DEPARTMENT OF COMPUTER SCIENCE & APPLICATION

PANJAB UNIVERSITY

ATTENDANCE MANAGEMENT SYSTEM

C++ MINOR PROJECT

DIKSHITA AGGARWAL (12) & NEHA (27)
MCA-I(MORNING)
SESSION: 2021-24

Preface

As a part of the MCA program and in order to gain practical knowledge in the field, the project report on the "Attendance Management System" has been prepared.

The report includes all the details regarding this minor project. It begins with a brief introduction as to what functions the system performs and eventually delves into a deeper understanding of the project.

The source code and screenshots for the system are included for the perusal of the reviewer to understand the functioning of this system.

The information stated in the report has been gathered from various sources available on the internet and with the help of our mentor.

Names of team members:

Dikshita Aggarwal(12)

Neha (27)

CERTIFICATE

We, students of MCA-I(Morning) hereby declare that this is our original work and it has never been submitted elsewhere.

PROJECT GUIDE	**SUBMITTED BY**
Prof. Sonal Chawla	Dikshita Aggarwal (12)
	Neha (27)

Acknowledgement

It is with great pleasure that we find ourselves penning down these lines to express our sincere gratitude to various people who helped us along in completing this project.

The nurturing and positive environment provided by our teachers at the University encouraged us to work hard and perform well. Thanks to the guidance and mentorship provided by Sonal ma'am that we were able to complete it on time and with utmost sincerity.

We'd also like to thank our parents and peers. They've been immensely helpful and supportive throughout the project.

INDEX

S.No	Topic	Page No	Remarks
1.	Preface	1	
2.	Certificate	2	
3.	Acknowledgement	3	
4.	Project Description	5	
5.	Objective	8	
6.	Future Scope	10	
7.	Hardware/Software Requirements	11	
8.	Flow Chart	12	
9.	Source Code	13	
10.	Screenshots	24	
11.	Conclusion	28	
12.	Bibliography	29	

Project description

Attendance Management System is an attendance maintaining software for educational institutions like schools and colleges. Only the administrative staff or teaching staff can have access to it as it is password protected.

The software is built with C++ Programming language, putting into use features of the Object-Oriented Programming language. The various features that are put into use in this particular project are as follows:

1. **CLASSES**: Classes are the building blocks of C++ that leads to Object Oriented Programming Language. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.

2. **OBJECTS**: An Object is an identifiable entity with some characteristics and behavior. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is

instantiated (i.e., an object is created) memory is allocated. They take up space in memory and have an associated address like a structure or union in C. When a program is executed the objects interact by sending messages to one another.

3. **ENCAPSULATION**: It is the binding together the data and the functions that manipulate them. It is a mechanism of bundling the data, and the functions that use them and **data abstraction** is a mechanism of exposing only the interfaces and hiding the implementation details from the user. C++ supports properties of encapsulation with the help of classes.

4. **INHERITANCE**: The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important features of Object-Oriented Programming. It supports the concept of “reusability”, i.e., when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

5. **ABSTRACTION:** Abstraction means displaying only essential information and hiding the details. Data abstraction refers to

providing only essential information about the data to the outside world, hiding the background details or implementation.

6. **MESSAGE PASSING**: Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and the information to be sent.

Objective

The project is built keeping in mind the amount of manual work involved in marking attendance, adding student details every time a new student enters, determining which students that are eligible to appear in the exams based on their attendance. This can be very difficult and tedious if done manually. So we built this system where all these tasks can be automated.

The following modules are incorporated in the project:

1. **Add New Student**: This module helps in adding new students in the database and automatically adds their roll no in the attendance file to mark attendance against their roll no every time attendance is recorded.

2. **Mark Student Attendance**: In this module, the roll no stored in attendance file are displayed and the user can mark attendance against it. A value of '1' means present and a value of '0' means absent. The attendance file adds the value to the already existing attendance.

3. **Display Student Attendance**: This module simply displays student's roll no, name and attendance in a tabular form. The staff can easily check the number of lectures attended by each student.

4. **Show Eligibility**: At the end of the year or semester, the teacher can enter the number of lectures delivered by her and hence check for students eligible to appear for exams. Only the students with 75% or more attendance are said to be eligible.

FUTURE SCOPE

The Attendance Management System is a simple yet dynamic software that can help maintain records of attendance along with student details. This makes it really beneficial for any educational institute that wants to maintain the attendance.

It can further be modified to store more details and perform more operations to make it more robust. For an instance, the attendance management system can further be enhanced to store and display the result of the student. We can even add a student leave management module to the this system to enhance the functionality of this system.

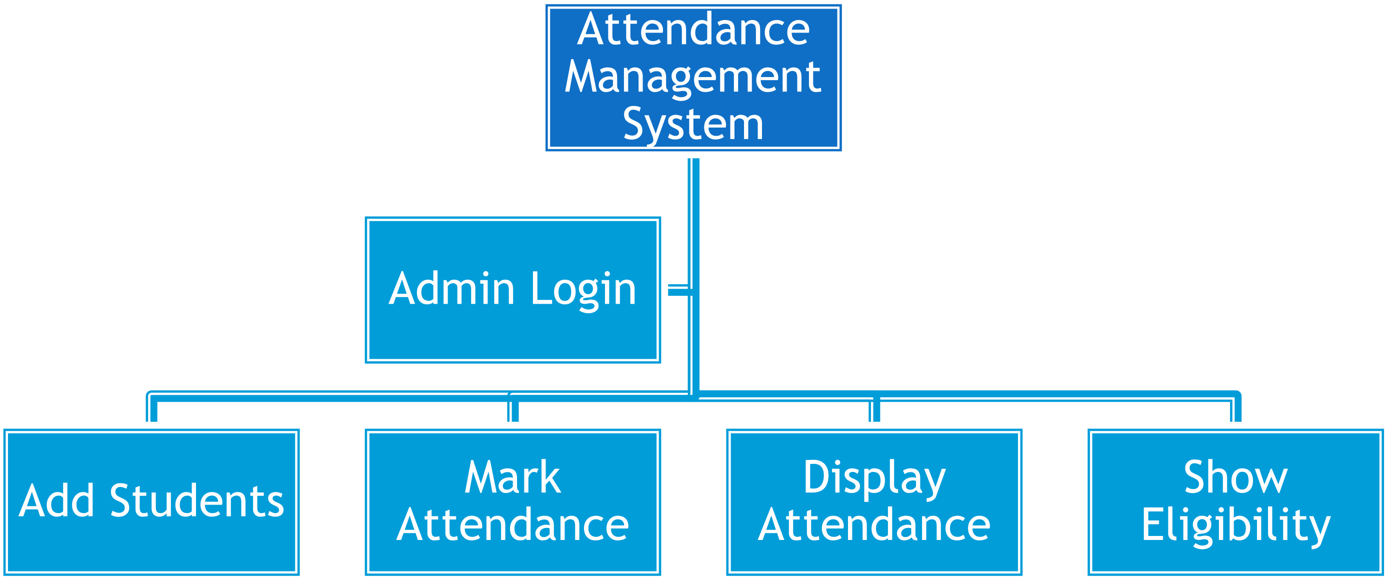
It is very easy to install and run on any operating system installed with a C++ compiler. It'll make an important task of marking the attendance everyday very quick. New records can be stored with ease. Instead of maintaining, a huge amount of paperwork of each student, all the details can be accessed from one place.

Hardware software requirements

- **SOFTWARE REQUIREMENTS**
    - Operating System: DOS based
    - Language: C language
    - Compiler: Dev C++

- **HARDWARE REQUIREMENTS**
    - RAM: 8 GB (baseline)
    - Secondary Memory:
        - 1.81 MB(for project)
        - 69 MB(for compiler)
    - Processor: i3 or i5 (minimum)

FLOW CHART


Attendance Management System
Admin Login
Add Students
Mark Attendance
Display Attendance
Show Eligibility

Source used

```
#include<iostream>

#include<conio.h>

#include <string>

#include<fstream>

#include<iomanip>

using namespace std;


struct student{

int rollno;

string name;

int attend;

}s;

class Student{

public:

addStudents(){

int i,n;

ofstream fout;//write mode

fout.open("Stu_Records.csv",ios::app);

ofstream fattend("attend.csv",ios::app);
```

```
cout<<"\n\t\t Enter number of students to add: ";

cin>>n;

for(i=0;i<n;i++){

cout<<"\n\t\t Enter Roll No: ";

cin>>s.rollno;

cout<<"\t\t Enter Name: ";

cin>>s.name;

fout<<s.rollno;

fout<<"\t"<<s.name<<"\n";

fattend<<s.rollno<<"\t";

fattend<<0<<"\n";

}

fout.close();

fattend.close();

system("cls");

cout<<"\n\n\t\t"<<n<<" Records inserted!!";

}

};

class Attendance:public Student{

public:

mark_attendance(){
```

```
int p;
ifstream fattend("attend.csv");
ofstream ftemp("temp.csv");
if(!fattend){
cout<<"Unable to open attend file";
getch();
return 0;
}
if(!ftemp){
cout<<"Unable to open temp file";
getch();
return 0;
}
cout<<"\n\n Mark attendance against each roll no. \n\t 1 for present\n\t 0 for absent.";
cout<<"\n\nRollNo\tAttendance";
while(fattend>>s.rollno>>s.attend){
cout<<"\n"<<s.rollno<<"\t";
cin>>p;
ftemp<<s.rollno<<"\t";
s.attend+=p;
ftemp<<s.attend<<"\n";
```

```
}
fattend.close();
ftemp.close();
remove("attend.csv");
rename("temp.csv","attend.csv");
cout<"\n\tAttendance Marked!!!!";
}

show_attendance(){
int temp;
ifstream fstud("Stu_Records.csv");
ifstream fattend("attend.csv");
if(!fattend){
cout<<"unable to open file";
getch();
}
cout<<"\n\nDISPLAYING STUDENT ATTENDANCE";
cout<<setw(8)<<"\n\nRollNo\t"<<setw(15)<<"\tName\t"<<setw(10)<<"\tAttendance";
while(fstud>>s.rollno>>s.name){
fattend>>temp;
fattend>>s.attend;
```

```
cout<<"\n"<<setw(8)<<s.rollno;

cout<<"\t"<<setw(15)<<s.name;

cout<<"\t\t"<<setw(10)<<s.attend;

}

fstud.close();

fattend.close();

}

elligible(int l){

double att_per;

ifstream fattend("attend.csv");

if(!fattend){

cout<<"Unable to open file";

getch();

return 0;

}

cout<<"\n\n----LIST OF ELLIGIBLE STUDENTS----";

cout<<"\nRollNo"<<"\tAttendance";

while(fattend>>s.rollno>>s.attend){

att_per=s.attend/l*100;

if(att_per>=75){

cout<<"\n"<<s.rollno<<"\t";
cout<<att_per<<"\t";

cout<<"Elligible";

}

}

fattend.close();

}

};


int main(){

string uname,pass; //user entered username and password

char pass_ch;

bool con; //to continue or not in case of some error

string user="admin", pwd="admin"; //actual username and password

int ch; //choice for menu

char ast;//set asterisk in password

Attendance a;

q:cout<<"\n\n\t";

cout<<"************************************************";

cout<<"\n\t\t ATTENDANCE MANAGEMENT SYSTEM";

cout<<"\n\t************************************************";

cout<<"\n\n\t\t Enter username: ";
```

```
 cin>>uname;
if(uname==user)
{
cout<<"\t\t Enter password: ";
  pass_ch = _getch();
  while(pass_ch != 13) //char 13 is return
{
 pass.push_back(pass_ch);
 cout << '*';
 pass_ch = _getch();
 }
if(pass==pwd){
menu: system("cls");
cout<<"\n\t************************************************";
cout<<"\n\t\t Access Granted!!";
cout<<"\n\t************************************************";
 cout<<"\n\t\t 1. Add Student";
 cout<<"\n\t\t 2. Mark Attendance";
 cout<<"\n\t\t 3. Show Attendance";
 cout<<"\n\t\t 4. Check Eligibility";
 cout<<"\n\t\t 5. Exit";
```

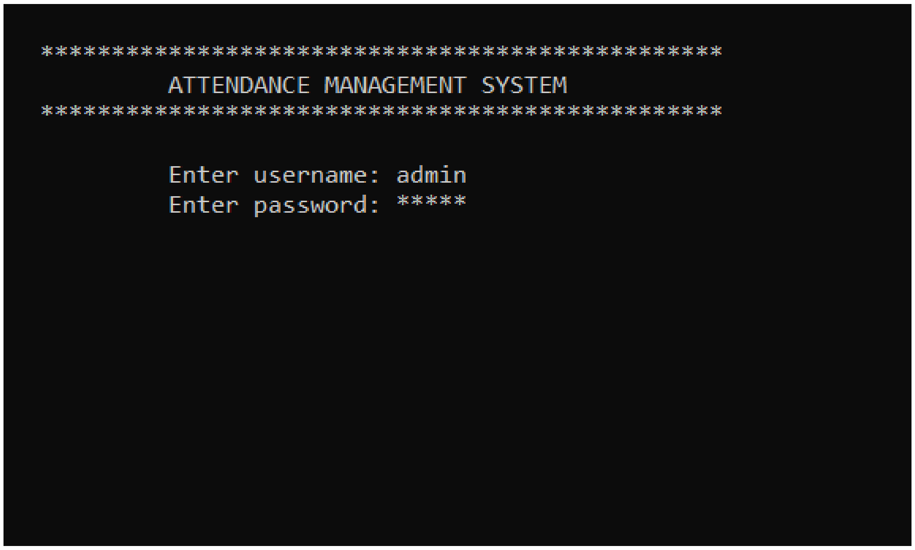
```
cout<<"\n\n\t\t Enter your choice: ";
cin>>ch;

switch(ch){
case 1:
system("cls");
a.addStudents();
goto menu;
case 2:
system("cls");
a.mark_attendance();
goto menu;
case 3:
system("cls");
a.show_attendance();
cout<<"\n\n\t\tPress any key to exit!!";
getch();
goto menu;
case 4:
system("cls");
int lec;
cout<<"\n\n\tEnter total number of lectures delivered: ";
cin>>lec;
a.elligible(lec);
getch();
goto menu;
case 5:
cout<<"\n\t\t Exiting....";
exit(0);
default:
cout<<"\n\t\t Invalid Choice!!";
cout<<"\n\t\t Press 1 to continue\n\t\t Press 0 to exit";
cin>>con;
if(con)
goto menu;
else{
cout<<"\n\t\t Exiting....";
exit(0);
}
}
}
else{
```

```
cout<<"\n\t\t Invalid Password!";
cout<<"\n\t\t Press 1 to try again\n\t\t Press 0 to exit: ";
cin>>con;
pass="";
if(con){
system("cls");
goto q;
}
else{
cout<<"\n\t\t Exiting....";
exit(0);
}
}
}
else{
cout<<"\n\t\t Invalid Username!";
cout<<"\n\t\t Press 1 to try again\n\t\t Press 0 to exit: ";
cin>>con;
if(con){
system("cls");
goto q;
```

```
}
else{
cout<<"\n\t\t Exiting....";
exit(0);
}
}
return 0;
}
```

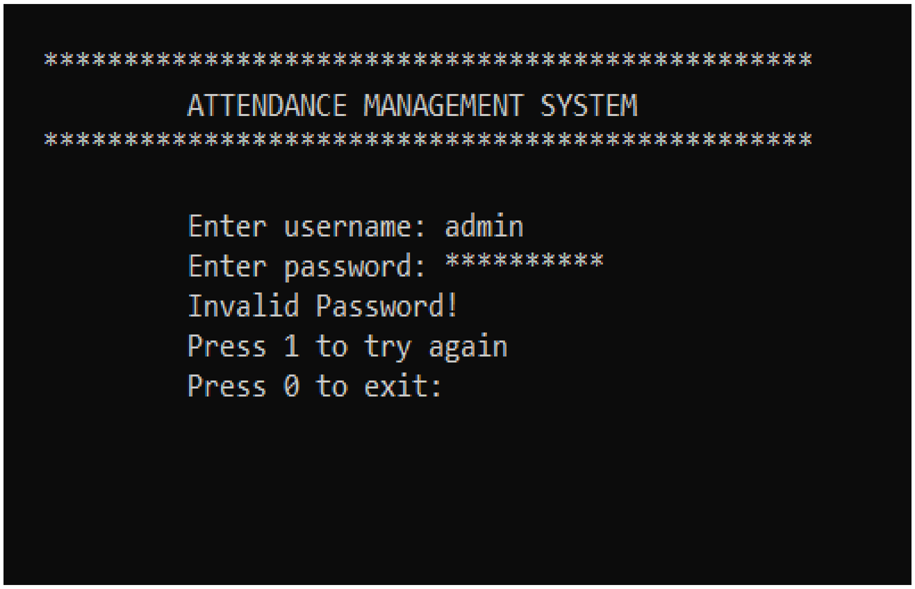
Screenshot

- This window appears as soon as the code is compiled
- Entry is granted only if the user name and password fields are correctly entered.



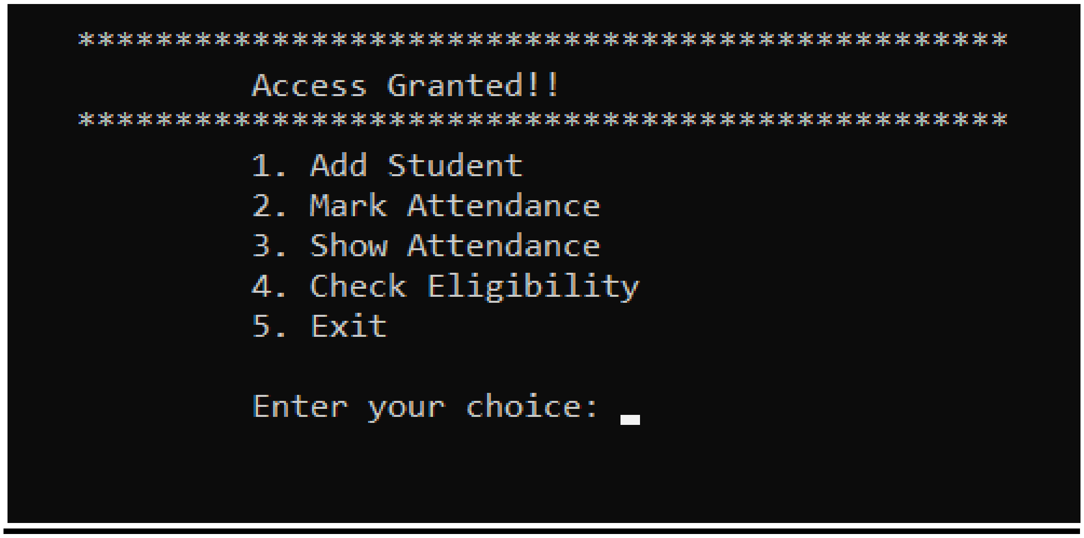
```
**************************************************
         ATTENDANCE MANAGEMENT SYSTEM
**************************************************

         Enter username: admin
         Enter password: *****
```

- Incorrect password



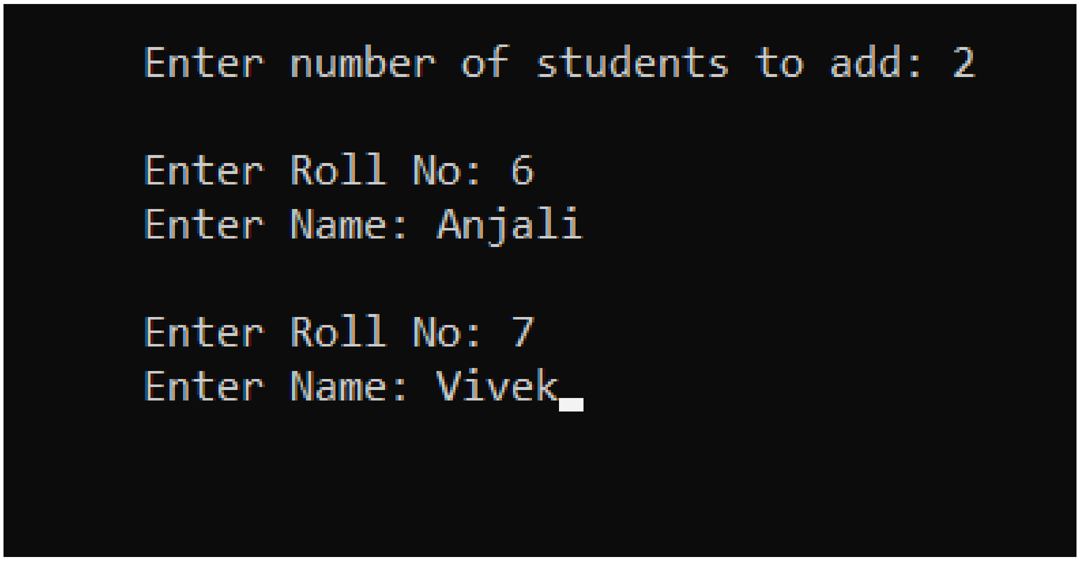
```
**************************************************
         ATTENDANCE MANAGEMENT SYSTEM
**************************************************

         Enter username: admin
         Enter password: **********
         Invalid Password!
         Press 1 to try again
         Press 0 to exit:
```

- Access granted and the list of choices appears



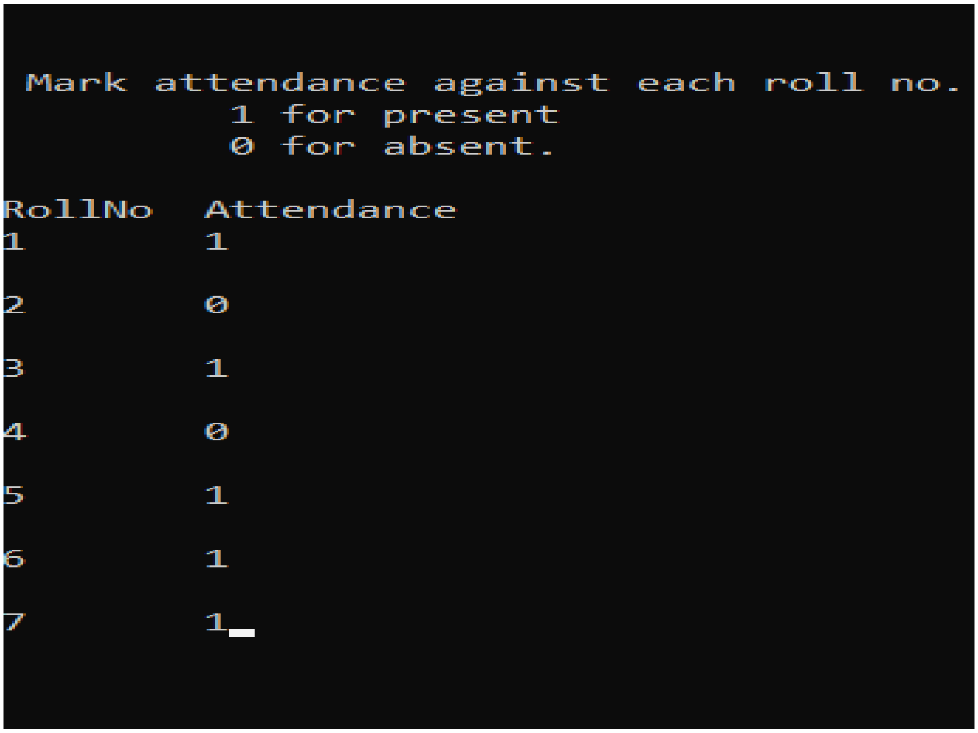
```
************************************************
         Access Granted!!
************************************************
         1. Add Student
         2. Mark Attendance
         3. Show Attendance
         4. Check Eligibility
         5. Exit

         Enter your choice: _
```

- If users choice is 1 (Add Student)



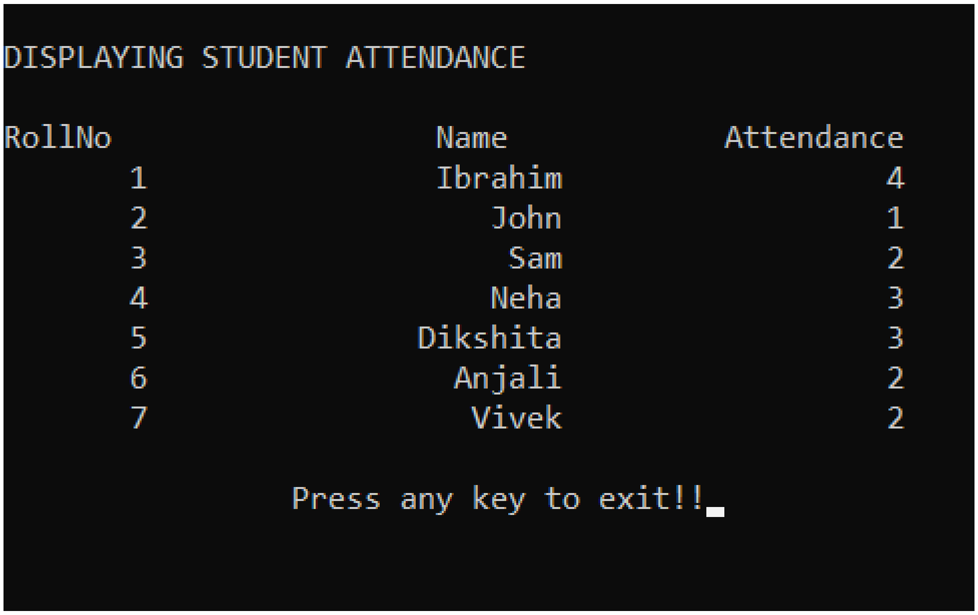
```
Enter number of students to add: 2

Enter Roll No: 6
Enter Name: Anjali

Enter Roll No: 7
Enter Name: Vivek_
```

- If users choice is 2 (Mark Attendance)



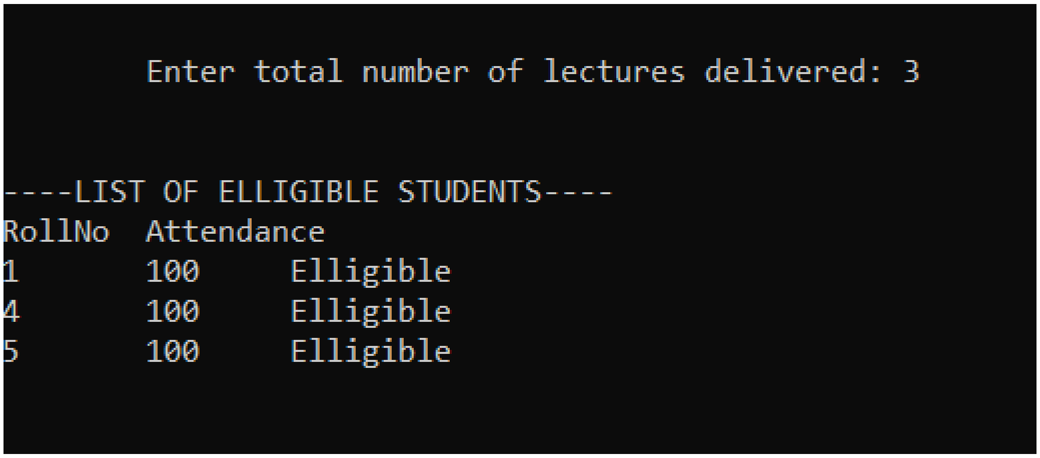
```
 Mark attendance against each roll no.
         1 for present
         0 for absent.

RollNo  Attendance
1       1

2       0

3       1

4       0

5       1

6       1

7       1_
```

- If users choice is 3 (Show Attendance)



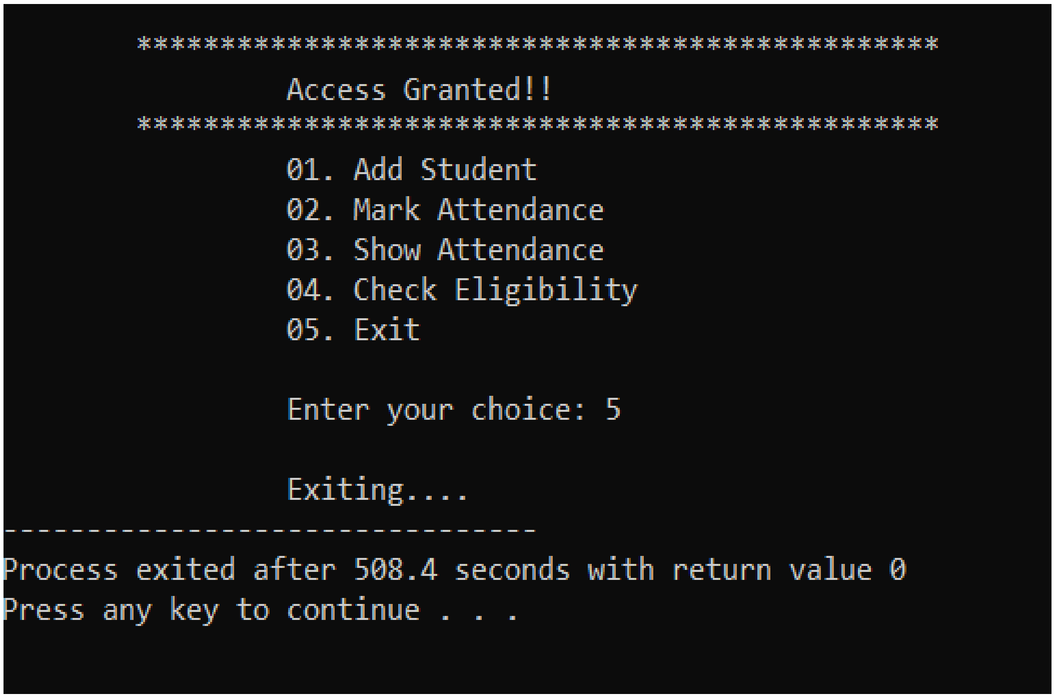
```
DISPLAYING STUDENT ATTENDANCE

RollNo                  Name            Attendance
       1                Ibrahim                  4
       2                   John                  1
       3                    Sam                  2
       4                   Neha                  3
       5               Dikshita                  3
       6                 Anjali                  2
       7                  Vivek                  2

                Press any key to exit!!_
```

- If users choice is 4 (Check Eligibility)



```
        Enter total number of lectures delivered: 3


----LIST OF ELLIGIBLE STUDENTS----
RollNo  Attendance
1       100     Elligible
4       100     Elligible
5       100     Elligible
```

- If users choice is 5 (EXIT)



```
        ************************************************
                 Access Granted!!
        ************************************************
                 01. Add Student
                 02. Mark Attendance
                 03. Show Attendance
                 04. Check Eligibility
                 05. Exit

                 Enter your choice: 5

                 Exiting....
--------------------------------
Process exited after 508.4 seconds with return value 0
Press any key to continue . . .
```

Conclusion

The Attendance Management system is built using the C++ Programming Language and fully utilizing as many features of the Object-Oriented Programming Language as could be in building this system. We've also used File Handling in the system to permanently store the data entered and not just while the program runs.

The system is managed in an efficient way and all the users associated with the system know its advantages. The system solves the problem. It is intended to solve the requirement specification of the system.

It reduces use of paper and saves time to generate accurate results from student's attendance. It provides security by password protecting it, and allowing only admins to enter or modify data.

Bibliography

1. https://www.geeksforgeeks.org/c-plus-plus/?ref=shm
2. https://www.tutorialspoint.com/cplusplus/index.htm
3. https://stackoverflow.com/
4. https://www.youtube.com/
5. https://www.ibm.com/docs/en/i/7.1?topic=programming
6. https://www.quora.com/
7. https://sourceforge.net/
8. https://www.educative.io/