# 



Movie Ticket Reservation System

# **Introduction:**

The movie ticket reservation system is a console-based C++ application that simulates the process of booking, viewing, and managing movie tickets. It provides two types of users — **Admin** and **Customer** — with role-specific features. The project applies key programming concepts like arrays, loops, functions, pointers, and file handling to implement **CRUD operations** in a structured and user-friendly manner.

# **Objectives:**

* Develop a practical application using C++
* Use course learned concepts learnt throughout the semester
* Apply **File handling** and **Pointers** to enhance functionality
* Allow users to interact with the system to book, view, and manage movie tickets.
* Log user actions in a file (history.txt) for persistent tracking.
* Implement **Create, Read, Update, Delete (CRUD)** features for data management.
* Introduce a user interface for both **Admin** and **Customer** roles.

# **System Design**

### Key Features

The program prompts the user to select their role and then displays the relevant menu.

**✅ Admin:**

* Add New Movies (Create)
* View All Movies (Read)
* Delete Movies (Delete)
* View Reservations (Read)
* View Action History (Read from file)

**✅ Customer:**

* View Movies (Read)
* Search Movies (Read)
* Book Tickets (Update)

## Functionalities:

1. Adds a Movie to the list (using pointers) to modify the arrays that store movie data, and the program automatically writes the record to history.txt file
2. Lists all the Movies with available seats
3. Users can search for a movie by its title or genre.
4. Reserves tickets for a movie and writes the record to history.txt file.
5. Displays all the Reservations for a specific Movie
6. Deletes Movie entry from the list by shifting arrays and writes the record to history.txt file
7. Displays History after reading it from the file (history.txt)

## **How the project implements CRUD**

#### **C-Create**

* **Add movie:** Users can add a movie using the function addMovie() by entering the movie title, genre and show time.
* **Book Tickets:** allow users to book tickets for a particular movie using the functionbookTickets() and store their names in a 2d array reservationNames and the number of booked seats is also tracked.

#### **R-Read**

* **View Movies:** The viewMovies() function lists all the movie names along with the genre, showtime and available seats.
* **Search Movies:** The search() function allows user to search for a movie either by its name, genre or showtime
* **View Reservations:** The viewReservations() function allows the admin to see the names of all the people who booked tickets for a specific movie
* **View History:** The viewHistory() function allows the admin to View the History

#### **U-Update**

* **Update Seat Count:** The system updates the seat count (bookedSeats[i]++) every time a ticket is booked and adds the reservation name to the array

#### **D-Delete**

* Users can remove a movie from the list using the function deleteMovie() and all the data (movie title, genre and showtime) is deleted by shifting the arrays.

## **Code Snippets**

#### **File Handling**

* Each major action (movie addition, booking, deletion) is logged to a file (history.txt) using **ofstream**:

saveTohistory("Movie added: " + movieTitle[movieCount]);

saveTohistory("User " + yourName + " booked a ticket for " + movieName);

saveTohistory("Movie deleted: " + name);

* The file is then later read by the admin using the **view History** function.

void viewHistory() {

ifstream file("history.txt");

string line;

cout << "\n--- Action History ---\n";

while (getline(file, line)) {

cout << line << endl;

}

file.close();

}

#### **Use of Pointers**

In the add movie function:

void addMovie(string\* titles, string\* genres, int\* times, int& count) {

cout << "Enter movie title: ";

getline(cin, titles[count]);

cout << "Enter movie genre: ";

getline(cin, genres[count]);

cout << "Enter show time(in pm): ";

cin >> times[movieCount];

cin.ignore();

saveTohistory("Movie added: " + movieTitle[movieCount]);

count++;

cout << "Movie added successfully..." << endl;

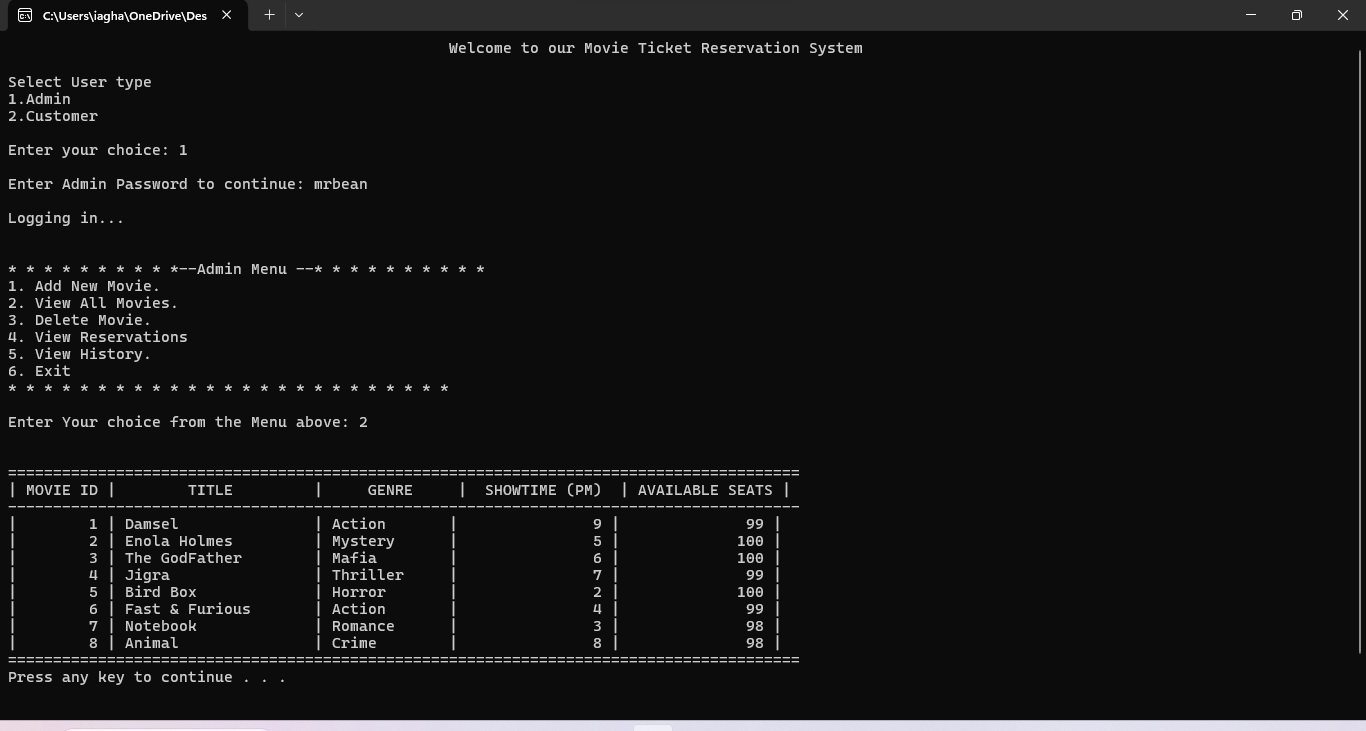
}

## Flow Chart:

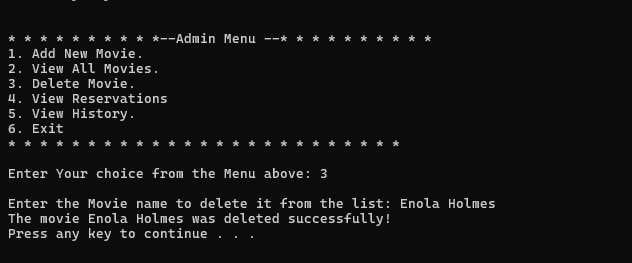
## Screenshots:

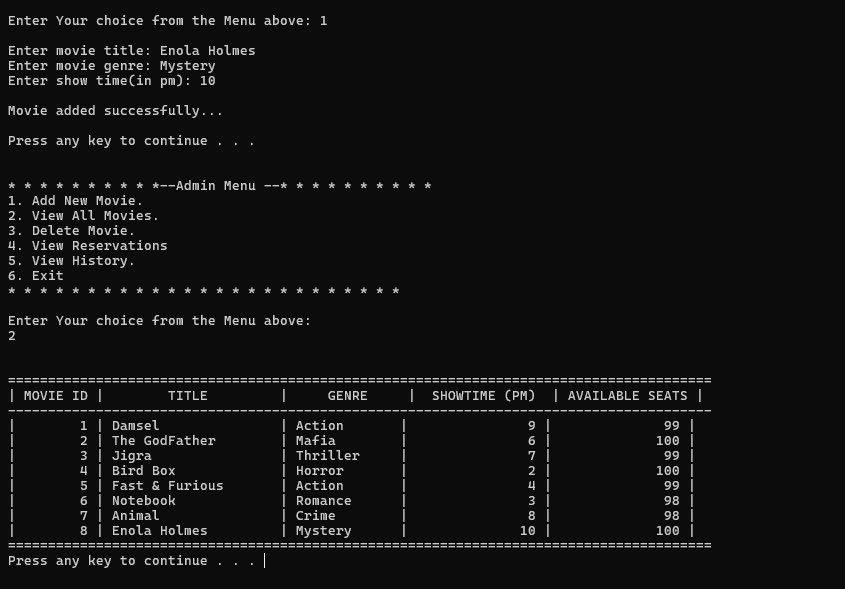
Admin:

* Loggin, menu & View Movies function

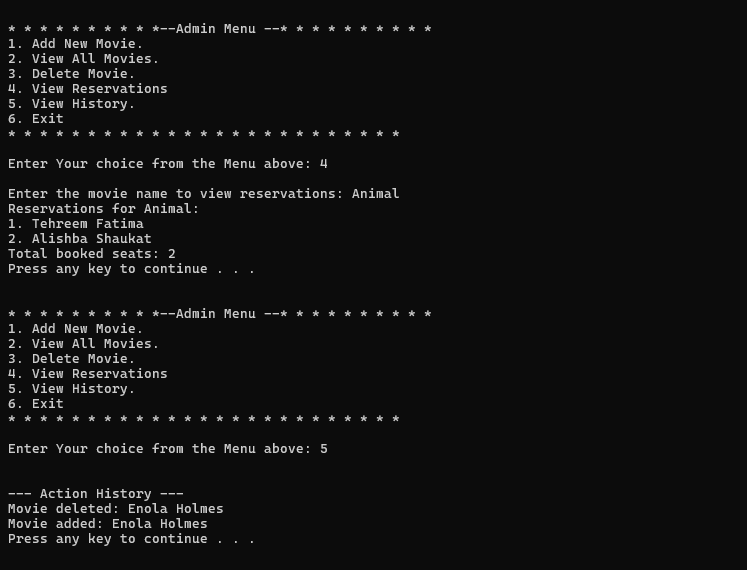


* Add & Delete Movies



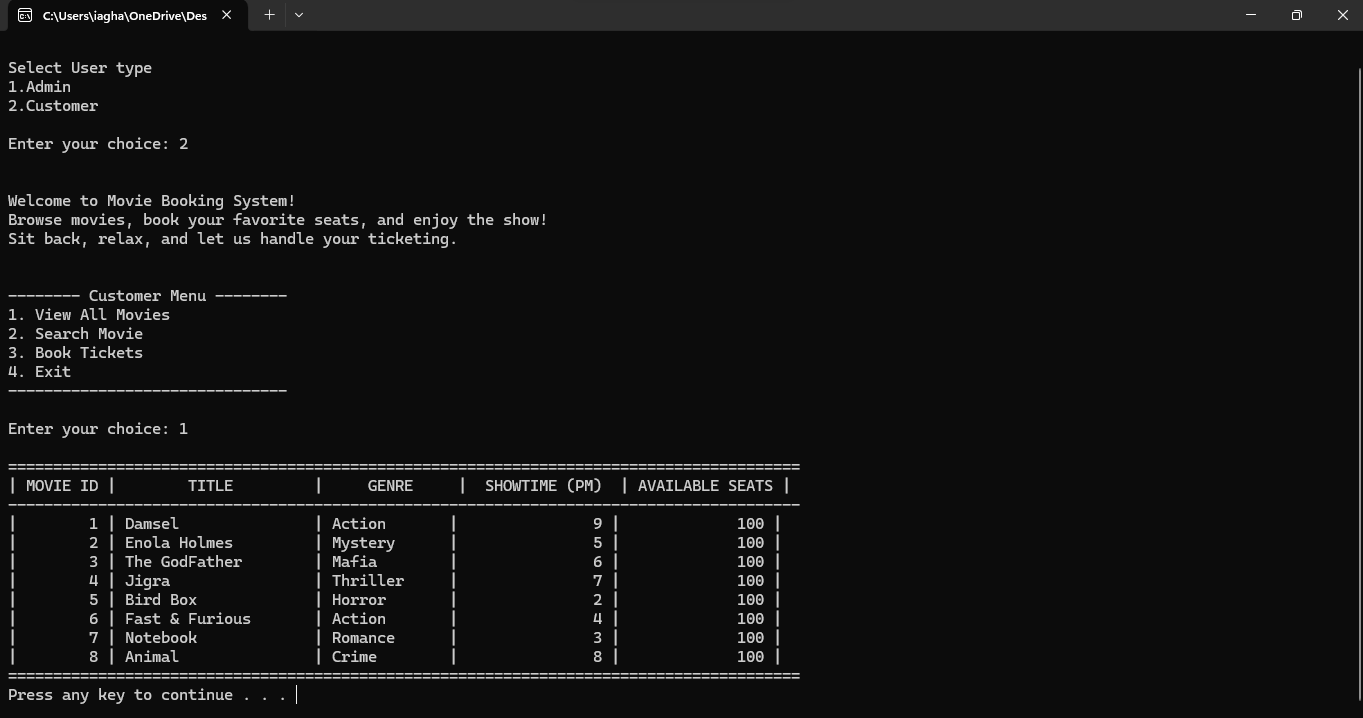


* **View Reservations & History**

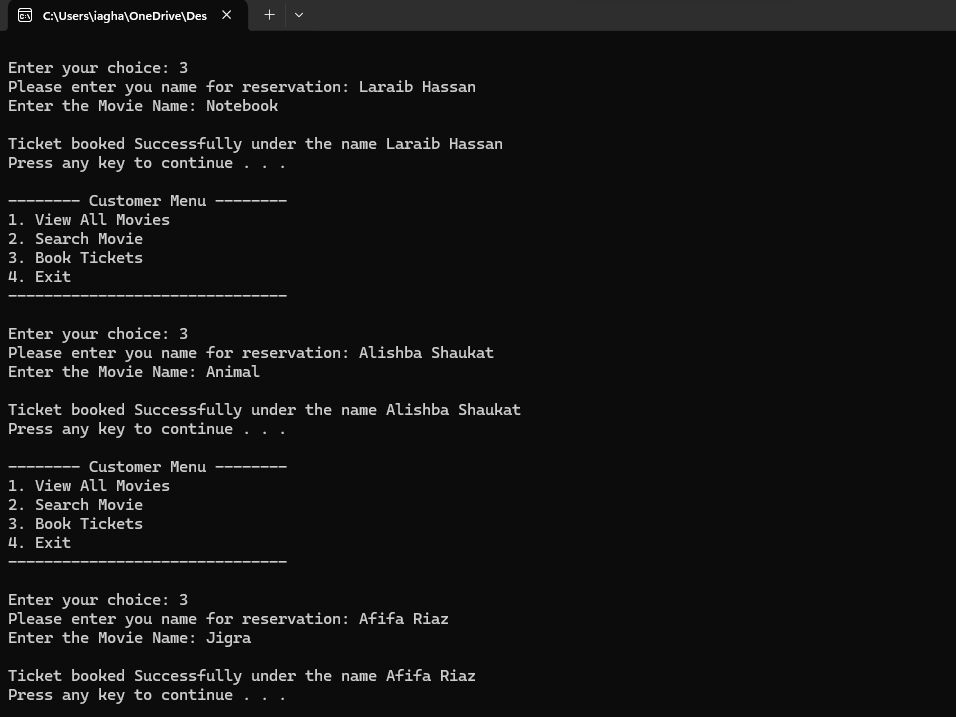
****

**Customer:**

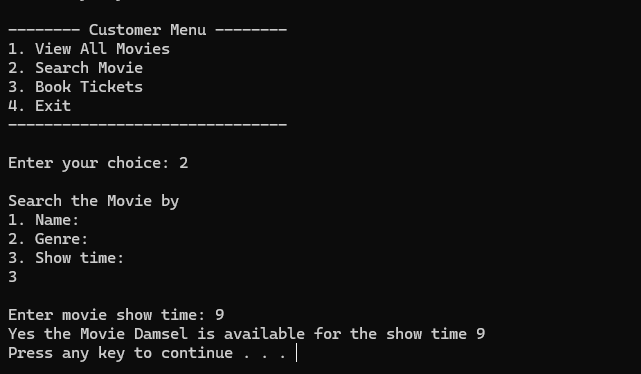
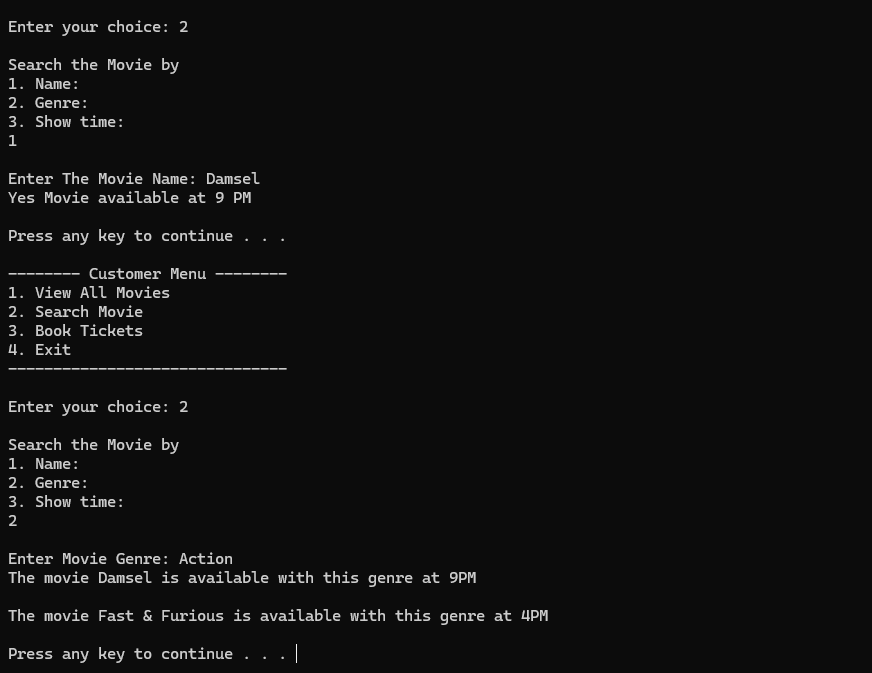
* Menu & View Movies Function



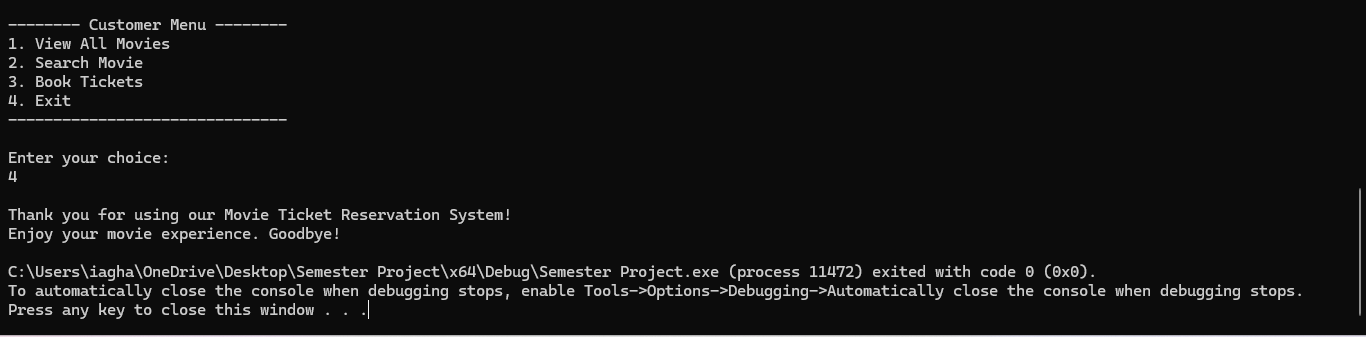
* Book Tickets:



* Searching Movies by Name/Genre/Showtime



* Exit



## GitHub Repository:

Following is the GitHub repository link:

👉 <https://github.com/Iqra039/Movie-Ticket-Reservation-System>

## LinkedIn Project Video:

Following is the LinkedIn video post link:

## Conclusion:

Working on this project helped us understand how to connect programming logic with user actions. We had to think through how to store and update information reliably, manage multiple records using arrays, and ensure smooth user experience. Overall, it was an excellent exercise in problem-solving, system design, and applying classroom knowledge in a meaningful way.