**DATA STRUCTURE & ALGORITHMS**

**DOCUMENTATION**

# **Project Topic:**

**MOVIE TICKET BOOKING SYSTEM**

# **Introduction:**

The Movie Ticket Booking System is a console-based application written in C++. It allows users to interact with a movie ticket booking system where they can book tickets, view available movies, check seat availability, and perform various other operations. The system is designed with the use of linked lists to manage users and movies efficiently.

# **Code Structure:**

The Code is organized into several parts:

## **Structures:**

* **User:** Represents user information, including username, age, and money balance.
* **Movie:** Represents movie information, including name, duration, available seats, booked seats, rating, genre, and ticket price.
* **Node\_User:** Node structure for the linked list of users.
* **Node\_Movie:** Node structure for the linked list of movies.

## **Classes:**

* **UserList:** Represents a stack-based linked list to manage users.
  + Functions: **pushUser, popUser, displayUserInfo.**
* **MovieList:** Represents a linked list to manage movies.
  + Functions: **addMovie, dequeueMovie, displayMoviesAlphabetically, displaySeats, bookSeats, displayMovie, findMovie, editMovie, deleteMovie, mergeSortedLists, mergeSortMovies.**

## **Functions:**

* **createUser():** Gets user input to create a new user.
* **writeUserToFile(const User& user):** Writes user information to a file.
* **displayUserProfileFromFile():** Displays user information from a file.
* **displayMenu(UserList& userList, MovieList& movieList):** Displays the main menu and handles user input.
* **bookTicket(UserList& userList, MovieList& movieList):** Handles the process of booking movie tickets.
* **checkSeats(const MovieList& movieList):** Displays available seats for a movie.
* **editMovie(MovieList& movieList):** Allows editing movie details.
* **deleteMovie(MovieList& movieList):** Deletes a movie from the list.
* **showTicket():** Displays tickets from a file.

## **Helper Functions:**

* **EntryPattern():** Displays an introductory pattern.
* **main():** Entry point of the program. Initializes the UserList and MovieList, adds initial movies, and starts the application.

# **User Interaction:**

Users can interact with the system through a menu-driven interface. They can perform operations such as booking tickets, viewing movies, checking seat availability, and managing user profiles. The system incorporates error handling for invalid data types.

# **Movie Management:**

The System allows Movies to **“Add”, “Edit” and “Delete”** Movies.  
Movies are displayed alphabetically using **“Merge-Sort”.**

# **File Handling:**

User profiles and ticket information are stored in text files ("userprofile.txt" and "ticket.txt", respectively). File operations are handled through functions like **writeUserToFile** and **displayUserProfileFromFile.**

# **Exception Handling:**

The code includes basic exception handling to catch and display runtime errors, ensuring a more robust and user-friendly experience.

# **Developer Options:**

The system includes additional options for developers to **add, edit, and delete** movies, providing flexibility for managing the movie catalog.

# **Introduction Display:**

The program starts with an introductory pattern using the **EntryPattern** function, creating an engaging visual display.

# **Conclusion:**

The Movie Ticket Booking System is a comprehensive console application that demonstrates basic data structures and file handling for managing user and movie information. The modular structure and user-friendly interface make it suitable for learning purposes and potential future expansion.

**FIN.**

Top of Form