandas package.

To enter the pandas package enter the command in the CMD.exe Command: **Pip install pandas**

Pandas:

Pandas is one of the most widely used Python libraries for data science. It provides powerful and easy-to-use structure and data analysis tools. This package comes pre-installed with Anaconda. An open source library built on top of the NumPy library. A Python package that provides various data structures and operations for

working with numerical data and time series. Mainly, it's common for data to be imported and analyzed much easier. Pandas is fast, providing users with high performance and productivity.



```
Anaconda Prompt (anaconda3) - pip install pandas

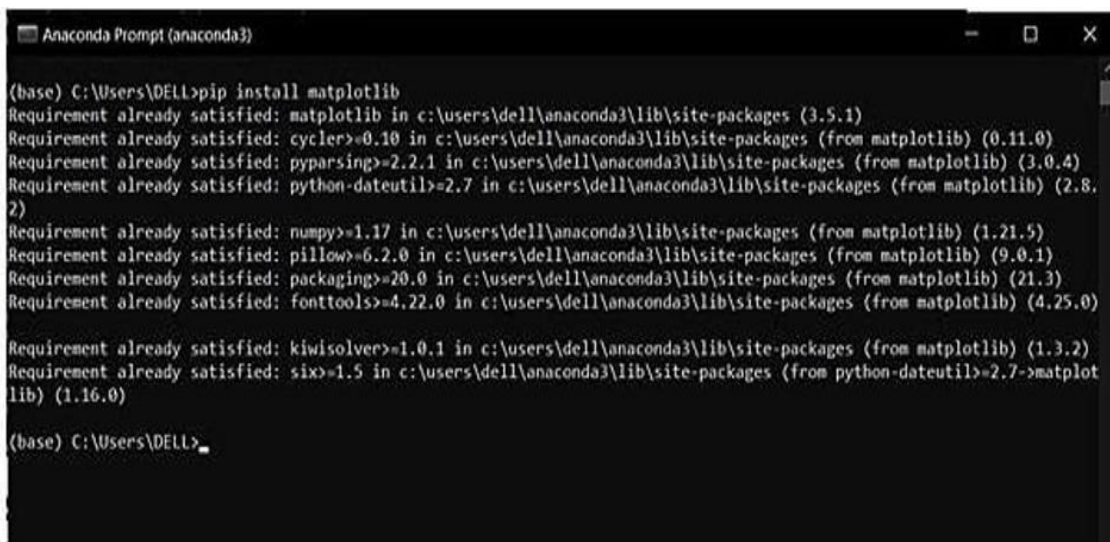
(base) C:\Users\DELL>pip install pandas
Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.18.5 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

Step 5: Install the Matplotlib package.

To enter the Matplotlib package enter the command In the CMD.exe Command: **Pip install Matplotlib**

Matplotlib:

Matplotlib is a comprehensive library for creating static, animated and interactive visualizations in Python. This package comes pre-installed with Anaconda. Matplotlib is a nice visualization library in Python for 2D plotting of arrays. Matplotlib is a cross-platform data visualization library based on NumPy arrays and designed to work with the wider SciPy stack. Introduced by John Hunter in 2002.



```
Anaconda Prompt (anaconda3)

(base) C:\Users\DELL>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib\site-packages (3.5.1)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: numpy>=1.17 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (1.21.5)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)

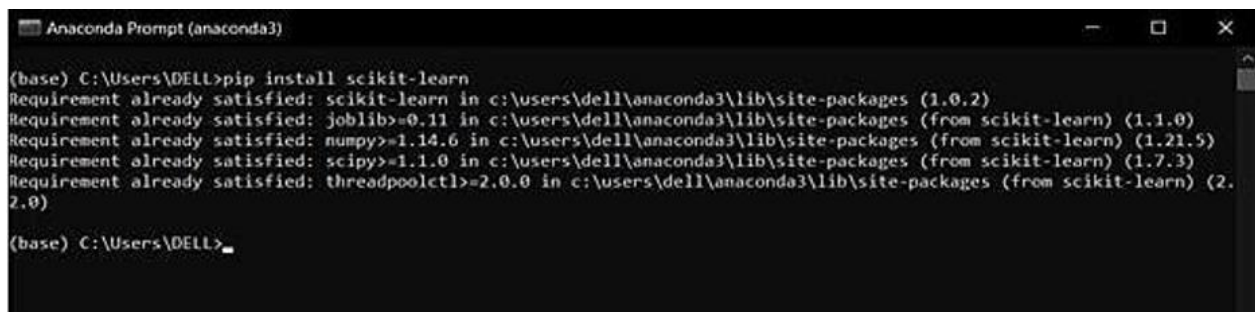
(base) C:\Users\DELL>
```

Step 6: Install the Scikit-learn package.

To enter the Scikit-learn package enter the command in the CMD.exe Command: **Pip install Scikit-learn**

Scikit-learn:

This is a machine learning library for the Python programming language. This package comes pre-installed with Anaconda. Scikit Learn in Python is primarily used to focus on modeling in Python. It was only focused on modeling, not loading data.

A screenshot of an Anaconda Prompt window with a dark background. The title bar reads 'Anaconda Prompt (anaconda3)'. The command prompt shows the command '(base) C:\Users\DELL>pip install scikit-learn'. The output displays that scikit-learn (1.0.2) is being installed from the local site-packages, and lists its dependencies: joblib (0.11), numpy (1.14.6), scipy (1.1.0), and threadpoolctl (2.0.0), all of which are already satisfied in the local environment. The prompt ends with '(base) C:\Users\DELL>_'.

```
(base) C:\Users\DELL>pip install scikit-learn
Requirement already satisfied: scikit-learn in c:\users\dell\anaconda3\lib\site-packages (1.0.2)
Requirement already satisfied: joblib>=0.11 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: numpy>=1.14.6 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.21.5)
Requirement already satisfied: scipy>=1.1.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.7.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
(base) C:\Users\DELL>_
```

Step 7: Install the Flask package.

To enter the Flask package enter the command in the CMD.exe Command: **Pip install Flask**

Flask:

Flask is a lightweight WSGI web application framework. Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts called Pocco. Flask is based on the WSGI toolkit tools and the Jinja2 template engine. Both are Pocco projects.


```
Anaconda Prompt (anaconda3)

(base) C:\Users\DELL>pip install flask
Requirement already satisfied: flask in c:\users\dell\anaconda3\lib\site-packages (1.1.2)
Requirement already satisfied: click>=5.1 in c:\users\dell\anaconda3\lib\site-packages (from flask) (8.0.4)
Requirement already satisfied: Werkzeug>=0.15 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.0.3)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.11.3)
Requirement already satisfied: itsdangerous>=0.24 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.0.1)
Requirement already satisfied: colorama in c:\users\dell\anaconda3\lib\site-packages (from click>=5.1->flask) (0.4.4)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\dell\anaconda3\lib\site-packages (from Jinja2>=2.10.1->flask) (2.0.1)

(base) C:\Users\DELL>
```

PROJECT FLOW

You will go through all the steps mentioned below to complete the project.

1. User interacts with the UI (User Interface) to enter Data
2. The entered data is analyzed by the model which is integrated
3. Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below

4. Data Collection.
 - a. Collect the dataset or Create the dataset
5. Data Preprocessing.
 - a. Import the Libraries.
 - b. Importing the dataset.
 - c. Checking for Null Values.
 - d. Data Visualization.
 - e. Taking care of Missing Data.
 - f. Label encoding.
 - g. One Hot Encoding.
 - h. Feature Scaling.
 - i. Splitting Data into Train and Test.

6. Model Building

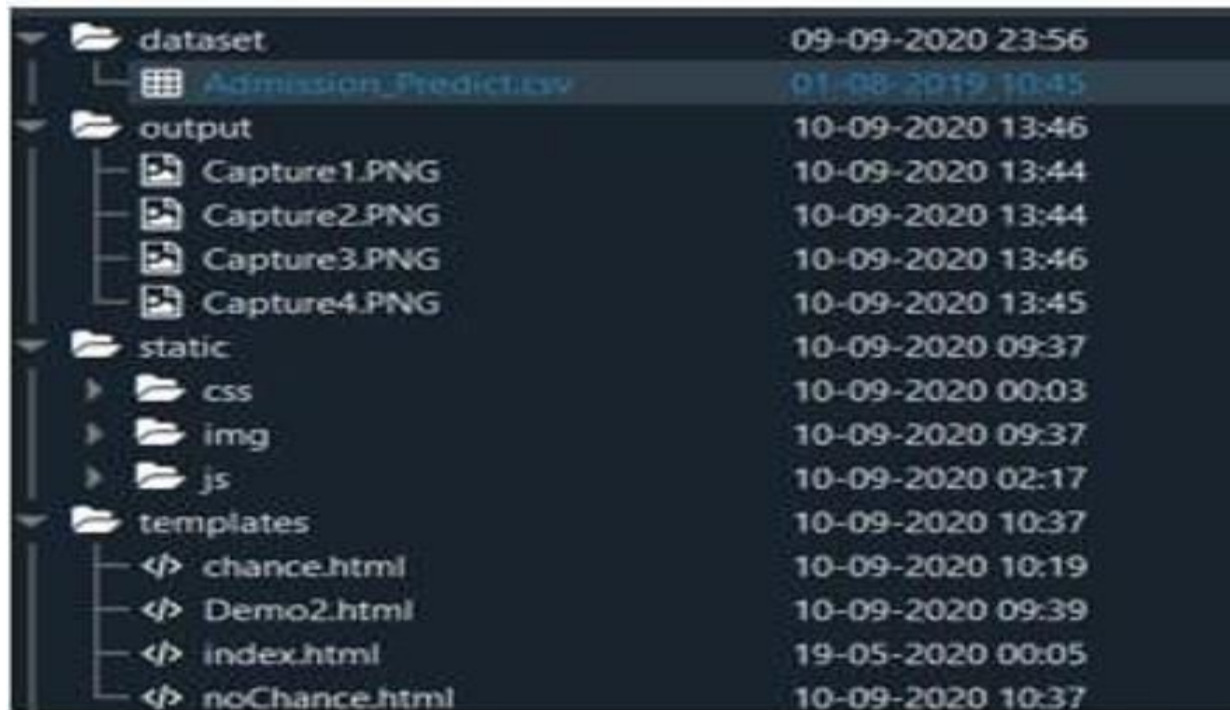
- a. Training and testing the model
 - b. Evaluation of Model
- 7. Application Building
 - a. Create an HTML file
 - b. Build a Python Code

PROJECT OBJECTIVES

- 8. To understand regression and classification problems.
- 9. To gain insights from data through visualization.
- 10. Applying different Machine Learning algorithms to determine the probability of acceptance in a particular university.
- 11. Evaluation metrics build a web application using the Flask framework

PROJECT STRUCTURE

Create a Project folder that contains files as shown below




DATA COLLECTION


The path to common information varies by project type. ML projects use real-time information. Information indexes can be collected from a variety of sources such as documents, data sets, sensors, and other sources, using free information collection from the Internet. Kaggle and the UCI Machinelearning Repository are the most commonly used repositories for sorting


information for machine learning models. Kaggle is probably the most visited website used for information gathering. Collect the dataset or Create the dataset.

481 lines (481 sloc) | 12.6 KB

RawBlame







🔍 Search this file...

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1	1	337	118	4	4.5	4.5	9.65	1	0.92
2	2	324	107	4	4	4.5	8.87	1	0.76
3	3	316	104	3	3	3.5	8	1	0.72
4	4	322	110	3	3.5	2.5	8.67	1	0.8
5	5	314	103	2	2	3	8.21	0	0.65
6	6	330	115	5	4.5	3	9.34	1	0.9
7	7	321	109	3	3	4	8.2	1	0.75
8	8	308	101	2	3	4	7.9	0	0.68
9	9	302	102	1	2	1.5	8	0	0.5
10	10	323	108	3	3.5	3	8.6	0	0.45
11	11	325	106	3	3.5	4	8.4	1	0.52
12	12	327	111	4	4	4.5	9	1	0.84
13	13	328	112	4	4	4.5	9.1	1	0.78
14	14	307	109	3	4	3	8	1	0.62
15	15	311	104	3	3.5	2	8.2	1	0.61
16	16	314	105	3	3.5	2.5	8.3	0	0.54
17	17	317	107	3	4	3	8.7	0	0.66
18	18	319	106	3	4	3	8	1	0.65
19	19	318	110	3	4	3	8.8	0	0.63
20	20	303	102	3	3.5	3	8.5	0	0.62
21	21	312	107	3	3	2	7.9	1	0.64
22	22	325	114	4	3	2	8.4	0	0.7
23	23	328	116	5	5	5	9.5	1	0.94
24	24	334	119	5	5	4.5	9.7	1	0.95
25	25	336	119	5	4	3.5	9.8	1	0.97

DATA PRE-PROCESSING

Importing the Libraries:

It is important to import all the necessary libraries such as pandas, numpy, matplotlib.

● Numpy-

It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.

- **Pandas-**

It is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.

- **Seaborn-**

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

- **Matplotlib-**

Visualisation with python. It is a comprehensive library for creating static, animated, and interactive visualizations in Python

Reading the Datasets:

You might have your data in .csv files, .excel files

Let's load a .csv data file into pandas using `read_csv()` function. We will need to locate the directory of the CSV file at first

```
#read_csv is a pandas function to read csv files  
data = pd.read_csv('Admission_Predict.csv')
```

If your dataset is in some other location, Then see

below command

```
pd.read_csv(r"File_location/filename.csv")
```

Note: `r` stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.

Our Dataset Admission_Predict contains
following Columns

- 1.Serial No.
- 2.GRE Score
- 3.TOEFL Score
- 4.University Rating
- 5..SOP
- 6.LOR
- 7.CGPA
- 8.Chance of Admit

Handling Missing Values:

After loading it is important to check the complete information of data as it can indicate many of the hidden information such as null values in a column or a row. Check for the null values. If it is present then the following steps can be performed

- Imputing data using the Imputation method in sklearn.
- Filling NaN values with mean, median, and mode using fillna() method. You can check the null values with the function isnull().any()

```
data.isnull().any()
```

```
GRE Score      False  
TOEFL Score    False  
University Rating  False  
SOP            False  
LOR            False  
CGPA           False  
Research       False  
Chance of Admit  False  
dtype: bool
```

- If the dataset contains null values then the above functions return as true. But if you look at the dataset you can observe that the dataset does not have any null values.
- You can also check the number of null values present in the columns by the using `isnull().sum()` function

As we don't have categorical data then we can skip the steps of label encoding and one-hot encoding

Data Visualization:

Data visualization is where a given dataset is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data. Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. Machine learning models will perform poorly on data that wasn't visualized and understood properly.

To visualize the dataset we need libraries called Matplotlib and Seaborn. The Matplotlib library is a Python 2D plotting library that allows you to generate plots, scatter plots, histograms, bar charts etc.

Splitting Dependent And Independent Columns:

We need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.

- To read the columns, we will use `iloc` of pandas (used to fix the indexes for selection) which takes two parameters —
[row selection, column selection].

Let's split our dataset into independent and dependent variables.

```
x=data.iloc[:,0:7].values  
x
```

```
y=data.iloc[:,7:].values  
y
```

From the above code “:” indicates that you are considering all the rows in the dataset and “0:7” indicates that you are considering columns 0 to 7 such as year, month, and day as input values and assigning them to variable x. In the same way in the second line “:” indicates you are considering all the rows and “7:” indicates that you are considering only the last column as output value and assigning them to variable y.

Let's Check the shape of x and Y

```
x.shape
(1991, 7)

y.shape
(1991, 1)
```

- You can see in x we have 1991 rows with 7 columns and y has 1 column with the same number of rows

Splitting The Data Into Train And Test:

To train the model, first split the model into two segments: "training data" and "testing data". The classifier is trained using a 'training data set' and the performance of the classifier is tested on a non-fitting 'test data set'.

Training Set: The training Set is material for computers to learn how to process data. The AI uses computation to do the training part. The training dataset is used to learn and tune the classifier parameters.

Test set: A set of unseen data used solely to evaluate the performance of the fully displayed classifier.

When you are working on a model and you want to train it, you obviously have a dataset. But after training, we have to test the model on some test dataset. For this, you will need a dataset that is different from the training set you used earlier. But it might not always be possible to have so much data during the development phase. In such cases, the solution is to split the dataset into two sets, one for training and the other for testing.

To help us with this task, the Scikit library provides a tool, called the Model

Selection library. There is a class in the library which is, 'train_test_split.' Using this we can easily split the dataset into the training and the testing datasets in various proportions.

The train-test split is a technique for evaluating the performance of a machinelearning algorithm.

- Train Dataset:Used to fit the machine learningmodel.
- Test Dataset:Used to evaluatethe fit machinelearning model.

```
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,random_state = 10)
```

In general, you can allocate 80% of the dataset to the training set and the remaining 20% to the test set. We will create 4 sets

- x_train
- x_test
- y_train
- y_test .

There are afew other parameters that we need to understand before we use the class:

- test_size: this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset and remaining a train dataset
- random_state: here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the Random_state class, which will become the number generator. If you don't pass anything,the Random_state instanceused by np.random will be used instead.

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

Existing Problem

It's almost admission season and I've know a couple of friends who are in panic mode waiting for a call from the universities they've applied.

This made me think— How can we predict whether a student will get an admission 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.

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.

References: M. Fatima and M. Pasha, "Survey of machine learning algorithms for disease diagnostic", *Journal of Intelligent Learning Systems and Applications*, vol. 9, no. 01, pp. 1, 2017.

> <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 WCE2011.

➤ <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 spend 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://medium.com/@jigar18011999/university-predictor-by-machine-learning-2d880e9f3a3>

Abstract:

This article describes the architecture and algorithms of the proposed system. ANN,

decision trees, and logistic regression were used to find admissions for a particular student. ML models take into account various parameters such as GRE and TOEFL scores, SOP, and LOR. Finally, after evaluation, the authors state that decision trees are the most accurate among the tree algorithms used.

> <https://github.com/satwik2663/Machine-Learning-Graduate-Student-Admission-Predictor>

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:

- i. Analyzed university admission statistics.
- ii. 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 users chances for getting into a certain college.

b. Problem Statement Definition

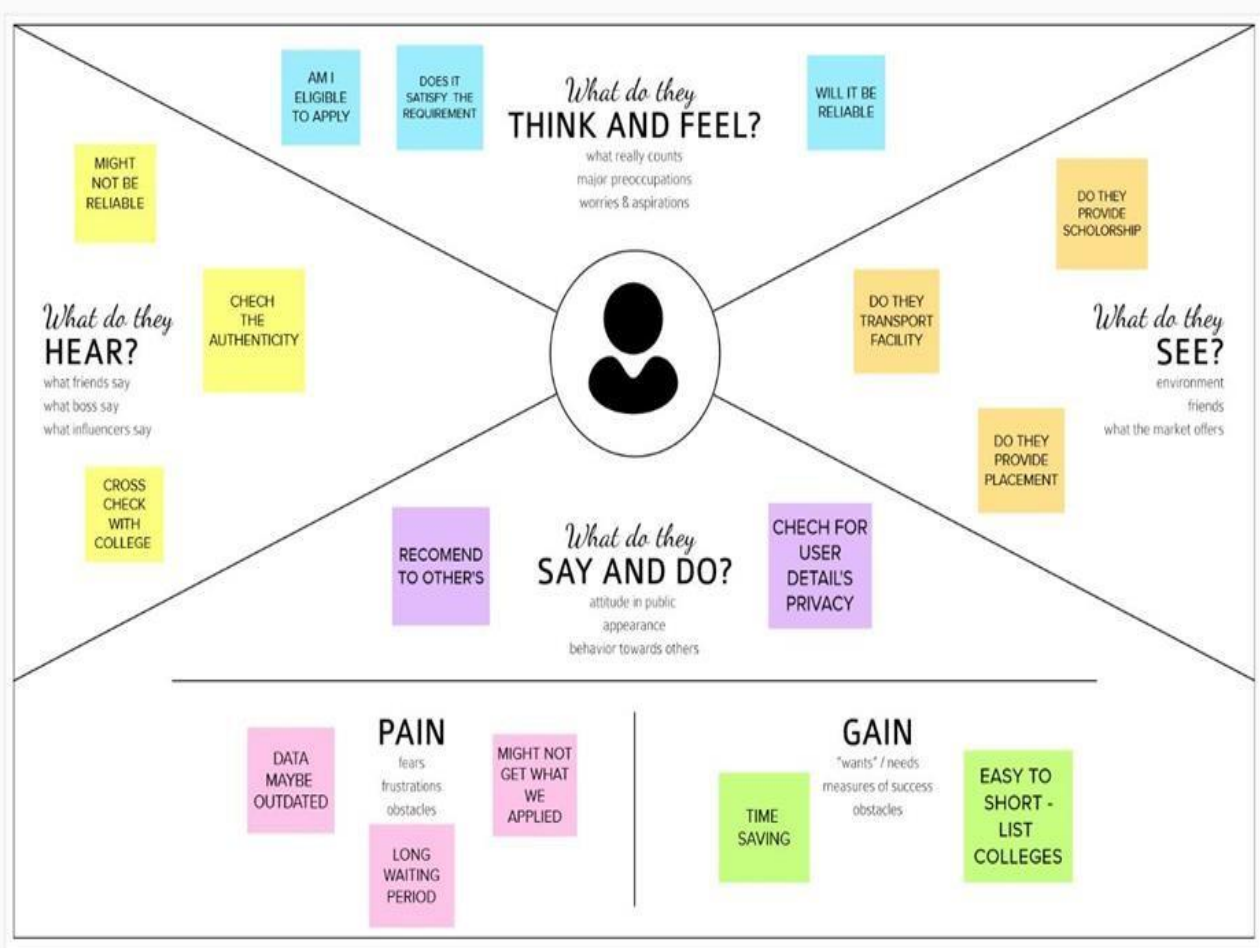
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A student	Enroll for a PG course in the university	I am unaware of eligible cut off marks	The calculation for the cut off marks is not available in the website.	Getting confused to admit university to get admission for different universities.
PS-2	Administrator	To download all the data's from the students.	Detailed furnished by the students are incomplete.	There is the problem in the server because of huge volume of data.	Frustrated and not able to download the documents.
PS-3	A student	Enroll in a PG degree	I am unaware of scholarship details are unknown	Unable to get the funds from different organizations	Depressed due to lack of financial support

3.IDEATION AND PROPOSEDSOLUTION

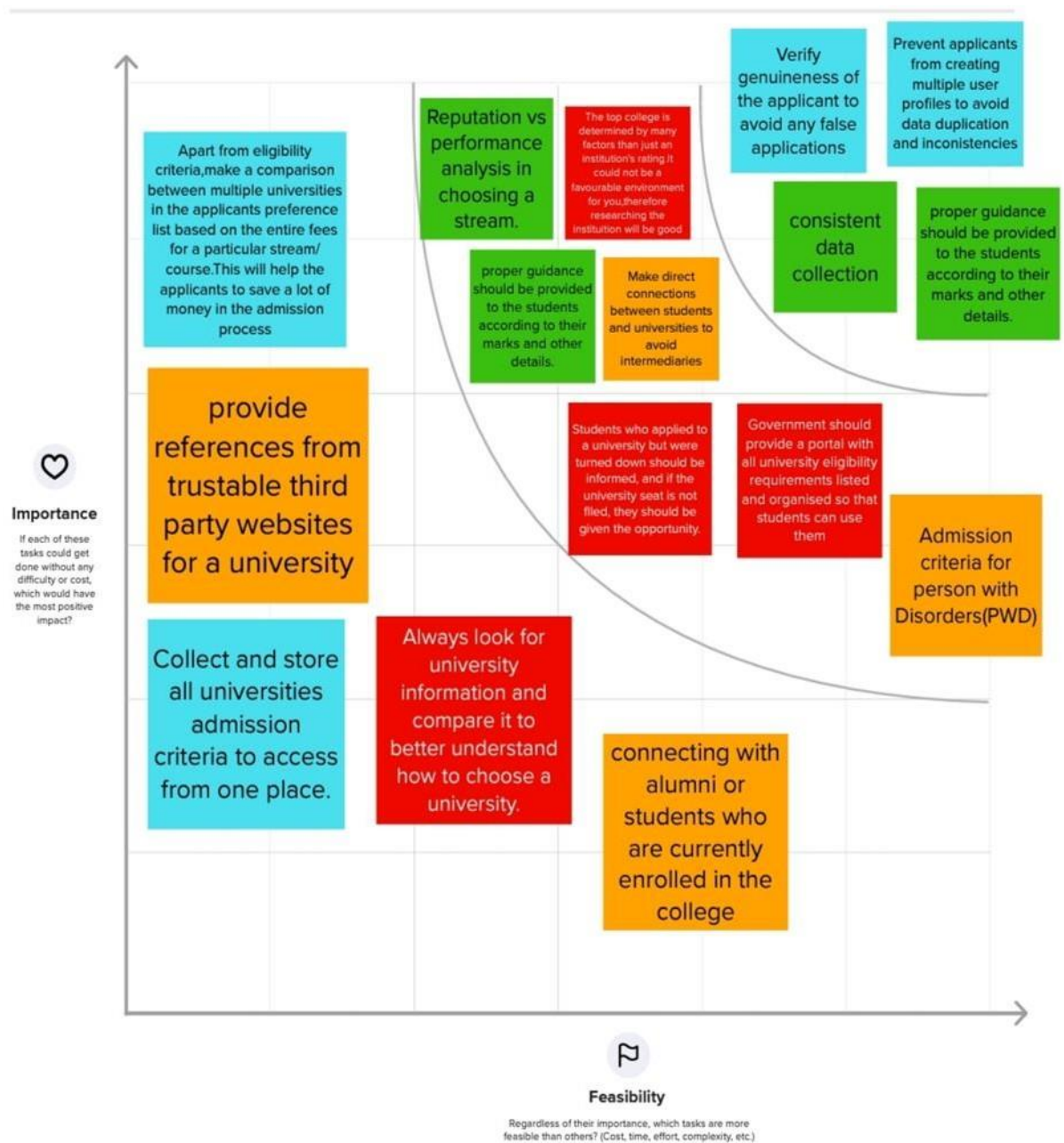
Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brainwriting, Worst Possible Idea, and a wealth of other ideation techniques. Ideation is also the third stage in the Design Thinking process. In this project the ideation phase consist of,

- Empathy Map
- Bíainstoíming
- Píoposed Solution
- Píoblem Solution Fit

Empathy Map



Ideation & Búainstoíming



Proposed Solution

1.	Problem Statement (Problem to be solved)	With the increased rates in the number of students/learners opting for abroad universities that provide niche and specialized courses/programs, higher job opportunities and salary packages, student want to be able to get in best Universities. To make such informed decision student look for online free prediction tools that can present them instant results – comprehensive and detailed and need no fees and can be accessed easily.
2.	Idea / Solution description	The Prediction model will be built to be efficient and effective using Machine Learning algorithms like – KNN, linear regression etc that provides accurate prediction based on past data collected. Users will get instant prediction results – the percentage chance of eligibility- on a given input of exam scores and university. Provided in graphical representation users can quickly grasp the output
3.	Novelty / Uniqueness	We'll be developing a straightforward user interface that will enable users to enter information about a student's profile and receive the application's forecasted result as an output. This project's goal is to assist students in narrowing down institutions based on their profiles. The anticipated results offer them a good indication of their prospects of admission to a particular university.

4.	Social Impact and Customer Satisfaction	<p>1. The prediction system will be embedded in an application service for ubiquitous access and free of charge, sign in requirements.</p> <p>2. The results of prediction will be visualized in the form of intelligible charts/graphs with the past score/cut-off of a university.</p> <p>3. Display of all possible eligible University and the requirements other than grades.</p> <p>4. Provision of necessary links/Blogs for users to discern</p>
5.	Business Model (Revenue Model)	<p>The need for online prediction system/service is in high demand after the covid-19. Student/Users need instant and reliable predictors. This system can cater to wide range of users apart from students and can employ subscriptions, sign in and other features to gain revenue. The system can be licensed to further generate revenue.</p>
6.	Scalability of the Solution	<p>The prediction system equipped with efficient ML algorithm can be further integrated with features like -location based university recommendation system, Colleges with high rate of admit predictions etc. By employing advanced frameworks, the application can be further improved for better user experience and usability.</p>

Problem Solution fit

Project Title: University Admit Eligibility Predictor
Team ID: PNT2022TMID25845

Project Design Phase-I - Solution Fit Template

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) Who is your customer? i.e. working parents of 0-5 y.o. kids <ul style="list-style-type: none"> A graduated student looking for higher studies A student preparing or will be preparing for exams An education consultancy provider tied with colleges to help students in admissions. 	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. <ul style="list-style-type: none"> Financial Limitations Unable to acquire costly consultancy services Obstacles in procuring required documents for various university application Unaware about each university's eligibility criteria's and in confusion about where to apply 	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. <p>Available solutions:</p> <ul style="list-style-type: none"> Online prediction websites/apps Education Consultancy Help from university alumni/Friends <p>Cons:</p> <ul style="list-style-type: none"> Unreliable prediction with varied output for same input Expensive Services
	Focus on J&P, map into BE, understand RC	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. <p>Problems Addressed:</p> <ol style="list-style-type: none"> Prediction system > Accurate, free of cost easy to understand Update university admit eligibility factors Providing instant results providing User-friendly web services with good visualizations Providing necessary details about the exams and score required. 	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. <p>Students of modern era and after COVID-19 rely on and want instant access to information/services that are:</p> <ol style="list-style-type: none"> Valid, relevant, and secure - available online services are accessible free of cost and are prompt in delivering correct information internet with its vastness is not free of fake,unreliable information and as such also has websites/app that unsafe/filled with ads and are irrelevant to students needs
Identify strong TR & EM		3. TRIGGERS What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. <ol style="list-style-type: none"> finding anything and everything is available online these days peers making use of expensive predictions services to reduce application 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. <p>lost, insecure, frustrated, anxious > aware, informed, driven to take informed steps, in control</p>	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. <ol style="list-style-type: none"> Use Effective ML algorithms and prediction model for efficient predictions using user-friendly free of cost web service Provide user-friendly visual graphs for ease of understanding Recommend probable colleges with high chance of admission Provision of prediction report in simple and legible format Provide all valuable info to users regarding prediction system to build trust

4. REQUIREMENT ANALYSIS

4.1 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 Email and Password
FR-3	User Login	Login through Email and Password.
FR-4	Administration work	Check qualified candidate detail Make allotment
FR-5	Admission Details	Check seat availability Check college infrastructure Check fees details
FR-6	Local counsellor	Issue the final allotment order

4.2 Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The UI/UX enhances the user experience. The entire journey of the customer throughout the application will be hustle free making it a smooth experience for the user.

NFR-2	Security	It is safe to use this application since no user data is stored
-------	-----------------	---

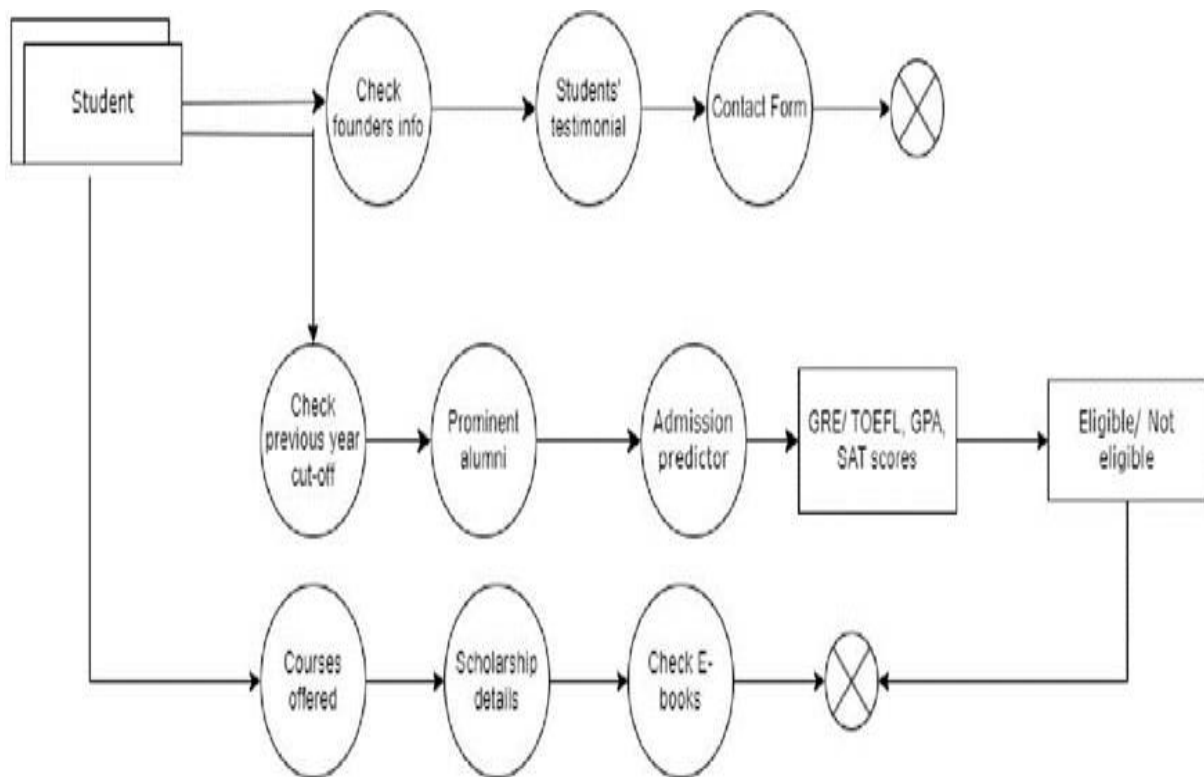
NFR-3	Reliability	<ul style="list-style-type: none">a. All data storage for use of variables will be committed to the database at the time of entry.b. Data corruption is prevented by applying the possible backup procedures and techniques.
-------	--------------------	---

NFR-4	Performance	<ul style="list-style-type: none"> a. The database should be able to accommodate a minimum of 10,000 records of students. b. At any instant the system should support use of multiple users at a time. c. Availability results of the Requested college should be presented to the student in max of two seconds, so retrieving of data should be reliable. d. As each student will be given a maximum time of 10 min, accessing from the database
		<p>should be</p> <p>done at relevant speed.</p>

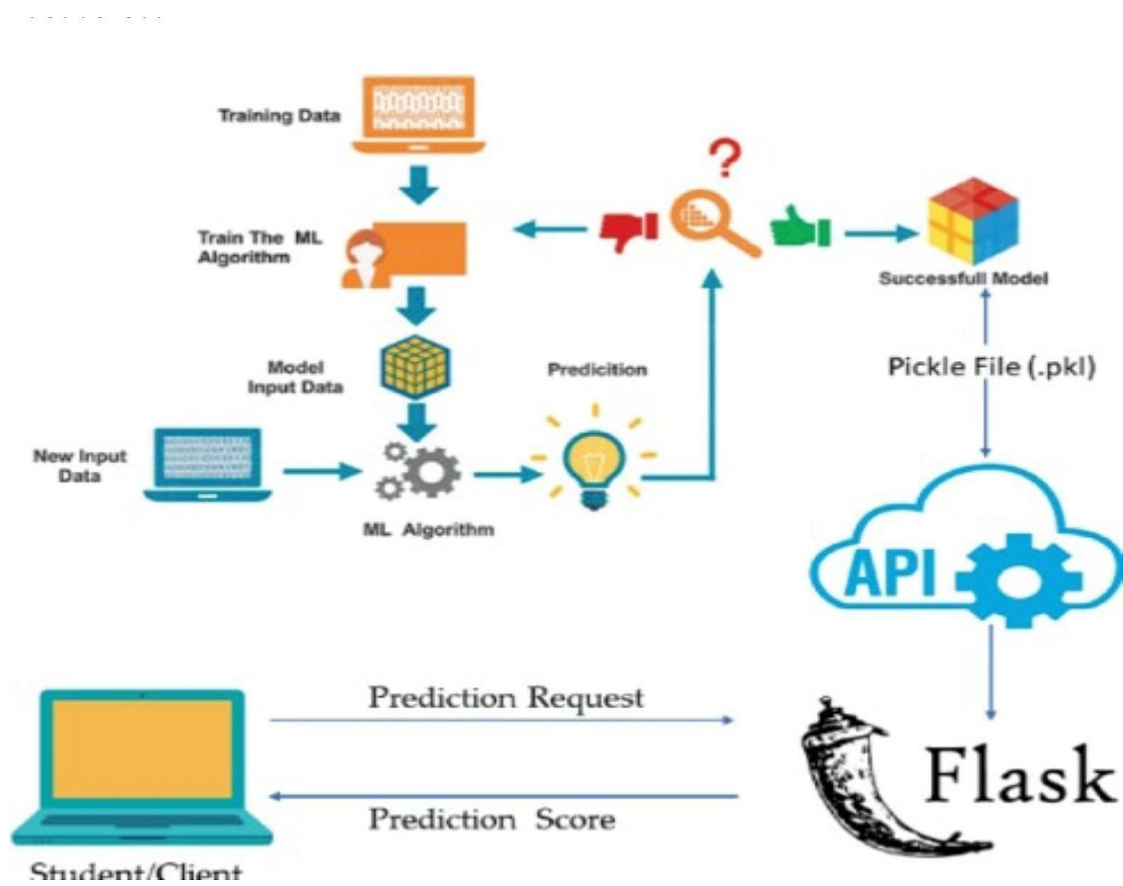
NFR-5	Availability	<p>The system should available at all the time meaning that the user can access easily.</p> <p>Increase of the hardware and data base failure a replacement page will be show andfor database back should be retrieved from data folder.</p>
NFR-6	Scalability	<p>Assesses the highest workloads under which the system will still meet the performance Deals with the measure of the system's response time under differentload conditions requirements.</p>

5.PROJECT DESIGN

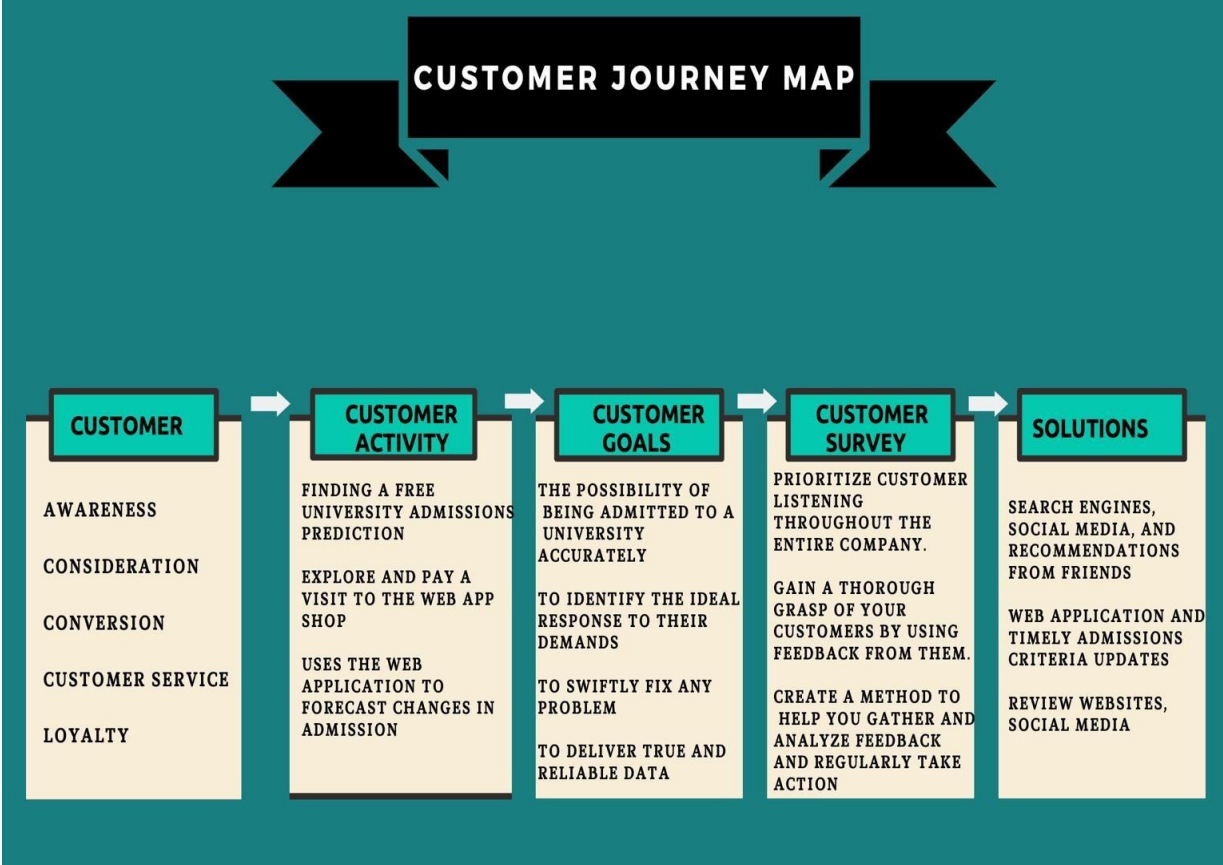
Data Flow Diagram



Solution & Technical Architecture



User Stories

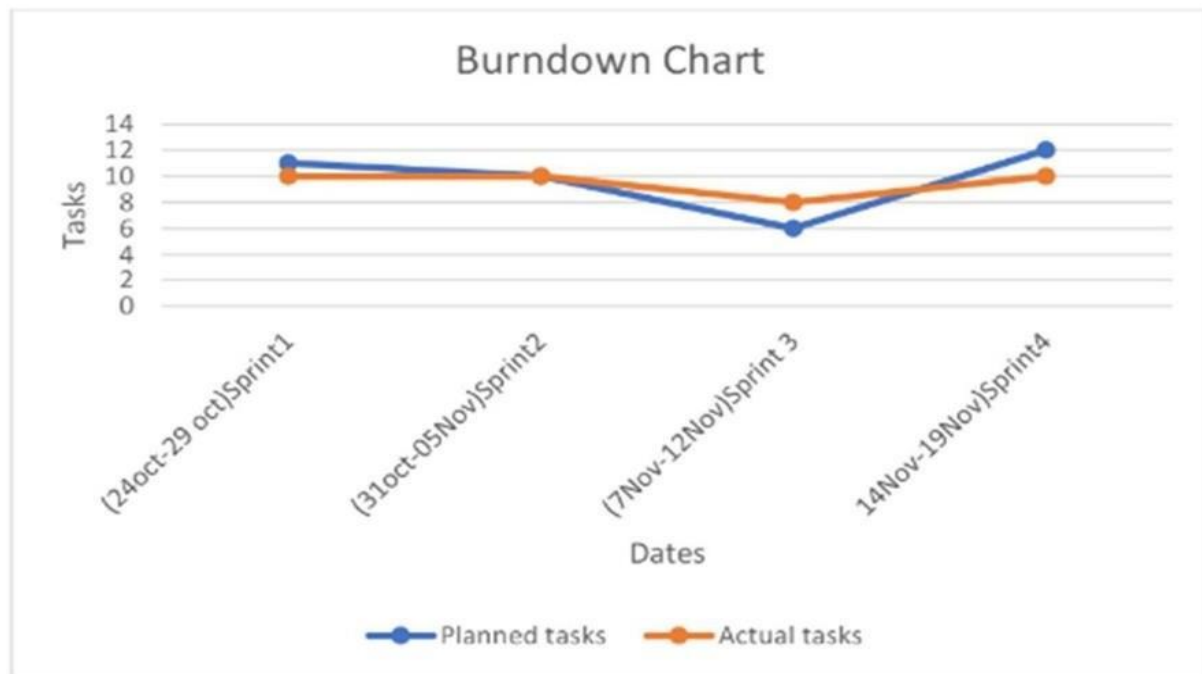


Debugging can sometimes be a relatively straight forward process, or rather finding out what you must debug can be. Since so many parts of the admissions system are integrated into one another, if an error occurs on one page, it may be a display error, for example; it may be the information is not correctly 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.

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

Burndown Chart:

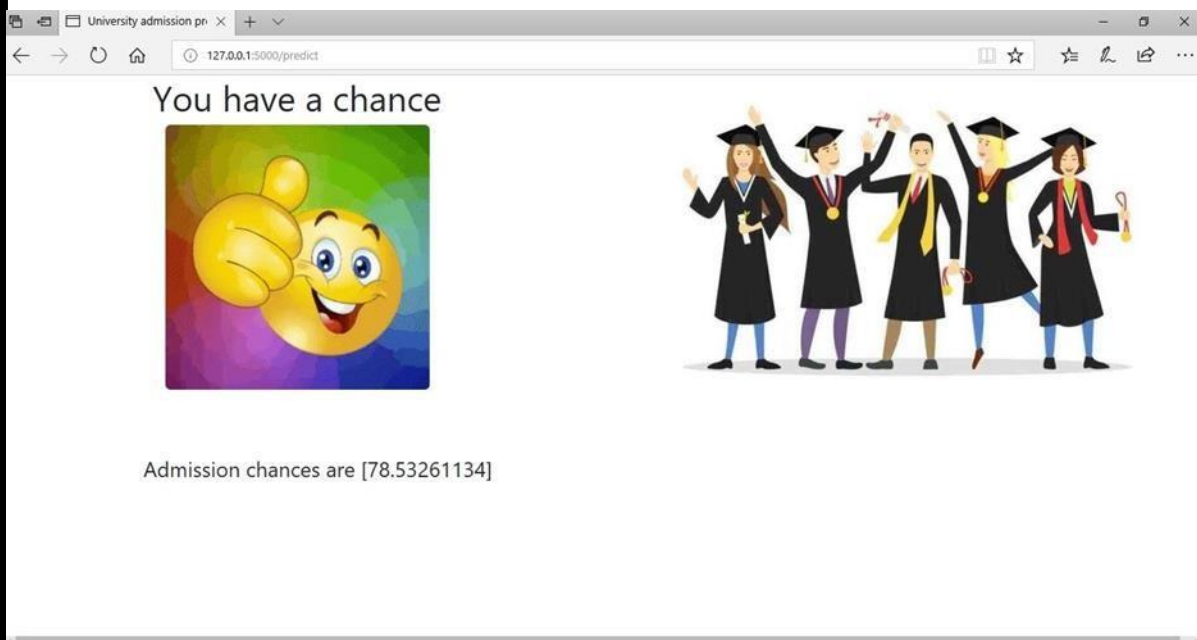
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.CODING & SOLUTIONING

Feature 1

The new feature will predict the chances in the admission of the university. The feature was designed in the html code connected with app.py as the backend.



Source Code :

```
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-
      gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNI/vI1Bx"
      crossorigin="anonymous">
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>University admission prediction System </title>
<link rel="icon" type="image/jpg" href="https://png.pngtree.com/png-
      vector/20200211/outline/pngtree-graduation-caps-vector-convocation-students-png-
      image_2144286.jpg">
<!--
<style>.center {
/*display: block;
margin-left:auto;
```

```

margin-right: auto;*/
width: 230px;
height: 161px;
padding-left:161px;
padding-top: 230px;
}
.body {
background-image: url('nochance_output.png');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}
</style>--></head>
<body >
<div class="row" >
<div class="col-md-6"><div class="col-md-12">
<h1 style="text-align: center;">You have a chance</h1>

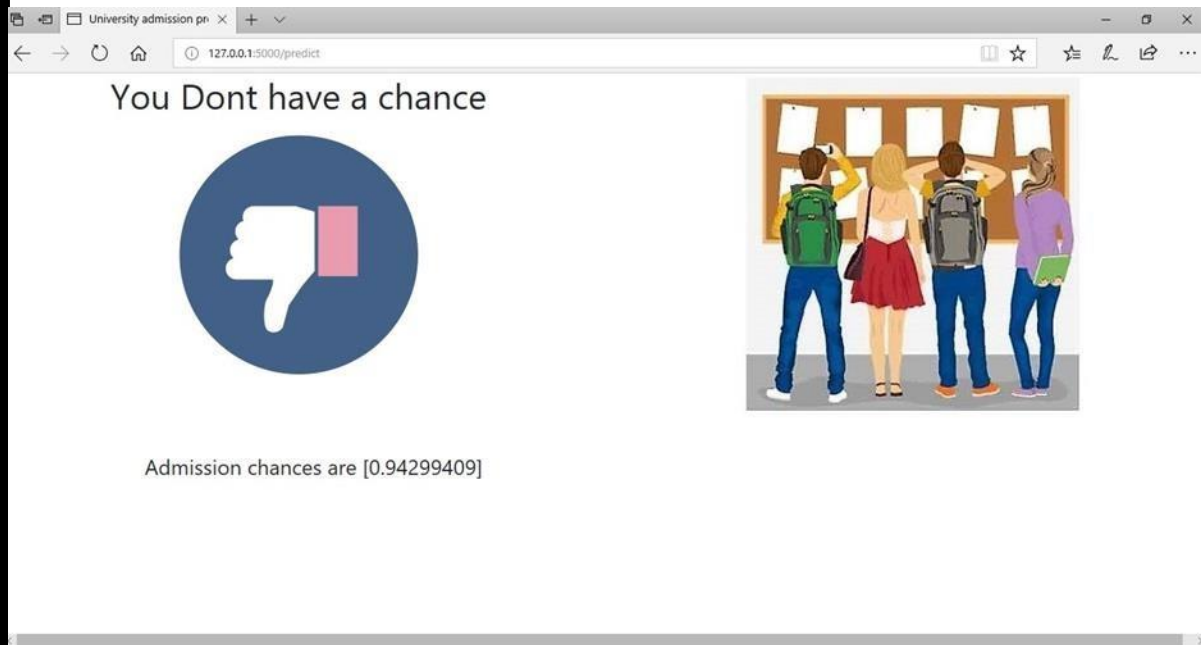
<h4 style="width: 750px;
height: 161px;
padding-left:161px;
padding-top: 75px;">{{prediction_text}}</h4></div></div><div class="col-md-6"><div class="col-md-
12">
</div></div></div>

</body>
</html>

```

Feature 2

The new feature will predict the low chances in the admission of the university. The feature was designed in the html code connected with app.py as the backend.



Source Code:

```
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNI/vI1Bx"
crossorigin="anonymous">
<meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>University admission prediction System </title>
<link rel="icon" type="image/jpg" href="https://png.pngtree.com/png-
vector/20200211/ourmid/pngtree-graduation-caps-vector-convocation-students-png-
image_2144286.jpg">
<!--
<style>.center {
/*display: block;
margin-left:auto;
margin-right: auto;*/
width: 230px;
```



```

height: 161px;
padding-left: 161px;
padding-top: 230px;
}
body {
background-image: url('nochance_output.png');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}
</style>--></head>
<body >
<div class="row" >
<div class="col-md-6"><div class="col-md-12">
<h1 style="text-align: center;">You Dont have a chance</h1>

<h4 style="width: 750px;
height: 161px;
padding-left: 161px;
padding-top: 75px;">{ { prediction_text } }</h4></div></div><div class="col-md-6"><div class="col-
md-12">
</div></div></div></body>
</html>

```

Database Schema

The database used here in this project was Admission_Predict.csv. The sample screenshot of the database are,

481 lines (481 sloc) | 12.6 KB

Raw Blame    

🔍 Search this file...

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1									
2	1	337	118	4	4.5	4.5	9.65	1	0.92
3	2	324	107	4	4	4.5	8.87	1	0.76
4	3	316	104	3	3	3.5	8	1	0.72
5	4	322	110	3	3.5	2.5	8.67	1	0.8
6	5	314	103	2	2	3	8.21	0	0.65
7	6	330	115	5	4.5	3	9.34	1	0.9
8	7	321	109	3	3	4	8.2	1	0.75
9	8	308	101	2	3	4	7.9	0	0.68
10	9	302	102	1	2	1.5	8	0	0.5
11	10	323	108	3	3.5	3	8.6	0	0.45
12	11	325	106	3	3.5	4	8.4	1	0.52
13	12	327	111	4	4	4.5	9	1	0.84
14	13	328	112	4	4	4.5	9.1	1	0.78
15	14	307	109	3	4	3	8	1	0.62
16	15	311	104	3	3.5	2	8.2	1	0.61
17	16	314	105	3	3.5	2.5	8.3	0	0.54
18	17	317	107	3	4	3	8.7	0	0.66
19	18	319	106	3	4	3	8	1	0.65
20	19	318	110	3	4	3	8.8	0	0.63
21	20	303	102	3	3.5	3	8.5	0	0.62
22	21	312	107	3	3	2	7.9	1	0.64
23	22	325	114	4	3	2	8.4	0	0.7
24	23	328	116	5	5	5	9.5	1	0.94
25	24	334	119	5	5	4.5	9.7	1	0.95
26	25	336	119	5	4	3.5	9.8	1	0.97

7. TESTING

Test Cases

Test Case Analysis

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

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

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**

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

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 Reproduc ed	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

Test Case Analysis

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

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. RESULTS

9.1 Performance Metrics

Measure the performance using Metrics

```
pd.crosstab(Y_Test,y_predict)
```

col_0	0	1	2
Sex			
0	108	29	112
1	33	223	35
2	123	52	121

```
print(classification_report(Y_Test,y_predict))
```

	precision	recall	f1-score	support
0	0.41	0.43	0.42	249
1	0.73	0.77	0.75	291
2	0.45	0.41	0.43	296
accuracy			0.54	836
macro avg	0.53	0.54	0.53	836
weighted avg	0.54	0.54	0.54	836

Measuring the performance using 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
```

9. ADVANTAGES &DISADVANTAGES

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.
4. Avoids data redundancy and inconsistency.
5. Very user-friendly.
6. Easy accessibility of data.

Dis-Advantages

7. Required active internet connection.
8. System will provide inaccurate results if data entered incorrectly.

10. 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 isnot 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 thefeatures 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.

11.FUTURE SCOPE

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

1. This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
2. 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.

12.APPENDIX

Source Code

PYTHON CODE

Uploading the pythoncode ,

IMPORT STATEMENTS

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```


LOAD THE DATASET

In [2]:

```
import os, types
import pandas as pd
from botocore.client
import Configimport
ibm_boto3

def __iter_

(self): return 0

#@hidden_cli

# The followingcode accesses a file in your IBM Cloud ObjectStorage. It includesyour
credentials.

# You might want to remove those credentials beforeyou share
the notebook.cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='T6FhPnWEPrnR91XKAfpiopbqTZ8j-
    gbLtjakMGexd6v0',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-
    storage.appdomain.cloud')
bucket = 'university-donotdelete-pr-
    ljujvyruwxy5c'object_key =
'Admission_Predict.csv'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas acceptsbody as file-like object
if not hasattr(body, "__iter__"): body.__iter__= types.MethodType(__iter_, body )
data =
pd.read_csv(
body)
```

```
data.head()
```

Out[2]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance ofAdmit
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 [3]:

```
data.drop(["Serial No."], axis=1, inplace=True)
```

In [4]:

```
data.describe()
```

Out[4]:

	GRE Score	TOEFL Score	Universit y Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.0000 00	400.0000 00	400.0000 00	400.0000 00	400.0000 00	400.0000 00	400.0000 00	400.0000 00
mean	316.8075 00	107.4100 00	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.47364 6	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	290.0000 00	92.00000 0	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	308.0000 00	103.0000 00	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.0000 00	107.0000 00	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.0000 00	112.0000 00	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	340.0000 00	120.0000 00	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

In [5]:

```
data.info()
```

Out[5]:

```
<class  
'pandas.core.frame.DataFrame'  
>RangeIndex: 400  
entries, 0 to 399
```

Data columns(total 8 columns):

#	Column	Non-Null Count	Dtype
---	--------	----------------	-------

- | | | | |
|----|-------------------|--------------|---------|
| 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 float64dtypes:

float64(4), int64(4)

memory usage: 25.1 KB

In [6]:

```
data.isnull().sum()
```

Out[6]:

GRE Score	0
TOEFL Score	0
University Rating	0
SOP	0
LOR	0
CGPA	0
Research	0

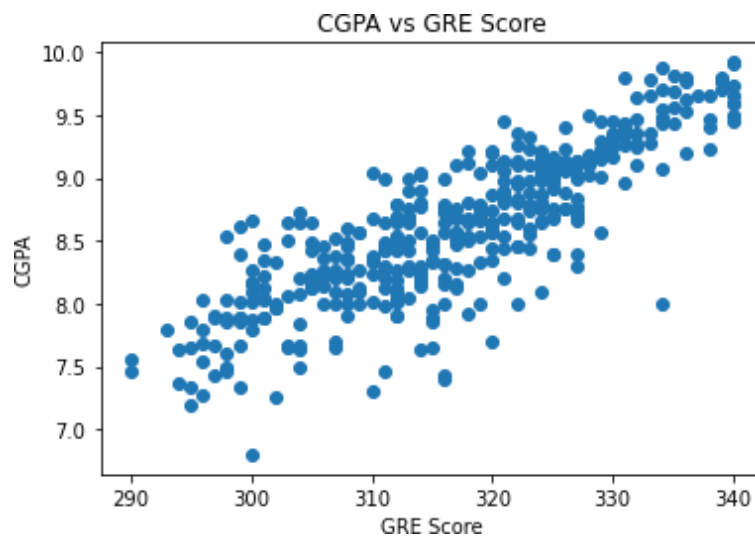
Chance of Admit 0
dtype: int64

VISUALIZATION

In [7]:

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

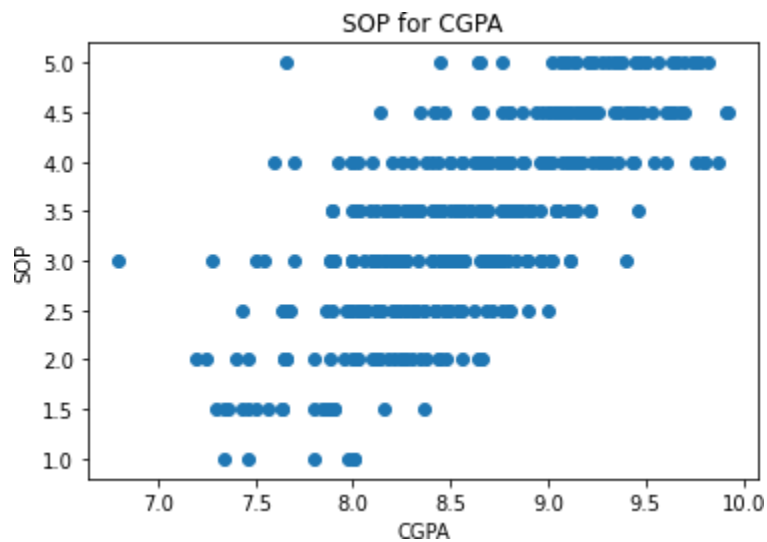
Out [7]:



In [8]:

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

Out [9]:



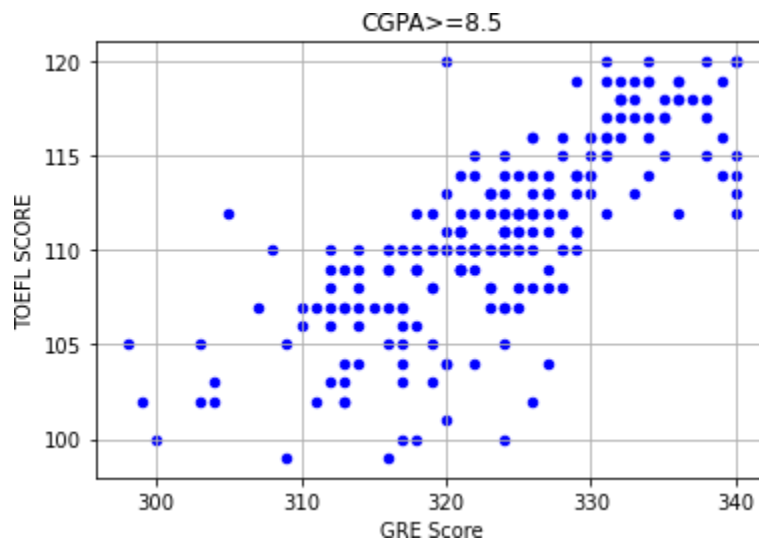
In [9]:

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

plt.xlabel("GREScore")
plt.ylabel("TOEFL SCORE")
plt.title("CGPA>=8.5")

plt.grid(True)
plt.show()
```

Out [9]:



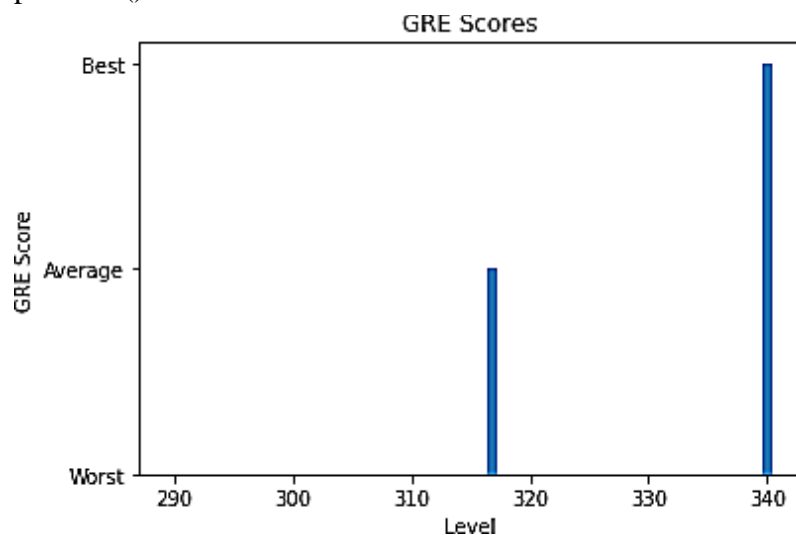
In [10]:

```
data["GRE Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))  
plt.title("GRE Scores")  
plt.xlabel("GRE Score")  
plt.ylabel("Frequency")  
plt.show()
```



In[11]:

```
p = np.array([data["TOEFL Score"].min(),data["TOEFL Score"].mean(),data["TOEFL Score"].max()])
r = ["Worst","Average","Best"]
plt.bar(p,r)
plt.title("TOEFL Scores")
plt.xlabel("Level")
plt.ylabel("TOEFL Score")
plt.show()
```



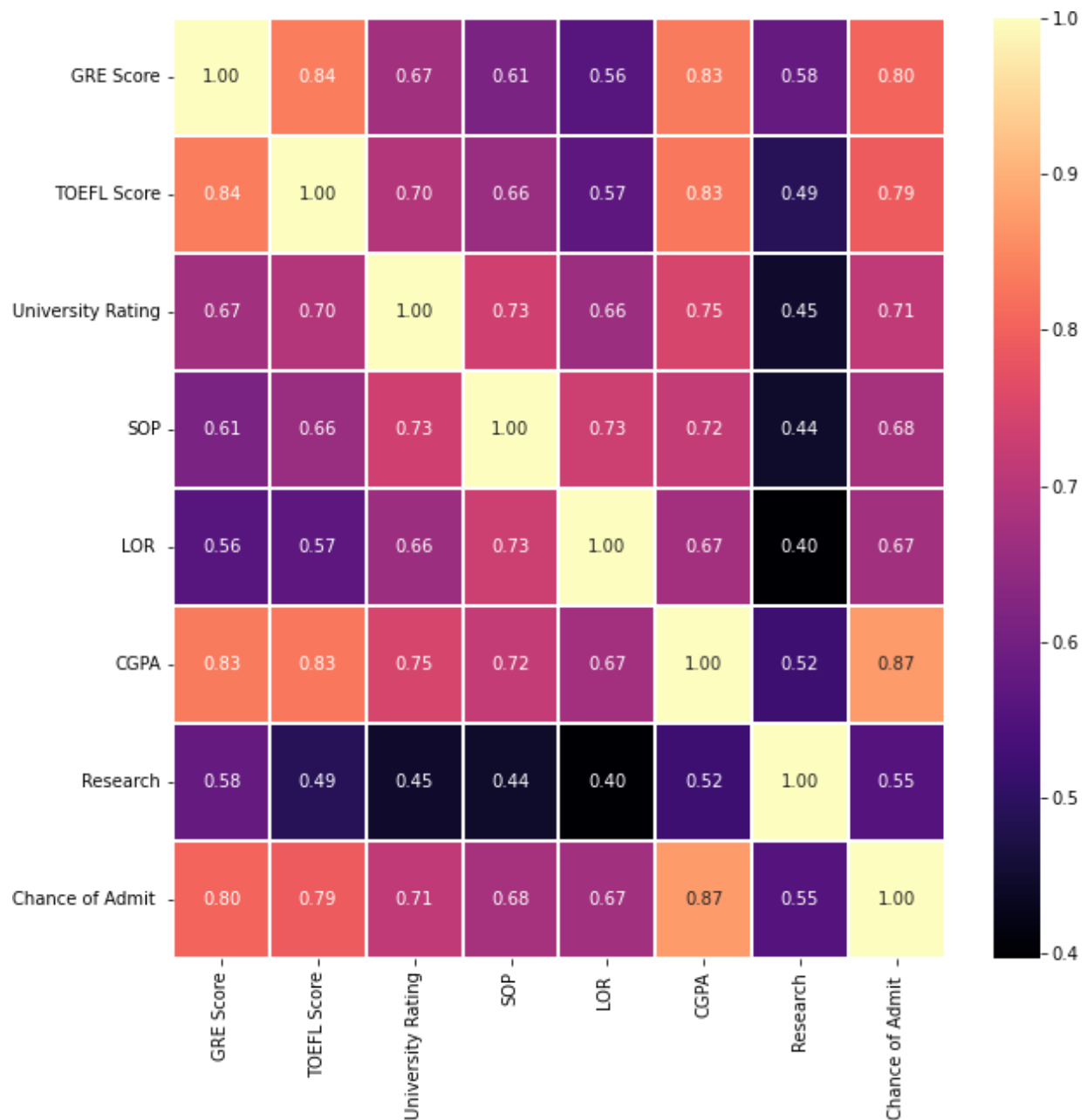
In[13]:

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

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

```
plt.show()
```

Out[13]:

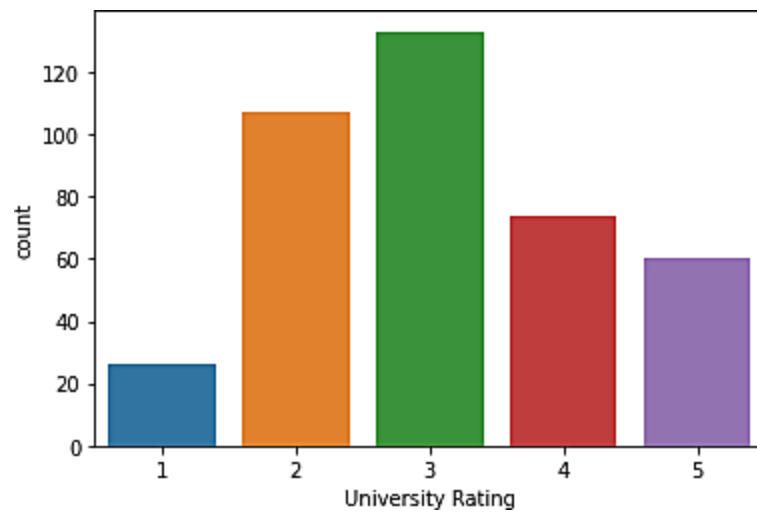


In[14]:

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

Out[14]:

<AxesSubplot:xlabel='University Rating',ylabel='count'>

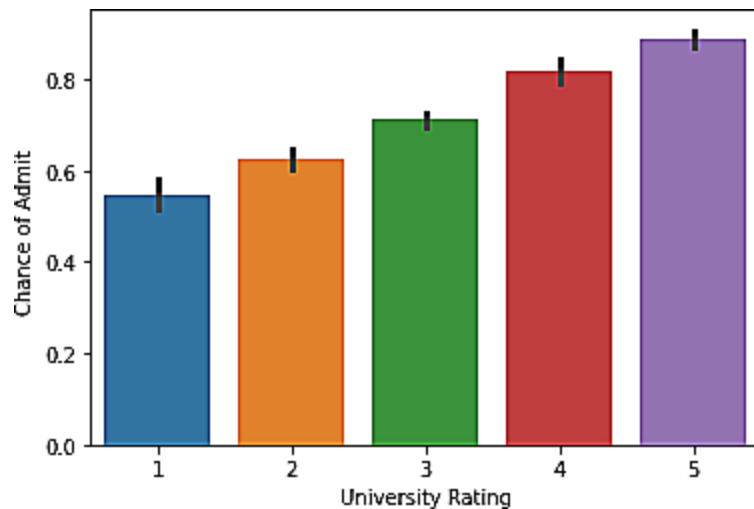


In[15]:

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

Out[15]:

<AxesSubplot:xlabel='University Rating',ylabel='Chance of Admit '>



TRAINING AND TESTING SPLIT

In [16]:

```
X=data.drop(['Chance of Admit '],axis=1)
#input data_sety=data['Chance of Admit ']
#output labels
```

In [17]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15)
```

MODELING AND TRAINING

In [18]:

```
from sklearn.ensemble import
GradientBoostingRegressor
gbr = GradientBoostingRegressor()
```

```
rgr.fit(X_train,y_train)
```

Out[18]:

```
GradientBoostingRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [19]:

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

Out[19]:

```
0.7214021715194154
```

In [20]:

```
y_predict=rgr.predict(X_test)
```

In [21]:

```
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
```

```
import numpy as np
```

```
print('Mean Absolute Error:', mean_absolute_error(y_test,
```

```
y_predict)) print('Mean Squared Error:',
```

```
mean_squared_error(y_test, y_predict))
```

```
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test,
```

```
y_predict))) Mean Absolute Error: 0.061115035673946834
```

```
Mean Squared Error:
```

0.007194293635482686 Root Mean
SquaredError: 0.08481918200196631

In [22]:

```
y_train =  
(y_train>0  
.5)y_test =  
(y_test>0.  
5)
```

In [23]:

```
from sklearn.linear_model._logistic  
import LogisticRegressionlore =  
LogisticRegression(random_state=0,  
max_iter=1000)  
lr = lore.fit(X_train, y_train)
```

In [24]:

```
y_pred = lr.predict(X_test)
```

In [25]:

```
from sklearn.metrics import accuracy_score, recall_score, roc_auc_score,  
confusion_matrixprint('Accuracy Score:', accuracy_score(y_test, y_pred))  
print('Recall Score:', recall_score(y_test,  
y_pred))print('ROC AUC  
Score:',roc_auc_score(y_test, y_pred))  
print('Confussion Matrix:\n', confusion_matrix(y_test, y_pred))
```

Out [25]:

```
Accuracy Score: 0.9166666666666666  
Recall Score: 1.0  
ROC AUC Score: 0.7222222222222222  
Confusion  
Matrix  
x: [[ 4  
      5]  
     [ 0 51]]
```

SAVING THE MODEL

In [26]:

```
import pickle
```

In [27]:

```
pickle.dump(lr, open("university.pkl", "wb")) #logistic regression model
```

HOSTING THE MODEL

In [28]:

```
import pickle
```

In [29]:

```
lr = pickle.load(open("university.pkl", "rb")) #logistic regression model
```

In [30]:

```
pip install-U ibm-watson-machine-learning
```

Out [30]:

Requirement already satisfied: ibm-watson-machine-learning in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.255)

Collecting ibm-watson-machine-learning

Downloading ibm_watson_machine_learning-1.0.256-py3-none-any.whl (1.8 MB)

 1.8 MB 22.1 MB/s eta

0:00:01 Requirement already satisfied: packaging in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (21.3)

Requirement already satisfied: importlib-metadata in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (4.8.2)

Requirement already satisfied: urllib3 in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)

Requirement already satisfied: lmond in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (0.3.3)

Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.3.4)

Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: requests in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)

Requirement already satisfied: tabulate in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)

Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)

Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (0.10.0)

Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.8.2)

Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (2021.3)

Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (1.20.3)

Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (1.15.0)

Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (2.0.4) Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (3.3)

Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm-watson-machine-learning) (3.6.0)

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm-

watson-machine-learning) (3.0.4) Installing collected packages: ibm-watson-machine-learning

Attempting uninstall: ibm-watson-machine-learning

Found existing installation: ibm-watson-machine-learning 1.0.255
Uninstalling ibm-watson-machine-learning-1.0.255:

Successfully uninstalled ibm-watson-machine-learning-1.0.255
Successfully installed ibm-watson-machine-learning-1.0.256

Note: you may need to restart the kernel to use updated packages.

In [31]:

```
from ibm_watson_machine_learning
import APIClient import json
```

In [32]:

```
uml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "Ct8qx91QEDmW9Xw2viaF7uGxI-jHrk0ogjKl17b_L4-K"
}
```

```
client = APIClient(uml_credentials)
```

In [33]:

```
def guid_from_space_name(client,
    space_name):
    space = client.spaces.get_details()
    idr = []
    for i in space['resources']
```

```
]:  
idr.append(i['meta  
data']['id'])  
return idr
```

In [34]:

```
space_uid = guid_from_space_name(client,  
“uni admit”)print(space_uid[0])  
424b35d3-9ffd-4890-a149-d7642a3f640c
```

In [35]:

```
client.set.default_space(space_uid[0])
```

Out[35]:

```
'SUCCESS'
```

In [36]:

```
client.software_specifications.list()
```

NAME	ASSET_ID	TYPE
default_py3.6	0062b8c9-8b7d-44a0-a9b9-46c416adcbd9	
base kernel-spark3.2-scala2.12		
020d69ce-7ac1-5e68-ac1a-31189867356a	base	pytorch-onnx_1.3-
py3.7-edt	069ea134-3346-5748-b513-	
49120e15d288	basescikit-learn_0.20-py3.6	
09c5a1d0-9c1e-4473-a344-eb7b665ff687	base	spark-mllib_3.0-

scala_2.12 09f4cff0-90a7-5899-b9ed-
1ef348aebdee base pytorch-onnx_rt22.1-py3.9
0b848dd4-e681-5599-be41-b5f6fccc6471 base ai-function_0.1-
py3.6 0cdb0f1e-5376-4f4d-92dd-
da3b69aa9bda base shiny-r3.6 0e6e79df-875e-4f24-8ae9-
62dcc2148306 base
tensorflow_2.4-py3.7-horovod 1092590a-307d-563d-9b62-
4eb7d64b3f22 basepytorch_1.1-py3.6
10ac12d6-6b30-4ccd-8392-3e922c096a92 base tensorflow_1.15-
py3.6-ddl 111e41b3-de2d-
5422-a4d6-bf776828c4b7 base runtime-22.1-py3.9
12b83a17-24d8-5082-900f-0ab31fbfd3cb base scikit-learn_0.22-
py3.6 154010fa-5b3b-
4ac1-82af-4d5ee5abbc85 base default_r3.6 1b70aec3-ab34-4b87-
8aa0-a4a3c8296a36 base
pytorch-onnx_1.3-py3.6 1bc6029a-cc97-56da-b8e0-
39c3880dbbe7 base kernel-spark3.3-r3.6
1c9e5454-f216-59dd-a20e-474a5cdf5988 base pytorch-
onnx_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f
basetensorflow_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-
3fbdf1665666 base spark-mllib_3.2
20047f72-0a98-58c7-9ff5-a77b012eb8f5 base tensorflow_2.4-
py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base
runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-
da66306ce658 base do_py3.8 295addb5-9ef9-547e-9bf4-
92ae3563e720 base
autoai-ts_3.8-py3.8 2aa0c932-798f-5ae9-abd6-
15e0c2402fb5 base tensorflow_1.15-py3.6
2b73a275-7cbf-420b-a912-eae7f436e0bc base

kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-
482c8368839a base pytorch_1.2-py3.6
2c8ef57d-2687-4b7d-acce-01f94976dac1 base spark-mllib_2.3
2e51f700-bca0-4b0d-88dc-5c6791338875 base pytorch-onnx_1.1-

py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base
spark-mllib_3.0-py37
36507ebe-8770-55ba-ab2a-eafe787600e9 base spark-mllib_2.4
390d21f8-e58b-4fac-9c55-d7ceda621326 base xgboost_0.82-
py3.6 39e31acd-5f30-41dc-ae44-
60233c80306e base pytorch-onnx_1.2-py3.6-edt
40589d0e-7019-4e28-8daa-fb03b6f4fe12 base default_r36py38
41c247d3-45f8-5a71-b065-8580229facf0 base autoai-ts_rt22.1-
py3.9 4269d26e-07ba-5d40-
8f66-2d495b0c71f7 base autoai-obm_3.0
42b92e18-d9ab-567f-988a-4240ba1ed5f7 base pmml-3.0_4.3
493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base spark-mllib_2.4-
r_3.6 49403dff-92e9-4c87-a3d7-
a42d0021c095 base xgboost_0.90-py3.6
4ff8d6c2-1343-4c18-85e1-689c965304d3 base pytorch-onnx_1.1-
py3.6 50f95b2a-bc16-43bb-
bc94-b0bed208c60b base autoai-ts_3.9-py3.8
52c57136-80fa-572e-8728-a5e7cbb42cde base spark-mllib_2.4-
scala_2.11 55a70f99-7320-4be5-
9fb9-9edb5a443af5 base spark-mllib_3.05c1b0ca2-4977- 5c2e-
9439-ffd44ea8ffe9 base
autoai-obm_2.0 5c2e37fa-80b8-5e77-840f-
d912469614ee base spss-modeler_18.1
5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base cuda-py3.8
5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base autoai-kb_3.1-
py3.7 632d4b22-10aa-5180-88f0-
f52dfb6444d7 base pytorch-onnx_1.7-py3.8
634d3cdc-b562-5bf9-a2d4-ea90a478456b base spark-mllib_2.3-
r_3.6 6586b9e3-ccd6-4f92-900f-
0f8cb2bd6f0c base tensorflow_2.4-py3.7
65e171d7-72d1-55d9-8ebb-f813d620c9bb base spss-
modeler_18.2 687eddc9-028a-
4117-b9dd-e57b36f1efa5 base

Note: Only first 50 records were displayed. To display more use 'limit' parameter.

In [37]:

```
import
sklearn
sklearn.___
version_
```

Out[37]:

```
'1.0.2'
```

In [38]:

```
MODEL_NAME =
'university'
DEPLOYMENT_N
AME = 'uni'
DEMO_MODEL =
lr
```

In [39]:

```
software_spec_uid = client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

In [40]:

```
model_props = {
    client.repository.ModelMetaNames.NAME: MODEL_NAME,
```

```
client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0 ',  
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid  
}
```

In [41]:

```
model_details =  
    client.repository.store_model(  
        model= DEMO_MODEL,  
        meta_props =  
        model_props,  
        training_data =  
        X_train,  
        training_target  
        = y_train  
    )  
model_details
```

Out[41]:

```
{'entity':  
  {'hybrid_pipeline_software_spec  
s': [], 'label_column': 'Chance of  
Admit ',  
'schemas': {'input': [{'fields': [{'name': 'GRE Score', 'type': 'int64'},  
    {'name': 'TOEFL Score', 'type': 'int64'},  
    {'name': 'University Rating', 'type': 'int64'},  
    {'name': 'SOP', 'type': 'float64'},  
    {'name': 'LOR ', 'type': 'float64'},  
    {'name': 'CGPA', 'type': 'float64'},  
  
    {'name': 'Research',
```

```
'type':'int64'}], 'id': '1',  
'type': 'struct'}],  
'output': []}]}
```

In [42]:

```
model_id =  
client.repository.get_model_id(model_details)  
model_id
```

Out[42]:

```
'85e3a479-6eb2-4104-9dd7-3c208aac046b'
```

In [43]:

```
deployment_props = {  
    client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,  
    client.deployments.ConfigurationMetaNames.ONLINE: {}  
}
```

```
deployment =  
    client.deployments.create(  
        artifact_uid =  
        model_id,  
        meta_props = deployment_props  
    )
```

Out [43]:

```
#####  
#####
```

Synchronous deployment creation for uid: '85e3a479-6eb2-4104-9dd7-3c208aac046b' started

```
#####  
#####
```

initializing

Note: online_url is deprecated and will be removed in a future release. Use

serving_urls instead.

Successfully finished deployment creation, deployment_uid='b76c9fcd-b8a6-4fa6-bca7-ced79cce8301'

HTML CODES

Uploading Html codes

Chance.html

```
<html>  
<head>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet"  
integrity="sha384-gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNI/vI1Bx"  
crossorigin="anonymous">  
<meta charset="utf-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <title>University admission prediction System </title>
```



```

<link rel="icon" type="image/jpg" href="https://png.pngtree.com/png-
vector/20200211/ourmid/pngtree-graduation-caps-vector-convocation-students-png-
image_2144286.jpg">
<!--
<style >.center {
/*display: block;
margin-left:auto;
margin-right: auto;*/
width: 230px;
height: 161px;
padding-left:161px;
padding-top: 230px;
}
.body {
background-image: url('nochance_output.png');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}
</style>--></head>
<body >
<div class="row" >
<div class="col-md-6"><div class="col-md-12">
<h1 style="text-align: center;">You have a chance</h1>

<h4 style="width: 750px;
height: 161px;
padding-left:161px;

padding-top: 75px;">{{ prediction_text }}</h4></div></div><div

class="col-md-6"><div class="col-md-12">

</div></div></div>

</body>
</html>

```

Nochance.html

```
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
gH2yIJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNI/vI1Bx"
crossorigin="anonymous">
<meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>University admission prediction System </title>
<link rel="icon" type="image/jpg" href="https://png.pngtree.com/png-
vector/20200211/ourmid/pngtree-graduation-caps-vector-convocation-students-png-
image_2144286.jpg">
  <!--
  <style >.center {
/*display: block;
margin-left:auto;
margin-right: auto;*/
width: 230px;
height: 161px;
padding-left:161px;
padding-top: 230px;
}
.body {
background-image: url('nochance_output.png');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: cover;
}
</style>--></head>
<body >
<div class="row" >
<div class="col-md-6"><div class="col-md-12">
  <h1 style="text-align: center;">You Dont have a chance</h1>

<h4 style="width: 750px;
height: 161px;
padding-left:161px;
padding-top: 75px;">{{ prediction_text }}</h4></div></div><div class="col-md-6"><div
class="col-md-12">
```

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

Index.html

```
<!DOCTYPE html>
<html>
<head>
  <SCRIPT language=Javascript>
    <!--
    function check(e, value) {
      //Check Charater
      var unicode = e.charCode ? e.charCode : e.keyCode;
      if (value.indexOf(".") != -1)
        if (unicode == 46) return false;
      if (unicode != 8)
        if ((unicode < 48 || unicode > 57) && unicode != 46) return false;
    }
    //-->
  </SCRIPT>
<title>University Admit Eligibility Predictor </title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/jpg" href="https://png.pngtree.com/png-
vector/20200211/ourmid/pngtree-graduation-caps-vector-convocation-students-png-
image_2144286.jpg">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">

<style>
body {
  font-family: Arial, Helvetica, sans-serif;
}

* {
  box-sizing: border-box;
}

/* style the container */
.container {
  position: relative;
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px 0 30px 0;
}
```

```

/* style inputs and link buttons */
input,
.btn {
    width: 100%;
    padding: 12px;
    border: none;
    border-radius: 4px;
    margin: 5px 0;
    opacity: 0.85;
    display: inline-block;
    font-size: 17px;
    line-height: 20px;
    text-decoration: none; /* remove underline from anchors */
}

input:hover,
.btn:hover {
    opacity: 1;
}

/* style the submit button */
input[type=submit] {
    content: " ";
    background-color: #4CAF50;
    /* background: linear-gradient(#00ccff,#d400d4);
    animation: animate 4s linear infinite;
    inset: 4px;
    border-radius: 20px;*/
    color: white;
    cursor: pointer;
}
@keyframes animate
{0% {
    transform: rotate(0deg);
}
100% {
    transform: rotate(360deg);
}
}
input[type=submit]:hover {
    background-color: #45a049;
}

/* Two-column layout */
.col {

```

```
float: left;
width: 50%;
margin: auto;
padding: 0 50px;
margin-top: 6px;
}
```

```
/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}
```

```
/* vertical line */
.vl {
  position: absolute;
  left: 50%;
  transform: translate(-50%);
  border: 2px solid #ddd;
  height: 490px;
}
```

```
/* text inside the vertical line */
.vl-innertext {
  position: absolute;
  top: 50%;
  transform: translate(-50%, -50%);
  background-color: #f1f1f1;
  border: 1px solid #ccc;
  border-radius: 50%;
  padding: 8px 10px;
}
```

```
/* hide some text on medium and large screens */
.hide-md-lg {
  display: none;
}
```

```
/* bottom container */
.bottom-container {
  text-align: center;
  background-color: #666;
  border-radius: 0px 0px 4px 4px;
}
```

```
/* Responsive layout - when the screen is less than 650px wide, make the two columns stack on top of
each other instead of next to each other */
@media screen and (max-width: 650px) {
```

```

.col {
  width: 100%;
  margin-top: 0;
}
/* hide the vertical line */
.vl {
  display: none;
}
/* show the hidden text on small screens */
.hide-md-lg {
  display: block;
  text-align: center;
}
}
</style>
</head>
<body style="background-image:
url('https://i.pinimg.com/564x/84/f1/f1/84f1f1cc416291c8c44e95c7c888d781.jpg'); background-
position: center;
background-repeat: no-repeat;
background-size: cover,contain; height: -15px;">

<div class="container" style="background-image:
url('https://i.pinimg.com/564x/84/f1/f1/84f1f1cc416291c8c44e95c7c888d781.jpg'); background-
position: center;
background-repeat: no-repeat;
background-size: cover,contain; ">
  <form action="/predict" method="post">
    <div class="row" style="padding:0px ; height:fit-content">
      <span class="align-middle"><p class="text-danger"><h1 class="align-middle" style="height:
min-content; color: #000080;">UNIVERSITY ADMIT ELIGIBILITY
PREDICTOR</h1></p></span>
      <div class="vl">
        <span class="vl-innertext"></span>
      </div>

    <div class="col">
      <p><b>In this project,We build a linear regression model and used it to predict the chance of
admission into a particular university based on the input given by the student.</p></b>
      <h3>Instructions for Input Features</h3>
      <ul>
        <li>GRE Score (out of 340)</li>
        <li>TOEFL Score (out of 120)</li>
        <li>University Rating (out of 5)</li>
        <li>Statment of Purpose {SOP} (out of 5)</li>
        <li>Letter of Recommendation {LOP} Strength (out of 5)</li>
        <li>Undergraduate CGPA (out of 10)</li>
        <li>Research Experience (Either 0 or 1)</li>

```


Login.html

```
<!DOCTYPE html>
<html>
<head>
    <title>University admission prediction System</title>
<link rel="icon" type="image/jpg" href="https://png.pngtree.com/png-
vector/20200211/ourmid/pngtree-graduation-caps-vector-convocation-students-png-
image_2144286.jpg"><link rel="stylesheet" type="text/css" href="{{ url_for('static', filename =
'main.css') }}">
</head>
<body>

    <!-- Icon -->
    <!-- Login Form -->
    <form action="/result" method="POST">

<div class="wrapper fadeInDown">
    <div id="formContent">
        <!-- Tabs Titles -->
        <h2 class="active"> Sign In </h2>
        <a href="{{ url_for('signup') }}"><h2 class="inactive underlineHover">Sign Up </h2></a> <div
class="fadeIn first">
            
        </div>

        <input type="email" id="login" class="fadeIn second" name="email" placeholder="email">
        <input type="password" id="password" class="fadeIn third" name="pass"
placeholder="password">
        <input type="submit" class="fadeIn fourth" value="Log In">
        {{error}}
    </form>
<!--
        Will add in future
    <div id="formFooter">
        <a class="underlineHover" href="#">Forgot Password?</a>
    </div>
-->
</div>
</div>
</body>
</html>
```


Sign Up.html

```
<!DOCTYPE html>
<html>
<head>
    <title>University admission prediction System</title>
<link rel="icon" type="image/jpg" href="https://png.pngtree.com/png-
vector/20200211/ourmid/pngtree-graduation-caps-vector-convocation-students-png-
image_2144286.jpg"><link rel="stylesheet" type="text/css" href="{{ url_for('static', filename =
'main.css') }}">
</head>
<body>
<div class="wrapper fadeInDown">
    <div id="formContent">
        <!-- Tabs Titles -->
        <a href="/"><h2 class="inactive underlineHover"> Sign In </h2></a>
        <h2 class="active">Sign Up </h2>

        <!-- Icon -->
        <div class="fadeIn first">
            
        </div>

        <!-- Login Form -->
        <form action="/register" method="POST">
            <input type="text" id="login" class="fadeIn second" name="name" placeholder="name">
                <input type="email" id="login" class="fadeIn second" name="email" placeholder="email">
                <input type="password" id="password" class="fadeIn third" name="pass"
placeholder="password">
                <input type="password" id="cpass" class="fadeIn second" name="cpass" placeholder="Confirm
Password">
                <input type="submit" class="fadeIn fourth" value="Sign Up">
                {{ cerror }}
            </form>

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

Welcome.css

```
body{
    margin: 0;
    background-color: #212121;
}

main{
    width: 60vw;
    padding-top: 100px;
    padding-bottom: 100px;
    text-align: center;
    margin-top: 20vh;
    background-color: #e0e0e0;
    margin-left: 20vw;
}

.name{
    font-size: 300%;
    font-family: monospace;
}

.email{
    font-size: 150%;
    color: #404040;
    font-family: monospace;
}
```

Main.css

```
@import url('https://fonts.googleapis.com/css?family=Poppins');

/* BASIC */

html {
    background-color: #56baed;
}

body {
    font-family: "Poppins", sans-serif;
    height: 100vh;
}

a {
```

```
color: #92badd;
display:inline-block;
text-decoration: none;
font-weight: 400;
}
```

```
h2 {
text-align: center;
font-size: 16px;
font-weight: 600;
text-transform: uppercase;
display:inline-block;
margin: 40px 8px 10px 8px;
color: #cccccc;
}
```

```
/* STRUCTURE */
```

```
.wrapper {
display: flex;
align-items: center;
flex-direction: column;
justify-content: center;
width: 100%;
min-height: 100%;
padding: 20px;
}
```

```
#formContent {
-webkit-border-radius: 10px 10px 10px 10px;
border-radius: 10px 10px 10px 10px;
background: #fff;
padding: 30px;
width: 90%;
max-width: 450px;
position: relative;
padding: 0px;
-webkit-box-shadow: 0 30px 60px 0 rgba(0,0,0,0.3);
box-shadow: 0 30px 60px 0 rgba(0,0,0,0.3);
text-align: center;
}
```

```
#formFooter {
background-color: #f6f6f6;
border-top: 1px solid #dce8f1;
```

```
padding: 25px;
text-align: center;
-webkit-border-radius: 0 0 10px 10px;
border-radius: 0 0 10px 10px;
}
```

```
/* TABS */
```

```
h2.inactive {
  color: #cccccc;
}
```

```
h2.active {
  color: #0d0d0d;
  border-bottom: 2px solid #5fbae9;
}
```

```
/* FORM TYPOGRAPHY*/
```

```
input[type=button], input[type=submit], input[type=reset] {
  background-color: #56baed;
  border: none;
  color: white;
  padding: 15px 80px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  text-transform: uppercase;
  font-size: 13px;
  width: 85%;
  -webkit-box-shadow: 0 10px 30px 0 rgba(95,186,233,0.4);
  box-shadow: 0 10px 30px 0 rgba(95,186,233,0.4);
  -webkit-border-radius: 5px 5px 5px 5px;
  border-radius: 5px 5px 5px 5px;
  margin: 5px 20px 40px 20px;
  -webkit-transition: all 0.3s ease-in-out;
  -moz-transition: all 0.3s ease-in-out;
  -ms-transition: all 0.3s ease-in-out;
  -o-transition: all 0.3s ease-in-out;
  transition: all 0.3s ease-in-out;
}
```

```
input[type=button]:hover, input[type=submit]:hover, input[type=reset]:hover {  
    background-color: #39ace7;  
}
```

```
input[type=button]:active, input[type=submit]:active, input[type=reset]:active {  
    -moz-transform: scale(0.95);  
    -webkit-transform: scale(0.95);  
    -o-transform: scale(0.95);  
    -ms-transform: scale(0.95);  
    transform: scale(0.95);  
  
}
```

```
input[type=email], input[type=password], input[type=text] {  
    background-color: #f6f6f6;  
    border: none;  
    color: #0d0d0d;  
    padding: 15px 32px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
    font-size: 16px;  
    margin: 5px;  
    width: 85%;  
    border: 2px solid #f6f6f6;  
    -webkit-transition: all 0.5s ease-in-out;  
    -moz-transition: all 0.5s ease-in-out;  
    -ms-transition: all 0.5s ease-in-out;  
    -o-transition: all 0.5s ease-in-out;  
    transition: all 0.5s ease-in-out;  
    -webkit-border-radius: 5px 5px 5px 5px;  
    border-radius: 5px 5px 5px 5px;  
}
```

```
input[type=email]:focus, input[type=password]:focus, input[type=text]:focus {  
    background-color: #fff;  
    border-bottom: 2px solid #5fb9e9;  
}
```

```
input[type=email]:placeholder, input[type=password]:placeholder, input[type=text]:placeholder {  
    color: #cccccc;  
}
```

```

/* ANIMATIONS */

/* Simple CSS3 Fade-in-down Animation */
.fadeInDown {
  -webkit-animation-name: fadeInDown;
  animation-name: fadeInDown;
  -webkit-animation-duration: 1s;
  animation-duration: 1s;
  -webkit-animation-fill-mode: both;
  animation-fill-mode: both;
}

@-webkit-keyframes fadeInDown {
  0% {

    opacity: 0;
    -webkit-transform: translate3d(0, -100%, 0);
    transform: translate3d(0, -100%, 0);
  }
  100% {
    opacity: 1;
    -webkit-transform: none;
    transform: none;
  }
}

@keyframes fadeInDown {
  0% {
    opacity: 0;
    -webkit-transform: translate3d(0, -100%, 0);
    transform: translate3d(0, -100%, 0);
  }
  100% {
    opacity: 1;
    -webkit-transform: none;
    transform: none;
  }
}

/* Simple CSS3 Fade-in Animation */
@-webkit-keyframes fadeIn { from { opacity:0; } to { opacity:1; } }
@-moz-keyframes fadeIn { from { opacity:0; } to { opacity:1; } }
@keyframes fadeIn { from { opacity:0; } to { opacity:1; } }

```

```
.fadeIn {  
  opacity:0;  
  -webkit-animation:fadeIn ease-in 1;  
  -moz-animation:fadeIn ease-in 1;  
  animation:fadeIn ease-in 1;  
  
  -webkit-animation-fill-mode:forwards;  
  -moz-animation-fill-mode:forwards;  
  animation-fill-mode:forwards;  
  
  -webkit-animation-duration:1s;  
  -moz-animation-duration:1s;  
  animation-duration:1s;  
}
```

```
.fadeIn.first {  
  -webkit-animation-delay: 0.4s;  
  -moz-animation-delay: 0.4s;
```

```
  animation-delay: 0.4s;  
}
```

```
.fadeIn.second {  
  -webkit-animation-delay: 0.6s;  
  -moz-animation-delay: 0.6s;  
  animation-delay: 0.6s;  
}
```

```
.fadeIn.third {  
  -webkit-animation-delay: 0.8s;  
  -moz-animation-delay: 0.8s;  
  animation-delay: 0.8s;  
}
```

```
.fadeIn.fourth {  
  -webkit-animation-delay: 1s;  
  -moz-animation-delay: 1s;  
  animation-delay: 1s;  
}
```

```
/* Simple CSS3 Fade-in Animation */  
.underlineHover:after {  
  display: block;
```

```
left: 0;
bottom: -10px;
width: 0;
height: 2px;
background-color: #56baed;
content: "";
transition: width 0.2s;
}
```

```
.underlineHover:hover {
  color: #0d0d0d;
}
```

```
.underlineHover:hover:after{
  width: 100%;
}
```

```
/* OTHERS */
```

```
*:focus {
  outline: none;
}
```

```
#icon {
  width:30%;
  margin-top: 2vh;
  margin-bottom: 2vh;
}
```

```
* {
  box-sizing: border-box;
}
```


REQUIRED IMAGES IN THE HTML CODES :







PYTHON CODE:

app.py

```
import pandas as pd

from flask import Flask, request, jsonify, render_template

import pickle

import pyrebase

app = Flask(__name__)

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

config = {

    "apiKey": "AIzaSyCpueysTCJlJW8t3-r-gV4NOPrZY2VZbA",

    "authDomain": "university-admit-predictor.firebaseio.com",

    "databaseURL": "https://university-admit-predictor-default-rtdb.firebaseio.com",

    "projectId": "university-admit-predictor",

    "storageBucket": "university-admit-predictor.appspot.com",

    "messagingSenderId": "471033088541",

    "appId": "1:471033088541:web:2d05bfca07ad298f2cd4f4",

    "measurementId": "G-DCEHDHRG4K"

}

#initialize firebase

firebase = pyrebase.initialize_app(config)

auth = firebase.auth()
```

```
@app.route("/register", methods = ["POST", "GET"])
```

```
def regiter():
```

```
    if request.method == "POST":
```

```
        global name    #Only if data has been posted
```

```
        name=request.form.get('name')
```

```
        email=request.form.get('email')
```

```
        password=request.form.get('pass')
```

```
        cpassword=request.form.get('cpass')
```

```
    try:
```

```
        if(password==cpassword):
```

```
            user=auth.create_user_with_email_and_password(email,password)
```

```
            return render_template("login.html")
```

```
        #return render_template("login.html")
```

```
    except:
```

```
        #return "Your passwaord could not be same Please Try Again"
```

```
        return render_template("signup.html",cerror="Your password could not be same or Already  
Exist account")
```

```
#Login
```

```
@app.route("/")
```

```

def login():

    return render_template("login.html")

@app.route("/signup")

def signup():

    return render_template("signup.html")


@app.route('/welcome')

def home():

    return render_template('index.html')

@app.route("/result", methods = ["POST", "GET"])

def result():

    """if('user' in session):

        return "Hi {}".format(session["user"])"""

    if request.method == "POST":      #Only if data has been posted

        email=request.form.get('email')

        password=request.form.get('pass')

        try:

            #Try signing in the user with the given information

            user = auth.sign_in_with_email_and_password(email, password)

            return render_template("index.html")

        except:

            return render_template("login.html",error="Your Email and Password Invalid Please Try
login again or SignUp")

```

```
@app.route('/predict', methods=['GET','post'])

def predict():

    GRE_Score = int(request.form['GRE Score'])

    TOEFL_Score = int(request.form['TOEFL Score'])

    University_Rating = int(request.form['University Rating'])

    SOP = float(request.form['SOP'])

    LOR = float(request.form['LOR'])

    CGPA = float(request.form['CGPA'])

    Research = int(request.form['Research'])

    final_features = pd.DataFrame([[GRE_Score, TOEFL_Score, University_Rating, SOP, LOR,
    CGPA, Research]])

    predict = model.predict(final_features)

    output = predict[0]

    if(output>50):

        return render_template('chance.html', prediction_text='Admission chances are
        {}'.format(output))

    else:

        return render_template('nochance.html', prediction_text='Admission chances are
        {}'.format(output))
```

```
if __name__ == "__main__":
```

```
    app.run(debug=True)
```

GITHUB LINK:

<https://github.com/IBM-EPBL/IBM-Project-18124-1659679726.git>

