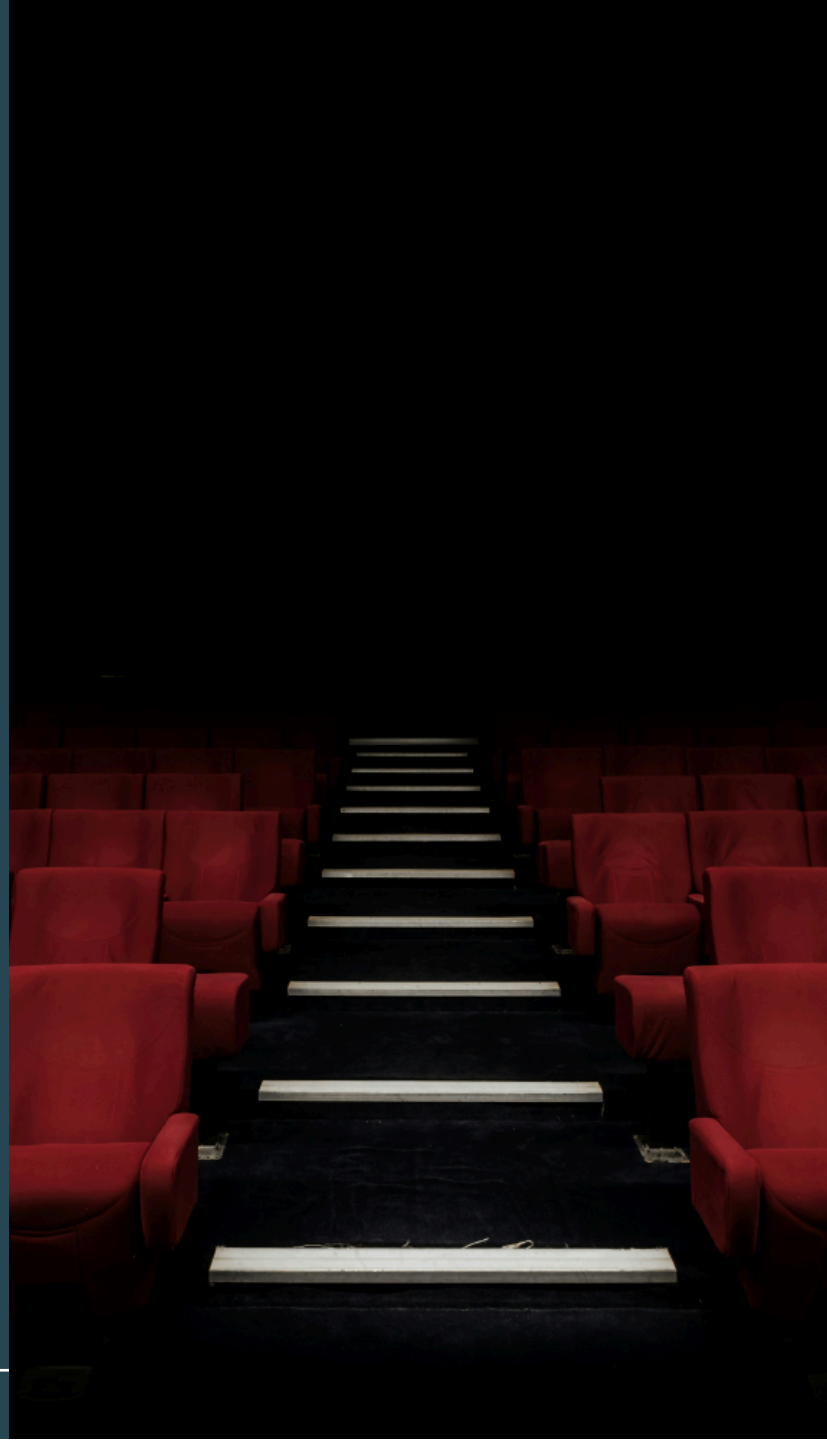
ONLINE MOVIE TICKET BOOKING SYSTEM

By Prasankumar N Gondali
Abhay Sharma

INTRODUCTION

The **Movie Ticket Booking System** is a simple Java-based console application designed to simulate the process of booking movie tickets. It provides users with a list of available movies along with the number of tickets remaining for each.
Users can select a movie, specify how many tickets they want to book, and the system processes the booking if enough tickets are available.
This project demonstrates the use of:

- **Object-Oriented Programming (OOP)** concepts like classes, objects, encapsulation
- **Collections** using ArrayList
- **Control structures** such as loops and conditionals
- **User input handling** using Scanner



```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Movie {
    private String title;
    private int availableTickets;

    public Movie(String title, int availableTickets) {
        this.title = title;
        this.availableTickets = availableTickets;
    }

    public String getTitle() {
        return title;
    }

    public int getAvailableTickets() {
        return availableTickets;
    }

    public void bookTickets(int numberOfTickets) {
        if (numberOfTickets <= availableTickets) {
            availableTickets -= numberOfTickets;
            System.out.println("Successfully booked " + numberOfTickets + "
tickets for " + title);
        } else {
            System.out.println("Not enough tickets available for " + title);
        }
    }
}

class BookingSystem {
    private List<Movie> movies;

    public BookingSystem() {
        movies = new ArrayList<>();
        // Adding some movies to the system
        movies.add(new Movie("Movie A", 10));
        movies.add(new Movie("Movie B", 5));
        movies.add(new Movie("Movie C", 0)); // No tickets available
    }

    public void displayMovies() {
        System.out.println("Available Movies:");
        for (int i = 0; i < movies.size(); i++) {
            Movie movie = movies.get(i);
            System.out.println((i + 1) + ". " + movie.getTitle() + " - Available Tickets: " + movie.
getAvailableTickets());
        }
    }

    public void bookMovie(int movieIndex, int numberOfTickets) {
        if (movieIndex >= 0 && movieIndex < movies.size()) {
            movies.get(movieIndex).bookTickets(numberOfTickets);
        } else {
            System.out.println("Invalid movie selection.");
        }
    }
}

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

        while (true) {
            bookingSystem.displayMovies();
            System.out.print("Select a movie by number (or 0 to exit): ");
            int movieChoice = scanner.nextInt() - 1;

            if (movieChoice == -1) {
                System.out.println("Exiting the booking system.");
                break;
            }

            System.out.print("Enter number of tickets to book: ");
            int numberOfTickets = scanner.nextInt();

            bookingSystem.bookMovie(movieChoice, numberOfTickets);
        }

        scanner.close();
    }
}
```

OUTPUT

plaintext Copy Edit

```
Available Movies:
1. Movie A - Available Tickets: 10
2. Movie B - Available Tickets: 5
3. Movie C - Sold Out
Select a movie by number (or 0 to exit): 1
Enter number of tickets to book: 3
Successfully booked 3 tickets for Movie A
```

Key points✅

Object-Oriented Design

USES CLASS LIKE MOVIE AND BOOKING SYSTEM

MOVIE MANAGEMENT

Stores movie titles and available tickets

TICKET BOOKING FUNCTIONALITY

Allows users to book tickets for a selected movie

User Interaction (Console-based UI)

Takes user input via the command line

Error Handling

Checks for invalid movie selections

How we get there

1.Movie Class

- **Represents a single movie.**
- Stores:
- Movie **title**
- Number of **available tickets**

2. BookingSystem Class

Maintains a **list of movies**.
Adds some movies with pre-defined ticket availability.
Has methods to:

display all movies with available tickets

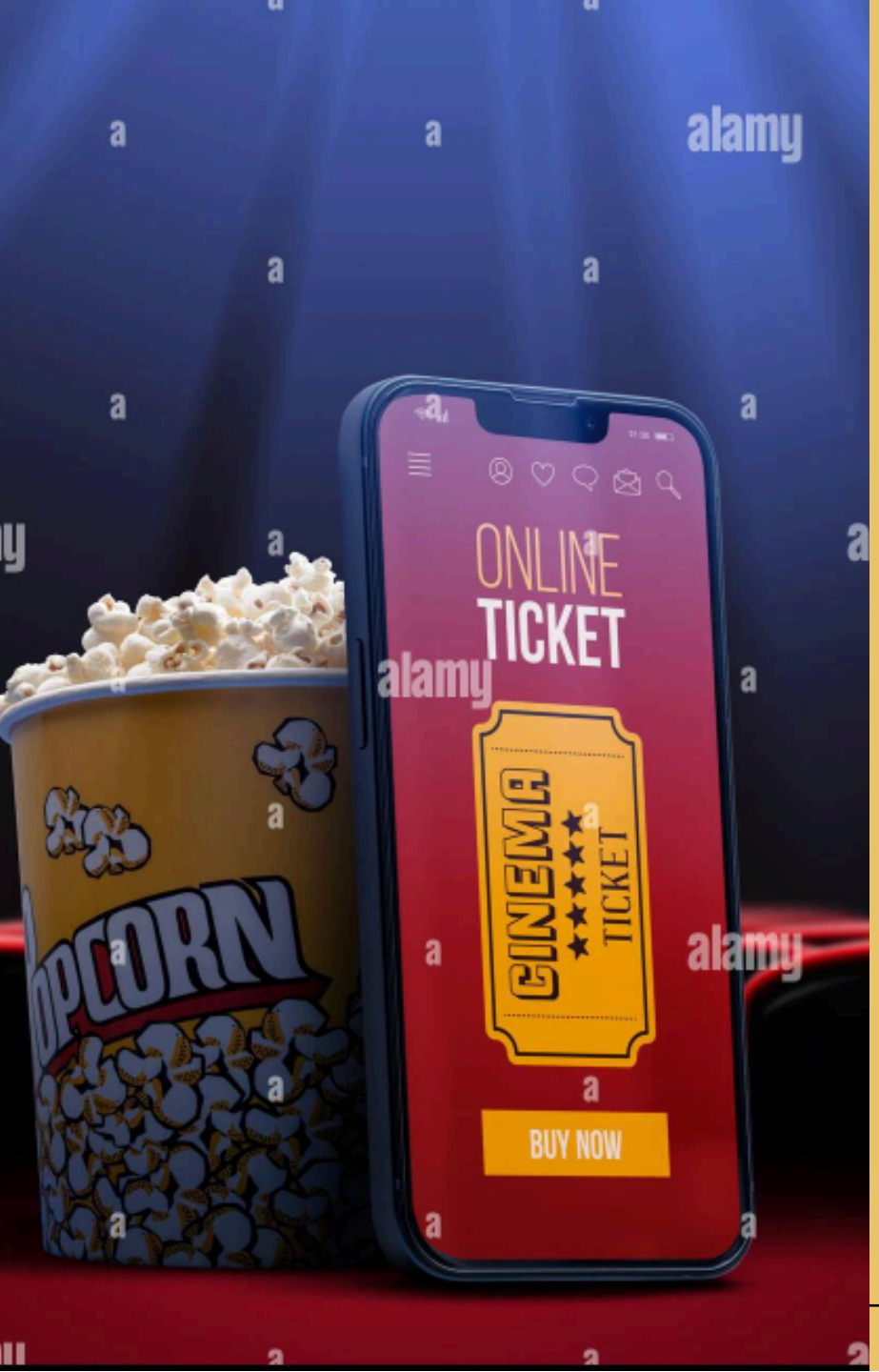
- **Book tickets** for a selected movie.

Ticket Booking Logic

Checks if the requested number of tickets is available.

If yes: books the tickets and shows success message.

If no: shows a message that not enough tickets are available



Summary

A simple Java-based movie ticket booking system.

Built using object-oriented programming with classes and methods

Allows users to view movies and book available tickets.

Includes input validation and user-friendly messages.

Uses a console-based menu with smooth program flow.

Thank you

Prasankumar N Gondali

(92400119112)

Abhay Sharma

(92400119104)

2ET1 Semester-2

Computer Science and engineering with Cyber Security