

#### MINOR PROJECT REPORT

**ON**



**FOR**

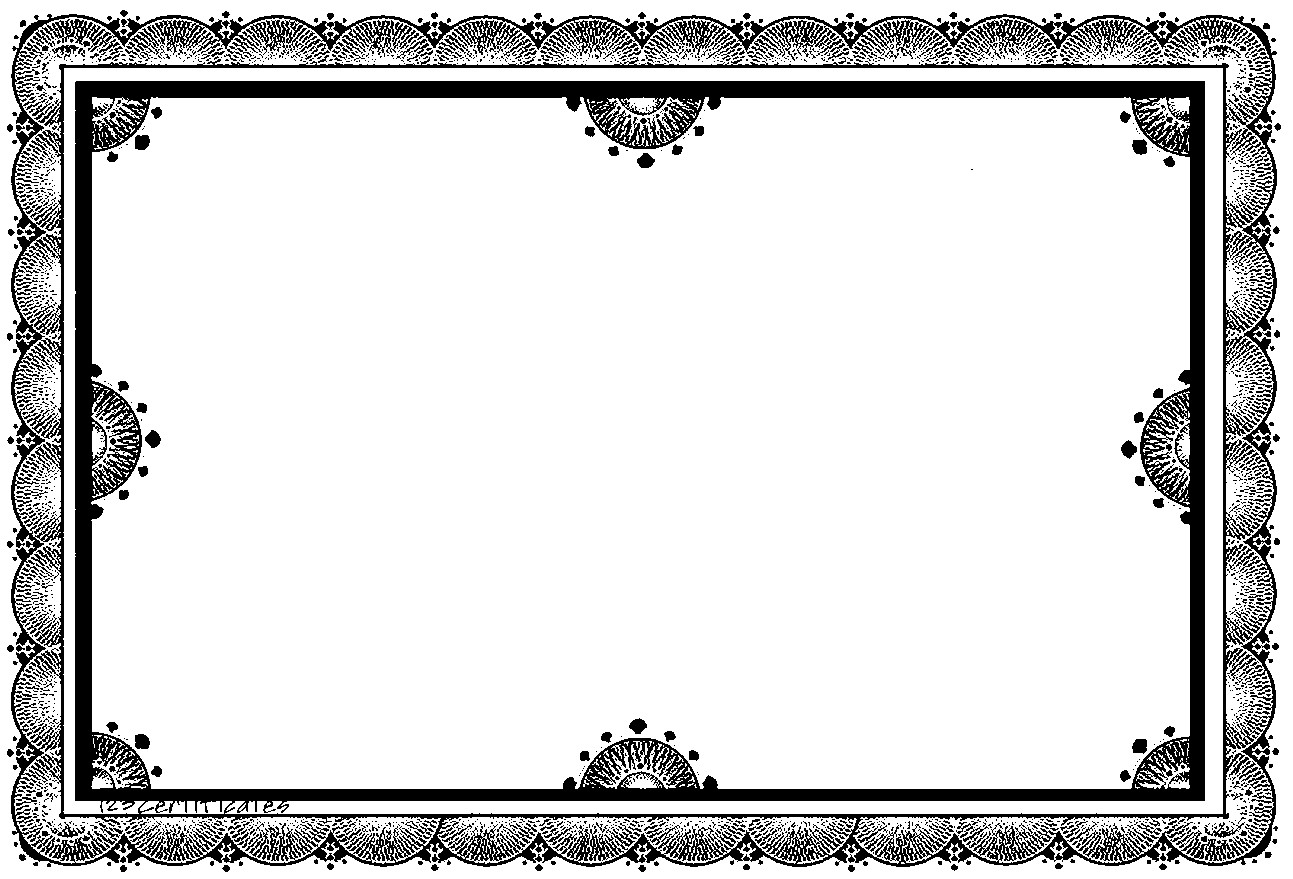
**THE PARTIAL FULFILLMENT OF AISSCE – 2018-2019**



**Submitted By :**

***Yash Gupta***

***12th A***



**2 |** P a g e

**Certificate**

*This is to certify that the project work* ***Cinema Management System*** *is a bonafide record of work done by* ***Yash Gupta*** *under my guidance and supervision.*

***Date : 9th December 9, 2018***

***Enrollment No. :\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***Principal***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**Signature of Internal ExaminerSignature of External Examiner**

*Acknowledgement*



I am extremely grateful to Rev. Bro. Monachan Kurian, our principal, for providing me with the necessary resources and guidelines for undertaking this project under the supervision of Ms. Gargee, Teacher of Department of Computer Science for her able guidance , useful suggestions And Non-Stop Guidance which helped me in completing the project work, on time.

I would also like to thank all the teaching and non-teaching staff of Computer Science department who helped me directly or indirectly in the completion of this project .

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents for their blessings, my friends/classmates for their help and wishes for the successful completion of this project.

***Yash Gupta***

**TABLE OF CONTENTS**



**S.No.**

**Topic**

**Page No.**

1.

2.

3.

4.

5.

6.

7.

8.

9.

Introduction Front-End(C++) Back-End(File) Requirements Source-Code

Output Screenshots Limitations

Future Enhancements Bibliography

5

10

13

15

16

75

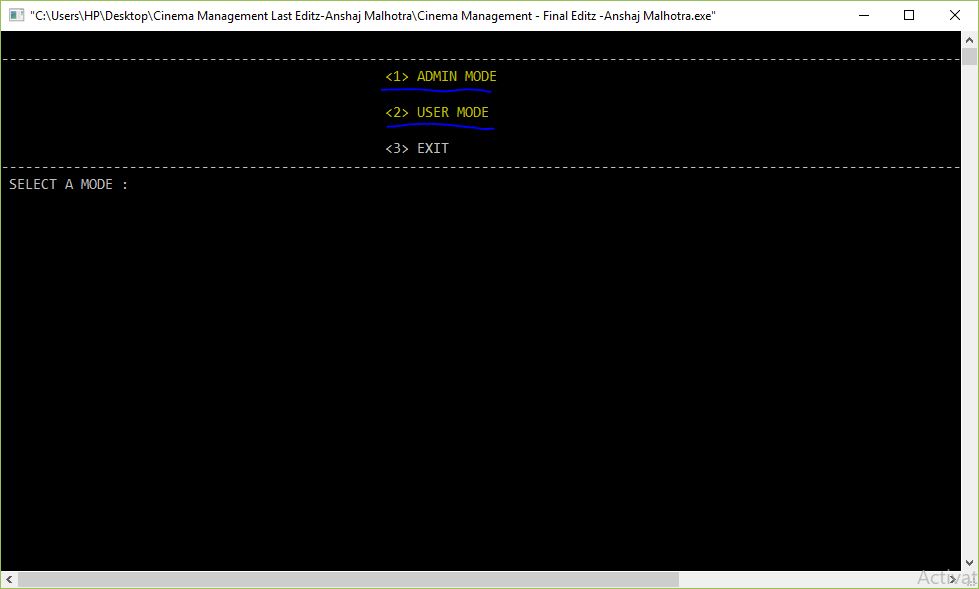
98

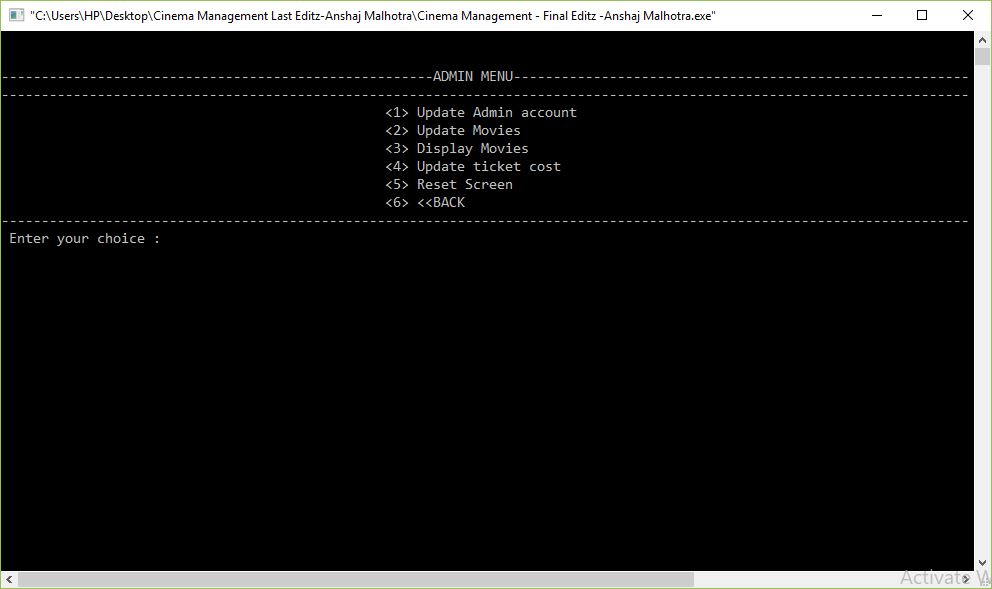
98

100

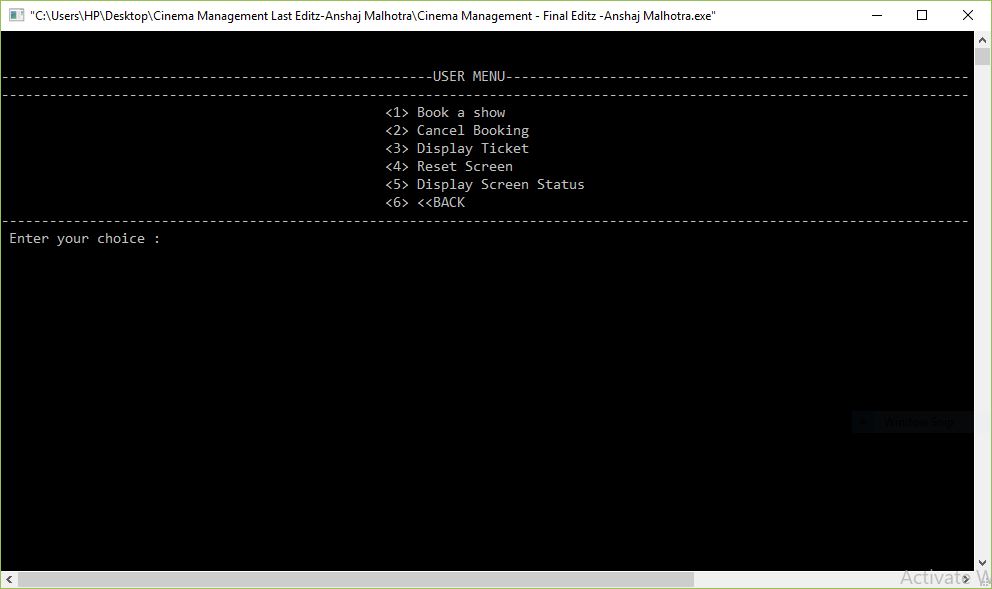
( Cineplex\_Management\_System) is movie booking cum Cineplex management program. It offers two modes-

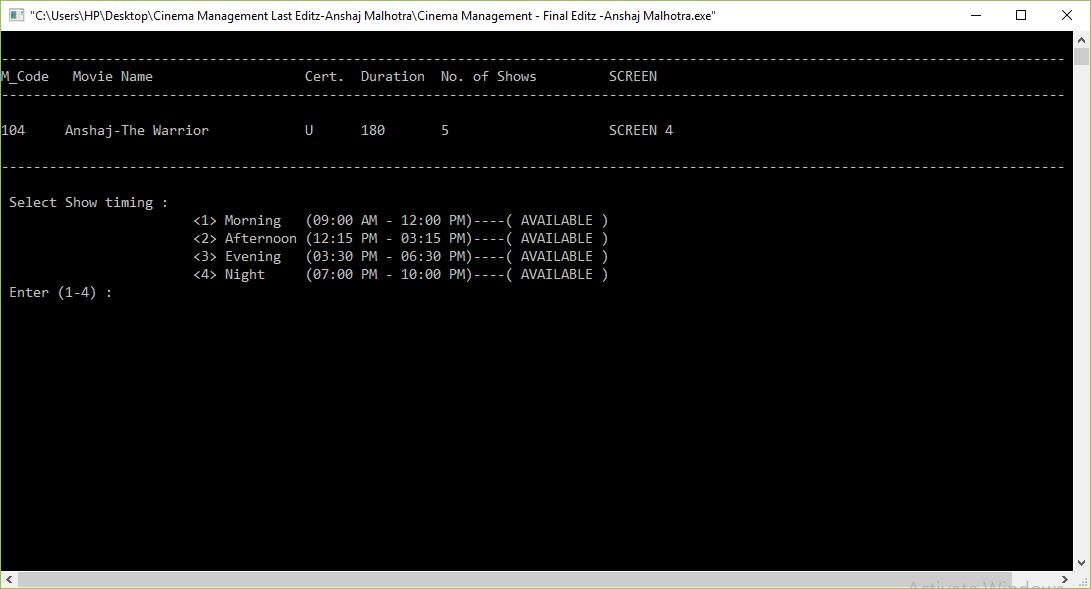
* Admin
  + Admin mode is password protected. The password can be changed any time. Admin has the authority to insert,update and delete a movie.



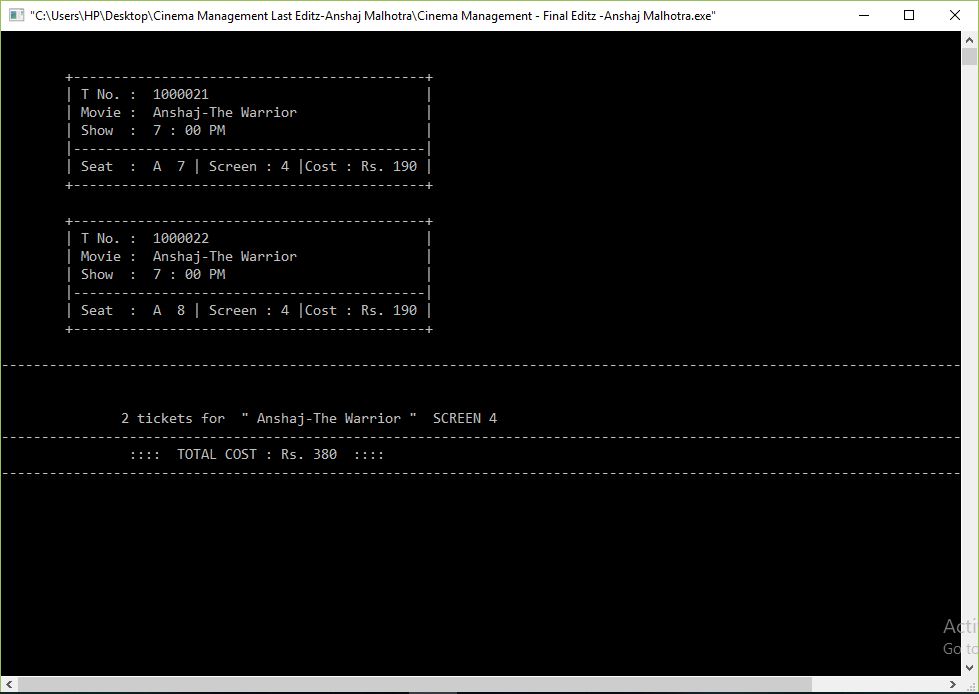


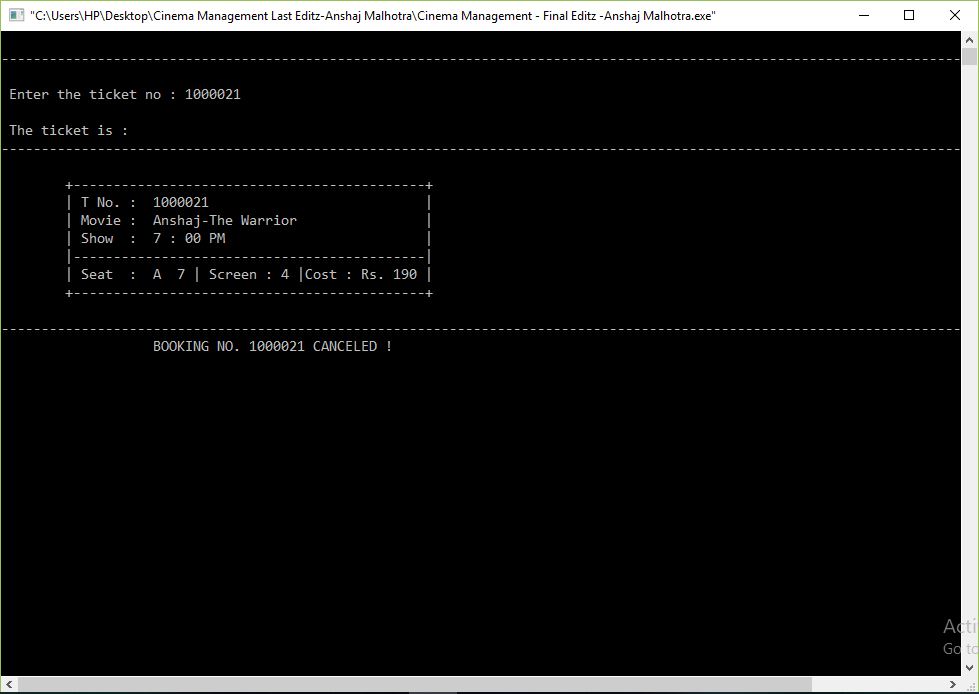
* User Mode
  + User mode facilitates booking a ticket, cancelling a ticket, displaying tickets and screen status in the back drop of an extremely interactive interface. The program offers user friendly functions . The program is completely tested and is flawless. It keeps the account of availability of seats, variation in ticket cost and even compute the total cost of tickets.











Both the modes of the program can be switched to any time. However, the program is pre customized according to a 4 screen 4 slot MEAN setting ( it is a technical abbreviation of Morning, Evening Afternoon and Night shows). In future the screen infrastructure can be customized by inculcating screen updation modules.



\*\*\*

#### Front-End



**The Front-End used in this project is C++.** C++ is a middle-level programming language developed by Bjarne Stroustrup starting in 1979 at Bell Labs. C++ runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming.

C++ is regarded as a **middle-level** language, as it comprises a combination of both high-level and low-level language features.

C++ was developed by Bjarne Stroustrup starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983.

C++ is a superset of C, and that virtually any legal C program is a legal C++ program.

**Note:** A programming language is said to use static typing when type checking is performed during compile-time as opposed to run-time.

##### Object-Oriented Programming

C++ fully supports object-oriented programming, including the four pillars of object-oriented development:

* Encapsulation
* Data hiding
* Inheritance
* Polymorphism

##### Standard Libraries

Standard C++ consists of three important parts:

* The core language giving all the building blocks including variables, data types and literals, etc.



* The C++ Standard Library giving a rich set of functions manipulating files, strings, etc.
* The Standard Template Library (STL) giving a rich set of methods manipulating data structures, etc.

##### Learning C++

The most important thing to do when learning C++ is to focus on concepts and not get lost in language technical details.

The purpose of learning a programming language is to become a better programmer; that is, to become more effective at designing and implementing new systems and at maintaining old ones.

C++ supports a variety of programming styles. You can write in the style of Fortran, C, Smalltalk, etc., in any language. Each style can achieve its aims effectively while maintaining runtime and space efficiency.

##### Use of C++

C++ is used by hundreds of thousands of programmers in essentially every application domain.

C++ is being highly used to write device drivers and other softwares that rely on direct manipulation of hardware under realtime constraints.

C++ is widely used for teaching and research because it is clean enough for successful teaching of basic concepts.

Anyone who has used either an Apple Macintosh or a PC running Windows has indirectly used C++ because the primary user interfaces of these systems are written in C++.

##### Local Environment Setup

If you are still willing to set up your environment for C++, you need following two softwares available on your computer.

##### Text Editor:



This will be used to type your program. Examples of few editors include Windows Notepad, OS Edit command, Brief, Epsilon, EMACS, and vim or vi.

Name and version of text editor can vary on different operating systems. For example, Notepad will be used on Windows and vim or vi can be used on windows as well as Linux, or UNIX.

The files you create with your editor are called source files and for C++ they typically are named with the extension .cpp, .cp, or .c.

Before starting your programming, make sure you have one text editor in place and you have enough experience to type your C++ program.

##### C++ Compiler:

This is actual C++ compiler, which will be used to compile your source code into final executable program.

Most C++ compilers don't care what extension you give your source code, but if you don't specify otherwise, many will use .cpp by default

Most frequently used and free available compiler is GNU C/C++ compiler, otherwise you can have compilers either from HP or Solaris if you have respective Operating Systems.

When we consider a C++ program, it can be defined as a collection of objects that communicate via invoking each other's methods. Let us now briefly look into what do class, object, methods and Instance variables mean.

* **Object -** Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors - wagging, barking, eating. An object is an instance of a class.
* **Class -** A class can be defined as a template/blueprint that describes the behaviors/states that object of its type support.
* **Methods -** A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.
* **Instance Variables -** Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.



##### Object – Oriented Programming using C++

The main purpose of C++ programming is to add object orientation to the C programming language and classes are the central feature of C++ that supports object-oriented programming and are often called user-defined types.

A class is used to specify the form of an object and it combines data representation and methods for manipulating that data into one neat package. The data and functions within a class are called members of the class.

##### C++ Class Definitions:

When you define a class, you define a blueprint for a data type. This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

One of the most important concepts in object-oriented programming is that of inheritance. Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and fast implementation time.

When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base**class, and the new class is referred to as the **derived** class.

#### Back-End : (Files)

A **data file** is an computer file which stores data to be used by a

computer [application](https://en.wikipedia.org/wiki/Application_software) or [system](https://en.wikipedia.org/wiki/System_software). It generally does *not* refer to files that contain instructions or code to be executed (typically called program files), or to files

which define the operation or structure of an application or system (which include configuration files, directory files, etc.); but specifically to information used as input, or written as output by some other software program. This is especially helpful when debugging a program.



Most computer programs work with [files.](https://en.wikipedia.org/wiki/Computer_file) This is because files help in storing information permanently. [Database](https://en.wikipedia.org/wiki/Database) programs create files of

information. [Compilers](https://en.wikipedia.org/wiki/Compiler) read source files and generate executable files. A file itself is a bunch of [bytes](https://en.wikipedia.org/wiki/Byte) stored on some storage device like [tape](https://en.wikipedia.org/wiki/Magnetic_tape_data_storage), [magnetic](https://en.wikipedia.org/wiki/Magnetic_disk) [disk,](https://en.wikipedia.org/wiki/Magnetic_disk) [Optical disk](https://en.wikipedia.org/wiki/Optical_disk) etc. The **data files** are the files that store data pertaining to a specific application, for later use.

*Storage types of Data file*

The data files can be stored in two ways:

1. Text files.
2. Binary files.

A **text file** (also called ASCII files) stores information in [ASCII](https://en.wikipedia.org/wiki/ASCII) characters. A text file contains visible characters. One can see the contents of file on the monitor or edit it using any of the text editors. In text files, each line of text is terminated, (delimited) with a special character known as [EOL (End of](https://en.wikipedia.org/wiki/Newline)

[Line)](https://en.wikipedia.org/wiki/Newline) character. In text files some internal translations take place when this EOL character is read or written.

##### Examples of text files

* A file containing a C++ program

A **binary file** is a file that contains information in the same format in which the information is held in memory i.e. in the binary form. In binary file, there is no delimiter for a line. Also no translations occur in binary files. As a result,binary files are faster and easier for a program to read and write than the text files. As long as the file doesn't need to be read or need to be ported to a different type of system, binary files are the best way to store program information.

##### Example of binary files

* A compiled C++ class file

Files are a means to store data in a storage device. C++ file handling provides a mechanism to store output of a program in a file and read from a file on the disk. So far, we have been using **<iostream>** header file which provide

functions **cin** and **cout** to take input from console and write output to a console respectively. Now, we introduce one more header file **<fstream>**which provides data types or classes ( **ifstream** ,**ofstream** , **fstream** ) to read from a file and write to a file.



## REQUIREMENTS

#### HARDWARE REQUIRED

 **Printer, to print the required documents of the project**

 **Compact Drive**

 **Processor : Pentium III**

 **Ram : 64 MB**

 **Harddisk : 20 Gb.**

**SOFTWARE REQUIRED**

 **Operating system : Windows XP**

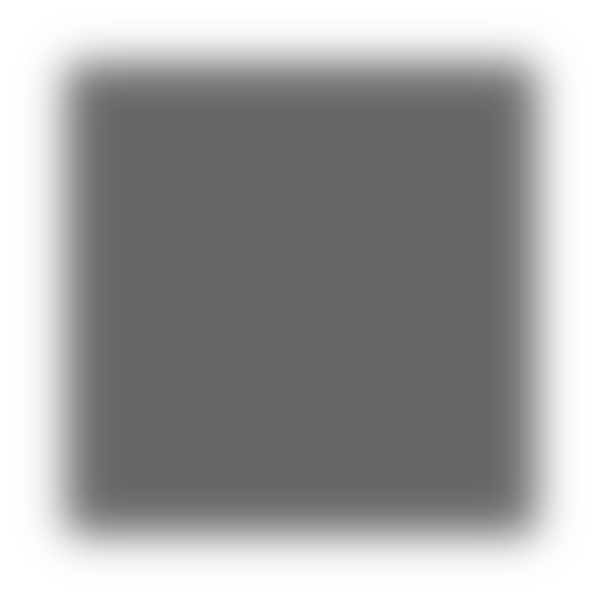
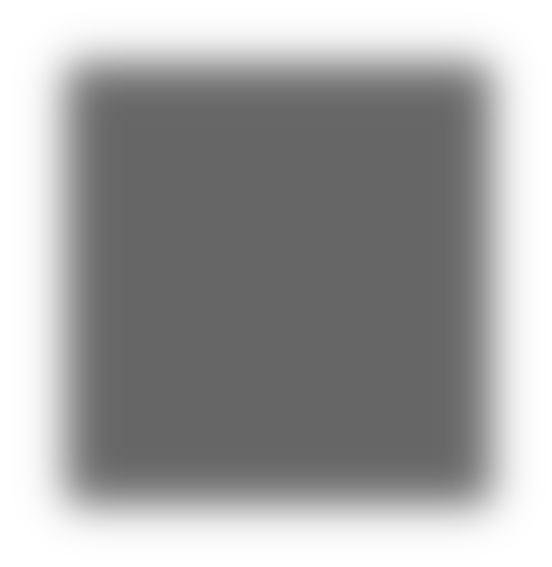
 **Turbo C++, for execution of program and**

 **Ms word, for presentation of output.**

**SOURCE**



**CODE**



/\*\* Description



Cinepolis\_System is movie ticket booking cum Cineplex management program. It offers two modes-

* 1. Admin

Admin mode is password protected. The password can be changed any time. Admin has the authority to insert,update and delete a movie.

* 1. User Mode

User mode facilitates booking a ticket, cancelling a ticket, displaying tickets and screen status in the back drop of an extremely interactive interface. The program offers user friendly functions . The program is completely tested and is flawless. It keeps the account of availability of seats, variation in ticket cost and even compute the total cost of tickets.

\*/

//===============================================

============

/\*---------------------------------Header Files used------------------------------

-------------------\*/

#include <iostream>

#include<string.h>

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<ctype.h>

#include<fstream>

#include<process.h>

#include<iomanip>

#define MAX 100

#define sk 'X'



using namespace std;

/\*-----------------------------------MISCELLANEOUS FUNCTIONS & CLASSES-----------------------------------------------------\*/

void show\_time\_slot(int n) //function to view the 4 time slots of screen

{

if(n==1)

cout<<" 9 : 00 AM";

if(n==2)

cout<<"12 : 15 PM";

if(n==3)

cout<<" 3 : 30 PM";

if(n==4)

cout<<" 7 : 00 PM";

}

int convert\_odd(int n) //function to convert the number to nearest odd natural number

{

if(n>24) n=n-12;

n=n/2; return n;

}

char \*disp\_avail(int n) //function to check if the screen or slot is available

{

if(n)

return "NOT AVAILABLE";

else

return "AVAILABLE";



ifstream ufil;

class seat //class representing seats in every screen

{

char r; //represents row number of seat

int c; //represents column number of the seat float price; //represents the price for seat category

char category[MAX]; //represents category of the seat like gold, premium or silver

void assign\_category() //assigns category to an object of seat

{

if(r<='D'&&r>='A') strcpy(category,"PREMIUM");

else if(r>='E'&&r<='H') strcpy(category,"GOLD");

else

strcpy(category,"SILVER");

}

void set\_price() //assigns price to an object of seat

{

if(strcmpi(category,"PREMIUM")==0) //checking the category and assigning price accordingly

price=180.0;

else if(strcmpi(category,"GOLD")==0) price=140.0;

else price=100.0;

}

public :

seat() //default constructor which sets the default values

{

r=0; c=0;

strcpy(category,"NA"); assign\_category(); set\_price();



void rset\_seat() //function called when a seat is canceled

{

r=0;c=0;strcpy(category,"NA"); assign\_category();

set\_price();

}

char get\_r(){return r;} //getter function to get number of rows

int get\_c(){return c;} //getter function to get number of columns

void get\_seat() //function to enter the seat, set its category and set its price

{

cout<<"\n Enter the seat : "; cout<<"\n\t\t\t\t\t\tROW : "; cin>>r; cout<<"\n\t\t\t\t\t\tCOLUMN : "; cin>>c;

r=toupper(r);

assign\_category(); //assigning category set\_price(); //assigning price

}

float get\_price() //function to set price by the user

{

assign\_category(); //getting category set\_price();

return price; //returning assigned price

}

void disp\_seat() //function to display seat number while displaying ticket

{

if(c<10) cout<<r<<" "<<c;

else



cout<<r<<' '<<c;

}

};

/\*----------------------------------ADMIN------------------------------------------

------------------------------\*/

class admin //class for setting a username and password for administrator

{

private:

char admin\_name[25]; char a\_pass[10];

long int admin\_ID;

public:

admin() //default constructor

{

strcpy(admin\_name,"Not Alotted"); strcpy(a\_pass,"password"); admin\_ID=12345678;

}

void set\_admin\_name() //getter function to get admin name

{

gets(admin\_name);

}

void set\_pass() //function to set password

{

char pass[10],ch;int i=-1; cout<<"\n\t\t\t\t\t\tEnter Password : "; do

{

ch=getch(); //this makes the entered password invisible pass[++i]=ch;

}

while(ch!='\r'); //take characters until the user press enter key



pass[i]='\0';

strcpy(a\_pass,pass); //setting the password

}

void get\_a\_id() //function to get admin ID

{

cin>>admin\_ID;

}

int check\_pass() //function to check the user entered password

{

pass\_check : //label for goto at the end of function char pass[10],ch;int i=-1;

cout<<"\n\t\t\t\t\t\tEnter Password : ";

do //loop for entering password

{

ch=getch(); pass[++i]=ch;

}

while(ch!='\r');

pass[i]='\0';

if(strcmp(pass,a\_pass)==0) //checking if the password is correct return 1;

else //else function moves the program to pass\_check

{

cout<<"\n \t\t\t\t\t\tWRONG PASSWORD !!!

\n\n\t\t\t\t\t\t<<Press a key to enter again>> ";

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

------------------------------------------------------------------";

cin.get();

goto pass\_check;

}



};

admin ADMIN; //object of class admin is declared

//--------------------------------SLOT-----------------------------------------------

--------------------------

class slot

{

int mat[12][12]; int slot\_num;

int tot\_s,booked\_seat; public:

slot()

{

for(int i=0;i<12;i++) for(int j=0;j<12;j++) mat[i][j]=0;

slot\_num=0; tot\_s=0; booked\_seat=0;

}

void bk\_seat(char r,int c) //function books the seat with r row and c column

{

int i=r; i-=65;

mat[i][c-1]=1; booked\_seat++;

}

void canc\_seat(char r,int c) //function cancels the seat with r row and c column

{

int i=r;

mat[i][c-1]=0; booked\_seat--;



}

void reset\_seat(char r,int c) //function that resets a seat when ticket is cancelled

{

int i=r; i-=65;

mat[i-1][c-1]=0;

}

void reset\_slot() //function to free all seats

{

for(int i=0;i<12;i++) for(int j=0;j<12;j++) mat[i][j]=0;

}

int get\_avail() //function to check if seat is available

{

if(booked\_seat==144) return 1;

else

return 0;

}

void disp\_slot() //function to display screen status

{

int i,j;

char k='A',ch='\_'; cout<<"\n\n\n";

// cout<<"\t\t\t 1 2 3 4 5 6 7 8 9 10 11

12 \n";

cout<<"\t\t --------------------------------------------------------------

---------\n";

cout<<"\t\t | 1 2 3 4 5 6 PREMIUM 7 8 9

10 11 12 |";

cout<<"\n\t\t ----------------------------------------------------------

-------------\n";

for(i=0;i<4;i++)

cout<<"\t\t\t"; for(j=0;j<36;j++)

{

if(j<24&&j>12)

{

if(j==18) cout<<k++;

else

cout<<" ";

}

else if(j%2==0) cout<<'|';

else if(mat[i][convert\_odd(j)]==1)

{

cout<<'\_'<<sk<<'\_';

}

else

}

cout<<" ";

cout<<"|\n";



}

cout<<"\t\t -----------------------------------------------------------------

------\n";

//cout<<"\t\t

|||||||||||||||||||||||||||||||||||||||||||||||||

||||||||||||||||||||||";

cout<<"\n\t\t -------------------------------------------------------------

----------\n";

cout<<"\t\t | 1 2 3 4 5 6 GOLD 7 8 9 10 11

12 |";

cout<<"\n\t\t --------------------------------------------------------------

---------\n";

for(i=4;i<8;i++)

cout<<"\t\t\t"; for(j=0;j<36;j++)

{

if(j<24&&j>12)

{

if(j==18) cout<<k++;

else

cout<<" ";

}

else if(j%2==0) cout<<'|';

else if(mat[i][convert\_odd(j)]==1)

{

cout<<'\_'<<sk<<'\_';

}

else

}

cout<<" ";

cout<<"|\n";



}

cout<<"\t\t ------------------------------------------------------------------

-----\n";

//cout<<"\t\t

|||||||||||||||||||||||||||||||||||||||||||||||||

||||||||||||||||||||||";

cout<<"\n\t\t --------------------------------------------------------------

---------\n";

cout<<"\t\t | 1 2 3 4 5 6 SILVER 7 8 9 10 11

12 |";

cout<<"\n\t\t --------------------------------------------------------------

---------\n";

for(i=8;i<12;i++)

{

cout<<"\t\t\t"; for(j=0;j<36;j++)

{

if(j<24&&j>12)

{

if(j==18) cout<<k++;

else

cout<<" ";

}

else if(j%2==0) cout<<'|';

else if(mat[i][convert\_odd(j)]==1)

{

cout<<'\_'<<sk<<'\_';

}

else

}

cout<<" ";

cout<<"|\n";



}

cout<<"\t\t --------------------------------------------------------------------

---\n";

cout<<"\n\n\n\t\t ---------------------------------------------------------

--------------\n";

cout<<"\t\t |>>>>>>>>>>>>>>>>>> SCREEN

<<<<<<<<<<<<<<<<<|";

cout<<"\n\t\t ----------------------------------------------------------------

-------\n";

}

};



//--------------------------------SCREEN-------------------------------------------

---------------------------

class screen //class that defines the screen which has 4 slots

{

private : slot s[4];

int s\_stat[4]; public:

int scr\_num; int max\_shows;

screen() //default constructor

{

scr\_num=0; max\_shows=4; for(int i=0;i<4;i++)

s\_stat[i]=0;

}

int get\_slot(int n) //getter function

{

return s\_stat[n-1];

}

void reset\_screen(int n) //function to reset screen

{

s[n-1].reset\_slot();

}

void set\_m\_slot(int n) //function to allocate a movie to a screen

{

s\_stat[n-1]=1; max\_shows--;

}

void reset\_m\_slot(int n) //function to deallocate a movie to a screen



{

max\_shows++; s\_stat[n-1]=0;

}

void book\_seat(char r,int c,int n) //function to book a seat

{

r=toupper(r);

s[n-1].bk\_seat(r,c);

}

void cancel\_seat(char r,int c,int n) //function to cancel a show

{

r=toupper(r);

s[n-1].canc\_seat(r,c);

}

int get\_slot\_stat(int n) //function to check the slot status

{

return s[n-1].get\_avail();

}

void disp\_screen(int n) //function to display a screen slot

{

s[n-1].disp\_slot();

}

};

//---------------------------------------------------------------------------------------

--------------------------

void reset\_screen\_status(int n) //function to reset the screen status

{

int pos; screen m; fstream fin; if(n==1)



fin.open("Screen\_Info1.dat",ios::binary|ios::in|ios::out);

//opening the screen file else if(n==2)

fin.open("Screen\_Info2.dat",ios::binary|ios::in|ios::out);

//opening the screen file else if(n==3)

fin.open("Screen\_Info3.dat",ios::binary|ios::in|ios::out);

//opening the screen file else

fin.open("Screen\_Info4.dat",ios::binary|ios::in|ios::out);

//opening the screen file

pos=fin.tellg(); //storing starting address fin.read((char\*)&m,sizeof(m)); //reading a screen for(int i=0;i<4;i++)

m.reset\_screen(i+1); fin.seekg(pos); fin.write((char\*)&m,sizeof(m)); fin.close();

}

void disp\_screen\_status() //function to display screen status

{

screen m; fstream filin;

filin.open("Screen\_Info1.dat",ios::binary|ios::in);

//opening screen 1

filin.read((char\*)&m,sizeof(m)); //displaying contents

cout<<"\n\t\t\t Screen "<<1<<" ( "<<disp\_avail(!m.max\_shows)<<" ) ";

filin.close();

filin.open("Screen\_Info2.dat",ios::binary|ios::in);



//opening screen 2 filin.read((char\*)&m,sizeof(m)); cout<<"\n\t\t\t Screen "<<2<<" (

"<<disp\_avail(!m.max\_shows)<<" ) "; filin.close();

filin.open("Screen\_Info3.dat",ios::binary|ios::in);

//opening screen 3 filin.read((char\*)&m,sizeof(m)); cout<<"\n\t\t\t Screen "<<3<<" (

"<<disp\_avail(!m.max\_shows)<<" ) "; filin.close();

filin.open("Screen\_Info4.dat",ios::binary|ios::in); filin.read((char\*)&m,sizeof(m)); cout<<"\n\t\t\t Screen "<<4<<" (

"<<disp\_avail(!m.max\_shows)<<" ) "; filin.close();

}

//----------------------------------------------------------MOVIE------------------

---------------------------------

class movie //class to define a movie

{

private :

int mcode;

char m\_name[100]; char cert[4];

int nos,num; //No. of shows float duration;

int m\_slot[4]; void gen\_mcode()

{

ifstream fmcode("Mcode.dat",ios::binary|ios::in); fmcode.read((char\*)&mcode,sizeof(mcode)); ofstream fitout("Mcode.dat",ios::binary|ios::out);

if(mcode==0)



{

mcode=101;

fitout.write((char\*)&mcode,sizeof(mcode)); // redundancy could be removed by writing it only once

}

else

{

mcode+=1;

fitout.write((char\*)&mcode,sizeof(mcode)); // redundancy could be removed by writing it only once

}

fitout.close(); fmcode.close();

}

public:

movie() //default constructor

{

mcode=0; strcpy(m\_name,"Not Alloted"); strcpy(cert,"NA");

nos=0; num=0; duration=0;

for(int i=0;i<4;i++) m\_slot[i]=0;

}

void rem\_movie() //function to remove a movie and free a screen slot

{

screen s;

int pos,n=num; fstream fout;

if(n==1) //checking screen number fout.open("Screen\_Info1.dat",ios::binary|ios::in|ios::out);



else if(n==2) fout.open("Screen\_Info2.dat",ios::binary|ios::in|ios::out);

else if(n==3) fout.open("Screen\_Info3.dat",ios::binary|ios::in|ios::out);

else

fout.open("Screen\_Info4.dat",ios::binary|ios::in|ios::out); pos=fout.tellg();

fout.read((char \*)&s,sizeof(s)); for(int i=0;i<4;i++)

{

if(m\_slot[i]==1)

{

s.reset\_m\_slot(i+1); //reseting the slot

}

}

fout.seekp(pos); fout.write((char \*)&s,sizeof(s)); fout.close(); //closing file

}

int get\_snum(){return num;}

void create\_movie(); //declaring a function void disp\_movie(); //declaring a function int get\_mcode() //getter function

{

return mcode;

}

char\* get\_m\_name() //getter function

{

return m\_name;

}

void set\_m\_name() //setter function



{

gets(m\_name);

}

char\* get\_cert() //getter function

{

return cert;

}

void set\_cert() //setter function

{

gets(cert);

}

int get\_nos() //getter function

{

return nos;

}

void set\_nos() //setter function

{

cin>>nos;

}

float get\_dura() //getter function

{

return duration;

}

void set\_dura() //setter function

{

cin>>duration;

}

void get\_slots() //function to update number of shows

{

rem\_movie(); int n=num;

screen m;



char str[]="Screen\_Info0.dat"; str[11]+=n;

ifstream fin;

ofstream fout("temp.dat",ios::out|ios::binary);

if(n==1) fin.open("Screen\_Info1.dat",ios::binary|ios::in);

else if(n==2) fin.open("Screen\_Info2.dat",ios::binary|ios::in);

else if(n==3) fin.open("Screen\_Info3.dat",ios::binary|ios::in);

else

fin.open("Screen\_Info4.dat",ios::binary|ios::in);

fin.read((char\*)&m,sizeof(m));

cout<<"\n Enter the number of shows ( max "<<m.max\_shows<<" ) ";

cin>>nos; cin.get();

cout<<"\n Select the show timings : ";

cout<<"\n \t\t\t<1> Morning (09:00 AM - 12:00 PM)---- ( "<<disp\_avail(m.get\_slot(1))<<" ) ";

cout<<"\n \t\t\t<2> Afternoon (12:15 PM - 03:15 PM)---

-( "<<disp\_avail(m.get\_slot(2))<<" ) ";

cout<<"\n \t\t\t<3> Evening (03:30 PM - 06:30 PM)---- ( "<<disp\_avail(m.get\_slot(3))<<" ) ";

cout<<"\n \t\t\t<4> Night (07:00 PM - 10:00 PM)----( "<<disp\_avail(m.get\_slot(4))<<" ) ";

for(int i=0;i<nos;i++)

{

int tm;

cout<<"\n\t Show #"<<i+1<<" : \n\t\t\t\t"; cin>>tm;

if(tm==1)

{

m.set\_m\_slot(1);m\_slot[tm-1]=1;

}

else if(tm==2)



{

m.set\_m\_slot(2);m\_slot[tm-1]=1;

}

else if(tm==3)

{

m.set\_m\_slot(3); m\_slot[tm-1]=1;

}

else if(tm==4)

{

m.set\_m\_slot(4);m\_slot[tm-1]=1;

}

else

cout<<"\n Wrong Choice !!!!";

;

}

fout.write((char\*)&m,sizeof(m)); fin.close();

fout.close(); remove(str); rename("temp.dat",str);

}

int return\_m\_slot(int n)

{

return m\_slot[n-1];

}

};

void movie::create\_movie() //function to create a movie

{

gen\_mcode(); cin.get();

cout<<"\n Enter movie name : "; gets(m\_name);



cout<<"\n Enter Certificate : "; gets(cert);cin.get();

cout<<"\n Enter the duration of movie(in minutes) : "; cin>>duration;cin.get();

cout<<"\n Select a Screen : "; disp\_screen\_status(); cout<<"\n Enter (1 -4) : "; cin>>num;

cin.get(); get\_slots();

}

void movie::disp\_movie() //member function to display a movie

{

cout.setf(ios::left);

cout<<endl<<setw(8)<<mcode<<setw(30)<<m\_name<<setw(7)<<c ert<<setw(10)<<duration<<setw(20)<<nos<<setw(7)<<" SCREEN "<<setw(1)<<num;

}

void insert\_movie() //function to insert a movie

{

movie m; //creating an object of class movie m.create\_movie();

ofstream fout("Movies\_this\_week.dat",ios::binary|ios::app); fout.write((char\*)&m,sizeof(m)); //writing data to file fout.close(); //closing file

}

void display\_movies() //function to display all movies

{

int r=0;

cout<<"\n\n------------------------------------------------------Display

Menu--------------------------------------------------------------------";

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



------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Search for a movie"; cout<<"\n\n\t\t\t\t\t\t<2> Display all movies"; cout<<"\n\n\t\t\t\t\t\t<3> <<BACK";

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

------------------------------------------------------------";

int ch;

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

int n; movie m;

system("cls"); //clearing screen switch(ch)

{

case 1:

{

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

---------------------------------------------------------------------";

cout<<"\n\n Enter the movie code : "; cin>>n;

ifstream fin("Movies\_this\_week.dat",ios::binary|ios::in); while(fin)

{

fin.read((char\*)&m,sizeof(m)); if(fin.eof())

break; if(m.get\_mcode()==n)

{

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

---------------------------------------------------------------------------\n";

cout.setf(ios::left); cout<<setw(8)<<"M\_Code"<<setw(30)<<" Movie

Name"<<setw(7)<<"Cert."<<setw(10)<<"Duration"<<setw(20)<<"N

o. of Shows"<<setw(7)<<" SCREEN ";

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

---------------------------------------------------------------------------\n";

m.disp\_movie();

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



-------------------------------------------------------------------------------";

cin.get(); break;

}

}

fin.close(); break;

}

case 2 :

{

ifstream fin("Movies\_this\_week.dat",ios::binary|ios::in); cout<<"\n--------------------------------------------------------------------

-----------------------------------------------------------------\n";

cout.setf(ios::left); cout<<setw(8)<<"M\_Code"<<setw(30)<<" Movie

Name"<<setw(7)<<"Cert."<<setw(10)<<"Duration"<<setw(20)<<"N

o. of Shows"<<setw(7)<<" SCREEN ";

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

-----------------------------------------------------------------\n";

while(fin)

{

cout.setf(ios::left); fin.read((char\*)&m,sizeof(m)); if(fin.eof())

break;

m.disp\_movie();

}

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

----------------------------------------------------------------------";

fin.close();

cin.get(); break;

}



case 3:

{

r=1; break;

}

}

}

void remove\_movie() //function to remove a movie

{

int r=0; scr:

cout<<"\n\n------------------------------------------------------Delete

Menu---------------------------------------------------------------------";

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

------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Delete a movie"; cout<<"\n\n\t\t\t\t\t\t<2> Delete all movies"; cout<<"\n\n\t\t\t\t\t\t<3> <<BACK";

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

------------------------------------------------------------";

int ch;

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

if(ch==1)

{

int n; movie m;

system("cls");

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

----------------------------------------------------------------";

cout<<"\n\n Enter the movie code : "; cin>>n;

ifstream fin("Movies\_this\_week.dat",ios::binary|ios::in);

//input stream file

ofstream fout("Dummy.dat",ios::binary|ios::out);



//output stream file

while(!fin.eof()) //getting cursor to eof

{

fin.read((char\*)&m,sizeof(m)); if(fin.eof())

break;

if(m.get\_mcode()==n) //searching for movie

{

cout<<"\n The movie is : ";

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

-----------------------------------------------------------------------\n";

cout.setf(ios::left); cout<<setw(8)<<"M\_Code"<<setw(30)<<" Movie

Name"<<setw(7)<<"Cert."<<setw(10)<<"Duration"<<setw(20)<<"N

o. of Shows"<<setw(7)<<" SCREEN ";

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

-----------------------------------------------------------------------";

m.disp\_movie();

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

----------------------------------------------------------------------------"; cout<<"\n MOVIE DELETED !";

m.rem\_movie(); cin.get();

break;

}

else

{

fout.write((char\*)&m,sizeof(m));

}

}

while(!fin.eof()) //getting cursor to eof

{

fin.read((char\*)&m,sizeof(m)); if(fin.eof())

break; fout.write((char\*)&m,sizeof(m));

}

fout.close();

fin.close();



remove("Movies\_this\_week.dat"); //removing original file rename("Dummy.dat","Movies\_this\_week.dat");

//renaming dummy as original file cin.get();

goto scr;

}

if(ch==2)

{

for(int i=0;i<4;i++)

{

reset\_screen\_status(i+1);

}

remove("Movies\_this\_week.dat"); cout<<"\n\t\t\t All Movies Deleted ! ";

cin.get(); goto scr;

}

if(ch==3) r=0;

cin.get();

}

void update\_movie() //function to update a movie

{

movie m; int n;

ifstream filin("Movies\_this\_week.dat",ios::binary|ios::in); ofstream filout("Dummy.dat",ios::binary|ios::out); cout<<"\nEnter the movie code : ";

cin>>n;

system("cls"); //clearing screen while(!filin.eof())

{

filin.read((char\*)&m,sizeof(m)); if(filin.eof())

break;

if(m.get\_mcode()==n)



{

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

------------------------------------------------------------------\n";

cout.setf(ios::left); cout<<setw(8)<<"M\_Code"<<setw(30)<<" Movie

Name"<<setw(7)<<"Cert."<<setw(10)<<"Duration"<<setw(20)<<"N

o. of Shows"<<setw(7)<<" SCREEN ";

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

------------------------------------------------------------------\n";

m.disp\_movie();

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

----------------------------------------------------------------------";

cin.get(); break;

}

else

filout.write((char\*)&m,sizeof(m));

}

int ch; do

{

cin.get();

system("cls");

cout<<"\n\n------------------------------------------------------Update

Menu--------------------------------------------------------------------";

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

---------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Change Movie name"; cout<<"\n\n\t\t\t\t\t\t<2> Change movie certificate "; cout<<"\n\n\t\t\t\t\t\t<3> Change movie length "; cout<<"\n\n\t\t\t\t\t\t<4> Change number of shows "; cout<<"\n\n\t\t\t\t\t\t<5> <<BACK";

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

---------------------------------------------------------------";

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

cin.get();

switch(ch) //getting new values



{

case 1: //getting new name

{

cout<<"\nEnter the new movie name

:";m.set\_m\_name();

break;

}

case 2: //getting new certificate

{

cout<<"\nEnter the new movie certificate

:";m.set\_cert();

break;

}

case 3: //getting new movie length

{

cout<<"\nEnter the new movie length :";m.set\_dura(); break;

}

case 4: //getting new slot

{

m.get\_slots(); break;

}

case 5: //exit

{

break;

}

}

}

while(ch!=5);

filout.write((char\*)&m,sizeof(m)); //writing updated record cout<<"\n\n\n\t\t\t\t\t\tRecord updation successful!";

while(!filin.eof()) //writing other records

{

filin.read((char\*)&m,sizeof(m)); if(filin.eof())

break;

filout.write((char\*)&m,sizeof(m));



}

filout.close(); //closing file filin.close(); //getting cursor to eof

remove("Movies\_this\_week.dat"); //deleting old record rename("Dummy.dat","Movies\_this\_week.dat"); //renaming

new record

getch(); //clearing buffer

}

//---------------------------------------------------------------------------------------

---------------------------

//-------------------------------TICKET---------------------------------------------

-----------------------------

class ticket //class that defines a ticket

{

long int ticket\_no; char mname[30]; seat s;

int slot\_num,snum,m\_code; float cost;

void gen\_ticket\_no() //function to generate ticket number

{

ifstream fmcode("Ticket\_Num.dat",ios::binary|ios::in); fmcode.read((char\*)&ticket\_no,sizeof(ticket\_no)); ofstream fitout("Ticket\_Num.dat",ios::binary|ios::out); if(ticket\_no==0)

{

ticket\_no=1000000; fitout.write((char\*)&ticket\_no,sizeof(ticket\_no)); //

redundancy could be removed by writing it only once

}

else



{

ticket\_no+=1; fitout.write((char\*)&ticket\_no,sizeof(ticket\_no)); //

redundancy could be removed by writing it only once

}

fitout.close(); fmcode.close();

}

public:

ticket() //default constructor to assign default values

{

ticket\_no=0;slot\_num=0; cost=0;snum=0;

}

void set\_cost() //function to set value of cost

{

cost=s.get\_price();

}

float get\_cost() //getter function

{

return cost;

}

void set\_movie(int code) //function to set a movie according to given code

{

movie m; m\_code=code; system("cls");

ifstream fin("Movies\_this\_week.dat",ios::binary|ios::in); while(fin)

{

fin.read((char\*)&m,sizeof(m));

if(fin.eof()) //giving end to loop by eof break;

if(m.get\_mcode()==code)

{

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

------------------------------------------------------------------------\n";

cout.setf(ios::left); cout<<setw(8)<<"M\_Code"<<setw(30)<<" Movie

Name"<<setw(7)<<"Cert."<<setw(10)<<"Duration"<<setw(20)<<"N

o. of Shows"<<setw(7)<<" SCREEN ";

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

------------------------------------------------------------------------\n";

m.disp\_movie(); //displaying selected movie cout<<"\n\n---------------------------------------------------------

----------------------------------------------------------------------------";

cin.get();

snum=m.get\_snum(); //to set the screen number of

ticket

int i=-1; do

{

++i;



mname[i]=\*(m.get\_m\_name()+i);

}

while(mname[i]!='\0'); break;

}

}

fin.close(); //file closed

}

int get\_t\_snum() //getter function

{

return snum;

}

void set\_slotnum(int n) //function to set slot number

{

slot\_num=n;

}

int get\_slotnum() //getter function



{

return slot\_num;

}

long int get\_tno() //getter function

{

return ticket\_no;

}

void book\_ticket() //function to book a ticket

{

gen\_ticket\_no(); //function to generate ticket number s.get\_seat(); //function to get seat number

int n=snum; screen sc;

char str[]="Screen\_Info0.dat"; //to select the file to be opened

str[11]+=n; ifstream fin;

ofstream fout("temp.dat",ios::out|ios::binary); if(n==1)

fin.open("Screen\_Info1.dat",ios::binary|ios::in); else if(n==2)

fin.open("Screen\_Info2.dat",ios::binary|ios::in); else if(n==3)

fin.open("Screen\_Info3.dat",ios::binary|ios::in); else

fin.open("Screen\_Info4.dat",ios::binary|ios::in); fin.read((char\*)&sc,sizeof(sc)); //to read data from file sc.book\_seat(s.get\_r(),s.get\_c(),slot\_num);

set\_cost();

fout.write((char\*)&sc,sizeof(sc)); fin.close();

fout.close(); //file closed remove(str); //file deletion

rename("temp.dat",str); //renaming temp.dat as str



}

void cancel\_book() //function to cancel a booked ticket

{

int n=snum; screen sc;

char str[]="Screen\_Info0.dat"; //using string to open user manipulated file

str[11]+=n; ifstream fin;

ofstream fout("temp.dat",ios::out|ios::binary); if(n==1)

fin.open("Screen\_Info1.dat",ios::binary|ios::in); else if(n==2)

fin.open("Screen\_Info2.dat",ios::binary|ios::in); else if(n==3)

fin.open("Screen\_Info3.dat",ios::binary|ios::in); else

fin.open("Screen\_Info4.dat",ios::binary|ios::in); fin.read((char\*)&sc,sizeof(sc)); sc.cancel\_seat(s.get\_r(),s.get\_c(),slot\_num); //cancelling

ticket

fout.write((char\*)&sc,sizeof(sc)); fin.close();

fout.close(); //file closed

remove(str); //deleting file rename("temp.dat",str); //renaming file

}

void disp\_ticket() //function to display ticket

{

int l=strlen(mname); l=34-l;

cout<<"\n\t+--------------------------------------------+";

cout<<"\n\t| T No. : "<<ticket\_no<<" |"; cout<<"\n\t| Movie : ";

cout<<mname; for(int i=0;i<l;i++)



cout<<' '; cout<<"|";

cout<<"\n\t| Show : ";

show\_time\_slot(slot\_num); //function to show time slot on ticket

cout<<" |";

cout<<"\n\t|--------------------------------------------|";

cout<<"\n\t| Seat : ";

s.disp\_seat(); //function to display seat

set\_cost(); //function to set cost of all booked tickets

cout<<" | Screen : "<<snum<<" | Cost : Rs. "<<cost<<"|"; cout<<"\n\t+--------------------------------------------+";

}

void change\_movie() //function to change movie in updating record

{

movie m;

ifstream fin("Movies\_this\_week.dat",ios::binary|ios::in); cout<<"\n\n------------------------------------------------------MOVIE

MENU----------------------------------------------------------------------";

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

---------------------------------------------------------------\n";

cout.setf(ios::left); cout<<setw(8)<<"M\_Code"<<setw(30)<<" Movie

Name"<<setw(7)<<"Cert."<<setw(10)<<"Duration"<<setw(20)<<"N

o. of Shows"<<setw(7)<<" SCREEN ";

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

---------------------------------------------------------------\n";

while(fin)

{

cout.setf(ios::left);

fin.read((char\*)&m,sizeof(m)); if(fin.eof())

break;

m.disp\_movie(); //function to display movie

}

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

-------------------------------------------------------------------";

fin.close();

cin.get(); int n;

cout<<"\n Enter the movie code : "; cin>>n;

set\_movie(n); //function called to set movie

}

void change\_show\_time() //function to change any show's timing

{

movie m;

ifstream fin("Movies\_this\_week.dat",ios::binary|ios::in); while(fin)

{

record

fin.read((char\*)&m,sizeof(m)); //to read data from file if(fin.eof())

break;

if(m.get\_mcode()==m\_code) //searching required

{

screen s;

int n=m.get\_snum(); ifstream finl; if(n==1)

finl.open("Screen\_Info1.dat",ios::binary|ios::in); else if(n==2) finl.open("Screen\_Info2.dat",ios::binary|ios::in);



else if(n==3) finl.open("Screen\_Info3.dat",ios::binary|ios::in); else finl.open("Screen\_Info4.dat",ios::binary|ios::in);



finl.read((char\*)&s,sizeof(s)); finl.close();

cin.get();

cout<<"\n Select Show timing : "; if(m.return\_m\_slot(1)==1)

cout<<"\n \t\t\t<1> Morning (09:00 AM - 12:00 PM)---

-( "<<disp\_avail(s.get\_slot\_stat(1))<<" ) "; if(m.return\_m\_slot(2)==1)

cout<<"\n \t\t\t<2> Afternoon (12:15 PM - 03:15 PM)---

-( "<<disp\_avail(s.get\_slot\_stat(2))<" ) "; if(m.return\_m\_slot(3)==1)

cout<<"\n \t\t\t<3> Evening (03:30 PM - 06:30 PM)---- ( "<<disp\_avail(s.get\_slot\_stat(3))<<" ) ";

if(m.return\_m\_slot(4)==1)

cout<<"\n \t\t\t<4> Night (07:00 PM - 10:00 PM)----( "<<disp\_avail(s.get\_slot\_stat(4))<<" ) ";

cout<<"\n Enter (1-4) : "; cin>>n;

set\_slotnum(n); //setting slot number of movie cin.get();

break;

}

}

fin.close();

}

char \*get\_m\_name() //getter function

{

return mname;

}

};

//---------------------------------------------------------------------------------------

-------------------------------------------------

//---------------------------------------------------------------------------------------

-------------------------------------------------

//-----------------------------------------------------USER--------------------------

--------------------------------------------------

class user

{

private:

int tickets\_sold; int user\_no;

char UserName[40]; float amount;

public:

user() //default constructor

{

user\_no=1;strcpy(UserName,"NA");tickets\_sold=0;amount=0;

}

void disp\_user() //function to display user details

{

cout.setf(ios::left);

cout<<endl<<setw(5)<<"USER"<<setw(6)<<user\_no<<setw(20)<<U serName<<setw(15)<<tickets\_sold<<setw(2)<<"Rs"<<setw(8)<<am ount;

}

void get\_user() //function to get user

{

-----";

}

user();

cout<<"\n---------------------USER "<<"DETAILS"<<" --------------

cout<<"\n\n Enter an User Name (max 10 characters) : "; gets(UserName); //function called for getting username cin.get();



int get\_uno() //getter function



{

return user\_no;

}

void set\_tsold(int n) //function to set number of tickets sold

{

tickets\_sold+=n;

}

void set\_amount(float n) //setter function

{

amount+=n;

}

};

user u;

void book\_a\_show() //function to book a show

{

ufil.open("User\_file.dat",ios::in|ios::binary); //file opening ufil.read((char\*)&u,sizeof(u));

ofstream tufil("temu.dat",ios::out|ios::binary);

ofstream fout("Ticket\_file.dat",ios::binary|ios::app|ios::ate);

//ofstream ftemp("Temp.dat",ios::binary|ios::out); int pos=fout.tellp(); //storing starting address

ticket t; //objects of classes declared movie m;

ifstream fin; fin.open("Movies\_this\_week.dat",ios::binary|ios::in); cout<<"\n\n------------------------------------------------------MOVIE

MENU----------------------------------------------------------------------";

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

---------------------------------------------------------------\n";

cout.setf(ios::left); cout<<setw(8)<<"M\_Code"<<setw(30)<<" Movie



Name"<<setw(7)<<"Cert."<<setw(10)<<"Duration"<<setw(20)<<"N

o. of Shows"<<setw(7)<<" SCREEN ";

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

---------------------------------------------------------------\n";

while(fin) //finishing loop till eof

{

cout.setf(ios::left);

fin.read((char\*)&m,sizeof(m)); if(fin.eof())

break;

m.disp\_movie(); //if found, displaying movie

}

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

-------------------------------------------------------------------";

fin.close(); //file closed cin.get();

int n;

cout<<"\n Enter the movie code : "; cin>>n;

t.set\_movie(n); //setting movie fin.open("Movies\_this\_week.dat",ios::binary|ios::in);

while(fin)

{

cout.setf(ios::left);

fin.read((char\*)&m,sizeof(m));

if(m.get\_mcode()==n) //checking if mcode is matching given value of n

break;

}

fin.close(); //file closed screen s;

n=t.get\_t\_snum();

ifstream finl; //string declared for opening file if(n==1)



finl.open("Screen\_Info1.dat",ios::binary|ios::in); else if(n==2)

finl.open("Screen\_Info2.dat",ios::binary|ios::in); else if(n==3)

finl.open("Screen\_Info3.dat",ios::binary|ios::in); else

finl.open("Screen\_Info4.dat",ios::binary|ios::in); finl.read((char\*)&s,sizeof(s)); //data reading

cin.get();

cout<<"\n Select Show timing : "; //selecting screen timing

if(m.return\_m\_slot(1)==1)

cout<<"\n \t\t\t<1> Morning (09:00 AM - 12:00 PM)----( "<<disp\_avail(s.get\_slot\_stat(1))<<" ) ";

if(m.return\_m\_slot(2)==1)

cout<<"\n \t\t\t<2> Afternoon (12:15 PM - 03:15 PM)----( "<<disp\_avail(s.get\_slot\_stat(2))<<" ) ";

if(m.return\_m\_slot(3)==1)

cout<<"\n \t\t\t<3> Evening (03:30 PM - 06:30 PM)----( "<<disp\_avail(s.get\_slot\_stat(3))<<" ) ";

if(m.return\_m\_slot(4)==1)

cout<<"\n \t\t\t<4> Night (07:00 PM - 10:00 PM)----( "<<disp\_avail(s.get\_slot\_stat(4))<<" ) ";

cout<<"\n Enter (1-4) : "; cin>>n; t.set\_slotnum(n);

int x=n; system("cls");

cout<<"\n\n Enter the number of seats : "; cin>>n;

cout<<"\n Choose "<<n<<" Seats : ";

s.disp\_screen(x); //displaying screen with booked ticket marked as x

finl.close();

float t\_cost=0;



cout<<"\n\n Enter "<<n<<" seats : "; for(int i=0;i<n;i++)

{

t.book\_ticket();

t\_cost+=(t.get\_cost()); //adding cost of ticket to total cost fout.write((char\*)&t,sizeof(t));

}

cin.get();

fout.close();

system("cls");

ifstream fil("Ticket\_file.dat",ios::binary|ios::in); fil.seekg(pos);

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

{

fil.read((char\*)&t,sizeof(t)); cin.get();

t.disp\_ticket(); //displaying all ticket

}

u.set\_tsold(n); //setting number of tickets sold cout<<"\n\n------------------------------------------------------------------

-------------------------------------------------------------------\n";

cout<<"\n\n\t\t "<<n<<" tickets for \" "<<t.get\_m\_name()<<" \""<<" SCREEN "<<t.get\_t\_snum();

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

----------------------------------------------------------------\n";

cout<<" \t \t:::: TOTAL COST : Rs. "<<t\_cost<<" ::::" ; cout<<"\n---------------------------------------------------------------------

----------------------------------------------------------------\n";

u.set\_amount(t\_cost); //setting amount of money t\_cost=0;

fil.close();

finl.close();

fin.close(); tufil.write((char\*)&u,sizeof(u)); tufil.close();

ufil.close();

remove("User\_file.dat"); //deleting user\_file.dat file rename("temu.dat","User\_file.dat"); //renaming temu as



user\_file

cin.get();

}

void display\_booking() //function to display booked tickets

{

int r=0; ticket t; long int n;

ifstream filin("Ticket\_file.dat",ios::binary|ios::in); cout<<"\n\n------------------------------------------------------Display

Menu--------------------------------------------------------------------";

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

------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Search for a ticket"; cout<<"\n\n\t\t\t\t\t\t<2> Display all tickets"; cout<<"\n\n\t\t\t\t\t\t<3> <<BACK";

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

------------------------------------------------------------";

int ch;

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

system("cls"); switch(ch)

{

case 1:

{

cout<<"\nEnter the ticket no. : "; cin>>n;

system("cls");

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

Ticket File---------------------------------------------------------------------";

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



------------------------------------------------------------------------";

while(!filin.eof())

{

filin.read((char\*)&t,sizeof(t)); if(filin.eof())

break; if(t.get\_tno()==n)

{

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

---------------------------------------------------------------------------------\n";

t.disp\_ticket();

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

-------------------------------------------------------------------------------------";

cin.get(); break;

}

}

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

------------------------------------------------------------------------";

break;

}

case 2 :

{

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

Ticket File---------------------------------------------------------------------";

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

---------------------------------------------------------------------";

while(!filin.eof())

{

filin.read((char\*)&t,sizeof(t)); if(filin.eof())

break;

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

---------------------------------------------------------------------";

t.disp\_ticket();

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



---------------------------------------------------------------------";

}

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

---------------------------------------------------------------------";

cin.get();

cin.get(); break;

}

case 3:

{

filin.close(); //function closed r=1; break;

}

}

}

void cancel\_booking() //function to cancel booking

{

int ch; ufil.open("User\_file.dat",ios::in|ios::binary); ufil.read((char\*)&u,sizeof(u));

ofstream tufil("temu.dat",ios::out|ios::binary);

int r=0; ticket t; long int n;

ifstream filin("Ticket\_file.dat",ios::binary|ios::in); do

{

system("cls");

cout<<"\n\n------------------------------------------------------Delete

Menu---------------------------------------------------------------------";

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

---------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Cancel a booking";

cout<<"\n\n\t\t\t\t\t\t<2> Reset ticket file"; cout<<"\n\n\t\t\t\t\t\t<3> <<BACK";



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

---------------------------------------------------------------";

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

switch(ch)

{

case 1 : //cancelling booking

{

long int n; system("cls");

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

------------------------------------------------------------------";

cout<<"\n\n Enter the ticket no : "; cin>>n;

ofstream fout("Dummy.dat",ios::binary|ios::out);

while(!filin.eof())

{

filin.read((char\*)&t,sizeof(t)); if(filin.eof())

break; if(t.get\_tno()==n)

{

cout<<"\n The ticket is : ";

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

-----------------------------------------------------------------------\n";

t.disp\_ticket();

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

---------------------------------------------------------------------------";

cout<<"\n BOOKING NO. "<<t.get\_tno()<<" CANCELED !";

t.cancel\_book(); u.set\_tsold(-1); float x=t.get\_cost(); u.set\_amount(-x); cin.get();

}



else

{

fout.write((char\*)&t,sizeof(t));

}

}

while(!filin.eof())

{

filin.read((char\*)&t,sizeof(t)); if(filin.eof())

break; fout.write((char\*)&t,sizeof(t));

}

fout.close(); filin.close();

remove("Ticket\_file.dat"); rename("Dummy.dat","Ticket\_file.dat"); cin.get();

break;

}

case 2 : //resetting ticket file

{

remove("Ticket\_file.dat"); cout<<"\n File Reseted ! ";

fstream f("Ticket\_file.dat",ios::binary|ios::out); f.close();

cin.get(); break;

}

case 3 : //exit break;

}

tufil.write((char\*)&u,sizeof(u)); tufil.close();

ufil.close();

remove("User\_file.dat"); //removing original file rename("temu.dat","User\_file.dat"); //renaming file

cin.get();



}while(ch!=3); //choice must be 1-3

}

void reset\_screen() //function to reset screen after each show

{

int n,s;

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

------------------------------------------------------------";

cout<<"\n Enter the screen no. : "; cin>>n;

cout<<"\n Enter the slot : "; cin>>s;

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

------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t Screen "<<n<<" :::: Slot "<<s<<" Reset Successful ! ";

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

------------------------------------------------------------";

cin.get(); int pos;

screen m;

fstream fin; //opening screen name by string if(n==1)

fin.open("Screen\_Info1.dat",ios::binary|ios::in|ios::out); else if(n==2)

fin.open("Screen\_Info2.dat",ios::binary|ios::in|ios::out); else if(n==3)

fin.open("Screen\_Info3.dat",ios::binary|ios::in|ios::out); else

fin.open("Screen\_Info4.dat",ios::binary|ios::in|ios::out);

pos=fin.tellg(); fin.read((char\*)&m,sizeof(m));

m.reset\_screen(s); //resetting screen n and slot s fin.seekg(pos);

fin.write((char\*)&m,sizeof(m)); fin.close();



}

void disp\_scr\_stat() //function to check availability of screen or slot for a movie

{

screen s; int n;

cout<<"\n Enter the screen number : "; cin>>n;

cin.get();

ifstream finl; //opening file by string if(n==1)

finl.open("Screen\_Info1.dat",ios::binary|ios::in); else if(n==2)

finl.open("Screen\_Info2.dat",ios::binary|ios::in); else if(n==3)

finl.open("Screen\_Info3.dat",ios::binary|ios::in);

else

finl.open("Screen\_Info4.dat",ios::binary|ios::in);

int a; finl.read((char\*)&s,sizeof(s));

cout<<"\n Enter the slot number : "; cin>>a;

cin.get();

system("cls"); //screen clear

cout<<"\n\n------------------------------------------------Screen "<<n<<"

:::: Slot "<<a<<" -----------------------------------------------------------------";

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

------------------------------------------------------------";

* 1. isp\_screen(a); //displaying screen

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

------------------------------------------------------------";

cin.get();

finl.close(); //file closed

}

void user\_menu() //function to get user's choice from his menu



{

int ch; int n; do

{

system("cls");

cout<<"\n\n------------------------------------------------------USER

MENU-----------------------------------------------------------------------";

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

---------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Book a show"; cout<<"\n\t\t\t\t\t\t<2> Cancel Booking"; cout<<"\n\t\t\t\t\t\t<3> Display Ticket"; cout<<"\n\t\t\t\t\t\t<4> Reset Screen"; cout<<"\n\t\t\t\t\t\t<5> Display Screen Status"; cout<<"\n\t\t\t\t\t\t<6> <<BACK";

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

---------------------------------------------------------------";

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

switch(ch)

{

case 1 :

{

system("cls");

book\_a\_show(); //function to book a show with slot s and screen n

break;

}

case 2 :

{

system("cls");

cancel\_booking(); //function to cancel booking break;

}



case 3 :

{

system("cls");

display\_booking(); //function to display bookings break;

}

case 4 :

{

system("cls");

reset\_screen(); //function to reset screen break;

}

case 5:

{

disp\_scr\_stat(); //function to display screen/slot status break;

}

case 6 :

{

break; //back

}

}

}while(ch!=6); //accepting numbers only 1-6

}

//===============================================

============

void admin\_menu() //function to get user's choice from his menu

{

int ch; do

{

system("cls");



cout<<"\n\n------------------------------------------------------ADMIN

MENU----------------------------------------------------------------------";

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

---------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Update Admin account";

//cout<<"\n\t\t\t\t\t\t<2> Insert user"; //future enhancements

// cout<<"\n\t\t\t\t\t\t<3> Display user";

//cout<<"\n\t\t\t\t\t\t<4> Remove user"; cout<<"\n\t\t\t\t\t\t<2> Update Movies"; cout<<"\n\t\t\t\t\t\t<3> Display Movies";

//cout<<"\n\t\t\t\t\t\t<4> Update ticket cost"; cout<<"\n\t\t\t\t\t\t<4> Reset Screen"; cout<<"\n\t\t\t\t\t\t<5> <<BACK";

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

---------------------------------------------------------------";

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

switch(ch)

{

case 1 : //updating admin account

{

int ch2;

fstream fin("AP.dat",ios::binary|ios::in|ios::out); int pos;

pos=fin.tellg(); fin.read((char\*)&ADMIN,sizeof(ADMIN));

do

{

system("cls");

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

Update Menu--------------------------------------------------------------------";

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



---------------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Change Admin name"; cout<<"\n\n\t\t\t\t\t\t<2> Change password "; cout<<"\n\n\t\t\t\t\t\t<3> <<BACK";

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

---------------------------------------------------------------------";

cout<<"\n\n Enter your choice : "; cin>>ch2;

cin.get(); switch(ch2)

{

case 1 : //setting admin name

{

cout<<"\n Enter the new Admin name : "; ADMIN.set\_admin\_name();

break;

}

case 2 : //setting admin password

{

cout<<"\n Enter the new password : "; ADMIN.set\_pass();

break;

}

case 3 : //exit

{break;}

}

}while(ch2!=3); fin.seekp(pos);

fin.write((char\*)&ADMIN,sizeof(ADMIN)); fin.close();

break;

}

//future enhancements

/\* case 2 :

{

user u1; system("cls");

ofstream tufil("User\_file.dat",ios::out|ios::binary); u1.get\_user();



tufil.write((char\*)&u1,sizeof(u1)); tufil.close();

cin.get(); break;

}

case 3 :

{

system("cls"); ufil.open("User\_file.dat",ios::in|ios::binary); ufil.read((char\*)&u,sizeof(u));

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

---------------------------------------------------------------\n";

cout.setf(ios::left);

cout<<setw(10)<<"USER No "<<setw(20)<<" User Name "<<setw(15)<<" Tickets Sold "<<setw(10)<<" Amount ";

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

----------------------------------------------------------------";

u.disp\_user();

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

------------------------------------------------------------------";

cin.get();

ufil.close();

cin.get(); break;

}

case 4 :

{

system("cls"); ufil.open("User\_file.dat",ios::in|ios::binary); ufil.read((char\*)&u,sizeof(u));

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

---------------------------------------------------------------\n";

cout.setf(ios::left);

cout<<setw(10)<<"USER No "<<setw(20)<<" User Name "<<setw(15)<<" Tickets Sold "<<setw(10)<<" Amount ";

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



----------------------------------------------------------------";

u.disp\_user();

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

------------------------------------------------------------------";

cout<<"\n\n\t\t\t\t\t\t :::: User Removed ::::";

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

------------------------------------------------------------------";

cin.get();

ufil.close(); remove("User\_file.dat"); cin.get();

break;

}\*/

case 2: //updating movies

{

int ch1; do

{

system("cls");

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

---MOVIE MENU-------------------------------------------------------------------

---";

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

------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> Insert Movie"; cout<<"\n\t\t\t\t\t\t<2> Delete Movie"; cout<<"\n\t\t\t\t\t\t<3> Update Movie"; cout<<"\n\t\t\t\t\t\t<4> <<BACK";

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

------------------------------------------------------------";

cout<<"\n Enter your choice : "; cin>>ch1;

if(ch1==1)

{

insert\_movie(); //inserting movies

}

else if(ch1==2)



{

remove\_movie(); //removing movies

}

else if(ch1==3)

{

update\_movie(); //updation of movie

}

else //exit break;

}while(ch1!=4); break;

}

case 3 : //displaying movies

{

display\_movies(); cin.get();

break;

}

/\* case 4 :

{

break;

}\*/

case 4 : //resetting screen status

{

for(int i=0;i<4;i++)

{

reset\_screen\_status(i+1);

}

cout<<"\n Screen Reset Successful !!! "; cin.get();

cin.get(); //for pressing enter key break;

}

case 5 : break; //exit

}

}while(ch!=5);



}

int main()

{

//-----------------------------------------ADMIN PASSWORD SET--------

---------------------------------

//remove("AP.dat");

/\*ADMIN.set\_admin\_name(); ADMIN.set\_pass();

ofstream fout("AP.dat",ios::binary|ios::out); fout.write((char\*)&ADMIN,sizeof(ADMIN)); fout.close();

\*///-----------------------------------------------------------------------------------

------------------------------------

//for creating files for first time

/\*ofstream fo; fo.open("Ticket\_file.dat",ios::out|ios::binary); fo.close(); fo.open("User\_file.dat",ios::out|ios::binary); ofstream f; f.open("Screen\_Info1.dat",ios::out|ios::binary); f.close(); f.open("Screen\_Info2.dat",ios::out|ios::binary); f.close(); f.open("Screen\_Info3.dat",ios::out|ios::binary); f.close(); f.open("Screen\_Info4.dat",ios::out|ios::binary); f.close();\*/

//-------------------------------------------------------------------------------------

--------------------------------------

system("cls"); int ch;

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

-------------------------------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\*\*\*\*\* Welcome To CINE\_CORNER \*\*\*\*\*";

cout<<"\n\n\t\t\t\t\t<<<PRESS ANY KEY TO CONTINUE>>>";



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

------------------------------------------------------------";

cin.get(); do

{

system("cls");

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

---------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\t<1> ADMIN MODE"; cout<<"\n\n\t\t\t\t\t\t<2> USER MODE"; cout<<"\n\n\t\t\t\t\t\t<3> EXIT";

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

---------------------------------------------------------------"; cout<<"\n SELECT A MODE : ";

cin>>ch;

system("cls"); switch(ch)

{

case 1:{ //admin mode

ifstream fin("AP.dat",ios::binary|ios::in); fin.read((char\*)&ADMIN,sizeof(ADMIN));

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

------------------------------------------------------------------------";

cout<<"\n\t\t\t\t\t\tEnter your Admin ID : "; ADMIN.get\_a\_id();

cin.get();

if(ADMIN.check\_pass()) //checking password for security concerns

{

fin.close();

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

---------------------------------------------------------------------------";

cin.get();

system("cls");

admin\_menu(); //for calling admin menu

}

break;



}

case 2:{ //user mode system("cls"); user\_menu(); break;

}

case 3 :{ //exit

break;

}

}

}while(ch!=3);

system("cls"); //clear screen

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

-------------------------------------------------------------------------------------";

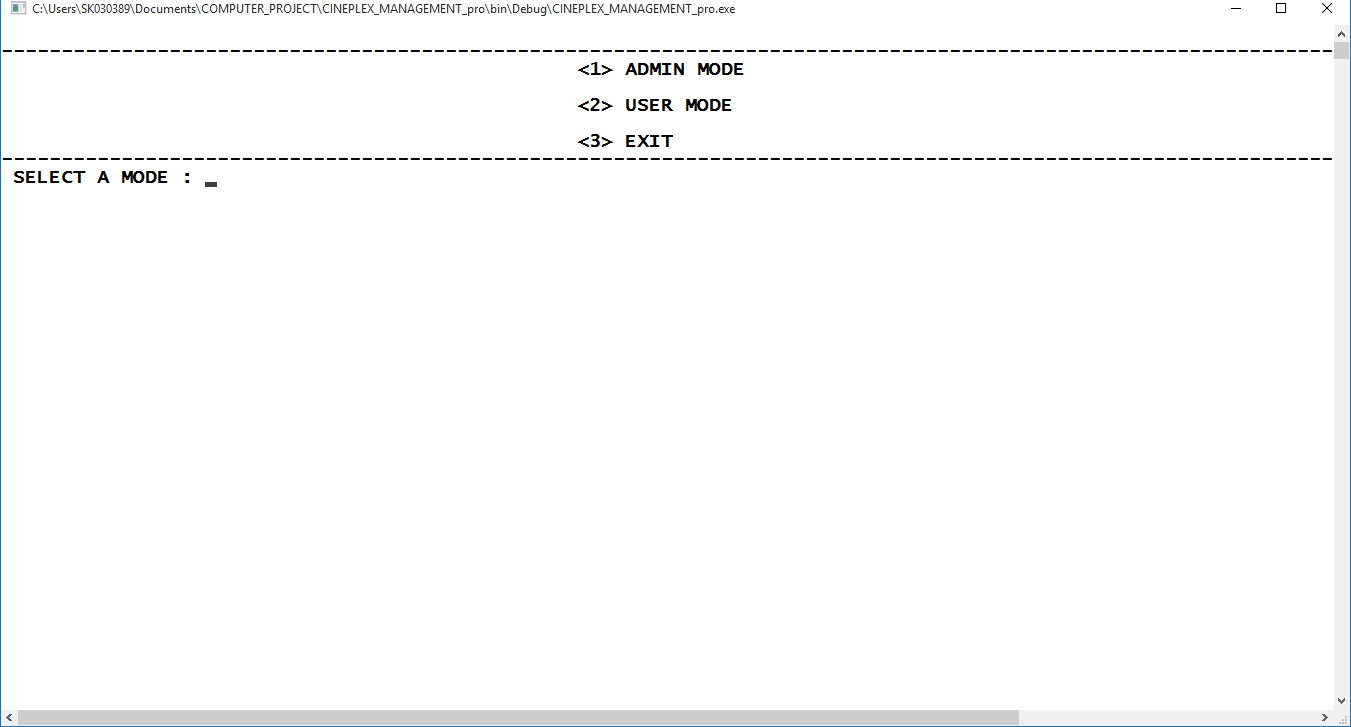
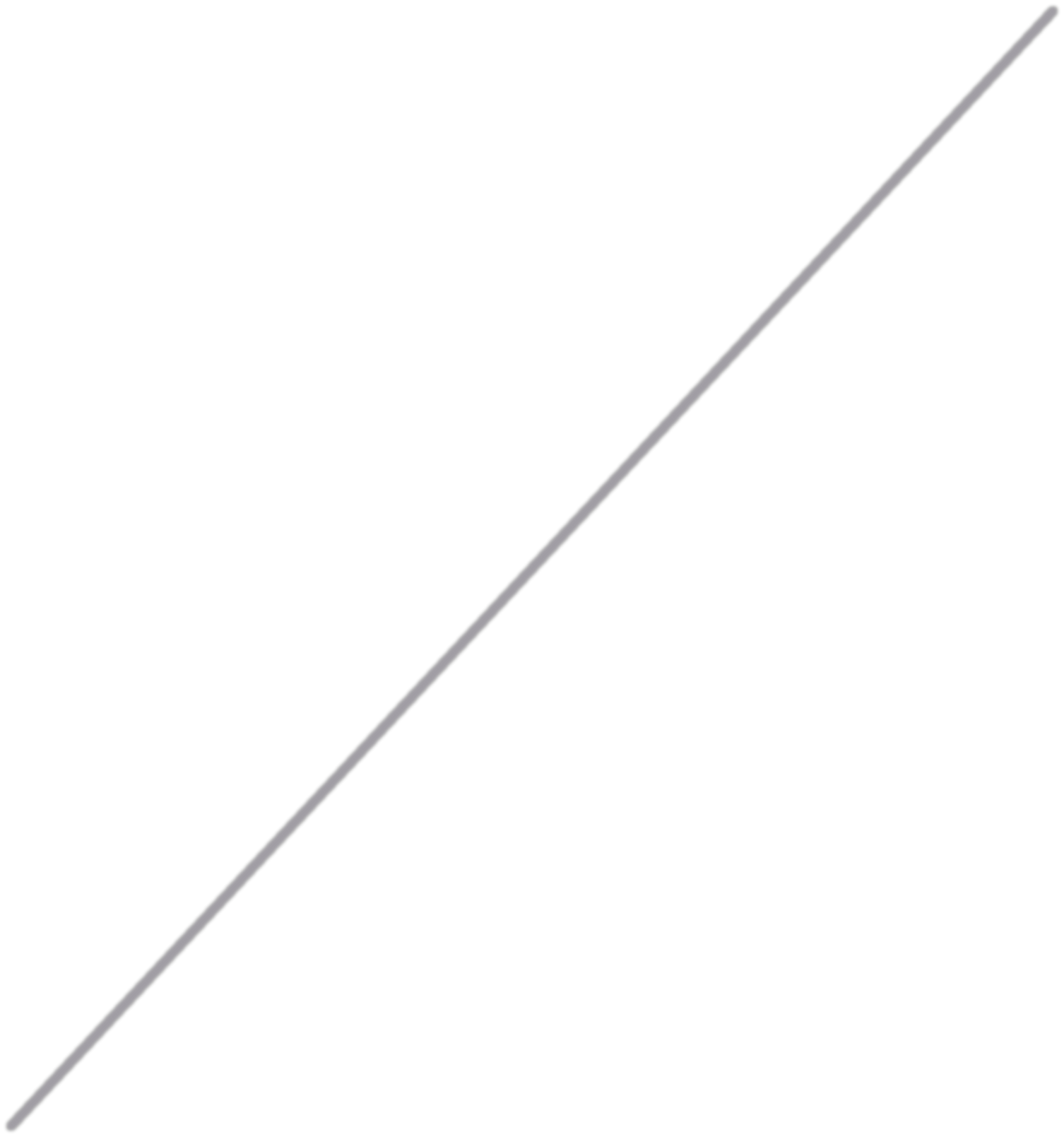
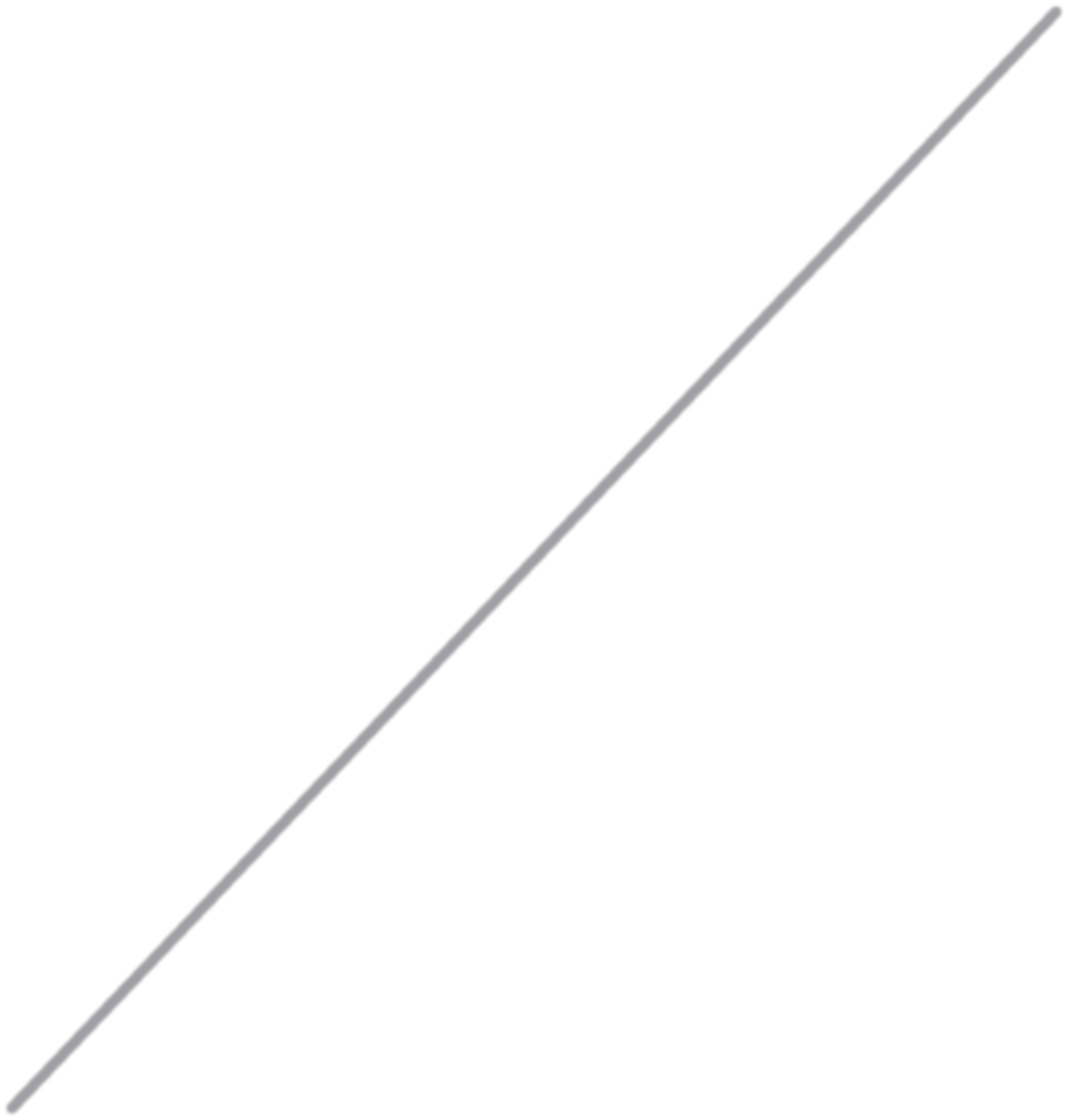
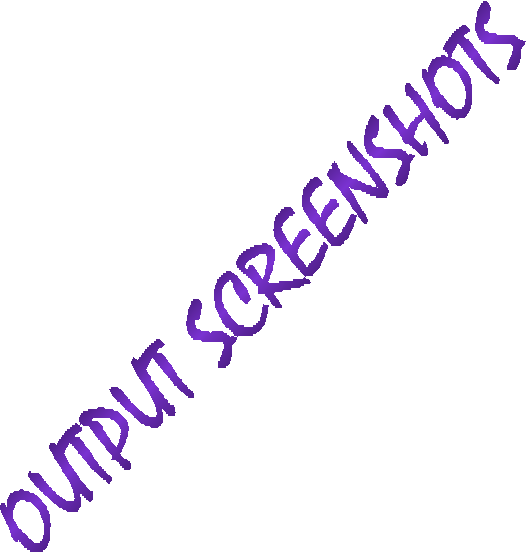
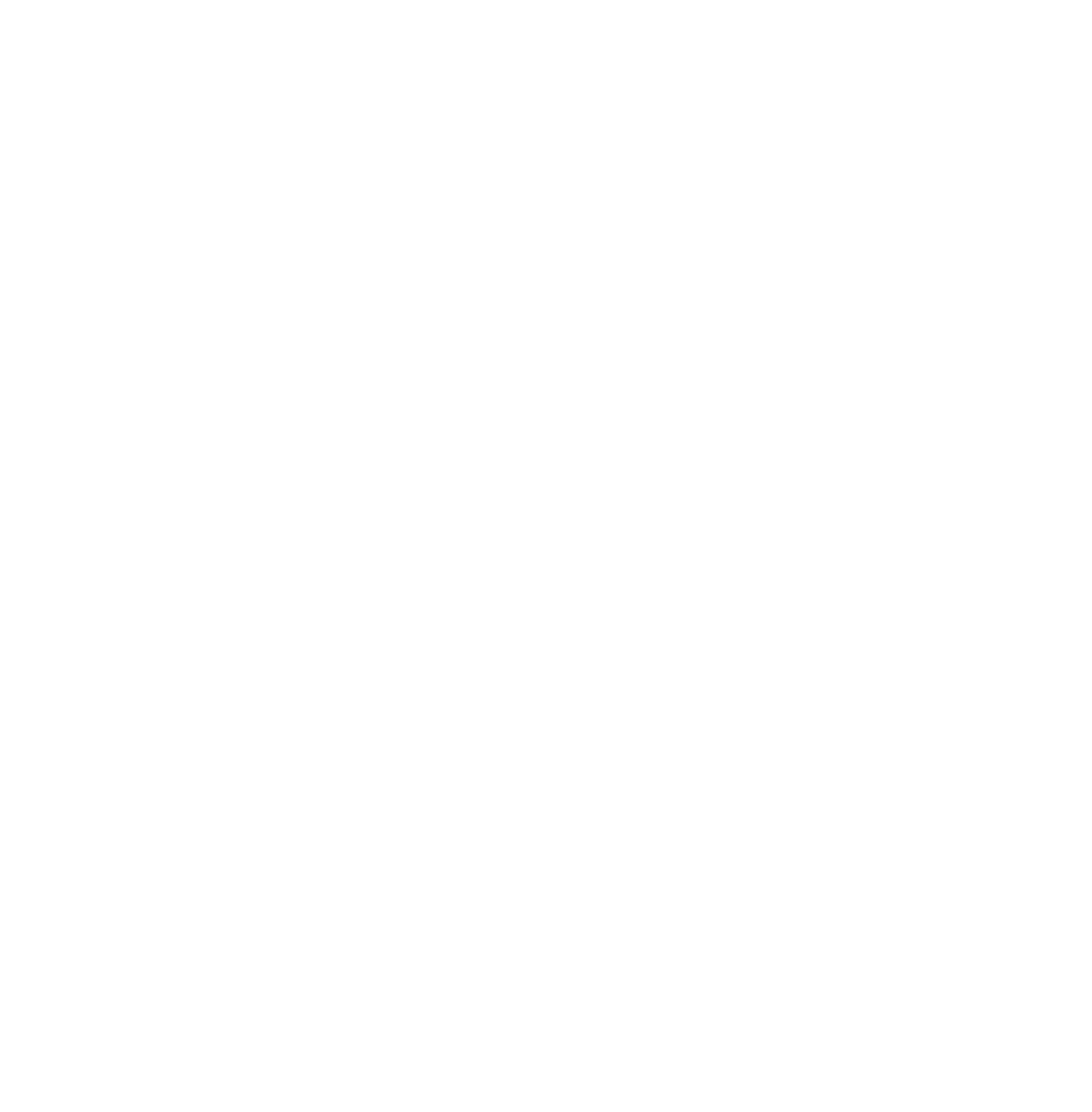
cout<<"\n\t\t\t\t\t\t\*\*\*\*\* THANK YOU \*\*\*\*\*";

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

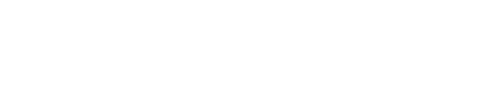
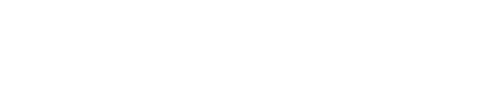
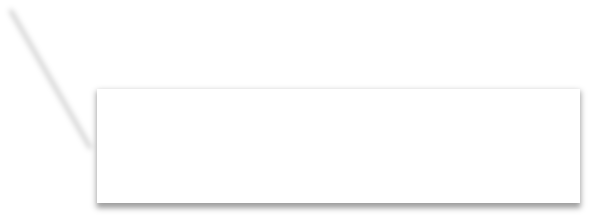
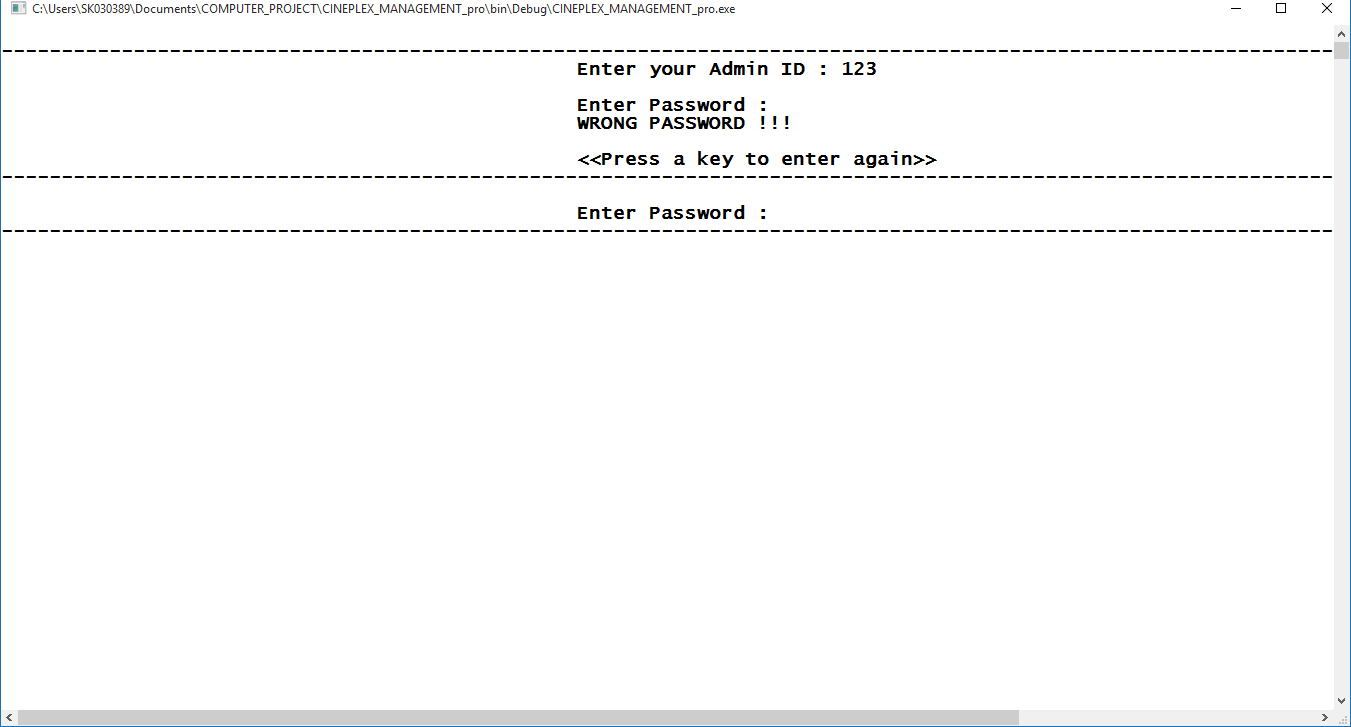
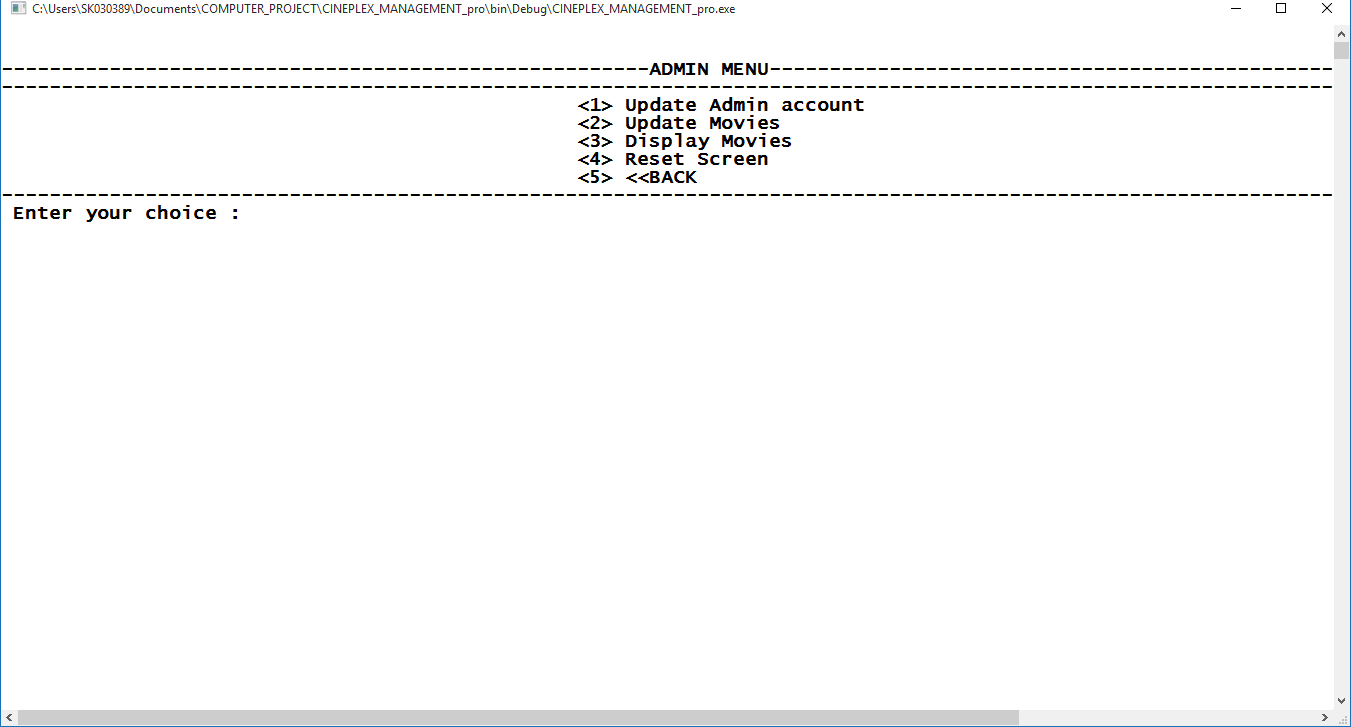
------------------------------------------------------------\n\n\n\n\n\n";

cin.get(); //for pressing enter key return 0;

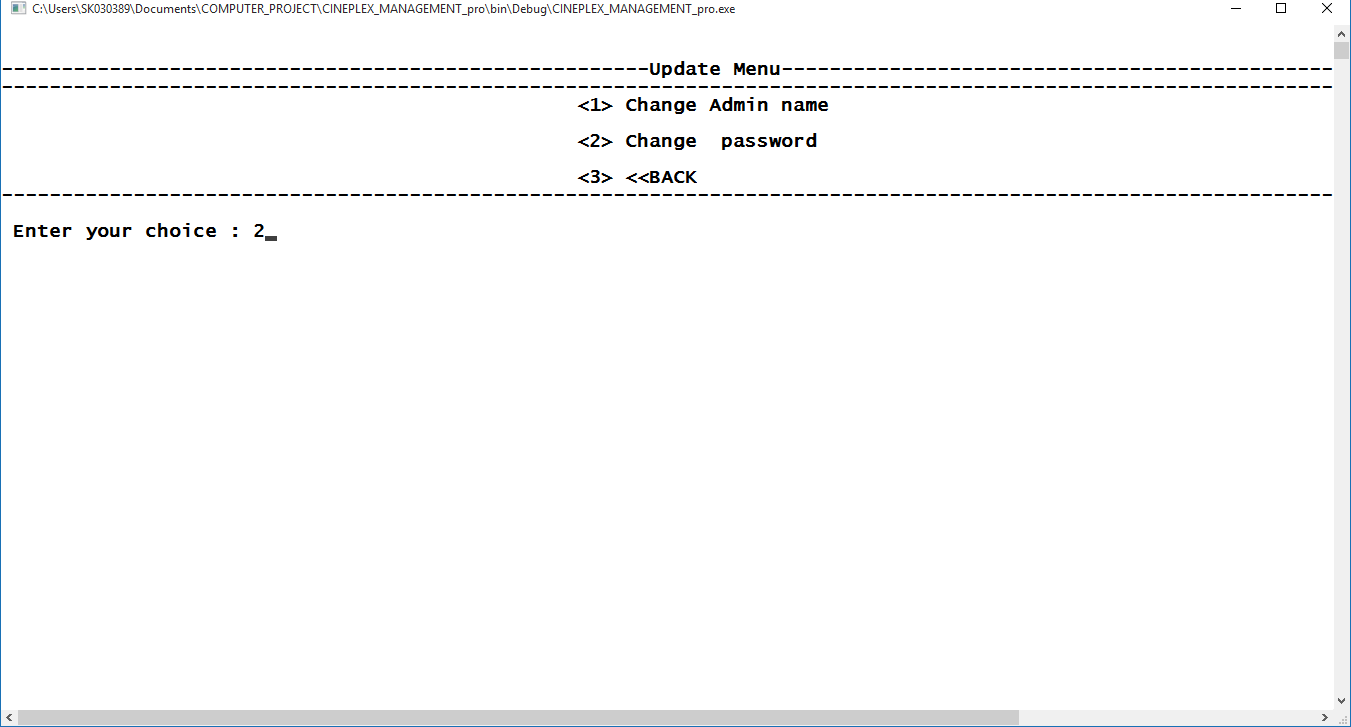
}



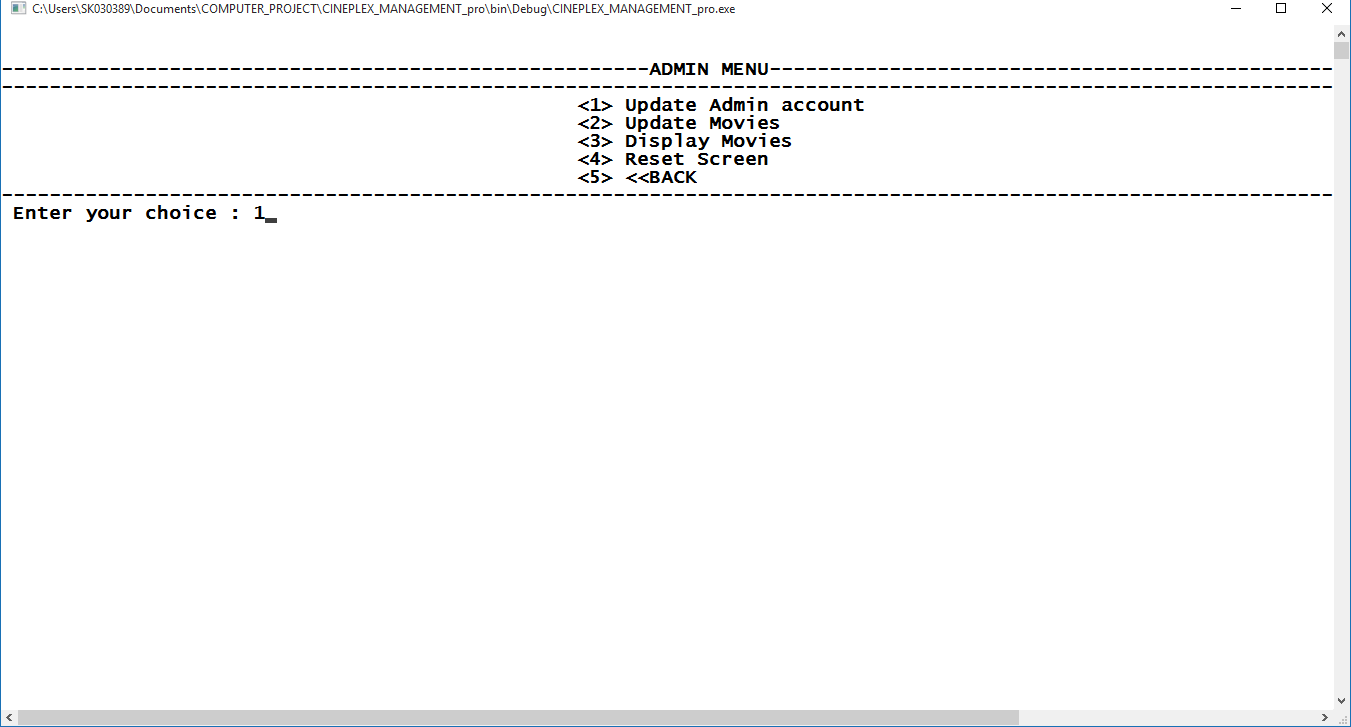
**ADMIN MODE**

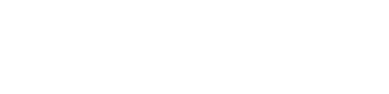
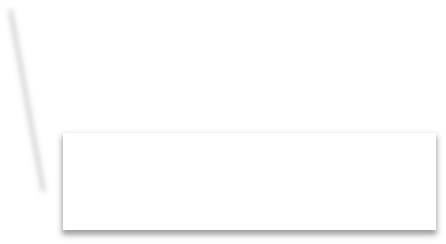
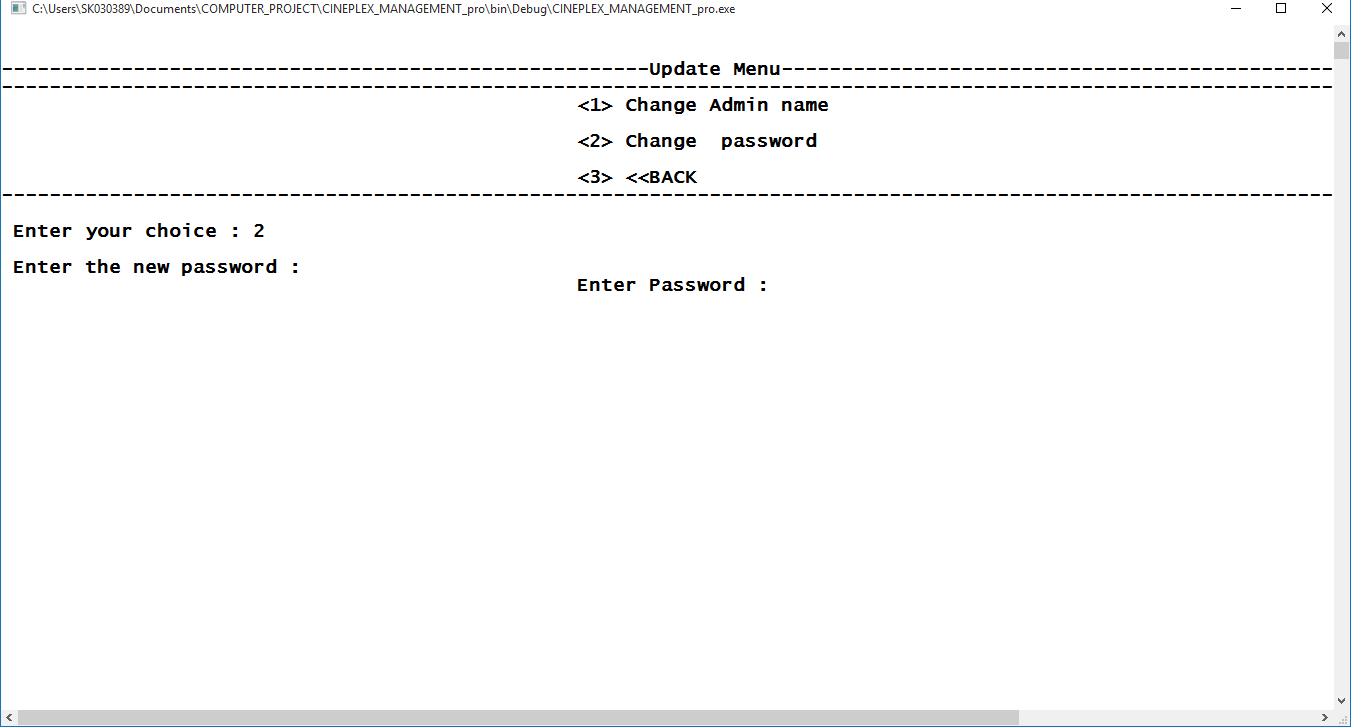
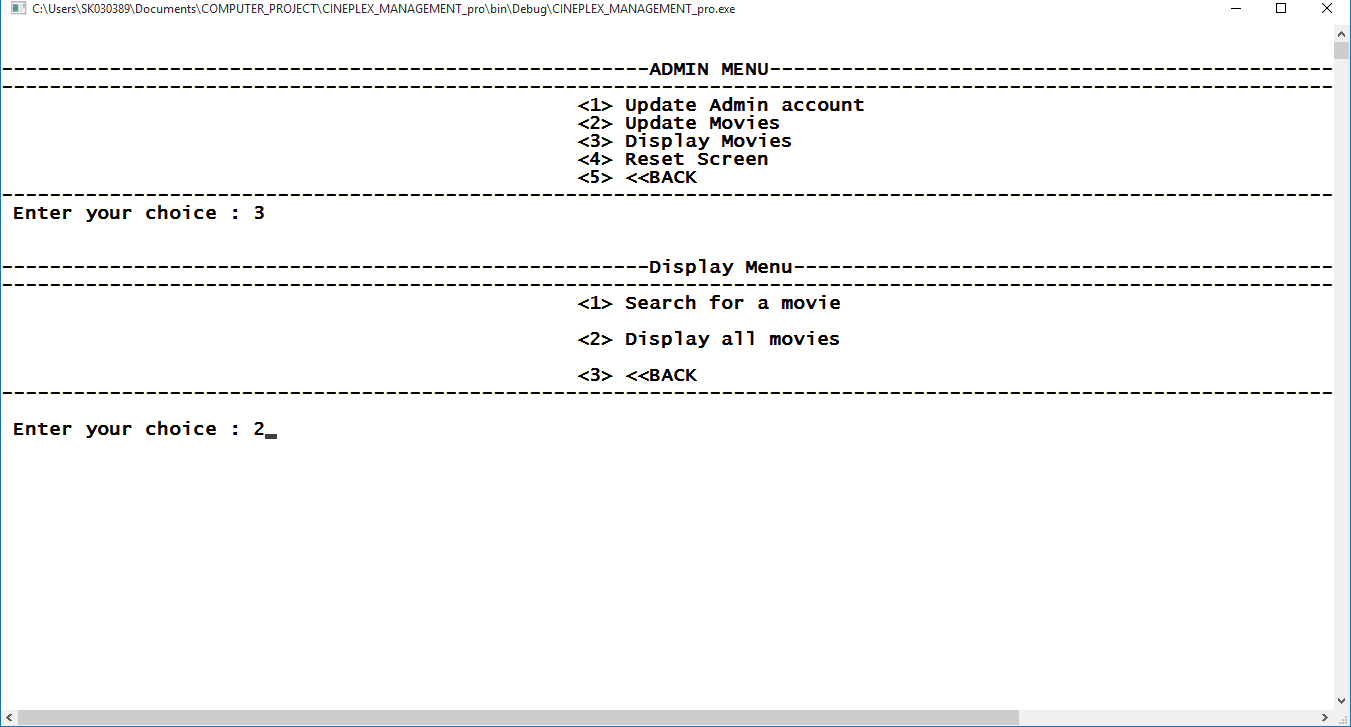


* ADMIN mode is password protected



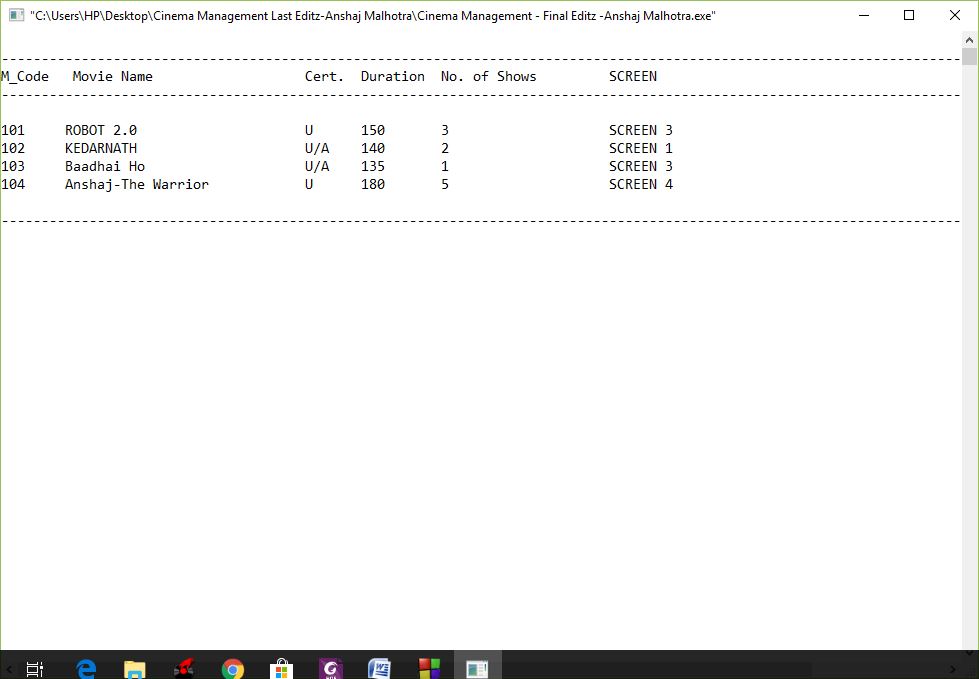
#### Changing Admin Password





Entering new password

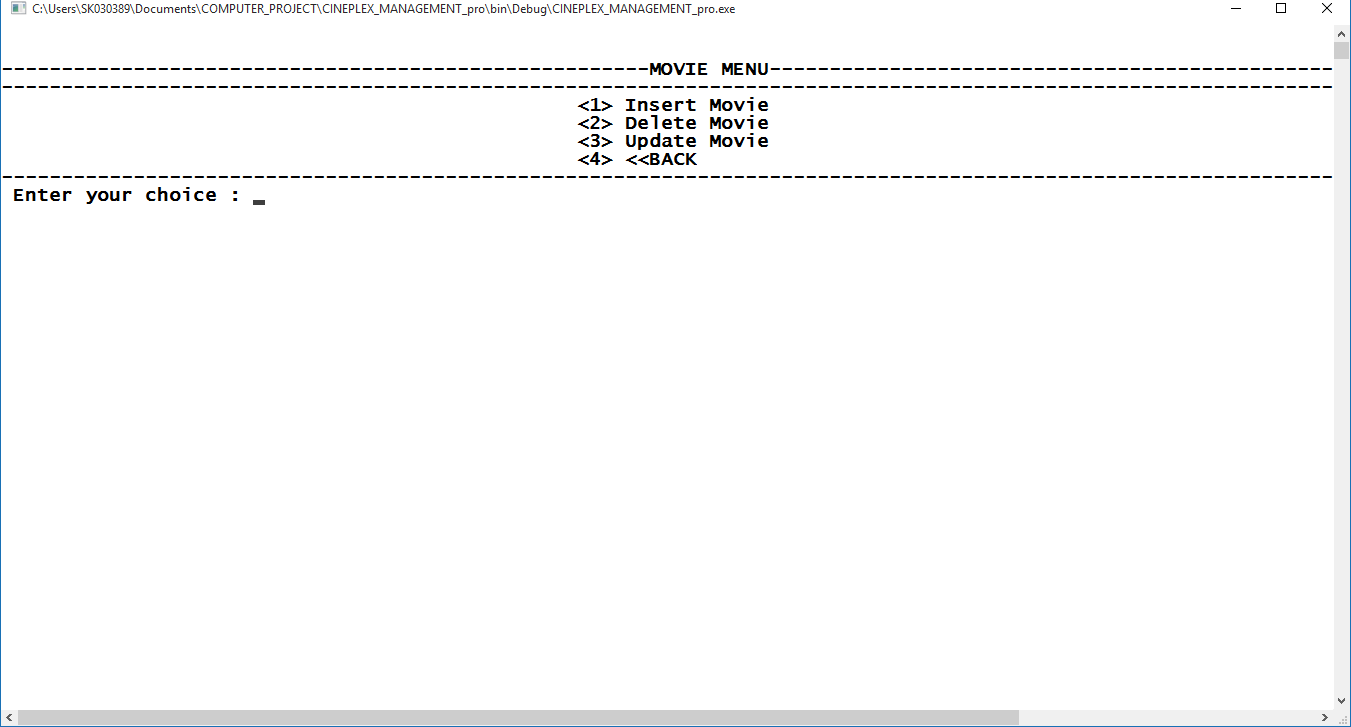
**DISPLAYING MOVIES**



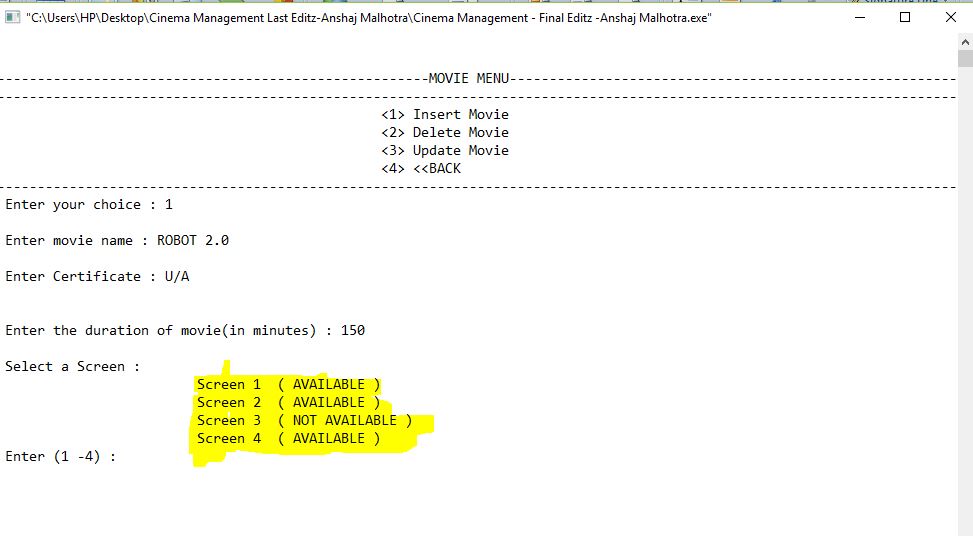
* + - There are 4 screens

#### Inserting Movie

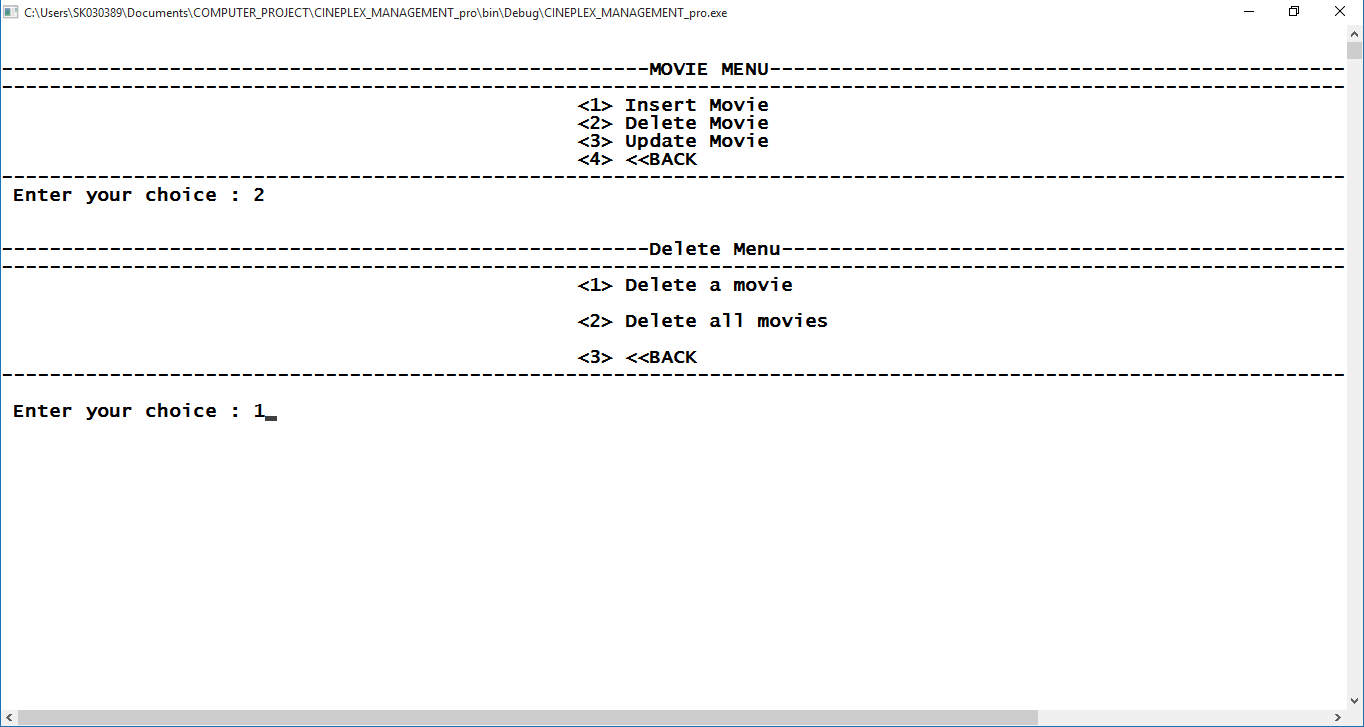
* + - Each screen has 4 slots – Morning, Afternoon, Evening & night

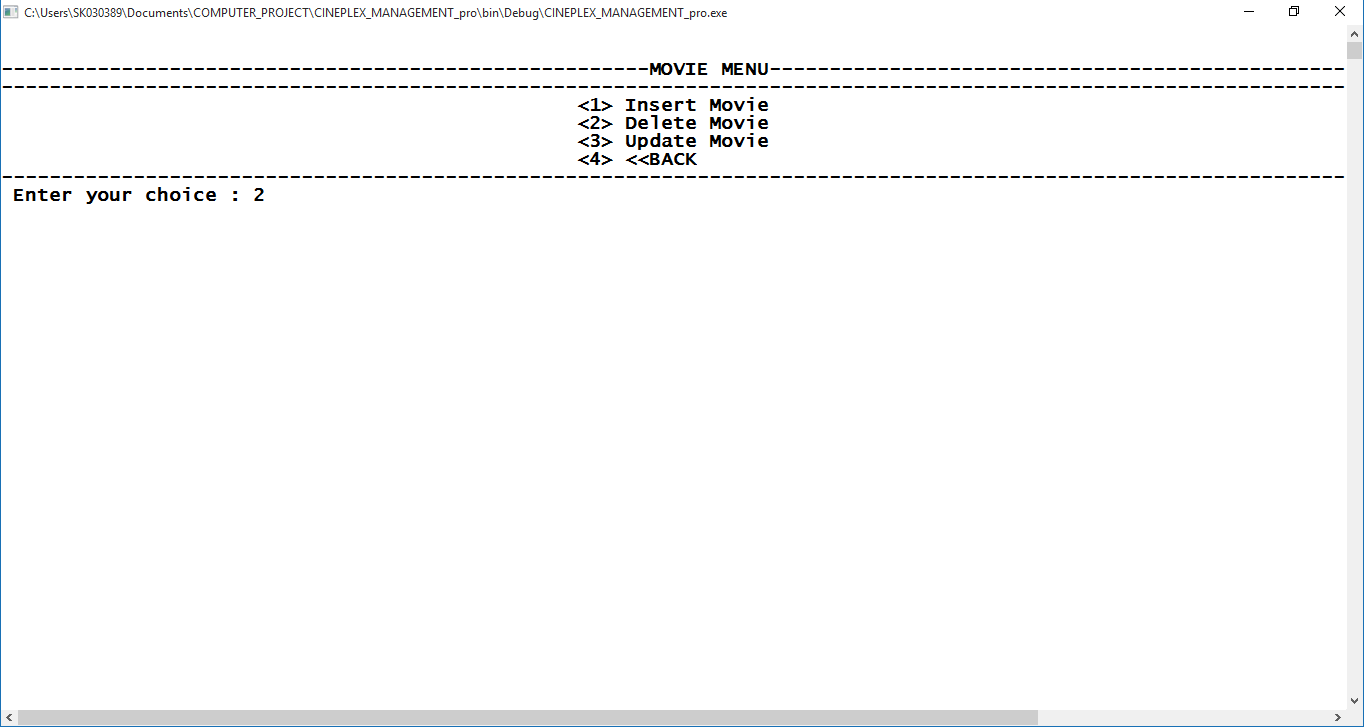


* + - The program checks if a screen has some slots left

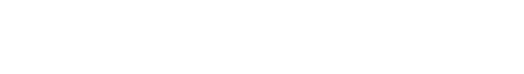
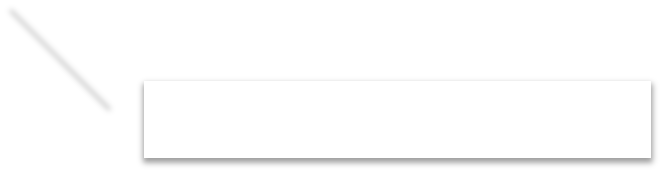
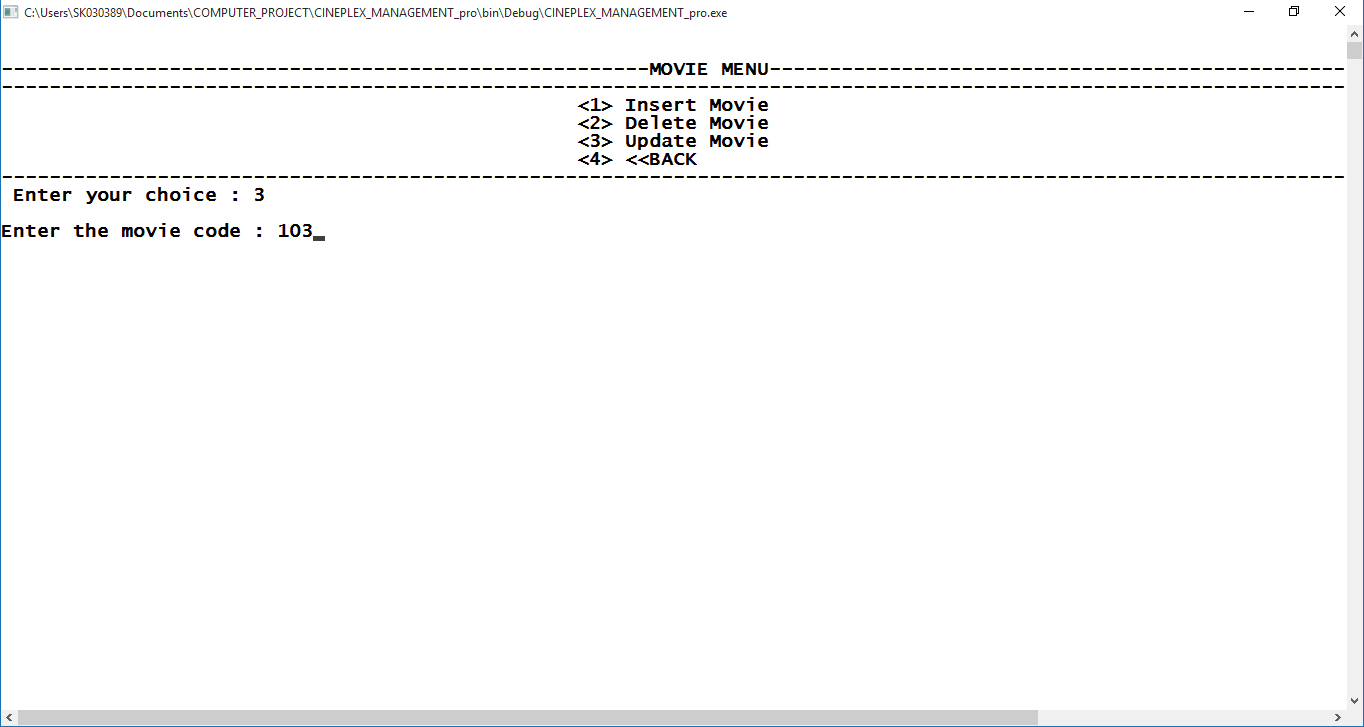


**Deleting a movie**



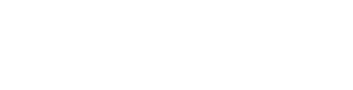
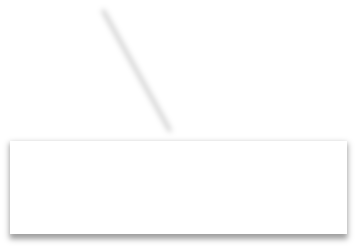
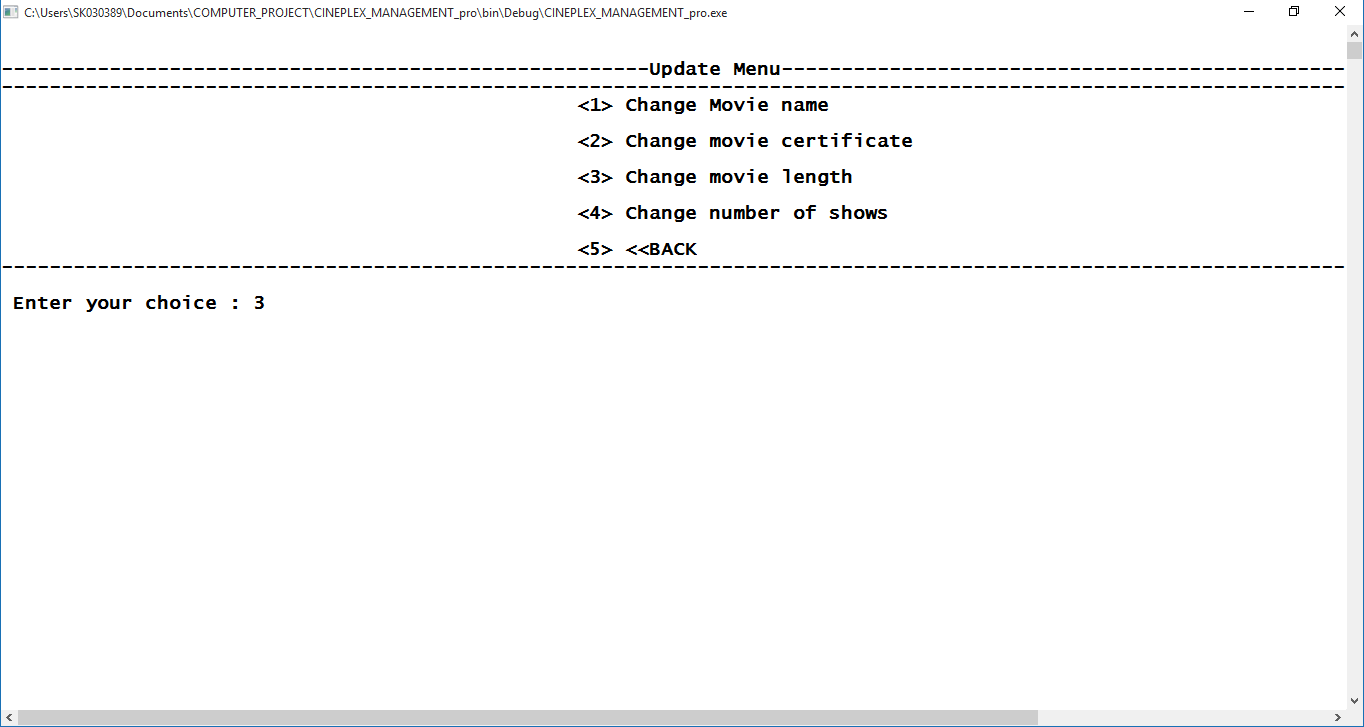


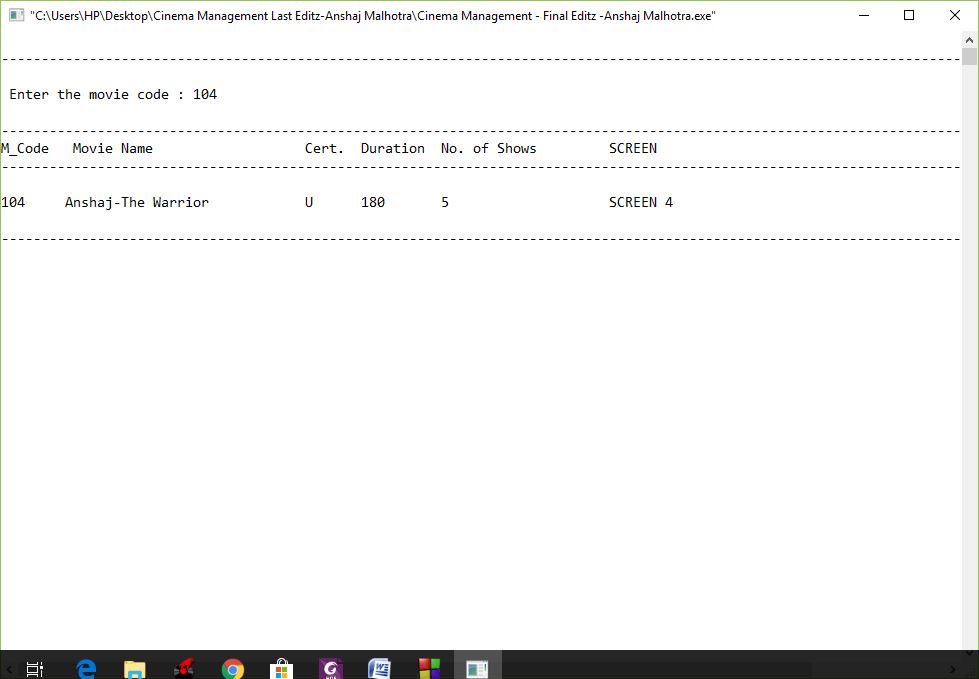




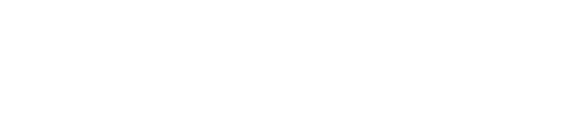
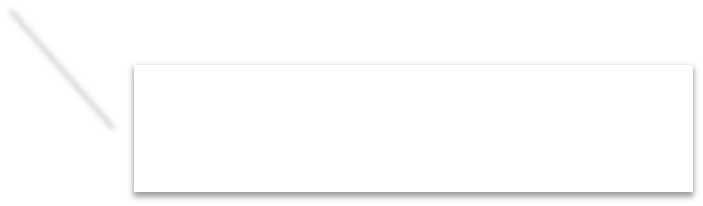
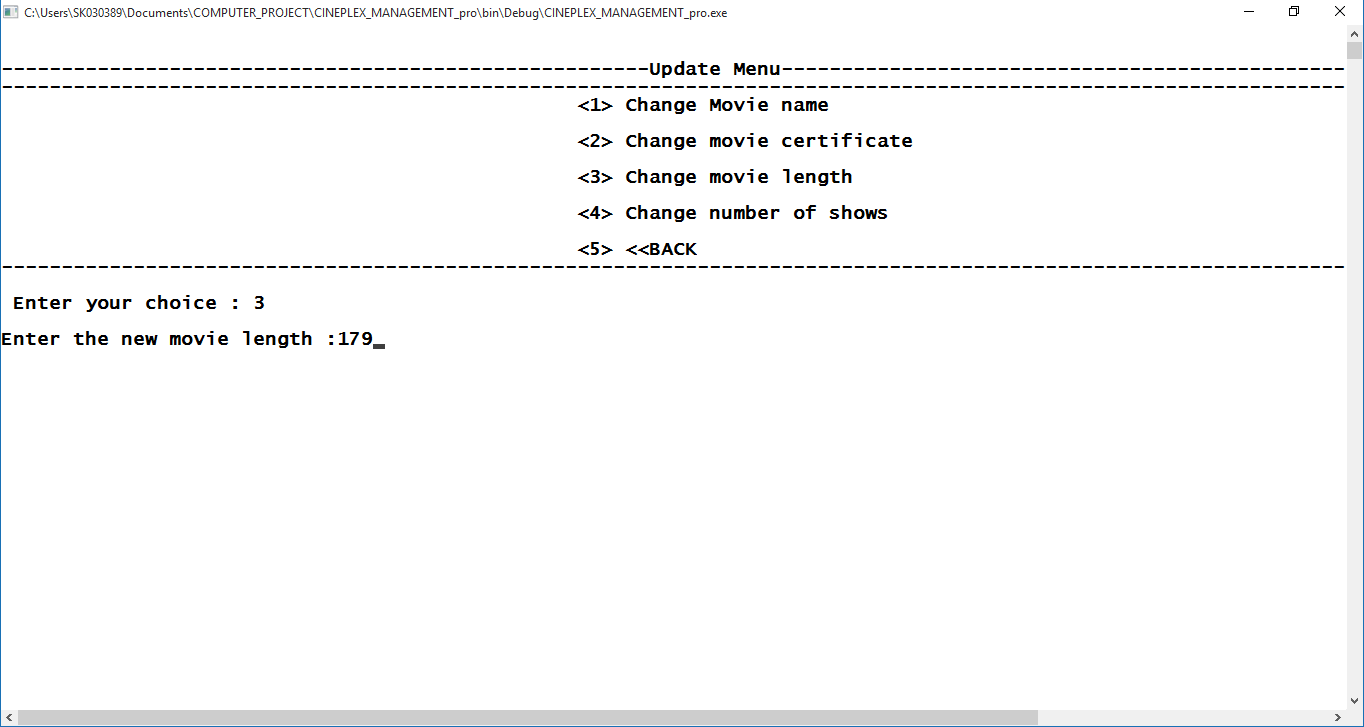
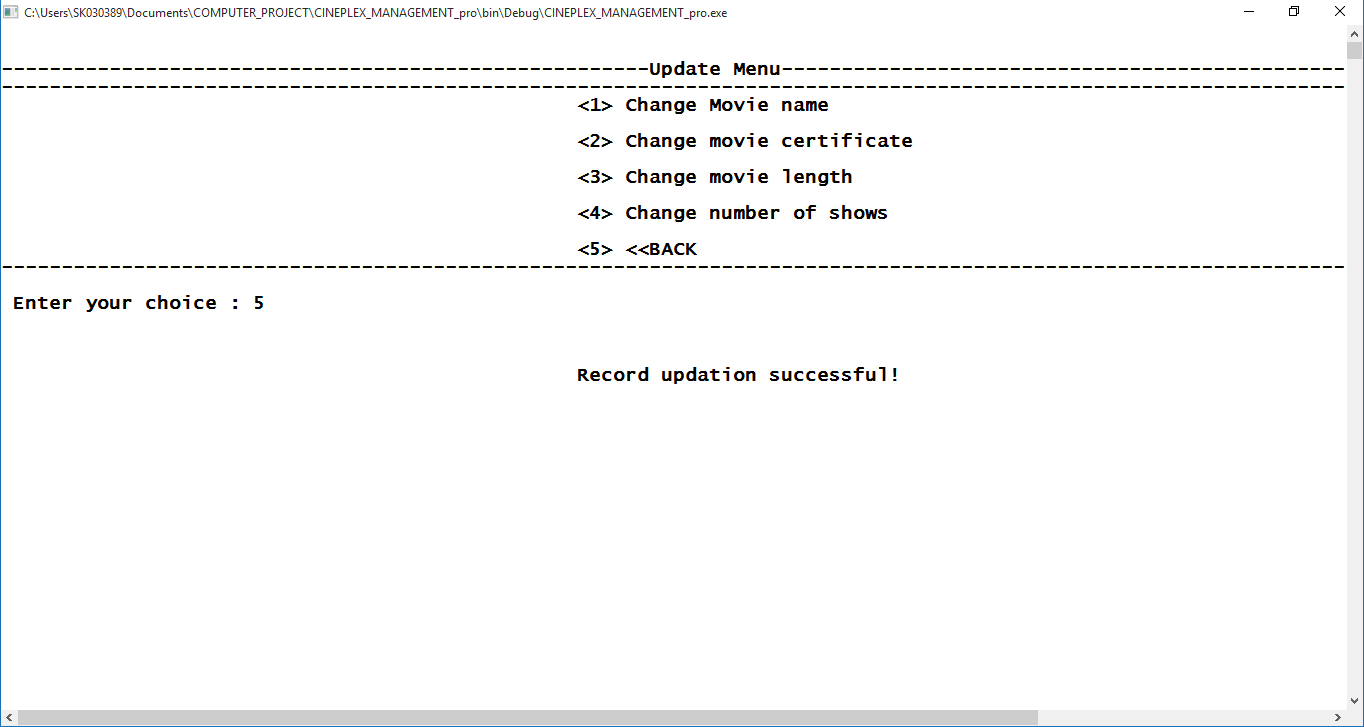
**Update Movie**

Searching a movie

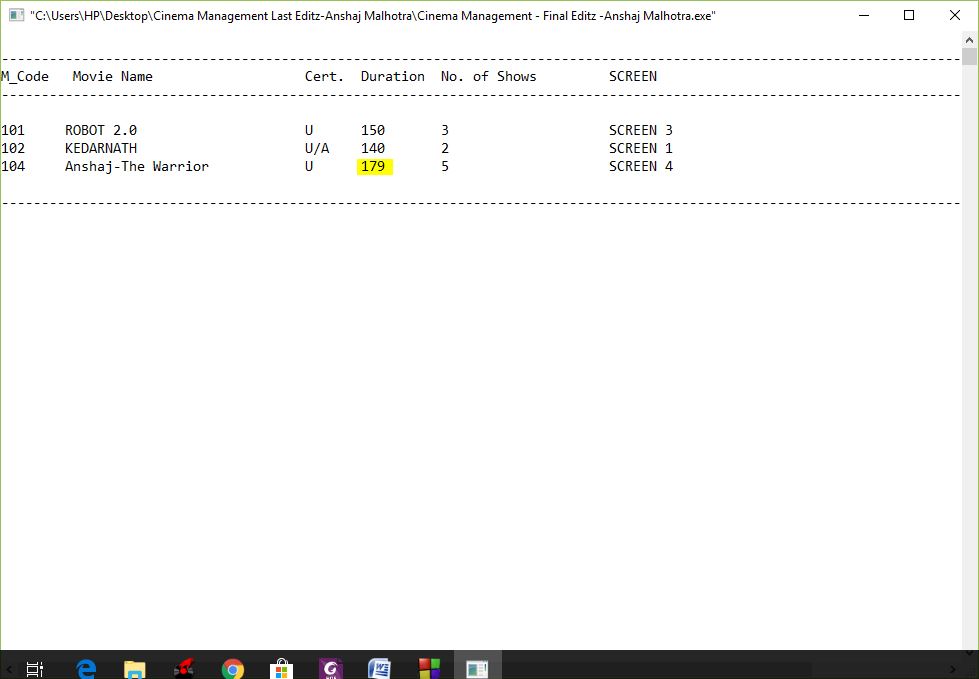




Update Menu

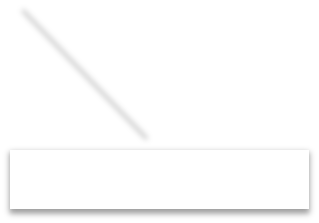
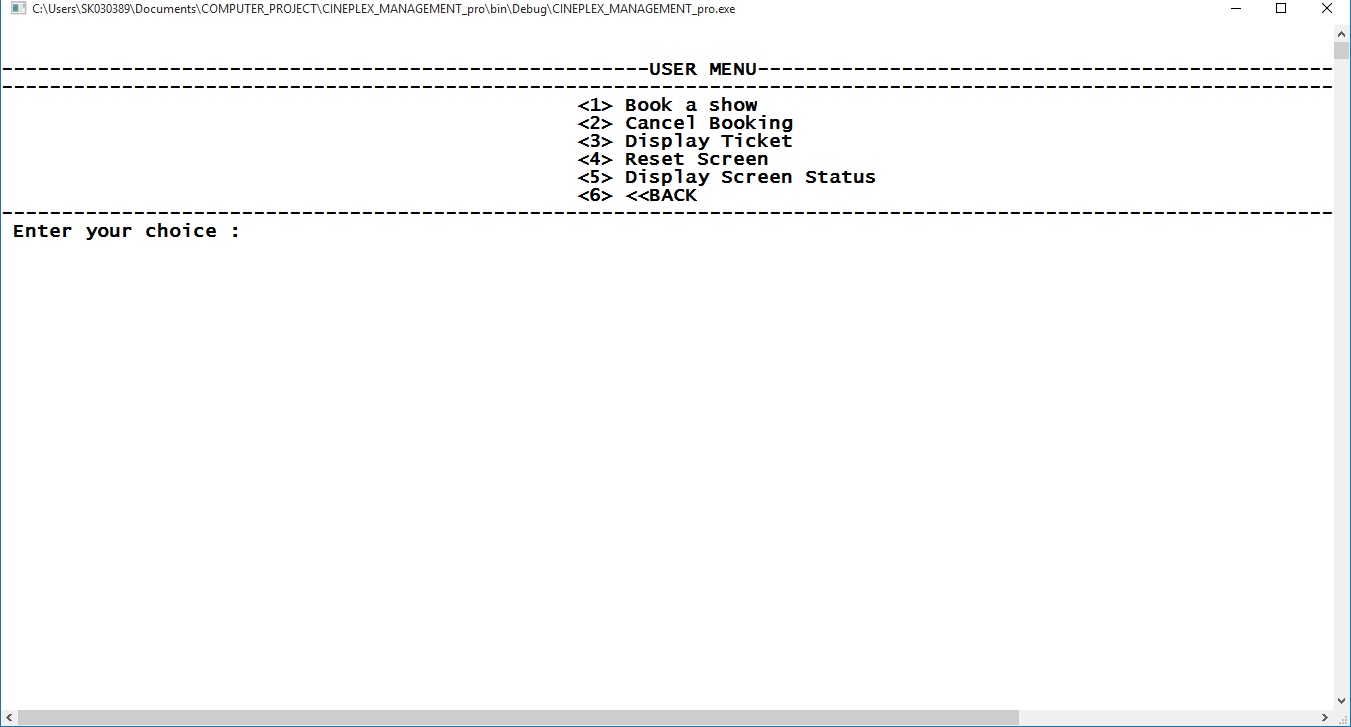


Entering new value

**Displaying updated record**

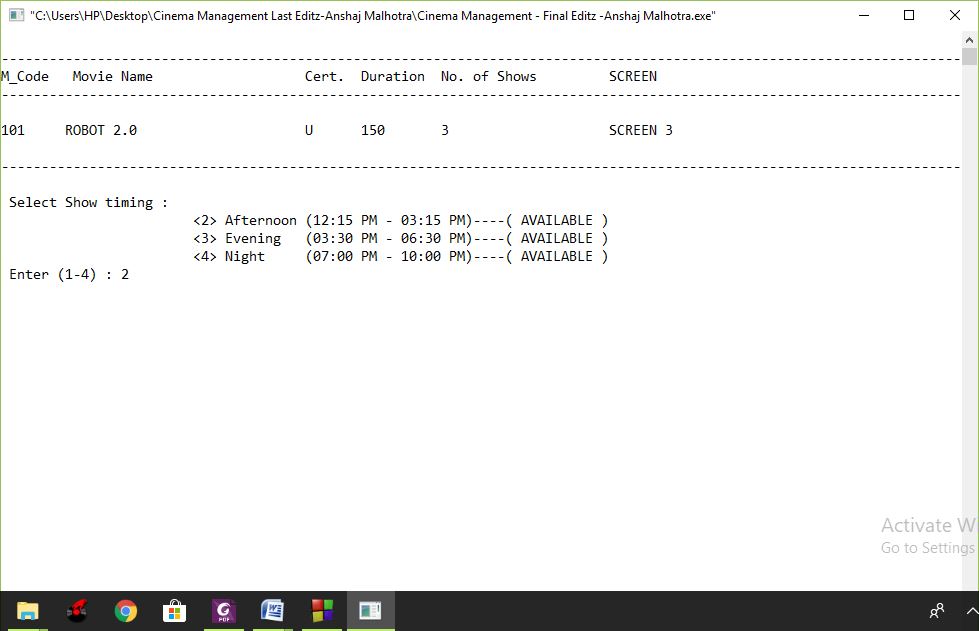
Movie length changed

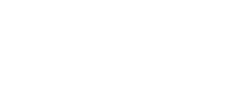
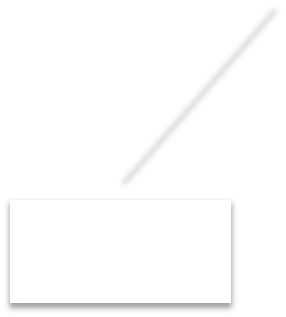
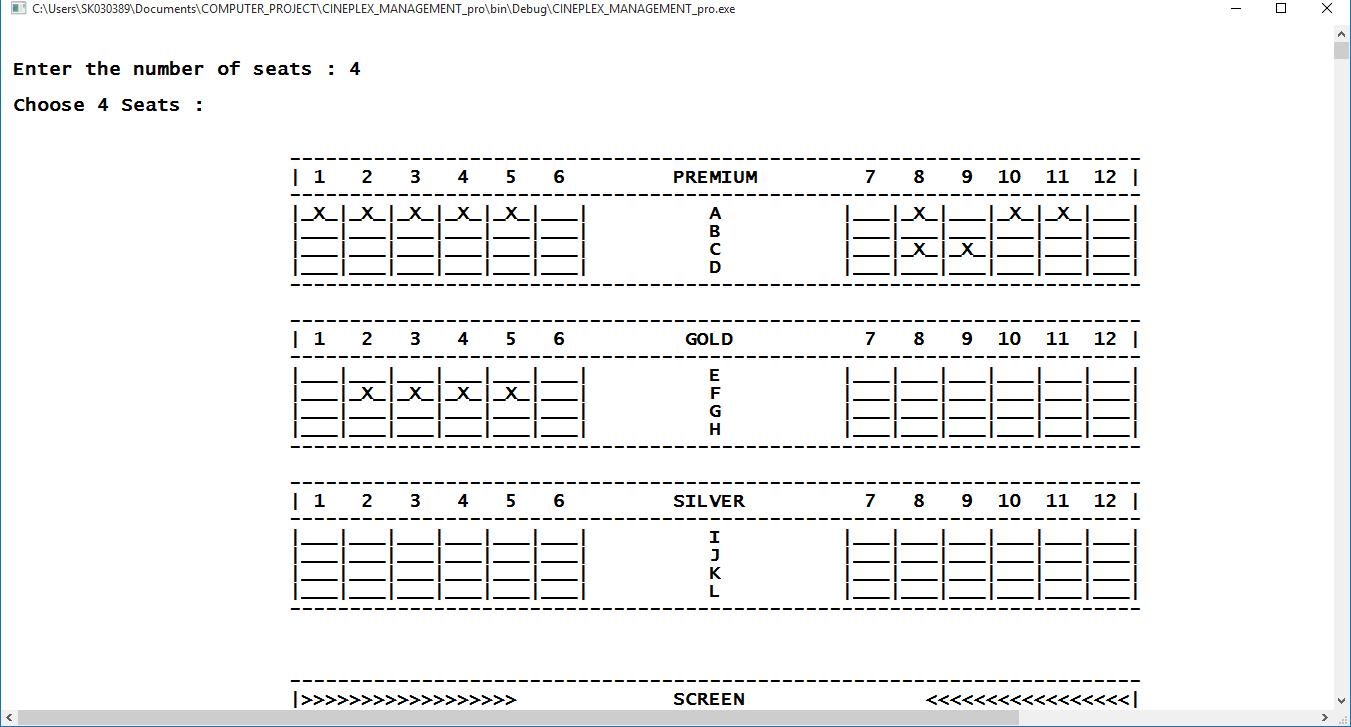
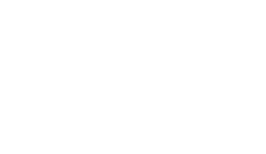
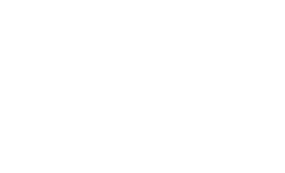
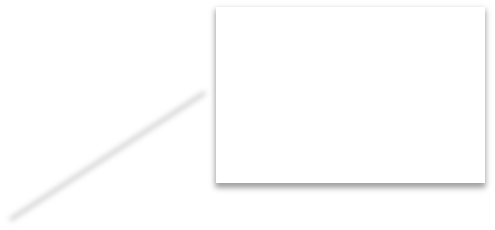
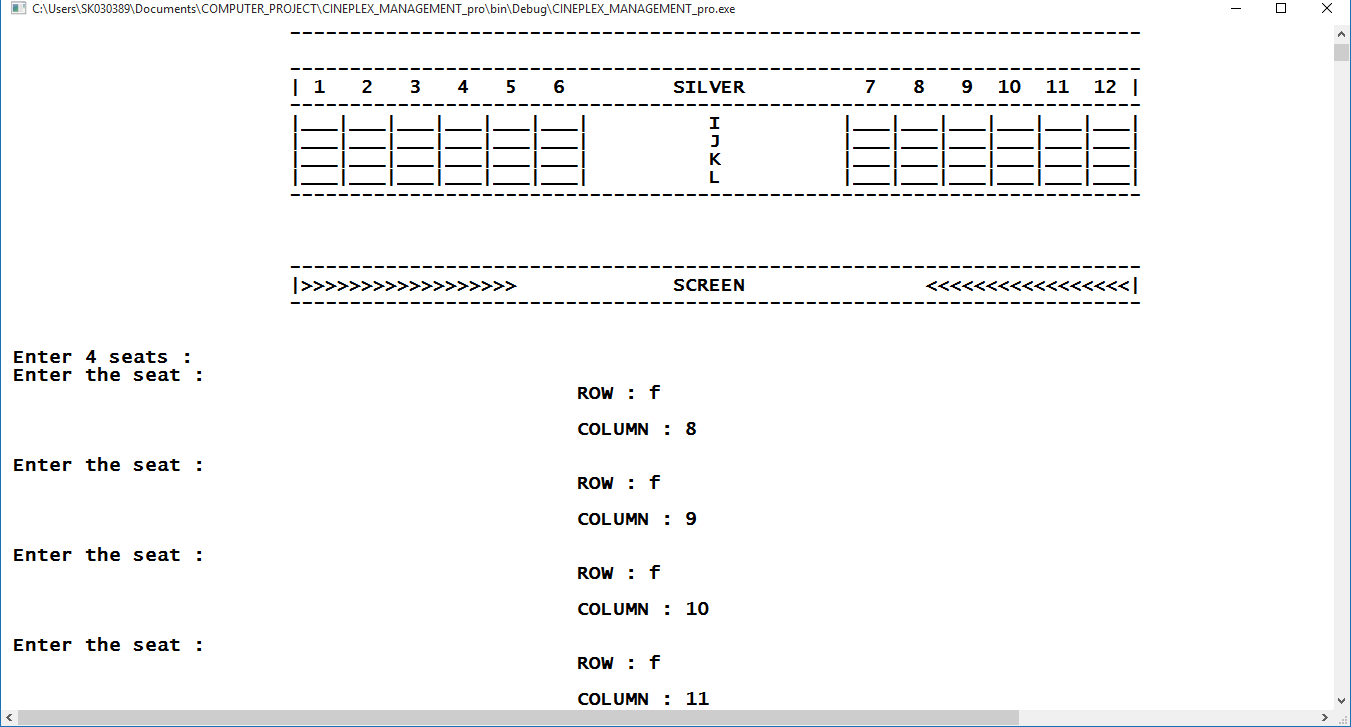
**User Mode**



* User Menu

**Booking a show**

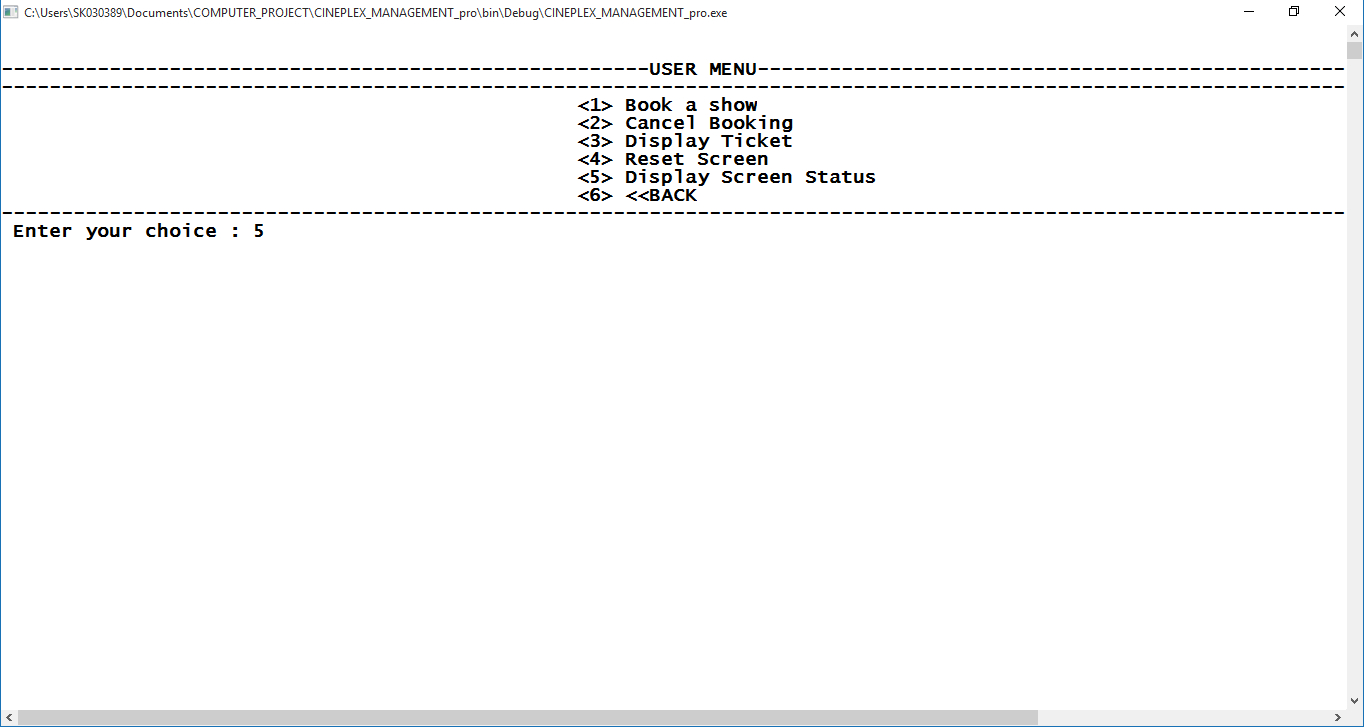
****



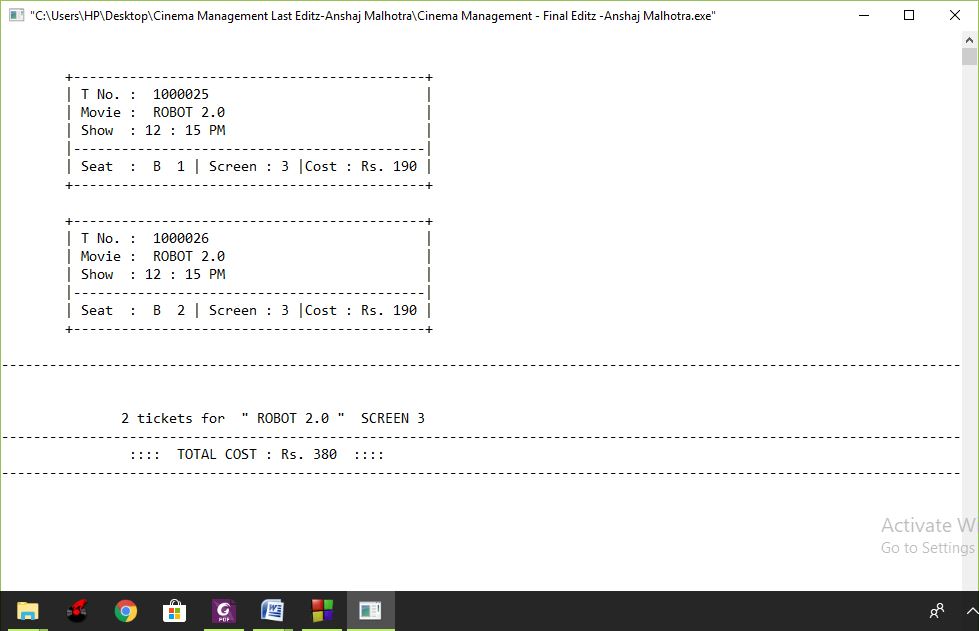
‘X’ seats are

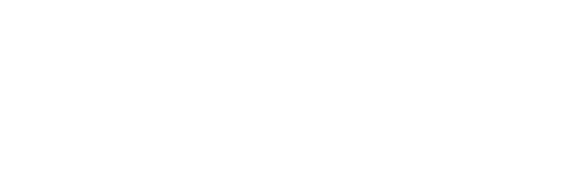
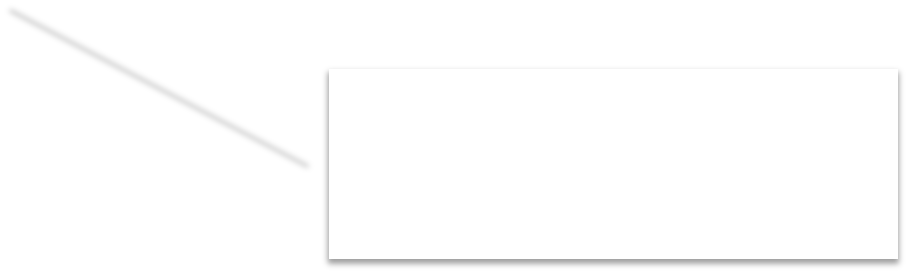
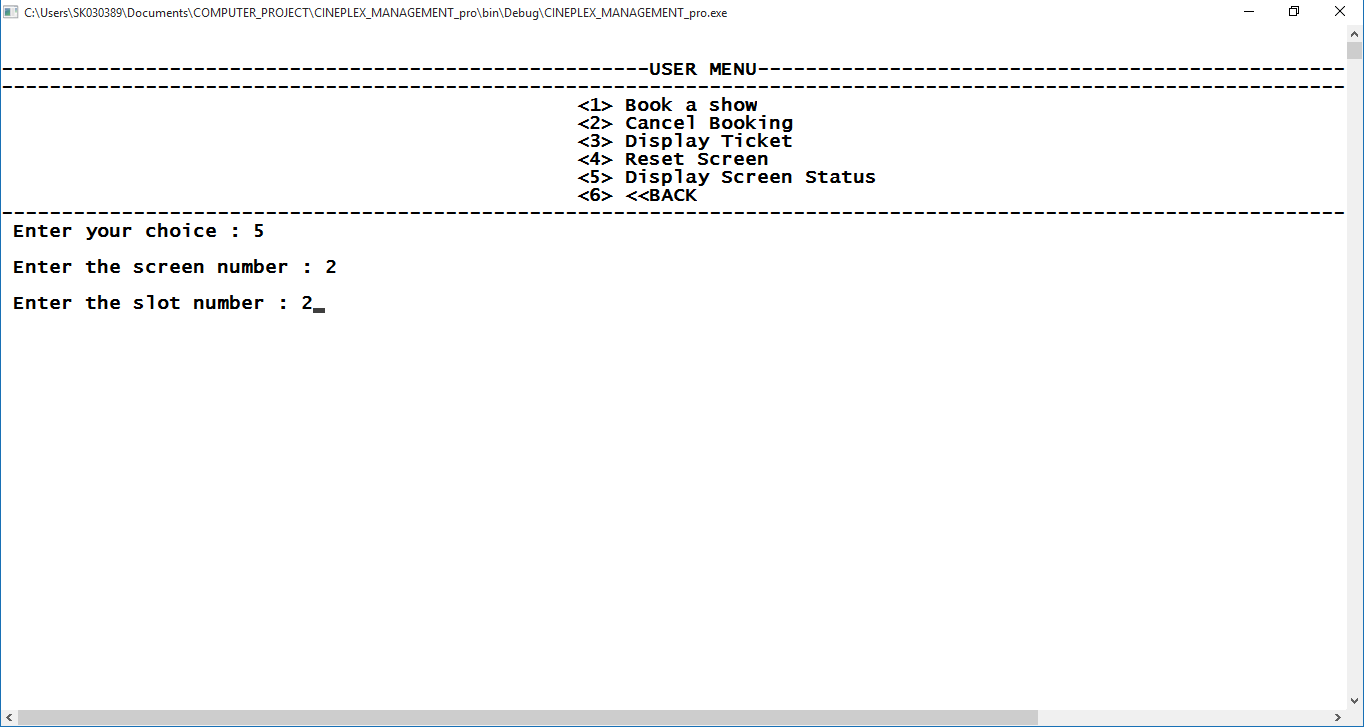
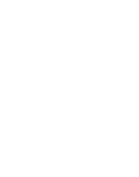
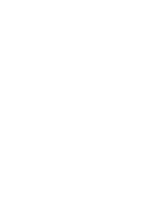
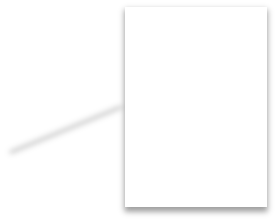
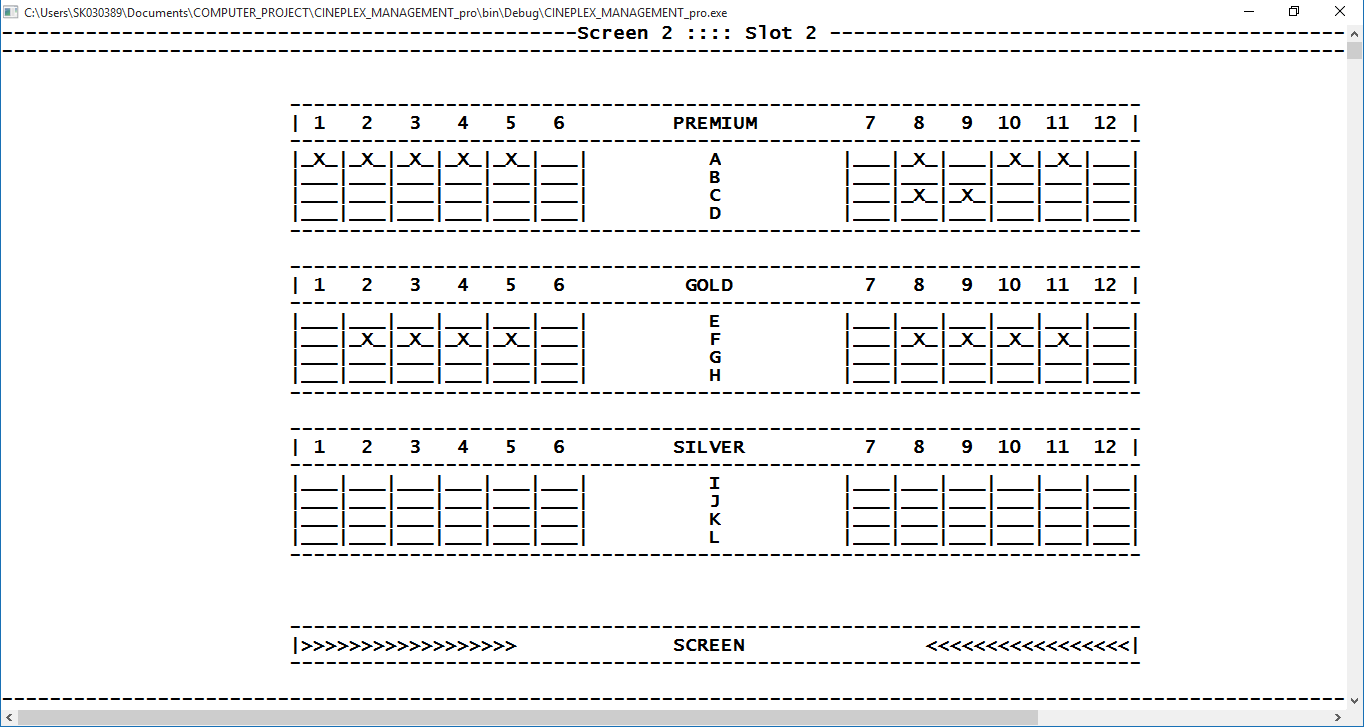
booked

Entering seats



**Displaying Screen Status**

****



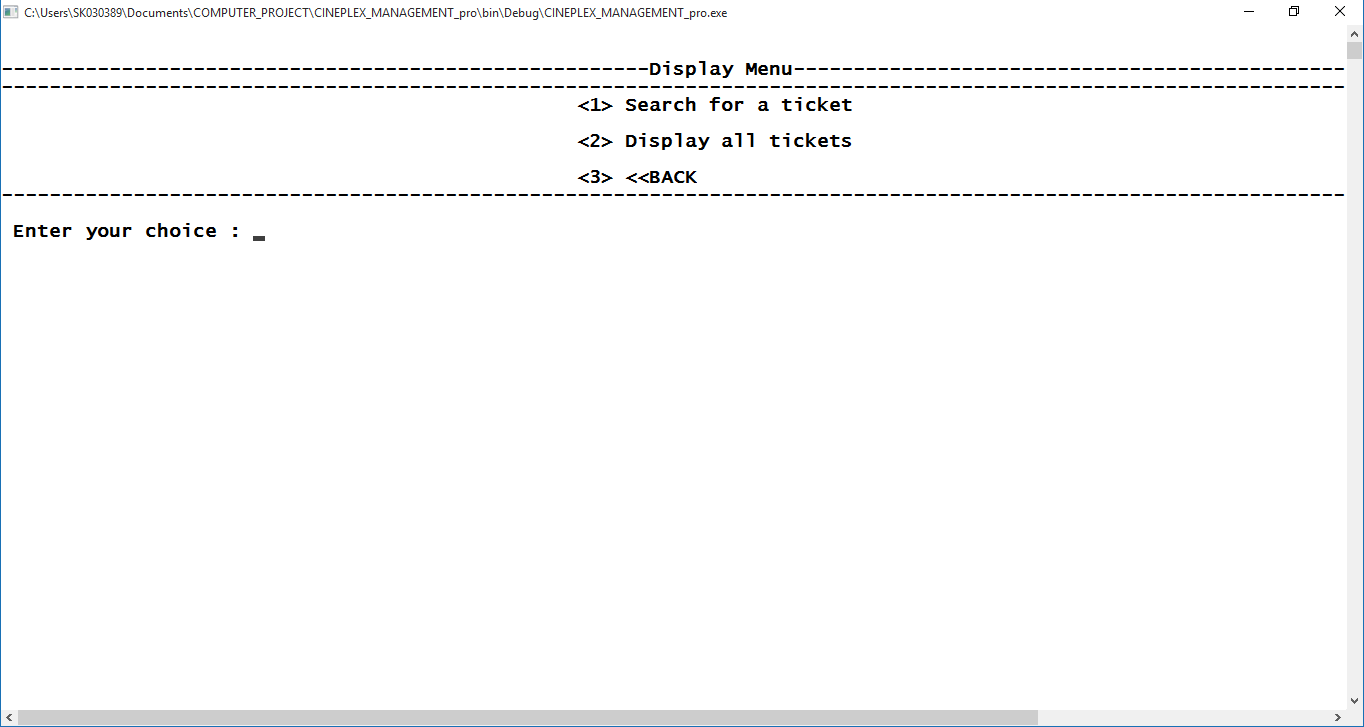
Enter screen & slot

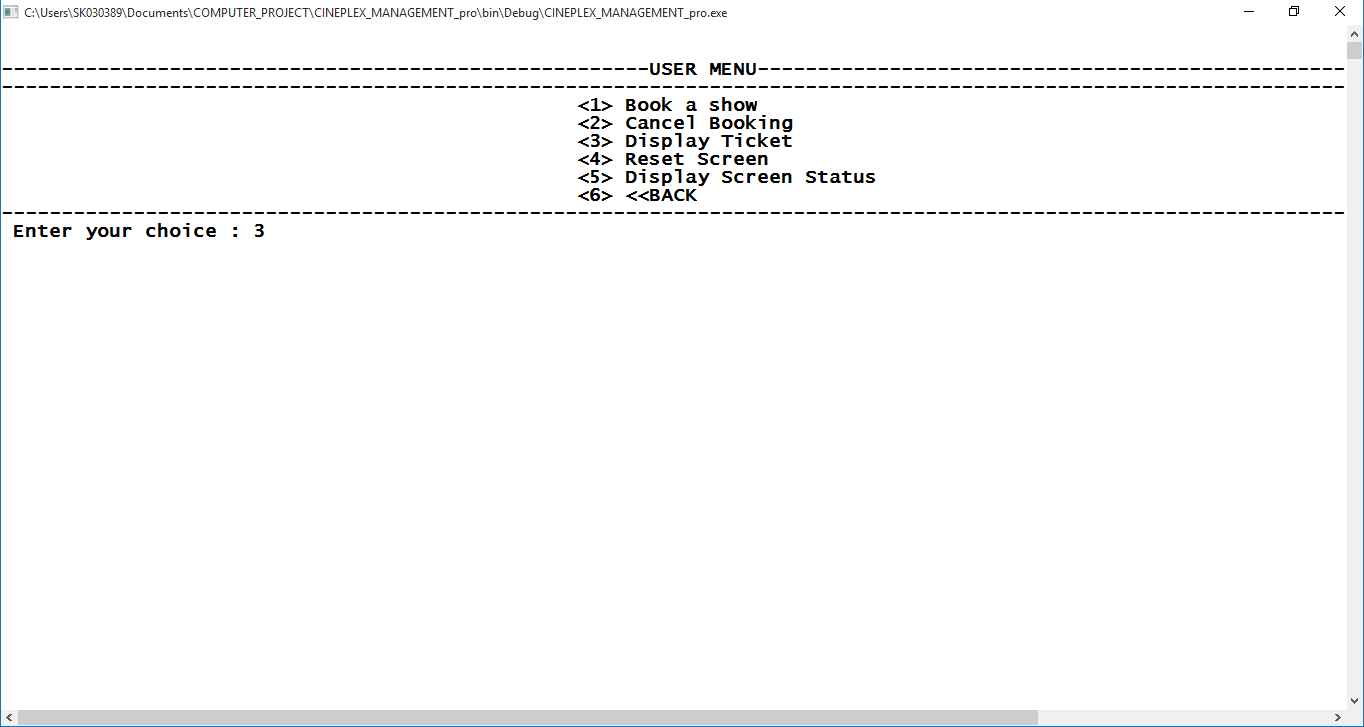
no.

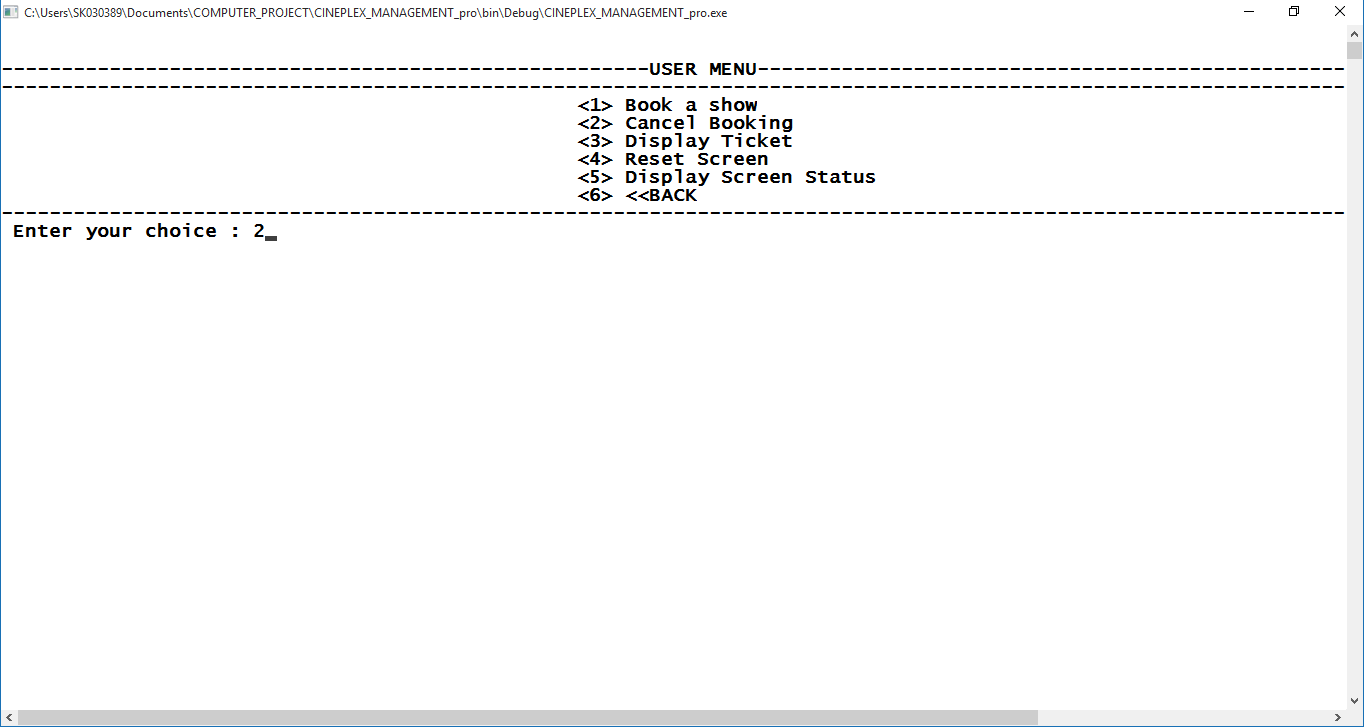
F8,F9, F10,F11

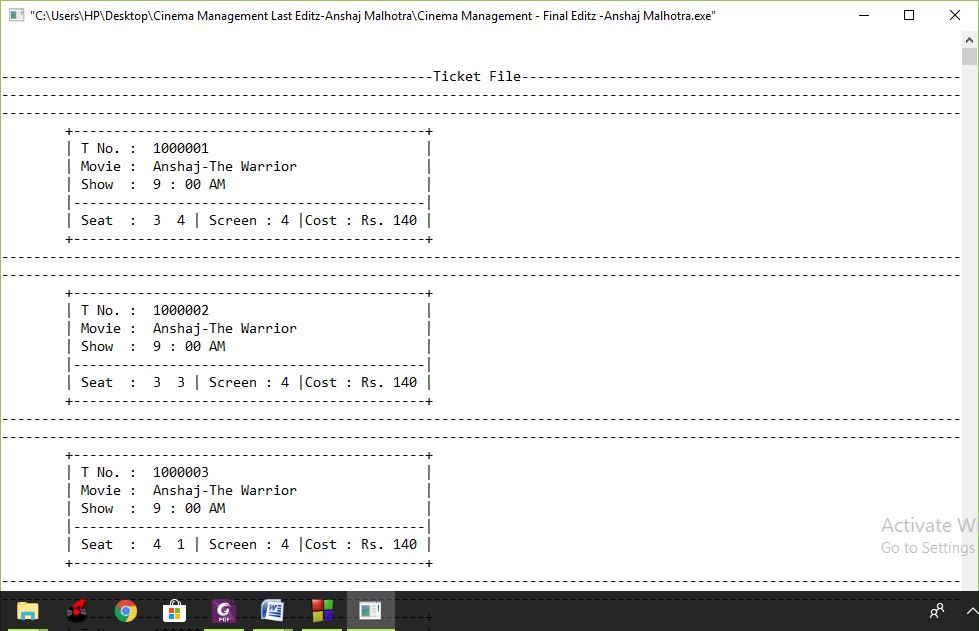
booked

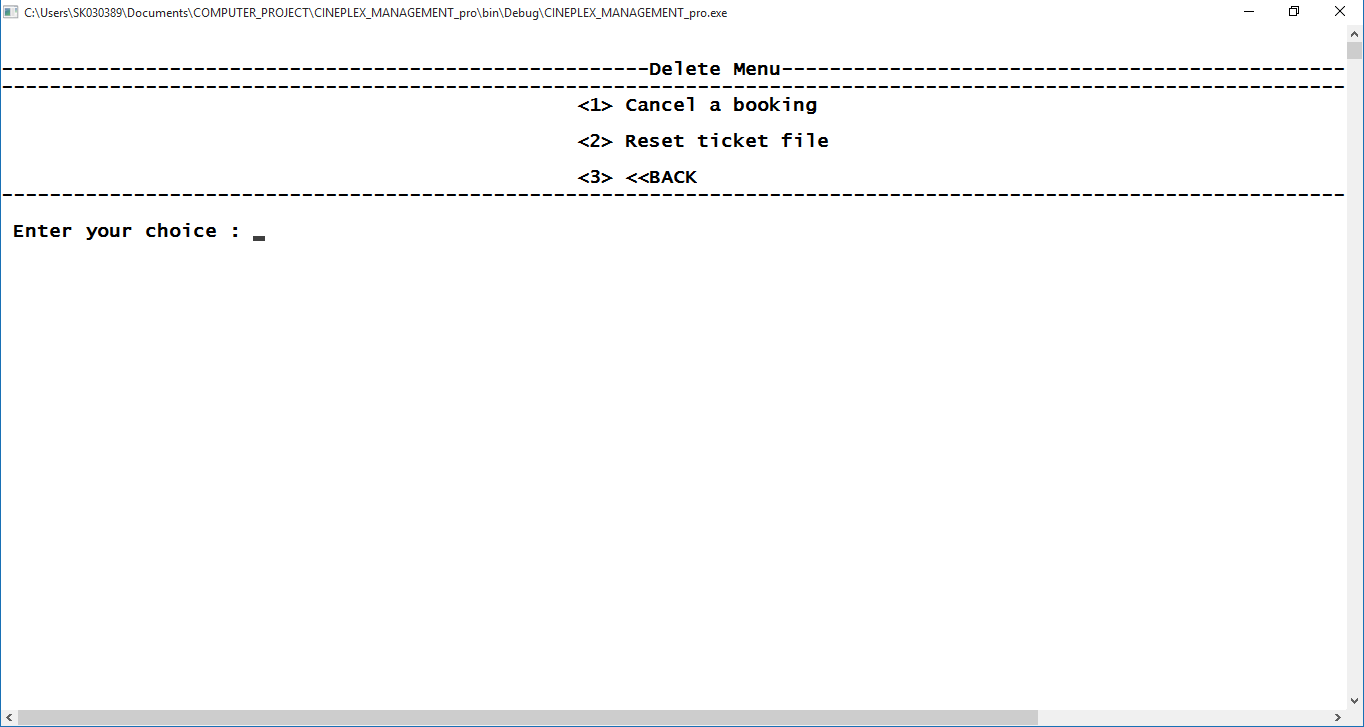
**Displaying Tickets**

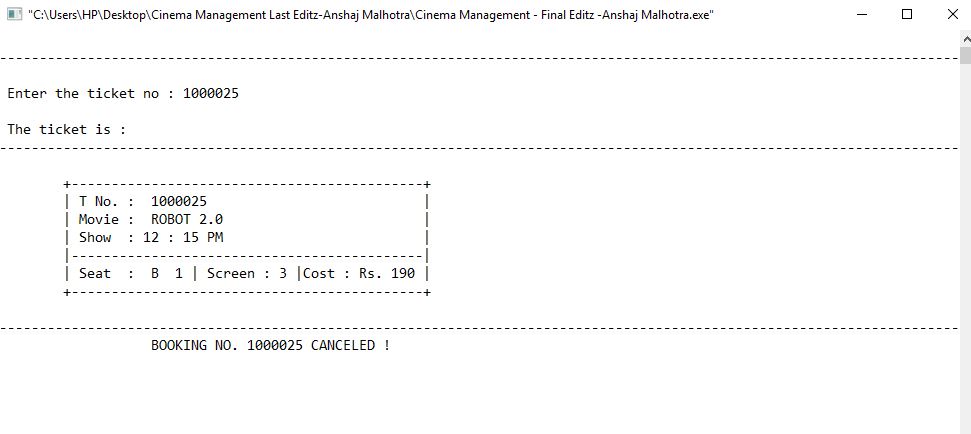


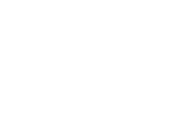
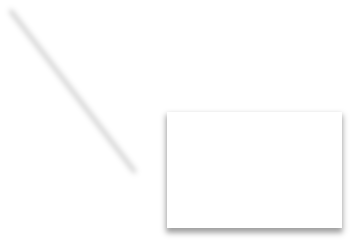
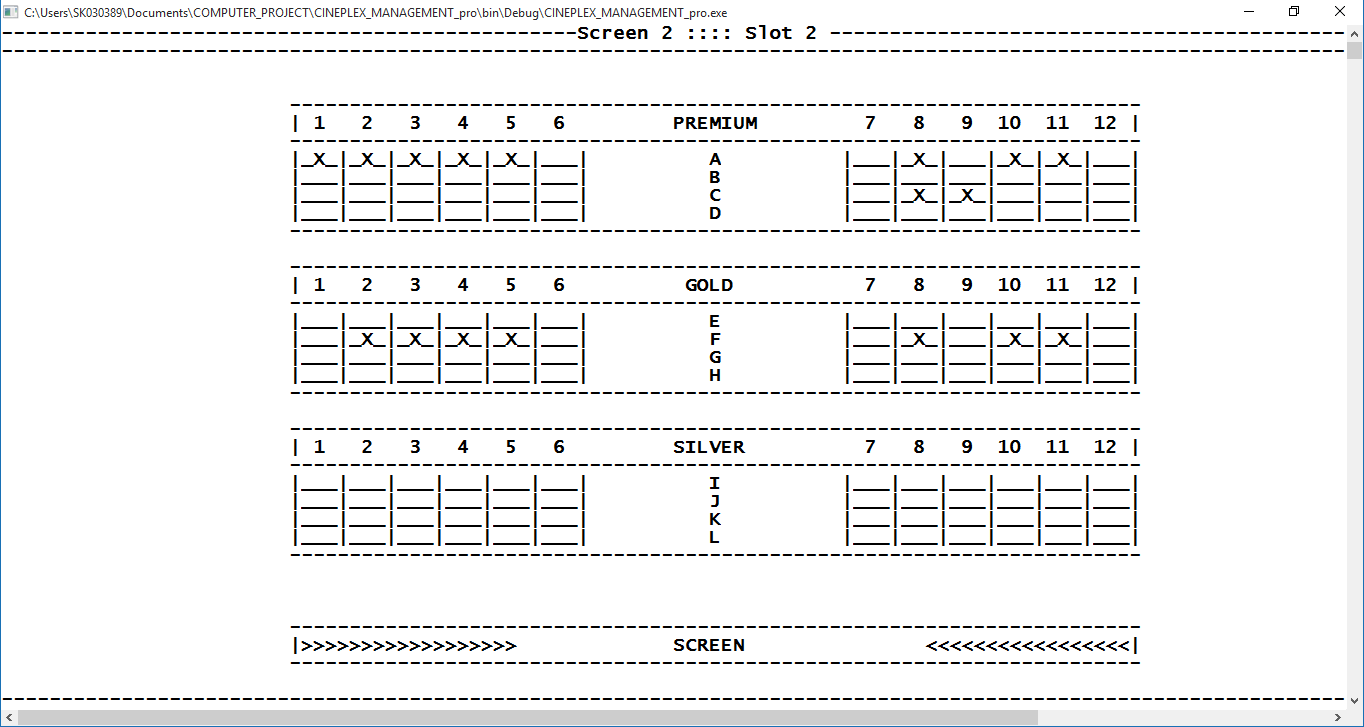




**Cancelling a booking**

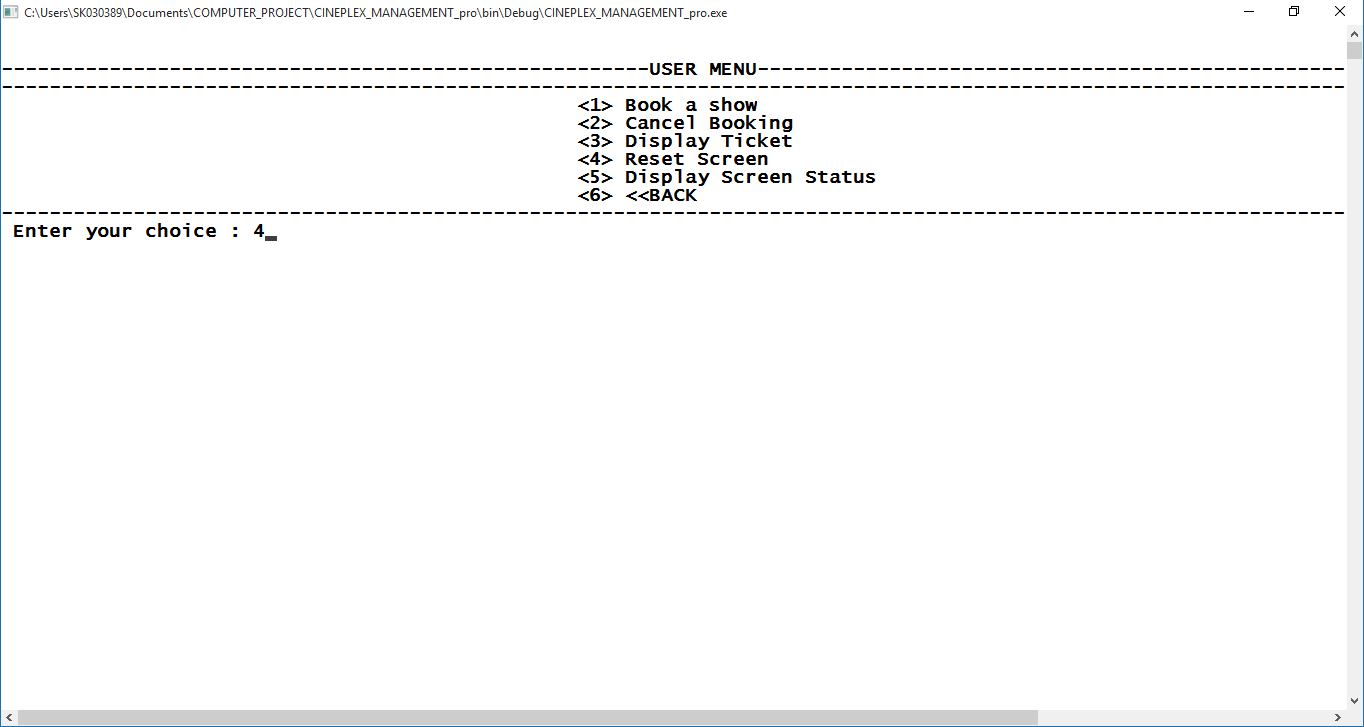


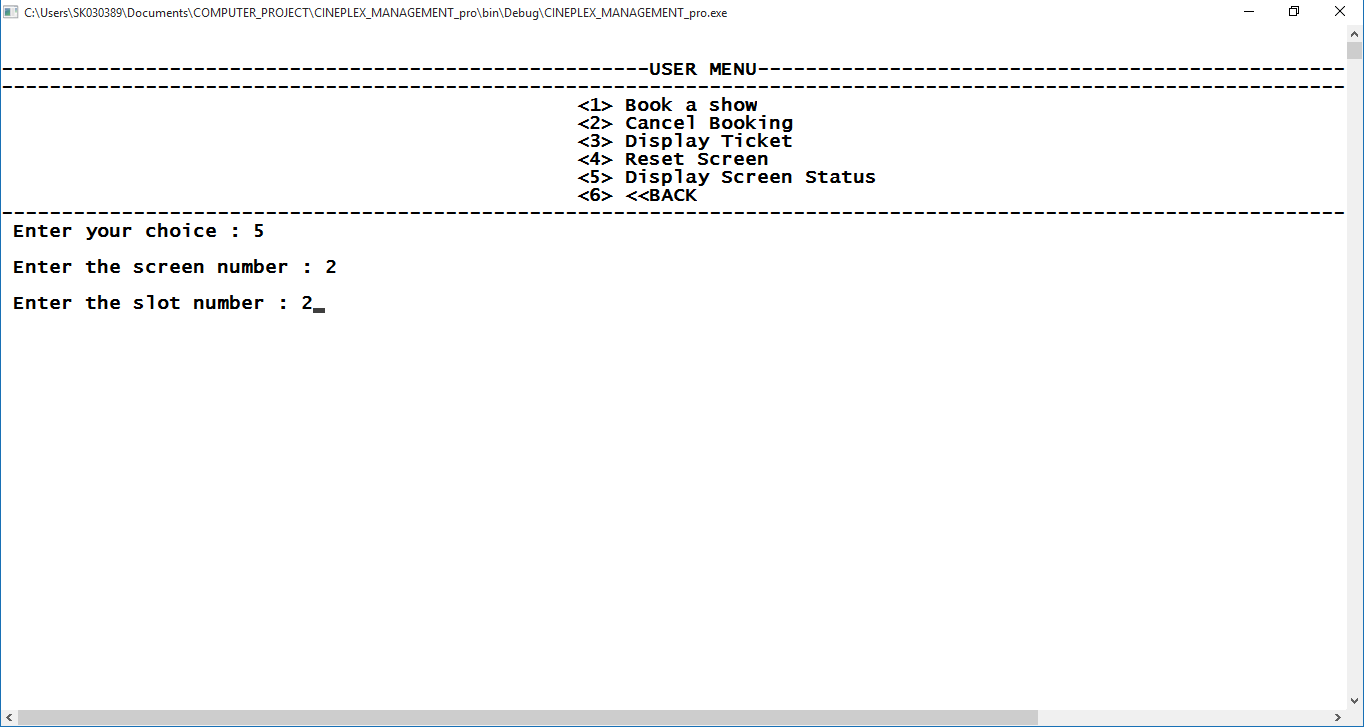
Booking cancelled



F9

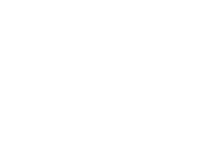
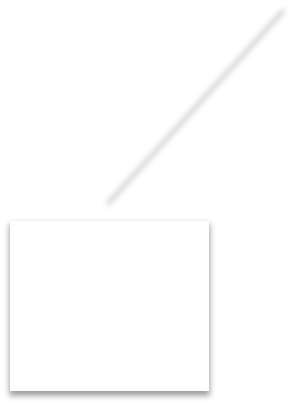
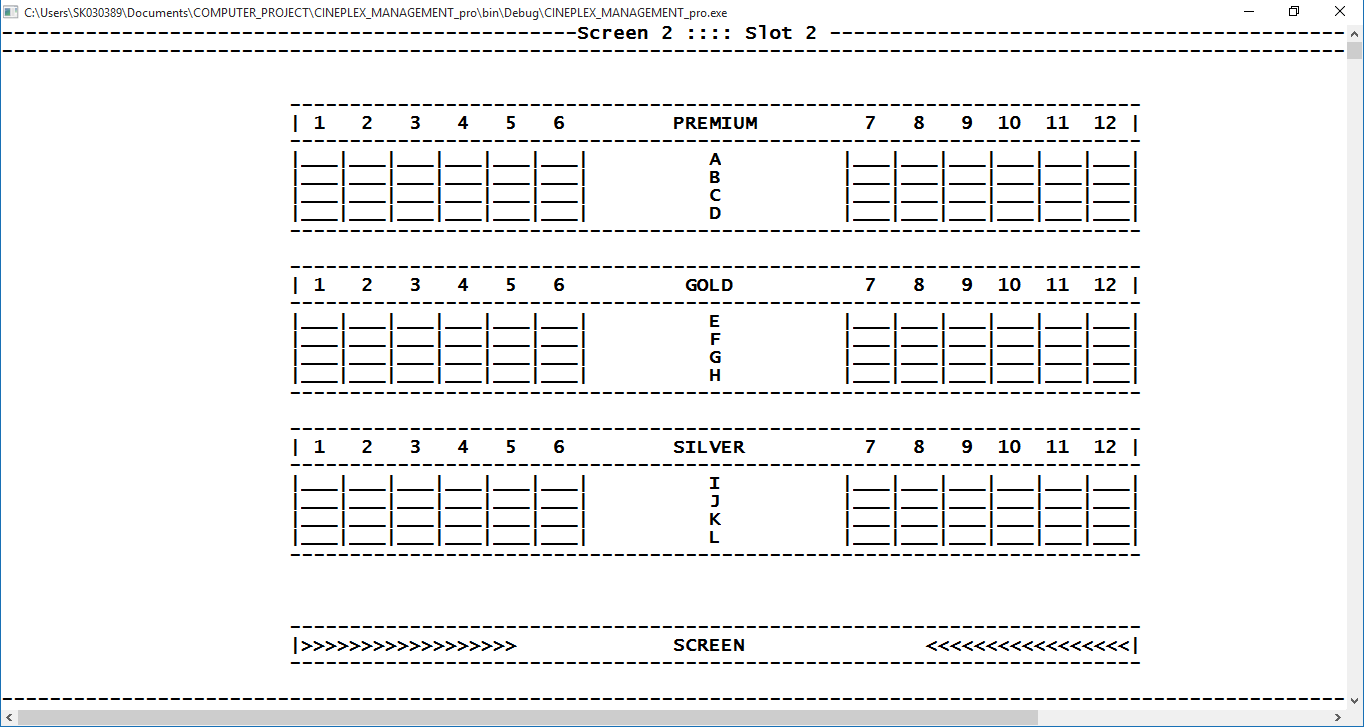
cancelled





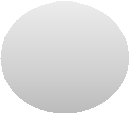
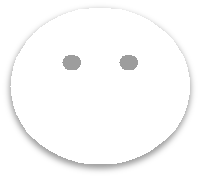
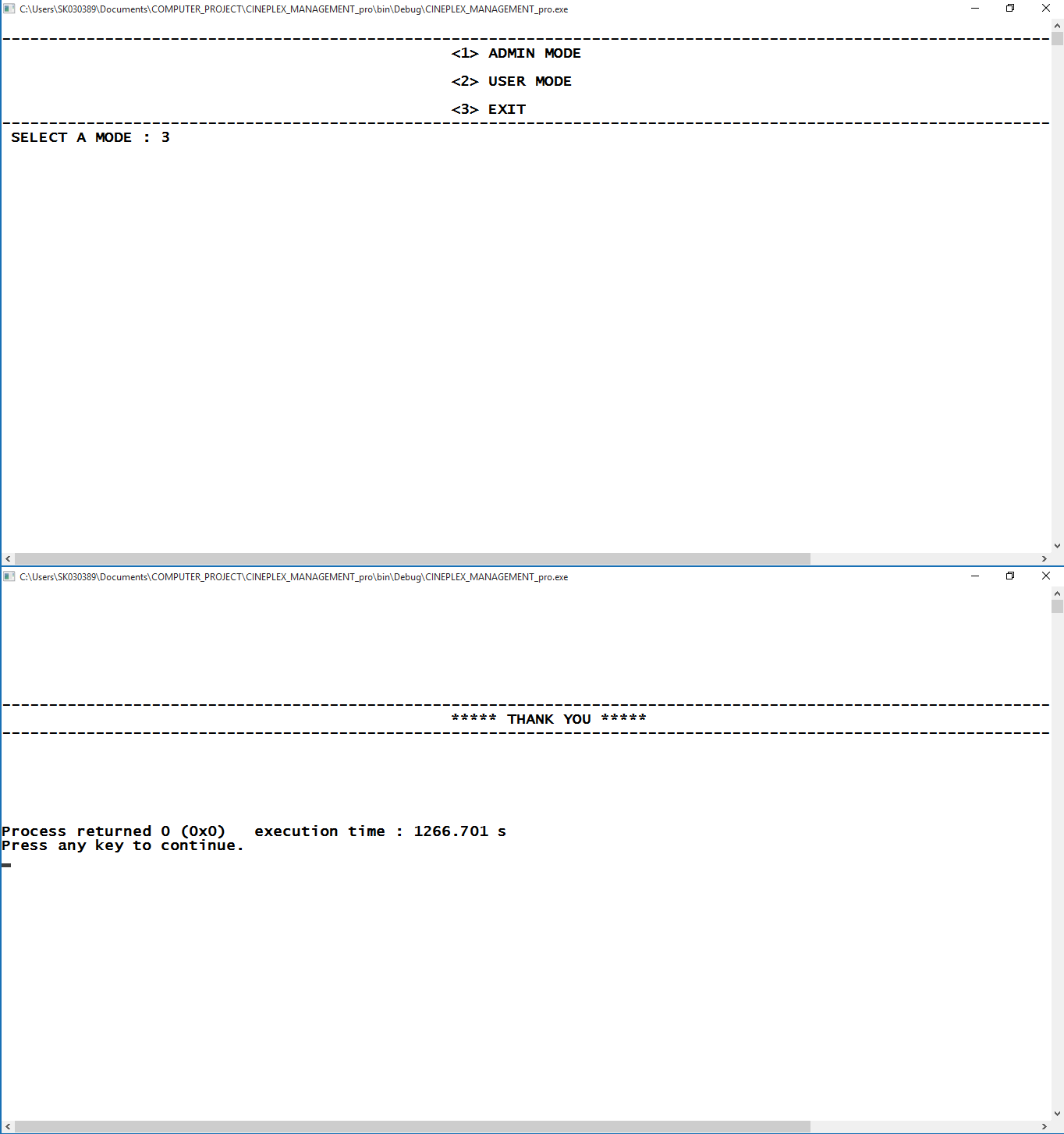
**The user has to reset a screen after each show**





The screen is reset

**EXIT**



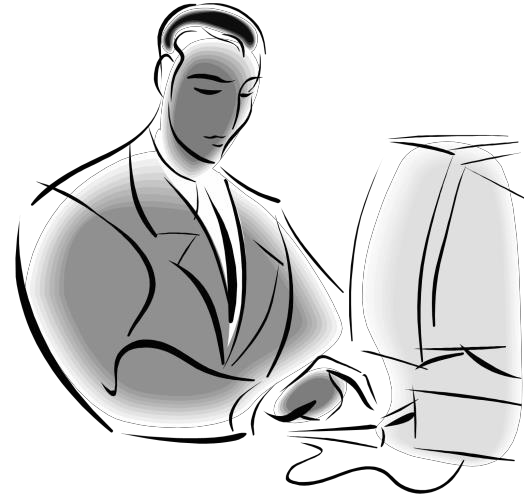
**That’s it!**

My Project lacks the following features :  
⮚**Screen infrastructure has to be pre-determined**⮚**There can be only one user at a time**⮚**2 Screens can not have 1 movie**⮚**Program does not facilitate easy group booking i.e.  
for booking 10 tickets we need to enter 10 seats.**Future Enhancements  
**In future I would like to embed the following enhancements, thereby,  
removing the short-comings of this C++ software :w.cbseportal.com  
1.Customized Screen infrastructure  
2.Multi user system  
3.multiple screens will have a common movie  
4.More interactive inter face  
5.Facilitating group booking  
6.Displaying statistics (total tickets sold etc.)**



## BIBLIOGRAPHY





Sumita Arora – Computer Science with C++

E. Balagurusami – C++ Robert Lafore – Turbo C++

website : [www.cppforschool.com](http://www.cppforschool.com/projects.html), [www.cbseguess.com](http://www.cbseguess.com/)