North South University

Department of Electrical and Computer Engineering

Project Report: Case & Lawyer Scheduling System

Name: Md. Atiq Shahriar

ID: 2212050042

Course: CSE115L

Section: 12

Submitted To: Shahriyar Zaman Ridoy

Submission Date: 12.12.2025

1. Introduction

The **Case & Lawyer Scheduling System** is a console-based software application designed to help law firms or court clerks manage legal case schedules efficiently. Traditionally, case details are maintained in physical registers, which can be difficult to search and maintain. This project aims to digitize this process using the C programming language, allowing users to store, retrieve, search, and delete case records using a simple file-based system.

2. Objectives

The primary objectives of this project are:

- To create a system that stores Case IDs, Lawyer names, Client names, and Hearing dates.
- To utilize **File Handling** in C for permanent data storage.
- To implement basic database operations: **Create (Add), Read (View), Search, and Delete**.
- To demonstrate the use of fundamental C concepts like Loops, Arrays, and Functions without using complex structures.

3. System Features

The system provides the following functionalities:

1. **Add New Case:** Allows the user to input details (ID, Lawyer, Client, Date) and saves them to a text file.
2. **View List:** Displays all scheduled cases in a tabular format.
3. **Search Case:** Enables the user to find specific case details using the Unique Case ID.
4. **Delete Case:** Removes a resolved or cancelled case from the records permanently.
5. **Data Persistence:** Data remains saved in `schedule.txt` even after the program is closed.

4. Technical Implementation

4.1 Tools & Language

- **Language:** C Programming Language
- **IDE/Compiler:** Code::Blocks
- **OS:** Windows

4.2 Key Concepts Used

- **File Handling:** Used `fopen()`, `fprintf()`, `fscanf()`, and `fclose()` to read and write data to `schedule.txt`.
- **Functions:** The code is modular, with separate functions for each feature (`addCase`, `viewAllCases`, etc.).

- **Control Statements:** if-else is used for decision making and while(1) loop is used to create the main menu.
- **Arrays:** Character arrays (char name[50]) are used to store string data.

4.3 Algorithm for "Delete" Logic

Since a specific line cannot be directly deleted from a text file in C, the following logic was used:

1. Open the original file (schedule.txt) in Read mode.
2. Open a temporary file (temp.txt) in Write mode.
3. Read every line from the original file.
4. If the Case ID matches the ID to be deleted, **skip** writing it to the temporary file.
5. If the Case ID does not match, **write** it to the temporary file.
6. Close both files.
7. Delete the original file and rename temp.txt to schedule.txt.

5. Function Descriptions

Function Name	Description
addCase()	Takes user input and appends it to the file.
viewAllCases()	Reads the file from start to end and prints data on the screen.
searchCase()	Scans the file for a matching Case ID and prints the specific record.
deleteCase()	Copies valid data to a new file and removes the selected record.
main()	Displays the menu and handles user choices using an infinite loop.

6. Limitations

- **Input Format:** Since `scanf("%s")` is used, names cannot contain spaces (e.g., "Mr.Rahim" is valid, "Mr. Rahim" is not).
- **No Structures:** To keep the code simple, `struct` was not used; instead, local variables were used within functions.
- **Console Interface:** The system uses a Command Line Interface (CLI), which is less user-friendly than a Graphical User Interface (GUI).

7. Future Scope

In the future, this project can be improved by:

- Implementing **Structures (`struct`)** for better data organization.
- Using `gets()` or `fgets()` to allow names with spaces.
- Adding a **Login System** for security.
- Sorting cases by Hearing Date.

8. Conclusion

The **Case & Lawyer Scheduling System** successfully demonstrates how to build a functional management system using basic C programming. It solves the problem of manual data entry and provides a simple, effective way to manage legal schedules. This project strengthens the understanding of file manipulation and modular programming in C.