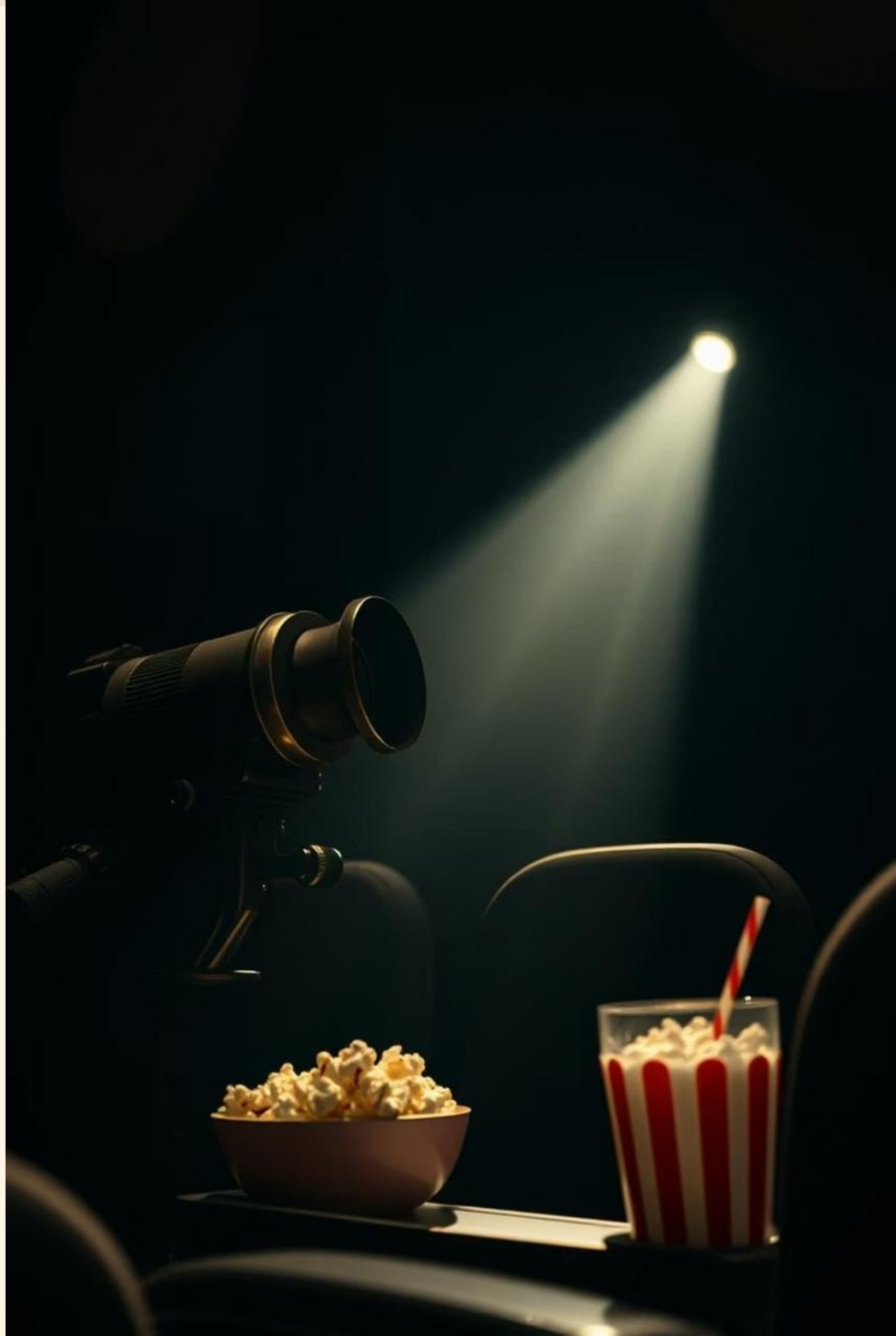
Movie Ticket Booking System

A CLI-based system in Python for booking movie tickets.

Developed by: Sudip Bhandari



INTRODUCTION

- The Movie Ticket Booking System is a Python-based command-line application designed to simplify the process of browsing, booking, and managing movie tickets.
- It supports both regular users and administrators, allowing users to view available movies, check showtimes, select seats, and manage their bookings, while admins can add or remove movies and showtimes.
- The system uses a JSON file for data storage, ensuring portability and ease of use without external dependencies.
- With features like seat maps, loyalty points, and fun Easter eggs, this project demonstrates practical object-oriented programming and user-friendly CLI design.

System Architecture

Login / Register
User authentication

Browse Movies
View available films

View Showtimes
Check schedules

Book / Cancel
Manage tickets

Save to JSON
Persistent storage

.

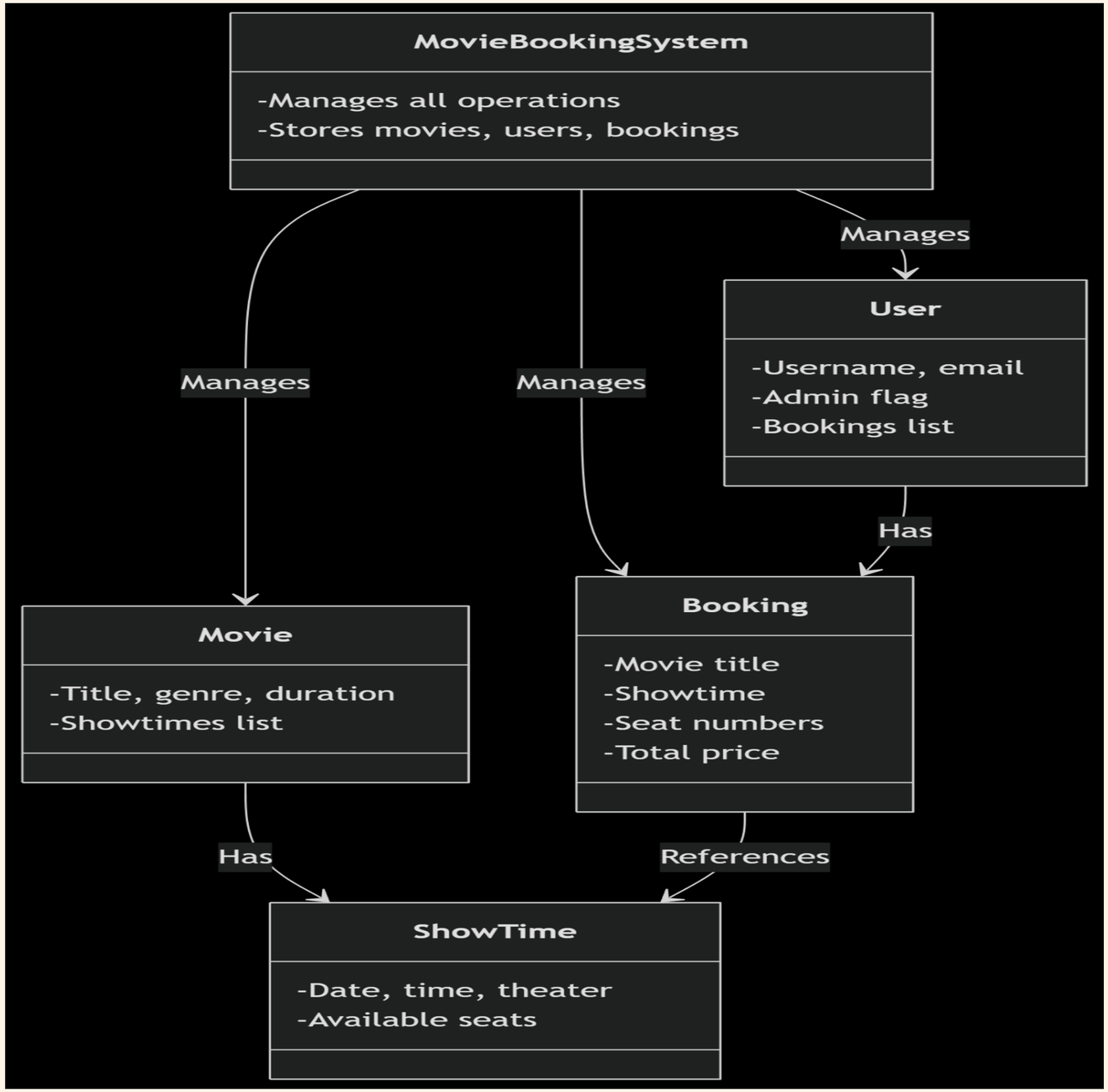


Fig: System Architecture

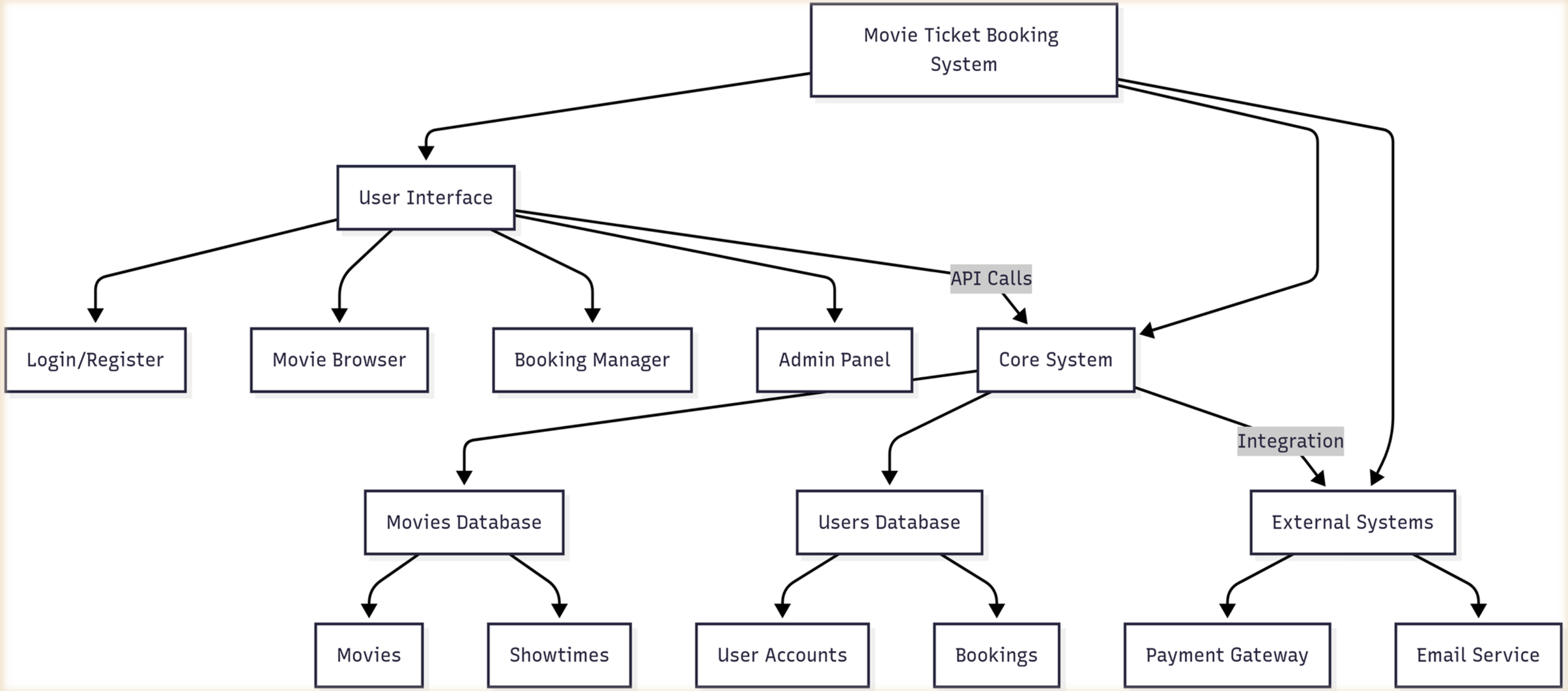


Fig: System Block Diagram

.

SYSTEM ARCHITECTURE

Modules:

- Movie: Movie details (title, genre, price, etc.)
- ShowTime: Showtimes and seat management
- Booking: User bookings and status
- User: User/admin accounts and bookings
- MovieBookingSystem: Main controller (data loading, user actions)

Data Flow:

- User/Admin interacts via CLI
- System loads/saves data from a JSON file (movie_system_data.json)
- All actions (browse, book, cancel, admin ops) update in-memory objects and persist to file

.

Key Features

User Features

Browse Movies

Explore available films 🎬

View Showtimes

Check movie schedules 🕑

Book/Cancel Tickets

Manage reservations 🎫

Booking History & Loyalty

Review past bookings and points 🏆

Admin Features

Add/Remove Movies

Manage film catalog ➕➖

Add/Remove Showtimes

Control screening times 🕐🗑️

View All Bookings

Access user reservation data 📊

User Workflows

User Login

Simple username-based login.

Admin users require a password.

Sample Credentials: Admin: admin / 12345 User: sudip (no password)

Booking Flow

Select Movie

Choose desired film.

Choose Showtime

Pick a convenient slot.

View Seat Map

Available (🎟) vs. Booked (❌).

Enter Seats

e.g., A,1;B,2.

Confirm & Get ID

Finalize booking.

Design Decisions

Why JSON Files?

Simplicity & Portability

No external dependencies

Easy manual inspection

Why Class Structure?

Object-Oriented Design

Modularity & Reusability

Movie ➡️ ShowTimes

Bookings 🔗 Users & Showtimes

CLI Focus

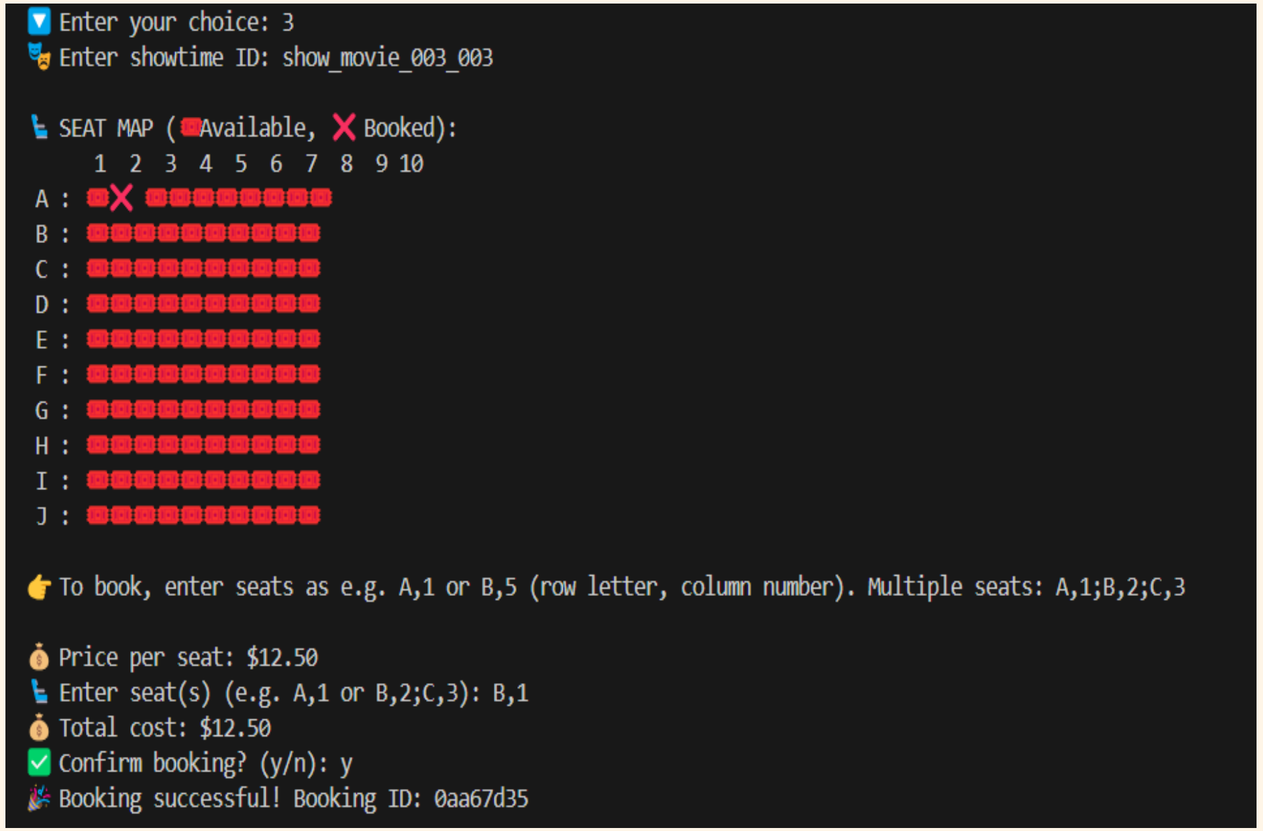
Platform-Independent

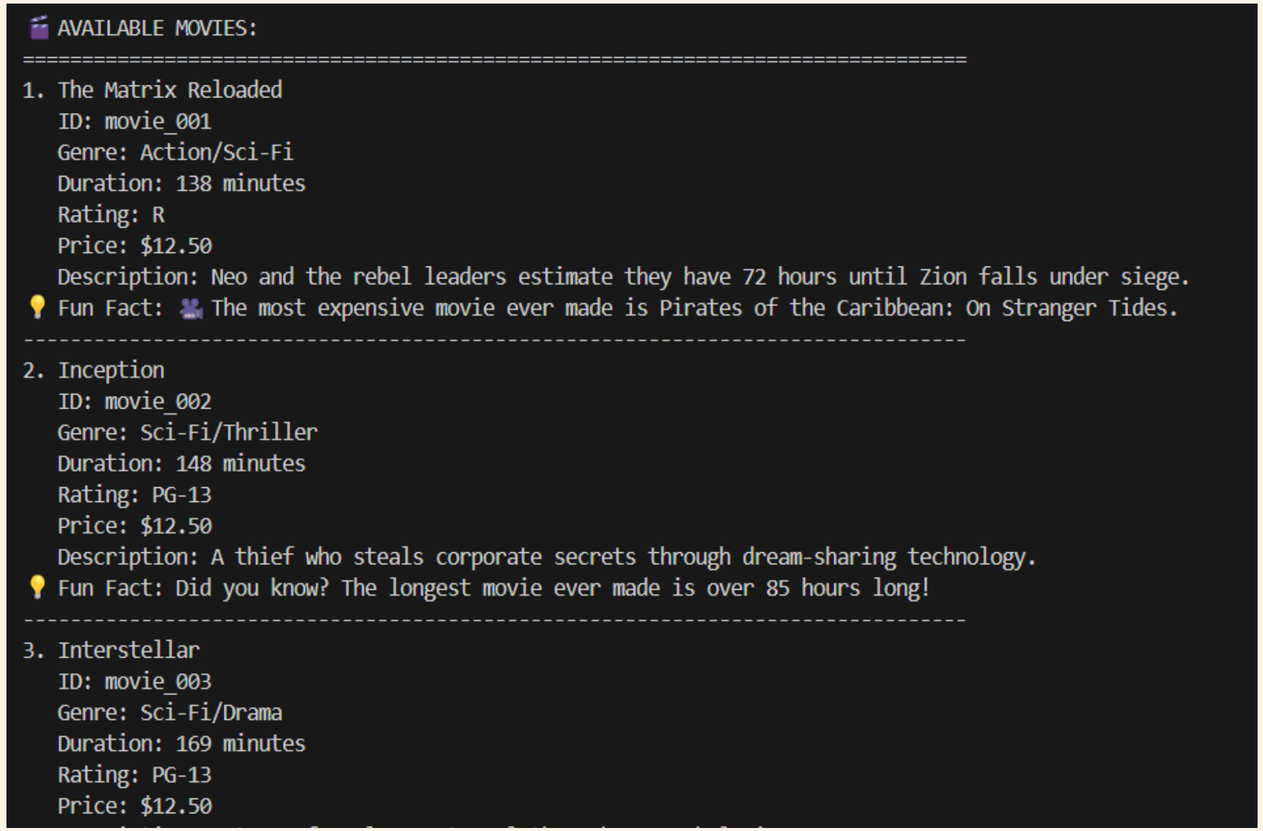
Easy Demo & Testing

Ideal for Learning

Screenshots

Visual examples of the CLI interface:





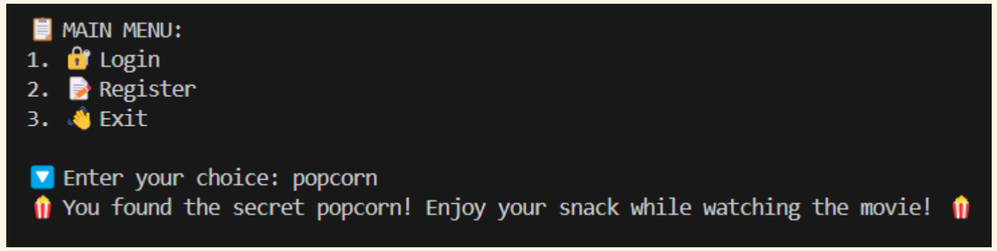
Fun Extras

Loyalty Points

Earn points per seat booked. Users earn loyalty points every time they complete a booking.. This is shown in the "My Bookings" section, where each booking displays the number of seats (and thus points) earned.

Easter Egg

Type "popcorn" for a surprise!

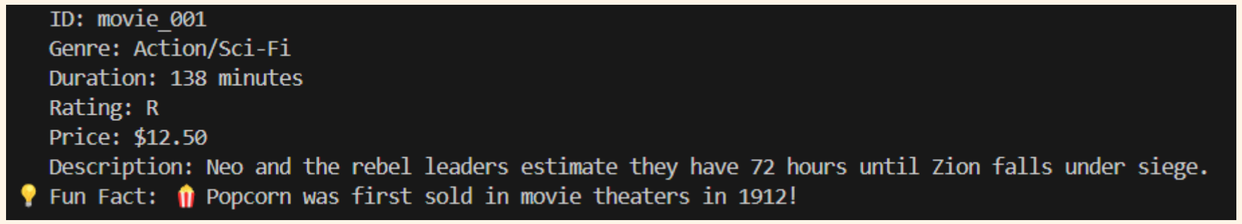


Lucky Draw

Random surprise on booking completion. After a successful booking, there is a mini-game: a random number is generated, and if the result matches a certain value (e.g., 3 out of 1–5), the user receives a surprise message:

Fun Facts

Discover trivia while browsing.



Data & File Handling

`movie_system_data.json`

Centralized storage for:

- Movies
- Users
- Bookings

Auto-initializes with sample data on first run.


```
{
  "movies": [
    {
      "movie_id": "movie_001",
      "title": "Avengers: Endgame",
      "genre": "Action/Adventure",
      "duration": 181,
      "rating": "PG-13",
      "description": "The Avengers assemble once more to reverse Thanos' actions.",
      "price": 15.0,
      "showtimes": [
        {
          "showtime_id": "show_movie_001_0_0",
          "movie_id": "movie_001",
          "date": "2025-07-07",
          "time": "10:00",
          "theater": "Theater A",
          "total_seats": 50,
          "booked_seats": []
        },
        {
          "showtime_id": "show_movie_001_0_1",
          "movie_id": "movie_001",
          "date": "2025-07-07",
          "time": "13:30",
          "theater": "Theater B",
          "total_seats": 50,
          "booked_seats": []
        },
```


I/O Operations

All input/output handled using Python's built-in json module.

Ensures data persistence between sessions.

```
import jsondef load_data(): with open('movie_system_data.json', 'r') as f: return json.load(f)def save_data(data): with open('movie_system_data.json', 'w') as f: json.dump(data, f, indent=4)
```

Requirements & Setup

3.7+ Python Version Required

No External Libraries

Run Instructions:

```
python "Movie_Ticket_Booking_system.py"
```

Thank You!

Your feedback is welcome!