- **Project Title:** ScholarSync a School Management System
- **Submitted By:** Garima Kapoor
- **GitHub Link:** https://github.com/garimakapoor1204/Garima_main_project_C.git

Body of Report:

Project Report: Student Record Management System

1. Abstract

The **Student Record Management System** is a console-based application developed in the C programming language to satisfy the requirements of the CSEG1032 Major Project. It automates the administrative and academic processes of an educational institution, addressing the inefficiencies of manual record-keeping. The system features a standardized directory structure and implements role-based access control for Admins, Teachers, and Students.

2. Problem Definition

2.1 Background

Managing student data manually leads to redundancy and errors. This project aims to digitize these records, ensuring data integrity and ease of access.

2.2 Objectives

- To design a modular C program with separate compilation units.
- To implement file handling for persistent storage of student and teacher records.
- To strictly adhere to the mandatory GitHub repository structure.

3. System Design

The system uses struct definitions to model real-world entities:

- **Student:** Stores entry number, personal details, and a list of enrolled subjects.
- **Teacher:** Stores ID, subject specialization, and assigned classes.
- **Class:** Links teachers to students.

Flow: The program begins by loading binary data files (.dat) into memory. Users are presented with a main menu to select their role. Authentication is verified against the loaded structures before access is granted.

4. Implementation Details

The repository is organized as follows:

- /src: Contains all .c source files.
- /include: Contains all .h header files.
- /docs: Contains this project report and documentation.
- README.md: Project description and usage instructions.

Key Logic:

- **Authentication:** A loop checks the entered credentials against the stored struct arrays.
- **GPA Calculation:** The system iterates through a student's subject list to calculate the average grade dynamically.
- **File Persistence:** fwrite() and fread() are used to dump memory structures to binary files on exit and load them on startup.

5. Testing & Results

The system was tested against the provided sample_input.txt criteria.

- **Test Case 1 (Admin):** Successfully added a teacher and assigned them to Class 10.
- **Test Case 2 (Student):** Successfully calculated GPA based on updated grades.
- **Test Case 3 (Error Handling):** Invalid IDs entered during login were caught gracefully without segmentation faults.

6. Conclusion

The project successfully implements a working Student Record Management System in C. It meets all evaluation criteria, including modular code separation, correct folder structure, and original logic implementation.

7. Appendix

Project Repository: https://github.com/garimakapoor1204/Garima_main_project_C.git