**Student Grade Management Program**

**Introduction**

**This program is designed to manage student details, calculate CGPA (Cumulative Grade Point Average), and determine departmental ranks based on the entered information. It allows users to enter, store, retrieve, and analyse student data efficiently.**

**Features**

**Student Details: Capture information such as roll number, name, department, study year, completed semesters, subjects, credits, and grades.**

**CGPA Calculation: Automatically calculates the CGPA based on the grades and credits of the subjects entered.**

**Departmental Ranks: Computes ranks within departments based on CGPA.**

**Storage: Saves student details to a CSV file for future reference.**

**User-Friendly Interface: Provides clear instructions and options for seamless interaction.**

**How to Use**

**Welcome Screen: Upon execution, the program displays a welcome message introducing its functionalities and guidelines for entering department codes and student names.**

**Menu Options:**

**Enter Student Details: Allows users to input information for a new student.**

**Display Student Details: Enables users to view details of a specific student by entering their roll number.**

**Display Departmental Ranks: Shows the ranks of students within their respective departments.**

**Exit: Saves student data and terminates the program.**

**Entering Student Details:**

**Users can input relevant details for each student, including roll number, name, department, study year, completed semesters, and subjects with their corresponding credits and grades.**

**CGPA Calculation:**

**The program automatically calculates the CGPA based on the entered grades and credits for each subject.**

**Departmental Ranks:**

**After entering student details, the program calculates ranks within each department based on CGPA.**

**Saving Data:**

**Student details are saved to a CSV file named "Student.csv" for future reference.**

**Exiting the Program:**

**Upon choosing the exit option, the program saves all student data and terminates gracefully.**

**Requirements**

**This program is written in C and requires a C compiler to build and execute.**

**Standard libraries such as stdio.h, stdlib.h, and string.h are utilized.**

**Ensure the provided constants, such as MAX\_stud and MAX\_sub, are appropriately configured based on system requirements and constraints.**

**Make sure the CSV file "Student.csv" is accessible and writable by the program.**

**Acknowledgments**

**This program was developed to streamline student data management and facilitate efficient analysis of academic performance. It serves as a versatile tool for educational institutions and administrators seeking to organize and evaluate student records effectively.**

**Contributions and Feedback**

**Contributions to this project are welcome via pull requests. If you encounter any issues or have suggestions for improvement, please feel free to open an issue on GitHub.**