“Student Result Management System”

Dept. of Computer Science and Engineering

A Project On Student Result Management System

Submitted 6th December, 2023
for the degree of BSc in CSE
Submitted to: Ashfia Jannat Keya (Lecturer)
Dept. of CSE BUBT

Submitted by:

Md. Sohel Rana	22235103083 ……………………….
Mithun Modak	22235103527……………………….
Tanbir Alam	22235103187……………………….
Kamrul Islam Emon	22235103202……………………….
Rashed Khan Eiad	22235103188……………………….

Abstract

The Student Result Management System (SRMS) is a console based application designed to provide a simple and effective method for managing student examination results. The aim of the project is to develop an easy system which can handle and manage the activities involved in an efficient result management system in an easy way.

Acknowledgements

We also want to thank Ashfia Jannat Keya, our project supervisor and a distinguished lecturer in the Department of Computer Science and Engineering, from the bottom of our hearts. Her steadfast support, astute advice, and persistent encouragement have been vital to the Student Result Management System project's progress. We have improved our technical proficiency and developed a thorough knowledge of software development concepts while working with her as a mentor.

We would also want to thank our friends and classmates for their positive attitudes, helpful debates, and contagious enthusiasm—all of which were essential in forming the project and ensuring its success.

We appreciate your support and donations, everyone.

Declaration

We hereby declare that the Student Result Management System Project, which is a partial fulfillment of the requirements for the Bangladesh University of Business and Technology (BUBT) Bachelor of Science in Computer Science and Engineering degree, is our original work and does not contain any content that has been approved for the awarding of any other degree or diploma to the candidate or candidates for any other degree or diploma. To the best of our knowledge, it doesn't contain any previously published or authored works by other people—unless the project properly credits them.

Contents

1. Introduction

1.1 Introduction

Efficient management of student performance data is crucial in the field of education. A student result management system simplifies and automates the process of recording, evaluating, and sharing this data. It acts as a centralized database for storing, organizing, and retrieving academic records, making it easy for users to access and analyze student progress. In the realm of education, a student result management system plays a pivotal role in streamlining and automating the process of recording, evaluating, and disseminating student performance data. It serves as a centralized repository for storing, organizing, and retrieving academic records, enabling users to access and analyze student progress effectively.

1.2 Objective

The aim of the project is to develop an easy and efficient system which can handle and manage the activities involved in an efficient student result management system in an easy way:

- Designing a computerized student result management system that makes it easy and would help evacuate the conventional paper-based exam result.
- To increase productivity and reduce manual work.
- Find out the problems which we will face and try to make sure they are corrected.
- Generate detailed reports for decision-making.
- Implement secure data handling.
- Ensure the protection of sensitive guest information and maintain data integrity.

1.3 Project Scope

There are two end users for the Student Result Management System.

- Administration
- Student

The whole system is controlled by institutional administration. The administrator can edit and update this system. And they have the permit to view the whole system. And the last end users are students. They log in with their id, and password and check their result, generate report by this system and many more. They could not update/edit something. But they can contact faculty/administration if they face any problems.

1.4 Our Contributions

In the development of the Student Result Management System, our team has played a pivotal role in the following aspects:

1. **User Interface and Experience Design:**
Crafting an intuitive and user-friendly interface to enhance the ease of use for both admin login and user login.
2. **Access Control Implementation:**
We are developing a robust authentication system to differentiate between normal users and administrators, ensuring security.
3. **Feature Implementation:** Implementing the core features of the system, including user registration, password retrieving, calculating CGPA, searching result, generating report, show statistics, and other functionalities.

2. Existing Literature

2.1 Introduction

While developing the Student Result Management System Software, a survey of existing literature was undertaken to identify major trends, technologies, and best practices in Student Result Management Systems. Several studies have stressed the importance of automation in improving operational efficiency in the educational industry. Adopting various programming techniques in system design has proven effective for developing modular and maintainable software systems.

2.2 Necessity of Student Result Management System

Operational Inefficiencies:

Challenge: Manual processes lead to errors and inefficiencies.

Necessity: Increases efficiency by replacing paper-based methods.

Data Security Concerns:

Challenge: Data handled by hand creates security issues.

Necessity: To protect user information, the project focuses on secure data management.

Need for User-Friendly Interfaces:

Challenge: The adoption of a system may be hampered by inefficient and confusing interfaces.

Necessity: To solve this, the software offers an easy-to-use command-line interface (CLI) for fluid communication.

Scalability and Future Expansion:

Challenge: Future growth is hampered by scalability issues.

Necessity: The scalable and maintainable system of the project is guaranteed by the OOP concepts.

2.3 Conclusion

There is a clear demand for the suggested Student Result Management System Software in the educational platform, according to a survey of the literature. The need to automate Student Result Management procedures is highlighted by the difficulties mentioned, which include operational inefficiencies, security issues, and a lack of analytical insights.

3. Proposed Model

3.1 Introduction

The proposed Student Result Management System Software is a strategic response to the evolving needs of the educational institute. Designed to streamline daily operations, the application focuses on automating tasks related to searching results and reporting. Developed using C++ Programming Language, the system ensures a modular and maintainable codebase. With a user-friendly command-line interface (CLI), the software aims to enhance operational efficiency, provide a seamless educational experience, and contribute to data security. The model's core objectives include simplifying processes, fostering automation, and delivering insightful analytics for informed decision-making in Student Result Management.

3.2 Hardware and Software

Software:

We used several tools and C++ Programming Languages to implement this project.

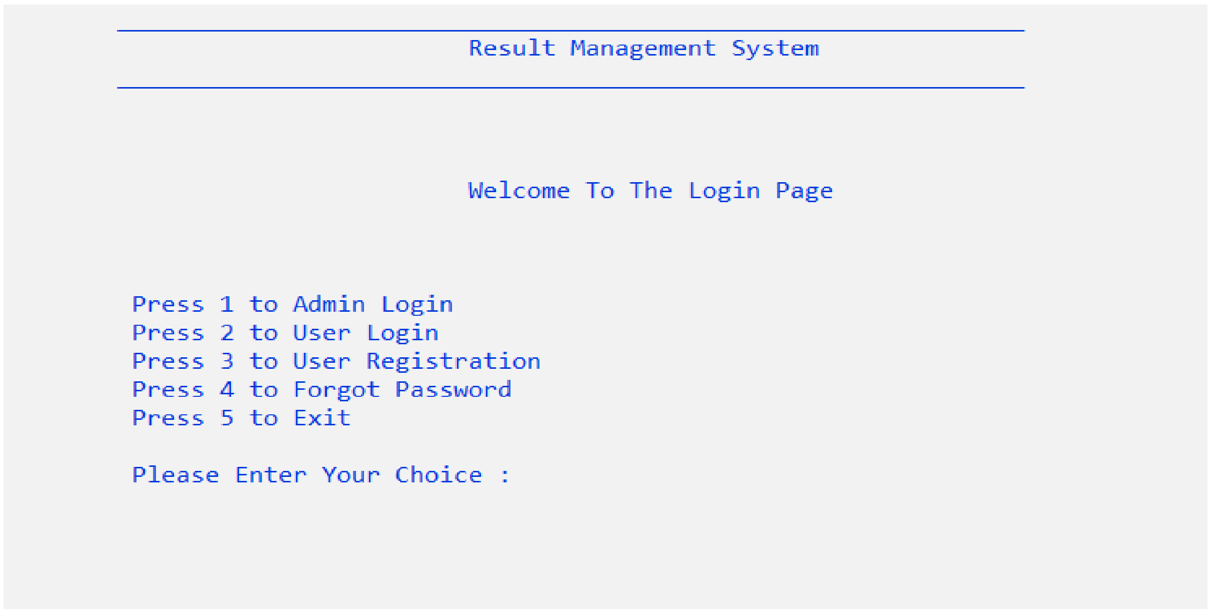
- OS: Windows 10
- Codeblocks
- GCC Compiler

Hardware:

- Processor: Dual-core Intel or AMD 64-bit processor
- 4 GB RAM (Minimum)
- 100 MB Hard Disk (Minimum)

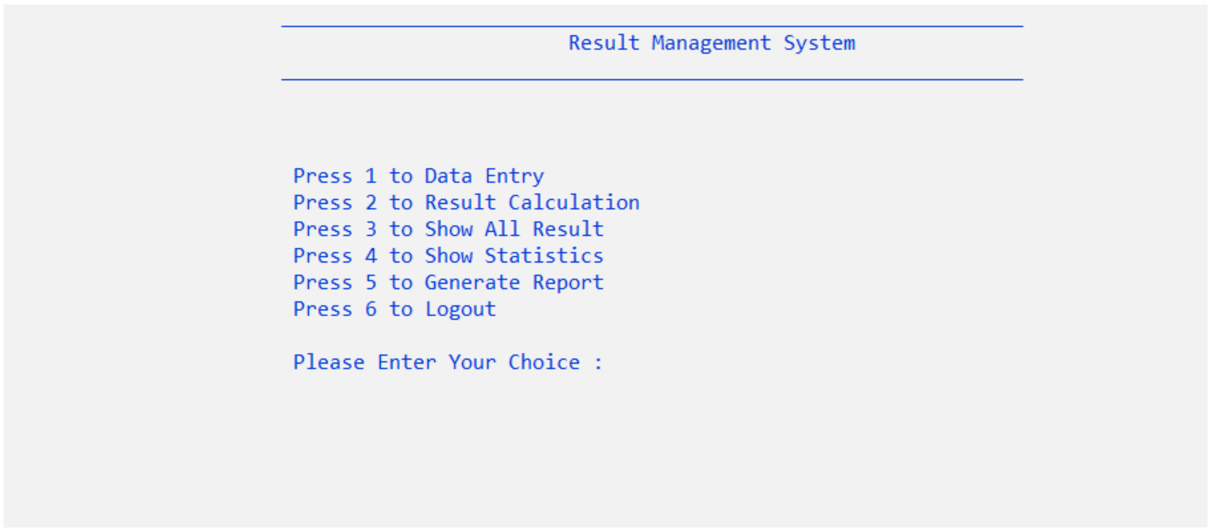
3.3 Screenshots of Encountered Interfaces

The Screenshots of the interfaces we are going to face inside the Program are Show in below:



```
_____________________________________________________________
                        Result Management System
_____________________________________________________________



                        Welcome To The Login Page



 Press 1 to Admin Login
 Press 2 to User Login
 Press 3 to User Registration
 Press 4 to Forgot Password
 Press 5 to Exit

 Please Enter Your Choice :
```

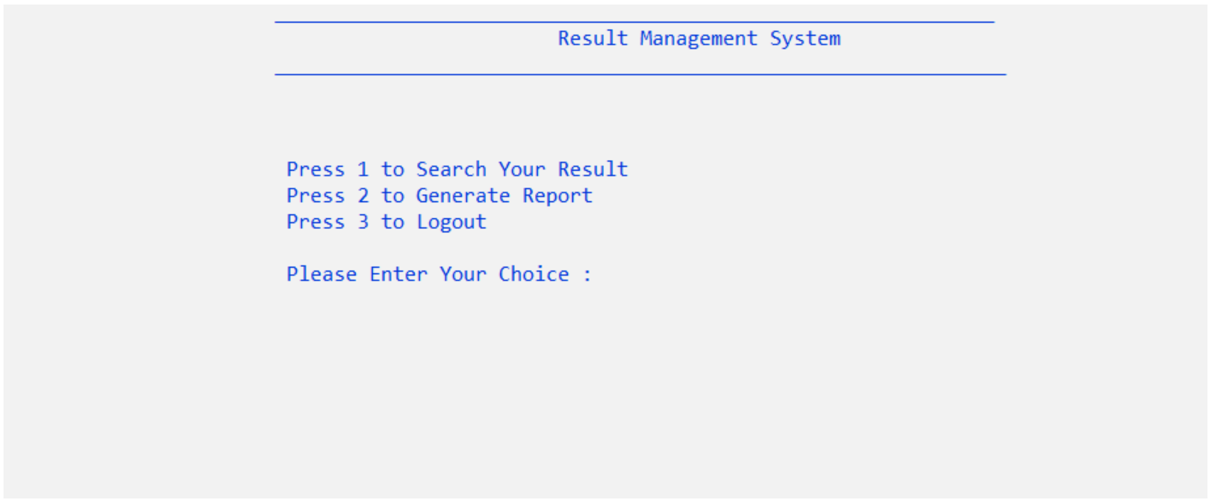
Figure 3.1: Main Menu Interface



```
_____________________________________________________________
                        Result Management System
_____________________________________________________________



 Press 1 to Data Entry
 Press 2 to Result Calculation
 Press 3 to Show All Result
 Press 4 to Show Statistics
 Press 5 to Generate Report
 Press 6 to Logout

 Please Enter Your Choice :
```

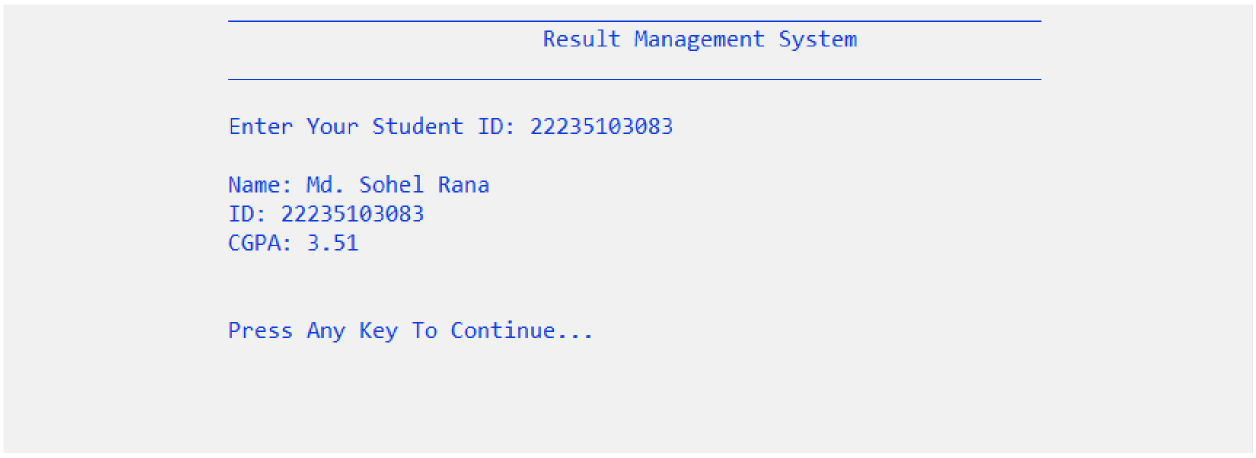
Figure 3.2: Admin Panel Interface

Proposed Model



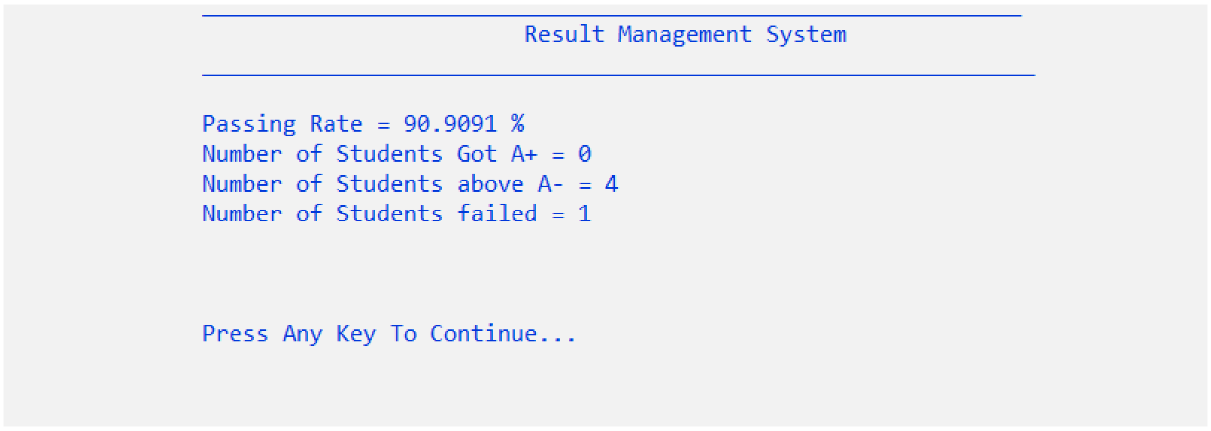
```
                        Result Management System

  Press 1 to Search Your Result
  Press 2 to Generate Report
  Press 3 to Logout

  Please Enter Your Choice :
```

Figure 3.3: User Interface



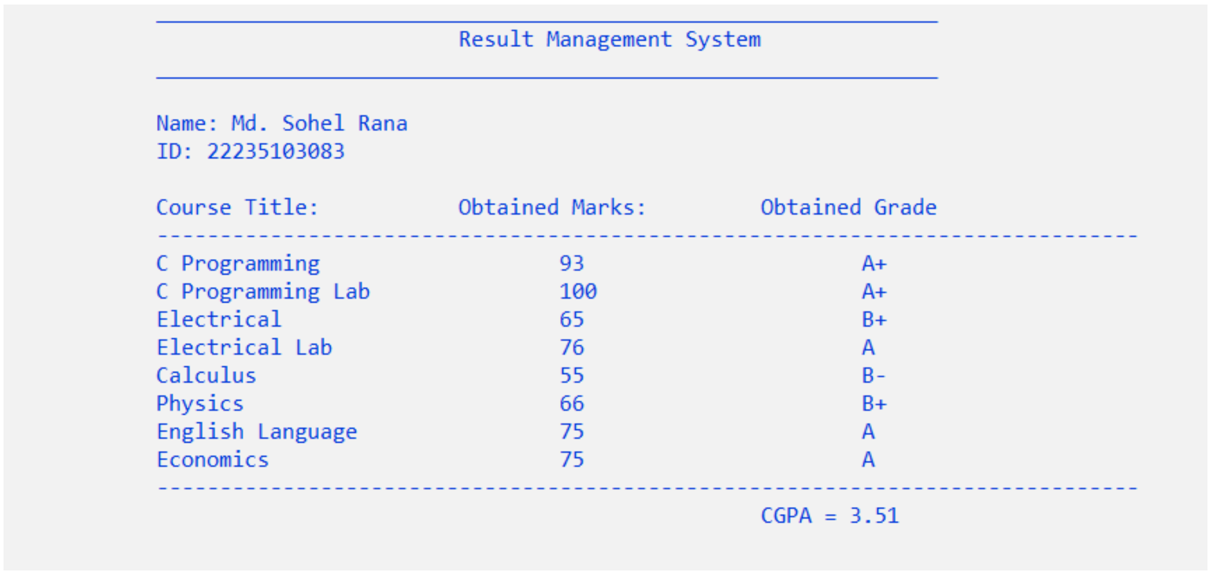
```
                        Result Management System

Enter Your Student ID: 22235103083

Name: Md. Sohel Rana
ID: 22235103083
CGPA: 3.51


Press Any Key To Continue...
```

Figure 3.4: Searching Result



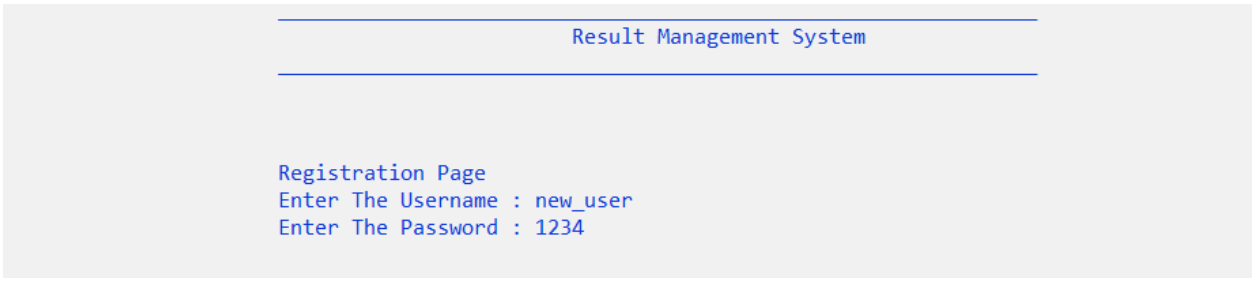
```
                        Result Management System

Passing Rate = 90.9091 %
Number of Students Got A+ = 0
Number of Students above A- = 4
Number of Students failed = 1


Press Any Key To Continue...
```

Figure 3.5: Statistics



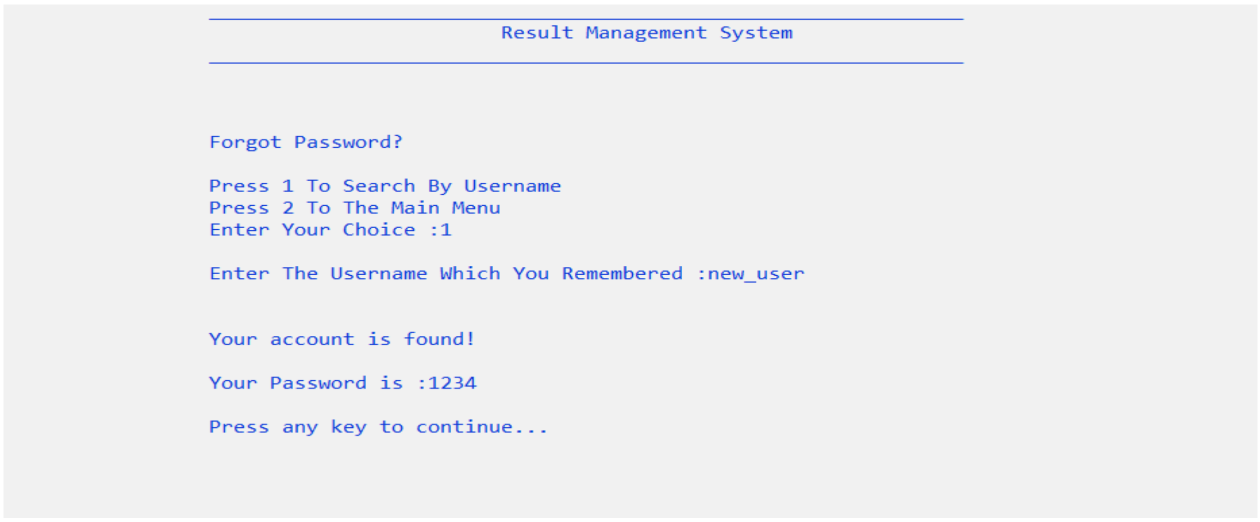
```
                         Result Management System

Name: Md. Sohel Rana
ID: 22235103083

Course Title:            Obtained Marks:          Obtained Grade
------------------------------------------------------------------------------
C Programming                    93                       A+
C Programming Lab                100                      A+
Electrical                       65                       B+
Electrical Lab                   76                       A
Calculus                         55                       B-
Physics                          66                       B+
English Language                 75                       A
Economics                        75                       A
------------------------------------------------------------------------------
                                                 CGPA = 3.51
```

Figure 3.5: Sample Report



```
                        Result Management System



Registration Page
Enter The Username : new_user
Enter The Password : 1234
```

Figure 3.6: User Registration Page



```
                        Result Management System


Forgot Password?

Press 1 To Search By Username
Press 2 To The Main Menu
Enter Your Choice :1

Enter The Username Which You Remembered :new_user


Your account is found!

Your Password is :1234

Press any key to continue...
```

Figure 3.5: Password Recovery

4. Implementation of Our System

4.1 Introduction

Explanation of Some functions:

- **User Registration:**

  The user registration function is responsible for registering new users and assigning them a username and password and securely storing this data in a file.

- **Login:**

  The login function is responsible for authenticating users and granting them access to the system. The system typically has two types of users: students and administrators.

  The student login process involves entering a unique username and password set by the user during registration. Once the credentials are verified, the student is granted access to their features.

  Administrators, on the other hand, admin have the power to access the system using their unique username and password. Once authenticated, they can perform various tasks

  We have implemented this feature by using file handling in C++ language.

- **Forgot Password:**

  The forgot password function is responsible for allowing users to retrieve their password in case they forget it. They need to remember username to recover the password.

- **Main Menu:**

  The main menu function provides an overview of the various modules available in the system. Such as Admin Login, User Login, Registration Etc.

Implementation of Our System

- **Searching:**

  This function is for users who want to search their result by their unique ID Number.
  It searches their result in the file ("result.dat").

- **Data Entry:**

  This function is for the admin who usually enters all the information about each student. Such as Student’s name, ID, their registered course names, Obtained marks, and credit hours of each course, etc.
  We store this information for future use in (data.dat) file.

- **Result Calculate:**

  We followed the grading system of BUBT to calculate CGPA.
  We first sum up the obtained (grade point * credit hour) of each course by a student, then we divide that by the total credit hour.

- **Display All Result:**

  This is a simple function which just prints all the students' results stored in the (result.dat) file which is generated by the Calculate Function.

- **Stats**

  This function generates a report after analyzing the result.
  Such as Passing Rate, Number of students got A+, Etc.

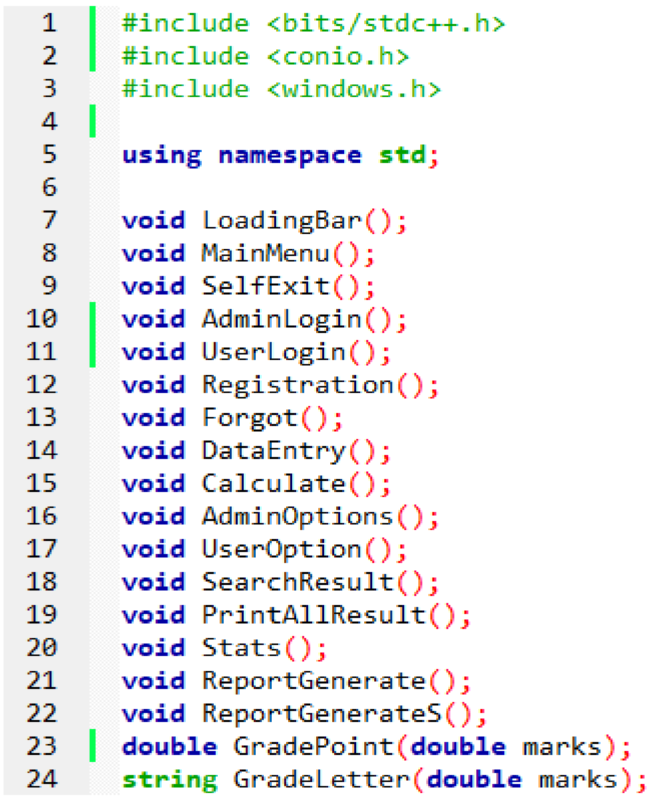
5. A Code Walkthrough

5.1 Introduction

Here we have analyzed all the results regarding to the project and also we have tried to show the source code of our program.

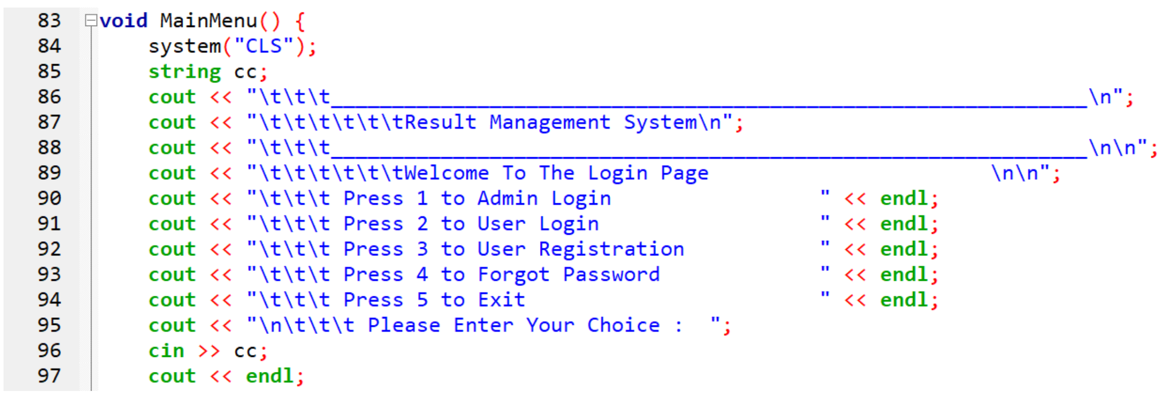
5.2 Result Analysis

Some screenshots of our program source code are shown below in order.



```cpp
#include <bits/stdc++.h>
#include <conio.h>
#include <windows.h>

using namespace std;

void LoadingBar();
void MainMenu();
void SelfExit();
void AdminLogin();
void UserLogin();
void Registration();
void Forgot();
void DataEntry();
void Calculate();
void AdminOptions();
void UserOption();
void SearchResult();
void PrintAllResult();
void Stats();
void ReportGenerate();
void ReportGenerateS();
double GradePoint(double marks);
string GradeLetter(double marks);
```

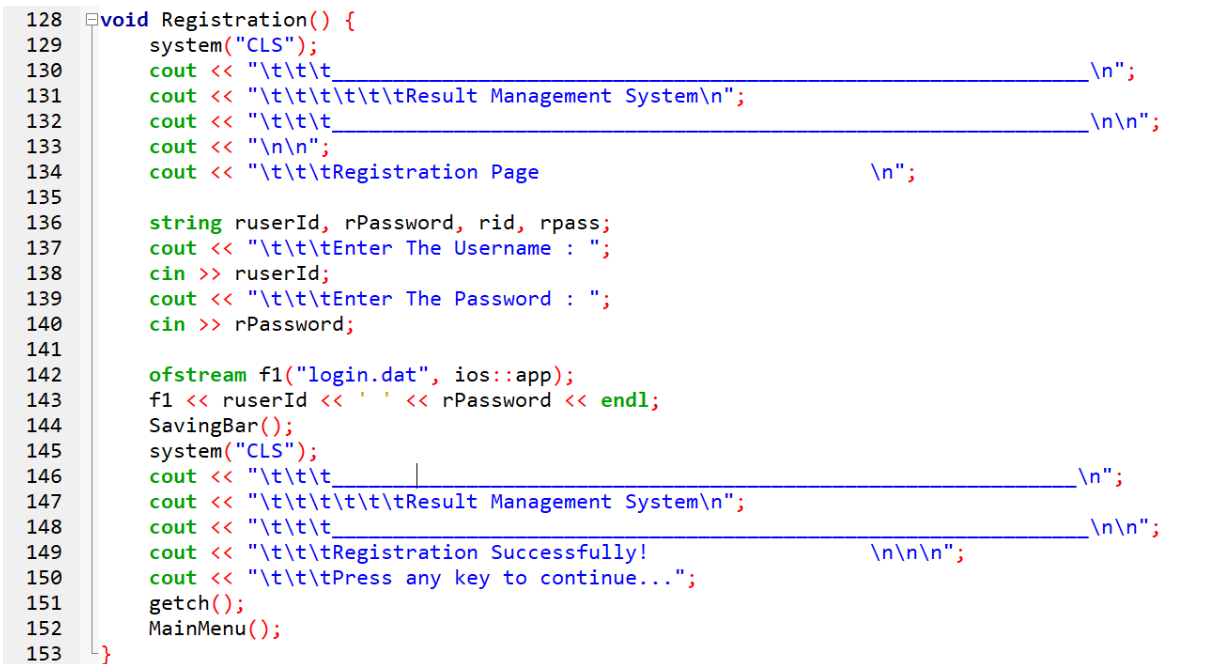
Figure 5.1: Screenshot of Function declaration



```cpp
void MainMenu() {
    system("CLS");
    string cc;
    cout << "\t\t\t______________________________________________________________\n";
    cout << "\t\t\t\t\t\t\tResult Management System\n";
    cout << "\t\t\t______________________________________________________________\n\n";
    cout << "\t\t\t\t\t\t\tWelcome To The Login Page                       \n\n";
    cout << "\t\t\t Press 1 to Admin Login                  " << endl;
    cout << "\t\t\t Press 2 to User Login                   " << endl;
    cout << "\t\t\t Press 3 to User Registration            " << endl;
    cout << "\t\t\t Press 4 to Forgot Password              " << endl;
    cout << "\t\t\t Press 5 to Exit                         " << endl;
    cout << "\n\t\t\t Please Enter Your Choice :  ";
    cin >> cc;
    cout << endl;
```

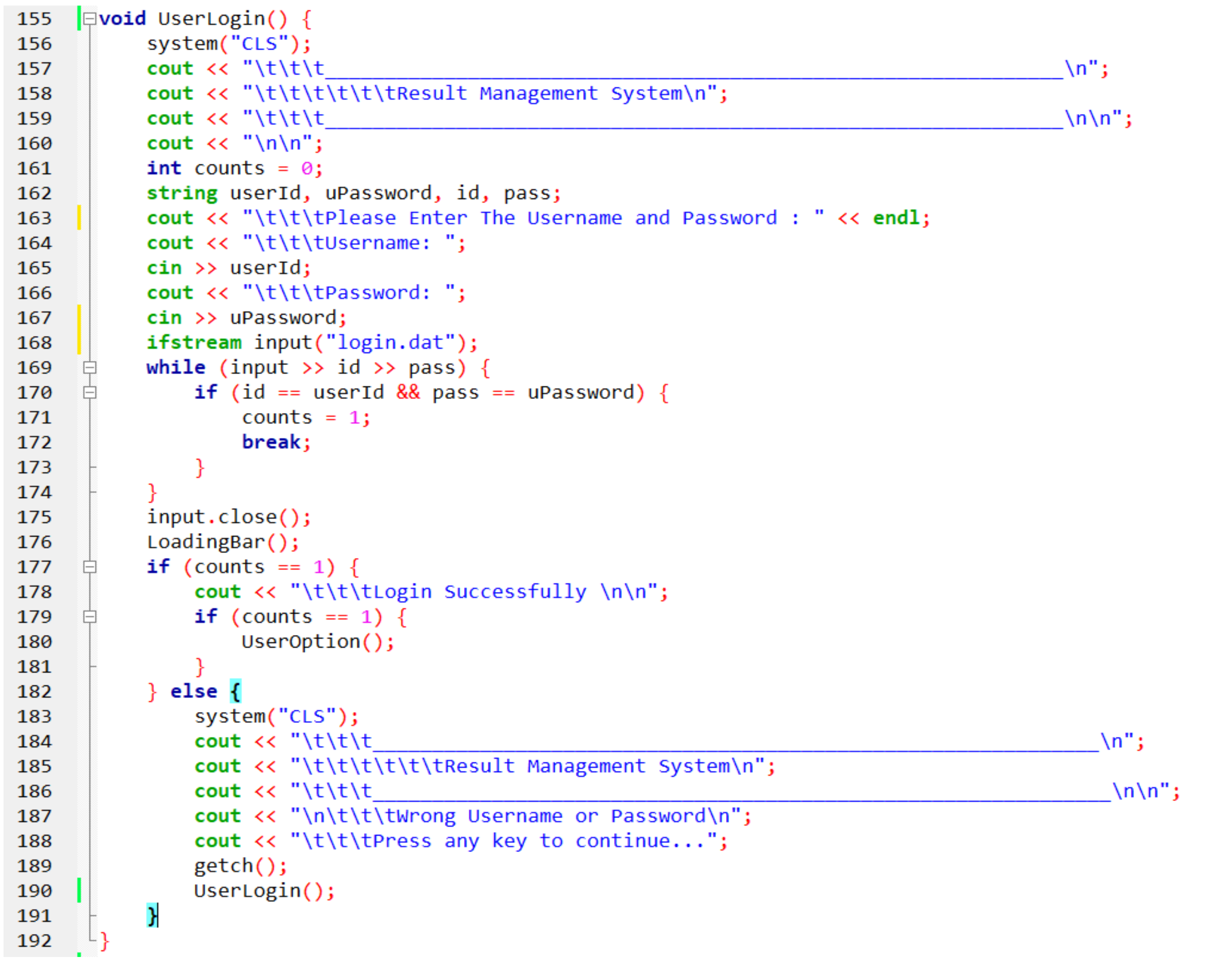
Figure 5.2: Screenshot of Main Menu Feature Code

A Code Walkthrough



```cpp
void Registration() {
    system("CLS");
    cout << "\t\t\t__________________________________________________________________\n";
    cout << "\t\t\t\t\t\tResult Management System\n";
    cout << "\t\t\t__________________________________________________________________\n\n";
    cout << "\n\n";
    cout << "\t\t\tRegistration Page                           \n";

    string ruserId, rPassword, rid, rpass;
    cout << "\t\t\tEnter The Username : ";
    cin >> ruserId;
    cout << "\t\t\tEnter The Password : ";
    cin >> rPassword;

    ofstream f1("login.dat", ios::app);
    f1 << ruserId << ' ' << rPassword << endl;
    SavingBar();
    system("CLS");
    cout << "\t\t\t_________________________________________________________________\n";
    cout << "\t\t\t\t\t\tResult Management System\n";
    cout << "\t\t\t__________________________________________________________________\n\n";
    cout << "\t\t\tRegistration Successfully!                 \n\n\n";
    cout << "\t\t\tPress any key to continue...";
    getch();
    MainMenu();
}
```

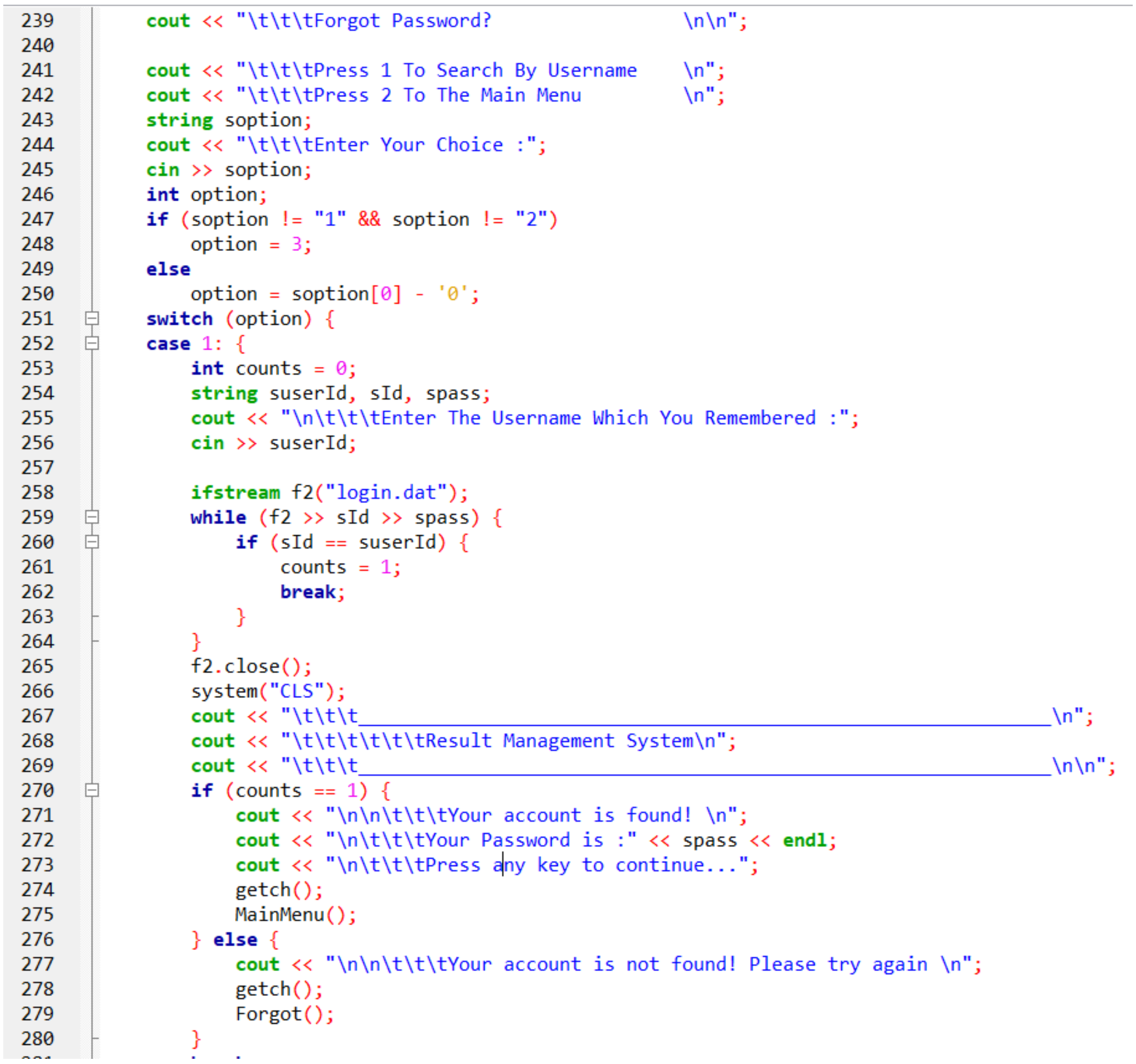
Figure 5.3: Screenshot of Registration Function



```cpp
void UserLogin() {
    system("CLS");
    cout << "\t\t\t________________________________________________________________\n";
    cout << "\t\t\t\t\t\tResult Management System\n";
    cout << "\t\t\t________________________________________________________________\n\n";
    cout << "\n\n";
    int counts = 0;
    string userId, uPassword, id, pass;
    cout << "\t\t\tPlease Enter The Username and Password : " << endl;
    cout << "\t\t\tUsername: ";
    cin >> userId;
    cout << "\t\t\tPassword: ";
    cin >> uPassword;
    ifstream input("login.dat");
    while (input >> id >> pass) {
        if (id == userId && pass == uPassword) {
            counts = 1;
            break;
        }
    }
    input.close();
    LoadingBar();
    if (counts == 1) {
        cout << "\t\t\tLogin Successfully \n\n";
        if (counts == 1) {
            UserOption();
        }
    } else {
        system("CLS");
        cout << "\t\t\t________________________________________________________________\n";
        cout << "\t\t\t\t\t\tResult Management System\n";
        cout << "\t\t\t________________________________________________________________\n\n";
        cout << "\n\t\t\tWrong Username or Password\n";
        cout << "\t\t\tPress any key to continue...";
        getch();
        UserLogin();
    }
}
```

Figure 5.4: Screenshot of Login Function

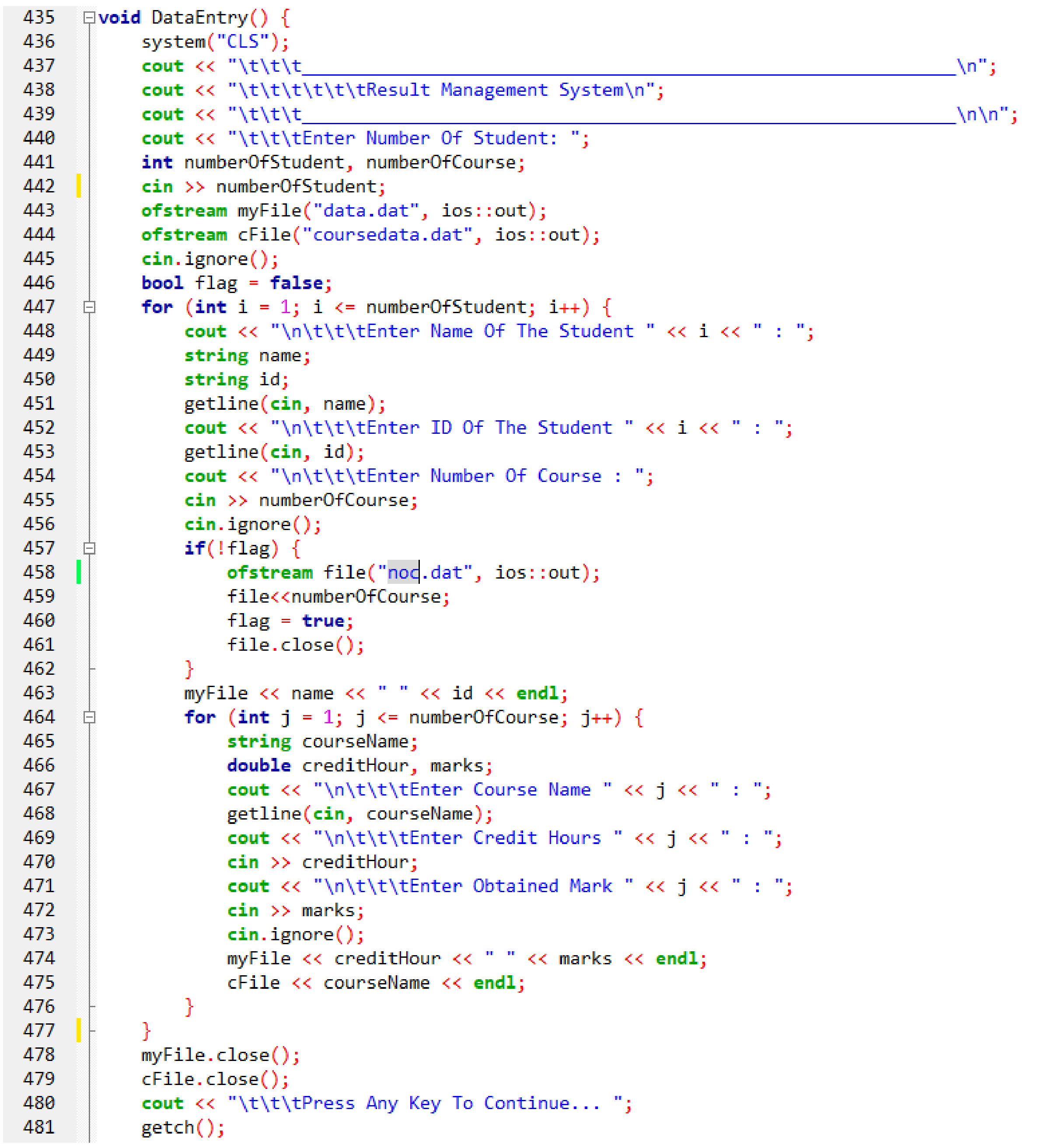
A Code Walkthrough



```
cout << "\t\t\tForgot Password?                 \n\n";

cout << "\t\t\tPress 1 To Search By Username    \n";
cout << "\t\t\tPress 2 To The Main Menu         \n";
string soption;
cout << "\t\t\tEnter Your Choice :";
cin >> soption;
int option;
if (soption != "1" && soption != "2")
    option = 3;
else
    option = soption[0] - '0';
switch (option) {
case 1: {
    int counts = 0;
    string suserId, sId, spass;
    cout << "\n\t\t\tEnter The Username Which You Remembered :";
    cin >> suserId;

    ifstream f2("login.dat");
    while (f2 >> sId >> spass) {
        if (sId == suserId) {
            counts = 1;
            break;
        }
    }
    f2.close();
    system("CLS");
    cout << "\t\t\t_______________________________________________________________\n";
    cout << "\t\t\t\t\t\tResult Management System\n";
    cout << "\t\t\t_______________________________________________________________\n\n";
    if (counts == 1) {
        cout << "\n\n\t\t\tYour account is found! \n";
        cout << "\n\t\t\tYour Password is :" << spass << endl;
        cout << "\n\t\t\tPress any key to continue...";
        getch();
        MainMenu();
    } else {
        cout << "\n\n\t\t\tYour account is not found! Please try again \n";
        getch();
        Forgot();
    }
```

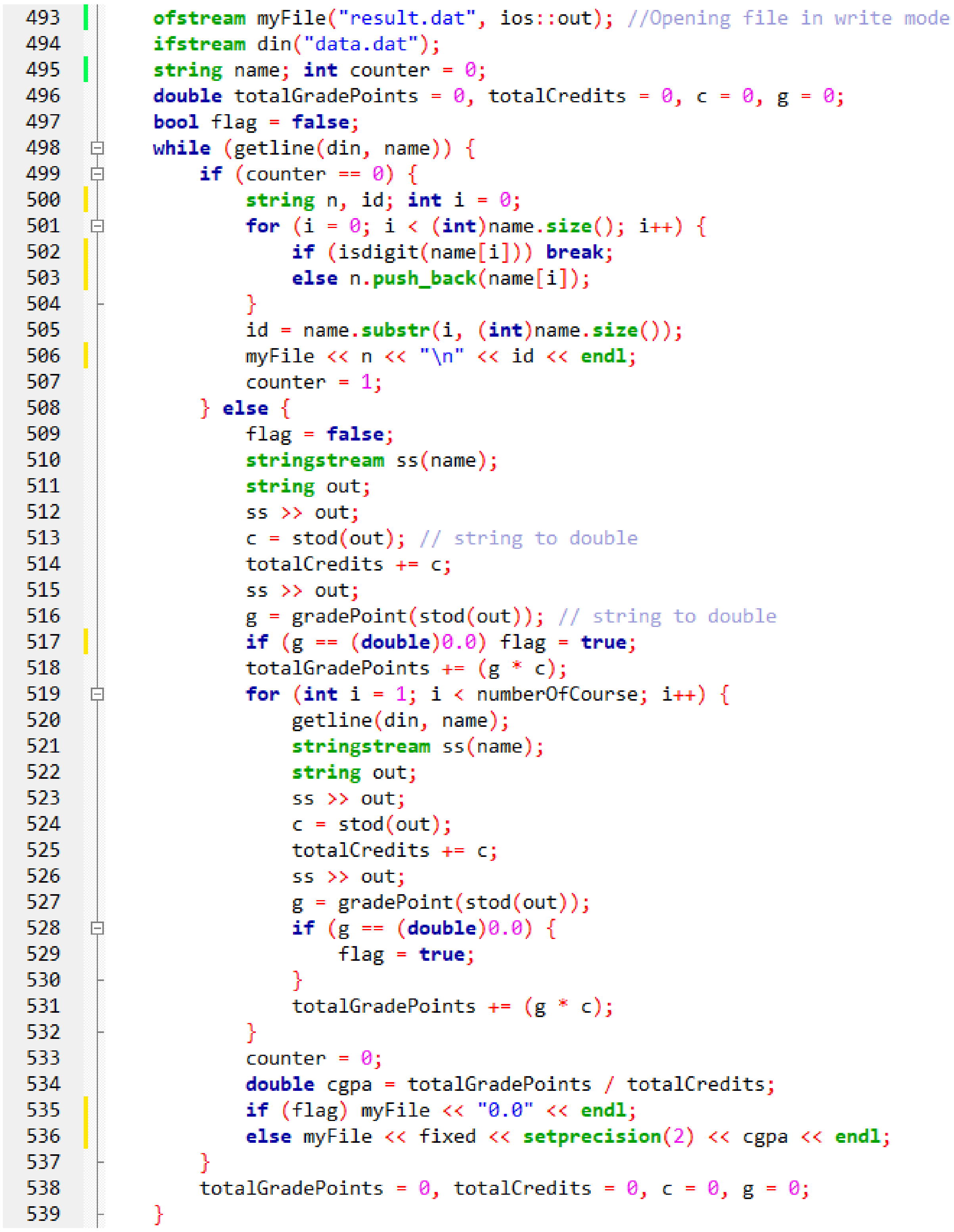
Figure 5.5 : Screenshot of Forgot Password Function

A Code Walkthrough



```
void DataEntry() {
    system("CLS");
    cout << "\t\t\t_____________________________________________________________\n";
    cout << "\t\t\t\t\t\tResult Management System\n";
    cout << "\t\t\t_____________________________________________________________\n\n";
    cout << "\t\t\tEnter Number Of Student: ";
    int numberOfStudent, numberOfCourse;
    cin >> numberOfStudent;
    ofstream myFile("data.dat", ios::out);
    ofstream cFile("coursedata.dat", ios::out);
    cin.ignore();
    bool flag = false;
    for (int i = 1; i <= numberOfStudent; i++) {
        cout << "\n\t\t\tEnter Name Of The Student " << i << " : ";
        string name;
        string id;
        getline(cin, name);
        cout << "\n\t\t\tEnter ID Of The Student " << i << " : ";
        getline(cin, id);
        cout << "\n\t\t\tEnter Number Of Course : ";
        cin >> numberOfCourse;
        cin.ignore();
        if(!flag) {
            ofstream file("noc.dat", ios::out);
            file<<numberOfCourse;
            flag = true;
            file.close();
        }
        myFile << name << " " << id << endl;
        for (int j = 1; j <= numberOfCourse; j++) {
            string courseName;
            double creditHour, marks;
            cout << "\n\t\t\t\tEnter Course Name " << j << " : ";
            getline(cin, courseName);
            cout << "\n\t\t\t\tEnter Credit Hours " << j << " : ";
            cin >> creditHour;
            cout << "\n\t\t\t\tEnter Obtained Mark " << j << " : ";
            cin >> marks;
            cin.ignore();
            myFile << creditHour << " " << marks << endl;
            cFile << courseName << endl;
        }
    }
    myFile.close();
    cFile.close();
    cout << "\t\t\tPress Any Key To Continue... ";
    getch();
```

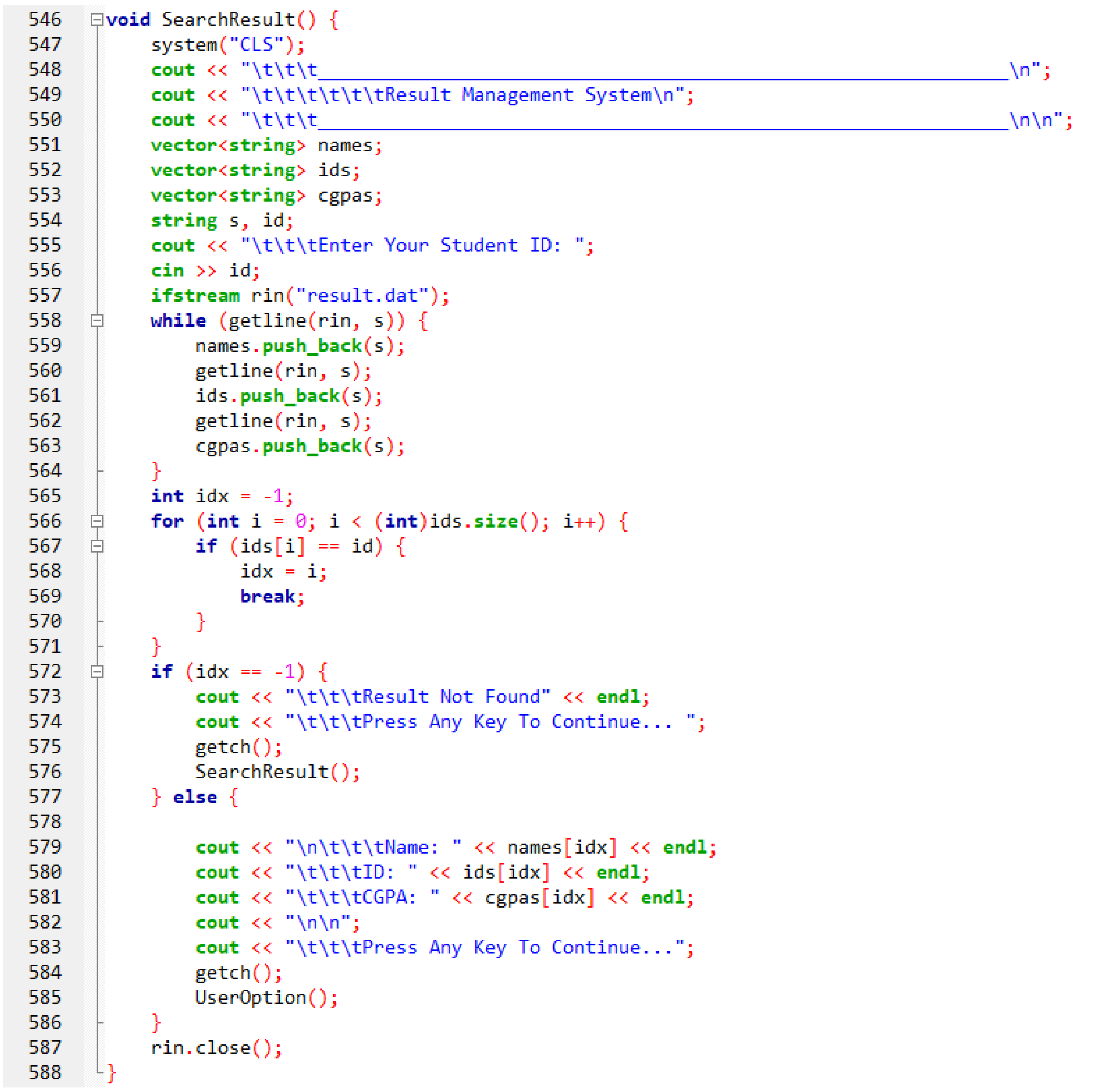
Figure 5.5: Screenshot of Data Entry Function



```
    ofstream myFile("result.dat", ios::out); //Opening file in write mode
    ifstream din("data.dat");
    string name; int counter = 0;
    double totalGradePoints = 0, totalCredits = 0, c = 0, g = 0;
    bool flag = false;
    while (getline(din, name)) {
        if (counter == 0) {
            string n, id; int i = 0;
            for (i = 0; i < (int)name.size(); i++) {
                if (isdigit(name[i])) break;
                else n.push_back(name[i]);
            }
            id = name.substr(i, (int)name.size());
            myFile << n << "\n" << id << endl;
            counter = 1;
        } else {
            flag = false;
            stringstream ss(name);
            string out;
            ss >> out;
            c = stod(out); // string to double
            totalCredits += c;
            ss >> out;
            g = gradePoint(stod(out)); // string to double
            if (g == (double)0.0) flag = true;
            totalGradePoints += (g * c);
            for (int i = 1; i < numberOfCourse; i++) {
                getline(din, name);
                stringstream ss(name);
                string out;
                ss >> out;
                c = stod(out);
                totalCredits += c;
                ss >> out;
                g = gradePoint(stod(out));
                if (g == (double)0.0) {
                    flag = true;
                }
                totalGradePoints += (g * c);
            }
            counter = 0;
            double cgpa = totalGradePoints / totalCredits;
            if (flag) myFile << "0.0" << endl;
            else myFile << fixed << setprecision(2) << cgpa << endl;
        }
        totalGradePoints = 0, totalCredits = 0, c = 0, g = 0;
    }
```

Figure 5.6: Screenshot of Calculate Function

A Code Walkthrough



```
void SearchResult() {
    system("CLS");
    cout << "\t\t\t______________________________________________________________\n";
    cout << "\t\t\t\t\t\t\tResult Management System\n";
    cout << "\t\t\t______________________________________________________________\n\n";
    vector<string> names;
    vector<string> ids;
    vector<string> cgpas;
    string s, id;
    cout << "\t\t\tEnter Your Student ID: ";
    cin >> id;
    ifstream rin("result.dat");
    while (getline(rin, s)) {
        names.push_back(s);
        getline(rin, s);
        ids.push_back(s);
        getline(rin, s);
        cgpas.push_back(s);
    }
    int idx = -1;
    for (int i = 0; i < (int)ids.size(); i++) {
        if (ids[i] == id) {
            idx = i;
            break;
        }
    }
    if (idx == -1) {
        cout << "\t\t\tResult Not Found" << endl;
        cout << "\t\t\tPress Any Key To Continue... ";
        getch();
        SearchResult();
    } else {

        cout << "\n\t\t\tName: " << names[idx] << endl;
        cout << "\t\t\tID: " << ids[idx] << endl;
        cout << "\t\t\tCGPA: " << cgpas[idx] << endl;
        cout << "\n\n";
        cout << "\t\t\tPress Any Key To Continue...";
        getch();
        UserOption();
    }
    rin.close();
}
```

Figure 5.7: Search Function

6. Conclusion

A result management system is a comprehensive software application that provides educational institutions with a centralized platform to manage and analyze student performance data. By leveraging its advanced features and functionalities, institutions can easily calculate, generate reports, and publish. This system promotes transparency and accountability, as it allows users to access and analyze student performance data in real time. Additionally, the implementation of such a system significantly enhances the quality of education and promotes a culture of academic excellence by empowering educators to identify areas where students need help and tailor their teaching accordingly.