OOPS WITH JAVA

CS23333

MINI PROJECT

Movie Ticket Booking System

DONE BY:

NAME: NAVEEN.V

ROLL NUMBER: 231401071

CLASS: CSBS-‘B’

AIM:

The objective of the **Movie Ticket Booking System** project is to develop a user-friendly application that simplifies the process of managing movie bookings. The system allows users to add movies with available seat information, view all available movies, and book tickets efficiently. By automating these operations, the project aims to enhance the user experience, reduce manual errors, and provide a streamlined solution for handling movie ticket reservations in a real-world scenario.

ALGORITHM:

1. **Start**.
2. Display the main menu with the following options:
    - Add Movie
    - View Movies
    - Book Tickets
    - Exit
3. Accept the user's choice.
4. Perform the following based on the choice:
    - **Add Movie**:
        1. Input the movie name and the number of available seats.
        2. Create a new movie object.
        3. Add the movie to the collection.
        4. Display a success message.
    - **View Movies**:
        1. Retrieve and display the list of movies with available seats.
    - **Book Tickets**:
        1. Input the movie name and the number of tickets to book.
        2. Search for the movie in the collection.
        3. If the movie exists:
            - Check if sufficient seats are available.
            - Deduct the booked seats from available seats.

        - Display a success or failure message based on availability.
    4. If the movie doesn't exist, display a "Movie not found" message.
  - **Exit**:
    1. Terminate the program with a goodbye message.

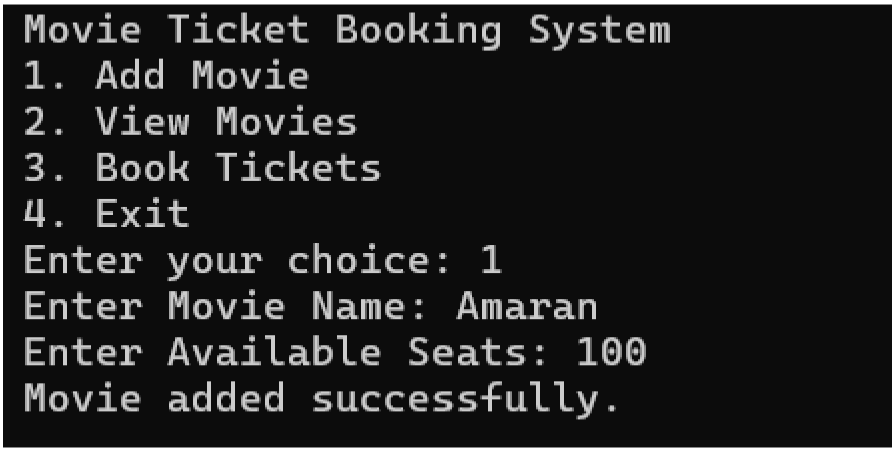
5. Repeat until the user chooses the **Exit** option.
6. **End**.

PROGRAM:

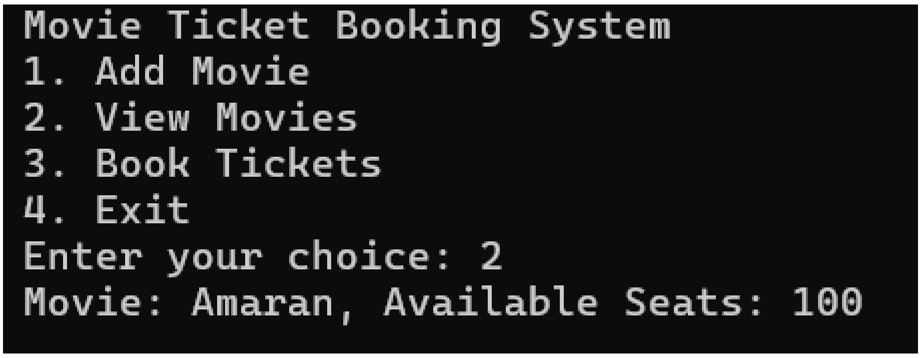
```
import java.util.*;


class Movie {
   private String name;
   private int availableSeats;


   public Movie(String name, int availableSeats) {
      this.name = name;
      this.availableSeats = availableSeats;
   }


   public String getName() {
      return name;
   }


   public int getAvailableSeats() {
      return availableSeats;
   }


   public boolean bookSeats(int numberOfSeats) {
      if (numberOfSeats <= availableSeats) {
            availableSeats -= numberOfSeats;
            return true;
        } else {
            return false;
        }
    }

    @Override
    public String toString() {
        return "Movie: " + name + ", Available Seats: " + availableSeats;
    }
}

class MovieBookingManager {
    private Map<String, Movie> movies = new HashMap<>();

    public void addMovie(Movie movie) {
        movies.put(movie.getName().toLowerCase(), movie);
    }

    public void viewMovies() {
        if (movies.isEmpty()) {
            System.out.println("No movies available.");
        } else {
            for (Movie movie : movies.values()) {
                System.out.println(movie);
            }
        }
    }
    public Movie getMovie(String movieName) {
        return movies.get(movieName.toLowerCase());
    }
}

public class MovieTicketBookingSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        MovieBookingManager manager = new MovieBookingManager();

        while (true) {
            System.out.println("\nMovie Ticket Booking System");
            System.out.println("1. Add Movie");
            System.out.println("2. View Movies");
            System.out.println("3. Book Tickets");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    System.out.print("Enter Movie Name: ");
                    String movieName = scanner.nextLine();
                    System.out.print("Enter Available Seats: ");
                    int seats = scanner.nextInt();
                    scanner.nextLine(); // Consume newline
                    manager.addMovie(new Movie(movieName, seats));
                    System.out.println("Movie added successfully.");
                    break;
                case 2:
                    manager.viewMovies();
                    break;
                case 3:
                    System.out.print("Enter Movie Name: ");
                    String movieToBook = scanner.nextLine();
                    Movie movie = manager.getMovie(movieToBook);
                    if (movie != null) {
                      System.out.print("Enter Number of Tickets: ");
                      int tickets = scanner.nextInt();
                      scanner.nextLine(); // Consume newline
                      if (movie.bookSeats(tickets)) {
                        System.out.println("Booking successful!");
                      } else {
                        System.out.println("Not enough seats available.");
                      }
                    } else {
                      System.out.println("Movie not found.");
                    }
                    break;
                case 4:
                    System.out.println("Exiting...");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
```

```
    }
  }
}
```

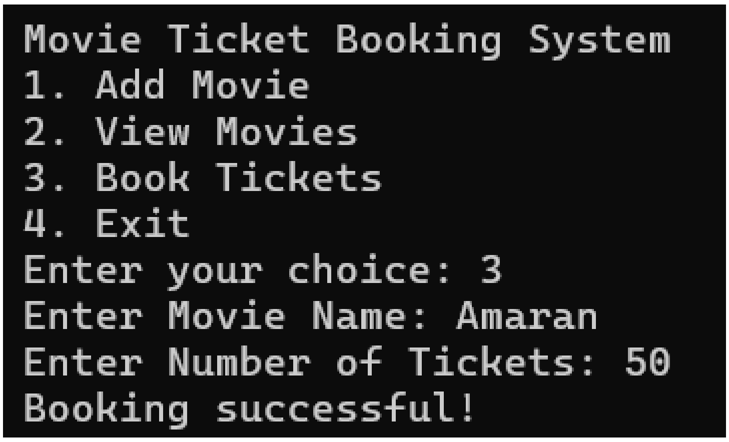
OUTPUT:



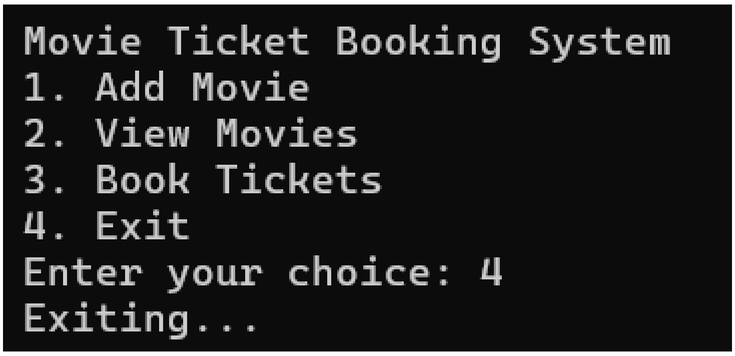
```
Movie Ticket Booking System
1. Add Movie
2. View Movies
3. Book Tickets
4. Exit
Enter your choice: 1
Enter Movie Name: Amaran
Enter Available Seats: 100
Movie added successfully.
```



```
Movie Ticket Booking System
1. Add Movie
2. View Movies
3. Book Tickets
4. Exit
Enter your choice: 2
Movie: Amaran, Available Seats: 100
```



```
Movie Ticket Booking System
1. Add Movie
2. View Movies
3. Book Tickets
4. Exit
Enter your choice: 3
Enter Movie Name: Amaran
Enter Number of Tickets: 50
Booking successful!
```



```
Movie Ticket Booking System
1. Add Movie
2. View Movies
3. Book Tickets
4. Exit
Enter your choice: 4
Exiting...
```

CONCLUSION:

In conclusion, the **Movie Ticket Booking System** successfully provides a streamlined and efficient solution for managing movie ticket reservations. It allows users to add movies, view available movies, and book tickets while ensuring data accuracy and ease of use. This project demonstrates the practical application of object-oriented programming concepts and showcases how software can simplify everyday tasks. It is a scalable and extendable solution that can be further enhanced with features like online payment integration, user authentication, and dynamic seat mapping, making it a valuable tool in the entertainment industry.