App Development 2 Project

May 2023 Term

1. Author

Name : Saranath P

Roll Number : 21F3002841

Student Email : 21f3002841@ds.study.iitm.ac.in

About me:

Hello, I am Saranath currently pursuing a BS in Data Science and Application at IIT Madras where I have cleared AppDev 2 theory course. I am a programming enthusiast and enjoyed this App Development where I got to know about how web applications are created and the process that involves in creating complex models which are currently very much required in the industry. I also understood how scheduling with batch processing works and how secure an app should be. I have learnt how frontend works and how to connect a standalone backend with a UX frontend.

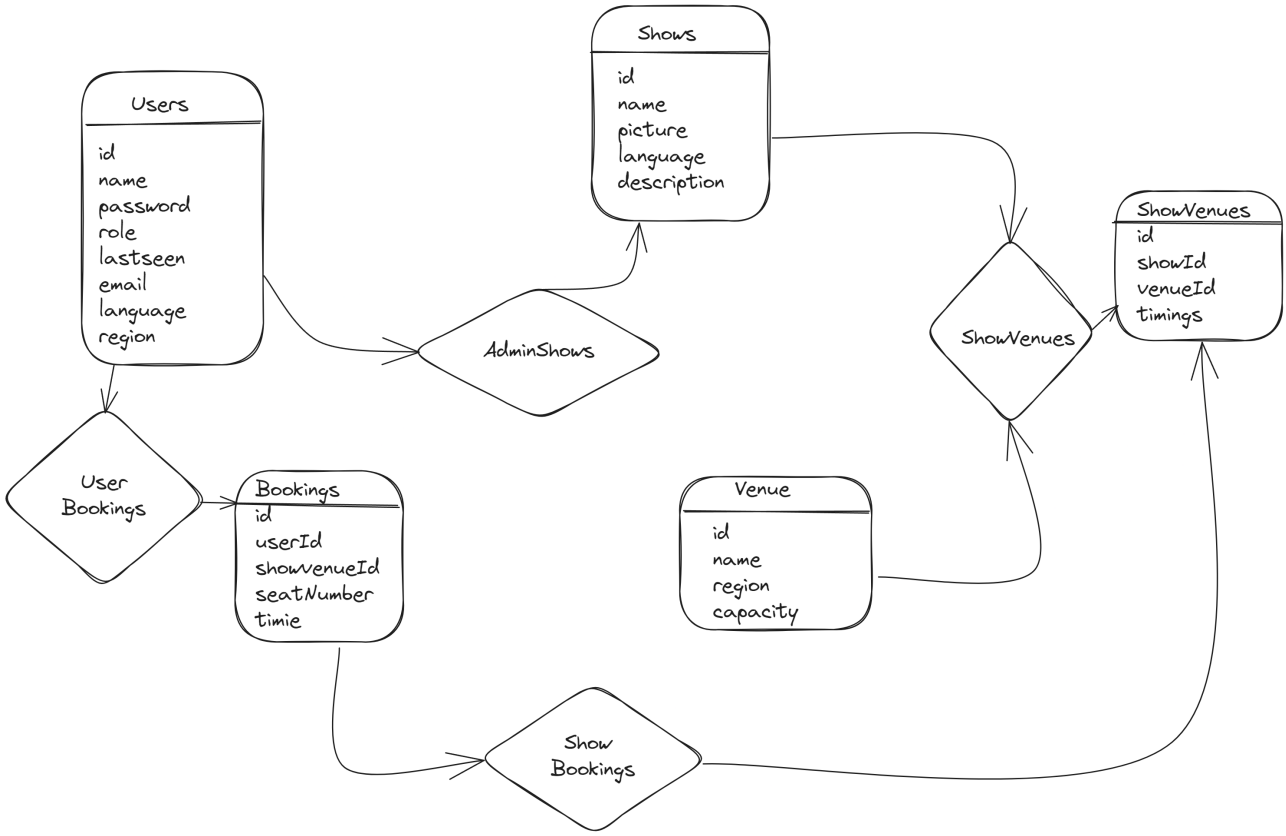
2. Description about my project

The basic idea of the project is to develop a Ticket Booking application which involves $admins$ to create movies, theatres and add them respectively and $users$ who can able to book tickets to a particular show. This can be achieved by RBAC (Role Based Authentication). The app must also be secured and proper hashing is to be done to achieve this agenda.

3. Technologies Used

- **Flask** : Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided.
- **SQL-Alchemy** : SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL. It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.
- **Vue.js** : For frontend and JavaScript for the application.
- **JWT** : JavaScript Web Tokens are used for achieving RBAC.
- **Celery** : A Celery system can consist of multiple workers and brokers, giving way to high availability and horizontal scaling.
- **Redis** : Redis (*Remote Dictionary Server*) is an in-memory data structure store, used as a distributed, in-memory key–value database, cache and message broker, with optional durability. Redis supports different kinds of abstract data structures(http such as strings, lists, maps, sets, sorted sets, HyperLogs, bitmaps, streams, and spatial indices.
- **SMTP**
- **Python3.10**
- **SQLite3**

4. DB Schema Design



5. API Design:

There are CRUD operations for Shows, Theatres, Bookings and Users tables where I create, read, update and delete for all. I have added sufficient endpoints for it and it is working fine. These are fetched by the frontend and displayed according to the type of user (normal or admin) based on their role.

6. Architecture and Features :

The architecture is simple and do not involve any complex processes. User can book tickets based on the availability of shows and the timings that they like. User can also see the tickets they booked for their reference. If not logged in for more than 1 day, a mail is sent to the user enquiring about their issue with the application. A monthly report is also sent to the user based on their booking details. There are dynamic pricing where if the popularity of the show in a particular theatre increases, then the show price also increases. For the admins, we have colorful graphs which can be used to visualize the booking details of the user.

7. Video Link

Demonstration Video