MOHD ARIZ SIDDIQUI
21F1002275

Modern Application Development II

Project – Ticket Show v2

Introduction

I am a B Tech Computer Science and Engineering graduate (graduated Jul 2023) from SRM Institute of Science and Technology. I have decent working experience in major fields in computer science such as web development, application development and ML/AI. As such, I've also completed an internship an Gahan AI Pvt Ltd in a collaboration with NVIDIA Corporation. Currently, I'm interning as a Software Apprentice as Techjockey Infotech Pvt Ltd, and will be leaving for Masters in the US at the University of Texas at Dallas in January 2024.

Description

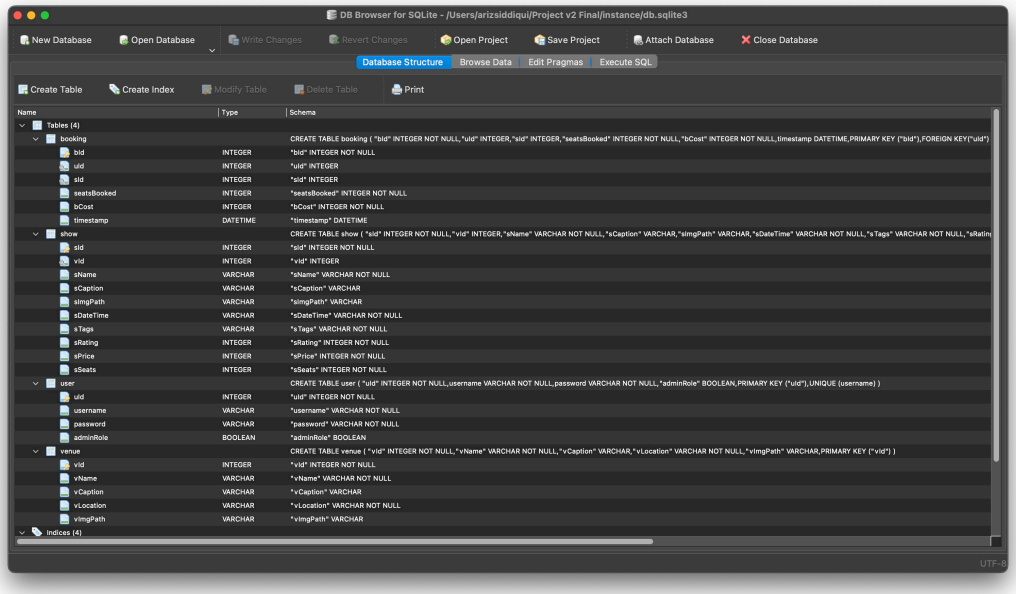
In this project, a ticketing system for movies was required to be constructed, where a user can browse and book movie tickets at a given venue. They can also check and delete their bookings. An administrator account and add, update and delete individual venues and shows and request the venue data to be exported as a CSV file.

Technologies Used

- Flask
- Vue 3
- Flask-SQL Alchemy
- Pinia
- Vue Router
- Flask-CORS
- Celery
- Flask-Mail
- Flask-Login
- Flask-Caching

DB Schema Design

The relevant data is stored in a database consisting of 4 tables: User, Venue, Show, Booking. Image files are received in base64 format, decoded and saved in the filesystem, with a uniquely generated name for each venue and show. The file path to this image file is then stored in the DB. The DB Schema is outlined in the image below:



API Design

The API handles all data-related requests in the application. It receives and sends the data in JSON format, with the relevant status codes. It also maintains a separate store of user information to make sure unauthorised access of DB functions is not possible. It also sets up background tasks using Celery (using Redis as a broker). It also implements a Cache storage to speed up recurring data retrievals.

Architecture and Features

The project was named ‘Toto Movies’, after a friend of mine who helped me fix bugs in the project.

The project can be run in two ways: Integrated Mode and Cross-Origin mode.

In Integrated mode, just run the main Python API file (‘main.py’) and the program works as intended.
In Cross-Origin Mode, the Vue3 front-end can be run on a separate server by running the command ‘npm run dev’ in the project’s command line. Before that do run ‘npm I’ to make sure all dependancies are installed.

The Vue 3 Frontend works on Vue3’s Composition API framework and uses SFCs to break down page structure. The frontend uses JavaScript for interactivity, HTML for layout, and a custom-made CSS library forked from W3.css for styling and aesthetics. The design is highly minimalistic, with the main motive to be simple yet stylish.

The production version code is saved in the ‘dist' folder, while the development source code, for reference is stored in the ‘src’ folder. The instance folder contains the SQLAlchemy database based on SQLite3.