# VISVESVARAYA TECHNOLOGY UNIVERSITY

BELAGAVI-590018

B.L.D.E.A's V.P. Dr.P.G.Halakatti College of Engineering

and Technology



DEPARTMENT OF INFORMATION SCIENCE AND

ENGINEERING

A MINI PROJECT REPORT ON :

“Movie Booking System”

UNDER THE GUIDANCE OF:

Prof. Pradeep Deshpande.

Prof. Supriya Patil.

SUBMITTED BY:

1.Adil Badekhan (2BL20IS003)

2.Maaz Rakkasagi(2BL20IS016)

# VISVESVARAYA TECHNOLOGY UNIVERSITY BELAGAVI - 590018

B.L.D.E.A's V.P. Dr.P.G.Halakatti College of Engineering and Technology



DEPARTMENT OF INFORMATION SCIENCE

AND ENGINEERING

CERTIFICATE

This is to certify that the mini project titled “ MOVIE BOOKING SYSTEM ” under File Structure laboratory[18CSL58] work carried out by ADIL BADEKHAN

(2BL20IS003) and MAAZ RAKKASGI (2BL20IS016) submitted in partial fulfillment for the Ⅵ Semester of Bachelor of Engineering degree of the Visvesvaraya Technological University, Belgaum during the year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for File Structure laboratory with Mini Project prescribed in Information Science & Engineering of Ⅵ semester.

## Guide H.O.D Principal

Prof. Pradeep Deshpande. Prof. P.H.UNKI Dr. V. G. Sangam

Prof. Supriya Patil.

SIGNATURE SIGNATURE SIGNATURE

Examiner 1:

Examiner 2:

ACKNOWLEDGEMENT

We would like to express deep sense of gratitude to our beloved principal Dr. V. G. Sangam providing all facilities in the college.

We would like to thank who heartedly to our Head of Department Prof . P.H.UNKI for providing facilities and fostering congenial academic environment in the college.

We feel deeply indebted to our esteemed project guides Prof. PRADEEP DESHPANDE & Prof. SUPRIYA PATIL for their impactful help right from the conception and visualization to the very presentation of the mini project. They have been our guiding light.

We would take this opportunity to thank all the faculty members and supporting staff for helping us in this endeavor.

Project Associates

1.Adil Badekhan

2.Maaz Rakkasagi

ABSTRACT

The Movie Book System is a mini project implemented using C++. It aims to provide a user-friendly interface for managing a collection of movies, allowing users to add, remove, search, and display movie details. The system utilizes object-oriented programming principles and file handling techniques to store and retrieve movie information efficiently.

The Movie Book System allows users to create a database of movies by providing essential details such as movie title, genre, release year, director, and cast members. Users can add new movies to the system, edit existing movie information, and delete movies from the collection. The system also includes a search functionality that enables users to find specific movies based on title, genre, or director.

To enhance the user experience, the Movie Book System provides a user-friendly command-line interface, where users can easily navigate through different options and perform desired operations on the movie database. The system ensures data integrity by validating user input and handling errors gracefully. It also incorporates file handling mechanisms to persist movie data between different sessions, enabling users to access and modify the database across multiple program executions.

The Movie Book System mini project demonstrates the practical application of core concepts in C++ programming, including classes, objects, inheritance, file handling, and user input validation. It serves as an excellent learning resource for beginners to understand the implementation of a basic database management system in C++ and provides a foundation for further enhancements and feature additions.

Keywords: C++, mini project, movie database, object-oriented programming, file handling, user interface.

## CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapter .No** | **TOPICS** | **PAGE NO** |
| 1 | Introduction | 6 |
| 2 | Major concepts used | 7 |
| 3 | Working of the project | 10 |
| 4 | Applications | 12 |
| 5 | Results & Discussion | 13 |
| 6 | Source Code | 18 |
| 7 | Benefits of projects | 22 |
| 8 | Bibliography | 23 |
| 9 | Conclusion | 24 |
|  | | |

S

C

**CHAPTER** 1

**INTRODUCTION**

The Movie Book System is a mini project implemented using C++, aiming to create a user-friendly and efficient movie management system. It provides a platform for users to organize, store, and manipulate a collection of movies. The project demonstrates the practical application of fundamental programming concepts such as classes, objects, inheritance, and file handling in C++.In today's digital era, movies have become an integral part of our entertainment. With a vast array of movies available across different genres, it can be challenging to keep track of one's favorite films. Moreover, movie enthusiasts often desire a comprehensive system to manage and organize their movie collections effectively. The Movie Book System addresses this need by providing a solution that allows users to create a personalized movie database.

The Movie Book System offers various functionalities to enhance the movie management experience. Users can add new movies to the database, including essential details such as title, genre, release year, director, and cast members. They can also edit the existing movie information, update it with the latest details, or remove movies from the collection. The system provides a search feature that enables users to find movies based on criteria like title, genre, or director, allowing for easy retrieval of specific films.

To ensure a seamless user experience, the Movie Book System incorporates a user-friendly command-line interface. This interface provides intuitive options and menu-driven navigation, allowing users to perform desired operations on the movie database effortlessly. Error handling mechanisms are implemented to validate user input and provide informative feedback, ensuring data integrity and preventing unexpected program behavior.

The project also leverages file handling techniques to store movie data persistently. By utilizing file operations, the Movie Book System enables users to access and modify their movie collections across different program executions. This feature ensures that the database remains intact even after closing the application, providing a seamless and continuous movie management experience.

In the following sections, we will delve into the detailed design, implementation, and features of the Movie Book System, showcasing how it provides an efficient and user-friendly.

**CHAPTER 2**

**MAJOR CONCEPTS USED**

In a file structure mini project like the Movie Book Ticket System implemented using C++, several major concepts are typically employed to manage and manipulate data effectively. Some of the key concepts utilized in such a project are:

**File Handling:**File handling is a crucial concept in managing data persistently. In the Movie Book Ticket System, file handling is used to read and write movie, booking, and ticket information to files. It involves opening files, reading from them, writing to them, and closing them properly. C++ provides file stream classes like fstream to facilitate file handling operations.

**Data Serialization:**Data serialization involves converting complex data structures into a format suitable for storage or transmission. In the context of the Movie Book Ticket System, serialization is used to convert movie, booking, and ticket objects into a serialized format before writing them to files. This allows for efficient storage and retrieval of data. Serialization can be achieved using techniques like writing object data member by member or utilizing libraries like Boost.Serialization.

**Data Structures:**Data structures play a vital role in organizing and managing data efficiently. In the Movie Book Ticket System, various data structures can be employed. For example, linked lists or arrays can be used to store and manage movie information, including details like title, genre, and director. Similarly, data structures like queues or stacks can be utilized to manage ticket bookings and maintain a record of ticket sales.

**Object-Oriented Programming (OOP) Concepts:**OOP concepts form the foundation of the Movie Book Ticket System. These include classes, objects, inheritance, and polymorphism. Classes are used to encapsulate data and behavior related to movies, bookings, and tickets. Objects are instances of these classes that store specific data. Inheritance allows for creating specialized classes based on existing ones, promoting code reusability. Polymorphism enables the usage of abstract classes and virtual functions to handle different types of movie bookings and tickets.

**Input/Output Validation:**Input/Output validation is crucial to ensure data integrity and prevent errors in the Movie Book Ticket System. User input for movie details, booking information, or ticket data needs to be validated to ensure it adheres to the required format and constraints. Input validation can involve techniques such as checking for valid data types, ranges, and formats. Similarly, output validation ensures that data displayed to users is accurate and properly formatted.

**Exception Handling:**Exception handling is employed to handle unexpected situations or errors that may occur during program execution. In the Movie Book Ticket System, exception handling is used to catch and handle errors related to file operations, input validation, or any other exceptional conditions that may arise. Proper error messages and fallback mechanisms can be implemented to guide users and maintain program stability.

**Functionalities of the Project:**

BOOK TICKETS

SEE MOVIE DETAILS

CHECK SEATS AVAILABILITY

ALLOWS MANAGER TO MANAGE MOVIE DETAILS

SHOWS TOTAL COST OF THE TICKETS BOOKED

**Major Features of the Project:**

FOR MANAGER:

ALLOWS MANAGER TO DECIDE:-

MOVIES TO BE SCREENED

NUMBER OF SHOWS IN A DAY

SHOW TIMINGS

CAN ALSO RESERVE SEATS FOR TICKETS BOOKED ONLINE

FOR USERS:

ALLOWS USERS TO DECIDE:-

BOOK TICKETS

SEE MOVIE DETAILS

AVAILABLE SEAT TYPES :

RECTILINEAR

DELUXE

SUPER DELUXE

**CLASS Movie Booking THAT CONTAINS FOLLOWING FUNCTIONS:**

first()

show\_seat\_frm\_file()

show()

seatdisp()

book()

seatin()

manager()

ticket\_print()

pass\_protect()

data\_str()

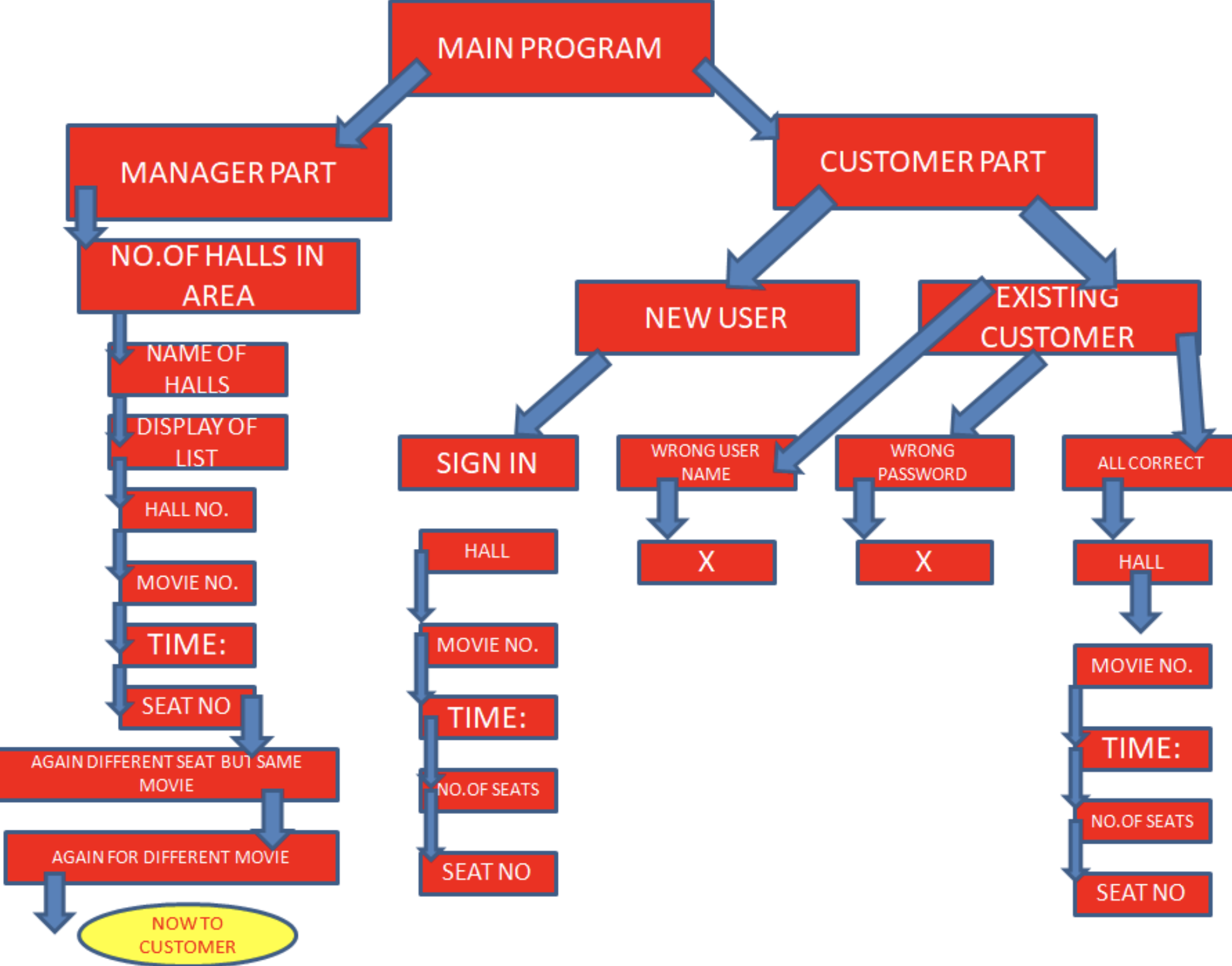
password\_check()

**CHAPTER 3**

**WORKING OF THE PROJECT**

The working of a file structure mini project like the Movie Book Ticket System implemented using C++ involves several key steps and functionalities. Here's an overview of how the system typically operates:

* **File Management:** The project starts by creating and managing the necessary files to store movie, booking, and ticket information. These files could include a movie database file, a booking record file, and a ticket sales file. The system ensures the files are present and accessible for reading and writing data.
* **Movie Management:** The user interacts with the system to perform movie-related operations. This includes adding new movies to the system by providing details such as title, genre, director, and cast. The system prompts the user to enter this information and then stores the movie data in the movie database file using file handling techniques.
* **Booking Management:** Users can book tickets for movies through the system. The system prompts the user to select a movie from the available list and enter the required details like the number of tickets, showtime, and customer information. It then stores this booking information in the booking record file, associating it with the respective movie.
* **Ticket Sales:** After a successful booking, the system generates tickets based on the provided information. It may create a unique ticket ID and print a ticket containing details like the movie title, showtime, seat number, and customer information. The system records this ticket information in the ticket sales file, associating it with the respective booking.
* **Data Retrieval:** The system allows users to retrieve information from the files. For instance, users can search for a specific movie by title, and the system reads the movie database file, retrieves the matching record, and displays the movie details. Similarly, users can search for booking information by ticket ID .
* **Data Updating and Deletion:** Users have the ability to update movie details, booking information, or customer data. The system prompts the user to select the specific record to be updated, retrieves the corresponding data from the file, and allows the user to modify it. Similarly, the system provides options to delete movies, bookings, or tickets, which involves locating the relevant record in the file and removing it.

****

**CHAPTER 4**

**APPLICATIONS OF PROJECT**

The Movie Book Ticket System, implemented as a file structure mini project using C++, has several practical applications. Here are some of the key applications of such a system:

* Movie Theater Ticket Booking: The Movie Book Ticket System can be used in movie theaters to facilitate ticket booking operations. It allows users to select movies, book tickets for specific showtimes, and generate tickets with relevant details like seat numbers. The system ensures efficient management of ticket bookings and provides a seamless experience for both customers and theater staff.
* Online Movie Ticketing Platforms: Online platforms that sell movie tickets can utilize the Movie Book Ticket System to manage their ticketing operations. The system can handle the process of adding new movies, managing showtimes, and processing ticket bookings. It allows customers to browse available movies, select showtimes, and book tickets online, providing a convenient and efficient ticketing solution.
* Film Festivals and Events: Film festivals or special events that feature multiple movies can benefit from the Movie Book Ticket System. The system can help organizers manage the screening schedules, ticket sales, and attendee information. It enables efficient management of movie screenings, ticket availability, and ensures smooth operations during the event.
* Movie Rental Services: Movie rental services can utilize the Movie Book Ticket System to manage their movie collections and rental operations. The system allows the rental service to add movies to their database, track movie availability, and manage customer rentals. It provides a structured approach to organizing movie collections and automating rental processes.
* Film Clubs and Societies: Film clubs or societies that organize regular movie screenings or events can employ the Movie Book Ticket System. It allows members to book tickets for screenings, keeps track of attendance, and provides an organized system for managing club activities. The system facilitates efficient coordination and management of movie-related events.
* Educational Institutions: Educational institutions offering film-related courses or organizing film screenings can benefit from the Movie Book Ticket System. It enables easy management of movie screenings, ticket reservations for students, and tracking attendance.

CHAPTER 5

RESULTS AND SCREENSHOTS

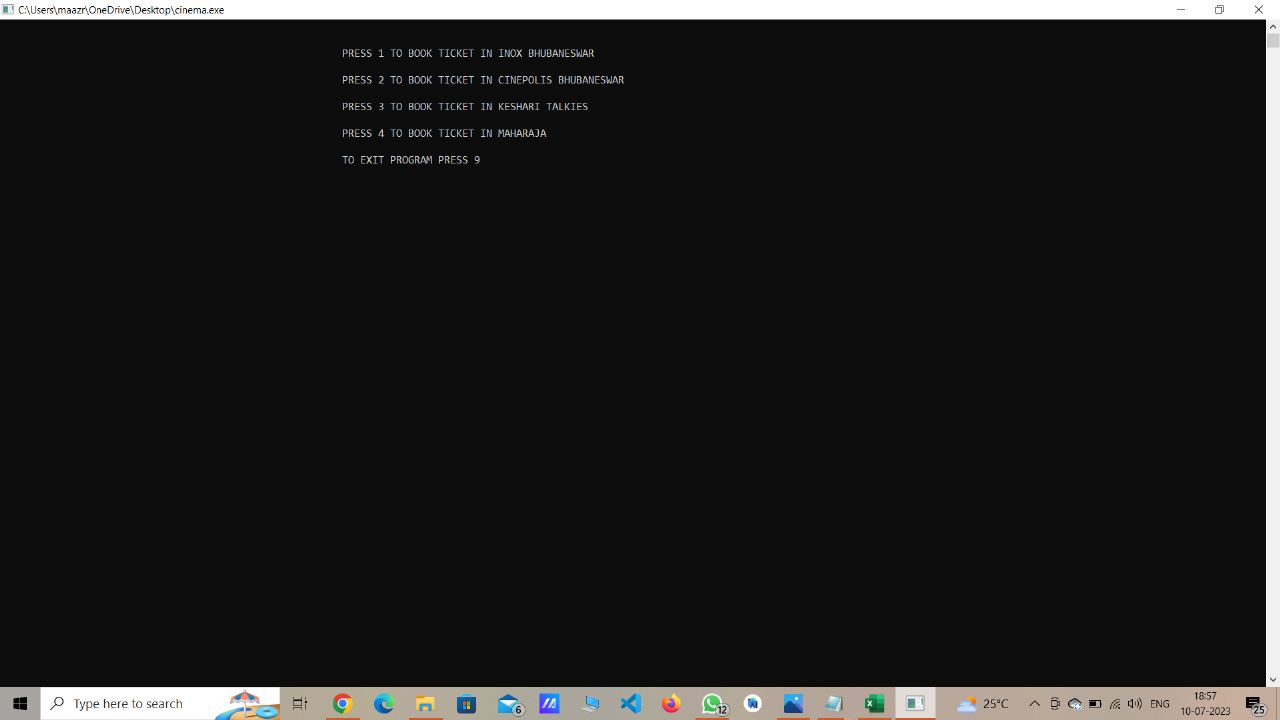


Fig .1 Home page

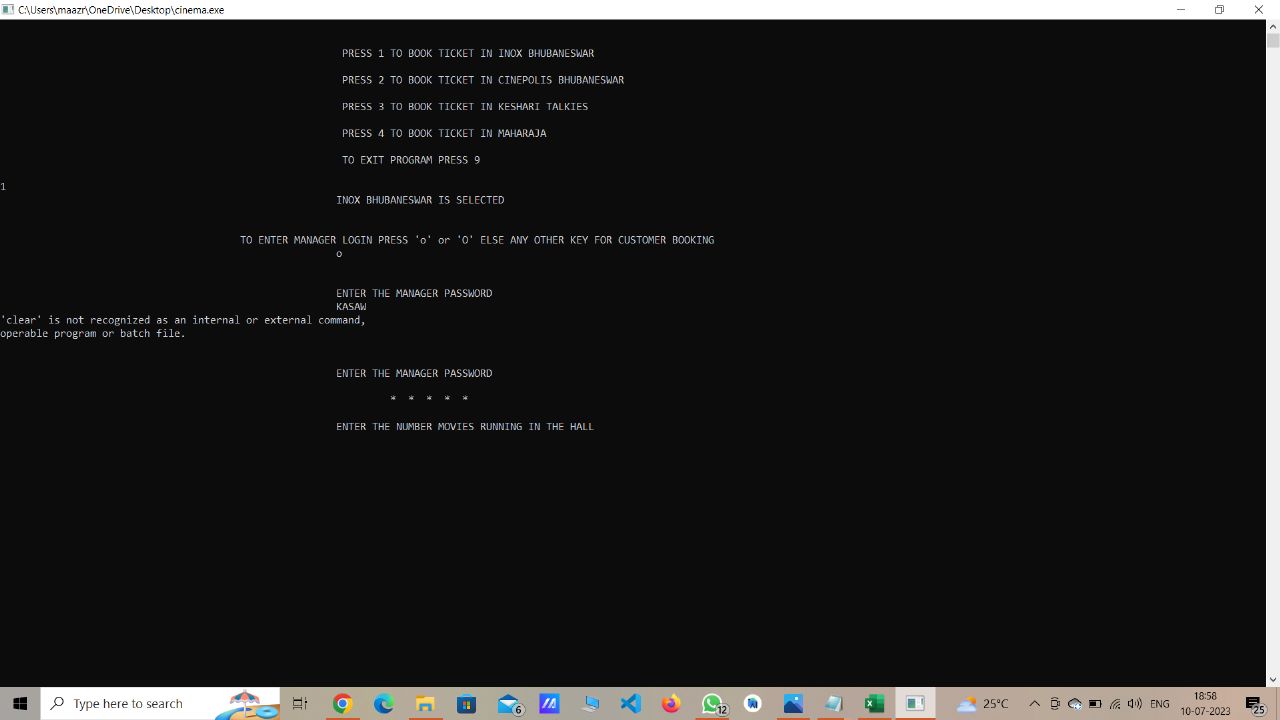


Fig2. Manager login page

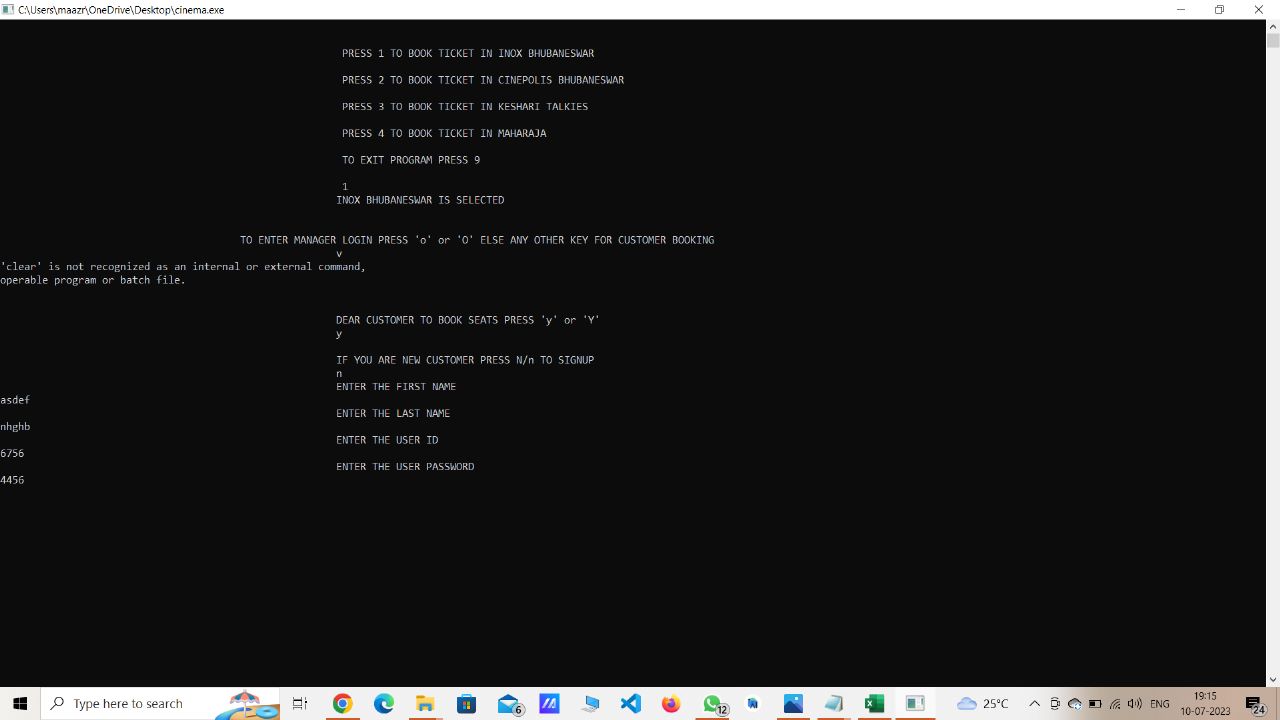


Fig3. Sign up page for new user

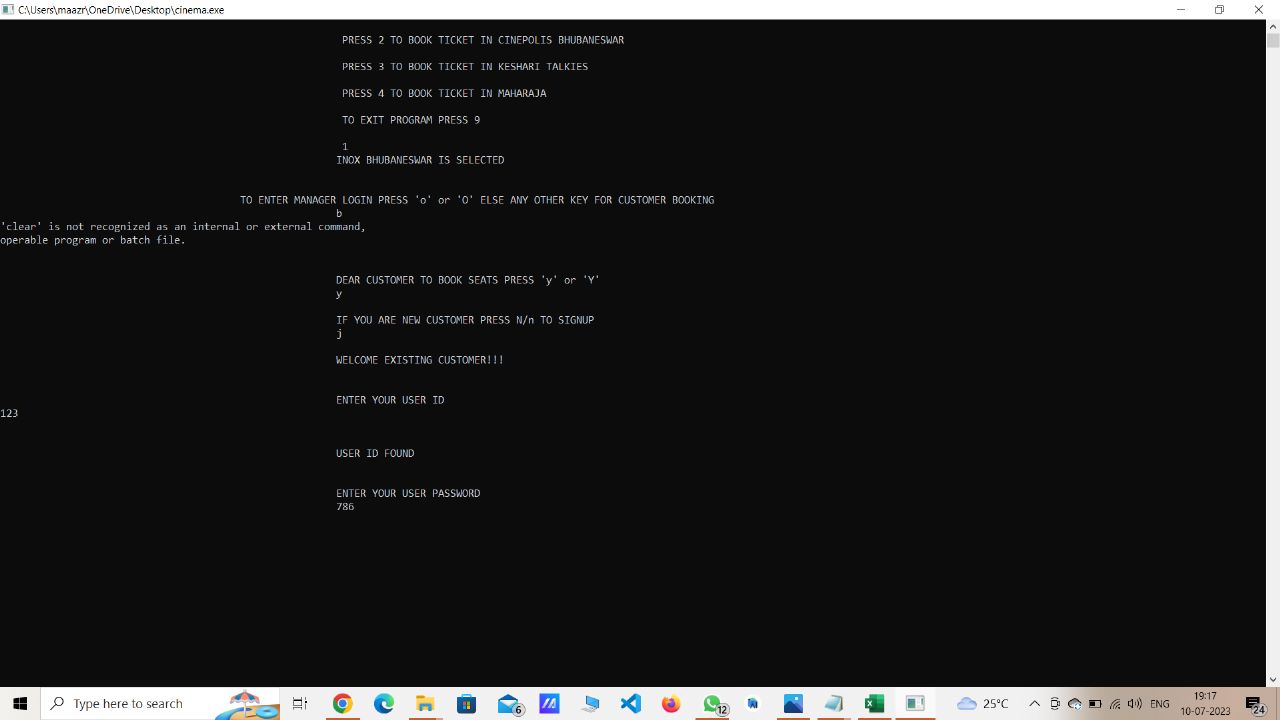


Fig4. User login page

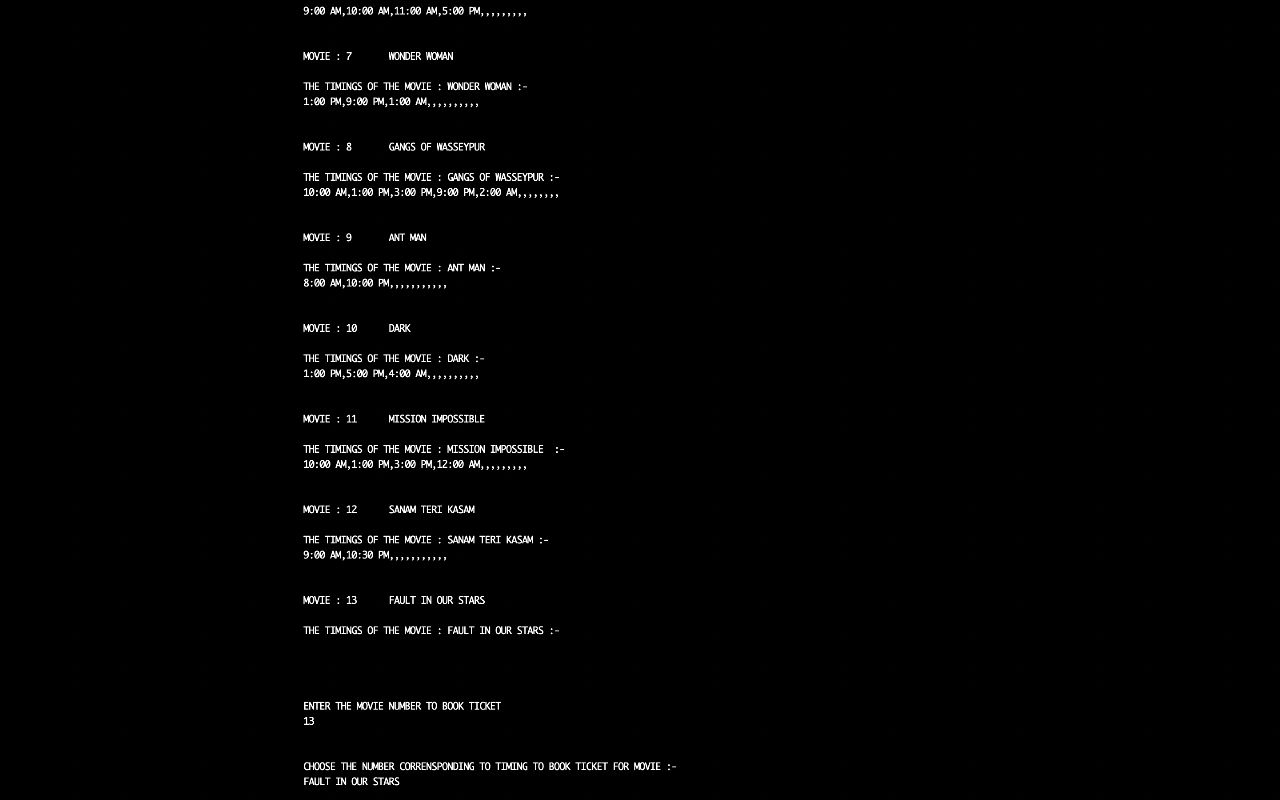


Fig5. Movies Display

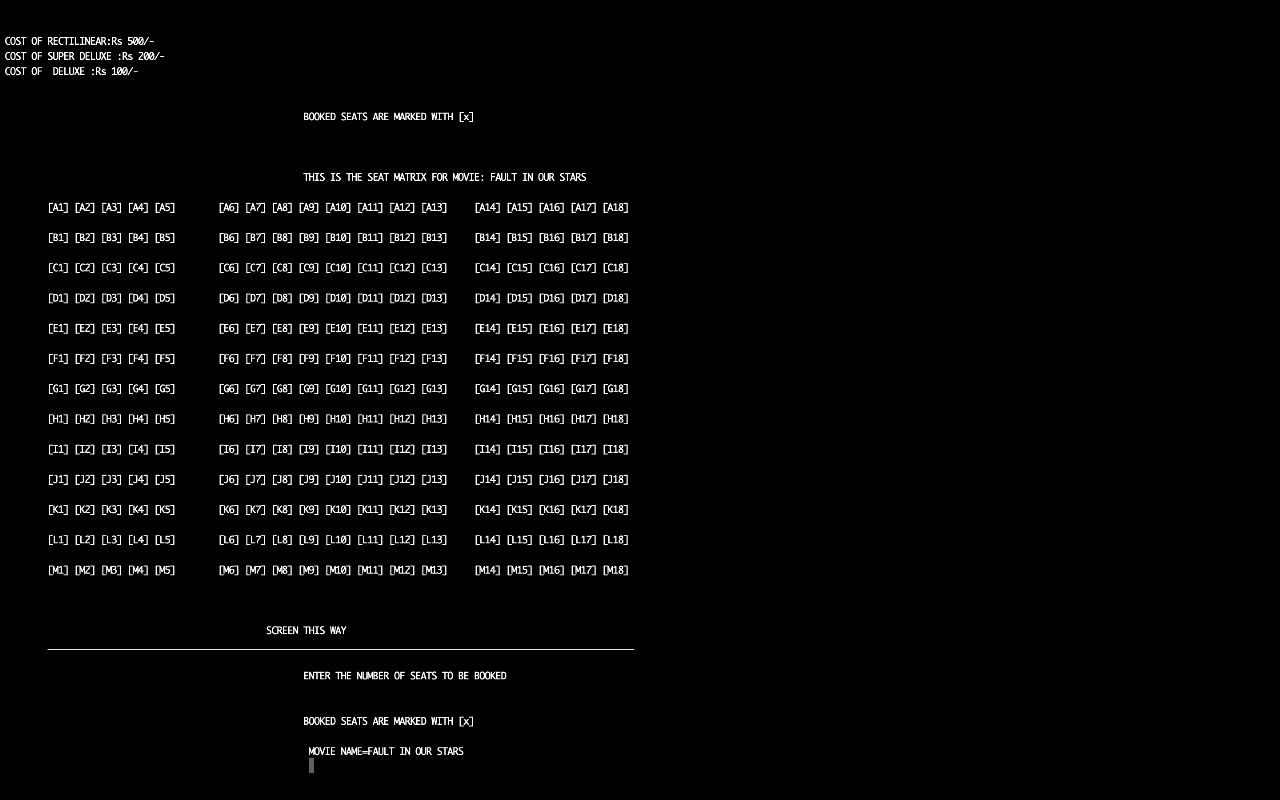


Fig6. Ticket Booking page



Fig7.Ticket Invoice

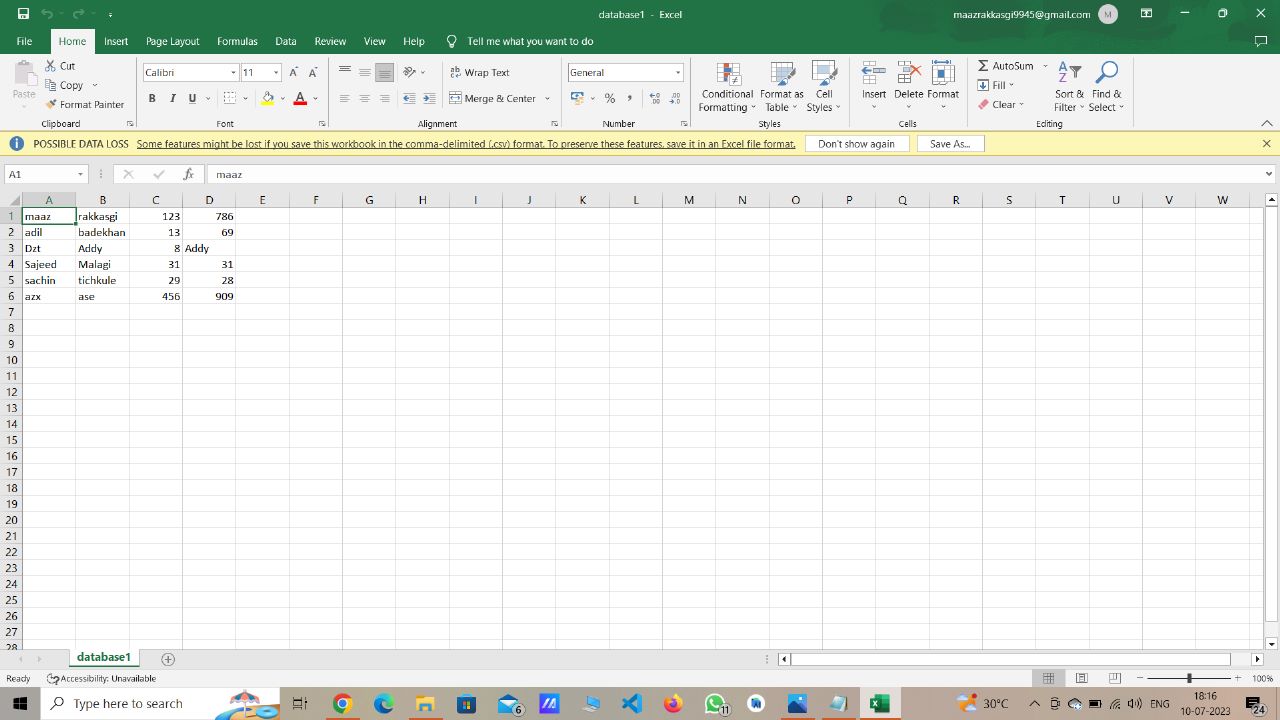


Fig8. User database

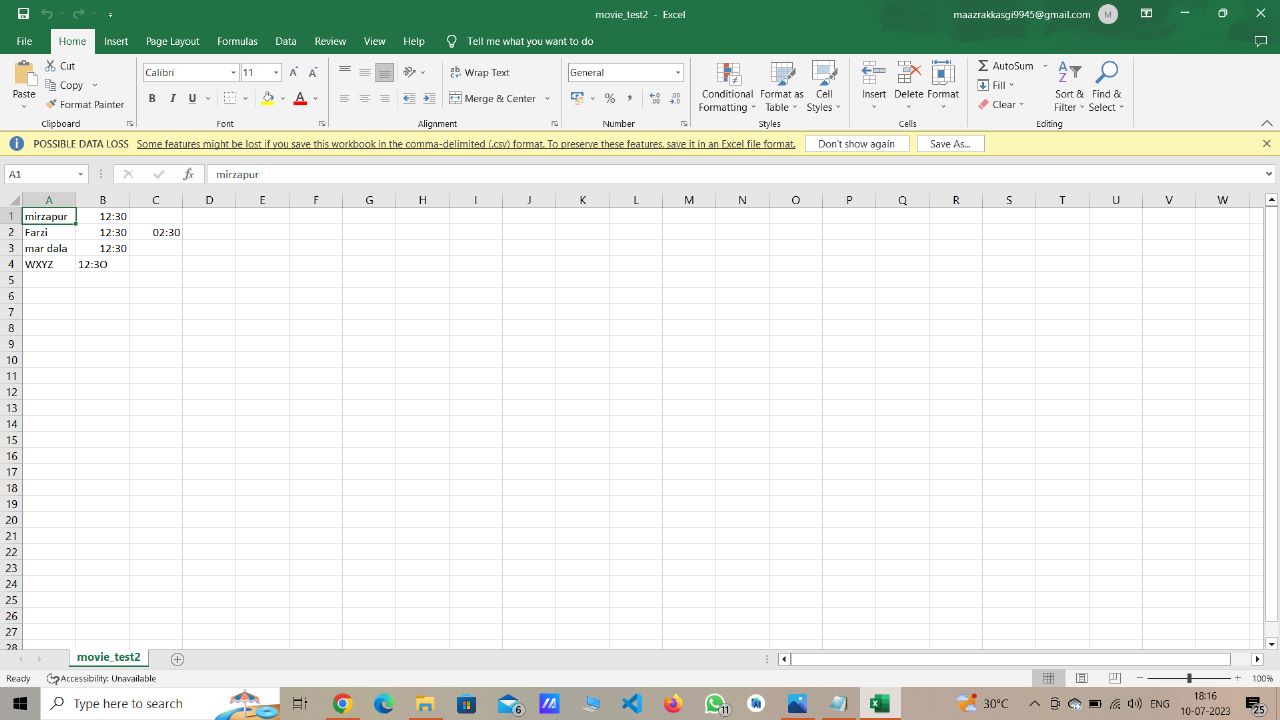


Fig9. Data Base of Movies

CHAPTER 6

SOURCE CODE

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <malloc.h>

#include <fstream>

using namespace std;

class movie\_booking

{

public:

void first();

void show\_seat\_frm\_file();

void show();

void seatdisp(int z,int wt);

void book(int z,int wt ,int a);

void seatin(int z,int wt);

void manager();

void ticket\_print();

void pass\_proctect();

void data\_str();

int password\_check();

void hal\_name()

{

switch(obn)

{

case 1:

hall\_name="INOX BHUBANESWAR";

break;

case 2:

hall\_name="CINEPOLIS BHUBANESWAR";

break;

case 3:

hall\_name="KESHARI TALKIES";

break;

case 4:

hall\_name="MAHARAJA";

break;

}

}

};

void movie\_booking::first()

{

int i;

char in='c',ind,cus;//cus is customer and ind is manager input

cout<<"\n\n\t\t\t\t\tTO ENTER MANAGER LOGIN PRESS 'o' or 'O' ELSE ANY OTHER KEY FOR CUSTOMER BOOKING\n";

cout<<"\t\t\t\t\t\t\t";

cin>>ind;

int pas\_count=0;

if((ind=='o')||(ind=='O'))

{

while(pas\_count<3)

{

(\*this).pass\_proctect();

if(pass=="KASAW")

{

(\*this).manager();//the manager mode to set the number of seats available for booking

pas\_count=3;

}

else

{

cout<<"PASSWORD ENTERED IS WRONG!!"<<" ATTEMPTS REMAINING = "<<2-pas\_count<<" \n";//pas\_count is to count no. of attempts left

pas\_count++;

}

}

}

system("clear");

cout<<"\n\n\t\t\t\t\t\t\tDEAR CUSTOMER TO BOOK SEATS PRESS 'y' or 'Y' \n";

cout<<"\t\t\t\t\t\t\t";

cin>>cus;

//HERE I HAVE TO ADD THE CUSTOMER LOGIN FUNCTION

if((cus=='y')||(cus=='Y'))//the customer part starts here

{ cout<<"\n\t\t\t\t\t\t\tIF YOU ARE NEW CUSTOMER PRESS N/n TO SIGNUP\n";

cout<<"\t\t\t\t\t\t\t";

cin>>cus\_in;

if(cus\_in=='N'||cus\_in=='n')

(\*this).data\_str();

else

{

cout<<"\n\t\t\t\t\t\t\tWELCOME EXISTING CUSTOMER!!!\n";

pas\_rak=(\*this).password\_check();

}

//cout<<"+++++++++++"<<pas\_rak<<"--------"<<data\_store<<endl;to check the values

cout<<"\t\t\t\t\t\t\t";

cout<<"PRESS ANYKEY TO CONTINUE\n";

// CLEAR();

system("clear");

if(pas\_rak!=0||data\_store!=0)

{ cout<<"\n\n\t\t\t\t\t\tTHE SEATS AVAILABLE ARE:\n";

while((in=='c')||(in=='C'))

{

cost=0;

(\*this).show\_seat\_frm\_file();

cout<<"\n\n\t\t\t\t\t\t\tENTER THE MOVIE NUMBER TO BOOK TICKET\n";

cout<<"\t\t\t\t\t\t\t";

cin>>arg;

ifstream fin;//this part is for retrieving moive name from file

fin.open("movie\_test2.csv");

for(i=0;i<arg-1;i++)

{

getline(fin,name[i],'\n');

}

getline(fin,name[i],',');

fin.close();

cout<<"\n\n\t\t\t\t\t\t\tCHOOSE THE NUMBER CORRENSPONDING TO TIMING TO BOOK TICKET FOR MOVIE :-"<<endl;

cout<<"\t\t\t\t\t\t\t"<<name[i];

cout<<"\t\t\t\t\t\t\t";

cin>>sti;

tkt\_pnt=name[i];

// cout<<"THE ARGUMENT IS ="<<arg<<endl;

(\*this).seatdisp(arg,sti);

(\*this).seatin(arg,sti);

system("clear");

(\*this).seatdisp(arg,sti);

/HERE ANOTHER FUNCTION IS TO BE WRITTEN TO PRINT THE TICKETS/

system("clear");

(\*this).ticket\_print();

cout<<"\n\n\t\t\t\t\t\t\tPRESS 'C' or 'c' IF U WANT TO BOOK SEAT AGAIN?\n";

cin>>in;

}

}

else{cout<<"\n\n\t\t\t\t\t\t\t YOU PRESSED WRONG USER PASSWORD\n";

cout<<"\n\n\t\t\t\t\t\t\t PROGRAM TERMINATED!!!\n";

cout<<"\n\n\t\t\t\t\t\t\t YOU ENTERED WRONG PASSWORD";}

}

}

int main()

{

prnt\_once=0;

movie\_booking obj[4];

while(1)

{

cout<<"\n\n\t\t\t\t\t\t\t PRESS 1 TO BOOK TICKET IN INOX BHUBANESWAR";

cout<<"\n\n\t\t\t\t\t\t\t PRESS 2 TO BOOK TICKET IN CINEPOLIS BHUBANESWAR";

cout<<"\n\n\t\t\t\t\t\t\t PRESS 3 TO BOOK TICKET IN KESHARI TALKIES ";

cout<<"\n\n\t\t\t\t\t\t\t PRESS 4 TO BOOK TICKET IN MAHARAJA";

cout<<"\n\n\t\t\t\t\t\t\t TO EXIT PROGRAM PRESS 9 ";

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

cin>>obn;

switch(obn)

{

case 1:

cout<<"\t\t\t\t\t\t\t";

cout<<"INOX BHUBANESWAR IS SELECTED\n";

obj[0].first();

break;

case 2:

cout<<"\t\t\t\t\t\t\t";

cout<<"CINEPOLIS BHUBANESWAR IS SELECTED\n";

obj[1].first();

break;

case 3:

cout<<"\t\t\t\t\t\t\t";

cout<<"KESHARI TALKIES BHUBANESWAR IS SELECTED\n";

obj[2].first();

break;

case 4:

cout<<"\t\t\t\t\t\t\t";

cout<<"MAHARAJA BHUBANESWAR IS SELECTED\n";

obj[3].first();

break;

case 9:

goto bye;

break;

default:

cout<<"enter a valid number\n";

goto bye;

break;}}

**CHAPTER 7**

**BENEFITS**

* Efficient Data Management: The project provides an efficient way to manage movie, booking, and ticket information. By utilizing file handling techniques, the system can read and write data to files, ensuring persistent storage and easy retrieval. This allows for organized and efficient management of movie collections, ticket bookings, and ticket sales.
* Data Persistence: The system ensures that data is persisted across different program executions. Movie information, booking records, and ticket sales data are stored in files, allowing users to access and modify the information even after closing and reopening the program. This ensures data integrity and enables seamless continuity in managing movie bookings and sales.
* User-Friendly Interface: The Movie Book Ticket System provides a user-friendly interface for users to interact with the system. It typically includes intuitive menus, prompts, and options that guide users through various functionalities. The system ensures a smooth user experience, making it easy for users to book tickets, search for movies, update information, and perform other operations.
* Streamlined Ticket Booking Process: The system simplifies the ticket booking process for users. It allows them to browse available movies, select showtimes, and book tickets with relevant details such as seat numbers. By automating ticket booking operations, the system eliminates manual processes, reduces errors, and saves time for both users and ticketing staff.
* Enhanced Organization and Searchability: The Movie Book Ticket System enables efficient organization and searchability of movie-related data. Users can search for movies based on title, genre, or director, making it easy to find specific movies of interest. The system also maintains booking records and ticket sales data, allowing for easy retrieval and analysis of information when needed.

**CHAPTER 8**

**BIBLIOGRAPHY**

We have taken help from the internet and books for making this movie booking system. This includes a lot of major concepts of oops. Our project was made with the help of information and concepts from the following:

**Sites:**

* <https://cppsecrets.com/users/22319897989712197103975756505164103109971051084699111109/C00-MOVIE-TICKET-BOOKING-SYSTEM.php>
* [www.youtube.com](http://www.youtube.com)
* <https://openai.com/blog/chatgpt>

**Books:**

* Object Oriented Programming with C++ by E Balagurusamy
* Fundamentals Of Computer programming & IT by Sumita Arora

CHAPTER 9

CONCLUSION

In conclusion, the file structure mini project Movie Book Ticket System implemented using C++ offers an efficient and user-friendly solution for managing movie information, ticket bookings, and ticket sales. By utilizing file handling techniques and data serialization, the system ensures persistent storage and retrieval of data, allowing users to access and modify information across different program executions. The project provides a streamlined ticket booking process, simplifying the selection of movies, booking of tickets, and generation of tickets with relevant details. The user-friendly interface and error handling mechanisms enhance the user experience and maintain data integrity. The system's scalability and flexibility make it suitable for various applications, such as movie theaters, online ticketing platforms, film festivals, and educational institutions. Overall, implementing the Movie Book Ticket System using C++ not only provides practical benefits for movie ticketing operations but also offers a valuable learning opportunity for developers to strengthen their programming skills and gain hands-on experience in file handling, object-oriented programming, and data management.