**PROJECT TITLE**

**ATTENDENCE MANAGEMENT SYSTEM**

**INTRODUCTION:**

Our project is attendance Management System. This software is based on C++ language (procedural language not the object oriented one), it’s integrated with GUI and various font and background color to make it look more attractive. The software is very user friendly and very simple to use. Its two main environments are Dev C++ and Visual Studio. (Code blocks is rarely preferred)

**OBJECTIVES**

The main objective of our project “attendance management system” is to manage the details of the profile of every student that is important to save their record. It manages all the information about student’s profile. This software will reduce the manual work for management hence reduce their work stress.

**INPUTS:**

* User enters choice according to the options provided (student, admin, exit)
* Menu Selection (with the help of “option number”).
* Input Student’s details such as their name, username, enrollment, father name, mother name, Address and Mobile number.
* Get all students by their roll no.
* Delete Existing Student Record.
* Update Existing Student’s record if wrong information is entered mistakenly.
* Checking list of students registered.
* Checking list of students by their presence count.
* Search Record of Existing Student.
* View Record of a student or an admin.
* Exit Console/Database window.

**OUTPUTS:**

* Student Details such as their name, username, enrollment, father name, mother name, Address and Mobile number. (Individual Student or Whole Class)
* Saving records of the student.
* Marking attendance of the students.
* Count the attendance of the student.
* Student Updated Details (if updated).

**ALGORITHM:**

1. Ask the user to enter choice according to the options provided (student, admin, exit).
2. Then the user has to login by the username and password to run the program.
3. If the credentials are wrong for three times, console terminates automatically.
4. If the credentials are correct then redirect the user to the main screen.
5. In Main Screen (admin login):

* Press 1 to register a student.
* Record will be saved in file of each class separately.
* If same roll number is entered twice, it will prompt that this roll number is already taken by another student.
* Press 2 to Delete Record.
* Chosen record will be deleted if that record exists.
* Press 3 to delete student by their roll no.
* Chosen record will be deleted by their roll no if that record exists.
* Press 4 to Check list of students that are registered.
* Show the list of all students that are registered earlier.
* Press 5 to check the presence count of any student.
* Displays the total presence of the student by just enter his roll no.
* Press 6 to get list of student with their attendance count.
* Displays the list of the students with their attendance count.
* Press 0 to go back to the main screen.

1. In main screen (student login):-

* Press 1 to mark attendance
* Press 2 to count my attendance
* Press 3 to go back

**COMPLETE SOURCE CODE:**

#include <iostream>

#include <string>

#include <fstream>

#include <cstring>

using namespace std;

int adminView();

int studentView();

int studentLogin();

int checkCredentials(string userName, string password);

int getAllStudentsbyRollNo();

int deleteAllStudents();

int deleteStudentbyRollno();

int checkListOfStudentsRegistered();

int checkPresenseCountbyRollno();

int getListOfStudentsWithTheirPresenseCount();

int registerStudent();

int adminLogin();

int registerStudent();

int markMyAttendance(string username);

int countMyAttendance(string username);

int delay();

int delay(){

for (int i = 0; i < 3; i++){

cout << "\n Saving Records ...";

for (int ii = 0; ii < 20000; ii++){

for (int iii = 0; iii < 20000; iii++){

}

}

}

cout << "\n Exiting Now ...";

for (int i = 0; i < 3; i++) {

for (int ii = 0; ii < 20000; ii++) {

for (int iii = 0; iii < 20000; iii++) {

}

}

}

return 0;

}

int adminView(){

int goBack = 0;

while (1){

system("cls");

cout << "\n 1 Register a Student";

cout << "\n 2 Delete All students name registered";

cout << "\n 3 Delete student by rollno";

cout << "\n 4 Check List of Student registered by userame";

cout << "\n 5 Check presense count of any student by Roll No";

cout << "\n 6 Get List of student with their attendance count";

cout << "\n 0. Go Back <- \n";

int choice;

cout << "\n Enter you choice: ";

cin >> choice;

switch (choice){

case 1: registerStudent(); break;

case 2: deleteAllStudents(); break;

case 3: deleteStudentbyRollno(); break;

case 4: checkListOfStudentsRegistered(); break;

case 5: checkPresenseCountbyRollno(); break;

case 6: getListOfStudentsWithTheirPresenseCount(); break;

case 0: goBack = 1; break;

default: cout << "\n Invalid choice. Enter again ";

getchar();

}

if (goBack == 1){

break; //break the loop

}

}

return 0;

}

int studentLogin(){

system("cls");

cout << "\n -------- Student Login ---------";

studentView();

delay();

return 0;

}

int checkCredentials(string userName, string password){

return 0;

}

int adminLogin(){

system("cls");

cout << "\n --------- Admin Login --------";

string username;

string password;

cout << "\n Enter username : ";

cin >> username;

cout << "\n Enter password : ";

cin >> password;

if (username == "sameeh" && password == "anees"){

adminView();

getchar();

delay();

}

else {

cout << "\n Error ! Invalid Credintials..";

cout << "\n Press any key for main menu ";

getchar(); getchar();

}

return 0;

}

int checkStudentCredentials(string username, string password){

// read file line by line & check if username-password.dat exist?

// if it exsist return 1 else 0

ifstream read;

read.open("db.dat");

if (read) {

// The file exists, and is open for input

int recordFound = 0;

string line;

string temp = username + ".dat";

cout << "\n file name is : " << temp;

while (getline(read, line)) {

if (line == temp){

recordFound = 1;

break;

}

}

if (recordFound == 0)

return 0;

else

return 1;

}

else{

return 0;

}

}

int getAllStudentsbyName(){

cout << "\n List of All Students by their Name \n";

cout << "\n Please any key to continue..";

getchar(); getchar();

return 0;

}

int getAllStudentsbyRollNo(){

cout << "\n List of All Students by their Roll No \n";

cout << "\n Please any key to continue..";

getchar(); getchar();

return 0;

}

int deleteStudentbyRollno(){

cout << "\n Delete any Student by their Roll No \n";

cout << "\n Please any key to continue..";

getchar(); getchar();

return 0;

}

int checkPresenseCountbyRollno(){

cout << "\n Check presense count of any Student by Roll No \n";

cout << "\n Please any key to continue..";

getchar(); getchar();

return 0;

}

int checkAllPresenseCountbyRollno(){

cout << "\n Check presense count of All Students by Roll No & Name \n";

cout << "\n Please any key to continue..";

getchar(); getchar();

return 0;

}

int studentView(){

cout << "\n ------- Student Login-------- \n";

string username, password;

cout << "\n Enter username : ";

cin >> username;

cout << "\n Enter password : ";

cin >> password;

if (checkStudentCredentials(username, password) == 0){

cout << "\n Invalid Credentials !!";

cout << "\n Press any key for Main Menu..";

getchar(); getchar();

return 0;

}

int goBack = 0;

while (1){

system("cls");

cout << "\n 1 Mark Attendance fo Today ";

cout << "\n 2 Count my Attendance";

cout << "\n 0. Go Back <- \n";

int choice;

cout << "\n Enter you choice: ";

cin >> choice;

switch (choice){

case 1: markMyAttendance(username); break;

case 2: countMyAttendance(username); break;

case 0: goBack = 1; break;

default: cout << "\n Invalid choice. Enter again ";

getchar();

}

if (goBack == 1){

break; //break the loop

}

}

}

int markMyAttendance(string username){

cout << "\n Mark Attendance for today !!";

cout << "\n Please any key to continue..";

//todo: implement this functionality

getchar(); getchar();

return 0;

}

int countMyAttendance(string username){

cout << "\n Count my attendace for today !!";

cout << "\n Please any key to continue..";

//todo: implment this functionality

getchar(); getchar();

return 0;

}

int deleteAllStudents(){

cout << "\n In delete all students !!";

cout << "\n Please any key to continue..";

//todo: implement this functionality

getchar(); getchar();

return 0;

}

int checkListOfStudentRegistered(){

cout << "\n List of All registered registered !!";

cout << "\n Please any key to continue..";

//todo: implment this functionality

getchar(); getchar();

return 0;

}

int getListOfStudentsWithTheirPresenseCount(){

cout << "\n All Students with their Presense count !!";

cout << "\n Please any key to continue..";

//todo: implement this functionality

getchar(); getchar();

return 0;

}

int checkListOfStudentsRegistered() {

cout << "\n - Check List of Student Registered by Username-- ";

//check if record already exist..

ifstream read;

read.open("db.dat");

if (read){

int recordFound = 0;

string line;

while (getline(read, line)) {

char name[100];

strcpy\_s(name, line.c\_str());

char onlyname[100];

strncpy\_s(onlyname, name, (strlen(name) - 4));

cout << " \n " << onlyname;

}

read.close();

}

else{

cout << "\n No Record found :(";

}

cout << "\n Please any key to continue..";

getchar(); getchar();

return 0;

}

int registerStudent(){

cout << "\n ----- Form to Register Student ---- \n";

string name, username, password, rollno, address, father, mother;

cout << "\n Enter Name : "; cin >> name;

cout << "\n Enter Username : "; cin >> username;

cout << "\n Enter password : "; cin >> password;

cout << "\n Enter rollno : "; cin >> rollno;

getchar();

char add[100];

cout << "\n Enter father : "; cin >> father;

cout << "\n Enter mother : "; cin >> mother;

//check if record already exist..

ifstream read;

read.open("db.dat");

if (read){

int recordFound = 0;

string line;

while (getline(read, line)) {

if (line == username + ".dat"){

recordFound = 1;

break;

}

}

if (recordFound == 1){

cout << "\n Username already Register. Please choose another username ";

getchar(); getchar();

delay();

read.close();

return 0;

}

}

read.close();

ofstream out;

out.open("db.dat", ios::app);

out << username + ".dat" << "\n";

out.close();

ofstream out1;

string temp = username + ".dat";

out1.open(temp.c\_str());

out1 << name << "\n"; out1 << username << "\n"; out1 << password << "\n";

out1 << rollno << "\n"; out1 << add << "\n"; out1 << father << "\n";

out1 << mother << "\n";

out1.close();

cout << "\n Student Registered Successfully !!";

cout << "\n Please any key to continue..";

getchar(); getchar();

return 0;

}

int main(int argc, char\*\* argv) {

while (1){

system("cls");

cout << "\n Attendance Management System \n";

cout << "-------------------------------------\n\n";

cout << "1. Student Login\n";

cout << "2. Admin Login\n";

cout << "0. Exit\n";

int choice;

cout << "\n Enter you choice: ";

cin >> choice;

switch (choice){

case 1: studentLogin(); break;

case 2: adminLogin(); break;

case 0:

while (1){

system("cls");

cout << "\n Are you sure, you want to exit? y | n \n";

char ex;

cin >> ex;

if (ex == 'y' || ex == 'Y')

exit(0);

else if (ex == 'n' || ex == 'N'){

break;

}

else {

cout << "\n Invalid choice !!!";

getchar();

}

}

break;

default: cout << "\n Invalid choice. Enter again ";

getchar();

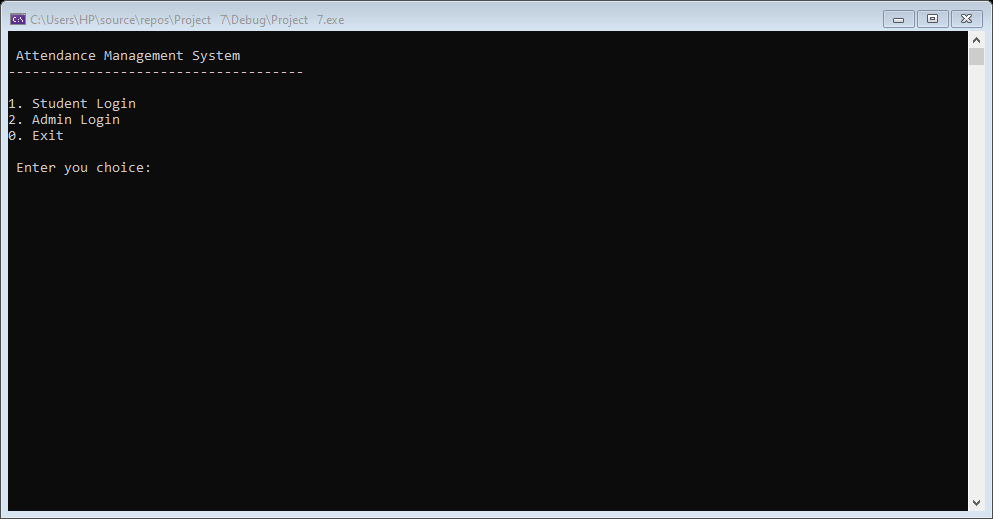
}

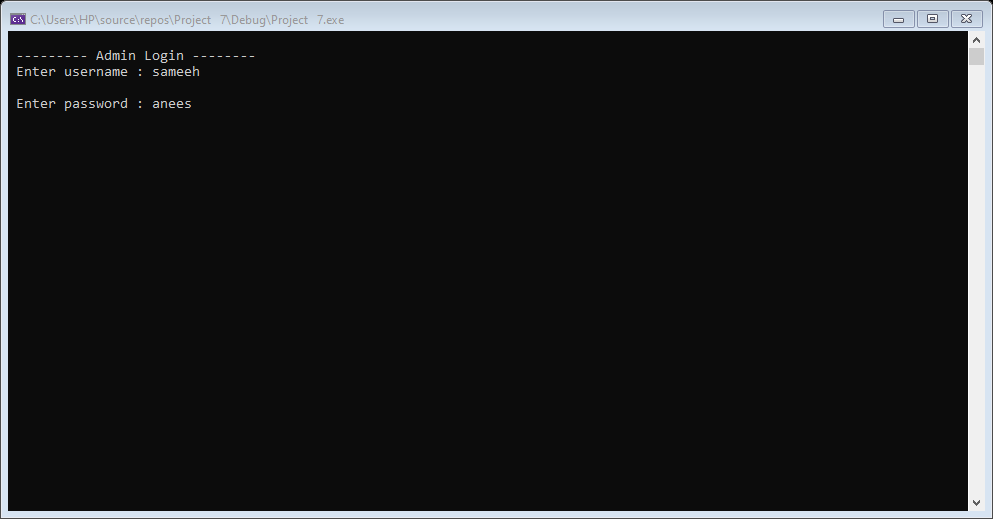
}

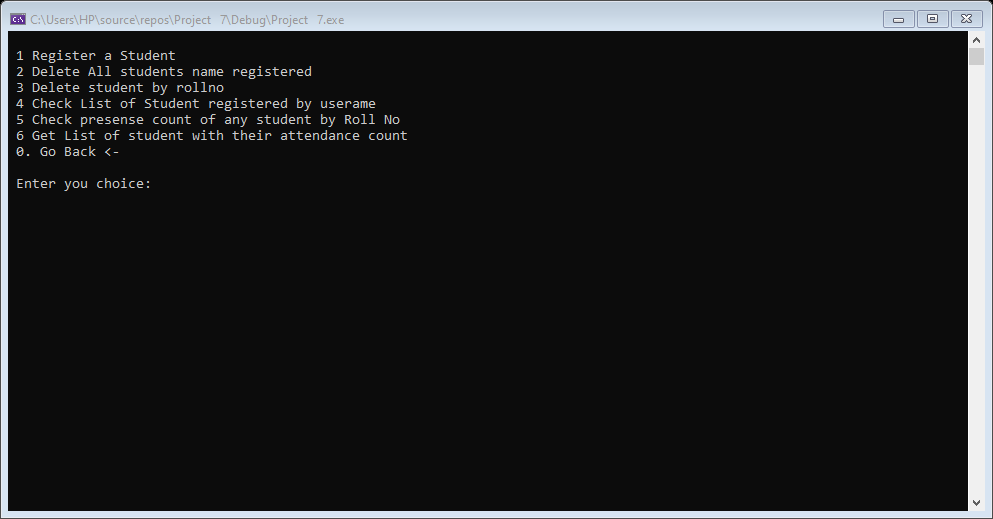
return 0;

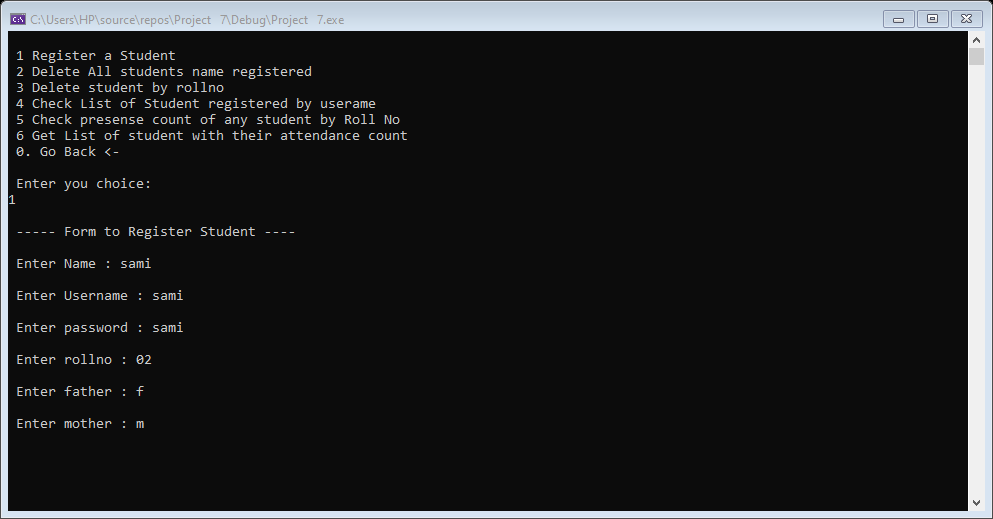
}

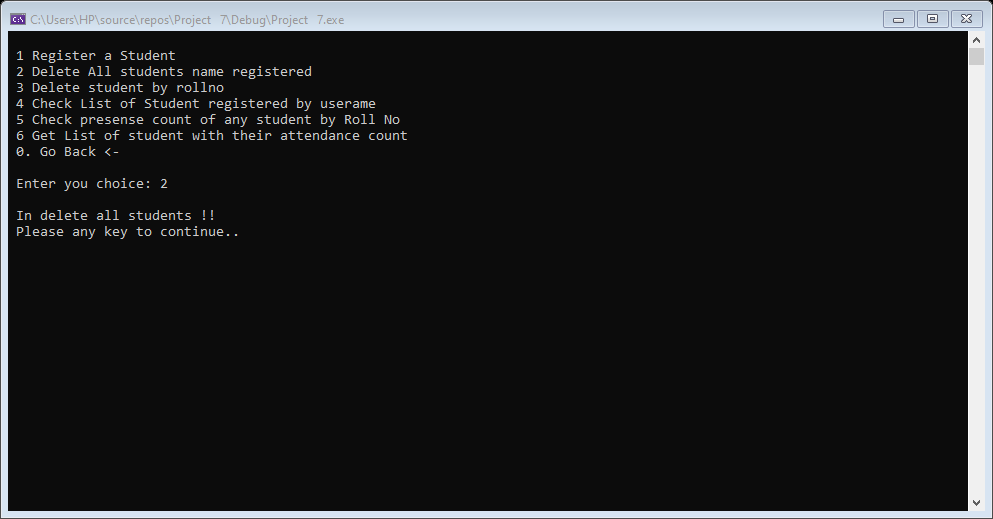
**SCREENSHOTS OF OUTPUT:**

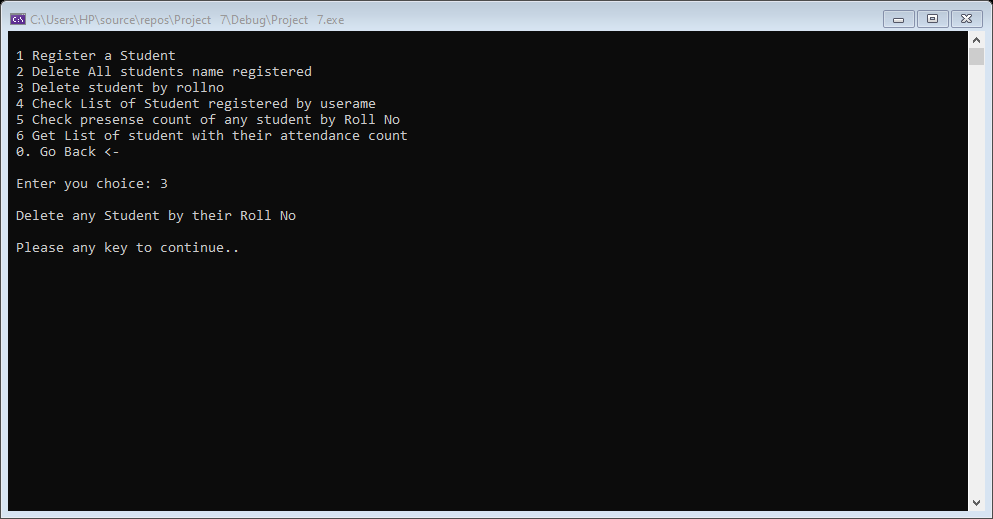


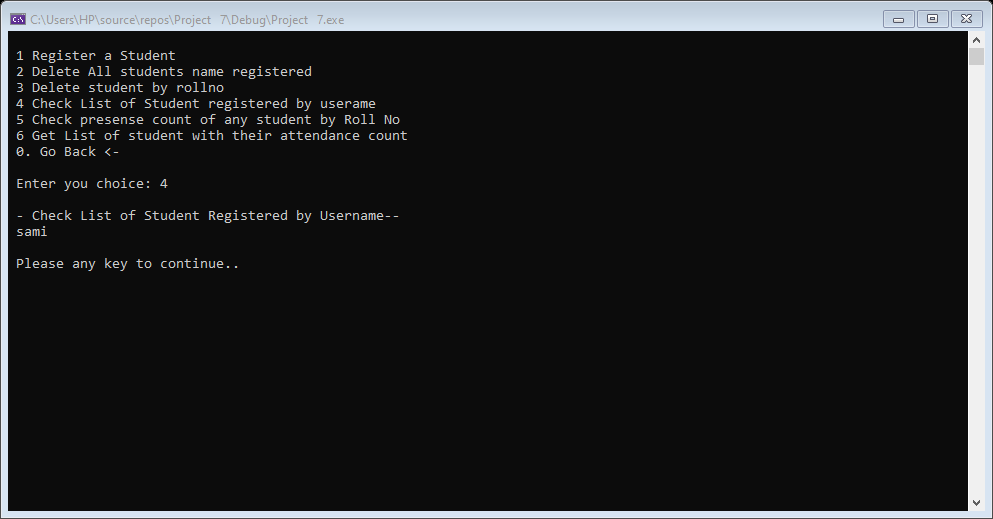


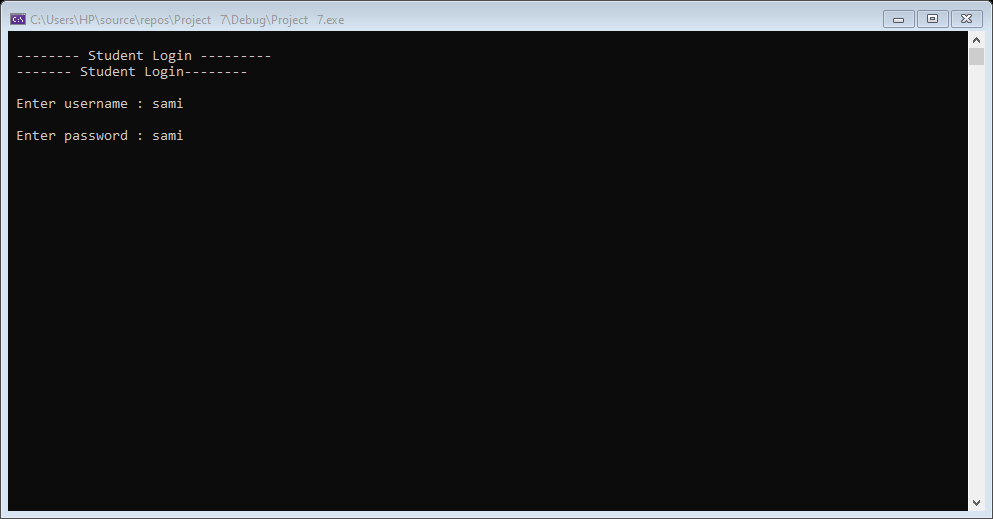


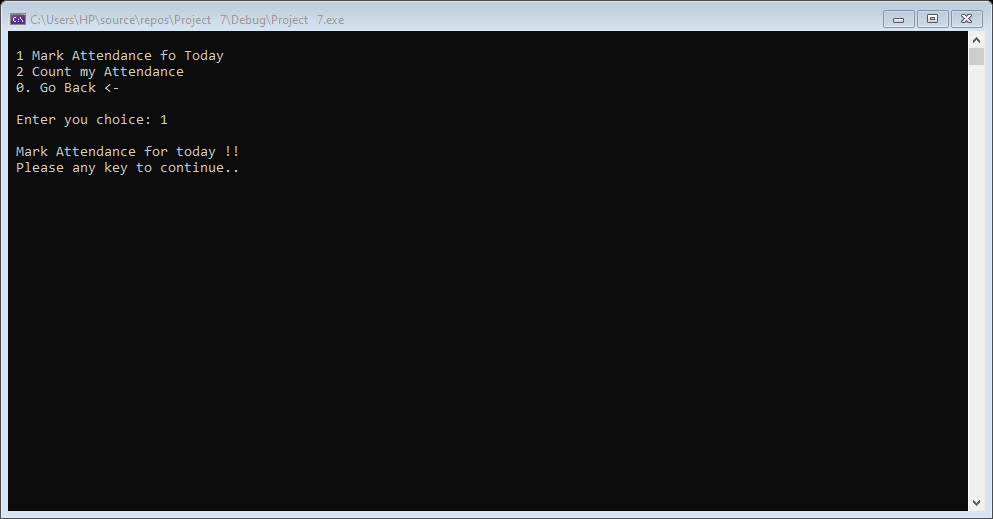


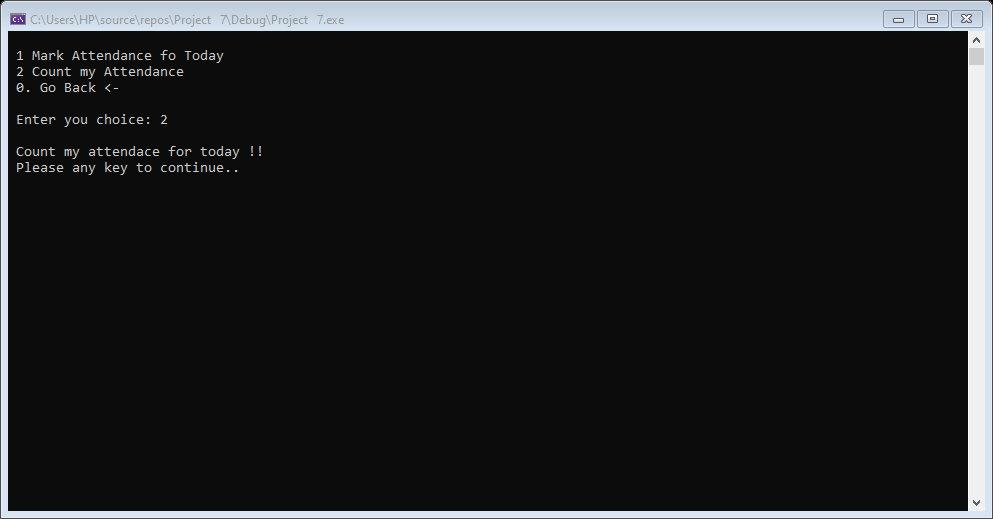


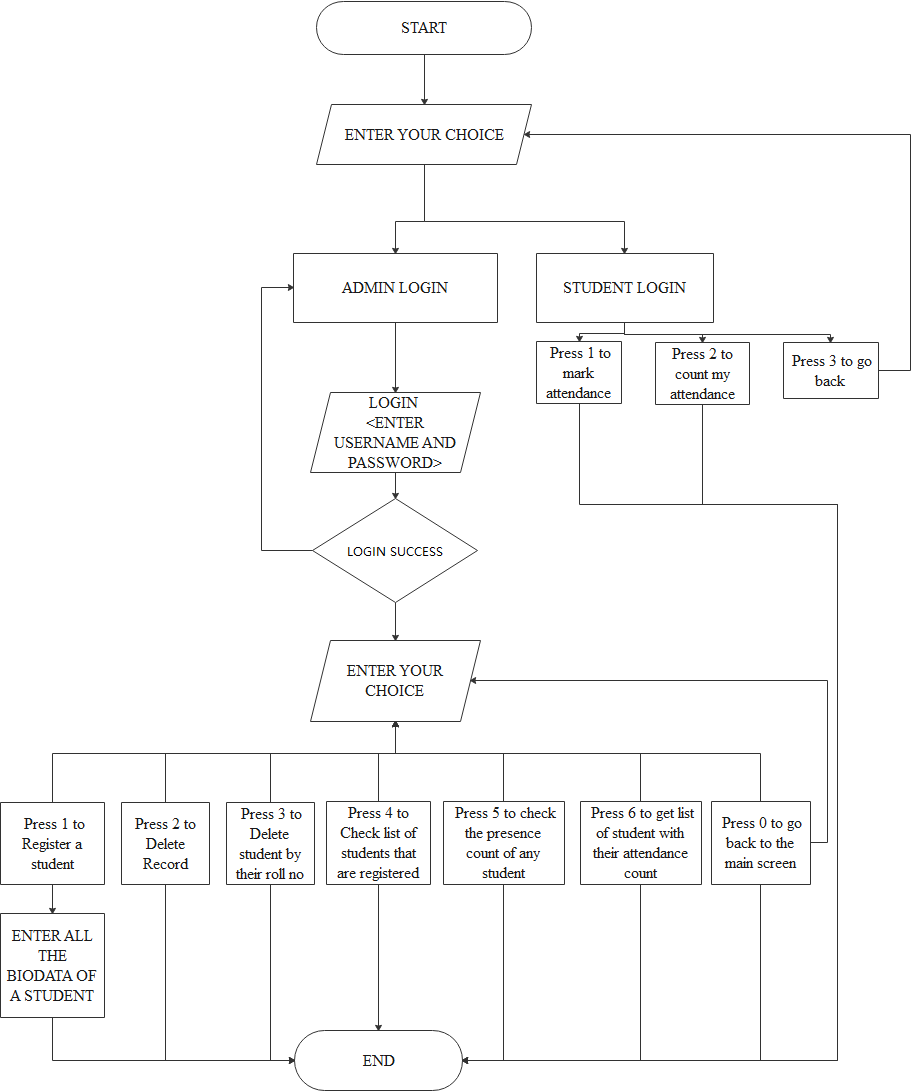












**System Testing**

The aim of the system testing process was to determine all defects in my project. The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not.

Our Project went through some levels of testing:

* **Unit Testing:**

In this level of testing I mainly focused on the small unit of the software design. In this we test an individual unit or group of inter related units by using sample input and observing its corresponding outputs.

* **Integration Testing:**

The main objective of this level testing was to take unit tested components and build a program structure that has been dictated by design in which a group of components are combined to produce output.

**Conclusion:**

The following results have been achieved after completing the system and relate back to the system’s objective.

* This program helps us to make choice according to the options provided (student, admin, exit).
* If the credentials are wrong for three times, console terminates automatically.
* If the credentials are correct then redirect the user to the main screen.

In Main Screen (admin login):

* Press 1 to register a student.
* Record will be saved in file of each class separately.
* If same roll number is entered twice, it will prompt that this roll number is already taken by another student.
* Press 2 to Delete Record.
* Chosen record will be deleted if that record exists.
* Press 3 to delete student by their roll no.
* Chosen record will be deleted by their roll no if that record exists.
* Press 4 to Check list of students that are registered.
* Show the list of all students that are registered earlier.
* Press 5 to check the presence count of any student.
* Displays the total presence of the student by just enter his roll no.
* Press 6 to get list of student with their attendance count.
* Displays the list of the students with their attendance count.
* Press 0 to go back to the main screen.

1. In main screen (student login):-

* Press 1 to mark attendance
* Press 2 to count my attendance
* Press 3 to go back

**References:**

No specific references other than the sample codes in lectures and a sample code provided by **Sir Tarwan Kumar** and **Miss** **Ambreen Akram**

**Link for google drive:**

[**https://drive.google.com/file/d/1YdO4Z4NsuujApBhxmFpmkmWqHpy4Na9f/view?usp=sharing**](https://drive.google.com/file/d/1YdO4Z4NsuujApBhxmFpmkmWqHpy4Na9f/view?usp=sharing)