UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

PROJECT REPORT

SUBMITTED BY

J.DHARANI(730319106009)

A.INDHUMALINI(730319106016)

N.PAVITHRA(730319106032)

J.PRIYADHARSHINI(730319106037)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING in

ELECTRONICS AND COMMUNICATION ENGINEERING

at

BUILDERS ENGINEERING COLLEGE

NATHAKADAIYUR-638108

NOV-2022

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

PROJECT OVERVIEW

The project is implemented using a Machine-Learning model that predicts whether the user is eligible for an admission in the selected rated universities with provided details such as marks and others. The algorithm works in such a way that when the user provides the details such as (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research) the percentage of chance of admit is displayed. The user is provided with a UI (Web based application) in which the user can enter the details mentioned above for prediction. The main advantage of this is that the user can avoid long process of having to check the eligibility of a university admission by himself and make use of this application to predict the eligibility / chance of admit.

PURPOSE

University and College research being one part of the university application process is itself an arduous and lengthy task. This issue is a big problem for students have not been solved till now. There are recognized sites which filter the best universities and colleges based on the location, tuition fees, major and degree but none of them have use machine learning algorithm to solve the issue.

Hence, we have done this research project to solve that issue to some extent with the use of data mining techniques

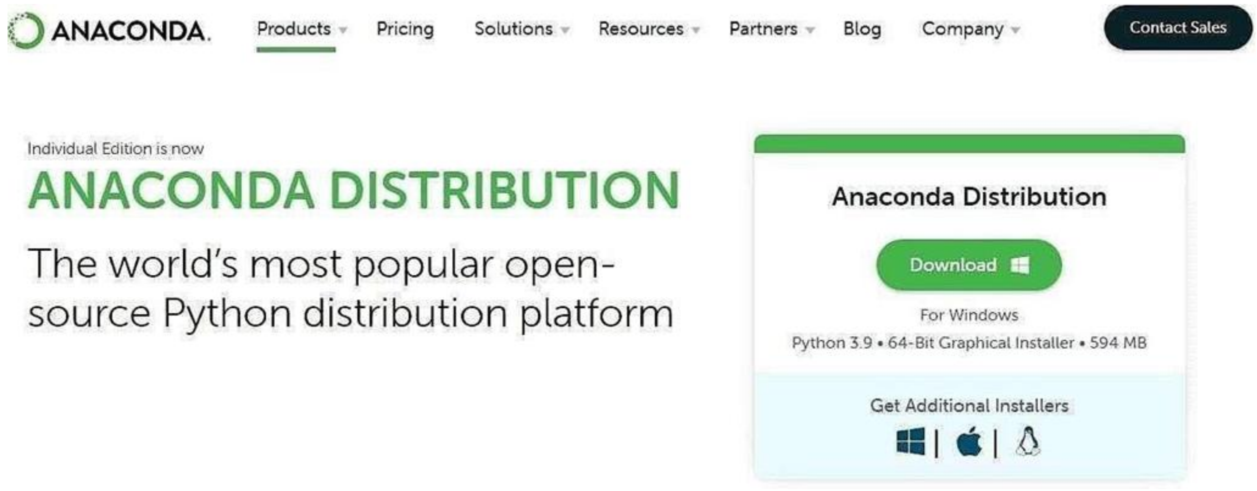
PRE-REQUISTIES:

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:

STEP1: Download and Anaconda



Individual Edition is now

ANACONDA DISTRIBUTION

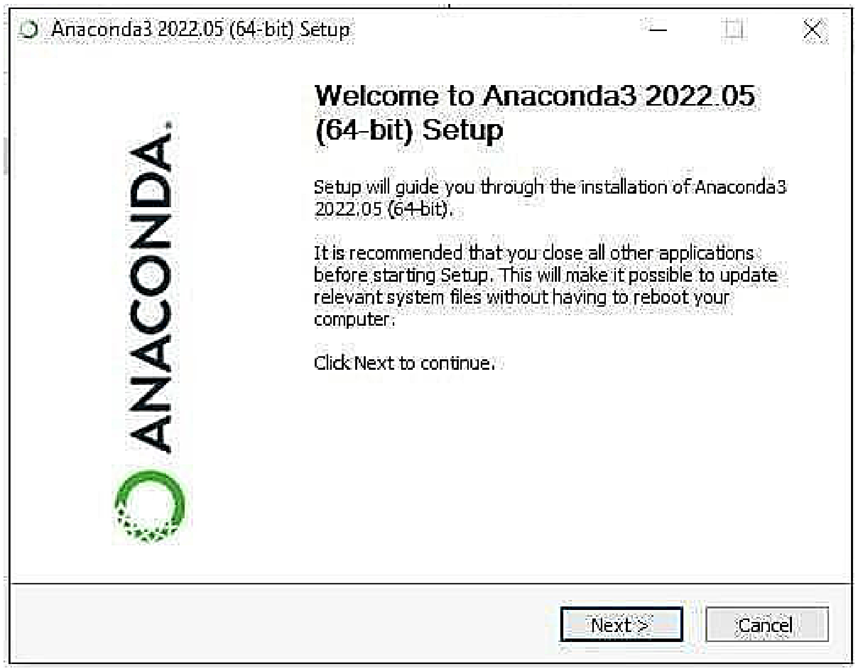
The world's most popular open-source Python distribution platform

Anaconda Distribution

For Windows

Python 3.9 • 64-Bit Graphical Installer • 594 MB

STEP2: Install the Anaconda



Anaconda3 2022.05 (64-bit) Setup

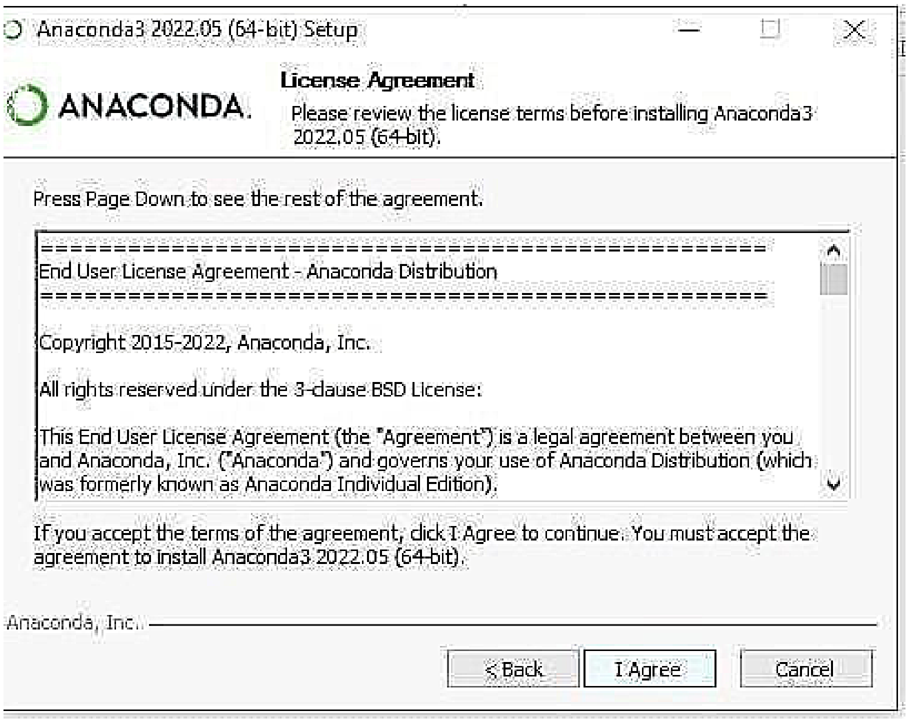
Welcome to Anaconda3 2022.05 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2022.05 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

STEP3: Click I Agree



Anaconda3 2022.05 (64-bit) Setup

License Agreement

Please review the license terms before installing Anaconda3 2022.05 (64-bit).

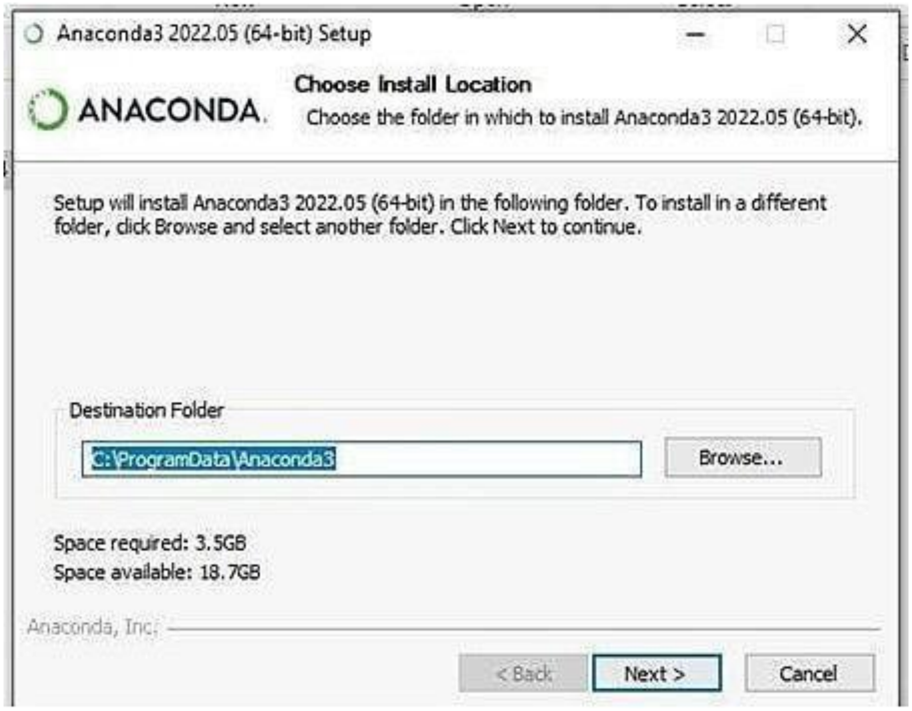
Press Page Down to see the rest of the agreement.

```
==================================================
End User License Agreement - Anaconda Distribution
==================================================

Copyright 2015-2022, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you
and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution (which
was formerly known as Anaconda Individual Edition).
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2022.05 (64-bit).

Anaconda, Inc.

STEP4: Choose the Installation Location


Anaconda3 2022.05 (64-bit) Setup

Choose Install Location

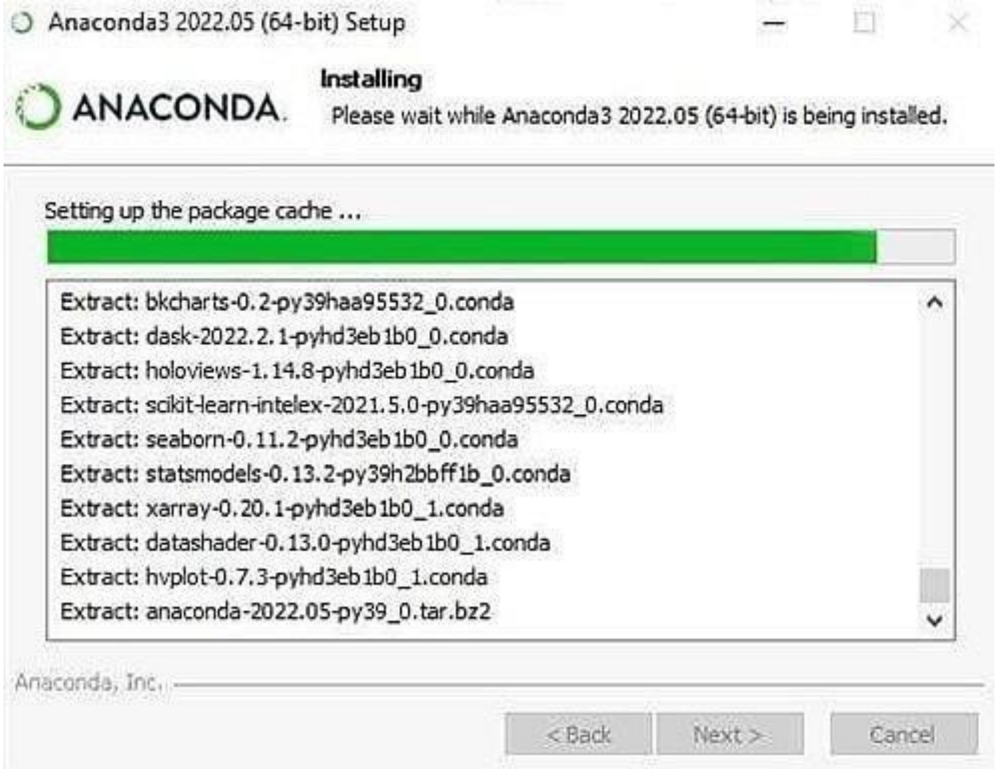
Choose the folder in which to install Anaconda3 2022.05 (64-bit).

Setup will install Anaconda3 2022.05 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\ProgramData\Anaconda3

Space required: 3.5GB
Space available: 18.7GB

STEP5: Installing the Requiring packages


Anaconda3 2022.05 (64-bit) Setup

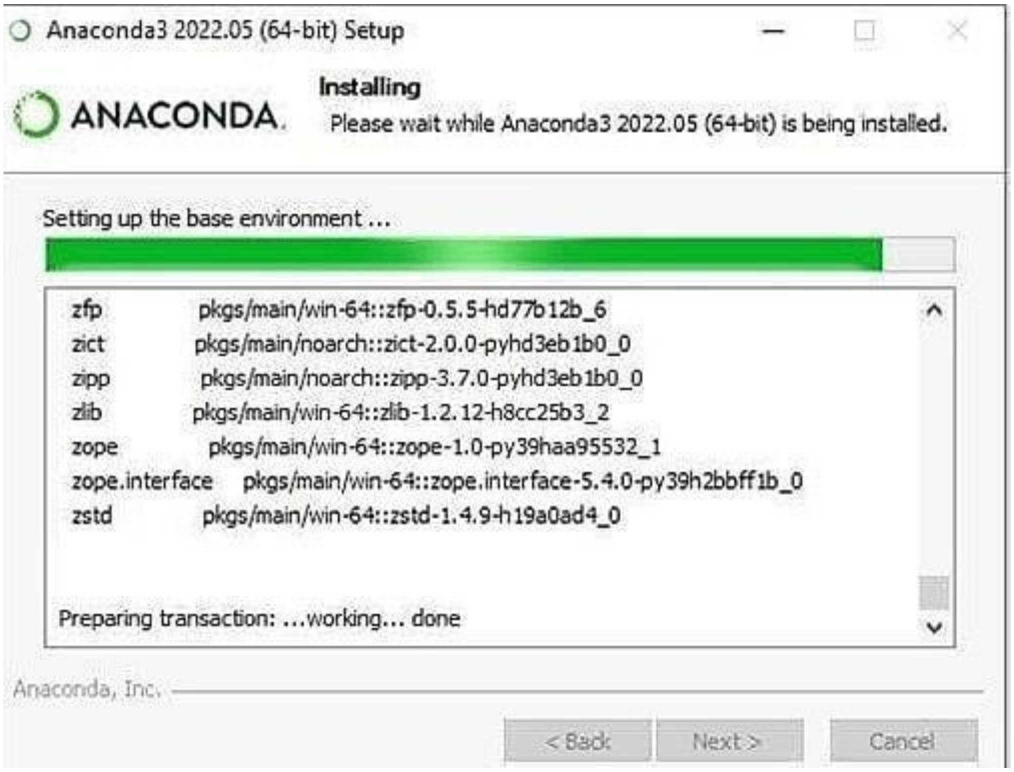
Installing

Please wait while Anaconda3 2022.05 (64-bit) is being installed.

Setting up the package cache ...

```
Extract: bkcharts-0.2-py39haa95532_0.conda
Extract: dask-2022.2.1-pyhd3eb1b0_0.conda
Extract: holoviews-1.14.8-pyhd3eb1b0_0.conda
Extract: scikit-learn-intelex-2021.5.0-py39haa95532_0.conda
Extract: seaborn-0.11.2-pyhd3eb1b0_0.conda
Extract: statsmodels-0.13.2-py39h2bbff1b_0.conda
Extract: xarray-0.20.1-pyhd3eb1b0_1.conda
Extract: datashader-0.13.0-pyhd3eb1b0_1.conda
Extract: hvplot-0.7.3-pyhd3eb1b0_1.conda
Extract: anaconda-2022.05-py39_0.tar.bz2
```

STEP6: Setting up the base environment



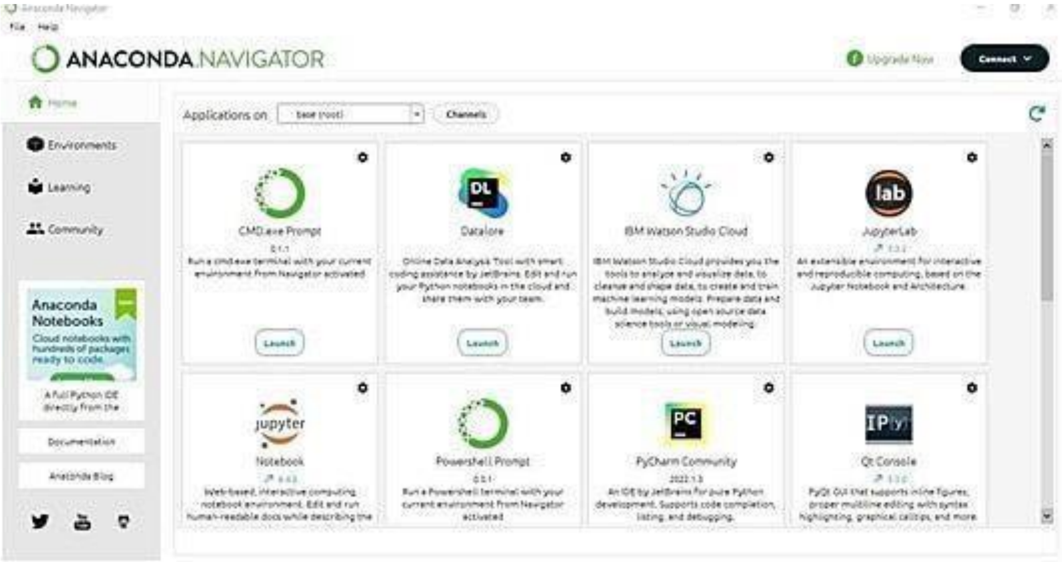
Installing

Please wait while Anaconda3 2022.05 (64-bit) is being installed.

Setting up the base environment ...

```
zfp             pkgs/main/win-64::zfp-0.5.5-hd77b12b_6
zict            pkgs/main/noarch::zict-2.0.0-pyhd3eb1b0_0
zipp            pkgs/main/noarch::zipp-3.7.0-pyhd3eb1b0_0
zlib            pkgs/main/win-64::zlib-1.2.12-h8cc25b3_2
zope            pkgs/main/win-64::zope-1.0-py39haa95532_1
zope.interface  pkgs/main/win-64::zope.interface-5.4.0-py39h2bbff1b_0
zstd            pkgs/main/win-64::zstd-1.4.9-h19a0ad4_0


Preparing transaction: ...working... done
```

STEP7: Successfully Installed and check the Anaconda Navigator working or not



CMD.exe Prompt 0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the

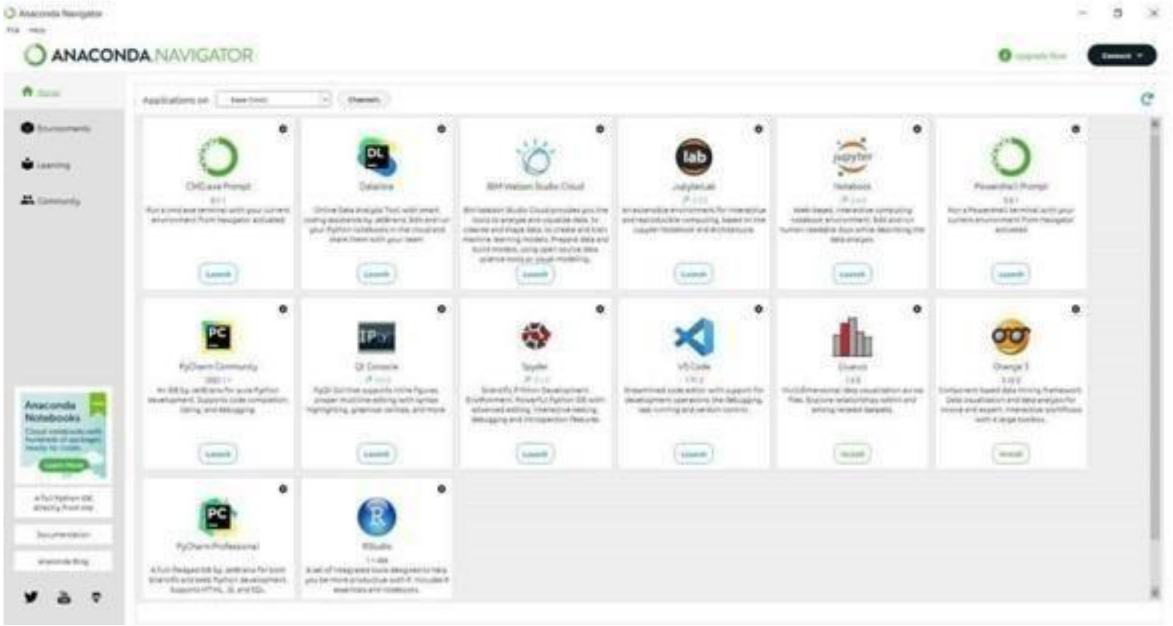
Powershell Prompt 0.0.1
Run a Powershell terminal with your current environment from Navigator activated

PyCharm Community 2022.1.3
An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.

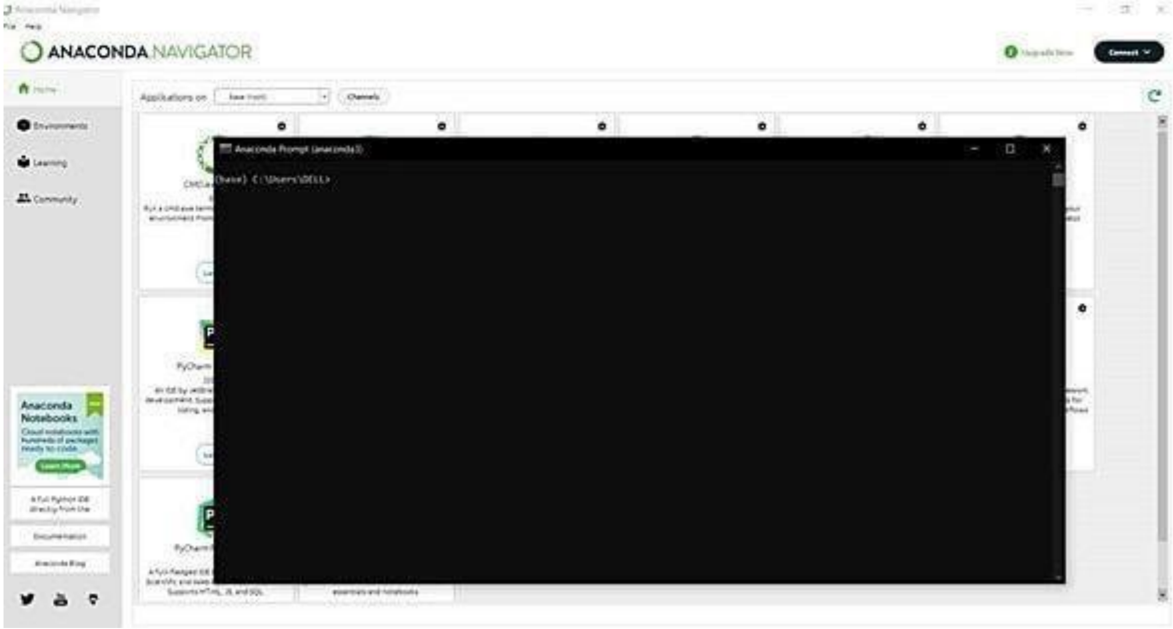
Qt Console
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Python packages installation:

Step1: Open the anaconda navigator in the start menu



Step 2: Open the CMD.exe prompt



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

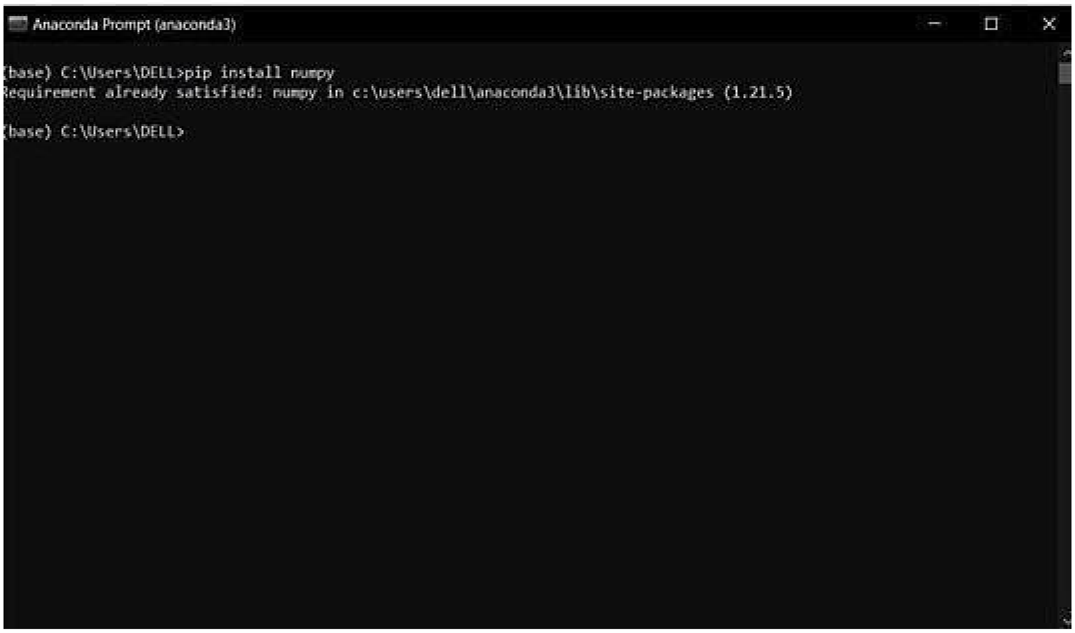
Step3: InstalltheNUMPYpackage

To enter the numpy package enter the command in the CMD.exe

Command:**Pipinstallnumpy**

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.



```
(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>
```

Step4: Installthepandaspackage.

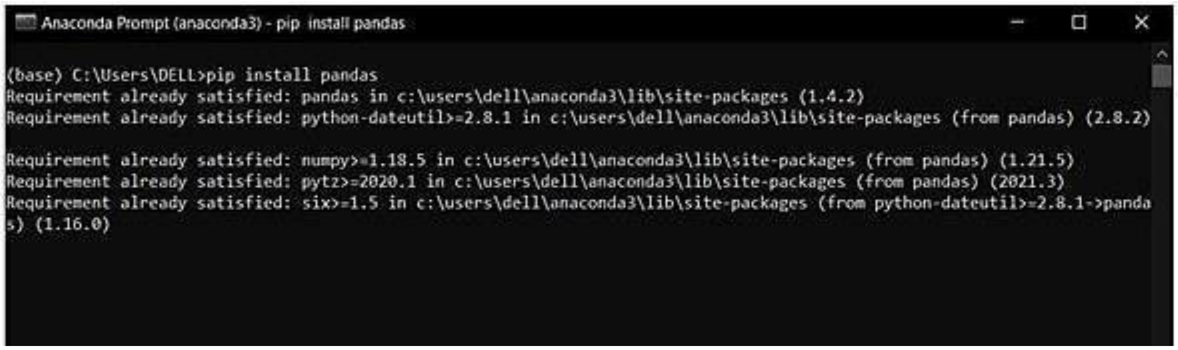
To Enter the Pandas Package enter the command in the CMD.exe Command:

Pipinstallpandas

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.



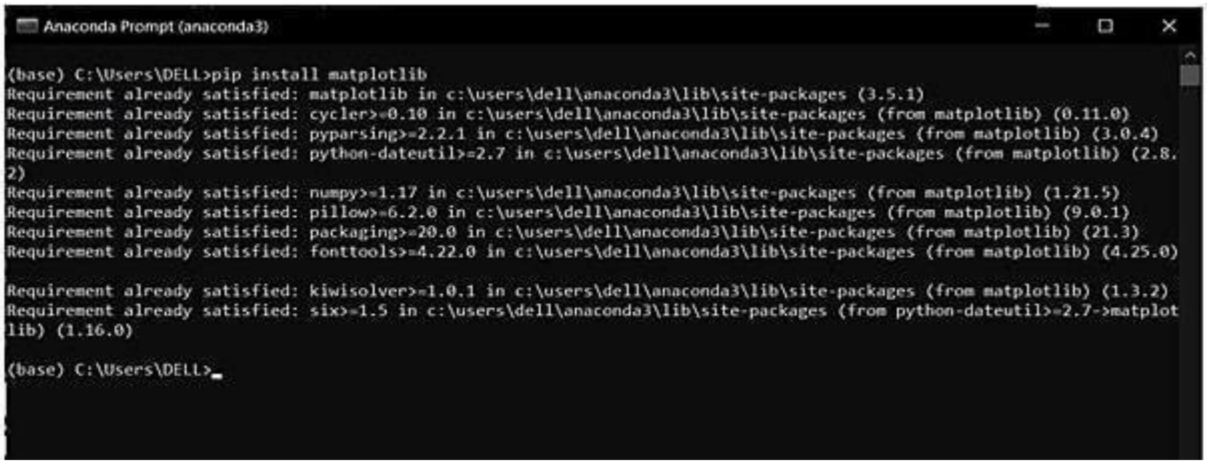
```
(base) C:\Users\DELL>pip install pandas
Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2.8.2)

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->panda
s) (1.16.0)
```

Step5: InstalltheMatplotlibpackage.

To enter the Matplotlib package enter the command in the CMD.exe Command: **PipinstallMatplotlib**

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.



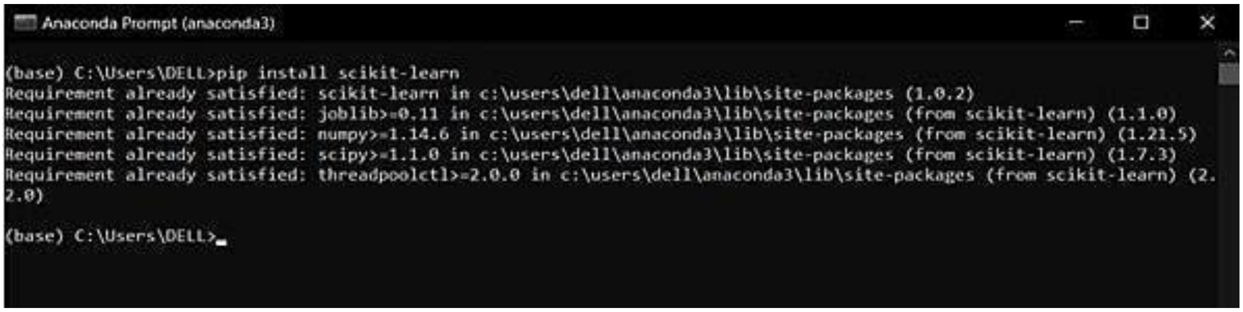
```
(base) C:\Users\DELL>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib\site-packages (3.5.1)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 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->matplot
lib) (1.16.0)

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

Step6: Install the Scikit-learn package.

To enter the Scikit-learn package enter the command in the CMD.exeCommand:**Pip installScikit-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.



```
(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>
```

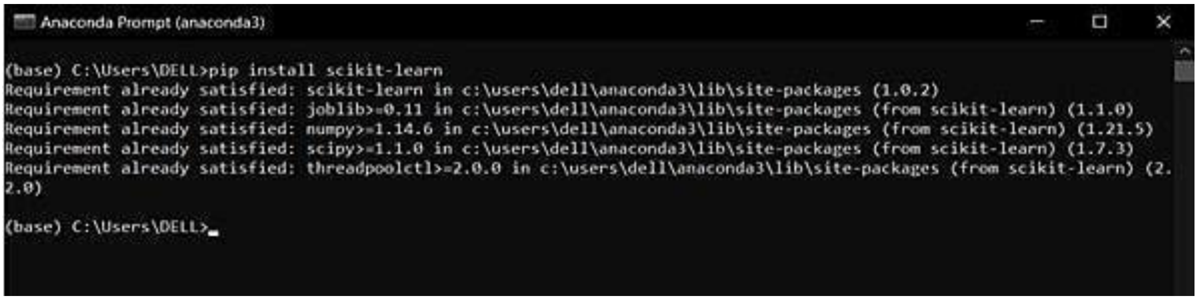
Step7: Install the Flask package.

To enter the Flask package enter the command in the CMD.exe

Command:**PipinstallFlask**

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.



```
(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>
```

CHAPTER 2

LITERATURE SURVEY

Related Work:

A considerable number of researches and studies have been done on graduation admission datasets using several types of machine learning algorithms. One impressive work by Acharya et al. has compared between 4 different regression algorithms, which are: Linear Regression, Support Vector Regression, Decision Trees, Random Forest, to predict the chance of admit based on the best model that showed the least MSE which was multilinear regression. In addition, Chakrabarty et al. compared between both linear regression and gradient boosting regression in predicting chance of admit; point out that gradient boosting regression showed better results. Gupta et al. developed a model that studies the graduate admission process in American universities using machine learning techniques. The purpose of this study was to guide students in finding the best educational institution to apply for. Five machine learning models were built in this paper including SVM (Linear Kernel), AdaBoost, and Logistic classifiers. Waters and Miikkulainen proposed a remarkable article that helps in ranking graduation admission application according to the level of acceptance andenhances the performance of reviewing applications using statistical machine learning. Sujay applied linear regression to predict the chance of admitting graduate students in master's programs as a percentage. However, no more models were performed.

References:

1. M. S. Acharya, A. Armaan, and A. S. Antony, "A comparison of regression models for prediction of graduate admissions," ICCIDS 2019 - 2nd Int. Conf. Compute. Intell. Data Sci. Proc., pp. 1–5, 2019.

2.N. Chakrabarty, S. Chowdhury, and S. Rana, “A Statistical Approach to Graduate Admissions’ Chance Prediction,” no. March, pp. 145–154,2020.

3. N. Gupta, A. Sawhney, and D. Roth, “Will I Get in? Modeling the Graduate Admission Process for American Universities,” IEEE Int. Conf.Data Min. Work. ICDMW, vol. 0, pp. 631–638, 2016.

4.A. Waters and R. Mikkilineni, “GRADE: Graduate Admissions,” pp. 64–75, 2014.

5. S. Sujay, “Supervised Machine Learning Modelling & Analysis for Graduate Admission Prediction,” vol. 7, no. 4, pp. 5–7, 2020.

Conclusion:

The main objective of this research was to develop a prototype of the system that can be used by the students aspiring to pursue their education in their dreamed universities. The model can be used by the students for evaluating their chances of getting shortlisted in a particular university with an average accuracy of 75%. It will help the students to make better and faster decisions regarding application to universities.

CHAPTER 3

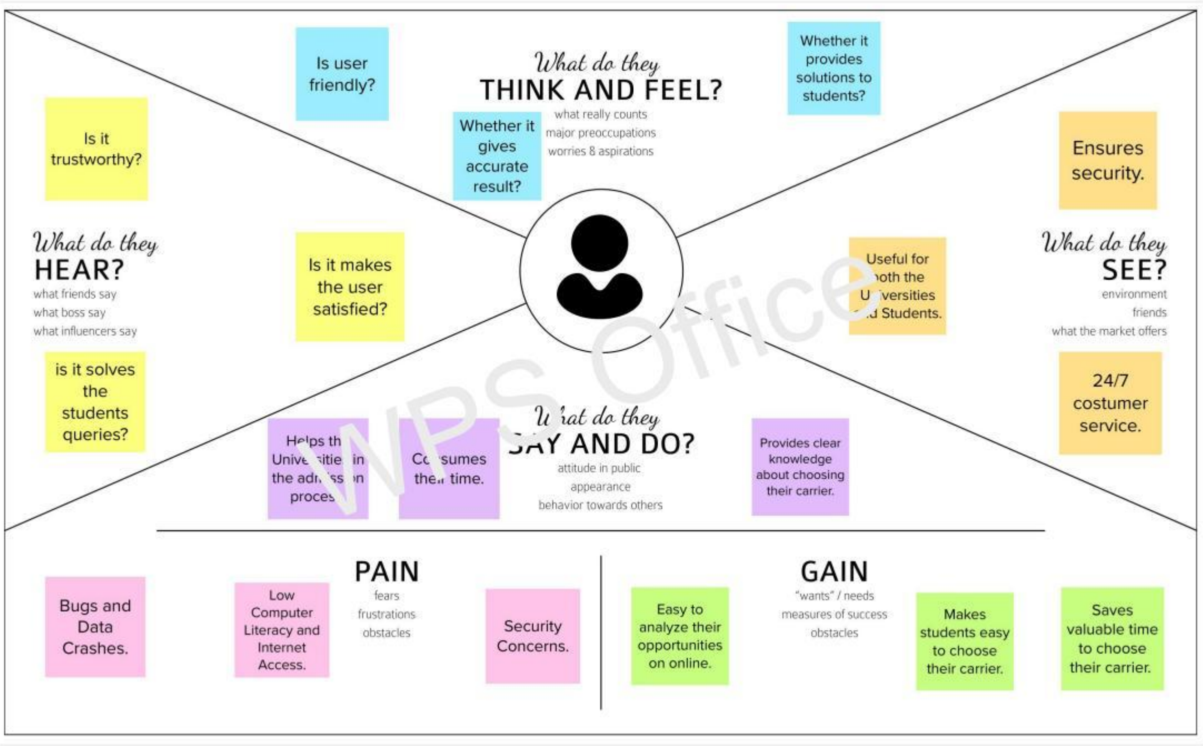
IDEATION & PROPOSED SOLUTION

EMPATHY MAP CANVAS:

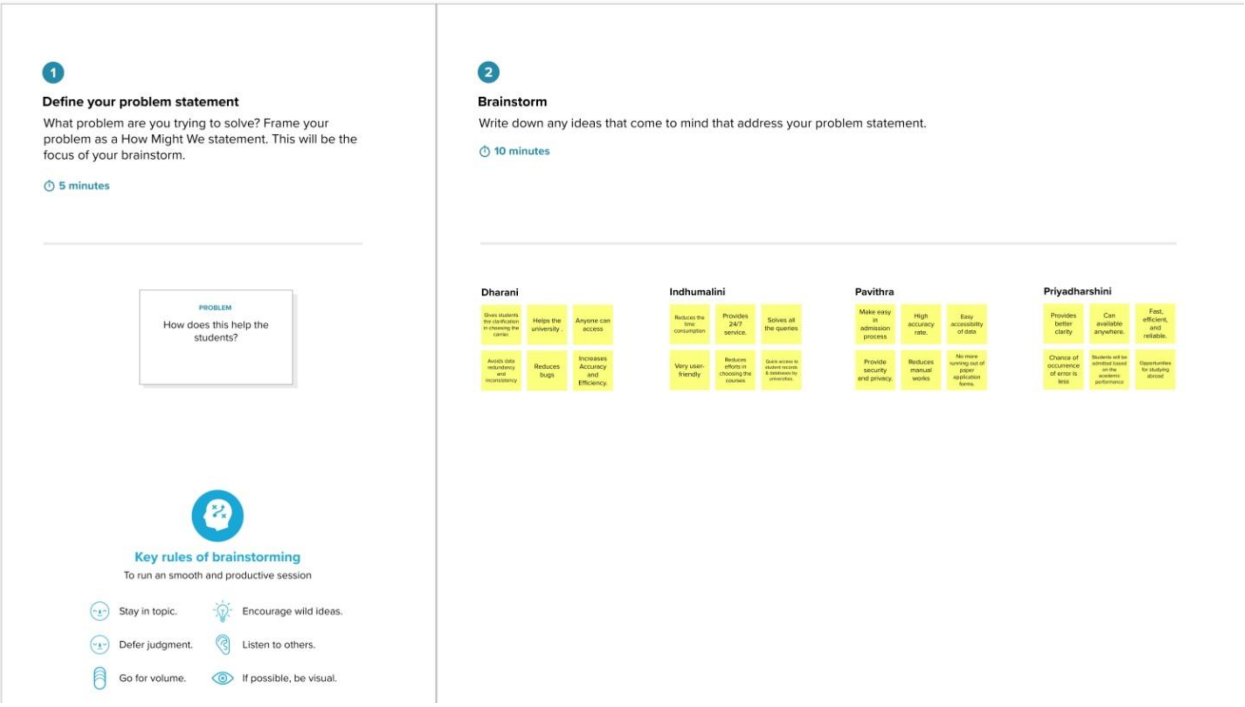
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it.

This exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges



IDEATION & BRAINSTORMING:



1

Define your problem statement

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

⏱ 5 minutes

PROBLEM

How does this help the students?

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

Encourage wild ideas.

Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

2

Brainstorm

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

⏱ 10 minutes

Dharani

Gives students the clarification in choosing the carrier.	Helps the university .	Anyone can access
Avoids data redundancy and inconsistency	Reduces bugs	Increases Accuracy and Efficiency.

Indhumalini

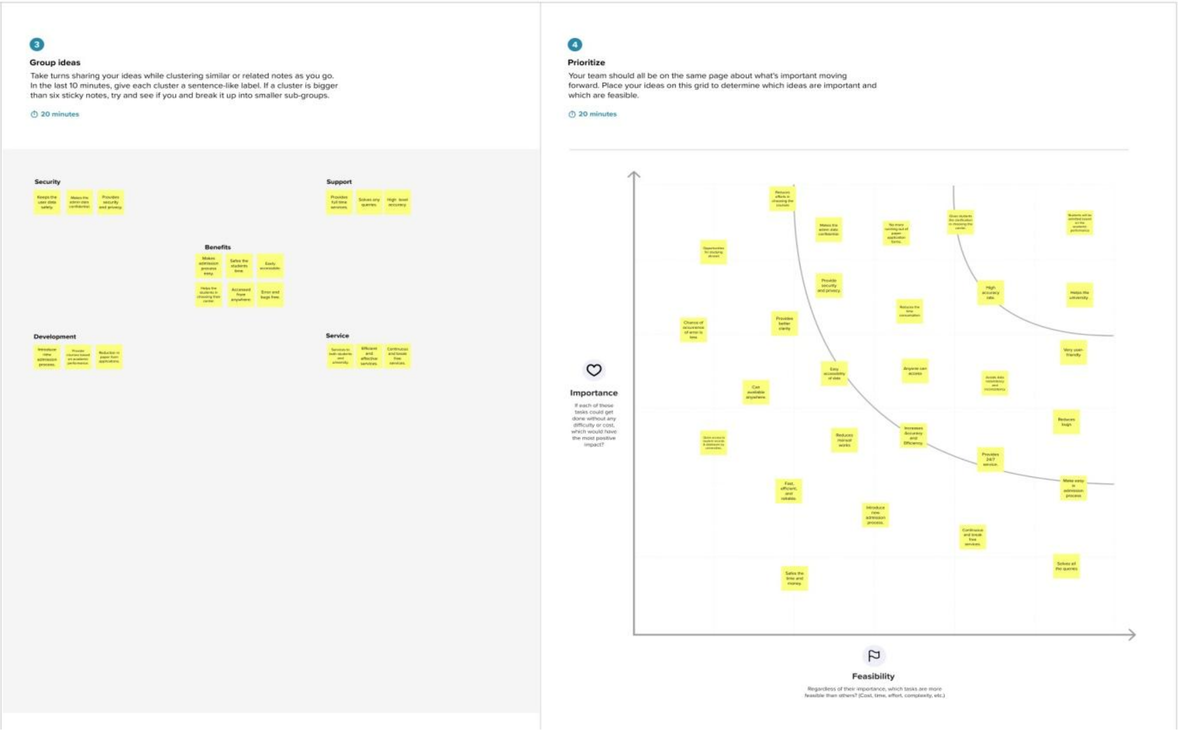
Reduces the time consumption	Provides 24/7 service.	Solves all the queries
Very user-friendly	Reduces efforts in choosing the courses	Quick access to student records & databases by universities.

Pavithra

Make easy in admission process	High accuracy rate.	Easy accessibility of data
Provide security and privacy.	Reduces manual works	No more running out of paper application forms.

Priyadharshini

Provides better clarity	Can available anywhere.	Fast, efficient, and reliable.
Chance of occurrence of error is less	Students will be admitted based on the academic performance	Opportunities for studying abroad



3

Group ideas

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

⏱ 20 minutes

Security

Keeps the user data safety.	Makes the admin data confidential	Provides security and privacy

Support

Provides full time services.	Solves any queries.	High level accuracy.

Benefits

Makes admission process easy.	Saves the students time.	Easily accessible.
Helps the students in choosing their carrier.	Accessed from anywhere.	Error and bugs free.

Development

Introduce new admission process.	Provide courses based on academic performance.	Reduction in paper from applications.

Service

Services to both students and university.	Efficient and effective services.	Continuous and break free services.

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⏱ 20 minutes

Reduces efforts in choosing the courses
Makes the admin data confidential
No more running out of paper application forms.
Gives students the clarification in choosing the carrier.
Students will be admitted based on the academic performance
Opportunities for studying abroad
Provide security and privacy.
High accuracy rate.
Helps the university.
Reduces the time consumption
Chance of occurrence of error is less
Provides better clarity
Very user-friendly
Easy accessibility of data
Anyone can access
Avoids data redundancy and inconsistency
Can available anywhere
Reduces bugs
Quick access to student records & databases by universities.
Reduces manual works
Increases Accuracy and Efficiency.
Provides 24/7 service.
Make easy in admission process
Fast, efficient, and reliable.
Introduce new admission process.
Continuous and break free services.
Solves all the queries
Saves the time and money.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

PROBLEM STATEMENT:

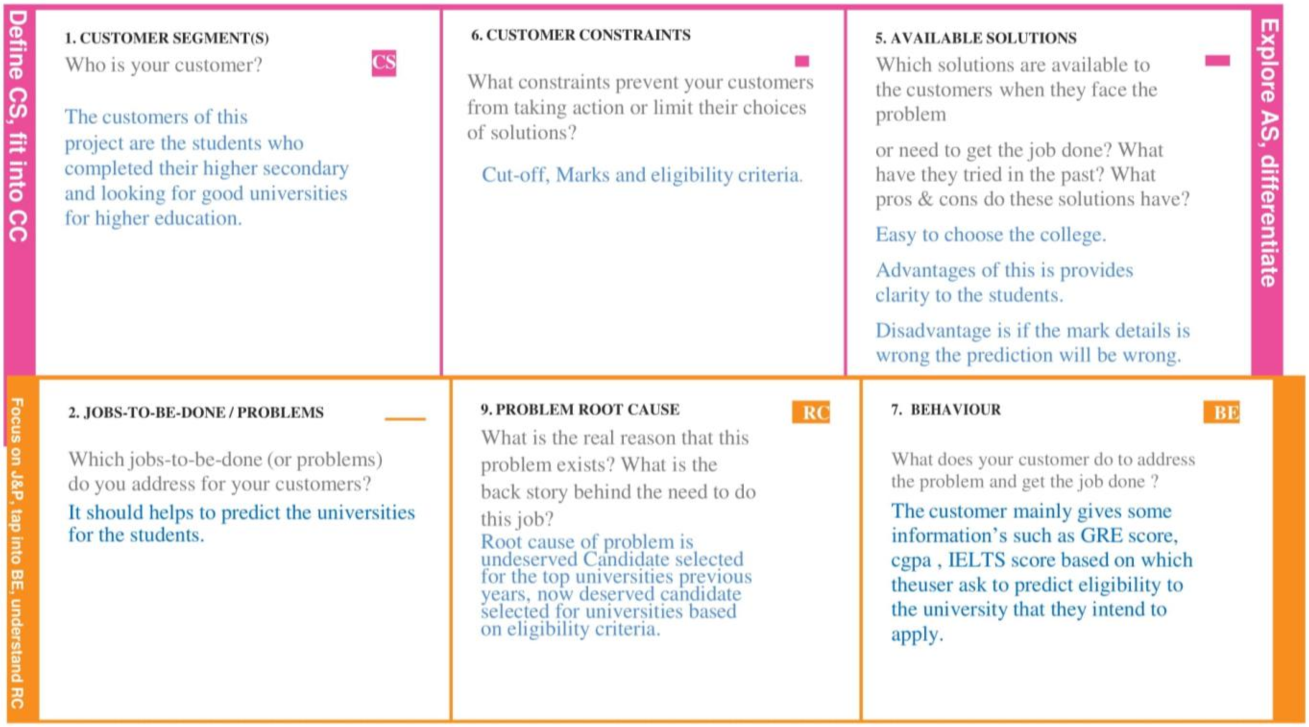
- Today in college's student details are entered manually .
- First of all,he/she takes admission form from reception and fills it and submits it into office.
- Filled form is first checked with documents like merit list an details came from university and verified by an official person ,if there is any mistake then it is corrected.
- Require much man power and more time.
- Since, all the work is done in papers so it is very hard to locate a particular student record when it is required.

PROPOSED SOLUTION:

- Reduce the time taken process the applications of students, admitting a student, conducting the online examination, verify student marks, and send call letters to selected students.
- Centralized data handling.
- Paperless admission with reduced man power.

PROBLEM SOLUTION FIT:

The Problem-Solution Fit simply means that we have found a problem with our customer and that the solution we have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why.



Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

Who is your customer?

CS

The customers of this project are the students who completed their higher secondary and looking for good universities for higher education.

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions?

Cut-off, Marks and eligibility criteria.

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?

Easy to choose the college.

Advantages of this is provides clarity to the students.

Disadvantage is if the mark details is wrong the prediction will be wrong.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers?

It should helps to predict the universities for the students.

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

Root cause of problem is undeserved Candidate selected for the top universities previous years, now deserved candidate selected for universities based on eligibility criteria.

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done ?

The customer mainly gives some information’s such as GRE score, cgpa , IELTS score based on which theuser ask to predict eligibility to the university that they intend to apply.

CHAPTER 4

REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Score sheets Curriculumvitae Letter of recommendation
FR-2	User Confirmation	Students should upload the records in the site. The candidate’s list will be shortlisted based on the marks and details they uploaded.

NON-FUNCTIONAL REQUIREMENT:

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

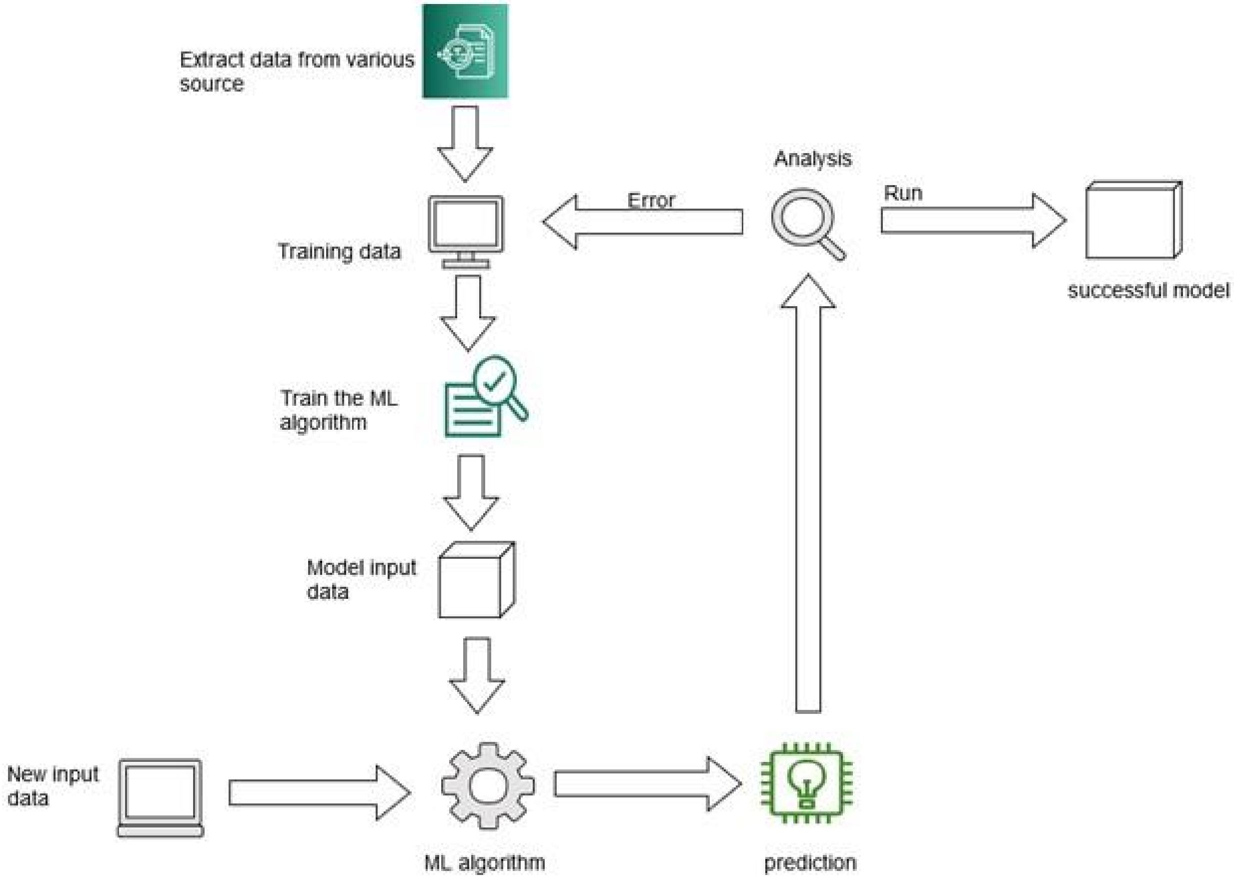
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User friendly. Easy to access by anyone.
NFR-2	**Security**	Data enter by the candidate will be secured safely.
NFR-3	**Reliability**	The possibility of errors will be reduced.
NFR-4	**Performance**	Efficient . Handle the data efficiently.
NFR-5	**Availability**	Provides 24/7 service. Anyone from anywhere.
NFR-6	**Scalability**	Provides accurate result. Can manage many users at same time.

CHAPTER 5

PROJECT DESIGN

DATA FLOW DIAGRAM:

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

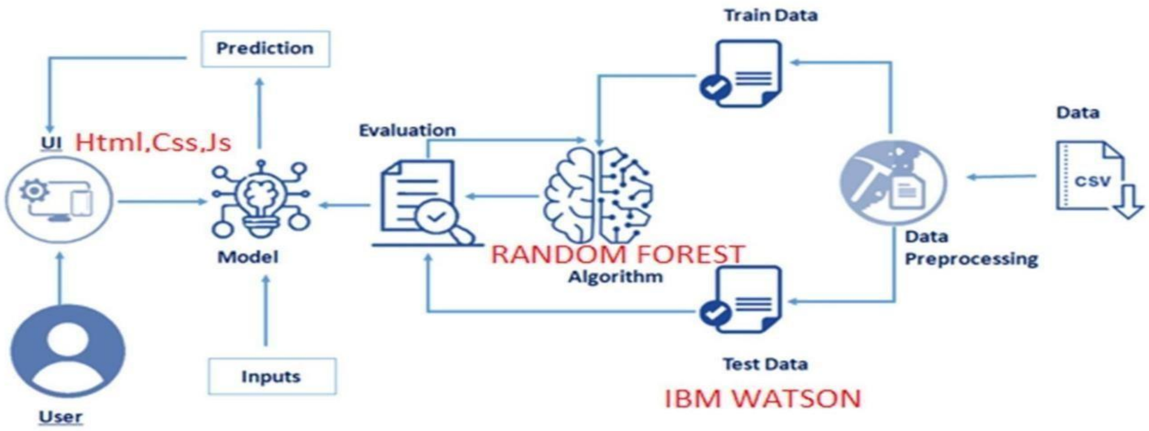


SOLUTION & TECHNICAL ARCHITECTURE:

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

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

- The aim of this project is to help students in shortlisting universities with their profiles.
- The predicted output gives them a fair idea about their admission chances in a particular university.
- This analysis should also help students who are currently preparing or will be preparing to get a better idea.



USER STORIES:

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson
4.	Application Logic-3	Logic for a process in the application	Random forest

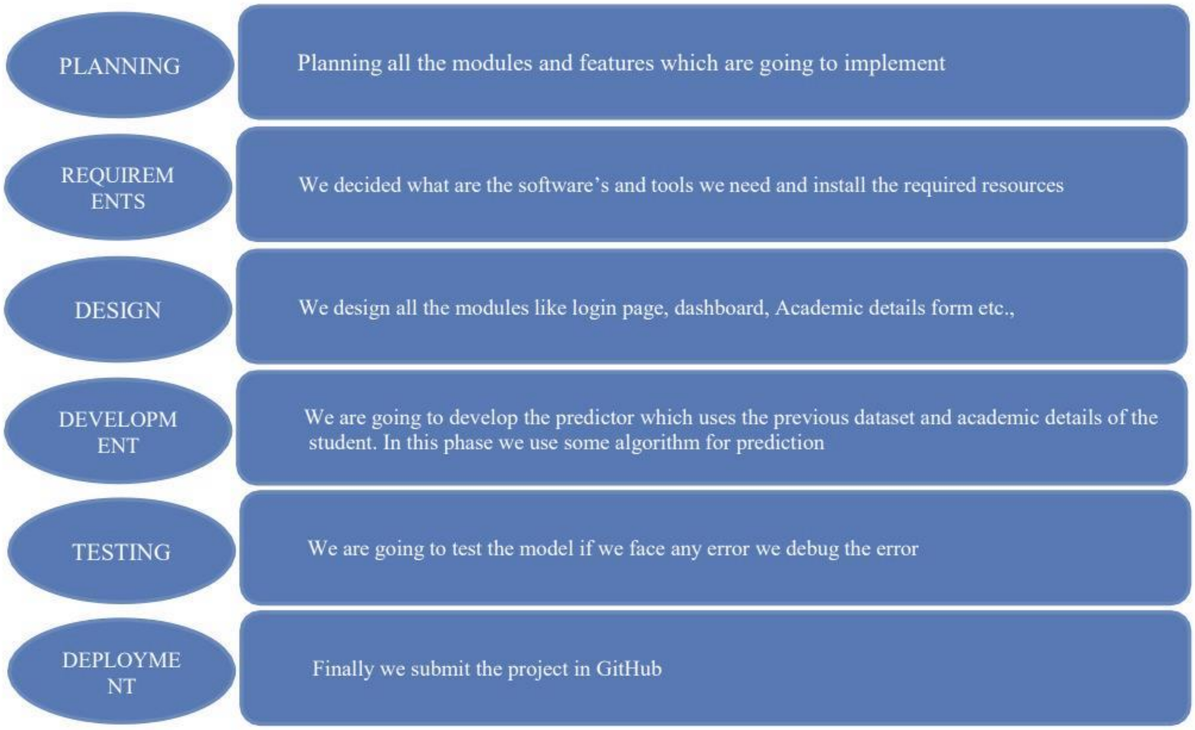
Table-2: Application Characteristics:

S.No	Characteristics	Technology
1.	Open-Source Frameworks	Flask
2.	Performance	It can handle many requests per second.

CHAPTER 6

PROJECT PLANNING & SCHEDULING

MILESTONE AND ACTIVITY:



PLANNING	Planning all the modules and features which are going to implement
REQUIREM ENTS	We decided what are the software's and tools we need and install the required resources
DESIGN	We design all the modules like login page, dashboard, Academic details form etc.,
DEVELOPM ENT	We are going to develop the predictor which uses the previous dataset and academic details of the student. In this phase we use some algorithm for prediction
TESTING	We are going to test the model if we face any error we debug the error
DEPLOYME NT	Finally we submit the project in GitHub

SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmy password	10	High	All
Sprint-1	Gmail notification	USN-2	As a user, I will receive confirmation email once I have registered for the application	5	Low	All
Sprint-2	Login	USN-3	As a user, I can login with my credential and geta personalized dashboard	5	Low	All
Sprint-2	View institutes	USN-4	As a user, I can view list of institutes	10	Low	All
Sprint-3	prediction	USN-5	As a user, I can provide my scores for prediction	10	High	All
Sprint-4	preferences	USN-6	As a user I can select and filter results based on my preferences	20	High	All

SPRINT DELIVERY SCHEDULE:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	15	6 Days	24 Oct 2022	29 Oct 2022	15	29 Oct 2022
Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	15	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

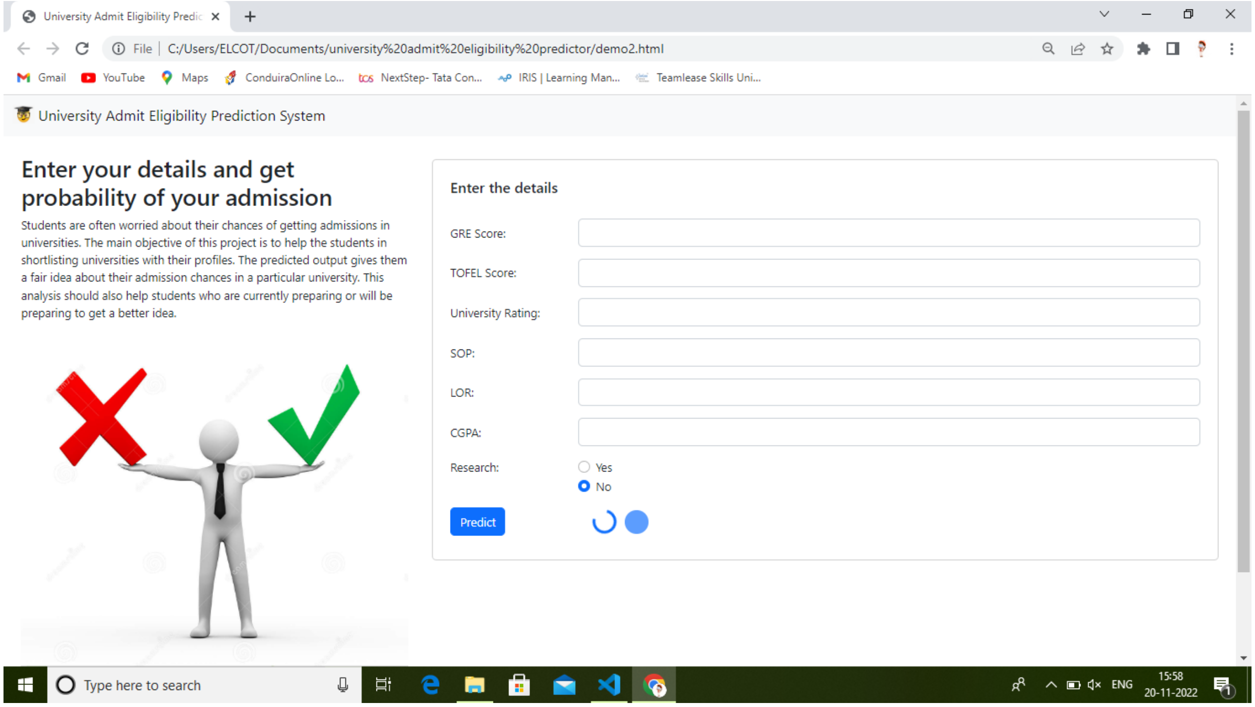
SOURCE CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
    <script type="text/javascript" src="../static/js/script.js" async></script>
    <title>University Admit Eligibility Predictor</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-light">
        <div class="container-fluid">
            <a class="navbar-brand text-responsive-h" href="/">
                <img src="images\hat.png" alt="Logo" width="30" height="24" class="d-inline-block align-text-top ">
                University Admit Eligibility Prediction System
            </a>
        </div>
    </nav>

    <div class="p-4">
        <div class="row mb-3">
            <div class="col-4">
                <h2 class="text-responsive-h">
        Enter your details and get probability of your admission
    </h2>
    <p class="text-responsive">
Students are often worried about their chances of getting admissions in universities. The main objective of this project is to help the students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.
    </p>
    <div class="d-flex justify-content-right">
        <img src="images\anime.jpg" class="card-img-top" alt="..." />
    </div>
    </div>
<div class="col-8">
    <div class="card p-2 ms-2 my-2">
        <div class="card-body">
            <h5 class="card-title pb-4">
                Enter the details
            </h5>
            <form action="/" method="post" id="theForm">
                <div class="row mb-3">
                    <label for="gre" class="col-lg-2 col-form-label">GRE Score:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="gre" name="gre" min="250" max="340" required>
                    </div>
                </div>
                <div class="row mb-3">
                    <label for="tofel" class="col-lg-2 col-form-label">TOFEL Score:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="tofel" name="tofel" min="50" max="120" required>
                    </div>
                </div>
```

```
                            <div class="row mb-3">
                                <label for="university_rating" class="col-lg-2 col-form-label">University
Rating:</label>
                                <div class="col-lg-10">
                                    <input type="number" class="form-control" id="university_rating" step="0.01"
name="university_rating" min="1" max="5" required>
                                </div>
                            </div>
                            <div class="row mb-3">
                                <label for="sop" class="col-lg-2 col-form-label">SOP:</label>
                                <div class="col-lg-10">
        <input type="number" class="form-control" id="sop" name="sop" step="0.01" min="1" max="5"
required>
                                </div>
                            </div>
                            <div class="row mb-3">
                                <label for="lor" class="col-lg-2 col-form-label">LOR:</label>
                                <div class="col-lg-10">
                                    <input type="number" class="form-control" id="lor" name="lor" step="0.01"
min="1" max="5" required>
                                </div>
                            </div>
                            <div class="row mb-3">
                                <label for="cgpa" class="col-lg-2 col-form-label">CGPA:</label>
                                <div class="col-lg-10">
                                    <input type="number" class="form-control" id="cgpa" name="cgpa" step="0.01"
min="5" max="10" required>
                                </div>
                            </div>
                            <fieldset class="row mb-3">
                                <legend class="col-form-label col-sm-2 pt-0">Research:</legend>
                                <div class="col-sm-10">
                                    <div class="form-check">
                                        <input class="form-check-input" type="radio" name="yes_no_radio"
```

```
id="gridRadios1" value="1">
                                        <label class="form-check-label" for="yes_no_radio">
                                        Yes
                                        </label>
                                    </div>
                                    <div class="form-check">
                                        <input class="form-check-input" type="radio" name="yes_no_radio"
id="gridRadios2" value="0" checked>
                                        <label class="form-check-label" for="yes_no_radio">
                                        No
                                        </label>
                                    </div>
                                </div>
                            </fieldset>
                            <div class="row lg-3">
                                <div class="col-lg-2 mb-2 me-3">
                                    <button type="submit" class="btn btn-primary" id="button">Predict</button>

  </div>
                                <div class="col-lg-2" id="spinner">
                                    <div class="spinner-border text-primary m-1" role="status">
                                        <span class="visually-hidden">Loading...</span>
                                    </div>
                                    <div class="spinner-grow text-primary m-1" role="status">
                                        <span class="visually-hidden">Loading...</span>
                                    </div>
                                </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <h1> Index Page </h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</body>
</html
```

OUTPUT:



University Admit Eligibility Prediction System

Enter your details and get probability of your admission

Students are often worried about their chances of getting admissions in universities. The main objective of this project is to help the students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

Enter the details

GRE Score:

TOFEL Score:

University Rating:

SOP:

LOR:

CGPA:

Research: ○ Yes ● No

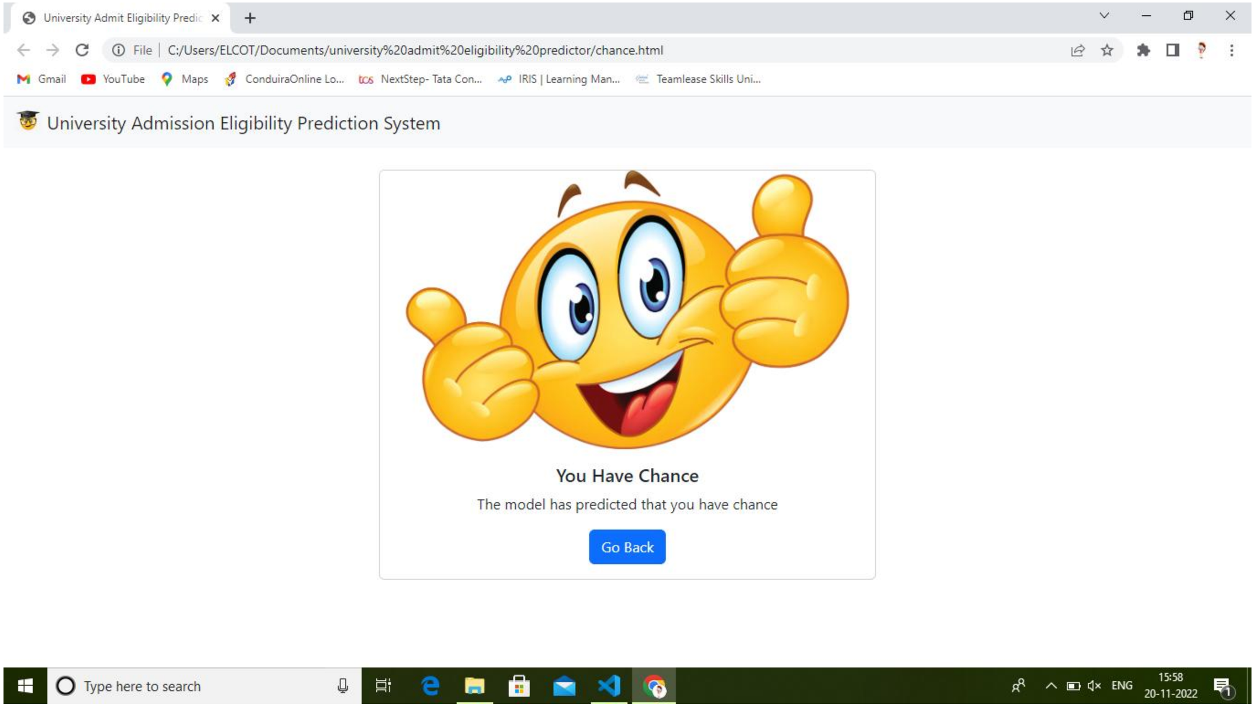
Predict

CHANCE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no">
```

```
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"

rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <script type="text/javascript" src="../static/js/script.js" async></script>
    <title>University Admit Eligibility Predictor</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-light">
        <div class="container-fluid">
            <a class="navbar-brand text-responsive-h" href="/">
                <img src="images\hat.png" alt="Logo" width="30" height="24" class="d-inline-block align-
text-top ">
                University Admission Eligibility Prediction System
            </a>
        </div>
    </nav>
    <div class="container text-center p-4">
        <div class="d-flex justify-content-center">

            <div class="card" style="width: 34rem;">
                <img src="images\chance.png" class="card-img-top" alt="...">
                <div class="card-body">
                    <h5 class="card-title">You Have Chance</h5>
                    <p class="card-text">The model has predicted that you have chance</p>
                    <a href="/home" class="btn btn-primary">Go Back</a>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

OUTPUT:



University Admission Eligibility Prediction System

You Have Chance

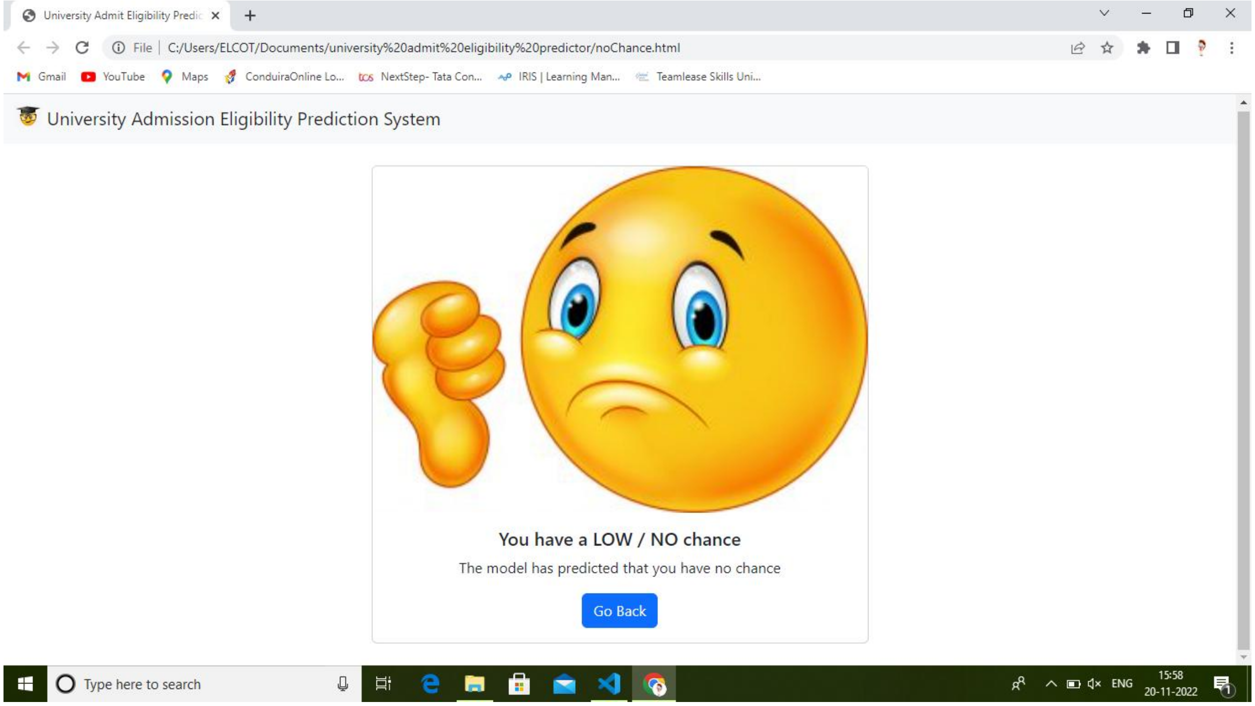
The model has predicted that you have chance

NO CHANCE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
    <script type="text/javascript" src="../static/js/script.js" async></script>
    <title>University Admit Eligibility Predictor</title>
</head>
<body>
```

```
 <nav class="navbar navbar-expand-lg bg-light">
     <div class="container-fluid">
         <a class="navbar-brand text-responsive-h" href="/">
             <img src="images\hat.png" alt="Logo" width="30" height="24" class="d-inline-block align-text-top ">
             University Admission Eligibility Prediction System
         </a>
     </div>
   </nav>
   <div class="container text-center p-4">
     <div class="d-flex justify-content-center">
         <div class="card" style="width: 34rem;">
             <img src="images\nochance.png" class="card-img-top" alt="...">
             <div class="card-body">
                 <h5 class="card-title">You have a LOW / NO chance</h5>
                 <p class="card-text">The model has predicted that you  have no chance</p>
                 <a href="/home" class="btn btn-primary">Go Back</a>
             </div>
         </div>
     </div>
   </div>
</body>
</html>
```

OUTPUT:



University Admission Eligibility Prediction System

You have a LOW / NO chance

The model has predicted that you have no chance

Go Back

CHAPTER 8

TESTING

TEST CASES:

1. Verify if user is able to see landing page
2. Verify if the menu bar is works on clicking
3. Verify user is able to navigate to admissions page
4. Verify user is able to see the testimonials
5. Verify the page elements are clear
6. Verify if the social media profiles of the university are accessible
7. Verify if the campus tour video is clear
8. Verify if proud alumni feature of the university are accessible
9. Verify if the chance of admission prediction page receives input correctly
10. Verify if the chance of admission prediction page gives accurate output
11. Verify if the course details table is clearly visible
12. Verify if the guide files are downloaded
13. Verify if the scholarship resources are accessible
14. Verify if the page is responsive
15. Verify if the page is suitable for all devices

USER ACCEPTANCE TESTING:

Defect Analysis:

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	2	18
Duplicate	0	0	0	0	0
External	2	3	0	2	7
Fixed	10	6	9	20	36
Not Reproduced	0	0	0	0	0
Skipped	0	0	2	1	3
Won't Fix	0	0	0	0	0
Totals	22	13	13	25	73

Test Case Analysis:

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

Section	TotalCases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	10	0	0	10
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	5	0	0	5
Final ReportOutput	4	0	0	4
Version Control	2	0	0	2

CHAPTER 9

RESULTS

PERFORMANCE METRICS:

1. Hours worked : 50 hours
2. Sticking to Timelines : 100%
3. Consistency of the product : 70%
4. Efficiency of the product : 85%
5. Quality of the product : 85%

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES:

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

DISADVANTAGES:

- The predictions made are not 100% accurate but accurate to an acceptable value.
- Required active internet connection.

  System will provide in accurate results if data entered incorrectly.

CHAPTER11

CONCLUSION

The project uses a Linear regressor to predict the output and a web application is built to make the UI more accessible and easy using various technologies such as python, React JS, HTML5, CSS, Flask, Scikit, Matplot, Numpy, Pandas, Seaborn and other libraries. After the deployment of the web application, it can be accessed from anywhere with internet connection.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.This project reduces the long hours of analysis to predict the eligibility of the admission to a rated university.The majority of features that are included work flawlessly and the errors that do exist are minor or graphical.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.This system helps the student to get clear idea about getting admission in the universities based on their academic performance.

CHAPTER 12

FUTURE SCOPE

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

- This can be implemented in less time for proper admission process.
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user need not travel a long distance for the admission and his/her time is also saved as a result of this automated system.
- Develop a community consisting of faculty, alumni and aspirants to get to know about the university more.

CHAPTER 13

APPENDIX

SOURCE CODE:

Landing Page:

App.py:

```
        from flask import Flask, render_template, redirect, url_for, request
import requests


app = Flask(__name__)


@app.route("/", methods = ['POST', 'GET'])
def index():
    if request.method == 'POST':
        arr = []
        for i in request.form:
            val = request.form[i]
            if val == '':
                return redirect(url_for("demo2"))
            arr.append(float(val))


        # deepcode ignore HardcodedNonCryptoSecret: <please specify a reason of ignoring this>
        API_KEY = "p-03a6131d4957c0415bcf495eba18c26430a70d9c"
        token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={
            "apikey": API_KEY,
            "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
            })
        mltoken = token_response.json()["access_token"]
        header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
        payload_scoring = {
            "input_data": [{"fields":[  'GRE Score',
                                        'TOEFL Score',
                                        'University Rating',
                                        'SOP',
                                        'LOR ',
                                        'CGPA',
                                        'Research'],
"values": [arr]
                            }]
                        }

        response_scoring = requests.post(
            'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/8308fd4c-24a5-46ab-96fa-
263657ae4ad0/predictions?version=2022-10-18',
            json=payload_scoring,
            headers=header
        ).json()

        result = response_scoring['predictions'][0]['values']

        if result[0][0] > 0.5:
            return redirect(url_for('chance', percent=result[0][0]*100))
        else:
            return redirect(url_for('no_chance', percent=result[0][0]*100))
    else:
        return redirect(url_for("demo2"))

@app.route("/home")
def demo2():
    return render_template("demo2.html")

@app.route("/chance/<percent>")
def chance(percent):
    return render_template("chance.html", content=[percent])

@app.route("/nochance/<percent>")
```

```
def no_chance(percent):
    return render_template("noChance.html", content=[percent])


@app.route('/<path:path>')
def catch_all():
    return redirect(url_for("demo2"))


if __name__ == "__main__":
    app.run()
```

Python Code:

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

LOAD THE DATASET

In [3]:

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

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It
includes your credentials.
# You might want to remove those credentials before you 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-1ijujvyruwxy5c'
object_key = 'Admission_Predict.csv'
```

```
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__,
body )

data = pd.read_csv(body)
data.head()
```

Out[3]:

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

In [4]:

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

Out[4]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
min	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

In [5]:

```
data.info()
```

```
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   GRE Score          400 non-null    int64  
 1   TOEFL Score        400 non-null    int64  
 2   University Rating  400 non-null    int64  
 3   SOP                400 non-null    float64
 4   LOR                400 non-null    float64
 5   CGPA               400 non-null    float64
 6   Research           400 non-null    int64  
 7   Chance of Admit    400 non-null    float64
dtypes: float64(4), int64(4)
memory usage: 25.1 KB
```

In [6]:

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

Out[6]:

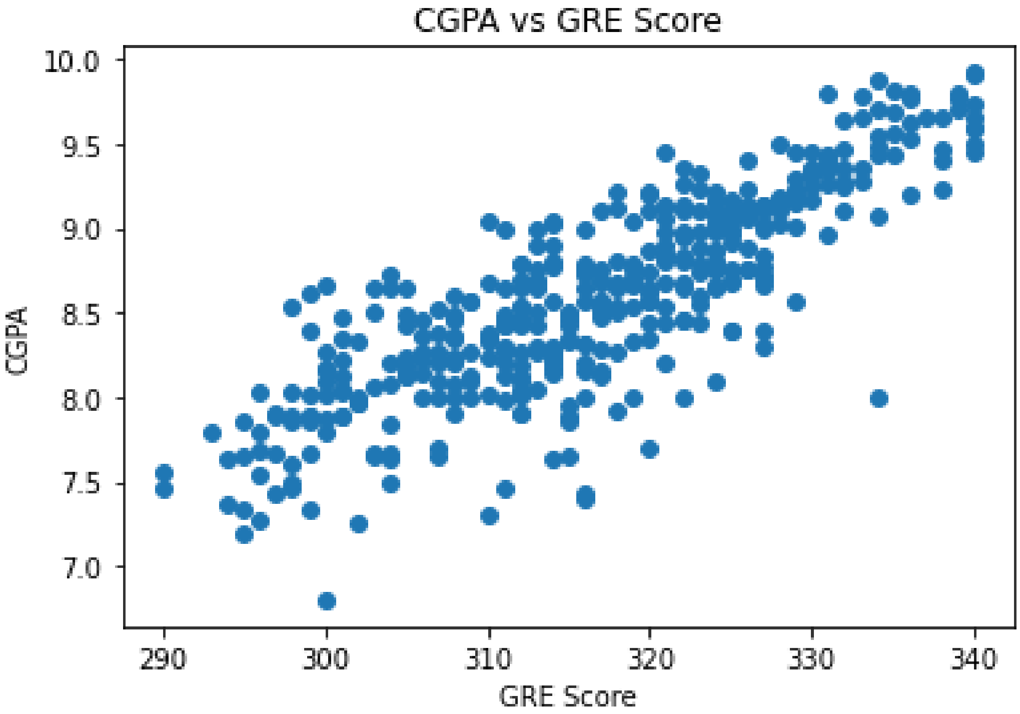
```
GRE Score            0
TOEFL Score          0
University Rating    0
```

```
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

VISUALIZATION

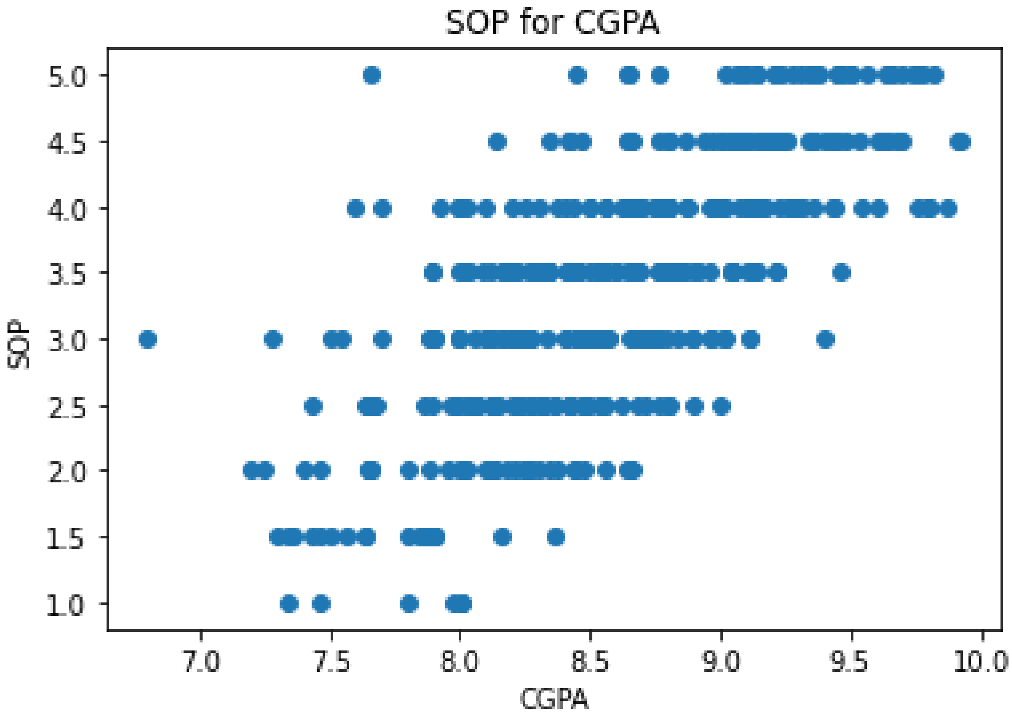
In [7]:

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



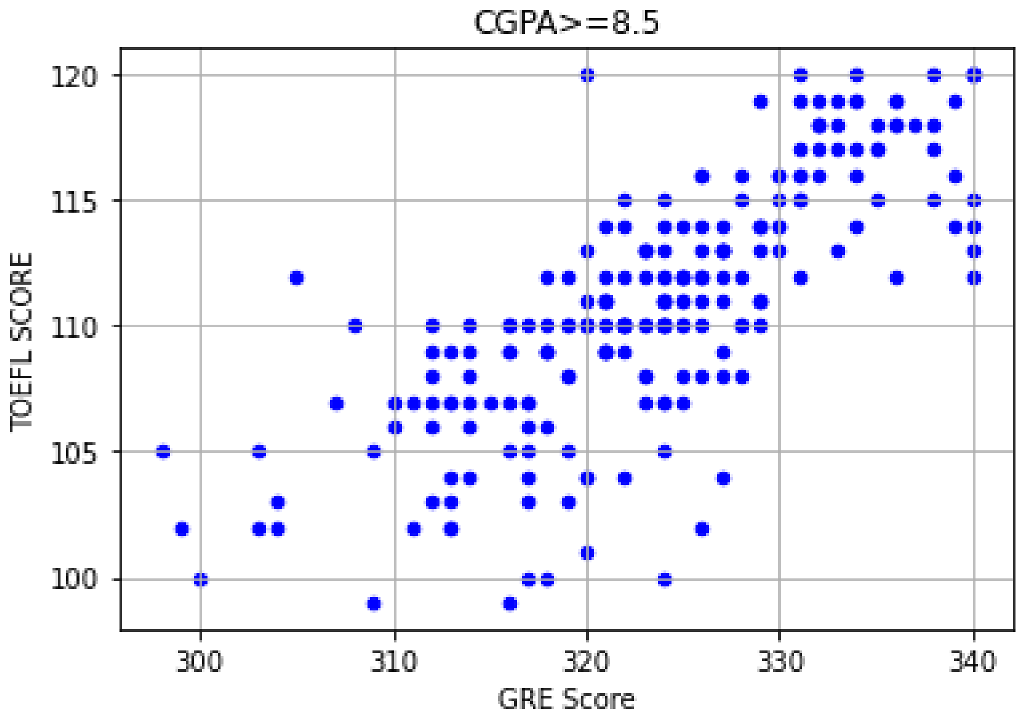
In [8]:

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



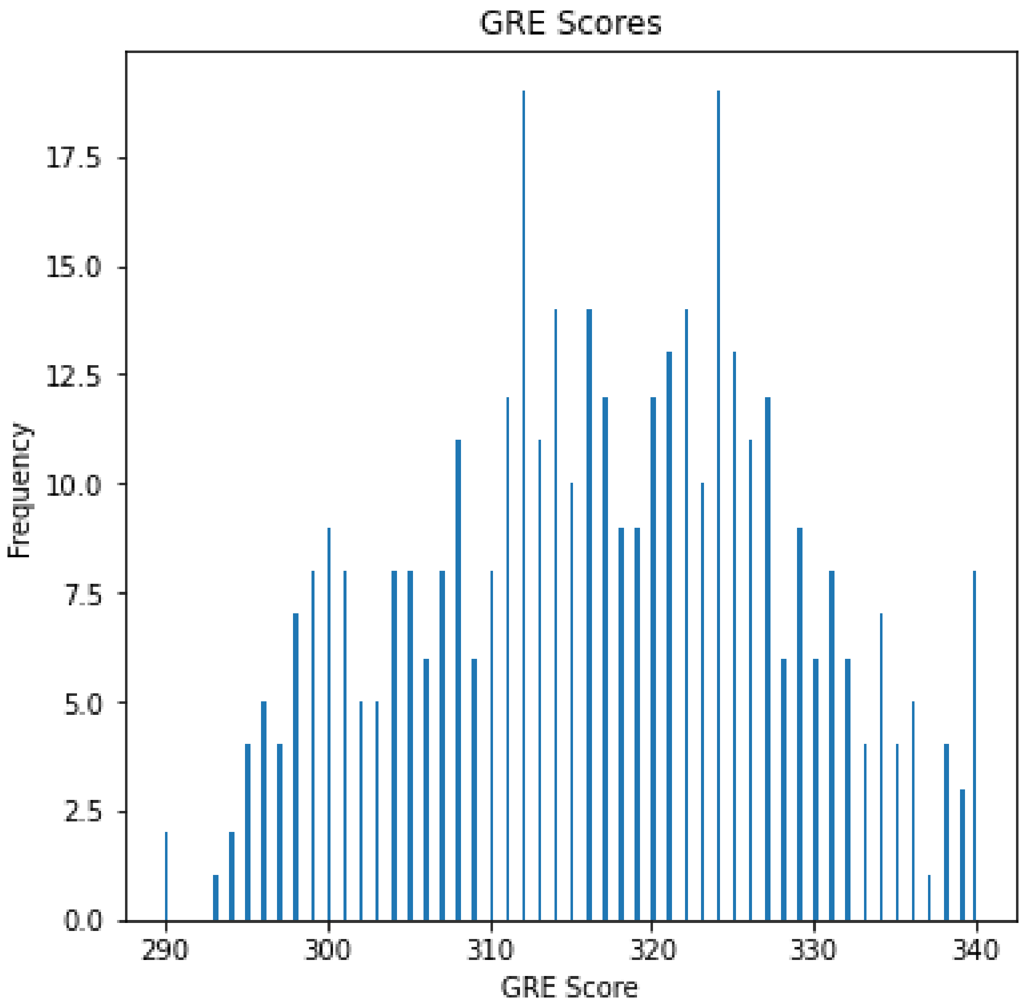
In [9]:

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

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

plt.show()
```



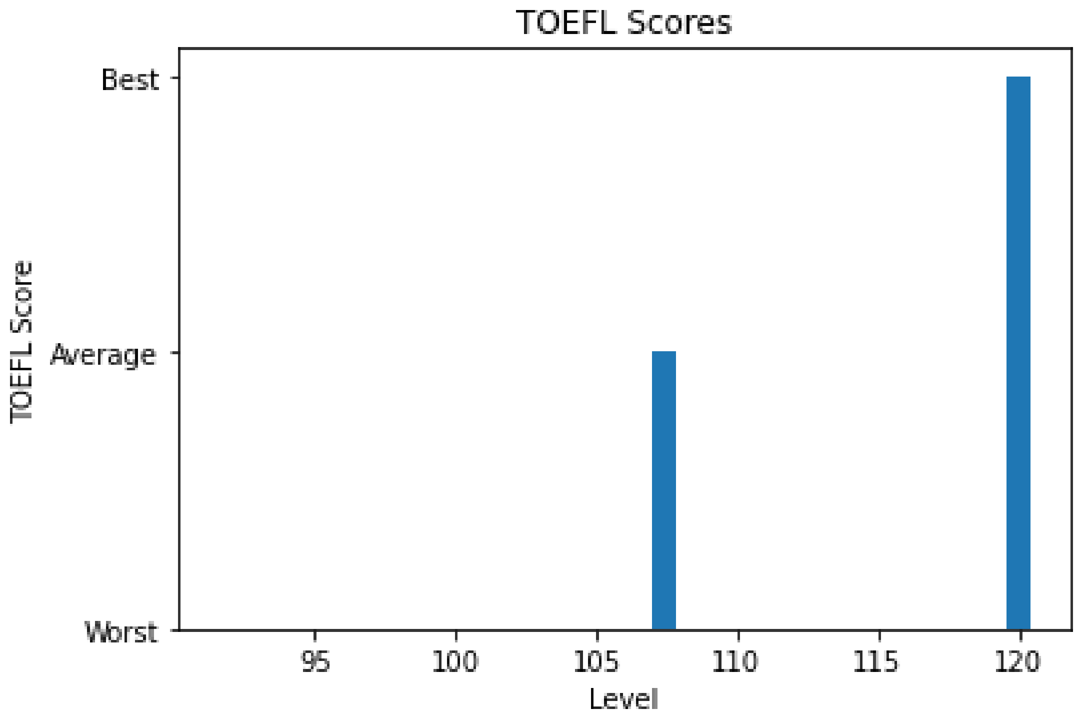
In [10]:

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

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

plt.show()
```



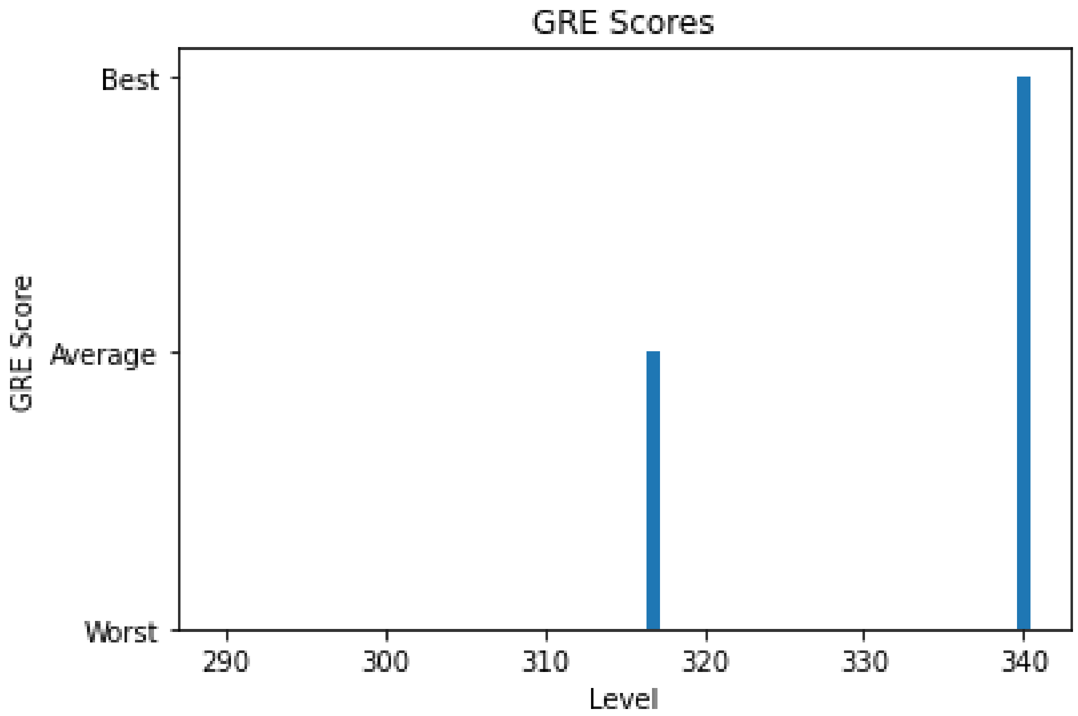
In [11]:

```
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()
```



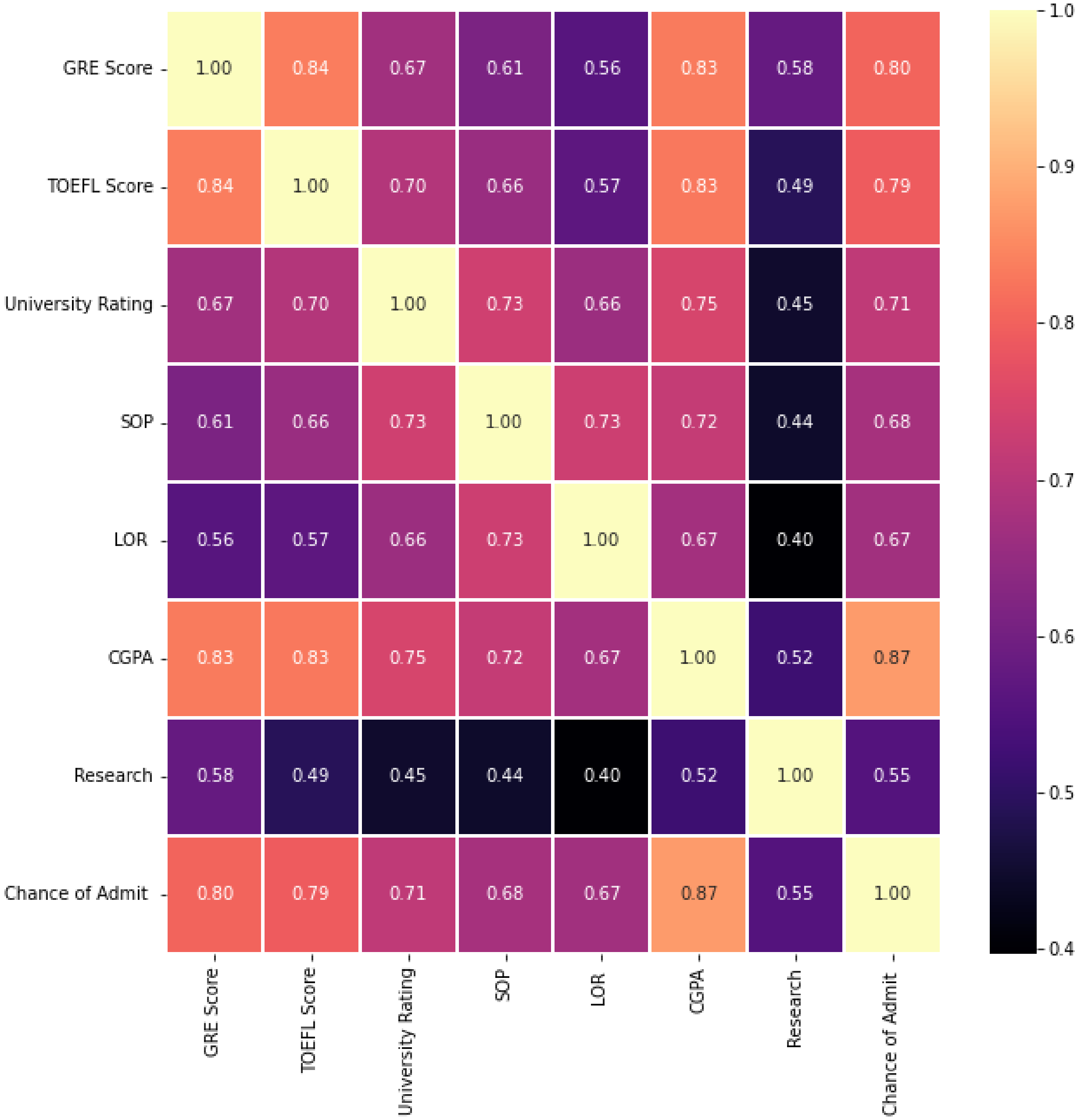
In [12]:

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

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

plt.show()
```



In [13]:

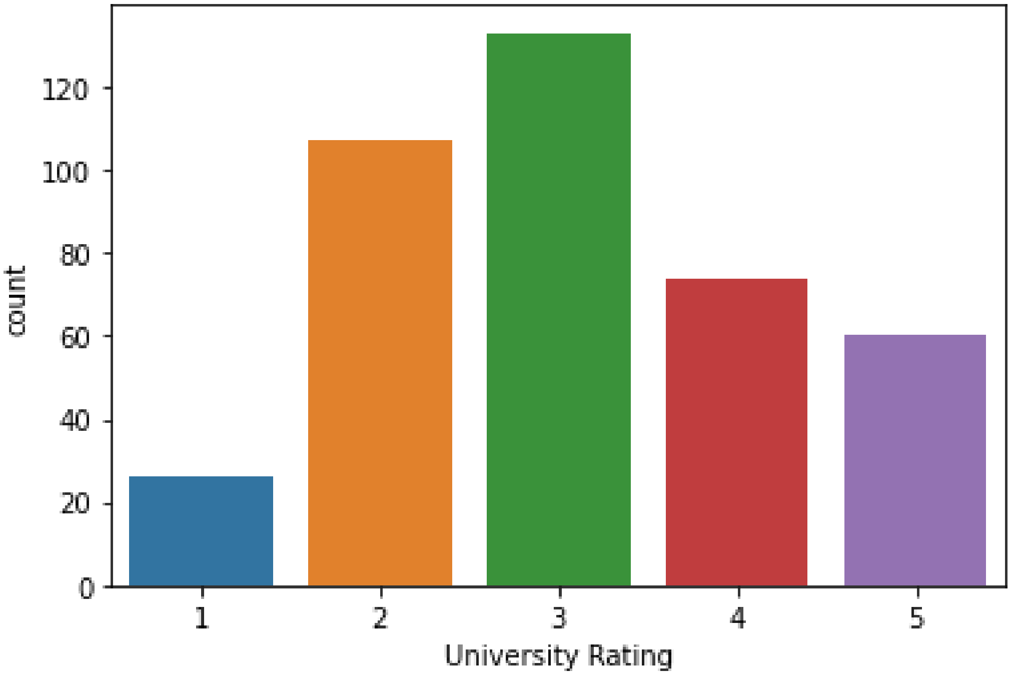
```
plt.figure(figsize=(10, 10))

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

plt.show()
```



In [14]:

```
data.Research.value_counts()

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

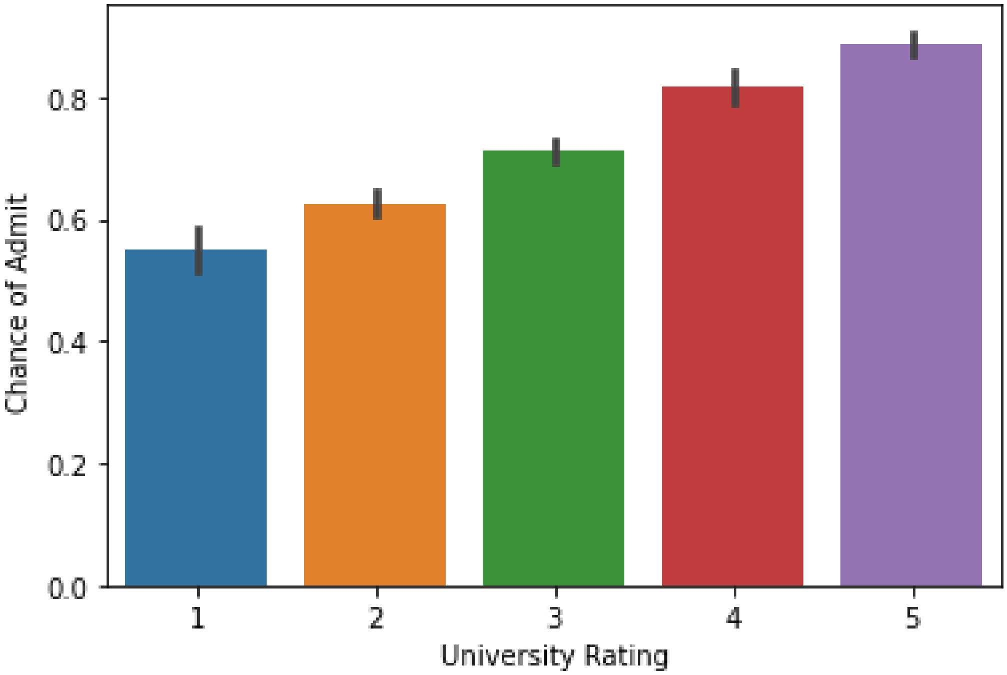
Out[14]:



In [15]:

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

Out[15]:



TRAINING AND TESTING SPLIT

In [16]:

```
X=data.drop(['Chance of Admit '],axis=1) #input data_set
y=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
rgr = GradientBoostingRegressor()
rgr.fit(X_train,y_train)
```

Out[18]:

```
GradientBoostingRegressor()
```

In [19]:

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

Out[19]:

```
0.8182951077367754
```

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: 0.04655745821447952
Mean Squared Error: 0.003985893658760451
Root Mean Squared Error: 0.06313393428862525
```

In [22]:

```
y_train = (y_train>0.5)
y_test = (y_test>0.5)
```

In [23]:

```
from sklearn.linear_model._logistic import LogisticRegression

lore = 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_matrix

print('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))
```

```
Accuracy Score: 0.9666666666666667
Recall Score: 0.9811320754716981
ROC AUC Score: 0.9191374663072777
Confussion Matrix:
 [[ 6  1]
 [ 1 52]]
```

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
```

```
Requirement already satisfied: ibm-watson-machine-learning in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
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: lomond 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: requests in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)
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: 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: 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: certifi in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning)
(2022.9.24)
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: 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-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: 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)
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": "poJ22ua6BCG9qY33B8fkgnz1bnP1f9DZqUlF9NkBM1bZ"
}

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['metadata']['id'])
    return idr
```

In [34]:

```
space_uid = guid_from_space_name(client, "university")
print(space_uid[0])
```

```
4f0253e5-f162-4eec-84ba-72e01fb69ab9
```

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  base
```

```
scikit-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  base
pytorch_1.1-py3.6              10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl      111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10        125b6d9a-5b1f-5e8d-972a-b251688ccf40  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  base
tensorflow_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
autoai-ts_rt22.2-py3.10        396b2e83-0953-5b86-9a55-7ce1628a406f  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
pytorch-onnx_rt22.2-py3.10     40e73f55-783a-5535-b3fa-0c8b94291431  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_3.0                5c1b0ca2-4977-5c2e-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
runtime-22.2-py3.10-xc         5e8cddff-db4a-5a6a-b8aa-2d4af9864dab  base
autoai-kb_3.1-py3.7            632d4b22-10aa-5180-88f0-f52dfb6444d7  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_NAME = '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_specs': [],
  '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': []},
  'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
   'name': 'runtime-22.1-py3.9'},
  'type': 'scikit-learn_1.0'},
 'metadata': {'created_at': '2022-11-20T11:29:56.678Z',
  'id': '0147b362-92b8-45b2-871d-b9db7e8811c4',
  'modified_at': '2022-11-20T11:30:03.018Z',
  'name': 'university',
  'owner': 'IBMid-6630041JHH',
  'resource_key': 'ab7b4c64-216d-42ad-a794-5a69c68bb9ae',
  'space_id': '4f0253e5-f162-4eec-84ba-72e01fb69ab9'},
 'system': {'warnings': []}}
```

In [42]:

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

Out[42]:

```
'0147b362-92b8-45b2-871d-b9db7e8811c4'
```

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
)
```

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

Synchronous deployment creation for uid: '0147b362-92b8-45b2-871d-
b9db7e8811c4' started

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

```
initializing
Note: online_url is deprecated and will be removed in a future release. Use
serving_urls instead.

ready
-----------------------------------------------------------------------------
-------------------
Successfully finished deployment creation, deployment_uid='4d1fe8ed-7ee3-
408c-b503-cfbe979bc2e0'
-----------------------------------------------------------------------------
------------------
```

DEMO2.HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank"
href="../static/css/styles.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <script type="text/javascript" src="../static/js/script.js" async></script>
    <title>University Admit Eligibility Predictor</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-light">
        <div class="container-fluid">
            <a class="navbar-brand text-responsive-h" href="/">
                <img src="images\hat.png" alt="Logo" width="30" height="24" class="d-inline-block
align-text-top ">
                University Admit Eligibility Prediction System
            </a>
        </div>
    </nav>

    <div class="p-4">
        <div class="row mb-3">
```

```
          <div class="col-4">
             <h2 class="text-responsive-h">
                Enter your details and get probability of your admission
             </h2>
             <p class="text-responsive">  Students are often worried about their chances of getting
admissions in universities. The main objective of this project is to help the students in shortlisting
universities with their profiles. The predicted output gives them a fair idea about their admission chances
in a particular university. This analysis should also help students who are currently preparing or will be
preparing to get a better idea.
            </p>
            <div class="d-flex justify-content-right">
               <img src="images\anime.jpg" class="card-img-top" alt="..." />
            </div>
            </div>
         <div class="col-8">
            <div class="card p-2 ms-2 my-2">
               <div class="card-body">
                  <h5 class="card-title pb-4">
                     Enter the details
                  </h5>
                  <form action="/" method="post" id="theForm">
                     <div class="row mb-3">
                        <label for="gre" class="col-lg-2 col-form-label">GRE Score:</label>
                        <div class="col-lg-10">
                           <input type="number" class="form-control" id="gre" name="gre" min="250"
max="340" required>
                        </div>
                     </div>
                     <div class="row mb-3">
                        <label for="tofel" class="col-lg-2 col-form-label">TOFEL Score:</label>
                        <div class="col-lg-10">
                           <input type="number" class="form-control" id="tofel" name="tofel" min="50"
max="120" required>
                        </div>
                     </div>
                     <div class="row mb-3">
                        <label for="university_rating" class="col-lg-2 col-form-label">University
Rating:</label>
                        <div class="col-lg-10">
                           <input type="number" class="form-control" id="university_rating" step="0.01"
name="university_rating" min="1" max="5" required>
                        </div>
                     </div>
                     <div class="row mb-3">
                        <label for="sop" class="col-lg-2 col-form-label">SOP:</label>  <div class="col-lg-
10">
                           <input type="number" class="form-control" id="sop" name="sop" step="0.01"
min="1" max="5" required>
                        </div>
                     </div>
```

```
                            <div class="row mb-3">
                                <label for="lor" class="col-lg-2 col-form-label">LOR:</label>
                                <div class="col-lg-10">
                                    <input type="number" class="form-control" id="lor" name="lor" step="0.01"
min="1" max="5" required>
                                </div>
                            </div>
                            <div class="row mb-3">
                                <label for="cgpa" class="col-lg-2 col-form-label">CGPA:</label>
                                <div class="col-lg-10">
                                    <input type="number" class="form-control" id="cgpa" name="cgpa" step="0.01"
min="5" max="10" required>
                                </div>
                            </div>
                            <fieldset class="row mb-3">
                                <legend class="col-form-label col-sm-2 pt-0">Research:</legend>
                                <div class="col-sm-10">
                                    <div class="form-check">
                                        <input class="form-check-input" type="radio" name="yes_no_radio"
id="gridRadios1" value="1">
                                        <label class="form-check-label" for="yes_no_radio">
                                        Yes
                                        </label>
                                    </div>
                                    <div class="form-check">
                                        <input class="form-check-input" type="radio" name="yes_no_radio"
id="gridRadios2" value="0" checked>
                                        <label class="form-check-label" for="yes_no_radio">
                                        No
                                        </label>
                                    </div>
                                </div>
                            </fieldset>
                            <div class="row lg-3">
                                <div class="col-lg-2 mb-2 me-3">
                                    <button type="submit" class="btn btn-primary" id="button">Predict</button>
                                </div>
                                <div class="col-lg-2" id="spinner">
                                    <div class="spinner-border text-primary m-1" role="status">
                                        <span class="visually-hidden">Loading...</span>
                                    </div>
                                    <div class="spinner-grow text-primary m-1" role="status">
                                        <span class="visually-hidden">Loading...</span>
                                    </div>
                                </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <h1> Index Page </h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</body>
</html>
```

CHANCE.HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank"
href="../static/css/styles.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <script type="text/javascript" src="../static/js/script.js" async></script><title>University Admit
Eligibility Predictor</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-light">
        <div class="container-fluid">
            <a class="navbar-brand text-responsive-h" href="/">
                <img src="images\hat.png" alt="Logo" width="30" height="24" class="d-inline-block align-
text-top ">
                University Admission Eligibility Prediction System
            </a>
        </div>
    </nav>
    <div class="container text-center p-4">
        <div class="d-flex justify-content-center">

            <div class="card" style="width: 34rem;">
                <img src="images\chance.png" class="card-img-top" alt="...">
                <div class="card-body">
                    <h5 class="card-title">You Have Chance</h5>
                    <p class="card-text">The model has predicted that you have chance</p>
                    <a href="/home" class="btn btn-primary">Go Back</a>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

NO CHANCE.HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank"
href="../static/css/styles.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-

Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
    <script type="text/javascript" src="../static/js/script.js" async></script>
    <title>University Admit Eligibility Predictor</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-light">
        <div class="container-fluid">
            <a class="navbar-brand text-responsive-h" href="/">
                <img src="images\hat.png" alt="Logo" width="30" height="24" class="d-inline-block align-
text-top ">
                University Admission Eligibility Prediction System
            </a>
        </div>
    </nav>
    <div class="container text-center p-4">
        <div class="d-flex justify-content-center">
            <div class="card" style="width: 34rem;">
                <img src="images\nochance.png" class="card-img-top" alt="...">
                <div class="card-body">
                    <h5 class="card-title">You have a LOW / NO chance</h5>
                    <p class="card-text">The model has predicted that you  have no chance</p>
                    <a href="/home" class="btn btn-primary">Go Back</a>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

SCRIPT.JS:

```
const button = document.getElementById('button');
const theForm = document.getElementById('theForm');
const loading = document.getElementById('spinner');

const disableButton = () => {
    console.log('Submitting form...');
    button.disabled = true;
    button.className = "btn btn-outline-primary";
    button.innerHTML = "Predicting..."

   loading.style.display = "block"
};

const enableButton = () => {
    console.log('Loading window...');
    button.disabled = false;
    button.className = "btn btn-primary"
    button.innerHTML = "Predict"
    loading.style.display = "none"
}

theForm.onsubmit = disableButton;

window.onload = enableButton;
```

STYLE.CSS:

```
* {
    margin: 0;
    padding: 0;
    border: 0;
}
body {
    font: 62.5%/1.5  "Lucida Grande", "Lucida Sans", Tahoma, Verdana, sans-serif;
    background: #e0eafc;
    background: -webkit-linear-gradient(to right, #e0eafc, #cfdef3);

 background: linear-gradient(to right, #e0eafc, #cfdef3);
    color: #000000;
    text-align:center;
}
```

```
h1 {

    font-size: 2.2em;

}


h2 {

    font-size: 2.0em;

}


h4 {

    font-size: 1.6em;


}

p {
    font-size: 1.2em;
}

input.text
  {
  padding: 3px;
  border: 1px solid #999999;
}

img {
    max-width: auto;
    height: auto;
}

.text-responsive {
    font-size: calc(50% + 0.6vw + 0.6vh);
}

.text-responsive-h {
    font-size: calc(80% + 0.6vw + 0.6vh);
}
```

DEMO VIDEO LINK:

https://drive.google.com/file/d/16-g-xvhjDakRpfNpuZDSPG_fjo89znSJ/view?usp=drivesdk