Swami Keshvanand Institute of Technology,
Management & Gramothan, Jaipur

PROJECT KIT

Title of the Project
Online Ticket Reservation System for Cinema Halls

Abstract of the Project

The **Online Ticket Reservation System** aims to streamline ticket booking for cinema halls by providing a seamless, feature-rich platform for users to reserve movie tickets. The system automates ticket reservations, prevents double bookings with real-time seat availability tracking, and sends email confirmations for successful reservations. It simplifies operations for cinema hall administrators by integrating features for movie and schedule management. Secure and a user-friendly interface ensure a modern, efficient movie ticket booking experience.

Generic keyword:

Online Ticket Booking, Cinema Reservation System, Seat Availability Management, Real-Time Ticket Updates, User Dashboard, Admin Panel, Email Notifications, Seat Mapping, Customer Experience Enhancement, Booking Automation, Cinema Management System.

Specific Technology keywords:

HTML, CSS, React.Js, MongoDb, Node.Js, Web Application, Database Management, API Development.

Functional Components:

1. **User Registration and Authentication**
    - Allows users to register and log in securely.
    - Social media login support for convenience.
2. **Movie Listings and Showtimes**
    - Displays current and upcoming movies with details such as synopsis, ratings, and showtimes.
    - Filters for date, time, and genre.
3. **Interactive Seat Selection**
    - Real-time interactive seat map displaying availability.
    - Differentiation of seat categories (e.g., VIP, Regular).
4. **Ticket Booking and Confirmation**
    - Sends detailed email confirmations with booking ID, QR code, and seat details.
5. **User Dashboard**
    - Displays booking history and user profile management.
6. **Admin Panel**

- Tools for movie schedule and seat management.
- Generates reports on revenue, user activity, and seat occupancy.

7. **Email Notifications**
   - Sends booking confirmations and reminders.
8. **Real-Time Availability**
   - Dynamic seat updates to prevent double bookings.

Functionality

Users of the System:

- **Customers**: Book movie tickets, view booking history, and receive notifications.
- **Administrators**: Manage movies, schedules, reservations, and monitor reports.

Core Functionalities:

- Secure registration, authentication, and profile management.
- Intuitive browsing and filtering for movies and showtimes.
- Interactive and synchronized seat selection.
- Admin tools for movie, schedule, and booking management.
- Reporting dashboard for performance insights.

Steps to Start-off the Project

1. Familiarize yourself with the core technologies (React, Node.js/Django, MySQL/PostgreSQL).
2. Research cinema management and user booking behavior.
3. Define user personas and user stories for clarity.
4. Design a user-friendly and responsive interface with accessibility considerations.
5. Create an API design document with endpoints for user and admin functionality.

Requirements

Hardware Requirements:

Number	Description	Alternative (if Available)
1.	Minimum requirements- Processor, x86-64 bit CPU	
2.	Ram - 4Gb, Disk Space - 5Gb.	

Software Requirements:

Numbers	Descriptions	Alternatives (if Available)
1	Development Environment: VS Code, Node.js	IntelliJ IDEA
2	Database Management: MySQL/PostgreSQL	SQLite
3	Client OS: Windows/Linux	macOS

Manpower requirements:
2 to 3 students working part-time for 4–6 months.

Milestones and Timelines

Number	Milestone Name	Milestone Description	Timeline (Week)	Remarks
1	Requirements Specification	Document detailed specifications and use cases.	Week 2-3	Ensure all functional and non-functional requirements are covered.
2	Technology Familiarization	Learn the technologies required for implementation.	Week 4	Focus on understanding practical application.
3	Database Creation	Design and implement the database schema.	Week 5-6	Validate the schema with use cases.
4	High-Level and Detailed Design	Create flowcharts and design the API.	Week 7-8	Include error handling in design.
5	Front-End Development	Implement the login and movie browsing interfaces.	Week 9-10	Begin preparing test cases.
6	Back-End Development	Develop APIs and integrate the database.	Week 11-12	Test APIs for scalability.

7	Integration Testing	Conduct thorough testing of the integrated system.	Week 13-14	Address bugs and finalize system for demo.
8	Final Review	Ensure all requirements are met and prepare for demo.	Week 15-16	Document any remaining issues and propose solutions.

Guidelines and References

- **React Documentation**: https://react.dev/reference/react
- **Node.js Documentation**: https://nodejs.org/docs/latest/api/
- **Database Design**: Navathe, “Database Management Systems”
- **Software Engineering Concepts**: Ian Sommerville, "Software Engineering"