**COLLEGE RECOMMENDATION SYSTEM**

A PROJECT REPORT

BY **TEAM NO. 6**

TEAM MEMBER 1 (E23CSE0381)

TEAM MEMBER 2 (E23CSE0383)

TEAM MEMBER 3 (E23CSE0371)

TEAM MEMBER 4 (E23CSE0372)

A blue and red text on a black background

Description automatically generated

SUBMITTED TO

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

April 2025

# DECLARATION

I/We hereby declare that the work which is being presented in the report entitled “College Recommendation System”, is an authentic record of my/our own work carried out during the period from JAN, 2025 to April, 2025 at School of Computer Science and Engineering and Technology, Bennett University Greater Noida.

The matters and the results presented in this report has not been submitted by me/us for the award of any other degree elsewhere.

Signature of Candidate

(MOKSH SHARMA)

MOKSH SHARMA

(Enroll. No. E23CSE0381)

LAKSHAY GUPTA

(Enroll. No. E23CSE0383)

YASH VERMA

(Enroll. No. E23CSE0372)

ANSH JAIN

(Enroll. No. E23CSE0371)

# ACKNOWLEDGEMENT

I/We would like to take this opportunity to express my/our deepest gratitude to my/our mentor, **Dr. Sanchali Das** for guiding, supporting, and helping me/us in every possible way. I/we was/were extremely fortunate to have him as my/our mentor as he provided insightful solutions to problems faced by me/us thus contributing immensely towards the completion of this capstone project. I/We would also like to express my/our deepest gratitude to VC, DEAN, HOD, faculty members and friends who helped me/us in successful completion of this capstone project. Any other name you can mentioned here. (Acknowledgement is your personal view, so you can write it in your way by maintaining integrity of technical report).

Signature of Candidate

(MOKSH SHARMA)

MOKSH SHARMA

(Enroll. No. E23CSE0381)

LAKSHAY GUPTA

(Enroll. No. E23CSE0383)

YASH VERMA

(Enroll. No. E23CSE0372)

ANSH JAIN

(Enroll. No. E23CSE0371)

**TABLE OF CONTENTS**

[LIST OF TABLES viii](#_Toc195542334)

[LIST OF FIGURES ix](#_Toc195542335)

[ABSTRACT x](#_Toc195542336)

[1. INTRODUCTION 1](#_Toc195542337)

[1.1. Problem Statement 1](#_Toc195542338)

[2. Background Research 2](#_Toc195542339)

[2.1. Proposed System 2](#_Toc195542340)

[2.2. Goals and Objectives 2](#_Toc195542341)

[3. Project Planning 3](#_Toc195542342)

[3.1. Project Lifecycle 3](#_Toc195542343)

[3.2. Project Setup 3](#_Toc195542344)

[3.3. Stakeholders 4](#_Toc195542345)

[3.4. Project Resources 4](#_Toc195542346)

[3.5. Assumptions 5](#_Toc195542347)

[4. Project Tracking 6](#_Toc195542348)

[4.1. Tracking 6](#_Toc195542349)

[4.2. Communication Plan 6](#_Toc195542350)

[4.3. Deliverables 8](#_Toc195542351)

[5. SYSTEM ANALYSIS AND DESIGN 9](#_Toc195542352)

[5.1. Overall Description 9](#_Toc195542353)

[5.2. Users and Roles 10](#_Toc195542354)

[5.3. Design diagrams/ UML diagrams/ Flow Charts 11](#_Toc195542355)

[5.3.1. Use Case Diagram 11](#_Toc195542356)

[5.3.2. Class Diagram 12](#_Toc195542357)

[5.3.3. Activity Diagrams 13](#_Toc195542358)

[5.3.4. Sequence Diagram 14](#_Toc195542359)

[5.3.5. Data Architecture 14](#_Toc195542360)

[6. User Interface 15](#_Toc195542361)

[6.1. UI Description 15](#_Toc195542362)

[6.2. UI Mockup 15](#_Toc195542363)

[7. Pseudo Code OF CORE FUNCTIONALITY 16](#_Toc195542364)

[8. Project Closure 17](#_Toc195542365)

[8.1. Goals / Vision 17](#_Toc195542366)

[8.2. Delivered Solution 18](#_Toc195542367)

[8.3. Remaining Work 18](#_Toc195542368)

[REFERENCES 19](#_Toc195542369)

LIST OF TABLES

Table Page

[Table 1: Goal and Objectives 2](#_Toc195491686)

[Table 2: Project Setup 3](#_Toc195491687)

[Table 3: Stakeholders 4](#_Toc195491688)

[Table 4: Project Resources 4](#_Toc195491689)

[Table 5: Assumptions 5](#_Toc195491690)

[Table 6: Project Tracking 6](#_Toc195491691)

[Table 7: Regularly Scheduled Meetings 6](#_Toc195491692)

[Table 8: Information To Be Shared Within Our Group 7](#_Toc195491693)

[Table 9: Information To Be Provided To Other Groups 7](#_Toc195491694)

[Table 10: Information Needed From Other Groups 7](#_Toc195491695)

[Table 11: Deliverables 8](#_Toc195491696)

[Table 12: User and Roles 10](#_Toc195491697)

LIST OF FIGURES

Figure Page

[Figure 1: Use-case diagram 11](#_Toc195491552)

[Figure 2: Class diagram 12](#_Toc195491553)

[Figure 3: Activity diagram 13](#_Toc195491554)

[Figure 4: Sequence diagram 14](#_Toc195491555)

[Figure 5: Data Architecture 14](#_Toc195491556)

[Figure 6: UI Mockup 15](#_Toc195491557)

ABSTRACT

The process of choosing a college can be daunting for students because of the plethora of options and the difficulty of matching individual interests with educational objectives. In order to counter this issue, we created a College Selection Assistant—a web-based tool developed with the Design Thinking approach. Our objective was to design an intuitive, student-oriented tool that eases decision-making by providing customized college suggestions in accordance with user-specified criteria like location, stream, price, and ranking.

The project was kickstarted with an empathize stage, where we surveyed and interviewed students and parents to gauge their pain points. The information that we gathered helped us frame clean problem statements and inform the ideation process. During the subsequent design process, we made various low-fidelity prototypes and flowcharts to sketch out the user journey. These were iteratively improved upon, based on feedback from peers and mentors.

The solution was created using HTML, CSS, and JavaScript. A user input capture form collects user information, which is filtered through a filtering algorithm comparing user requirements against a predefined data set of colleges. The user interface is simple and displays a ranked list of suggested institutions with corresponding details. Static college data was utilized for demonstration purposes, and results were verified using user testing sessions.

The ultimate prototype effectively accomplishes its stated goals by streamlining college choice and enhancing user confidence in making well-informed decisions. This project illustrates the strength of user-centered design in educational resources and lays the foundation for future integration with real-time data feeds and sophisticated recommendation algorithms.

1. INTRODUCTION

The college search process is usually stressful and overwhelming to students and parents. There is a lack of straightforward, guided tools that take student interests and provide refined recommendations. Our project fills this gap by developing a College Selection Assistant web site following design thinking methodology.

* 1. Problem Statement

The students tend to be confused in making their college admissions choices because they face too many decisions, unreliable advice, and no friendly digital tools to assist them according to their requirements.

1. Background Research

Our early research involved looking at current college recommendation systems and conducting interviews with prospective users (parents and students). We observed that most systems were non-personalized and not intuitive. We researched Shiksha.com, CollegeDunia.com, and NIRF rankings. With insights from these, we created a system that enables interactive preference selection and smart filtering through basic algorithms.

* 1. Proposed System

We suggest a College Selection Assistant web application wherein users can input specific preferences (location, branch, fees, etc.) and get appropriate college suggestions. The tool is designed with empathy, usability, and ease of access in mind.

* 1. Goals and Objectives

Table 1: Goal and Objectives

|  |  |
| --- | --- |
| **#** | **Goal or Objective** |
| 1 | Make the system extensible – future updates that can be done easily |
| 2 | Make the system easy to support – provide good documentation, configuration/build files, administrator’s manual |
| 3 | Make the system very easy to use – users would agree that minimal to no training is needed |
| 4 | Build a prototype that demonstrates the user interface to get early feedback from the customer/users |
| 5 | Have fun working on the project |

1. Project Planning

This section covers the details of the project planning. Selecting the lifecycle of the development, project stakeholders, resources required, assumptions made (if any) are detailed in the sections below.

* 1. Project Lifecycle

We used an Agile lifecycle of multiple design and test iterations. Each sprint saw planning, prototyping, feedback, and iterative refinements.

* 1. Project Setup

Table 2: Project Setup

|  |  |
| --- | --- |
| **#** | **Decision Description** |
| 1 | HTML/CSS/JavaScript stack |
| 2 | GitHub for source control |
| 3 | Flowcharts and diagrams to visualize |

* 1. Stakeholders

Table 3: Stakeholders

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| Students | Primary consumers |
| Parents | Secondary consumers |
| University mentors | Project critics |
| Moksh Sharma | Team member |
| Lakshay Gupta | Team member |
| Yash Verma | Team member |
| Ansh Jain | Team member |

* 1. Project Resources

Table 4: Project Resources

|  |  |  |
| --- | --- | --- |
| **Resource** | **Resource Description** | **Quantity** |
| Database Server | A database server provided by the sponsoring company. | 1 |
| Capstone Team | Our team of students who will be the primary developers of the project. | 4 |
| Jim Somebody | The mentor who will be able to provide us with technical assistance. | 1 |
| Mac Workstation | An OS X workstation with X Code for developing the OS X version of the software. | 1 |
| Android Phone | An Android phone to be used as test hardware for the mobile version of the software. | 2 |

* 1. Assumptions

Table 5: Assumptions

|  |  |
| --- | --- |
| **#** | **Assumption** |
| A1 | The capstone team and mentors will be able to meet face to face once a week. |
| A2 | Users possess basic internet ability |
| A3 | College data will be static in prototype |
| A4 | Team will have sufficient time to complete a working model to present by mid-semester |
| A5 | Recommendations are suggestive rather than prescriptive |
| A6 | The development test data provided will be sufficient to create an accurate prediction of user actions |
| A7 | The models developed will be easily extended to other forms within the time frame |

1. Project Tracking
   1. Tracking

|  |  |
| --- | --- |
| **Information** | **Description** |
| Code Storage | Source control through GitHub |
| Bug Tracking | Bug tracking will be done with Trac. |
| Project Documents and Assignments | Weekly reports, specification and design documents, etc. |
| Continuous Integration | Continuous integration will be done. |

Table 6: Project Tracking

* 1. Communication Plan

Table 7: Regularly Scheduled Meetings

|  |  |  |
| --- | --- | --- |
| Meeting Type | Frequency/Schedule | Who Attends |
| Conference Call/Skype | Weekly | Project team and mentor |
| Team Meeting | Weekly | Project team |
| Short Meeting | Weekly in class | Project team |
| Sprint Planning Meeting | Start of each sprint | Project team and mentor |
| Sprint Retrospective Meeting | End of each sprint | Project team |
| Sprint Review Meeting | End of each sprint | Project team, ***mentor, and sponsor*** |

Table 8: Information To Be Shared Within Our Group

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Project team | Task assignments & General scrum information | Weekly | Team meetings, listing in Project Specification. |

Table 9: Information To Be Provided To Other Groups

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Sponsor and mentor | Final deliverables | At completion of project | Project specification doc., code, Power Point presentation |
| Sponsor and mentor | Weekly report | Weekly | Email and Trac site access |
| Sponsor and mentor | Project baselines ***(optional)*** | At the end of each sprint | Onsite customer demo, access to repository |

Table 10: Information Needed From Other Groups

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Sponsor and mentor | Requirement changes | Start of each sprint | Conference call or meeting with sponsor and mentor. |

* 1. Deliverables

Table 11: Deliverables

|  |  |
| --- | --- |
| **#** | **Deliverable** |
| 1 | Study results ***(if any)*** |
| 2 | Code |
| 3 | Test and test results |
| 4 | Build process documents***(if any)*** |
| 5 | Install process documents***(if any)*** |
| 6 | Administrator or user manual***(if any)*** |
| 7 | Postmortem document |
| 8 | Final report (final PowerPoint presentation, 3 minute video, and final sprint) |

1. SYSTEM ANALYSIS AND DESIGN

This section describes in detail about the design part of the system.

* 1. Overall Description

This project seeks to develop a web-based College Selection Assistant that assists students in selecting the appropriate educational institution according to their individual preferences. With a robust base in the Design Thinking methodology, the solution places user empathy at the center of its development. The tool enables users to enter parameters like desired location, course stream, fee range, and ranking order. From these inputs, the assistant applies a pre-defined set of colleges and suggests the most appropriate ones to the user in a clean and simple interface.

Technically, the project employs standard web technologies—HTML, CSS, and JavaScript—to construct the front end. Conditional filtering over a structured dataset (locally stored in JSON format) is applied by the recommender logic to mimic personalized recommendations. Flowcharts and design documents produced using Mermaid.js and other graphing libraries established the logic flow and enhanced the usability of the interface. The application was repeatedly prototyped and tested in order to incorporate user feedback along the way to ensure that the final product is both usable and functional.

Future developments of the system may involve integrating dynamic college databases via APIs, using machine learning models for more intelligent and adaptive recommendations, and adding support to mobile platforms. The simplicity of the existing prototype renders it extremely extensible and provides a good basis for further development according to contemporary ed-tech trends.

* 1. Users and Roles

Table 12: User and Roles

|  |  |
| --- | --- |
| **User** | **Description** |
| Student User | The primary end-user of the system who inputs their preferences such as course, location, and fee range to receive a personalized list of colleges. |
| Parent/Guardian | A secondary user who may assist or guide the student in making preference choices or reviewing recommended colleges. |
| System (Agent) | The internal logic that handles user inputs and sifts through college choices according to established rules and conditions. |
| Developer | Members who work on coding, interface building, logic development, and incorporation of design features. |
| Mentor/Evaluator | Guide or instructor offering critique, checking on progress, and determining the system's functionality and usability. |

* 1. Design diagrams/ UML diagrams/ Flow Charts
     1. Use Case Diagram

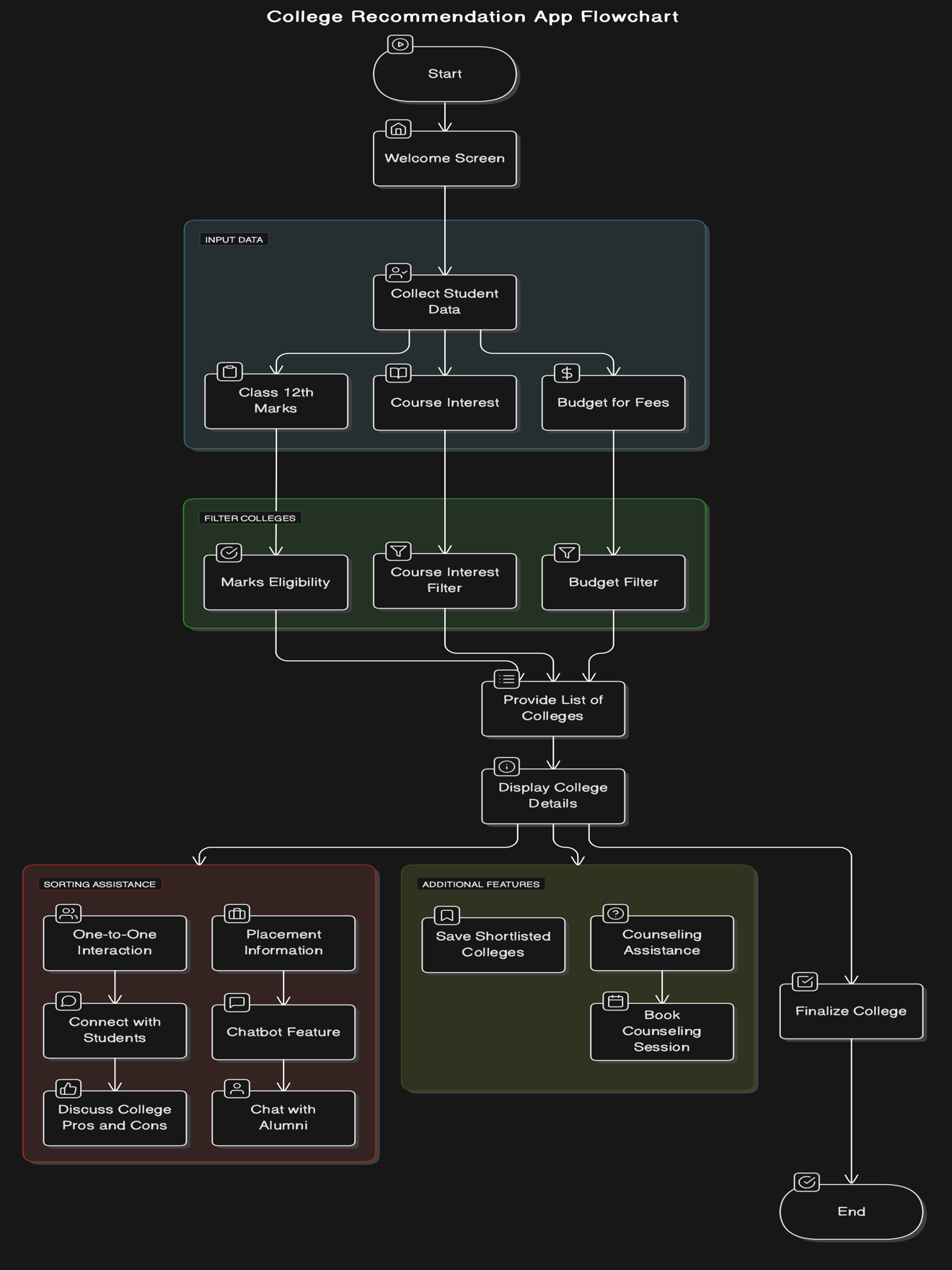


Figure : Use-case diagram

* + 1. Class Diagram

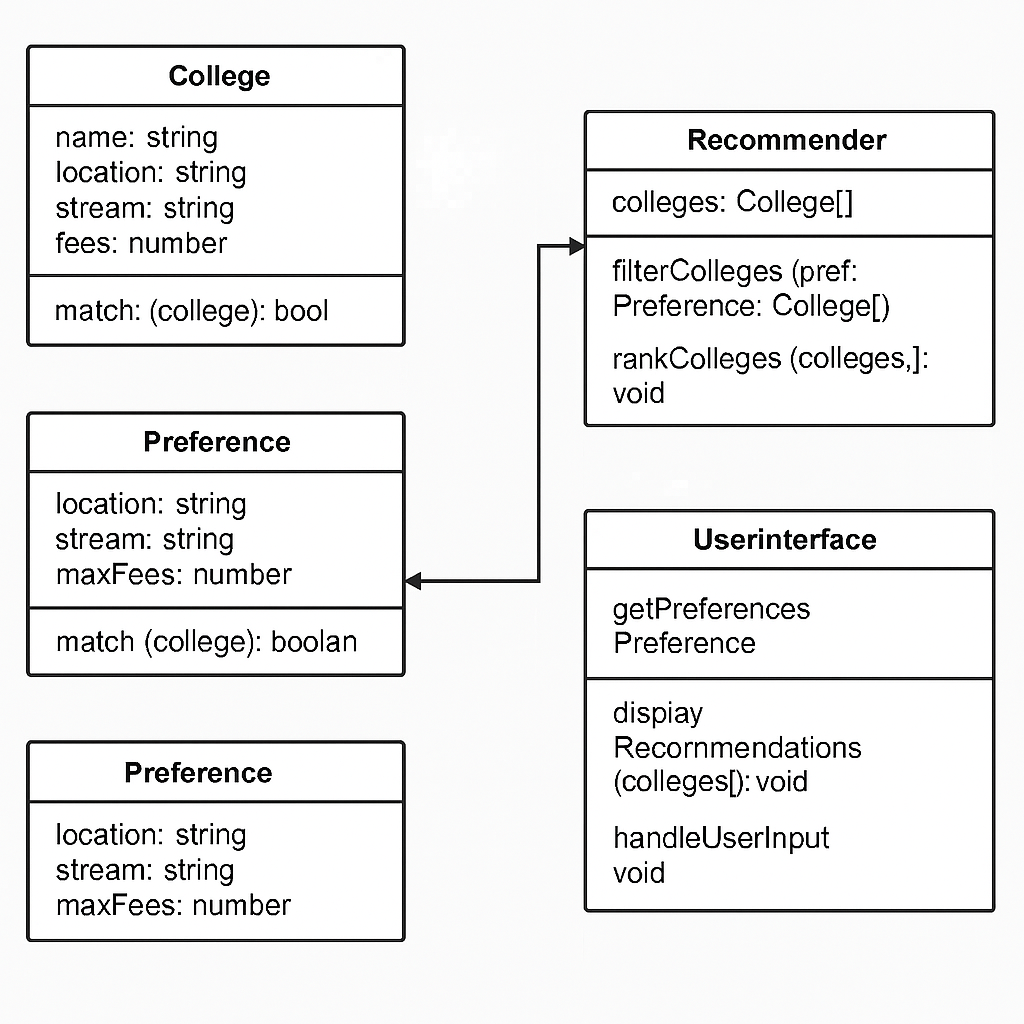


Figure : Class diagram

* + 1. Activity Diagrams

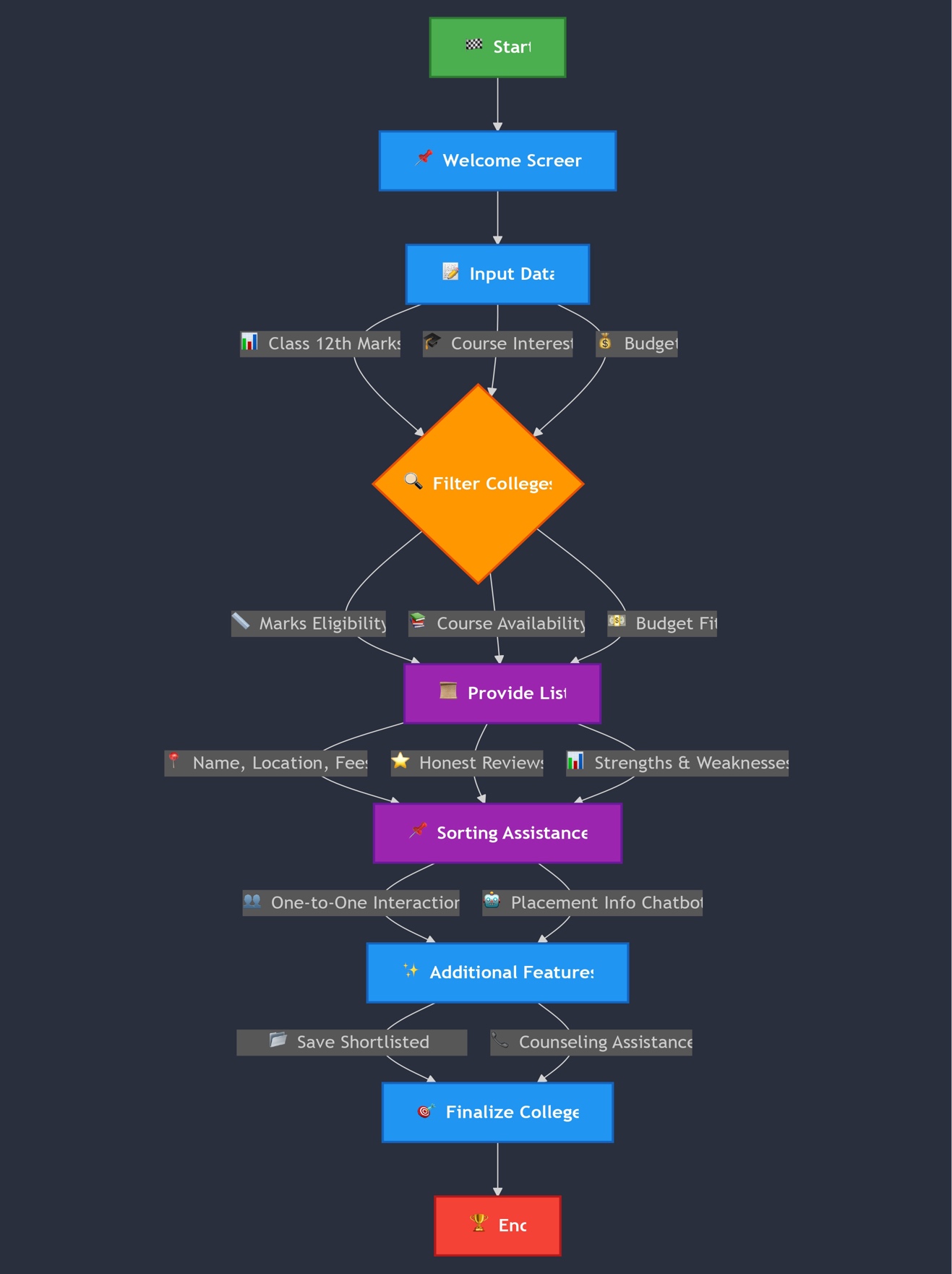


Figure : Activity diagram

* + 1. Sequence Diagram

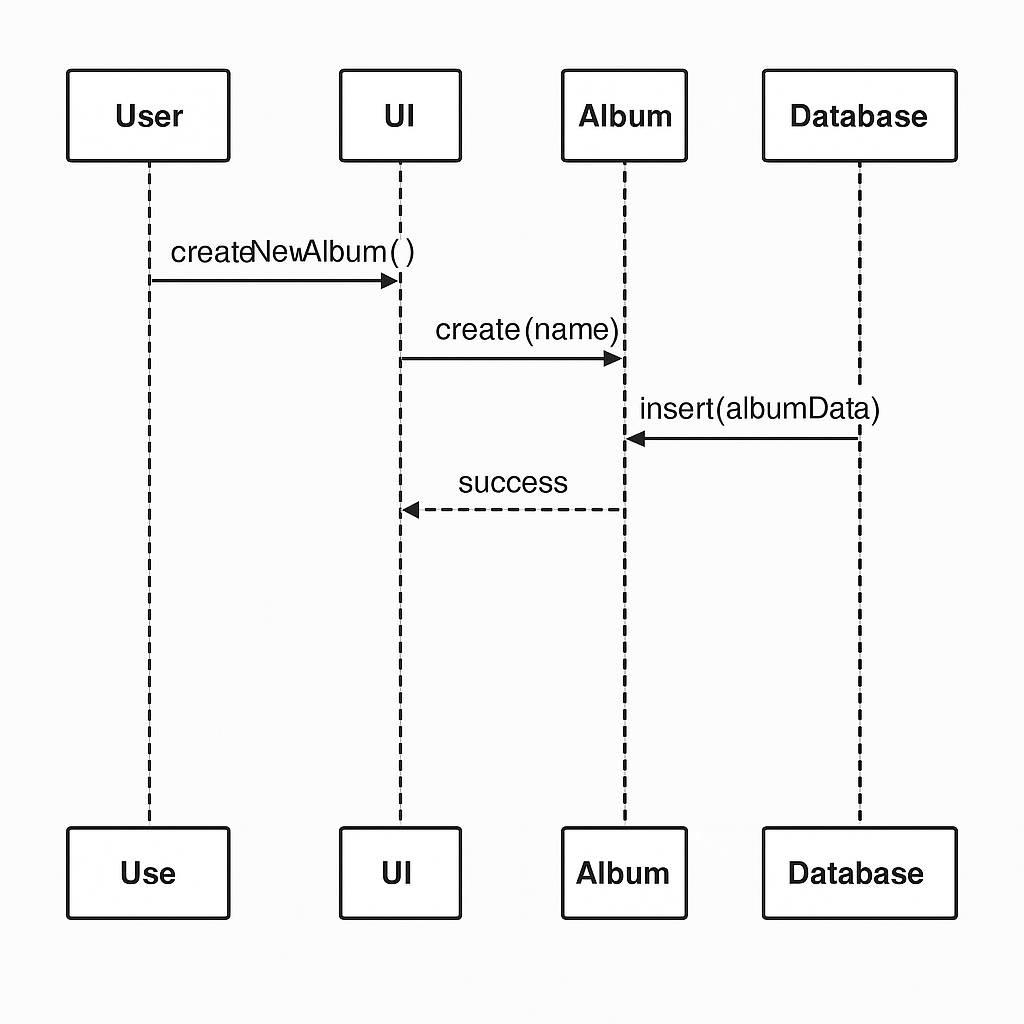


Figure : Sequence diagram

* + 1. Data Architecture

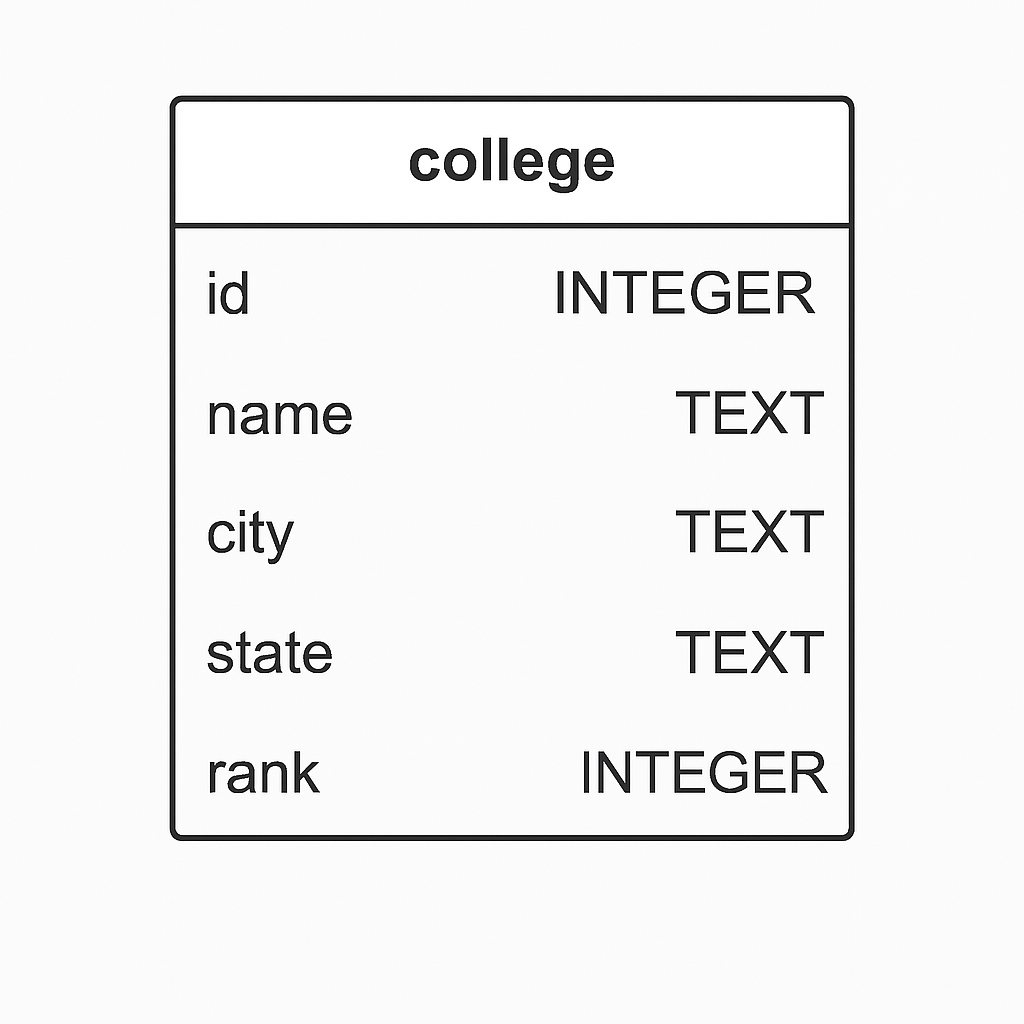


Figure : Data Architecture

1. User Interface
   1. UI Description

The College Selection Assistant is a web application coded with HTML, CSS, and JavaScript. The major user interface is browser-based with an interactive form where users mainly students enter their preferences. The UI is clean, responsive, and user-friendly to provide easy navigation on various devices, such as desktops and mobile phones. Interacting with the system is done through dropdown menus, sliders, and buttons. Upon submission of the preferences, a dynamically loaded list of suggested colleges is presented, sorted based on how well each college fits the user's criteria. There is no console interaction nor backend database in this version; everything is client-side processed using JavaScript and static JSON-based dataset. Future development can include a backend with real-time updated data and more sophisticated filtering using AI/ML modules.

* 1. UI Mockup

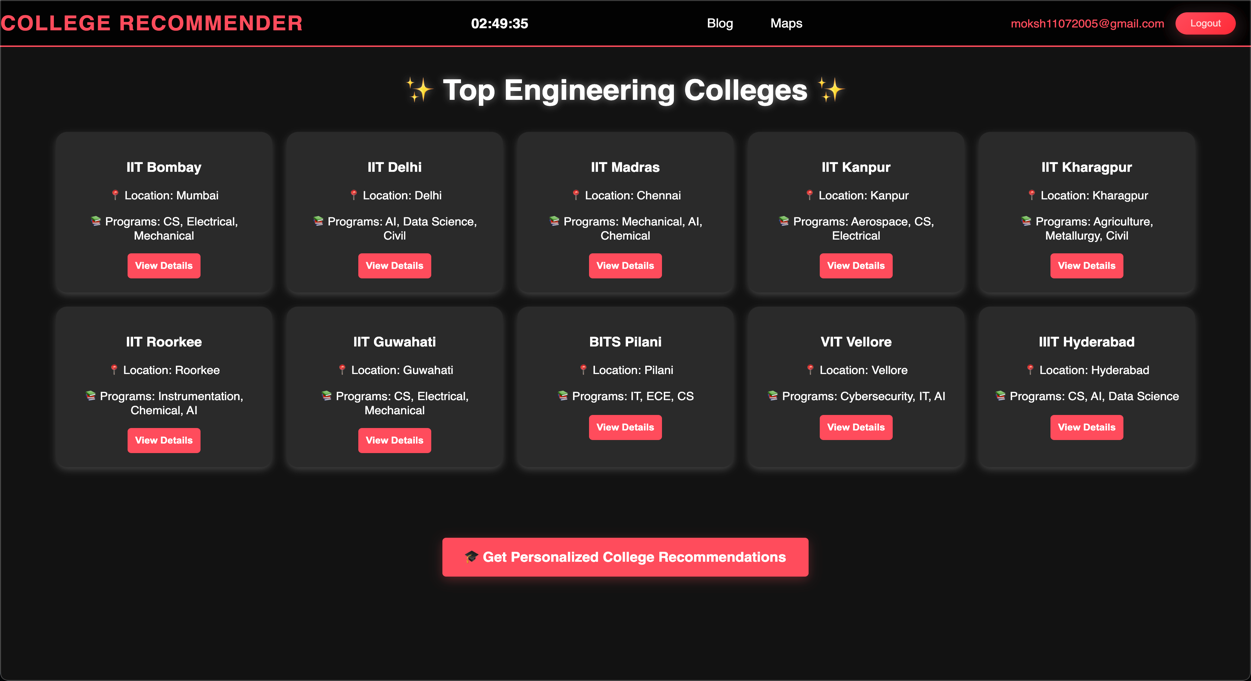


Figure : UI Mockup

1. Pseudo Code OF CORE FUNCTIONALITY

function recommendColleges(stream, location, maxFee, minRank):

recommendedList = []

for each college in collegeDataset:

if college.stream == stream or stream == "Any":

if college.location == location or location == "Any":

if college.fee <= maxFee:

if college.rank <= minRank:

recommendedList.append(college)

if recommendedList is empty:

return "No matching colleges found."

else:

sort recommendedList by college.rank ascending

return recommendedList

1. Project Closure

This section elucidates the overall lookup at the project and some of the future works that may enhance the solution.

* 1. Goals / Vision

Our initial intention for this project was to create a recommendation web application that would assist students in making rational college choices by streamlining the selection process based on more than one preference. The central idea was to establish a simple and effective system through Design Thinking principles such that the students could enter their priorities and obtain corresponding college recommendations.

As the project went along, the scope changed to concentrate more on usability, accessibility, and interactivity. Instead of incorporating advanced machine learning algorithms during this phase, our focus changed to creating a solid and working prototype that showcased the end-to-end user flow and logic of a smart college recommender. We made sure to prioritize developing clear flowcharts, clean UI designs, and modular code to make the system intuitive and easily extensible.

This revised vision enabled us to provide a fully functional prototype supported by considered design, concise logic, and potential for future growth into real-time data integration and AI-based recommendation models.

* 1. Delivered Solution

Our intended deliverable was a web-based College Selection Assistant application that would gather user preferences and provide customized college suggestions through a simple-to-use interface. The system was to be constructed on fundamental web technologies, with a focus on transparent user interaction, accessible design, and logic transparency.

The solution delivered is a complete working web prototype built with HTML, CSS, and JavaScript, including all essential features conceived during the planning stage. The application allows the user to choose preferences such as location, stream, fees, and ranking and filters colleges based on these using a formal rule-based system.

While the existing version relies on a static dataset, the application itself is modular and documented, so it can be scaled easily in later versions. Dynamic database support, API-based college directory, and AI-powered personalization can be some of the improvements.

* 1. Remaining Work

To further develop the existing prototype, future efforts should involve adding a real-time college database and applying machine learning for more intelligent recommendations. Further features such as user login, bookmarking, and feedback collection can enhance user experience. A mobile-optimized design and admin panel for data management are also suggested for increased adoption and scalability.

REFERENCES

1. Shiksha.com

2. NIRF India Rankings

3. CollegeDunia.com

4. https://mermaid.js.org

5. https://developer.mozilla.org/en-US/docs/Web

6. https://www.w3schools.com