

Movie Ticket Booking System Programming

Overview

In this assignment, you will be developing a simple text-based movie ticket booking system using Python. The system will include displaying available movies, booking tickets, and viewing a cart. You'll write several functions to accomplish these tasks.

Function 1: `moviesList()` - 3 pts

Write a function called `moviesList()` that doesn't take any arguments and returns a dictionary representing the list of available movies. Each movie should have a title, a seating chart, and a price per ticket.

Function 2: `display_movies(movies)` - 3 pts

Write a function called `display_movies()` that takes the movies dictionary as an argument and prints the available movies. The display should include the movie ID, title, number of available seats, and the price per ticket.

Function 3: `book_ticket(movies, cart)` - 5 pts

Write a function called `book_ticket()` that takes the movies dictionary and the user's cart as arguments. The function should allow the user to select a movie, choose the number of tickets, and select specific seats. If the booking is confirmed, the selected seats should be marked as booked, and the booking details should be added to the cart.

Steps:

a. **Display Movies:** Call `display_movies()` to show the available movies. b. **Select Movie:** Prompt the user to enter the movie ID they wish to book. c. **Display Seats:** Show the seating chart for the selected movie with available seats marked. d. **Select Tickets:** Prompt the user to enter the number of tickets they wish to book. e. **Select Seats:** Allow the user to select specific seats by entering row and column numbers. f. **Confirm Booking:** Calculate the total price, prompt the user to confirm the booking, and update the cart and seating chart accordingly.

Function 4: `view_cart(cart)` - 4 pts

Write a function called `view_cart()` that takes the user's cart as an argument and displays the current bookings. The function should show the movie titles, number of tickets, price, and seat numbers for each booking. It should also prompt the user to proceed to checkout.

Steps:

a. **Check Cart:** Display a message if the cart is empty. b. **Display Cart:** Show the details of each booking in the cart, including the total price. c. **Checkout:** Prompt the user to proceed to checkout and clear the cart if the payment is successful.

Function 5: `bookmyShow()` - 10 pts

Write a function called `bookmyShow()` that ties everything together and runs the main application loop.

Steps:

a. **Setup:** Initialize the movies list and display a welcome message. b. **Main Menu:** Create a while-loop that continues until the user chooses to exit. The menu should offer options to book tickets, view the cart, or exit. c. **Book Ticket:** Call `book_ticket()` if the user chooses to book a ticket. d. **View Cart:** Call `view_cart()` if the user chooses to view the cart. e. **Exit:** End the application if the user chooses to exit.

Example Walkthrough

1. **Start:** The user launches the application and sees the main menu.
2. **Book Ticket:** The user selects the option to book a ticket.
3. **Select Movie:** The user views the available movies and enters the ID of the movie they wish to book.
4. **Select Seats:** The user chooses the number of tickets and selects specific seats from the seating chart.
5. **Confirm Booking:** The user confirms the booking, and the details are added to the cart.
6. **View Cart:** The user selects the option to view the cart, sees the booking details, and proceeds to checkout.
7. **Checkout:** The user confirms the checkout, and the cart is cleared after successful payment.
8. **Exit:** The user chooses to exit the application.

Note

Ensure the system correctly handles invalid inputs and provides clear feedback to the user. The seating chart should be updated to reflect booked seats, and the cart should display accurate booking details.