***ABSTRACT***

The present project is aimed to be creating a C++ program that can store, add, display, delete, and modify a student's marks in their subjects. The program is completely menu driven without any graphical interface. The data is stored in a binary file for future reference.

Student Report Card Generating System is a mini project is being developed record student personal details and generate report for school student. The report card generated using this software can be retained permanently on the computer which can be referred permanently on the computer which can be referred. This project is doneusing turbo C++ compiler. This system will present a comprehensive picture of the student in overall development. It also provides glance on the strengths and weakness of the student across various academic subjects in which the student and thus enables the teacher to focus on the subject in which student has to improve his knowledge.

## GRAPHICAL ABSTRACT



Fig. 1.1 Graphical Abstract

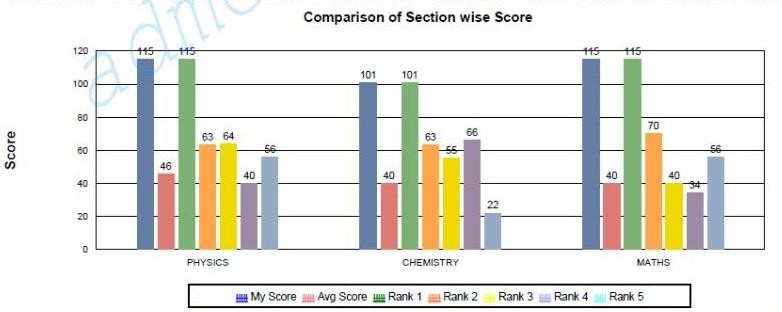


Fig 1.2 Graphical Abstract

## ABBREVIATIONS

*Abbreviations Definitions*

1. AS Alternative Schooling
2. Avg Average
3. BAS Baseline Assessment Studies
4. BRC Block Resource Centre
5. CR Completion Rate
6. CRC Cluster Resource Centre
7. CWSN Child With Special Needs
8. DIET District Institute of Education and Training
9. DISE District Information System for Education
10. DOR Dropout Rate
11. DPEP District Primary Education Programme
12. DRC District Report Cards
13. Ed. CIL Educational Consultants India Limited
14. EGS Education Guarantee Scheme
15. EMIS Educational Management Information System
16. GER Gross Enrolment Ratio
17. GOI Government of India
18. Govt. Government
19. GPI Gender Parity Index
20. M. Phil. Master of Philosophy
21. MHRD Ministry of Human Resource Development
22. NCERT National Council of Educational Research and Training
23. NER Net Enrolment Ratio
24. NIEPA National Institute of Educational Planning and Administration
25. No Res No Response
26. No. Numbe

## SYMBOLS

*Symbols Definitions*

1. % Percent
2. \* Asterisk
3. → Right Arrow
4. × Multiplication Symbol

# CHAPTER 1 INTRODUCTION

A report card system is software designed to create and manage student report cards. A report card system helps to execute all the activities related to report cards in the system. By using a good report card system, educators can design, create, distribute, and maintain student report cards easily.

A report card system should have features such as multiple report templates, customization, data entry, automatic calculations, automatic grading, graphs, charts, comment generator, digital signatures etc.

The Student Report Card Management System in **C++** is a desktop application coded in a **C++** programming language. The project contains basic functionality that allow user to generate the student report card. This project was created to fasten the distribution of grade record to each student. This will make it easier for the teacher to compute the General Average for the student grade. This Student Report Card Management System is a simple project that benefits students taking computer related course. This is helpful for those who are looking grading system related project. This Student Report Card Management System in **C++** can provide new techniques in **C++** programming.

The Student Report Card is required in education system that uses some form of reporting to identify student's performance. All Report cards are list of marks scored that analyst the student capabilities to each subject such as: Math, English, Science, etc.

The project Student Report Card Management is aimed to efficiently store and retrieve student examination reports.

This project in C++ is a simple console application built without any graphical Interface. In this project, users can perform typical report card related functions like adding a new student record and displaying, modifying, editing and deleting it. File handling has been effectively used to perform all these. This project will teach you how to use file handling in C++, add, read, display, search, modify and delete record from file.

## PROJECT DESCRIPTION

* + 1. **Create student report card record**: This feature creates a new student record containing his marks. Void write student (); function writes a record in a binary file. For this the information to be provided are the name and roll no. of the student, and the marks obtained by him/her in 5 subjects – Mathematics 3, Data Structure and Algorithm, Software Engineering, Microprocessor and Principles of Programming Language.
    2. **Read all students report card record**: The void displays all (); function in this student report card management system project in C++ has been used for this feature. It shows the progress report of all the students added in file.

This feature displays the roll no. and name of all the students, the marks obtained by them in 5 subjects – Mathematics 3, Data Structure and Algorithm, Software Engineering, Microprocessor and Principles of Programming Language along with the percentage and grade of each student.

* + 1. **Read specific student’s report card record**: This feature is same as the one explained above, except it shows the progress report and relevant data related to a particular student. Void displays (in) are used for this purpose.
    2. **Display all students’ grade report**: This feature enlists all the students‘ record saved in file. The grade report is displayed in a tabular form with roll no. and name of the students, marks achieved in the five subjects, and the grade and percentage obtained by them. Void class result (); is thus used.
    3. **Modify student’s report card record**: void modify student (in); function is used to edit the report card record of a particular student. For this, the name and roll no. of the student is sought. Upon successful modification, the program displays the message ―Record Updated‖. If no record of student is found in file, it displays the message ―Record not found‖.
    4. **Delete student record:** void delete student (in); function deletes the report card record of a particular student; it first of all asks for the name and roll no. of the student whose record is to be deleted.

### Object oriented data centered approach

The component of a system encapsulates data and the operation that must be applied to manipulated data .Communication and coordination between components is accomplished via message passing.

## Benefits of using a Report Card Management System

* + 1. Saves time

A report card system will help teachers to create report cards without any of the time consuming manual tasks. Teachers and educators are able to create report cards quickly and in time. Features like automatic calculations and grading, preset report templates, teacher comment bank will save a huge amount of time while creating reports.

* + 1. Customizations

A report card system allows teachers to customize report card templates to their liking. Teachers can customize elements such as logos, address, names, attendance, widget styles, signatures, grades, comments, colors and themes.

* + 1. Improved efficiency

As a report card system allows teachers and educators to create report cards in a short time, without any manual data entry and manual calculations, their efficiency will increase multifold. The time they save in creating report cards can be used in other critical tasks.

* + 1. Automations

Teachers don't need to worry about manual data entry and complex manual calculations. A report card system eliminates the need for manual calculations thereby eliminating errors and miss-calculations.

* + 1. Graphical elements

Schools can use graphical elements such as charts, line graphs, graphs etc. in their student report cards. Graphs help to convey meaningful information in a better way which is otherwise not inferred just from the marks. Use of such elements ensures that parents can get useful insights from the report cards.

### OBJECTIVE (OF PROJECT)

A single report grade for each academic subject is the most common and accepted system in elementary, middle and secondary schools. They show what learning is being assessed and if the student is below grade level, at grade level or above grade level.

Report cards use a multi-dimensional system of reporting which includes attitude, effort, school behavior and academics. It communicates to parents how a child is doing academically, socially and their life skills.

Schools use a conventional, subject-based report card or a report card that represents these intended learning outcomes as standards. A report card identifies the skills in which students are struggling, and provides a clear understanding of a student‘s strengths.

### ANALYSIS AND FEATURE

When finalizing the specifications/features for a student report card system, it is important to consider any constraints that may impact the implementation and maintenance of the system. Some common constraints include:

1. Budget: The cost of the system should be within the budget allocated by the school or district.
2. Time: The system should be implemented within a reasonable timeframe and not disrupt regular school operations.
3. Resources: The school or district should have the necessary resources, such as IT support and training, to implement and maintain the system.
4. Compatibility: The system should be compatible with existing hardware and software used by the school or district.
5. Legal and regulatory requirements: The system should comply with legal and regulatory requirements, such as data privacy laws and student record-keeping regulations.
6. User feedback: User feedback should be taken into account when finalizing the specifications/features to ensure that the system meets the needs of all stakeholders.

By considering these constraints, the school or district can ensure that the selected specifications/features are feasible and sustainable in the long term

## TASK MANAGEMENT

From an articulation standpoint, task management entails more than just organizing virtual andphysical collections and scheduling activities.

Recent research has begun to address the problem of generic task management in the context of email. This development is hardly surprising, given that many digital device users are overloadedby the number of chores done through email. According to this research, any successful productivity tool must be tightly connected with email functionalities. Recent researches looked at task management strategies more generally because email and related technologies are unlikely to be the whole picture.

### 1.3.1 Student Report Card System as a solution to task management

The concept of student report card system has existed for a very long time and it is one of the primary methods for management of tasks, use of a report card as a reminder system, to report card as a system for note management, etc. In the simplest and most primitive form, a report card list can be implemented on a penand paper as a checklist of items which can be crossed of or ticked against when completed.

The prevailing offline paper based system has several disadvantages such as excessive use of time for manual record keeping, wasting resources, inefficiency in data logging as inconsistent data can be recorded. Most of the academic institutions still use this traditional method for student management. The data and information of any student is not accessible globally as the system is not online due to which the concerning users have to physically meet for the exchange of information.

Students have to stand in long queue and wait for hours to perform basic tasks such as fee payments and submit railway concession forms. Students are also not able to view their own attendance record and college notices when needed. To overcome the limitations of an existing system, we have proposed a web-based student management system. It is an online automated approach, advanced for everyday record keeping in academic organizations.

### FEATURES OF A STUDENT REPORT CARD SYSTEM:

A student report card system should have the following features:

1. User-friendly interface: The system should have an easy-to-use interface that allows

teachers to input grades and comments easily.

1. Customization: The system should be customizable to meet the specific needs of the school, such as grading scales, attendance, and behavior.
2. Accessibility: The system should be accessible to parents and students, allowing them to view grades and comments online.
3. Security: The system should be secure to prevent unauthorized access to student information.

### GANTT C[HART](https://www.studocu.com/in?utm_campaign=shared-document&utm_source=studocu-document&utm_medium=social_sharing&utm_content=final-project-report-20bcs1330)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of the task** | **Start date** | **End date** | **Duration** |
| Project kick off | 28/2/2023 | 02/03/2023 | 1 |
| Initial design | 02/03/2023 | 09/03/2023 | 4 |
| Making of prototype | 09/03/2023 | 16/03/2023 | 7 |
| Research | 16/03/2023 | 21/03/2023 | 6 |
| Adding functionality to prototype | 21/03/2023 | 23/03/2023 | 6 |
| Initiating development | 23/03/2023 | 28/03/2023 | 5 |
| Finalizing design | 28/03/2023 | 11/04/2023 | 3 |
| Developing prototype | 11/04/2023 | 13/04/2023 | 3 |
| Testing functionality | 13/04/2023 | 18/04/2023 | 2 |
| Implementing the solution | 18/04/2023 | 20/04/2023 | 4 |
| Initiating integration of modules | 20/04/2023 | 25/04/2023 | 5 |
| Finalizing essentialfunctionalities | 25/04/2023 | 27/04/2023 | 7 |
| Planning add on functionalities | 27/04/2023 | 02/05/2023 | 5 |
| Implementing add on functionalities | 02/05/2023 | 04/05/2023 | 3 |
| Testing of the project | 04/05/2023 | 09/05/2023 | 5 |
| Implement improvements and debugging | 09/05/202323 | 11/05/2023 | 3 |

Table 1.1

# CHAPTER 2 L[ITERATURE SURVEY](https://www.studocu.com/in?utm_campaign=shared-document&utm_source=studocu-document&utm_medium=social_sharing&utm_content=final-project-report-20bcs1330)

A student report card system is an essential tool in the education sector that is used to track and record student's academic performance. The system is used to evaluate students' progress, identify areas of improvement, and provide feedback to parents and teachers. This literature survey aims to review the existing literature on student report card systems, their features, and their impact on student performance.

## Evolution of Student report card system

Today's report cards typically include information on attendance, behavior, academic progress, and teacher comments. They provide a snapshot of a student's overall performance, highlighting areas of strength and weakness and identifying specific skills that need improvement. However, the process of creating report cards has remained a time-consuming and labor-intensive task for educators, until the advent of report card generators.

are software tools that automate the process of creating report cards. These tools allow educators to input student data, select a report card template, and generate customized report cards for each student. The rise of report card generators has transformed the report card creation process, making it faster, easier, and more accurate.

[Report card generator](https://www.teachmint.com/features/student-report-card)

Report card generators come with a wide range of templates, allowing educators to select a format that best fits their needs. They can also be customized with school logos, images, and color schemes, giving them a professional and personalized look.

### Essential Functionalities

1. Student Information

The system should be able to store and retrieve student information such as name, address, contact details, and class.

1. Grade Management

The system should be able to manage student grades for different subjects and terms. It should calculate the average grade for each subject and term.

1. Attendance Management

The system should be able to track student attendance for each class and generate attendance reports

1. Report Card Generation

The system should be able to generate report cards for each student at the end of each term. The report card should include the student's grades, attendance record, and any additional comments from teachers.

1. Parent-Teacher Communication

The system should allow parents and teachers to communicate easily about a student's progress. This could include messaging features or parent-teacher conferences.

1. Security

The system should have appropriate security measures to protect student information and prevent unauthorized access.

1. User Management

The system should allow administrators to manage user accounts for teachers, students, and parents. This includes creating new accounts, resetting passwords, and deactivating accounts when necessary.

### Integration with other technologies

Technology-based formative assessments can offer real-time reporting of results, allowing stakeholders to understand students‘ strengths and weaknesses, while guiding them to make valid, actionable interpretations of the assessment data.

Technology integration is the approach that companies use to choose and refine the technologies employed in a new product, process, or service. Access to great research is still immensely important, but if a company selects technologies that don‘t work well together, it can end up with a product that is hard to manufacture, is late getting to market, and does not fulfill its envisioned purpose. An effective technology-integration process starts in the earliest phases of an R&D project and provides a road map for all design, engineering, and manufacturing activities. It defines the interaction between the world of research and the worlds of manufacturing and product application.

## Need of project

Student report cards are important because they provide a comprehensive overview of a student's academic performance, attendance, and behavior. The report card helps parents and teachers to track the progress of the student over a period of time and identify areas where the student needs improvement. It also provides an opportunity for parents and teachers to discuss the student's academic progress and work together to support the student's learning. Additionally, report cards are often used as a basis for academic recognition, such as awards

or scholarships. Overall, student report cards are an essential tool for monitoring and supporting student success in their academic journey.

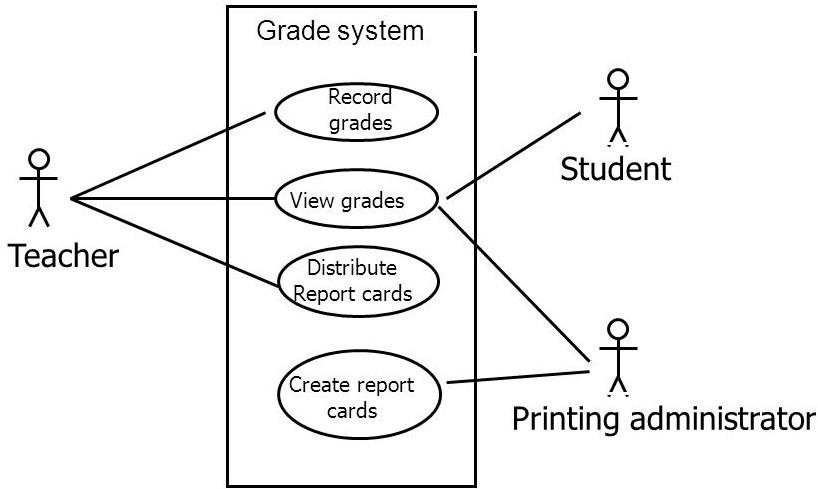
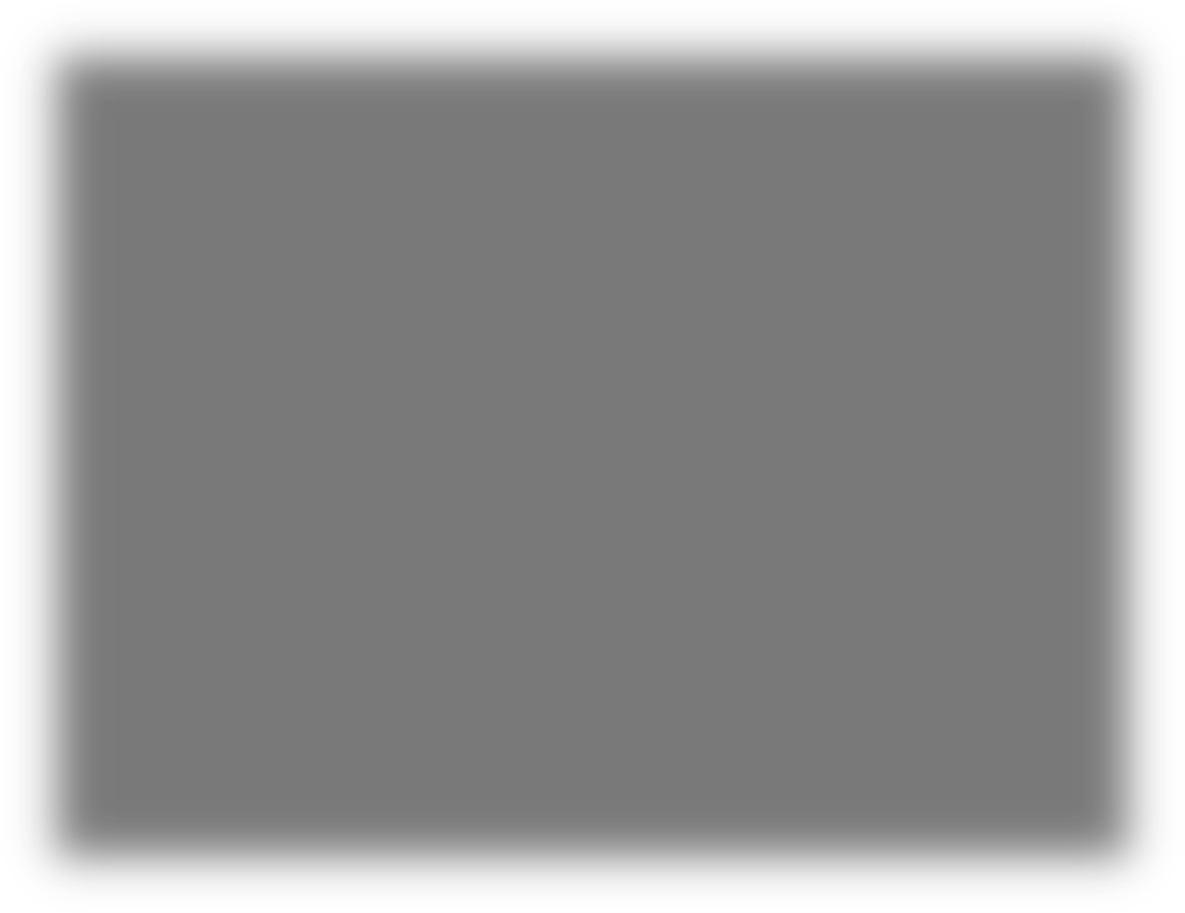


Fig 2.1 Use-case diagram

# CHAPTER 3 SYSTEM DESIGN

## PROBLEM DEFINITION

### INTRODUCTION

The project Student Report Card Management is aimed to efficiently store and retrieve student examination reports. This project in C++ is a simple console application built without any graphical Interface. In this project, users can perform typical report card related functions like adding a new student record and displaying, modifying, editing and deleting it.

File handling has been effectively used to perform all these. This project will teach you how to use file handling in C++, add, read, display, search, modify and delete record from file. The term report card, in its broadest sense, refers to a report at any level in the education system—from the student report cards familiar to parents in the United States to school level report cards to national reports such as the Partnership for Education Revitalization in the Americas‘ (PREAL‘s) education report cards in Latin America.

For the purposes of this brief, discussion will be confined to only reports providing data at the school level, including simple school profiles and the type of school report cards used in the United States under No Child Left behind (NCLB). In order to cast the net more widely, reports providing data at a broader geographic level, such as a district, and contain data for individual schools are considered, as well.

A number of countries are experimenting with school-level information systems known as

‗school report cards‘ to increase accountability and transparency. These systems have different formats and purposes, ranging from strict accountability systems that measure student performance to participatory diagnostic and management tools that support school managers. Efforts are relatively novel, and substantial evaluation information is not yet available. The purpose of this report is to present the various types of school report cards and information systems currently being used and establish a typology for understanding the range of audiences and purposes for such systems, as well as the continuum of cost and sophistication involved.

### OPEN SOURCE

Student report card system is open source and its code repository is publicly available on GitHub. It isopen to contributions and improvements by anyone who wishes to contribute to it. It is alsofree for modification and personal use.

A brief overview of open source projects:

Open source software is software with source code that anyone can inspect, modify, and enhance. "Source code" is the part of software that most computer users don't ever see;it's the code computer programmers can manipulate to change how a piece of software—a"program‖. Programmers who have access to a computer program's source code can improve thatprogram by adding features to it or fixing parts that don't always work correctly.

# Chapter 4

**SOFTWARE AND HARDWARE REQUIREMENTS**

## Software Requirements:

* + 1. Operating System: Microsoft Windows11.
    2. G++/Gcc complier.

## Hardware Requirements:

* + 1. Pointing Device: Touch Pad or Keys.
    2. Desktop or Laptop.

# CHAPTER 5 IMPLEMENTATION

## HEADER FILES.

**1. iosteam** :- iostream stands for **standard input-output stream**. #include iostream declares objects that control reading from and writing to the standard streams. In other words, the iostream library is an object-oriented library that provides input and output functionality using streams.

A stream is a sequence of bytes. You can think of it as an abstraction representing a device. You can perform I/O operations on the device via this abstraction. You must include iostream header file to input and output from a C++ program.

**Syntax**

#include iostream provides the most used standard input and output streams, cin and cout. The syntax for using them is as follows

**Standard Output Stream -- cout**

* It is an [instance](https://www.scaler.com/topics/basic-programs-in-cpp/) of the ostream class.
* It produces output on the standard output device, i.e., the display screen.
* We need to use the **stream insertion operator** << to insert data into the standard output stream cout, which has to be displayed on the screen.

Syntax:

cout << variable\_name;

### Standard Input Stream -- cin

* It is an instance of the istream class.
* It reads input from the standard input device, i.e., the keyboard.
* We need to use the **stream extraction operator** >> to extract data entered using the keyboard.

Syntax:

**2. String**: - A string in C++ is a type of object representing a collection (or sequence) of

cin >> variable\_name;

different characters. Strings in C++ are a part of the standard string class (std::string).

The string class stores the characters of a string as a **collection of bytes** in contiguous memory locations. Strings are most commonly used in the programs where we need to work with texts. We can perform various operations on the strings in C++. For example, reversing, concatenating, or passing as an argument to a function.

### Syntax

The syntax to create a string in C++ is quite simple. We use the *string* keyword to create a string in C++. However, we also need to include the standard string class in our program before using this keyword.

string str\_name = "This is a C++ string";

* + 1. **fstream** :- His class is the combination of both ofstream and ifstream. It provides the capability of creating, writing and reading a file. To access the following classes, you must include the stream as a header file like how we declare upstream in the header.

File Operations in C++

C++ provides us with four different operations for file handling. They are:

1. **Open ()** – This is used to create a file.
2. **Read () –** This is used to read the data from the file.
3. **Write () –** This is used to write new data to file.
4. **Close() –** This is used to close the file

## CLASSES USED IN PROJECT

class student{ }; //class ends here void student::calculate()

{

}

void student::getdata()

{

}

void student::show\_tabular() const

{

}

## SOURCE CODE

#include<iostream> #include<fstream> #include<iomanip> using namespace std; class student

{

int rollno;

char name[50];

int DSA\_marks, Micro\_marks, M3\_marks, SE\_marks, PPL\_marks; double per;

char grade;

void calculate(); public:

void getdata();

void showdata() const; void show\_tabular() const; int retrollno() const;

};

void student::calculate()

{

per=(DSA\_marks + Micro\_marks + M3\_marks + SE\_marks + PPL\_marks)/5.0; if(per>=75)

grade='A'; else if(per>=50)

grade='B'; else if(per>=33)

grade='C';

else

}

grade='F';

void student::getdata()

{

cout<<"\n Enter The roll number of student "; cin>>rollno;

cout<<"\n Enter The Name of student "; cin.ignore();

cin.getline(name,50);

cout<<"\n Enter the marks in DSA out of 100 : "; cin>>DSA\_marks;

cout<<"\n Enter the marks in Micro out of 100 : "; cin>>Micro\_marks;

cout<<"\n Enter the marks in M3 out of 100 : "; cin>>M3\_marks;

cout<<"\n Enter the marks in SE out of 100 : "; cin>>SE\_marks;

cout<<"\nEnter the marks in PPL out of 100 : "; cin>>PPL\_marks;

calculate();

}

void student::showdata() const

cout<<"\n Roll number of student : "<<rollno; cout<<"\n Name of student :- "<<name;

cout<<"\n Marks in Data Structure & Algorithm :-"<<DSA\_marks; cout<<"\n Marks in Microprocessor :- "<<Micro\_marks;

cout<<"\n Marks in Mathematics 3 :- "<<M3\_marks; cout<<"\n Marks in Software Engineering :- "<<SE\_marks;

cout<<"\n Marks in Principle of Programming Language :-"<<PPL\_marks; cout<<"\n Percentage of student is :-"<<per;

cout<<"\n Grade of student is :-"<<grade;

}

void student::show\_tabular() const

{

cout<<rollno<<setw(6)<<" "<<name<<setw(10)<<DSA\_marks<<setw(4)<<Micro\_marks<<setw(4)<<M3\_marks<<setw(4)

<<SE\_marks<<setw(4)<<PPL\_marks<<setw(8)<<per<<setw(6)<<grade<<endl;

}

int student::retrollno() const

{

return rollno;

}

void write\_student(); void display\_all(); void display\_sp(int);

void modify\_student(int); void delete\_student(int); void class\_result();

void result(); void intro();

void entry\_menu();

int main()

char ch; cout.setf(ios::fixed|ios::showpoint); cout<<setprecision(2);

intro(); do

{

system("cls");

cout<<"\n\n\n\tMAIN MENU";

cout<<"\n\n\t01. RESULT MENU"; cout<<"\n\n\t02. ENTRY/EDIT MENU"; cout<<"\n\n\t03. EXIT";

cout<<"\n\n\tPlease Select Your Option (1-3) ";

cin>>ch;

switch(ch)

{

case '1': result();

break;

case '2': entry\_menu(); break;

case '3':

break; default :cout<<"\a";

}

}while(ch!='3');

return 0;

}

void write\_student()

{

student st; ofstream outFile;

outFile.open("student.txt",ios::in); st.getdata();

outFile.write(reinterpret\_cast<char \*> (&st), sizeof(student)); outFile.close();

cout<<"\n\nStudent record Has Been Created "; cin.ignore();

cin.get();

}

void display\_all()

{

student st; ifstream inFile;

inFile.open("student.txt",ios::in); if(!inFile)

{

cout<<"File could not be open !! Press any Key..."; cin.ignore();

cin.get(); return;

}

cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n"; while(inFile.read(reinterpret\_cast<char \*> (&st), sizeof(student)))

{

st.showdata(); cout<<"\n\n====================================\n";

}

inFile.close();

cin.ignore();

cin.get();

}

void display\_sp(int n)

{

student st; ifstream inFile;

inFile.open("student.txt",ios::in); if(!inFile)

{

cout<<"File could not be open !! Press any Key..."; cin.ignore();

cin.get(); return;

}

bool flag=false;

while(inFile.read(reinterpret\_cast<char \*> (&st), sizeof(student)))

{

if(st.retrollno()==n)

{

st.showdata(); flag=true;

}

}

inFile.close(); if(flag==false)

cout<<"\n\nrecord not exist"; cin.ignore();

cin.get();

}

void modify\_student(int n)

{

bool found=false; student st; fstream File;

File.open("student.txt",ios::app|ios::in|ios::out); if(!File)

{

cout<<"File could not be open !! Press any Key..."; cin.ignore();

cin.get(); return;

}

while(!File.eof() && found==false)

{

File.read(reinterpret\_cast<char \*> (&st), sizeof(student));

if(st.retrollno()==n)

{

st.showdata();

cout<<"\n\nPlease Enter The New Details of student"<<endl; st.getdata();

int pos=(-1)\*static\_cast<int>(sizeof(st)); File.seekp(pos,ios::cur);

File.write(reinterpret\_cast<char \*> (&st), sizeof(student));

cout<<"\n\n\t Record Updated"; found=true;

}

}

File.close(); if(found==false)

cout<<"\n\n Record Not Found "; cin.ignore();

cin.get();

}

void delete\_student(int n)

{

student st; ifstream inFile;

inFile.open("student.txt",ios::app); if(!inFile)

{

cout<<"File could not be open !! Press any Key..."; cin.ignore();

cin.get(); return;

}

ofstream outFile;

outFile.open("Temp.txt",ios::out); inFile.seekg(0,ios::beg);

while(inFile.read(reinterpret\_cast<char \*> (&st), sizeof(student)))

{

if(st.retrollno()!=n)

{

outFile.write(reinterpret\_cast<char \*> (&st), sizeof(student));

}

}

outFile.close(); inFile.close(); remove("student.txt");

rename("Temp.txt","student.txt"); cout<<"\n\n\tRecord Deleted .."; cin.ignore();

cin.get();

}

void class\_result()

{

student st; ifstream inFile;

inFile.open("student.txt",ios::app); if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

cin.ignore();

}

cin.get(); return;

cout<<"\n\n\t\tALL STUDENTS RESULT \n\n"; cout<<"==========================================================\n";

cout<<"R.No Name DSA Micro M3 SE PPL %age Grade"<<endl; cout<<"==========================================================\n";

while(inFile.read(reinterpret\_cast<char \*> (&st), sizeof(student)))

{

st.show\_tabular();

}

cin.ignore();

cin.get(); inFile.close();

}

void result()

{

char ch; int rno;

system("cls"); cout<<"\n\n\n\tRESULT MENU"; cout<<"\n\n\n\t1. Class Result"; cout<<"\n\n\t2. Student Report Card"; cout<<"\n\n\t3. Back to Main Menu"; cout<<"\n\n\n\tEnter Choice (1/2/3)? "; cin>>ch;

system("cls"); switch(ch)

{

case '1' : class\_result(); break;

case '2' : cout<<"\n\n\tEnter Roll Number Of Student : "; cin>>rno;

display\_sp(rno); break;

case '3' : break;

default: cout<<"\a";

}

}

void intro()

{

cout<<"\n\n\n\t\t STUDENT";

cout<<"\n\n\t\tREPORT CARD"; cout<<"\n\n\t\t PROJECT";

cout<<"\n\n\n\tMADE BY : VAISHNAVI TALELE "; cout<<"\n\tCOLLEGE : JIT COLLEGE ";

cin.get();

}

void entry\_menu()

{

char ch;

int num; system("cls");

cout<<"\n\n\n\tENTRY MENU"; cout<<"\n\n\t1.CREATE STUDENT RECORD"; cout<<"\n\n\t2.DISPLAY ALL STUDENTS RECORDS"; cout<<"\n\n\t3.SEARCH STUDENT RECORD "; cout<<"\n\n\t4.MODIFY STUDENT RECORD";

cout<<"\n\n\t5.DELETE STUDENT RECORD";

cout<<"\n\n\t6.BACK TO MAIN MENU"; cout<<"\n\n\tPlease Enter Your Choice (1-6) "; cin>>ch;

system("cls"); switch(ch)

{

case '1': write\_student(); break; case '2': display\_all(); break;

case '3': cout<<"\n\n\tPlease Enter The roll number "; cin>>num; display\_sp(num); break;

case '4': cout<<"\n\n\tPlease Enter The roll number "; cin>>num; modify\_student(num);break;

case '5': cout<<"\n\n\tPlease Enter The roll number "; cin>>num; delete\_student(num);break;

case '6': break;

default: cout<<"\a"; entry\_menu();

}

}

# CHAPTER 6 RESULT

## 5.1 Output

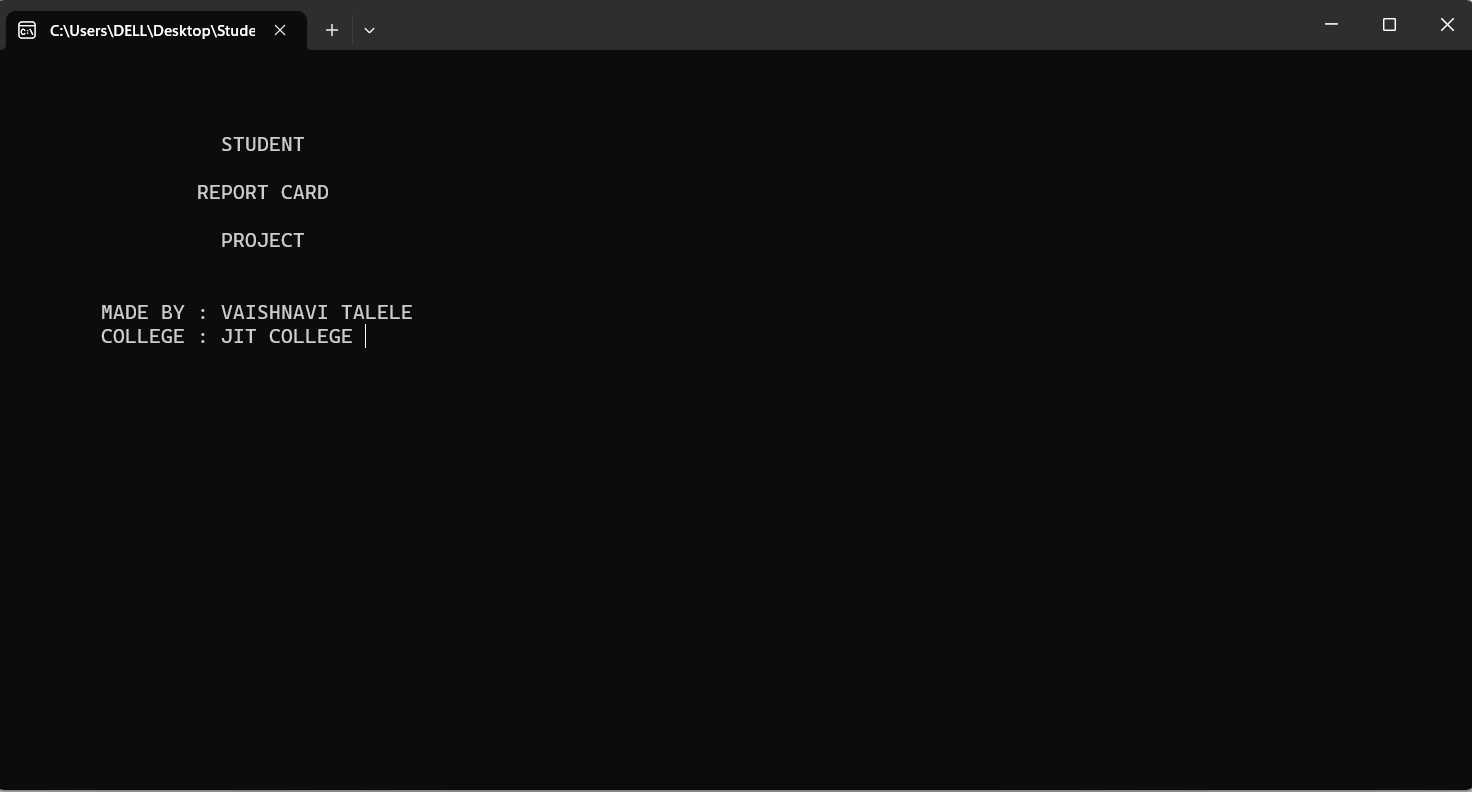


Fig 5.1.1 Main Screen

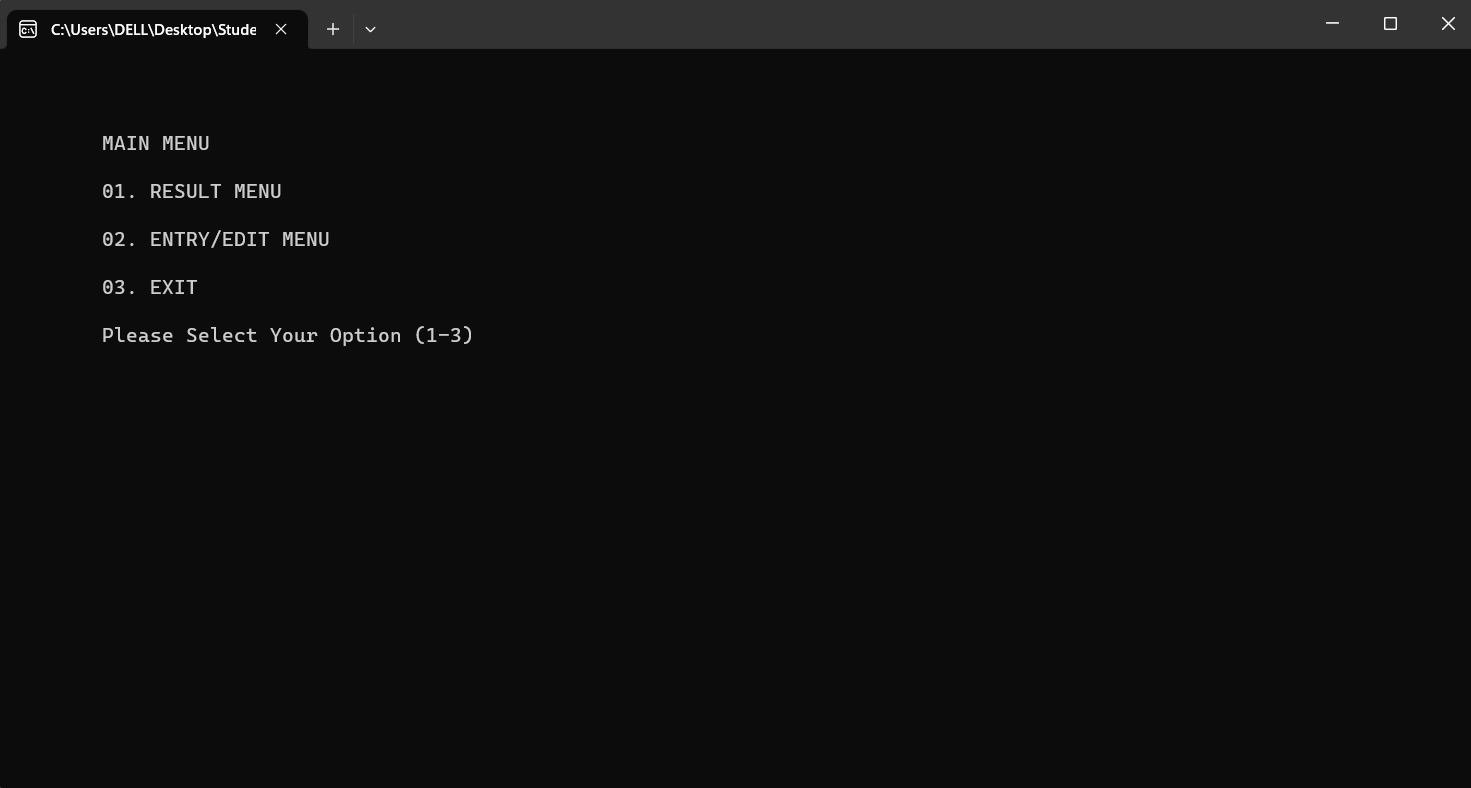


Fig 5.1.2 Main Menu

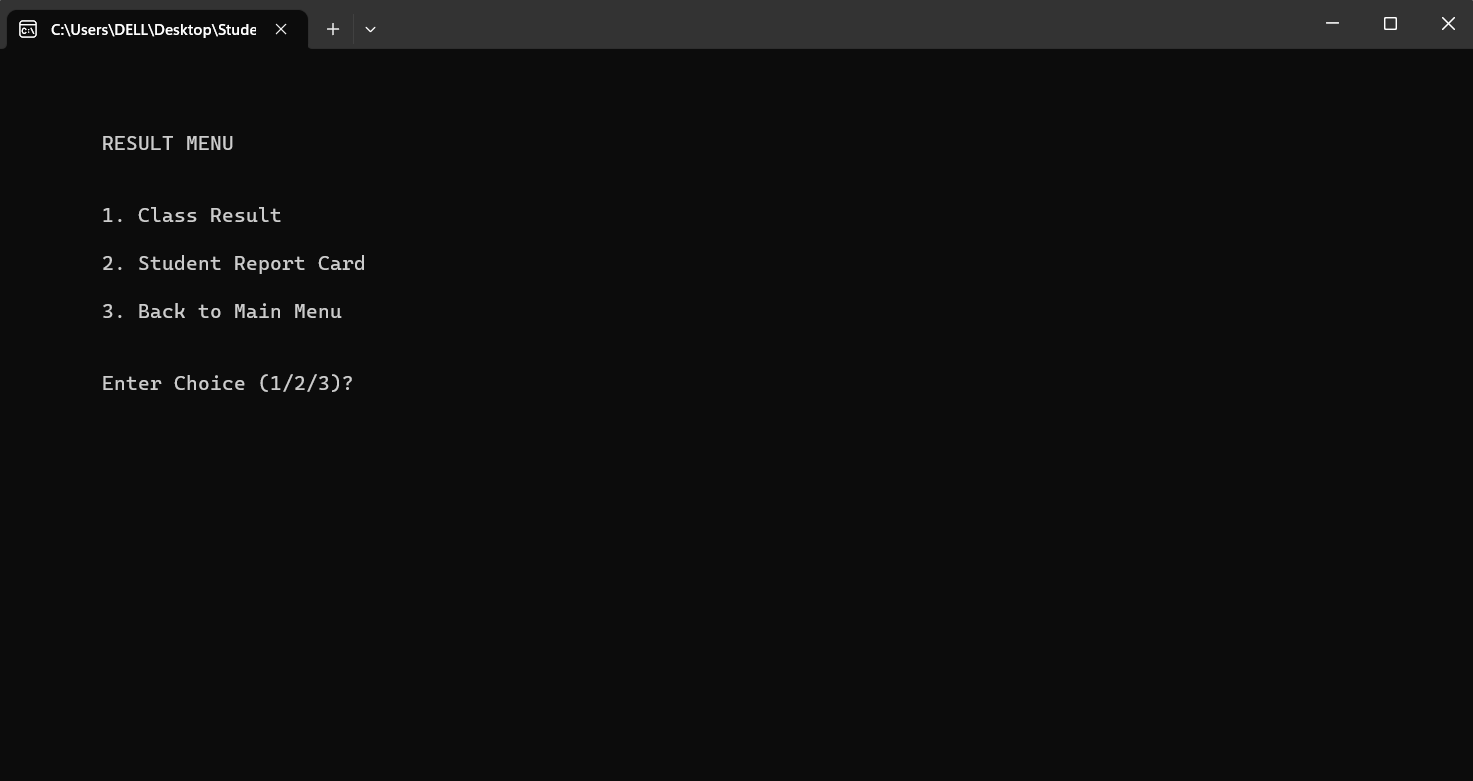


Fig 5.1.3 Result Menu

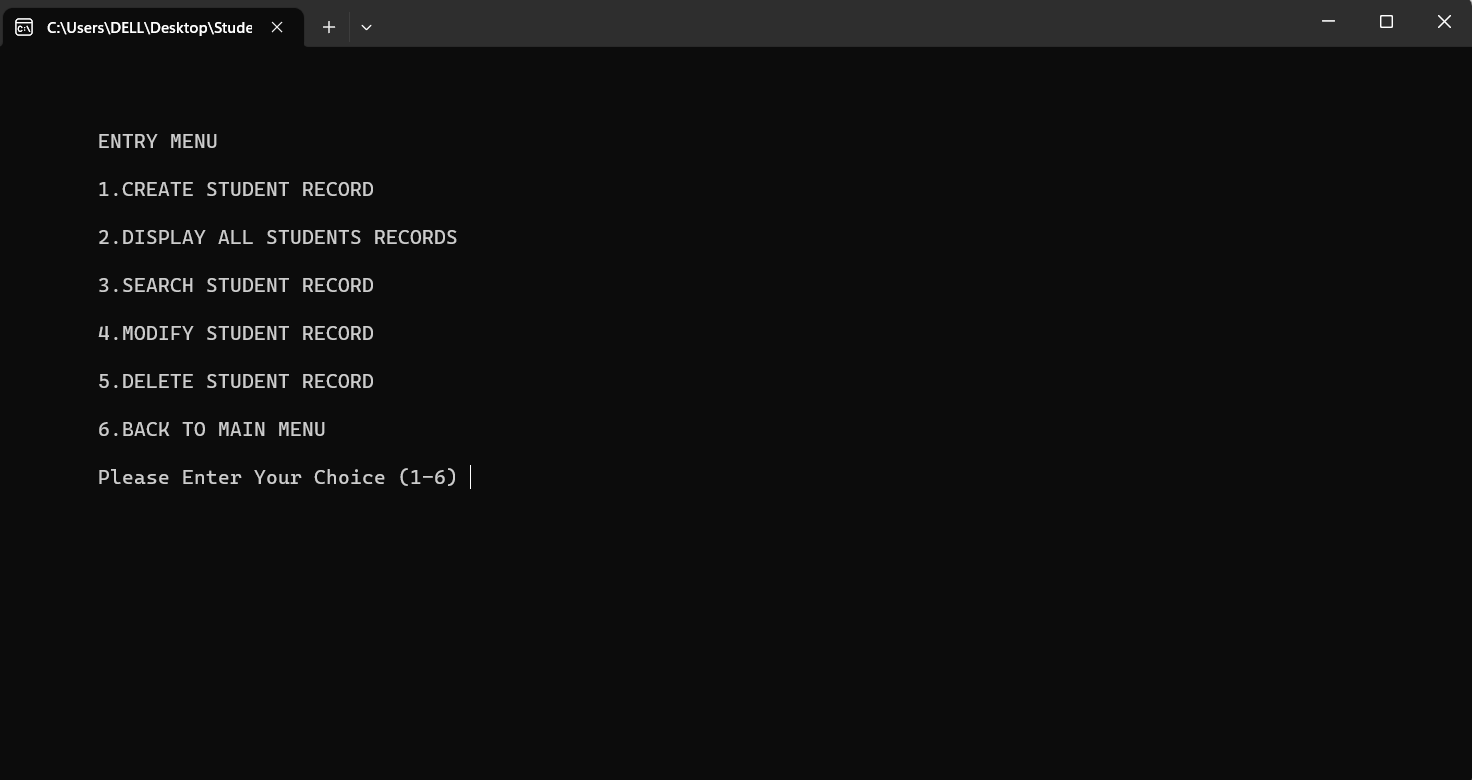


Fig 5.1.4 Entry Menu

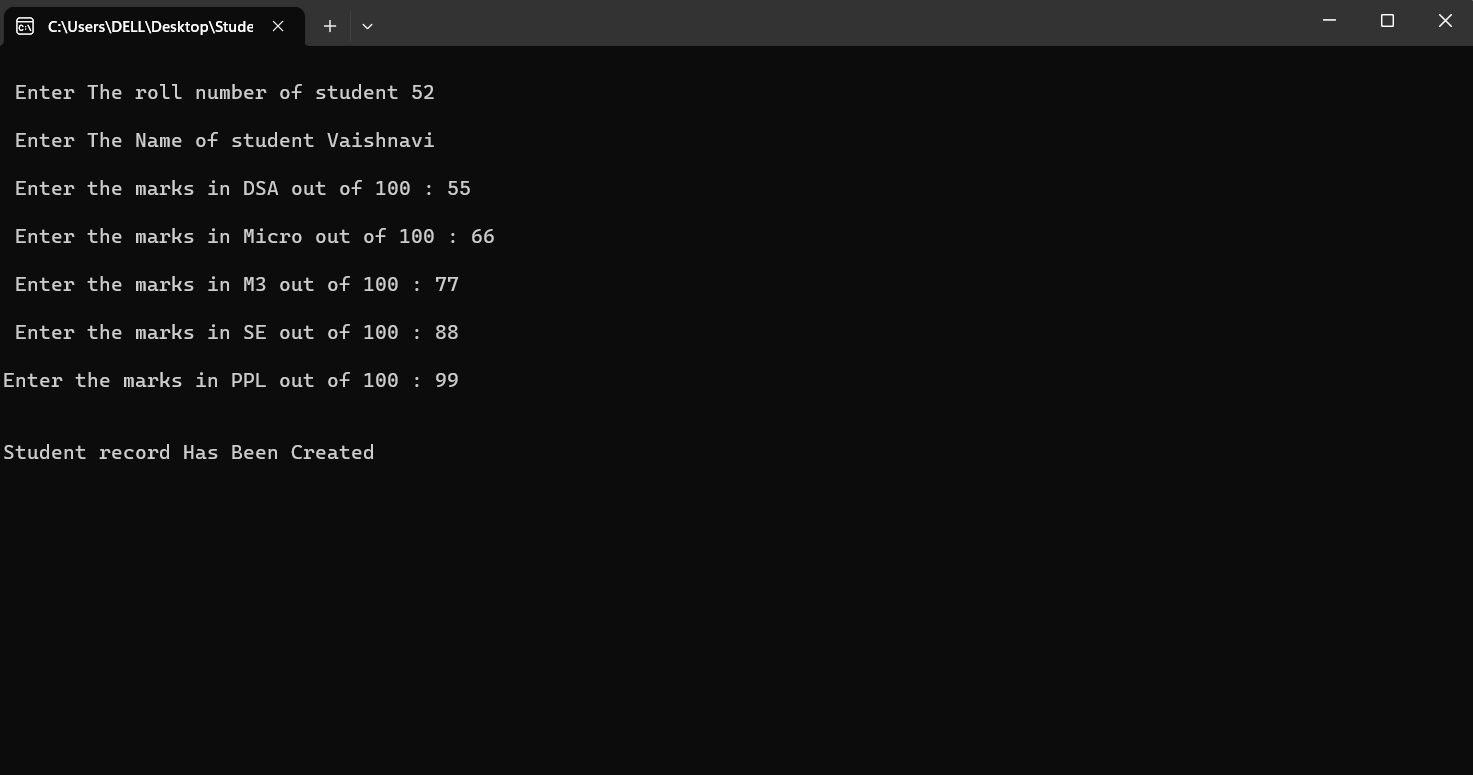


Fig 5.1.5 Create a new student data

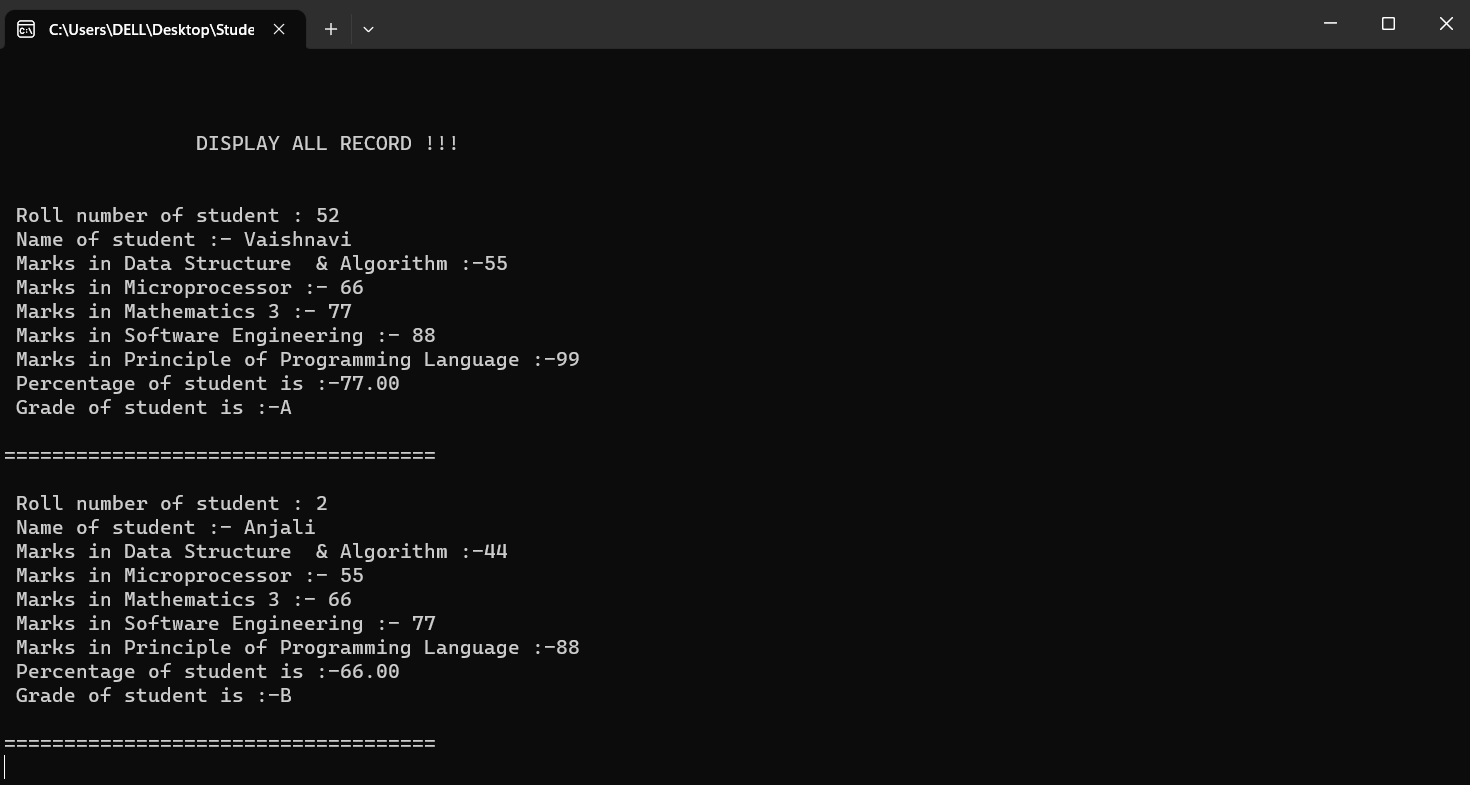


Fig 5.1.6.Display Student record

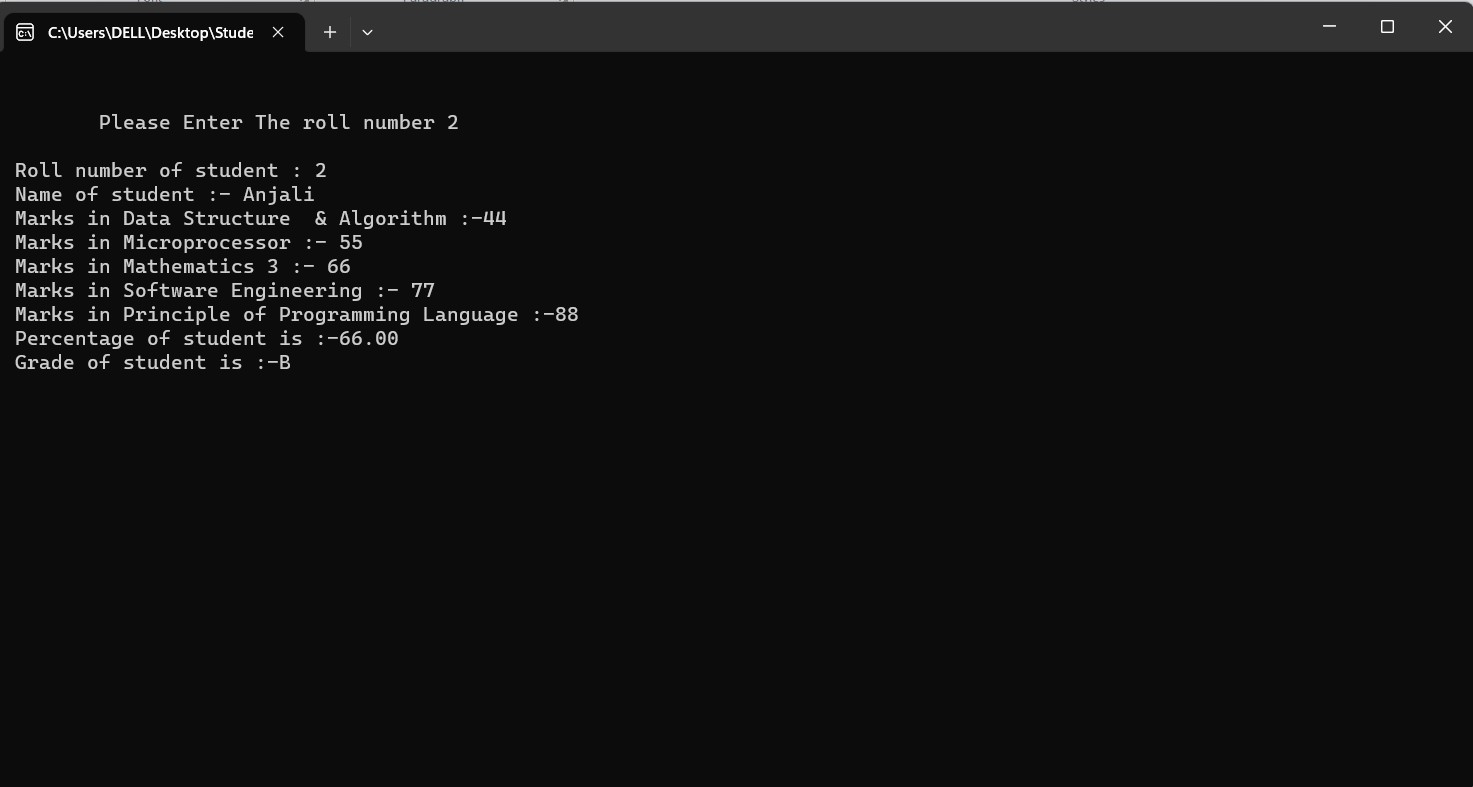


Fig 5.1.7.Search Student Record

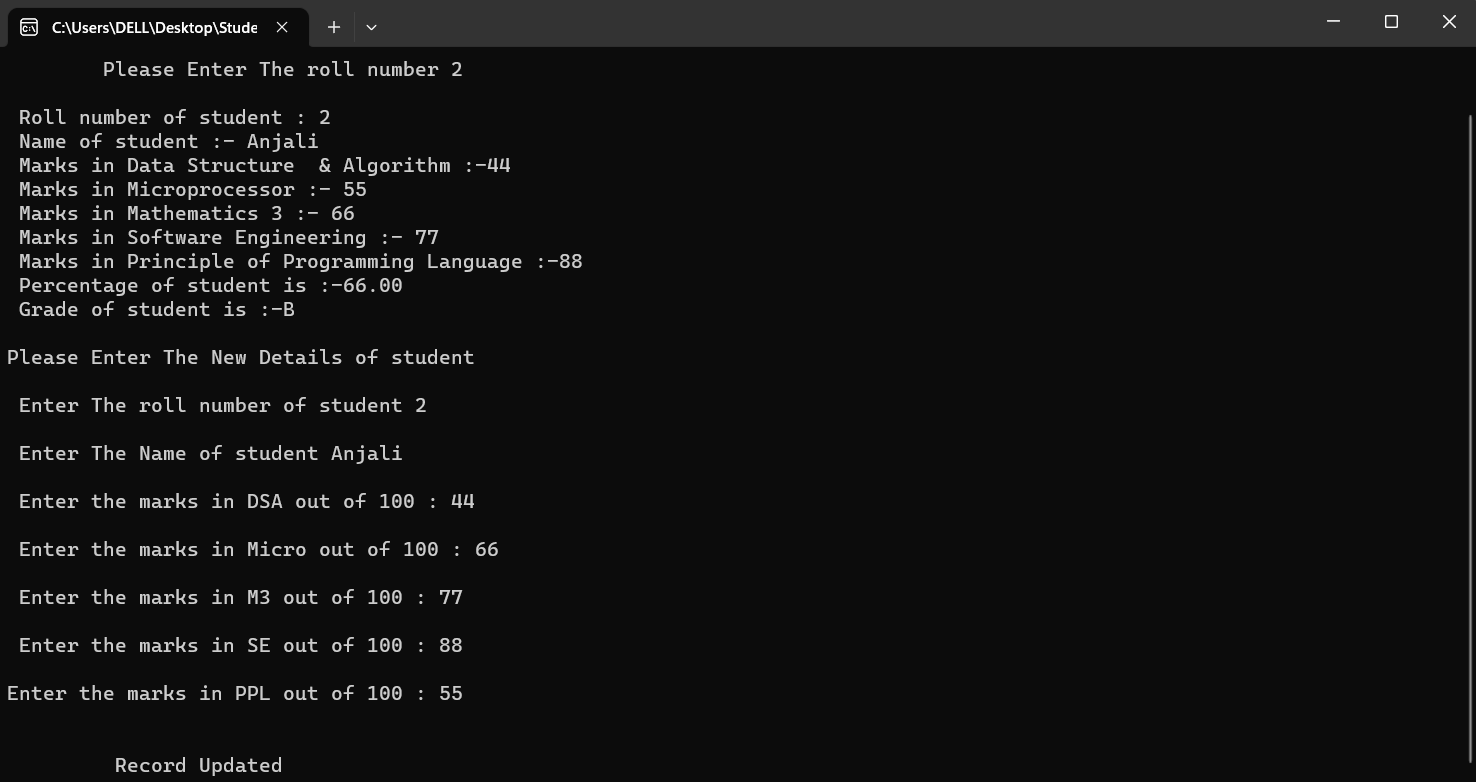


Fig 5.1.8 Modify Student Record

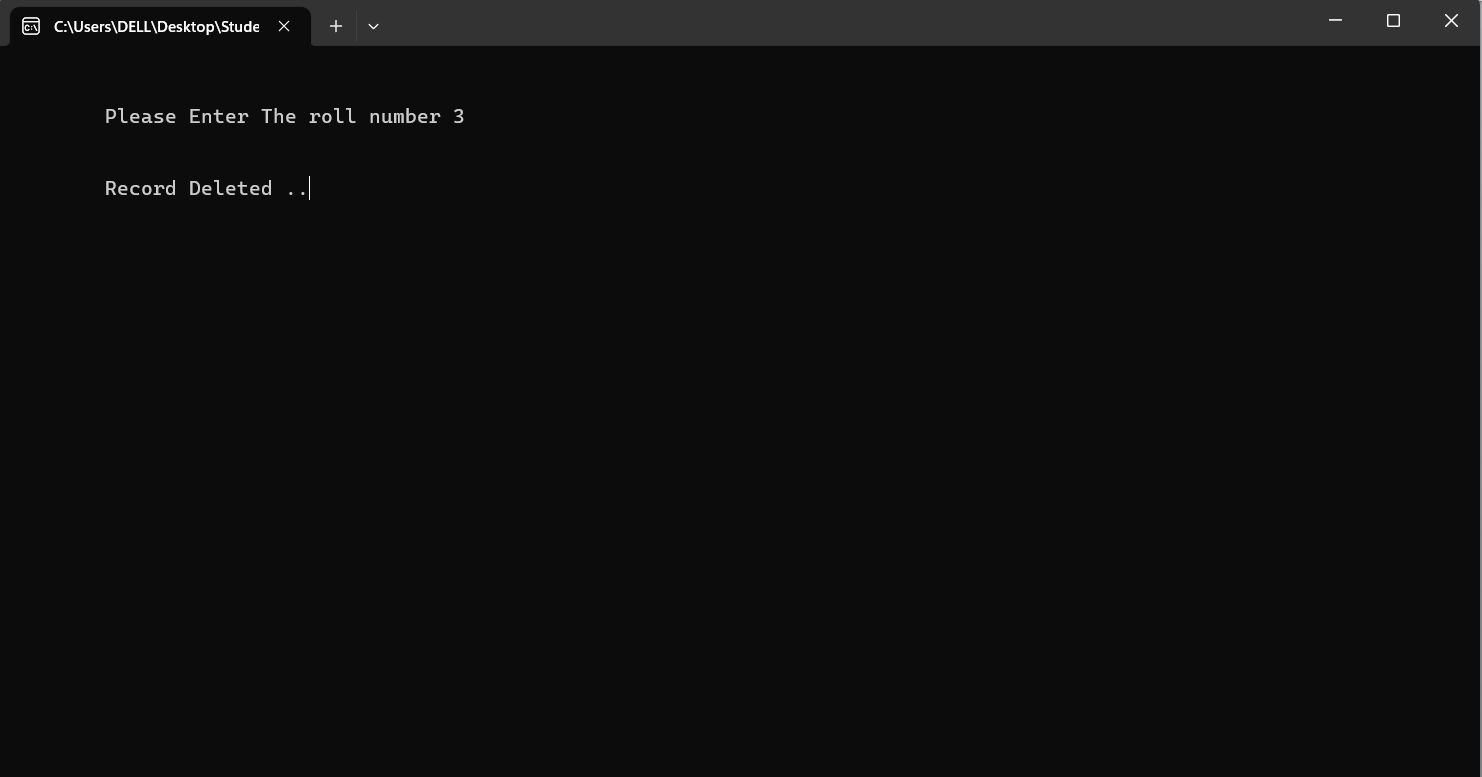


Fig 5.1.9 Delete Student Record

# CONCLUSION

This software has its advantages and disadvantages but it can surely help with the record storage system. We don‘t have to worry about the misplacing of the record which is a great clash while storing the record on separate files. It is efficient in maintaining student‘s details and can easily perform operations on student‘s records.

This software also reduces the work load of the teachers in the school as all the details are stored in computer system and whenever the detail marks of student needed it can be searched and displayed on the screen. In future, this system can launch on a web portal for easy online entry of students details, marks and their parents can login and check the marks and download the reports of their children.

This report contains the Student Report Card Management project made using CPP submitting for our Object-Oriented Programming using C++ mini project in the academic year 2022-23.

# REFERENCES

1. <http://www.gjesr.com/Issues%20PDF/ICITAIC-2019/13.pdf>
2. <https://www.geeksforgeeks.org/file-handling-c-classes/?ref=lbp>
3. <https://www.geeksforgeeks.org/header-files-in-c-cpp-and-its>
4. <https://gist.github.com/Synxed/1c1d7cc840d70f935315e32ff233348b>
5. <https://www.codewithc.com/student-report-card-system-project-in-c/>
6. <https://www.scribd.com/doc/113441907/C-Student-Report-Card-System-Project>
7. [https://www.programmingwithbasics.com/2023/03/student-report-card-system-project-](https://www.programmingwithbasics.com/2023/03/student-report-card-system-project-cpp.html) [cpp.html](https://www.programmingwithbasics.com/2023/03/student-report-card-system-project-cpp.html)