CSLR31-DSA

Mentorship platform

Manual

The Base Case

USER MANUAL – STUDENT ALUMNI MENTOR BACKEND SYSTEM

Overview:

The Student Alumni Mentor Backend System is a C++ program designed to manage data for a mentorship platform.

It handles the registration, login, mentorship request, and feedback functionalities for students and alumni.

This backend uses **file handling** to store all information in text files instead of using any database or user interface.

Project Structure:

The project consists of several C++ source and header files, along with a folder to store data files.

- **main.cpp** – Entry point of the program that runs the main menu and core logic.
- **functions.cpp / functions.h** – Contain all backend functions such as registration, login, and data storage.
- **user.h, student.h, alumni.h** – Contain class structures and definitions for user types.
- **data folder** – Stores all persistent data as text files:
    - *students.txt* – Contains student registration data.
    - *alumni.txt* – Contains alumni registration data.
    - *requests.txt* – Stores mentorship requests made by students.
    - *feedback.txt* – Stores feedback details from mentoring sessions.

Purpose of the System:
This backend acts as the data management layer for a Student-Alumni Mentoring Platform.
It supports the following backend operations:

- Registering new students or alumni
- Logging in existing users

- Recording and storing mentorship requests
- Saving feedback from students or alumni after sessions
- Retrieving and displaying stored information when needed

How to Use:

1. Compile the C++ files using a compiler such as g++.
2. Run the compiled program (mentor or mentor.exe).
3. Use the console menu to perform operations such as registration or login.
4. The program automatically creates and updates the necessary text files under the “data” folder.
5. Each action — such as registering, sending requests, or giving feedback — is reflected in these files.
6. You can open the text files to view the stored information.

Data Handling Details:

- Every student or alumni registration creates a new entry in their respective text file.
- Mentorship requests are stored line by line in *requests.txt* with details of both student and alumni.
- Feedback is stored in *feedback.txt* for each interaction.
- All operations rely on file I/O (fstream) for reading and writing.

Key Features:

- Works entirely through command-line (no GUI).
- Uses text files for persistent storage.
- Cleanly separates data for students, alumni, requests, and feedback.
- Can be extended or integrated later into a frontend interface or web platform.

Notes:

- Ensure the “data” folder exists in the project directory before running the program.
- No external libraries or databases are required.

- To stop execution, use the Exit option from the main menu or close the terminal.

End of User Manual