Theatre Management System

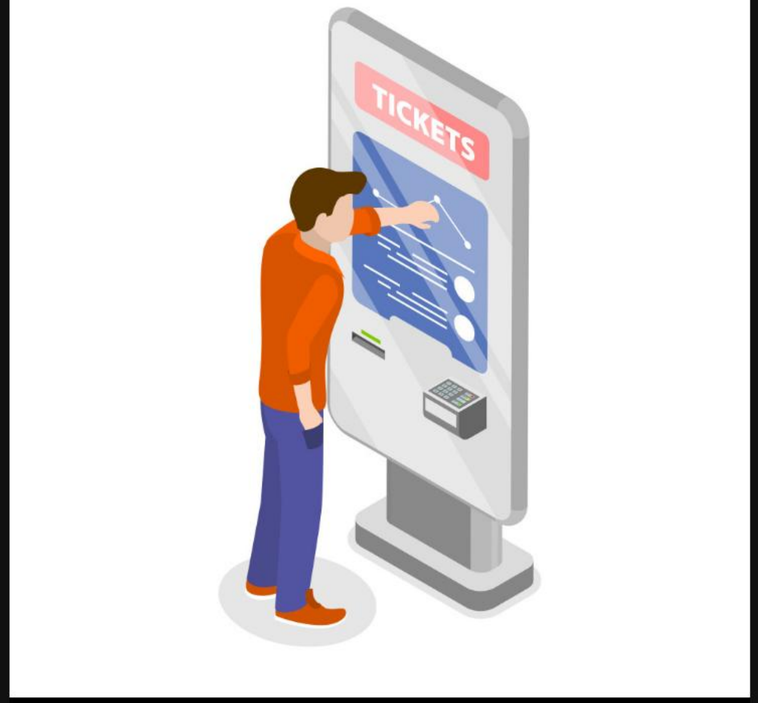
A Comprehensive Code Report & Analysis

```
> python run_theatre.py_
```

System Overview

The Theatre Management System is a robust, console-based application designed to streamline the process of movie ticket bookings.

- **Console Interface:** Simple, text-based user interaction loop.
- **Show Management:** Handles multiple movie titles with specific showtimes.
- **Dynamic Booking:** Real-time seat tracking and updates.



Code Architecture

Understanding the Object-Oriented Design

The `Show` Class

The `Show` class acts as the blueprint for individual movie sessions. It encapsulates the state of a specific screening.

- **Attributes:** Stores `title`, `time`, `total_seats`, and `booked_seats`.
- **Identity:** Each instance represents a unique movie slot.
- **Representation:** Implements `__str__` for formatted string output.

Core Functionality

book_seats(n)

Validates if `n > 0` and checks seat availability. Updates the `booked_seats` counter upon success.

cancel_seats(n)

Ensures that the number of seats to cancel does not exceed the currently booked amount before deducting.

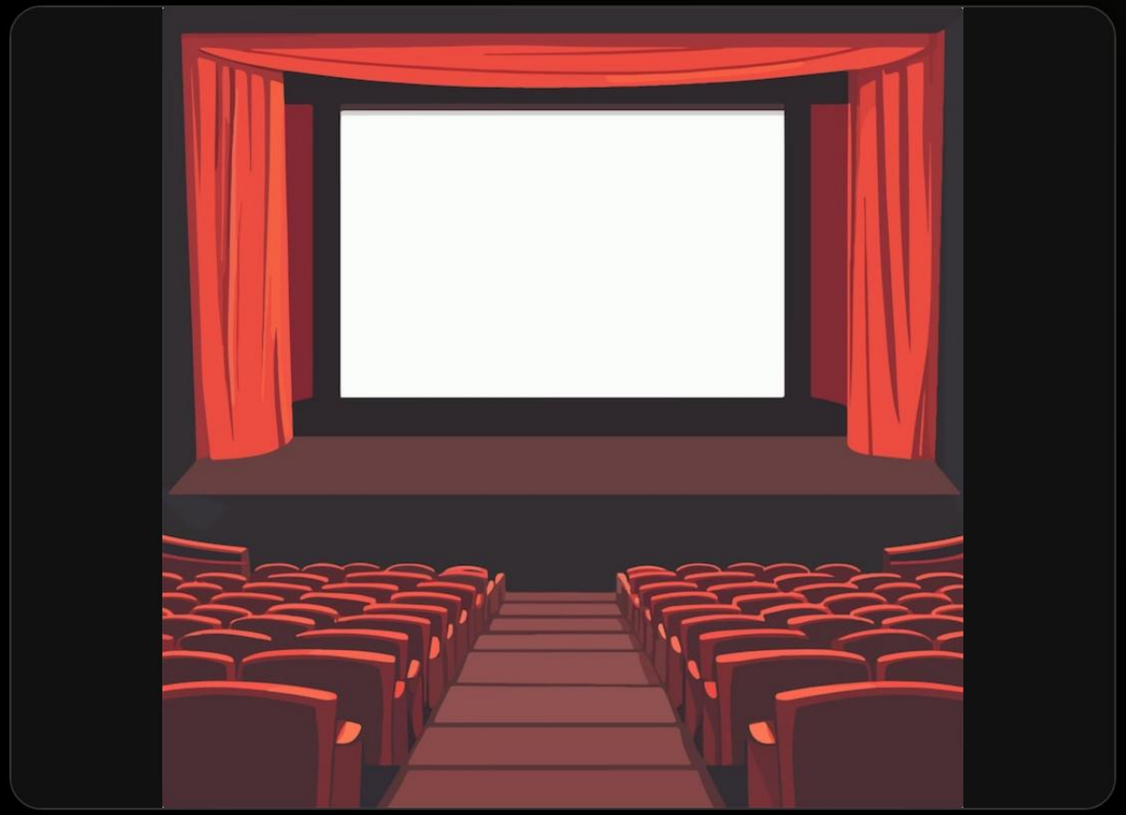
available_seat()

A helper method that returns the difference between `total_seats` and `booked_seats`.

The `Theatre` Class

The `Theatre` class functions as the manager or container. It aggregates multiple `Show` objects and provides an interface to manipulate them.

- **Collection:** Maintains a list `self.shows = []`.
- **Search Logic:** Iterates through the list to find shows by title for booking or cancellation.
- **Delegation:** Delegates the actual seat logic back to the specific `Show` instance.



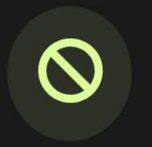
User Interface Loop

1. View Shows

Calls `theatre.display_shows()` to list all currently running movies and their availability.

2. Book Tickets

Prompts user for movie title and seat count, then calls `theatre.book_ticket()`.



3. Cancel Tickets

Allows users to reverse a booking by specifying the movie title and number of seats.

Robustness & Validation

- **Input Validation:** Checks for non-positive integers (e.g., trying to book -5 seats).
- **Error Handling:** Uses `try-except ValueError` blocks to prevent crashes when users enter text instead of numbers.
- **Case Insensitivity:** Converts user input and titles to lowercase (`.lower()`) to ensure "RRR" matches "rrr".
- **Logic Safety:** Prevents booking more seats than available or cancelling more than booked.

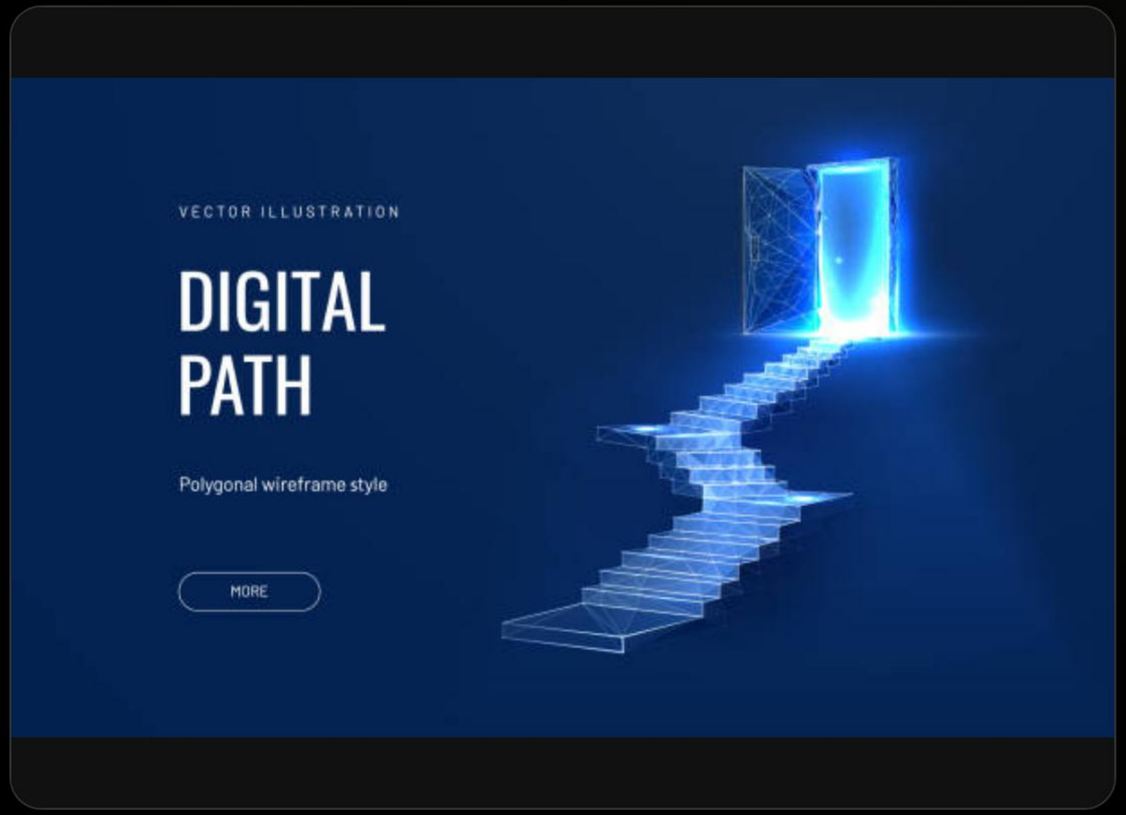
Initialized Show Data

Movie Title	Duration	Total Seats
Sholay	3 hrs 48 mins	380
RRR	3 hrs 02 mins	380
KGF	2 hrs 48 mins	380
Pushpa	3 hrs 17 mins	380

Future Improvements

While functional, the system can be expanded with enterprise-level features:

- **Persistent Storage:** Integrate SQL or JSON storage to save bookings after the program exits.
- **Graphical GUI:** Replace the CLI with Tkinter or a web interface.
- **User Accounts:** Implement login systems for distinct customers.
- **Seat Mapping:** Allow selection of specific seat numbers (e.g., A1, B2).



"The code demonstrates a clean, modular object-oriented approach to solving real-world management problems."

— Code Analysis

Questions?

Thank you for reviewing the code report.

Image Sources

https://cdn.vectorstock.com/i/1000v/42/58/3d-isometric-flat-set-of-ticket-kiosks-vector-48404258.jpg

Source: www.vectorstock.com

https://iisecurity.in/images/courses/source-code-analysis.webp

Source: iisecurity.in

https://img.freepik.com/premium-vector/modern-movie-theater-stage-with-cinema-theatre-scene-big-screen_1322206-76226.jpg?semt=ais_hybrid&w=740&q=80

Source: www.freepik.com

https://media.istockphoto.com/id/1398867347/vector/open-door-at-digital-path-futuristic-science-fiction-concept-of-doorway-technology-portal-in.jpg?s=612x612&w=0&k=20&c=pMc9g58h3-2J-NEUZHmXtjUbyUCx-h8dP-pWvzJ0xlA=

Source: www.istockphoto.com