**Student Management System**



**Session 2023 - 2027**

**Submitted by:**

Mian Saad Tahir 2023-CS-62

**Supervised by:**

Prof. Dr. Awais Hassan Ma’am Maida Shahid

**Course:**

CSC-102 Programming Fundamentals

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Table of Contents

[ About SMS 4](#_Toc154010849)

[ Users Types 4](#_Toc154010850)

[ Functional Requirements 4](#_Toc154010851)

[ Wireframes 5](#_Toc154010852)

[ Data Structures 16](#_Toc154010853)

[ Function Prototypes 17](#_Toc154010854)

[ Functions Working Flow 19](#_Toc154010898)

[ Complete Code 20](#_Toc154010899)

#  About SMS

o SMS is a learning management software that provides a centralized platform for students and faculty to access and manage learning resources, keep record of progress, and manage student’s data easily.

o It consists of 5 classes, each having 3 subjects.

#  Users Types

1. Admin:

Admin has access to add student, view and update their record, maintain their attendance and fee submission status, assign them marks, change their classes and their login password. Admin can also view subjects of all classes.

1. Student:

User has the access to view his/her class, subjects, marks, attendance, fee status, and his/her result card.

#  Functional Requirements

|  |  |  |
| --- | --- | --- |
| **User type** | **Functions** | **Result** |
| **ADMIN** | Add Student | Adds student record |
| Remove Student | Removes student record |
| View Student | Displays student record |
| Update Fee Status | Updates student fee status |
| Update Attendance | Updates student attendance |
| Update Marks | Update student marks |
| Change Class | Updates student class |
|  | View Subjects | Displays subjects of all classes |
| Change Admin Password | Changes admin password |
| Change Student Password | Changes student password |
|  | Logout | Logs out of admin menu |

|  |  |  |
| --- | --- | --- |
| **STUDENT** | View Class | Displays student class |
| View Subjects | Displays student subjects |
| View Attendance | Displays student attendance |
| View Fee Status | Displays student fee status |
| View Result Card | Displays student result card |

#  Wireframes

* This is the log in menu
* If you enter invalid choice, it displays the following



* Otherwise, sign in screen appears on pressing 1. By default, admin’s username is “admin” and password is “1,2,3”.



* If you enter invalid credentials, following appears

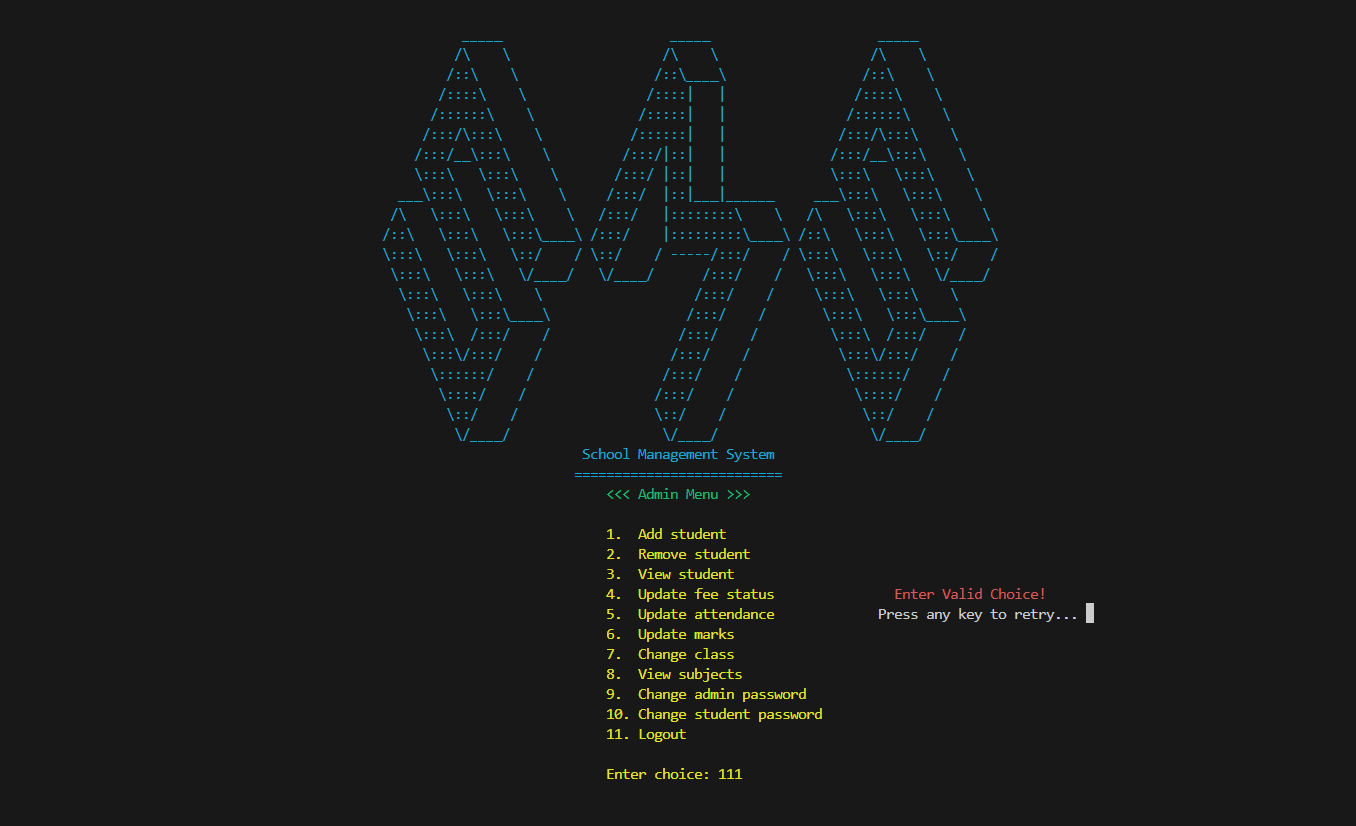


**Admin Menu**

* If you enter valid admin credentials, admin menu appears



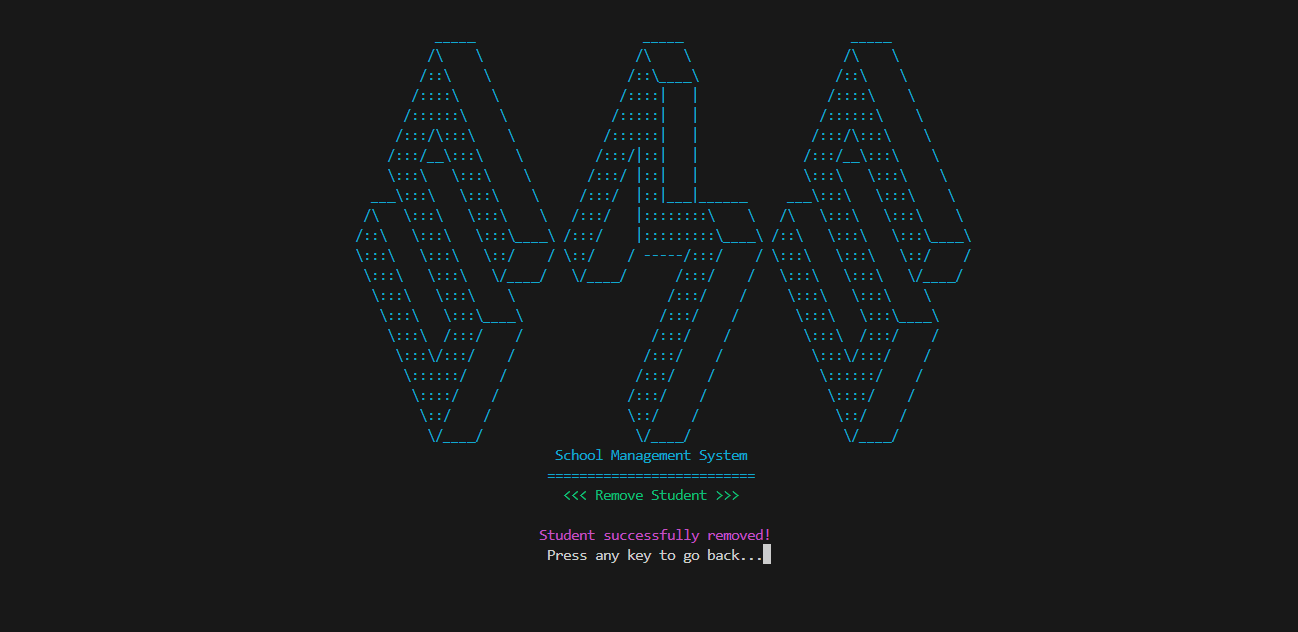
* If you enter any choice greater than 11 or less than 1, it appears invalid



* From choice 1, admin can add student



* From choice 2, admin can remove student



* From choice 3, admin can view attributes of student



s

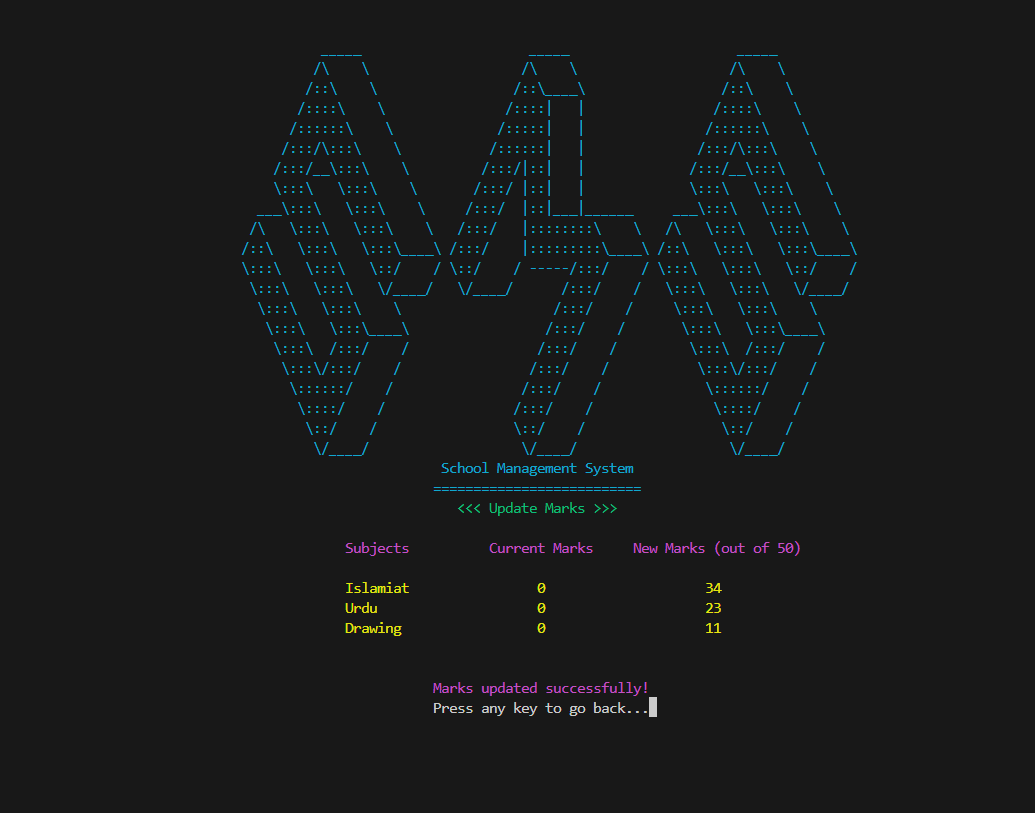
* From choice 4, admin can update student’s fee status



* From choice 5, admin can update student’s attendance



* From choice 6, admin can add student’s marks



* From choice 7, admin can change student’s class



* From choice 8, admin can view subjects of all classes



* From choice 9, admin can update admin’s password



* From choice 10, admin can update student’s password



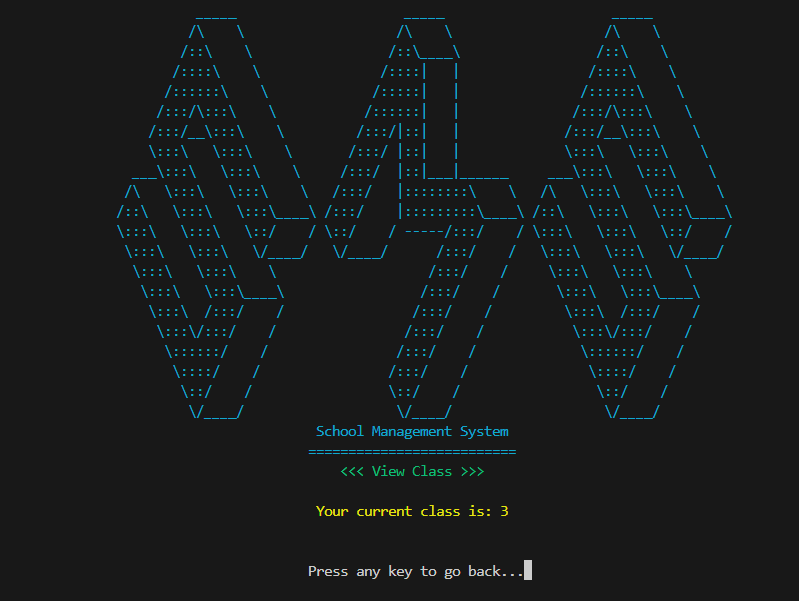
* From choice 11, admin can log out

**Student Menu**

* If you enter valid student credentials, student menu appears



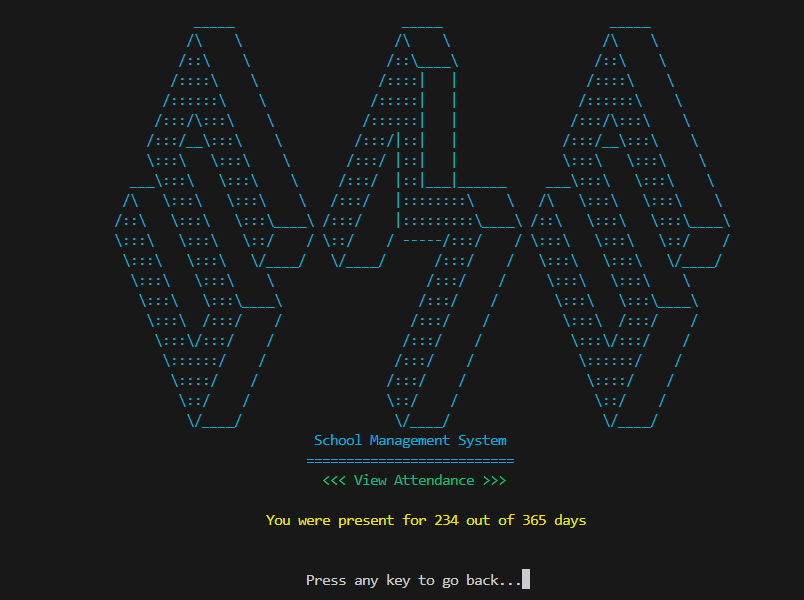
* From choice 1, student can view his/her class



* From choice 2, student can view his/her subjects



* From choice 3, student can view his/her attendance



* From choice 4, student can view his/her fee status



* From choice 5, student can view his/her result card



* From choice 6, student can log out

#  Data Structures

* For storing student’s marks, I have made a single array

**int studentMarks[3]**

* For storing subjects of each class, I have made 5 arrays

**string class1subjects[3],**

**string class2subjects[3],**

**string class3subjects[3],**

**string class4subjects[3],**

**string class5subjects[3]**

#  Function Prototypes

# void printHeader();

# void loginScreen();

# void clearScreen();

# void gotoxy(int, int);

# void resizeConsole();

# void pressTOgoBACK();

# void notExists();

# void alreadyExists();

# void addStudent();

# void removeStudent();

# void viewStudent();

# void updateFees();

# void printSubjects(string);

# void printMarks();

# void updateAttendance();

# void updateMarks();

# void changeClass();

# void viewAdminSubjects();

# void changeAdminPass();

# void changeUserPass();

# void viewStudentClass();

# void viewStudentSubjects();

# void viewStudentAttendance();

# void viewStudentFees();

# void viewStudentResultCard();

# void invalidInput();

# void incorrectCredentials();

# void addNewMarks1();

# void addNewMarks2();

# void addNewMarks3();

# bool loadAdminPass();

# bool loadStudentID();

# bool loadStudentPass();

# void writeStudentName();

# void writeStudentUserName();

# void writeStudentPass();

# void writeStudentClass();

# void removeStudentName();

# void removeStudentID();

# void removestudentPass();

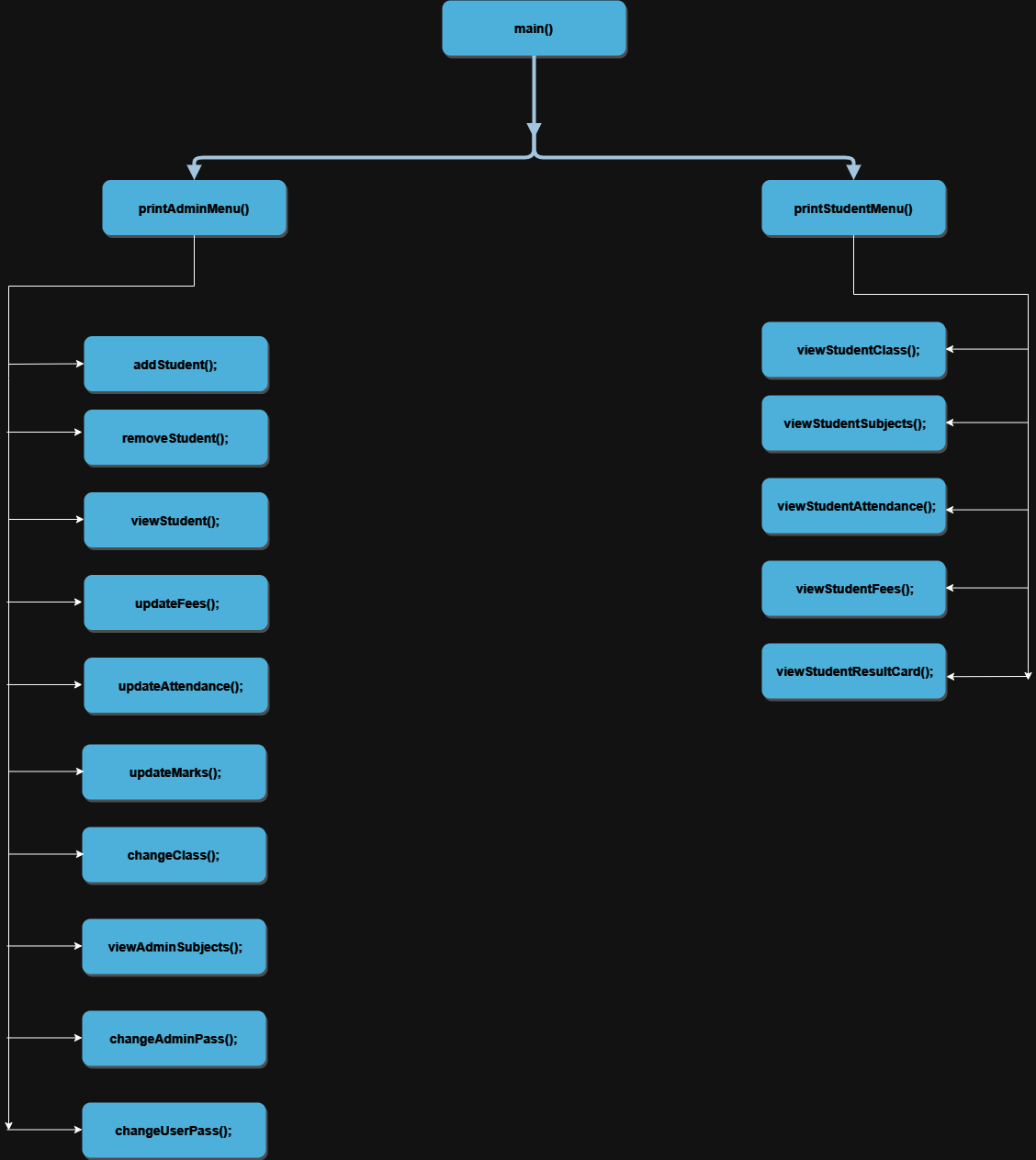
# void removestudentFees();

# void removestudentAttendance();

# void removeStudentClass();

* void validateAttendance(int newAttendance);
* void printAdminMenu();
* void printStudentMenu();

#  Functions Working Flow



#  Complete Code

#include <iostream>

#include <windows.h>

#include <conio.h>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

// Functions

void printHeader(); // prints header

void loginScreen(); // prints login options

void clearScreen(); // clears screen

void gotoxy(int, int); // prints anything at desired grid

void resizeConsole(); // opens cmd in full screen

void pressTOgoBACK(); // prints "go back" statement

void notExists(); // tells that student doesnot exist

void alreadyExists(); // tells that student already exist

void addStudent(); // adds student

void removeStudent(); // removes student

void viewStudent(); // displays all attributes of student

void updateFees(); // updates student's fees

void printSubjects(string); // prints subjects of desired class

void printMarks(); // prints student's marks

void updateAttendance(); // updates student's attendance

void updateMarks(); // updates student's marks

void changeClass(); // changes student's class

void viewAdminSubjects(); // shows all subjects of all classes

void changeAdminPass(); // changes admin password

void changeUserPass(); // changes student's password

void viewStudentClass(); // shows student's class

void viewStudentSubjects(); // shows student's subjects

void viewStudentAttendance(); // shows student's attendance

void viewStudentFees(); // shows student's fee status

void viewStudentResultCard(); // displays student's result card

void invalidInput(); // entered invalid choice

void incorrectCredentials(); // entered invalid credentials

void addNewMarks1();

void addNewMarks2();

void addNewMarks3();

// file loading functions

bool loadAdminPass();

bool loadStudentID();

bool loadStudentPass();

// file writing functions

void writeStudentName();

void writeStudentUserName();

void writeStudentPass();

void writeStudentClass();

// file deleting functions

void removeStudentName();

void removeStudentID();

void removestudentPass();

void removestudentFees();

void removestudentAttendance();

void removeStudentClass();

void validateAttendance(int newAttendance);

string setcolor(unsigned short); // for using colors

int x, y; // coordinates for gotoxy

int studentCount = 0; // number of students

string classOption; // student's class

int studentAttendance = 0; // student's attendance

// marks input

string newMarks1 = "0";

string newMarks2 = "0";

string newMarks3 = "0";

string studentMarks[3] = {"0", "0", "0"}; // student's marks

string adminID = "admin", adminPass = ""; // admin credentials

string CHECKstudentID = " "; // input student's username

string studentID = " ", studentPass = " ", studentName = " ", studentFees = "Nill", newFees = " "; // student's credentials

string usernameSignIn = " ", passwordSignIn = " "; // input username and password

// subjects of each class

string class1subjects[3] = {"English", "Maths", "G.Knowledge"};

string class2subjects[3] = {"Islamiat", "Urdu", "Drawing"};

string class3subjects[3] = {"Social.St", "Communication", "Geography"};

string class4subjects[3] = {"G.Science", "Arabic", "Art & Craft"};

string class5subjects[3] = {"Computer.Sc", "Algebra", "Humanities"};

// Users Main Function Prototypes

void printAdminMenu(); // Options for admin menu

void printStudentMenu(); // Options for student's menu

bool mainLoop = true; // main loop condition

bool admin = true; // admin loop condition

bool user = true; // user loop condition

main()

{

string choice; // input choice

string adminOption; // input admin choice

string studentOption; // input student choice

resizeConsole();

while (mainLoop)

{

clearScreen();

loadAdminPass();

{

if (loadAdminPass() == true)

{

mainLoop = false;

break;

}

}

loadStudentID();

{

if (loadStudentID() == true)

{

mainLoop = false;

break;

}

}

loadStudentPass();

{

if (loadStudentPass() == true)

{

mainLoop = false;

break;

}

}

loginScreen();

getline(cin >> ws, choice);

if (choice == "1" || choice == "2")

{

if (choice == "1")

{

clearScreen();

x = 76, y = 25;

gotoxy(x + 2, y);

setcolor(2);

cout << "<<< Sign in >>>";

gotoxy(x, y + 2);

setcolor(6);

cout << "Enter username: ";

getline(cin >> ws, usernameSignIn);

gotoxy(x, y + 3);

cout << "Enter password: ";

getline(cin >> ws, passwordSignIn);

if (usernameSignIn == adminID && passwordSignIn == adminPass)

{

admin = true; // valid admin credentials

while (admin)

{

clearScreen();

printAdminMenu();

getline(cin >> ws, adminOption);

if (adminOption == "1" || adminOption == "2" || adminOption == "3" || adminOption == "4" || adminOption == "5" || adminOption == "6" || adminOption == "7" || adminOption == "8" || adminOption == "9" || adminOption == "10" || adminOption == "11")

{

if (adminOption == "1")

{

clearScreen();

if (studentCount < 1)

{

x = 73, y = 25;

addStudent();

studentCount++;

}

else

{

x = 73, y = 25;

alreadyExists();

}

}

else if (adminOption == "2")

{

clearScreen();

if (studentCount == 0)

{

x = 73, y = 25;

gotoxy(x + 2, y);

setcolor(2);

cout << "<<< Remove Student >>>";

notExists();

}

else

{

x = 73, y = 25;

removeStudent();

studentCount--;

}

}

else if (adminOption == "3")

{

clearScreen();

if (studentCount == 0)

{

x = 73, y = 25;

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< View Student >>>";

notExists();

}

else

{

x = 45, y = 27;

viewStudent();

}

}

else if (adminOption == "4")

{

clearScreen();

if (studentCount == 0)

{

x = 73, y = 25;

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Update Fees >>>";

notExists();

}

else

{

x = 73, y = 25;

updateFees();

}

}

else if (adminOption == "5")

{

clearScreen();

if (studentCount == 0)

{

x = 73, y = 25;

gotoxy(x + 1, y);

setcolor(2);

cout << "<<< Update Attendance >>>";

notExists();

}

else

{

x = 73, y = 25;

updateAttendance();

}

}

else if (adminOption == "6")

{

clearScreen();

if (studentCount == 0)

{

x = 73, y = 25;

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Update Marks >>>";

notExists();

}

else

{

x = 73, y = 25;

updateMarks();

}

}

else if (adminOption == "7")

{

clearScreen();

if (studentCount == 0)

{

x = 73, y = 25;

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Change Class >>>";

notExists();

}

else

{

x = 73, y = 25;

changeClass();

}

}

else if (adminOption == "8")

{

clearScreen();

viewAdminSubjects();

}

else if (adminOption == "9")

{

clearScreen();

changeAdminPass();

}

else if (adminOption == "10")

{

clearScreen();

if (studentCount == 0)

{

x = 73, y = 25;

gotoxy(x - 2, y);

setcolor(2);

cout << "<<< Change Student Password >>>";

notExists();

}

else

{

clearScreen();

changeUserPass();

}

}

else if (adminOption == "11")

{

admin = false; // admin logout

}

}

else

{

x = 112, y = 30;

invalidInput();

}

}

}

else if (usernameSignIn == studentID && passwordSignIn == studentPass)

{

user = true; // valid student credentials

while (user)

{

clearScreen();

printStudentMenu();

getline(cin >> ws, studentOption);

if (studentOption == "1" || studentOption == "2" || studentOption == "3" || studentOption == "4" || studentOption == "5" || studentOption == "6")

{

if (studentOption == "1")

{

clearScreen();

viewStudentClass();

}

else if (studentOption == "2")

{

clearScreen();

viewStudentSubjects();

}

else if (studentOption == "3")

{

clearScreen();

viewStudentAttendance();

}

else if (studentOption == "4")

{

clearScreen();

viewStudentFees();

}

else if (studentOption == "5")

{

clearScreen();

viewStudentResultCard();

}

else if (studentOption == "6")

{

user = false; // student logout

}

}

else

{

x = 112, y = 30;

invalidInput();

}

}

}

else

{

x = 72;

y = 30;

incorrectCredentials();

}

}

else if (choice == "2")

{

mainLoop = false;

}

}

else

{

x = 76;

y = 31;

invalidInput();

}

}

}

void viewStudentResultCard()

{

x = 73, y = 25;

gotoxy(x + 1, y);

setcolor(2);

cout << "<<< View Result Card >>>";

setcolor(5);

gotoxy(x + 1, y + 2);

cout << "Subjects";

gotoxy(x + 16, y + 2);

cout << "Marks ( out of 10 )";

setcolor(6);

x = 74, y = 29;

printSubjects(classOption);

x = 96, y = 29;

printMarks();

gotoxy(73, 34);

pressTOgoBACK();

}

void viewStudentFees()

{

x = 73, y = 25;

gotoxy(x + 2, y);

setcolor(2);

cout << "<<< View Fee Status >>>";

setcolor(6);

gotoxy(x + 1, y + 2);

cout << "Your fee status is " << studentFees;

gotoxy(73, 30);

pressTOgoBACK();

}

void viewStudentSubjects()

{

x = 73, y = 25;

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< View Subjects >>>";

gotoxy(x + 4, y + 2);

setcolor(5);

cout << "Your subjects are:";

setcolor(6);

x = 82, y = 29;

printSubjects(classOption);

gotoxy(73, 33);

pressTOgoBACK();

}

void viewStudentAttendance()

{

x = 73, y = 25;

gotoxy(x + 2, y);

setcolor(2);

cout << "<<< View Attendance >>>";

setcolor(6);

gotoxy(x - 5, y + 2);

cout << "You were present for " << studentAttendance << " out of 365 days";

gotoxy(73, 30);

pressTOgoBACK();

}

void viewStudentClass()

{

x = 73, y = 25;

gotoxy(x + 4, y);

setcolor(2);

cout << "<<< View Class >>>";

setcolor(6);

gotoxy(x + 1, y + 2);

cout << "Your current class is: " << classOption;

gotoxy(73, 30);

pressTOgoBACK();

}

void changeAdminPass()

{

string checkAdminPass;

x = 73,

y = 25;

gotoxy(x - 2, y);

setcolor(2);

cout << "<<< Change Admin Password >>>";

setcolor(6);

gotoxy(x - 2, y + 2);

cout << "Enter current admin password: ";

getline(cin >> ws, checkAdminPass);

if (checkAdminPass == adminPass)

{

gotoxy(x - 2, y + 3);

cout << "Enter new admin password: ";

getline(cin >> ws, adminPass);

gotoxy(x - 4, y + 5);

setcolor(5);

cout << "Admin password updated successfully!";

gotoxy(x, y + 6);

pressTOgoBACK();

}

else

{

gotoxy(x + 4, y + 4);

setcolor(4);

cout << "Incorrect password!";

gotoxy(x + 1, y + 5);

setcolor(7);

cout << "Press any key to retry...";

getch();

clearScreen();

changeAdminPass();

}

}

void changeUserPass()

{

string checkUserPass;

x = 73,

y = 25;

gotoxy(x - 2, y);

setcolor(2);

cout << "<<< Change student Password >>>";

setcolor(6);

gotoxy(x - 2, y + 2);

cout << "Enter current student password: ";

getline(cin >> ws, checkUserPass);

if (checkUserPass == studentPass)

{

gotoxy(x - 2, y + 3);

cout << "Enter new student password: ";

getline(cin >> ws, studentPass);

gotoxy(x - 5, y + 5);

setcolor(5);

cout << "Student password updated successfully!";

gotoxy(x, y + 6);

pressTOgoBACK();

}

else

{

gotoxy(x + 4, y + 4);

setcolor(4);

cout << "Incorrect password!";

gotoxy(x + 1, y + 5);

setcolor(7);

cout << "Press any key to retry...";

getch();

clearScreen();

changeUserPass();

}

}

void viewAdminSubjects()

{

x = 73, y = 25;

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< View Subjects >>>";

setcolor(5);

gotoxy(x, y + 2);

cout << "Class";

setcolor(6);

gotoxy(x + 2, y + 4);

cout << "1";

gotoxy(x + 2, y + 5);

cout << "2";

gotoxy(x + 2, y + 6);

cout << "3";

gotoxy(x + 2, y + 7);

cout << "4";

gotoxy(x + 2, y + 8);

cout << "5";

gotoxy(x + 19, y + 2);

setcolor(5);

cout << "Subjects";

setcolor(6);

gotoxy(x + 16, y + 4);

cout << class1subjects[0] << " | " << class1subjects[1] << " | " << class1subjects[2];

gotoxy(x + 16, y + 5);

cout << class2subjects[0] << " | " << class2subjects[1] << " | " << class2subjects[2];

gotoxy(x + 16, y + 6);

cout << class3subjects[0] << " | " << class3subjects[1] << " | " << class3subjects[2];

gotoxy(x + 16, y + 7);

cout << class4subjects[0] << " | " << class4subjects[1] << " | " << class4subjects[2];

gotoxy(x + 16, y + 8);

cout << class5subjects[0] << " | " << class5subjects[1] << " | " << class5subjects[2];

gotoxy(73, 35);

pressTOgoBACK();

}

void changeClass()

{

string newClass;

x = 73, y = 25;

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Change Class >>>";

gotoxy(x + 5, y + 2);

setcolor(6);

cout << "Current class: " << classOption;

gotoxy(x + 5, y + 3);

cout << "Updated class: ";

getline(cin >> ws, newClass);

if (newClass == "1" || newClass == "2" || newClass == "3" || newClass == "4" || newClass == "5")

{

if (newClass == "1")

{

classOption = newClass;

}

else if (newClass == "2")

{

classOption = newClass;

}

else if (newClass == "3")

{

classOption = newClass;

}

else if (newClass == "4")

{

classOption = newClass;

}

else if (newClass == "5")

{

classOption = newClass;

}

studentMarks[0] = "0";

studentMarks[1] = "0";

studentMarks[2] = "0";

gotoxy(73, 31);

setcolor(5);

cout << "Class updated successfully!";

gotoxy(73, 32);

pressTOgoBACK();

}

else

{

gotoxy(77, 31);

setcolor(4);

cout << "Enter valid class!";

gotoxy(74, 32);

setcolor(7);

cout << "Press any key to retry... ";

getch();

clearScreen();

changeClass();

}

}

void updateMarks()

{

x = 73, y = 25;

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Update Marks >>>";

gotoxy(x - 11, y + 2);

setcolor(5);

cout << "Subjects";

gotoxy(x + 7, y + 2);

cout << "Current Marks";

gotoxy(x + 25, y + 2);

cout << "New Marks (out of 10)";

setcolor(6);

x = 62, y = 29;

printSubjects(classOption);

x = 86, y = 29;

printMarks();

addNewMarks1();

addNewMarks2();

addNewMarks3();

gotoxy(73, 34);

setcolor(5);

cout << "Marks updated successfully!";

gotoxy(73, 35);

pressTOgoBACK();

}

void addNewMarks1()

{

setcolor(6);

gotoxy(78, 34);

cout << " ";

gotoxy(75, 35);

cout << " ";

x = 107, y = 29;

newMarks1 = "0";

gotoxy(x, y);

getline(cin >> ws, newMarks1);

if (newMarks1 == "0" || newMarks1 == "1" || newMarks1 == "2" || newMarks1 == "3" || newMarks1 == "4" || newMarks1 == "5" || newMarks1 == "6" || newMarks1 == "7" || newMarks1 == "8" || newMarks1 == "9" || newMarks1 == "10")

{

studentMarks[0] = newMarks1;

}

else

{

gotoxy(78, 34);

setcolor(4);

cout << "Enter valid marks!";

gotoxy(75, 35);

setcolor(7);

cout << "Press any key to retry... ";

getch();

gotoxy(107, 29);

cout << " ";

addNewMarks1();

}

}

void addNewMarks2()

{

setcolor(6);

gotoxy(78, 34);

cout << " ";

gotoxy(75, 35);

cout << " ";

x = 107, y = 30;

newMarks2 = "0";

gotoxy(x, y);

getline(cin >> ws, newMarks2);

if (newMarks2 == "0" || newMarks2 == "1" || newMarks2 == "2" || newMarks2 == "3" || newMarks2 == "4" || newMarks2 == "5" || newMarks2 == "6" || newMarks2 == "7" || newMarks2 == "8" || newMarks2 == "9" || newMarks2 == "10")

{

studentMarks[1] = newMarks2;

}

else

{

gotoxy(78, 34);

setcolor(4);

cout << "Enter valid marks!";

gotoxy(75, 35);

setcolor(7);

cout << "Press any key to retry... ";

getch();

gotoxy(107, 30);

cout << " ";

addNewMarks2();

}

}

void addNewMarks3()

{

setcolor(6);

gotoxy(78, 34);

cout << " ";

gotoxy(75, 35);

cout << " ";

x = 107, y = 31;

newMarks3 = "0";

gotoxy(x, y);

getline(cin >> ws, newMarks3);

if (newMarks3 == "0" || newMarks3 == "1" || newMarks3 == "2" || newMarks3 == "3" || newMarks3 == "4" || newMarks3 == "5" || newMarks3 == "6" || newMarks3 == "7" || newMarks3 == "8" || newMarks3 == "9" || newMarks3 == "10")

{

studentMarks[2] = newMarks3;

}

else

{

gotoxy(78, 34);

setcolor(4);

cout << "Enter valid marks!";

gotoxy(75, 35);

setcolor(7);

cout << "Press any key to retry... ";

getch();

gotoxy(107, 31);

cout << " ";

addNewMarks3();

}

}

void validateAttendance(int newAttendance)

{

bool error;

do

{

error = 0;

if (cin.fail())

{

error = 1;

cin.clear();

cin.ignore(123, '\n');

gotoxy(x + 2, y + 5);

setcolor(4);

cout << "Enter valid attendance!";

gotoxy(x + 1, y + 6);

setcolor(7);

cout << "Press any key to retry...";

getch();

clearScreen();

updateAttendance();

}

} while (error);

}

void updateAttendance()

{

int newAttendance = 0;

gotoxy(x + 1, y);

setcolor(2);

cout << "<<< Update Attendance >>>";

gotoxy(x + 2, y + 2);

setcolor(6);

cout << "Current attendance: " << studentAttendance;

gotoxy(x - 6, y + 3);

cout << "Updated attendance (out of 365 days): ";

cin >> newAttendance;

validateAttendance(newAttendance);

if (newAttendance >= 1 && newAttendance <= 365)

{

studentAttendance = newAttendance;

gotoxy(x + 4, y + 5);

setcolor(5);

cout << "Attendance updated!";

gotoxy(x + 1, y + 6);

pressTOgoBACK();

}

else

{

gotoxy(x + 4, y + 5);

setcolor(4);

cout << "Invalid attendance!";

gotoxy(x + 1, y + 6);

setcolor(7);

cout << "Press any key to go back...";

getch();

}

}

void printMarks()

{

gotoxy(x, y);

cout << studentMarks[0];

gotoxy(x, y + 1);

cout << studentMarks[1];

gotoxy(x, y + 2);

cout << studentMarks[2];

}

void printAdminMenu()

{

x = 77, y = 25;

gotoxy(x, y);

setcolor(2);

cout << "<<< Admin Menu >>>";

gotoxy(x, y + 2);

setcolor(6);

cout << "1. Add student";

gotoxy(x, y + 3);

cout << "2. Remove student";

gotoxy(x, y + 4);

cout << "3. View student";

gotoxy(x, y + 5);

cout << "4. Update fee status";

gotoxy(x, y + 6);

cout << "5. Update attendance";

gotoxy(x, y + 7);

cout << "6. Update marks";

gotoxy(x, y + 8);

cout << "7. Change class";

gotoxy(x, y + 9);

cout << "8. View subjects";

gotoxy(x, y + 10);

cout << "9. Change admin password";

gotoxy(x, y + 11);

cout << "10. Change student password";

gotoxy(x, y + 12);

cout << "11. Logout";

gotoxy(x, y + 14);

cout << "Enter choice: ";

}

void printStudentMenu()

{

x = 77, y = 25;

gotoxy(x - 1, y);

setcolor(2);

cout << "<<< Student Menu >>>";

setcolor(6);

gotoxy(x, y + 2);

cout << "1. View class";

gotoxy(x, y + 3);

cout << "2. View subjects";

gotoxy(x, y + 4);

cout << "3. View attendance";

gotoxy(x, y + 5);

cout << "4. View fee status";

gotoxy(x, y + 6);

cout << "5. View result card";

gotoxy(x, y + 7);

cout << "6. Logout";

gotoxy(x, y + 9);

cout << "Enter choice: ";

}

void incorrectCredentials()

{

gotoxy(x - 1, y);

setcolor(4);

cout << "Incorrect Username or Password!";

gotoxy(x + 2, y + 1);

setcolor(7);

cout << "Press any key to retry... ";

getch();

}

void addStudent()

{

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Add Student >>>";

gotoxy(x, y + 2);

setcolor(6);

cout << "Enter Student's name: ";

do

{

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Add Student >>>";

gotoxy(x - 55, y + 2);

setcolor(4);

cout << "(Please enter valid name of less than 9 characters)";

gotoxy(x, y + 2);

setcolor(6);

cout << "Enter Student's name: ";

getline(cin >> ws, studentName);

if (studentName.length() > 8)

{

gotoxy(x + 15, y + 2);

cout << " ";

}

} while (studentName.length() > 8);

gotoxy(x - 55, y + 2);

cout << " ";

gotoxy(x, y + 3);

cout << "Enter Student's username: ";

getline(cin >> ws, CHECKstudentID);

if (CHECKstudentID == adminID)

{

gotoxy(x, y + 7);

setcolor(4);

cout << "Username cannot be 'admin'!";

gotoxy(x + 1, y + 8);

setcolor(7);

cout << "Press any key to retry... ";

getch();

clearScreen();

addStudent();

}

else

{

studentID = CHECKstudentID;

gotoxy(x, y + 4);

cout << "Enter Student's password: ";

getline(cin >> ws, studentPass);

gotoxy(x, y + 5);

cout << "Enter Student's class (1-5): ";

getline(cin >> ws, classOption);

if (classOption == "1" || classOption == "2" || classOption == "3" || classOption == "4" || classOption == "5")

{

gotoxy(x, y + 7);

setcolor(5);

cout << "Student added successfully!";

gotoxy(x, y + 8);

setcolor(7);

cout << "Press any key to continue... ";

getch();

}

else

{

gotoxy(x + 4, y + 7);

setcolor(4);

cout << "Enter valid class!";

gotoxy(x + 1, y + 8);

setcolor(7);

cout << "Press any key to retry... ";

getch();

clearScreen();

addStudent();

}

}

writeStudentName();

writeStudentUserName();

writeStudentPass();

writeStudentClass();

}

void removeStudent()

{

studentName = " ";

studentID = " ";

studentPass = " ";

studentFees = " ";

studentAttendance = 0;

classOption = " ";

studentMarks[0] = "0";

studentMarks[1] = "0";

studentMarks[2] = "0";

gotoxy(x + 2, y);

setcolor(2);

cout << "<<< Remove Student >>>";

gotoxy(x - 1, y + 2);

setcolor(5);

cout << "Student successfully removed!";

gotoxy(x, y + 3);

setcolor(7);

pressTOgoBACK();

removeStudentName();

removeStudentID();

removestudentPass();

// removestudentFees();

// removestudentAttendance();

// removeStudentClass();

// // removeStudentMarks();

}

void alreadyExists()

{

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Add Student >>>";

gotoxy(x + 2, y + 2);

setcolor(4);

cout << "Student already exists!";

gotoxy(x, y + 3);

setcolor(7);

cout << "Press any key to continue... ";

getch();

}

void viewStudent()

{

gotoxy(75, 25);

setcolor(2);

cout << "<<< View Student >>>";

gotoxy(45, 27);

setcolor(5);

cout << R"(Name Class Fee Status Attendance(out of 365) Subjects Marks(out of 10))";

gotoxy(45, 30);

setcolor(6);

cout << studentName;

gotoxy(55, 30);

cout << classOption;

gotoxy(64, 30);

cout << studentFees;

gotoxy(83, 30);

cout << studentAttendance << " days";

x = 102, y = 30;

printSubjects(classOption);

x = 121, y = 30;

printMarks();

gotoxy(73, 35);

pressTOgoBACK();

}

void updateFees()

{

gotoxy(x + 3, y);

setcolor(2);

cout << "<<< Update Fees >>>";

gotoxy(x, y + 2);

setcolor(6);

cout << "Current fee status: " << studentFees;

gotoxy(x - 3, y + 3);

cout << "New fee status (paid/unpaid): ";

getline(cin >> ws, newFees);

if (newFees == "paid" || newFees == "unpaid")

{

if (newFees == "paid")

{

studentFees = newFees;

}

else if (newFees == "unpaid")

{

studentFees = newFees;

}

gotoxy(x + 4, y + 5);

setcolor(5);

cout << "Fee status updated";

gotoxy(x + 1, y + 6);

pressTOgoBACK();

}

else

{

gotoxy(x + 2, y + 5);

setcolor(4);

cout << "Enter valid fee status!";

gotoxy(x + 1, y + 6);

setcolor(7);

cout << "Press any key to retry...";

getch();

clearScreen();

updateFees();

}

}

void pressTOgoBACK()

{

setcolor(7);

cout << "Press any key to go back...";

getch();

}

void printSubjects(string classOption)

{

setcolor(6);

if (classOption == "1")

{

gotoxy(x, y);

cout << class1subjects[0];

gotoxy(x, y + 1);

cout << class1subjects[1];

gotoxy(x, y + 2);

cout << class1subjects[2];

}

else if (classOption == "2")

{

gotoxy(x, y);

cout << class2subjects[0];

gotoxy(x, y + 1);

cout << class2subjects[1];

gotoxy(x, y + 2);

cout << class2subjects[2];

}

else if (classOption == "3")

{

gotoxy(x, y);

cout << class3subjects[0];

gotoxy(x, y + 1);

cout << class3subjects[1];

gotoxy(x, y + 2);

cout << class3subjects[2];

}

else if (classOption == "4")

{

gotoxy(x, y);

cout << class4subjects[0];

gotoxy(x, y + 1);

cout << class4subjects[1];

gotoxy(x, y + 2);

cout << class4subjects[2];

}

else if (classOption == "5")

{

gotoxy(x, y);

cout << class5subjects[0];

gotoxy(x, y + 1);

cout << class5subjects[1];

gotoxy(x, y + 2);

cout << class5subjects[2];

}

}

void notExists()

{

gotoxy(x + 4, y + 2);

setcolor(4);

cout << "No student exists!";

gotoxy(x, y + 3);

setcolor(7);

pressTOgoBACK();

}

void gotoxy(int x, int y)

{

COORD coordinates;

coordinates.X = x;

coordinates.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coordinates);

}

void invalidInput()

{

gotoxy(x + 1, y);

setcolor(4);

cout << "Enter Valid Choice!";

gotoxy(x - 1, y + 1);

setcolor(7);

cout << "Press any key to retry... ";

getch();

}

bool loadAdminPass()

{

fstream loadFile;

loadFile.open("adminPass.txt", ios::in);

if (!loadFile.is\_open())

{

x = 74, y = 25;

gotoxy(x, y);

setcolor(4);

cerr << "Error opening file";

return true;

}

getline(loadFile, adminPass);

loadFile.close();

}

bool loadStudentID()

{

fstream loadFile;

loadFile.open("studentUserName.txt", ios::in);

if (!loadFile.is\_open())

{

x = 74, y = 25;

gotoxy(x, y);

setcolor(4);

cerr << "Error opening file";

return true;

}

getline(loadFile, studentID);

loadFile.close();

}

bool loadStudentPass()

{

fstream loadFile;

loadFile.open("studentPass.txt", ios::in);

if (!loadFile.is\_open())

{

x = 74, y = 25;

gotoxy(x, y);

setcolor(4);

cerr << "Error opening file";

return true;

}

getline(loadFile, studentPass);

loadFile.close();

}

void writeStudentName()

{

fstream writeFile;

writeFile.open("studentName.txt", ios::out);

writeFile << studentName;

writeFile.close();

}

void writeStudentUserName()

{

fstream writeFile;

writeFile.open("studentUserName.txt", ios::out);

writeFile << studentID;

writeFile.close();

}

void writeStudentPass()

{

fstream writeFile;

writeFile.open("studentPass.txt", ios::out);

writeFile << studentPass;

writeFile.close();

}

void writeStudentClass()

{

fstream writeFile;

writeFile.open("studentClass.txt", ios::out);

writeFile << classOption;

writeFile.close();

}

void removeStudentName()

{

fstream removeFile;

removeFile.open("studentName.txt", ios::out);

removeFile << studentName;

removeFile.close();

}

void removeStudentID()

{

fstream removeFile;

removeFile.open("studentUserName.txt", ios::out);

removeFile << studentID;

removeFile.close();

}

void removestudentPass()

{

fstream removeFile;

removeFile.open("studentPass.txt", ios::out);

removeFile << studentID;

removeFile.close();

}

void resizeConsole()

{

COORD coord;

SMALL\_RECT rect;

HWND console = GetConsoleWindow();

coord.X = 1920;

coord.Y = 1080;

rect.Left = 0;

rect.Top = 0;

rect.Right = 1919;

rect.Bottom = 1079;

SetConsoleScreenBufferSize(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

SetConsoleWindowInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), TRUE, &rect);

ShowWindow(console, SW\_MAXIMIZE);

}

void printHeader()

{

setcolor(3);

gotoxy(50, 1);

cout << R"(

\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_

/\ \ /\ \ /\ \

/::\ \ /::\\_\_\_\_\ /::\ \

/::::\ \ /::::| | /::::\ \

/::::::\ \ /:::::| | /::::::\ \

/:::/\:::\ \ /::::::| | /:::/\:::\ \

/:::/\_\_\:::\ \ /:::/|::| | /:::/\_\_\:::\ \

\:::\ \:::\ \ /:::/ |::| | \:::\ \:::\ \

\_\_\_\:::\ \:::\ \ /:::/ |::|\_\_\_|\_\_\_\_\_\_ \_\_\_\:::\ \:::\ \

/\ \:::\ \:::\ \ /:::/ |::::::::\ \ /\ \:::\ \:::\ \

/::\ \:::\ \:::\\_\_\_\_\ /:::/ |:::::::::\\_\_\_\_\ /::\ \:::\ \:::\\_\_\_\_\

\:::\ \:::\ \::/ / \::/ / -----/:::/ / \:::\ \:::\ \::/ /

\:::\ \:::\ \/\_\_\_\_/ \/\_\_\_\_/ /:::/ / \:::\ \:::\ \/\_\_\_\_/

\:::\ \:::\ \ /:::/ / \:::\ \:::\ \

\:::\ \:::\\_\_\_\_\ /:::/ / \:::\ \:::\\_\_\_\_\

\:::\ /:::/ / /:::/ / \:::\ /:::/ /

\:::\/:::/ / /:::/ / \:::\/:::/ /

\::::::/ / /:::/ / \::::::/ /

\::::/ / /:::/ / \::::/ /

\::/ / \::/ / \::/ /

\/\_\_\_\_/ \/\_\_\_\_/ \/\_\_\_\_/

)";

gotoxy(74, 23);

cout << "School Management System";

gotoxy(73, 24);

cout << "==========================";

}

void clearScreen()

{

system("cls");

printHeader();

cout << endl

<< endl;

}

void loginScreen()

{

x = 77, y = 25;

gotoxy(x, y);

setcolor(2);

cout << "<<< Log In Menu >>>";

gotoxy(x + 4, y + 2);

setcolor(6);

cout << "1. Sign In";

gotoxy(x + 4, y + 3);

cout << "2. Exit";

gotoxy(x + 1, y + 4);

cout << "Enter choice : ";

}

string setcolor(unsigned short color)

{

HANDLE hcon = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hcon, color);

return "";

}

* **Weakness in the SMS**

Before starting the project, I misintercepted the requirement of the project. The requirement of the project was to have atleast 2 users so I designed the app just for two users ( admin and student ), meaning that the admin cannot add more than one student at a time. For this purpose, I used variables instead of parallel arrays. As I did not use parallel arrays, I could not implement file handling with comma separated files. I realized this mistake but very late. My TA told me to rebuild the entire app just two days before the deadline but it was nearly impossible. So I had to submit this version. I do hope not make such a mistake next time.

* **Improvements in future**

I shall implement single responsibility principle in functions and opt for local variables and local parallel arrays with comma separated file handling.