**Student-Record-Management-System**

This project belongs to

1)Sudarshan Dhakal

2)Rohit Khanal

3)Tilak Rokaya

4)Udaya Bhatta

# # Introduction

The Student Management System is a C++ program designed to manage student records efficiently. It allows users to create, read, update, and delete (CRUD) student information, including marks in various subjects. The program stores student data in a binary file and provides a user-friendly menu to interact with the system.

# # Methodologies

*Class Definition:*

student:This class stores the student details such as roll number, name, marks in various subjects, average marks, and grade. It also contains methods to input data, calculate average marks and grade, and display student details.

*File Handling:*

The program uses binary files to store student records. It employs file streams (fstream, ifstream, ofstream) to perform read and write operations.

*Menu-Driven Interface:*

A loop provides a menu-driven interface that allows users to choose different operations such as adding a new student, displaying a student's details, displaying all students' details, deleting a student's record, and modifying a student's details.

# #Code Explanation

*Class Definition and Methods:*

The student class encapsulates the student data and provides methods for data manipulation.

class student {

int rollno;

char name[50];

int eng\_marks, math\_marks, sci\_marks, lang2\_marks, cs\_marks;

double average;

char grade;

public:

void getdata();

void showdata() const;

void calculate();

int retrollno() const;

};

*Data Input and Calculation:*

The getdata method collects the student's information, and the calculate method computes the average marks and assigns a grade.

void student::getdata() {

// Collect student information

}

void student::calculate() {

// Calculate average marks and grade

}

*Displaying Data:*

The showdata method displays the student's details.

void student::showdata() const {

// Display student details

}

*File Operations:*

The program includes functions for creating, displaying, updating, and deleting student records.

void create\_student();

void display\_sp(int);

void display\_all();

void delete\_student(int);

void change\_student(int);

*Menu-Driven Interface:*

The main function presents a menu to the user and performs the chosen operation.

int main() {

char ch;

do {

// Display menu and handle user choice

} while(ch != '6');

return 0;

}

# # Discussion

The Student Management System demonstrates fundamental concepts of C++ such as classes, file handling, and user input/output. It provides a practical application of these concepts by managing student records. The use of a binary file ensures efficient storage and retrieval of data.

# # Conclusion

This Student Management System in C++ effectively manages student records by providing essential CRUD operations. It showcases the power of C++ in handling data through classes and file operations. The program can be further enhanced by adding features such as data validation, more comprehensive error handling, and a graphical user interface.