

1. INTRODUCTION

- a. Project Overview
- b. Purpose

2. LITERATURE SURVEY

- a. Existing problem
- b. References
- c. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- a. Empathy Map Canvas
- b. Ideation & Brainstorming
- c. Proposed Solution
- d. Problem Solution fit

4. REQUIREMENT ANALYSIS

- a. Functional requirement
- b. Non-Functional requirements

5. PROJECT DESIGN

- a. Data Flow Diagrams
- b. Solution & Technical Architecture
- c. User Stories

6. PROJECT PLANNING & SCHEDULING

- a. Sprint Planning & Estimation
- b. Sprint Delivery Schedule
- c. Reports from JIRA

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

- a. Feature 1
- b. Feature 2
- c. Database Schema (if Applicable)

8. TESTING

- a. Test Cases

- b. User Acceptance Testing
- 9. **RESULTS**

- a. Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

Source Code

GitHub & Project Demo Link

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

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.

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

1. Admin
2. Students
3. Account
4. Student section.

This Database follows a typical event flow seen in such a system. The design and implementation of a comprehensive student information system and user interface is to replace the current paper records. This system is intended for communication purpose between users of academic institutions. This system helps the administrator to easy access the information of students. This system is also helpful for the administrator because he/she can easily bring changes to the records of the students. The mobile application would require connecting to the database on a remote server using Wi-Fi technology.

Our system primarily focuses on building an efficient and user-friendly Android mobile application for a Cloud based Intra-College Communication Information System using Mobile Clients. The application will be installed on the users (student/teachers) Smartphone. Here the concept of unique ID is also included using which each student gets one unique identification number by email. This id will help to access his info or find him from multiple students.

After XII, students desiring to take admissions in professional colleges like engineering face lots of problems. Admissions in engineering colleges in the state of Maharashtra or any

state is based upon common entrance test (CET) and since more than 1.5lakh seats are to be allotted in more than 200 engineering colleges and over 35 different branches of engineering , for students belonging to many categories like open, home university, outside home university, reserved category(SC,ST, OBC, etc.) the problem becomes more serious and students struggle to understand which colleges they are likely to get admitted in, even after going through cut-off data of previous years. Many students fill wrong Options and fail to get admission. To minimize the stress of students we came up with the idea of a computer aided method which will help the students get the list of all colleges in which they could get the admission at the click of a button, making the admission process fast and easy.

1.2 PURPOSE

College admission predictor is a boon to many students. This helps the student not only to help in filling out the application forms but also give the students an idea about their future college by calculating their cut off.

When students come from rural places, they find it hard to go along with the formal procedures. So, this application helps them a lot and eases out their fear. Whatever may be their scores, this application helps to find the best colleges. Hence, our proposed computer aided system will help the students to get the list of all colleges in which they could get the admission at the click of a button. The students only have to enter their marks of XII, AIEEE etc.

With this application, the students can very easily obtain the list of colleges even branch wise and course wise. This will not only make the admission process easy but also minimizes stress for students. The main objective of our system is to make the right choice of colleges.

This is a Requirements Specification Document for a new web-based University Admissions Predictor. It is an AI 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.

In the current world scenario, it is not enough for a student to just have an Under Graduate degree. Most employers now look for higher qualifications in their new recruits. As a result, the demands for a good higher education are at an all-time high. A lot of students from India prefer to continue their higher education with foreign universities, especially in the United States. In order to get admitted to these foreign universities, a set of academic requirements are needed. However, because of the sheer number of universities of different levels, students are often stuck in a dilemma till the very last minute as to whether or not their applications will be accepted or not as no concrete documentation is available which lists the requirements.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

2.1.1 University admit eligibility predictor existing system

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

1. When the student comes in college.
2. First of all, he/she takes admission form from reception.
3. Fills it and submits it into office.
4. Filled form is first checked with documents like merit list and details came from university and verified by an official person, if there is any mistake then it is corrected.
5. At the time of submission of it the fees are deposited by the candidate.

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

7. Candidate gets the receipt of fees deposition.

Advantages

- Students can easily choose their universities based on their eligibility for the universities.
- University admit eligibility predictor is mostly user friendly.

Disadvantages

- Require much man power i.e. much efforts, much cost and hard to operate and maintain.
- Since, all the work is done in papers so it is very hard to locate a particular student record when it is required.

2.1.2 University admit eligibility predictor proposed system

The main goal of the system is to automate the process carried out in the organization with improved performance and realize the vision of paperless admission. Some of the goals of the system are listed below:

- Manage large number of student details.
- Manage all details of student who registered for the course
- Create student accounts and maintain the data is effectively.
- View all the details of the students.
- Reduce the work load in interview the students for selection
- Activities like updating, modification, deletion of records should be easier.

Advantages

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also

based on the feedback from users of previous metrics tools. Following are the objectives of the proposed system:

- Reach to geographically scattered student: One of the important objectives of the admission system is communicate with all the students scattered geographically.
- Reducing time in activities: 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: Transfer the data smoothly to all the departments involved and handle the data centralized way.
- Paperless admission with reduced man power: Reduce the manpower needed to perform all the admission and administration task by reducing the paper works Cost cutting. Reduce the cost involved in the admission process.
- Operational efficiency: Improve the operational efficiency by improving the quality of the process.

2.2 REFERENCES

- [1]"The Daily Star," 2017. [Online]. Available: <https://www.thedailystar.net/backpage/public-universities-admissionstill-uphill-battle-1438285>. [Accessed 2020].
- [2] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, 3 ed., San Francisco:Morgan Kaufmann Publishers, 2011.
- [3] Aggarwal and C. C, Data mining: the textbook, Springer, 2015.
- [4] C. C. Aggarwal, Data Mining: The Textbook, 1 ed., Springer International Publishing Switzerland, 2015.
- [5] T. Calders and M. Pechenizkiy, "Introduction to the special section on educational data mining," SIGKDD Explorations, vol. 13, no. 2, pp. 3-6, 2012.
- [6] J. P. Bigus, Data mining with neural networks: Solving business problems from application development to decision support, McGrawHill Companies, 1996.
- [7] C. Romero and S. Ventura, "Educational data mining: a review of the state of the art," IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 6, pp. 601-618, 2010.
- [8] B. K. Baradwaj and S. Pal, "Mining educational data to analyze students' performance," ArXiv Preprint ArXiv:1201.3417, 2012.

- [9] El-Halees and A. M, "Mining students data to analyze e-Learning behavior: A Case Study," Mining students data to analyze e-Learning behavior: A Case Study, vol. 29, 2009.
- [10] C. Romero, S. Ventura and E. Garcia, "Data mining in course management systems: Moodle case study and tutorial," Computers & Education, vol. 51, no. 1, pp. 368-384, 2008.
- [11] P. Binu, A. Chandran and M. Rahul, "A Cloud-Based Data Analysis and Prediction System for University Admission," in 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2019.
- [12] M. S. Acharya, A. Armaan and A. S. Antony, "A comparison of regression models for prediction of graduate admissions," in International Conference on Computational Intelligence in Data Science (ICCIDS), 2019.
- [13] A. M. Roa, N. Dharani, A. S. Raghava, J. Buvanambigai and K. Sathish, "College Admission Predictor," Journal of Network Communications and Emerging Technologies (JNCET), vol. 8, no. 4, April 2018.
- [14] B. Ghai, "Analysis & Prediction of American Graduate Admissions Process," 2018.

2.3 PROBLEM STATEMENT DEFINITION

Create a web application that predicts a user's chances of admissions in the universities of their choice for PG Studies abroad.

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

Stuart Schmill, Dean of Admissions at the Massachusetts Institute of Technology (MIT), US, echoes this sentiment, emphasizing the importance of looking at the different degree structures on offer: The key thing for an applicant to understand is the kind of educational environment that will best suit how they like to learn.

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

Once you've weighed up all the factors and carefully made your decision, it's time for the really fun part: applying.

Though this might seem obvious, ensure you take care over this. You don't want to miss out simply because you forgot to submit the required evidence or applied too late. Should check entry requirements and deadlines before applying to make sure that they have the best possible chance of gaining a place on their chosen course.

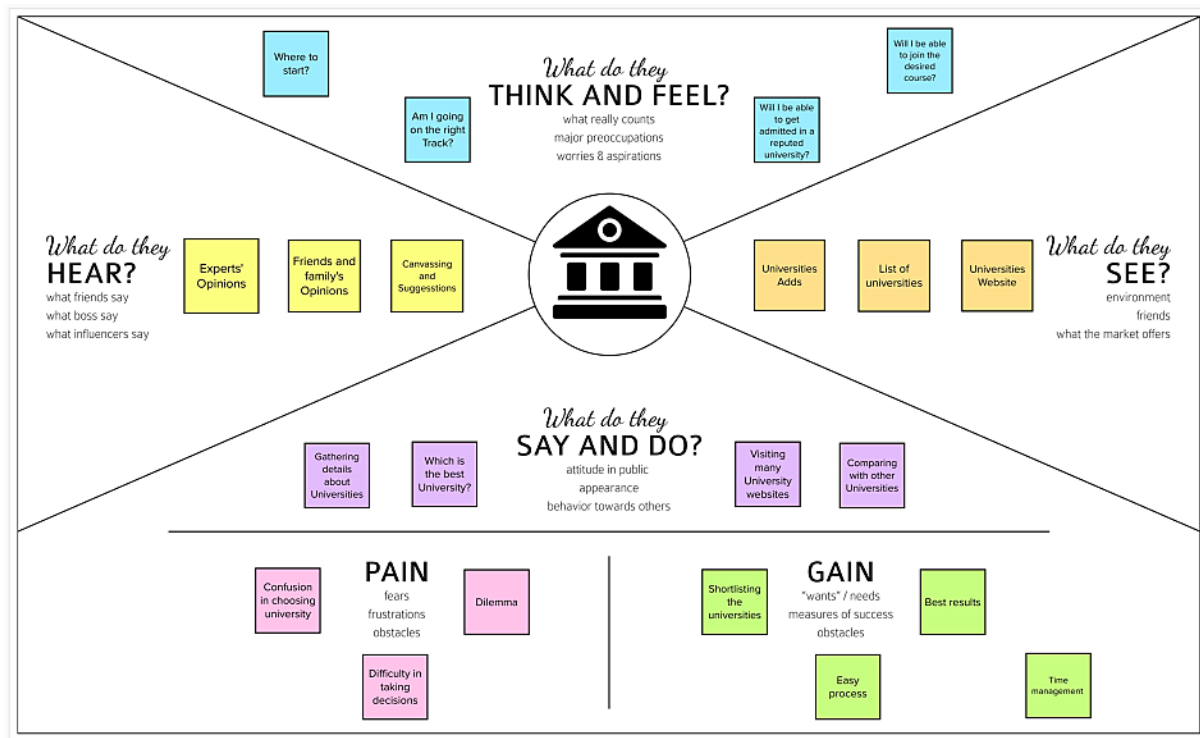
If they are unsure whether their qualifications are acceptable, they might like to contact the admissions office or international office in their chosen institution to check before submitting a full application.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is **a collaborative tool teams can use to gain a deeper insight into their customers**. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.




3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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

Reference: <https://www.mural.co/templates/empathy-map-canvas>

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



Brainstorm & idea prioritization

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

- 10 minutes to prepare
- 1 hour to collaborate
- 2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

- A Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
- B Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.
- C Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

[Open article](#) →


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 might we build an efficient solution for students to evaluate their chances of admit?



Key rules of brainstorming

To run a smooth and productive session

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

Step-2: Brainstorm, Idea Listing and Grouping

S

Deepika

- future-proof the model
- compare to preexisting work
- Deploy application on the web to facilitate easy use over various devices
- Enhance user experience by collecting early feedback on ease of use

Kaviya

- data visualisation to understand historical
- English statistical models to produce reproducible results for accurate prediction
- Analyse previous models and identify their drawbacks
- Enhance prediction reliability through prevalidated data

Kowsalya,Pavithra

- Innovative solution to address the given problem statement
- Additional data for better results by reducing bias
- Follow up on user satisfaction post deployment
- User friendly application

Sudharsanan

- Information will be presented in evaluation
- Personal user friendly interaction
- Analyse exceptions and outliers to build robust data model
- Quota for Disabled Persons and Scholarship

User Experience

- Prioritize user friendly interaction
- Deploy application on the web to facilitate easy use over various devices
- User friendly application

Analysis

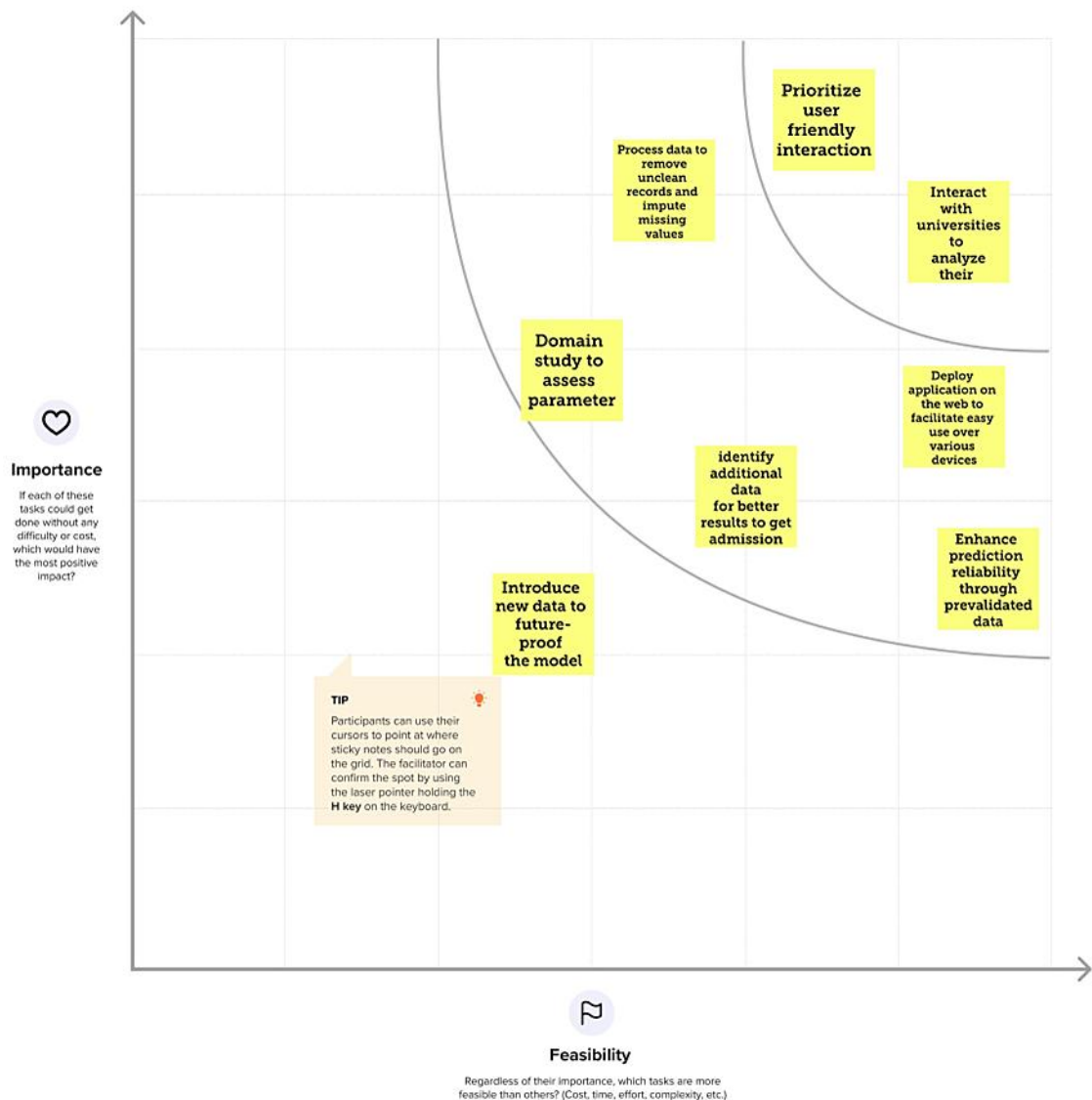
- Enhance prediction reliability through prevalidated data
- Process data to remove uniform records and inputs resulting values
- Domain study to assess parameter

System Modeling

- Introduce new data to future-proof the model
- Identify additional data for better results to get admission
- Interact with universities to analyse their

TP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as they surface to you mind.

Step-3: Idea Prioritization



3.3 PROPOSED SOLUTION

3.3.1 Problem Statement

A student who is trying to choose a right college based on score but I couldn't because I am not aware of eligibility criteria which makes me feel frustrated.

3.3.2 Idea / Solution description

This idea helps students to get the list of college to which they can apply as the system shortlist the colleges by comparing the students' marks and cut off to predict the possibility of admission or not.

3.3.3 Novelty / Uniqueness

The main advantages of the project are the computerization of the entrance seat allotment process. This makes the process easier and takes very less time when compared to the existing system.

3.3.4 Social Impact / Customer Satisfaction

It helps the students for making decision for right college. Here the chance of occurring errors is comparatively less. It is fast and efficient. Avoid the data repetition and inconsistency. It helps you to understand as to how your profile can be further improved to secure an admit in your preferred college. It can guide you whether you need to take GRE or not, in order to improve your chances of admit in your target university. It is user friendly.

3.3.5 Business Model (Revenue Model)

Like most areas of the educational world these days, technology is forcing higher educational institutions to do more with less. Institutions are under increasing pressure to admit more students, retain these students, and do their best to ensure the students success. This project not only relieves students from pressure but also helps these institutions to get relieved from their pressure.

3.3.6 Scalability of the Solution

This will also help you to finalize your dream university with a realistic roadmap, with the help of your information coupled with a bit of reality check on your academic scores such as GRE, TOFEL, etc., working experience and your CGPA. On the other hand, we have connoisseurs who shall work with you to amplify your prospects of receiving offers by ensuring that the universities that you apply, do not digress from your profile, and chiefly your ambition. The concern is privacy and ownership for both students and teachers.

3.4 Problem Solution Fit

DefineCS,frinnoCC	1. CUSTOMER SEGMENT(S) CS Who is your customer? i.e. working parents of 0-5 y.o. kids <div>Students who have completed HSC.</div>	6. CUSTOMER CONSTRAINTS CC 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. <div>Waste of Money, Time and Energy due to Poor Knowledge.</div>	5. AVAILABLE SOLUTIONS AS 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? <div>Seat allotment, Eligibility Criteria due to Entrance Exam like NEET, JEE, etc.,.</div>	ExploreAS,differentiate
	2. JOBS -TO -BE -DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <div>Want to get admission in preferred University.</div>	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? i.e. customers have to do it because of the change in regulations. <div>Due to high Competition and not meeting the required criteria.</div>	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) <div>At the Last Moment Spend more Money and Time to get into the preferred University.</div>	
	3. TRIGGERS T What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. <div>Watching Other Students trying into getting into one's Favourite University.</div>	10. YOUR SOLUTION SL 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. <div>We would create an application that helps students to get the list of colleges by comparing the student's marks and college's cut off and predicting admission probability. It is fast, efficient and reliable. It helps you to understand as to how your profile can be further improved to secure an admit in your preferred university.</div>	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 <div>They will search online about the preferred university and the criteria to join the University</div> 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. <div>Visiting College campus, Enquire students, Academic representatives and nearby people about the University.</div>	
Identifystron&TR&FM	4. EMOTIONS: BEFORE / AFTER EM 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. <div>Confused, Anxious over whether one getting Admission in the University.</div>			ExtraOnline & OfflineCH an dBE

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

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

FR No.	Functional Requirement	Process
FR-1	User Interface	A beautiful interface will be created with UX Research in mind to give users the best possible User Interface and Experience
FR-2	User Enters Marks	The user will enter the marks of their Grade 12 board exam
FR-3	Frontend To Backend	This mark will be forwarded to the backend
FR-4	Algorithm	The algorithm for this particular program will take the inputs and process it.
FR-5	Pre-Defined CSV File	The algorithm will fetch the data from the predefined CSV file which contains the list of Universities.
FR-6	Data Processing	Now this data will be processed by using Applied Data Science method
FR-7	Specified Model	This ADS method will also train the data by using a specified model for better predictions.
FR-8	Backend To Frontend	Now the data that is fetched by the Algorithm is now transmitted from Backend to Frontend User Interface

4.2 Non-functional Requirements:

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

NFR No.	Non-Functional Requirement	Description
NFR-1	Reliability	<ul style="list-style-type: none"> • The system shall be completely operational all hours of the day unless system failure or upgradation work is to be performed. • Down time after a failure shall not exceed 24 hours.
NFR-2	Usability	<ul style="list-style-type: none"> • No training is required to use the website. The form, home, about, FAQ and analysis pages load up within 10 seconds. • The results from the predictor should not take more than 30 seconds.
NFR-3	Performance	<ul style="list-style-type: none"> • The system can support any number of users at a time. • The mean time to view a web page over a 56Kbps modem connection shall not exceed 5 seconds.
NFR-4	Security	<ul style="list-style-type: none"> • The system shall provide password protected access to the website to all users, students and admins both.
NFR-5	Supportability	<ul style="list-style-type: none"> • The system will be able to incorporate more features without major reengineering. • The system web site shall be viewable from Internet Explorer 4.0 or later, Netscape Navigator/Communicator 3.0 or later and the America Online web browser version 3.0 or later.
NFR-6	Online User Documentation And Help	<ul style="list-style-type: none"> • The system shall provide a web page that explains how to navigate the site. This page should be customized based on what pages that user is allowed to access. • This help page should be accessible from all other pages.
NFR-7	Purchased Components	<ul style="list-style-type: none"> • No purchased components.
NFR-8	Interfaces	<ul style="list-style-type: none"> • The system must interface with the MongoDB database and the

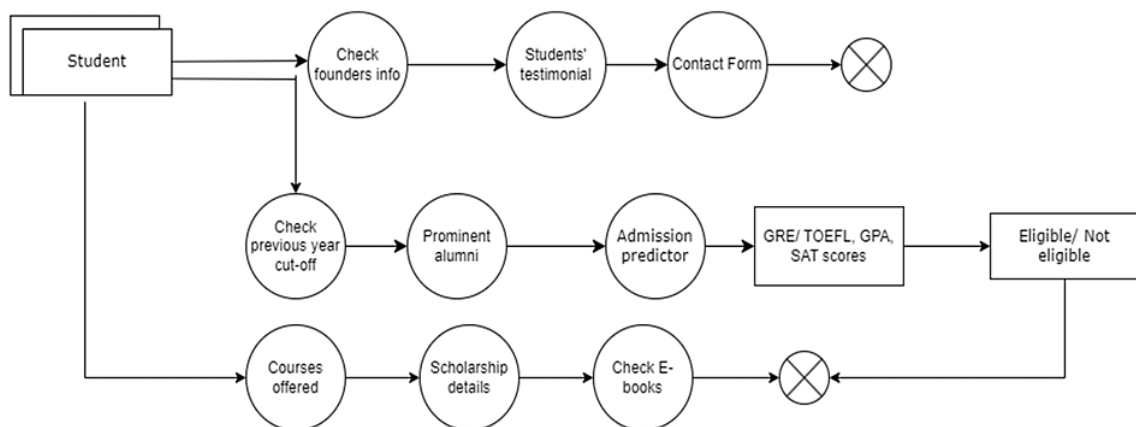
		web search engine it will be launched on.
--	--	---

CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagram

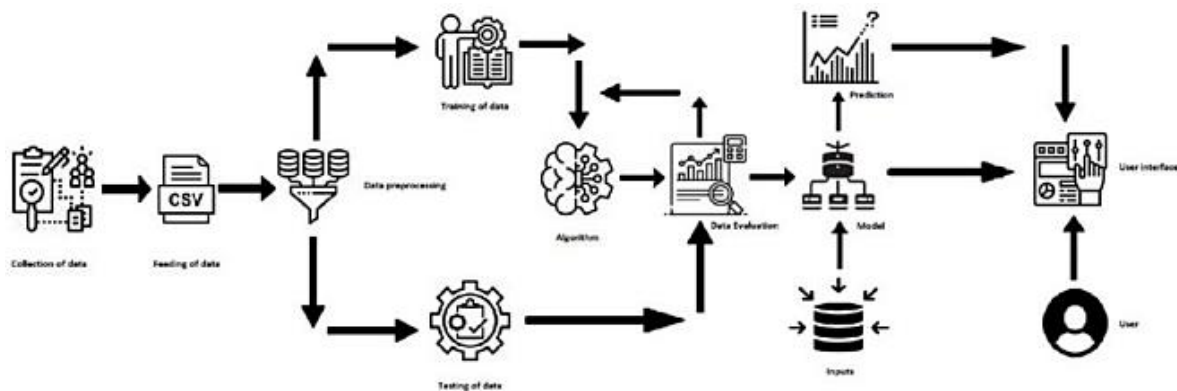
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD depicts the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution and Technical Architecture

5.2.1 Solution Architecture:

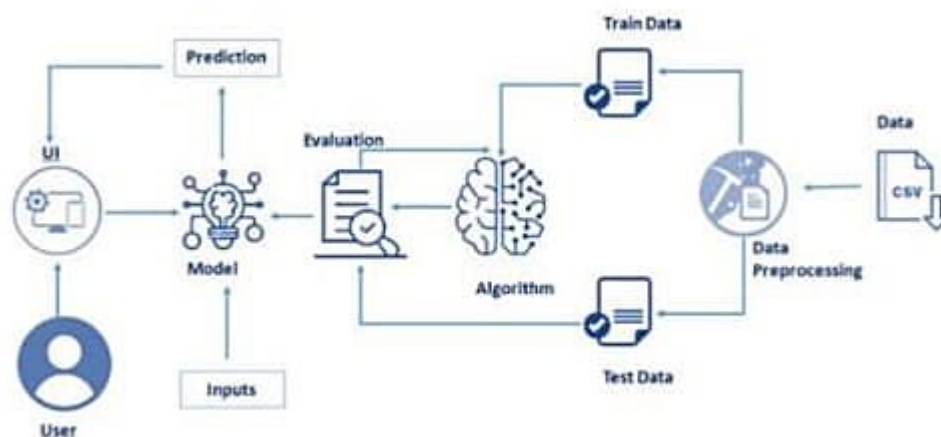
Solution architecture is a practice to provide ground for software development projects by tailoring IT solutions to specific business needs and defining their functional requirements and stages of implementation. It is comprised of many subprocesses that draw guidance from various enterprise architecture viewpoints.



5.2.2 Technical Architecture:

Technical Architecture (TA) is **a form of IT architecture that is used to design computer systems**. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

+



5.3 User Stories

User Stories	Functional Requirement	User Story Number	User Story/Task	Acceptance Criteria
Student	Home	USN-1	As a User, I will be able to view the Details of the Predictor.	Can be able to create an account.
	Details Form	USN-2	I have to fill the Form with Academic Details and provide information to the Predictor.	Can register and access the predictor.
University	Analysis	USN-3	The Form Results are sent to the Model to predict the Possibilities and Results are Obtained.	Can predict the results.
	Result	USN-4	I have to be able to view the Results of the Prediction.	Can view the results.
		USN-5	I have to be able to view Alternate Universities if the Chances of My Preferred University is very less.	Can choose the alternate universities

Student	Feedback	USN-6	After Completing my Prediction, I have to Share a Feedback to the Prediction for further Improvements.	Can be able to fill the feedback the form
	Share To Friends	USN-7	I will be able to share the Predictor to Others.	Can be able to share the predictor.
Administrator Services	Model Generation	USN-8	I have to train a Model that is able to predict the Chances of Admission Accurately.	Can be able to train the model
	Model Updating	USN-9	I have to change the Existing Model with a new Model as it gets Outdated as time passes.	Can be able to change the model
	Technical Improvement	USN-10	I should be able to view Feedbacks and Try to resolve their Query.	Can be able to resolve the queries.

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning and Estimation

Product backlog, sprint schedule and estimation

Use the below template to create product backlog and sprint schedule

S.NO	MILESTONE	ACTIVITIES	DATE
1.	Preparation Phase	Pre-requisites	23 Aug 2022
		Prior knowledge	24 Aug 2022
		Project Structure	23 Aug 2022
		Project Flow	23 Aug 2022
		Project Objectives	22 Aug 2022
		Registrations	26 Aug 2022
		IBM Cognossignin	27 Aug 2022
2.	Ideation Phase	Literature Survey	29 Aug 2022 - 03 Sept 2022
		Empathy Map	05 Sept2022 – 8 Sept 2022
		Problem Statement	09 Sept 2022- 11 Sept2022

	Ideation	12 Sept 2022 - 16 Sept2022
--	----------	----------------------------

3.	Project Design Phase-I	Proposed Solution	19 Sept 2022 - 23 Sept2022
		Problem Solution Fit	25 Sept 2022 - 27 Sept2022
		Solution Architecture	28 Sept 2022- 30 Sept 2022
4.	Project Design Phase-II	Customer Journey	05 Oct2022 - 08 Oct 2022
		Requirement Analysis	09 Oct 2022 - 11 Oct 2022
		Data FlowDiagrams	11 Oct 2022 - 13 Oct 2022
		Technology Architecture	14 Oct 2022- 15 Oct 2022
5.	Project Planning Phase	Milestones &Tasks	17 Oct2022 - 19 Oct 2022
		Sprint Schedules	20 Oct2022 - 22 Oct 2022
6.	Project Development Phase	Sprint-1	24 Oct2022 - 30 Oct 2022
		Sprint-2	01 Oct2022 - 05 Nov 2022
		Sprint-3	07 Nov 2022- 12 Nov 2022
		Sprint-4	13 Nov 2022 - 18 Nov2022

Sprint	Sprint-1	<ul style="list-style-type: none"> • Download the Dataset • Register IBM Cognos 	24 Oct 2022– 30 Oct 2022
		<ul style="list-style-type: none"> • Login • Working with dataset • Upload the dataset 	

	Sprint – 2	<ul style="list-style-type: none"> • Exploratory data • Visualization • Remove the unnecessary details 	30 Oct 2022 – 05 Nov 2022
	Sprint – 3	<ul style="list-style-type: none"> • Predicting based on the datasheet. • Top10 Sales by Year Using Line Graph • Top10 Revenue by Year Using Line Graph • Dashboard Creation 	06 Nov 2022 –11 Nov 2022
	Sprint – 4	<ul style="list-style-type: none"> • Summary Cards of Total Revenue, Sales, Stock, Price • Report 	13 Nov 2022 –18 Nov 2022

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement	User Story Number	User Story Task	Priority
Sprint-3	Home	USN -1	As a User, I will be able to view the Details of the Predictor	Low
Sprint-2	Details Form	USN -2	I have to fill the Form with Academic Details and provide information to the Predictor.	Medium

Sprint-1	Analysis	USN -3	The Form Results are sent to the Model to predict the Possibilities and Results are Obtained	High
Sprint-2	Result	USN -4	I have to be able to view the Results of the Prediction	Medium
Sprint-3		USN -5	I have to be able to view Alternate Universities if the Chances of My Preferred University is very less.	Medium
Sprint-4	Feedback	USN -6	After Completing my Prediction, I have to Share a Feedback to the Prediction for further Improvements	Low
Sprint-4	Share to Friends	USN -7	I will be able to share the Predictor to Others	Low
Sprint-1	Model Generation	USN -8	I have to train a Model that is able to predict the Chances of Admission Accurately	High
Sprint-1	Model Updating	USN -9	I have to change the Existing Model with a new Model as it gets Outdated as time passes.	Medium
Sprint-4	Technical Improvement	USN -10	I should be able to view Feedbacks and Try to resolve their Query.	Low

Project Tracker, Velocity& Burndown Chart:

Sprint	Duration	Sprint StartDate	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint ReleaseDate(Actual)
Sprint-1	6 Days	24 Oct 2022	29 Oct 2022	30	29 Oct 2022
Sprint-2	6 Days	31 Oct 2022	05 Nov 2022	14	05 Nov 2022
Sprint-3	6 Days	07 Nov 2022	12 Nov 2022	15	12 Nov 2022
Sprint-4	6 Days	14 Nov 2022	19 Nov 2022	21	19 Nov 2022

Velocity:

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

$$AV = \text{Sprint duration} / \text{Velocity}$$

SPRINT	Total Story Points	Duration	Average Velocity
Sprint-1	30	6 Days	30/6=5
Sprint-2	14	6 Days	14/6=2.33
Sprint-3	15	6 Days	15/6=2.5
Sprint-4	21	6 Days	21/6=3.5
Total	80	24 Days	80/24=3.33

Burndown Chart:

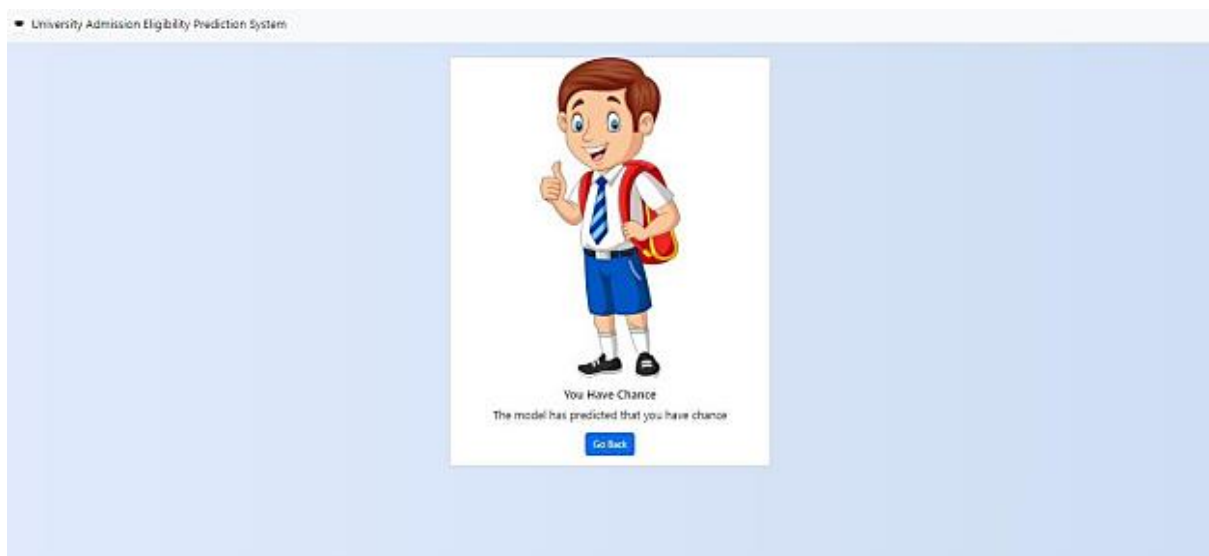
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

CHAPTER 7

CODING & SOLUTION

7.1 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:

```

{% extends 'index.html' %}

{% block body %}

<div class="container text-center p-4">

<div class="d-flex justify-content-center">

<div class="card" style="width: 34rem;">



<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>

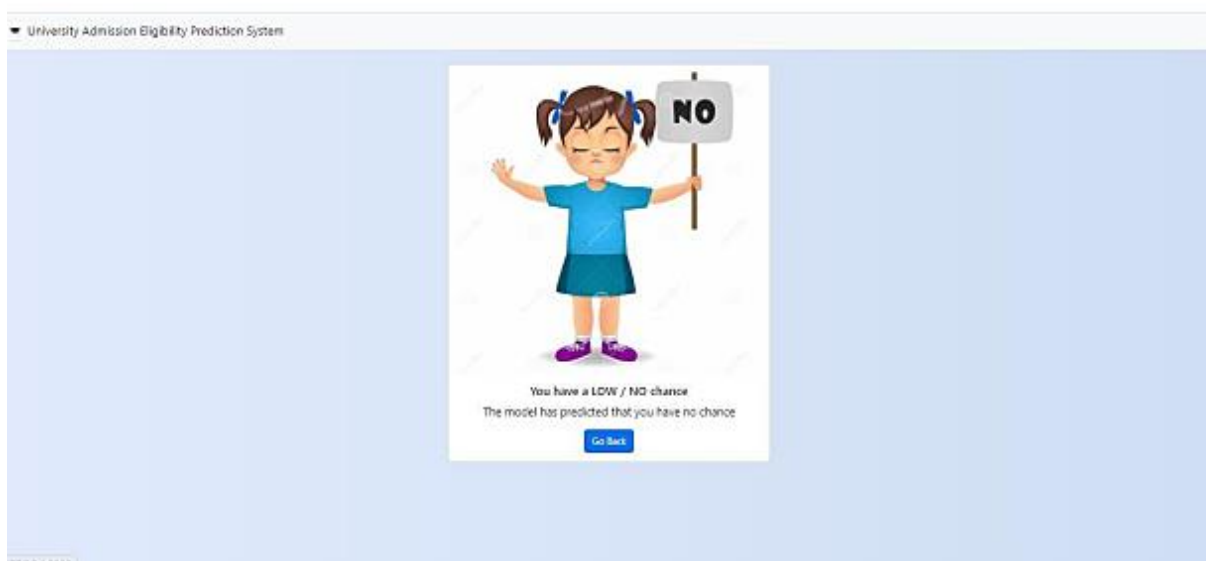
</div>

{% endblock %}

```

7.2 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:

```

{% extends 'index.html' %}

```

```
{% block body %}

<div class="container text-center p-4">
  <div class="d-flex justify-content-center">
    <div class="card" style="width: 34rem;">
      
      <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>





{% end block %}
```

7.3 Database Schema

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

401 lines (401 sloc) | 12.6 KB

RawBlame



Q Search this file...

1	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
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

CHAPTER 8

TESTING

8.1 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

a. 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).

1. 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 Reproduce D	0	0	0	0	0
Skipped	0	0	1	1	2
Wont Skipped	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

CHAPTER 9

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
```

CHAPTER 10

ADVANTAGES & DISADVANTAGES

10.1 Advantages

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

5. Very user-friendly.
6. Easy accessibility of data.

10.2 Disadvantages

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

CHAPTER 11

CONCLUSION

This system, being the first we have created in Python using ML algorithms and other front end languages such as html, CSS, java script, has proven more difficult than originally imagined. While it may sound simple to fill out a few forms and process the information, much more is involved in the selection of applicants than this.

Every time progress was made and features were added, ideas for additional features or methods to improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself. Debugging can sometimes be a relatively straight forward process, or rather 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

CHAPTER 12

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.

CHAPTER 13

APPENDIX

13.1 Source Code

PYTHON CODE

Uploading the python code,

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 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(servicename='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 accept body 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 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 [3]:

```
data.drop(["SerialNo."],axis=1,inplace=True)
```

In [4]:

```
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
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 ()
```

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	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()
```

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='TOEFL  
Score',color="BLUE")  
plt.xlabel("GRE Score")  
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()
```

Out [10]

In [11]:

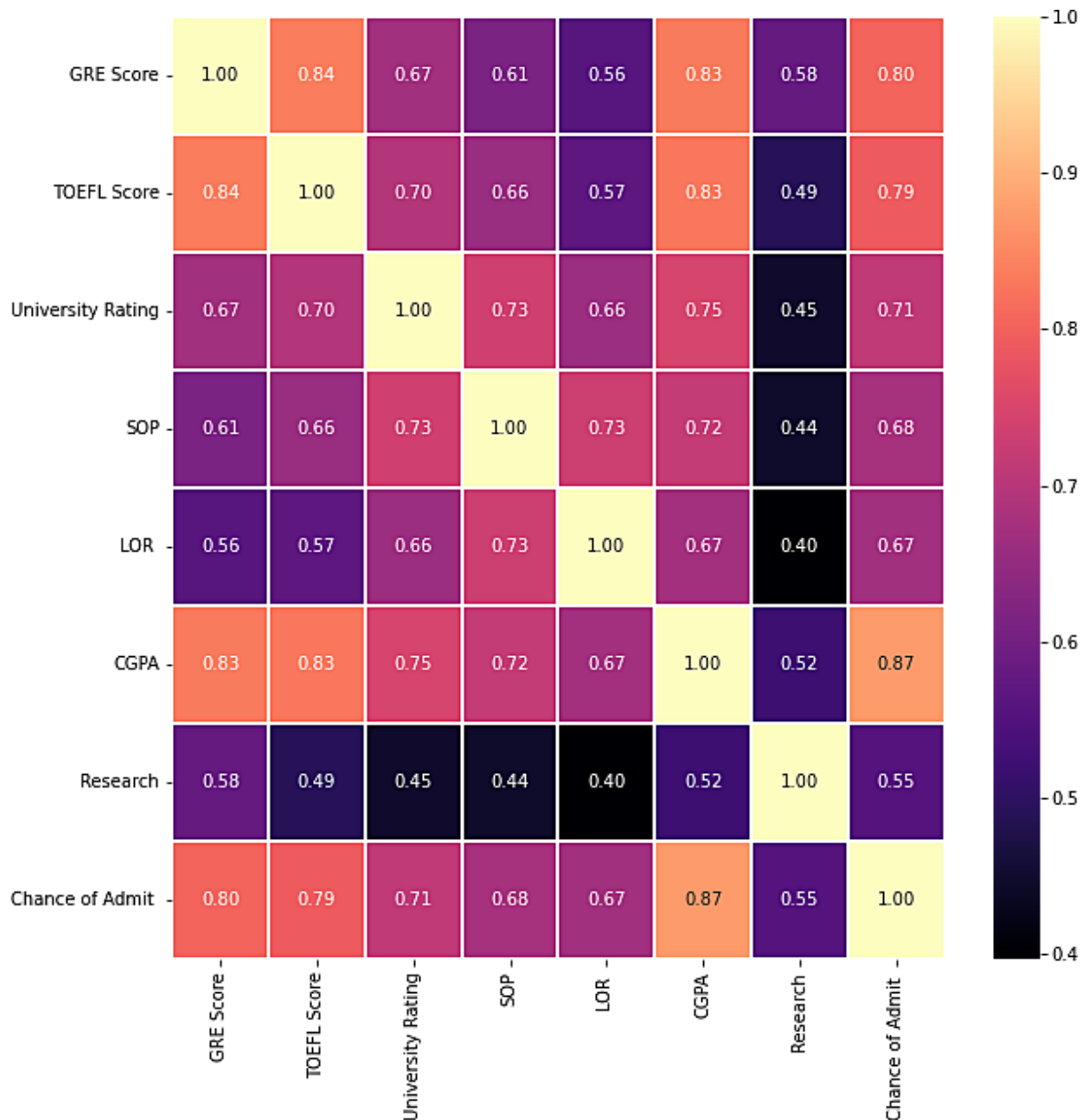
```
p = np.array([data["TOEFL Score"].min(),data["TOEFLScore"].mean(),data["TOEFL  
Score"].max()])  
r = ["Worst","Average","Best"] plt.bar(p,r)  
plt.title("TOEFL Scores")  
plt.xlabel("Level")  
plt.ylabel("TOEFL Score")  
plt.show()
```

Out [11]:

In [12]:

```
plt.figure(figsize=(10, 10))
sns.heatmap(data.corr(), annot=True, linewidths=0.05,
fmt='.2f',cmap="magma")
plt.show()
```

Out [12]:

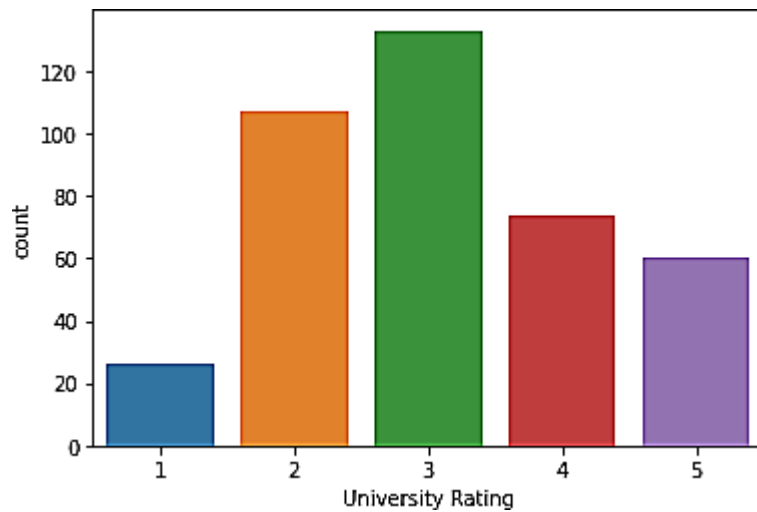


In [14]:

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

Out[14]:

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

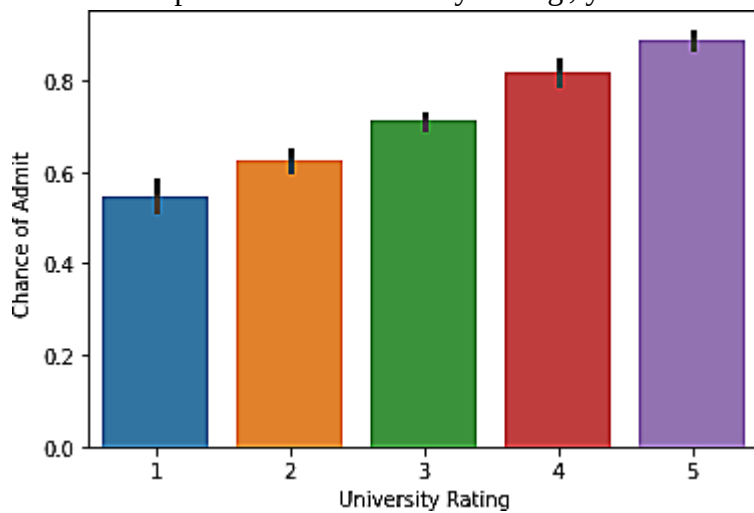


In [15]:

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

Out [15]:

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

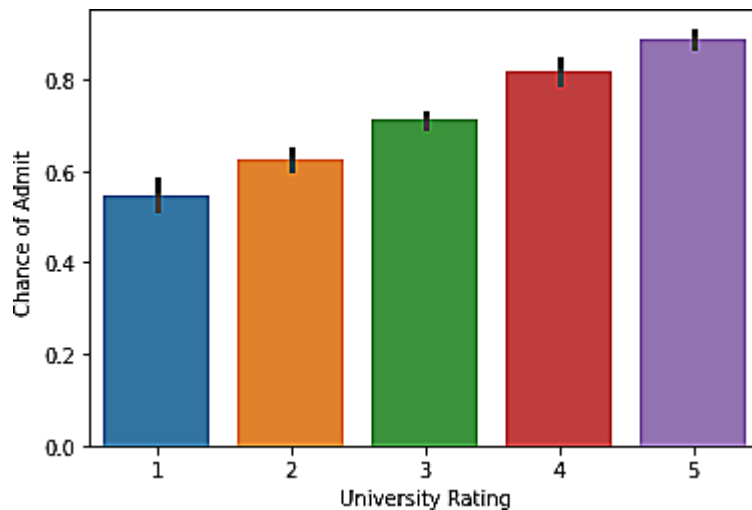


In [15]:

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

Out [15]:

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



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 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]:

```
0.721402171514154
```

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 Squared Error: 0.08481918200196631
```

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))
```

Out [25]:

```
Accuracy Score: 0.9166666666666666
Recall Score: 1.0
ROC AUC Score: 0.7222222222222222
Confussion Matrix:
```

```
[[ 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)

HTML CODES

Uploading Html codes

Chance.html

```
{% extends 'index.html' %}
```

```
{% block body %}
```

```
<div class="container text-center p-4">
```

```
<div class="d-flex justify-content-center">
```

```
<div class="card" style="width: 34rem;">
```

```

```

```
<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>
```

```
{% endblock %}
```

Demo2.html

```
{% extends 'index.html' %}

{% block body %}

<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 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. </p>
<div class="d-flex justify-content-right">

</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="the Form">
<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>
{% endblock %}

```


Index.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="sha384Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URyB
v1WTRi" 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="/">

 University Admission Eligibility Prediction System </a>
</div>

</nav>

{% block body %}

<h1> Index Page </h1>

{% endblock %}

<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>
```

Nochance.html

```
{% extends 'index.html' %}

{% block body %}

<div class="container text-center p-4">
<div class="d-flex justify-content-center">
<div class="card" style="width: 34rem;">

<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>

{% endblock %}
```

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;

```

Styles.css

```

* {margin: 0;
padding: 0;
border: 0;
}

{

font: 62.5%/1.5 "Lucida Grande", "Lucida Sans", Tahoma, Verdana, sans-serif;
background: #e0eafc;

background: -web kit-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);

}

Footer
```

PYTHON CODE

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 inrequest.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 = "poJ22ua6BCG9qY33B8fkgnz1bnP1f9DZqUIF9NkBM1bZ"

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]

                }]

}
```


OUTPUT IMAGES:

127.0.0.1:5000/home

University Admission Eligibility Prediction System

Enter your details and get probability of your admission

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



Enter the details

GRE Score:

TOFEL Score:

University Rating:

SOP:

LOR:

CGPA:

Research:

☐ Yes

☒ No


Predict

127.0.0.1:5000/home

University Admission Eligibility Prediction System

Enter your details and get probability of your admission

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



Enter the details

GRE Score:279

TOFEL Score:68

University Rating:3.50

SOP:2.50

LOR:4.51

CGPA:8.70

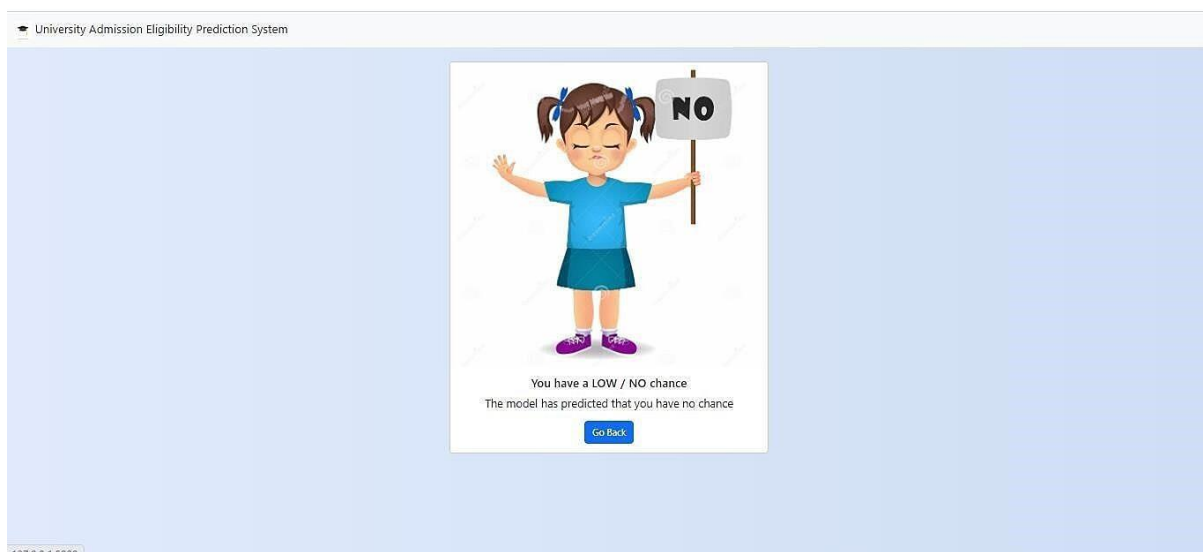
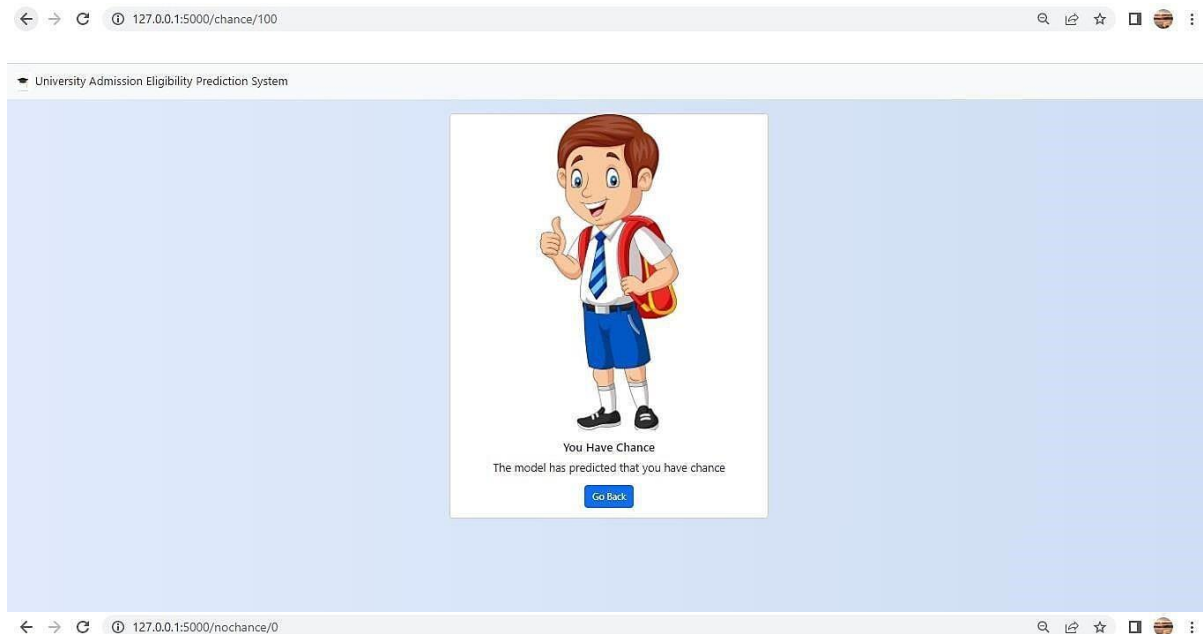
Research:

☐ Yes

☒ No

Predicting...

127.0.0.1:5000



13.2 GitHub Link

<https://github.com/IBM-EPBL/IBM-Project-10469-1664355350>

13.3 Project Demo Link

<https://youtu.be/4TLsHvUKv1g>