Design and Implementation of a ticket booking website

07-08-2023
MAD 2

Irshad.A.Sareshwala

Introduction

The project encompasses the design and implementation of a ticket booking website, employing a variety of technologies and tools to ensure high performance, security, and user-friendly experience. The system comprises a backend developed using Flask and SQLite, and a frontend designed with Vue.js. The intermediary between the frontend and backend is a set of APIs, which are used for various operations like create, read, and update data. To enhance the performance of these APIs, caching has been implemented with the help of Redis.

Backend Design:

The backend of the system relies on Flask, a micro web framework written in Python, and SQLite, a C library that implements a small, disk-based database engine. Flask-SQLAlchemy and ORM (Object-Relational Mapping) have been used to interact with the database. The backend consists of several data tables:

Admin: This table stores information about the admin users, such as ID, name, email, and password.
User: User information, such as ID, email, password, name, phone number, and address, is stored in this table. The User model also includes functions for password hashing and verification.
Role: This table manages the roles of users in the system.

Venue: This table keeps records of the venues, including their ID, name, location, and capacity.
Show: Information about the shows, like ID, name, time, venue ID, available seats, price, rating, date, poster, synopsis, and description, is stored in this table.
Booking: This table stores booking details, including ID, user ID, show ID, number of tickets, and timestamp.
Ratings: This table manages the ratings given by users to the shows.
Frontend Design:

The frontend has been developed using Vue.js, a progressive JavaScript framework used for building user interfaces. The design incorporates a Single Page Application (SPA) structure, reducing server load and providing an efficient and seamless user experience. Vue Router has been used to enable navigation between different components. The aesthetic appeal of the frontend has been enhanced using CSS and Bootstrap containers. This design approach makes it easy for users to distinguish between different shows and venues and understand which show is running in which venue.

Security and User Management:

Security is a critical aspect of the system. JWT (JSON Web Tokens) has been used to secure user data. JWT not only helps to distinguish between users and admin but also provides Role-Based Access Control (RBAC). This way, unauthorized users are prevented from

booking tickets, thus ensuring the integrity and privacy of user data.

Performance Optimization:

To optimize the performance of the APIs, caching has been implemented using Redis, an in-memory data structure store. When a GET request is made from the frontend to the backend, the data is retrieved from the cache or the database. If the data is fetched from the database, it is also stored in the cache for future use. This approach significantly reduces the response time, leading to an improved user experience.

Conclusion:

The ticket booking website leverages several technologies and design principles to provide a secure, efficient, and user-friendly platform for users to book tickets for shows. The use of Flask and SQLite for backend, Vue.js for frontend, APIs as an intermediary, and Redis for caching, along with JWT for security, contributes to a robust and high-performance system. The system is designed keeping in mind both the user experience and system efficiency, and it demonstrates a good example of a full-stack web application.

Link for my video Presentation
https://drive.google.com/file/d/13WfM3EQncfcT1kzAdcUGBtufSz2n7jO2/view?usp=sharing