

Author

Pranav Kumar Iyengar

21f1006080

21f1006080@ds.study.iitm.ac.in

Diploma Student at IITM online degree

Description

This project provides a platform where users can book tickets for shows in various theaters, leave reviews for shows, and view summaries of show ratings and top reviewers. Admins can get summarized data on various metrics.

Technologies used

Flask: A micro web framework written in Python.

Flask-SQLAlchemy: Used for ORM (Object-Relational Mapping) functionalities.

SQLite: As a lightweight database for development.

Flask-JWT-Extended: For creating JWTs (JSON Web Tokens) for user authentication.

Vue.js: Progressive JavaScript framework for building user interfaces.

Bootstrap: Front-end framework for designing responsive web pages.

Purpose of these technologies

Flask provides a quick and easy way to set up a web application. Using Flask-SQLAlchemy simplifies database operations, and Flask-JWT-Extended allows for secure user authentication. Vue.js provides a modern way to build dynamic front-ends, and Bootstrap ensures the UI is responsive and aesthetic.

DB Schema Design

User: Contains user details like ID, username, password.

Show: Details of different shows, such as ID, name, rating.

Theatre: Info on various theatres, including ID, location.

Booking: Records user bookings, with fields like ID, user ID, show ID, number of tickets.

Tag: Stores tags for shows.

Review: Stores user reviews for shows, fields include ID, user ID, show ID, rating.

Reasons for Design

The database is normalized to reduce data redundancy. Separate tables for users, shows, theatres, bookings, and reviews ensure easy maintainability. Foreign keys are used to establish relationships between tables, ensuring data integrity.

API Design

The system provides APIs for:

User registration and login.

Creating, reading, updating, and deleting shows and theatres.

Making bookings and writing reviews.

An admin summary endpoint to fetch metrics like total users, top shows, and top reviewers.

Architecture and Features

The project follows the MVC (Model-View-Controller) architecture. Flask is used for the backend, handling API routes (controllers), interacting with the SQLite database (models), and rendering views. Vue.js is used for building dynamic frontend components (views).

Controllers: Located in server/routes.py.

Models: Defined in the server/models.py.

Views: Vue components are structured in client/src/components.

Features

Default Features: User registration and login, CRUD operations for theatres and shows, making bookings, leaving reviews.

Additional Features: Admin dashboard providing summarized metrics, top shows by bookings, top reviewers, and average ratings. Celery tasks.

Video

https://drive.google.com/file/d/1p_PwcAp5DXjehU1U2XGZP_rTn0XfID8a/view?usp=sharing