ALUMNI MENTORSHIP SYSTEM

A Project Report Submitted by

Team: The Base Case

Members: Rajyalakshmi A,Rithika P N , Vedhavarshini E S

Department of Computer Science
NIT TRICHY

Academic Year: 2025

GitHub Repository:
`https://github.com/eduri06-sketch/THEBASECASE`

Abstract

The Alumni Mentorship System is a comprehensive full-stack project designed to connect students and alumni for mentorship, guidance, and networking. The project includes both frontend and backend components. The frontend, built using HTML and CSS, provides an intuitive and user-friendly interface. The backend, developed in C++, manages data processing, user information, mentorship requests, and feedback storage using structured text files. A dedicated `/data` directory ensures organized file-based storage for persistent data handling. Together, these components create a complete system that promotes academic and professional collaboration between students and alumni.

1 Introduction

Mentorship is an essential process that helps bridge the gap between academic learning and real-world professional experience. Many students face uncertainty when choosing career paths or preparing for placements, and guidance from alumni provides clarity and confidence. The **Alumni Mentorship System**, created by Team *The Base Case*, was developed to build this connection digitally. It enables students to log in, explore alumni profiles, send mentorship requests, and share feedback. By combining frontend design and backend logic, the system achieves a balance between user accessibility and functional reliability, making it both interactive and practical.

2 Objective

The objectives of this project are:

- To design and develop a complete web-based system connecting alumni and students.
- To implement both frontend and backend modules that communicate effectively.
- To manage and store user, request, and feedback data efficiently.
- To enhance teamwork, coding discipline, and problem-solving skills.
- To demonstrate modular programming and file handling techniques in C++.

3 System Overview

The Alumni Mentorship System consists of two major components — the **Frontend Interface** and the **Backend Processing Unit**.

Frontend

The frontend was developed using HTML and CSS. It provides a user-friendly interface for both students and alumni. Major pages include:

- **index.html** – Landing page introducing the portal.
- **login.html** – Handles user authentication.
- **profile.html** – Displays user details.
- **requests.html** – Allows students to send or track mentorship requests.
- **search.html** – Enables alumni search by name or domain.
- **style.css** – Ensures visual consistency and layout design.

Backend

The backend is implemented in C++ and is responsible for data management, request handling, and feedback operations. It adopts an object-oriented structure with modular files:

- **main.cpp** – Controls program execution and coordinates backend logic.
- **functions.cpp / functions.h** – Contain definitions and declarations of major backend operations like adding users, viewing requests, and saving data.
- **alumni.h, student.h, user.h** – Represent object classes for each user category with attributes like name, ID, and contact details.

Data Directory

A dedicated `/data` subfolder was created within the backend to store information persistently. The backend interacts with this folder to read, write, and update records. The files include:

- **students.txt** – Stores registered student details such as ID, name, and department.

- **alumni.txt** – Contains alumni mentor details including designation, graduation year, and expertise.
- **requests.txt** – Records mentorship requests raised by students and their current status.
- **feedback.txt** – Stores comments or feedback submitted by users after mentorship sessions.

This structure ensures modularity, clarity, and easy debugging. The file-based storage model provides a lightweight yet effective way to simulate database functionality, making it ideal for prototype-level systems.

4 Technologies Used

- **Frontend:** HTML5, CSS3
- **Backend:** C++ (Object-Oriented Programming, File Handling)
- **Data Storage:** Text-based files in the `/data` folder
- **Version Control:** Git and GitHub
- **Development Tools:** Visual Studio Code, Code::Blocks, Browser Developer Tools

5 Results and Discussion

The final version of the Alumni Mentorship System successfully integrates both frontend and backend modules. Students can access the interface to view mentors, send mentorship requests, and submit feedback. The backend reads and updates the relevant text files within the `/data` folder dynamically, ensuring that all user interactions are persistently stored. This modular and scalable design allows future teams to easily transition from file-based storage to a database-driven system. The structured separation between UI and logic makes maintenance and upgrades straightforward.

6 Teamwork and Contribution

The project's completion was the result of effective coordination, communication, and shared responsibility among the team members:

- **Rajyalakshmi A** – Designed and developed the complete frontend using HTML and CSS.
- **Rithika P N** – Created the flowchart representation of the system, prepared the LaTeX project report, and contributed to backend development.
- **Vedhavarshini** – Assisted in creating the flowchart and compiled the project manual, while also contributing to backend functionality.
- **All Members** – Collaborated in backend programming, integrating the `/data` module, debugging, and testing.

The collaboration process improved technical proficiency, problem-solving, and documentation clarity, while strengthening teamwork and time management skills.

7 Conclusion and Future Scope

The Alumni Mentorship System developed by Team *The Base Case* demonstrates a complete working model for student-alumni interaction. The backend's modular structure and the organized `/data` directory showcase effective file handling and data management in C++. Future improvements may include:

- Transitioning from file-based storage to MySQL or Firebase databases.
- Implementing secure authentication with encrypted credentials.
- Adding automated email notifications and chat integration.
- Developing a mobile or cloud-based version of the portal.

This project highlights how collaboration and technology can build strong bridges between academic and professional communities.

References

- Project Repository: `https://github.com/eduri06-sketch/THEBASECASE`
- HTML and CSS Documentation – `https://developer.mozilla.org`