Project Report

Ticket Show v2

Author:

- **Name:** Sushmithasumukhi N
- **Roll no:** 21f2001423
- **Email:** 21f2001423@ds.study.iitm.ac.in
- **About me:** I'm in my 3rd year BTech ECE from UVCE, Bangalore University and pursuing BSc in Data science from IIT-Madras and I'm at the diploma level

Description:

TicketShow is a web-based application made with Vue3 and Python-Flask. It enables multiple admins to create shows in respective theaters. It enables multiple users to browse through the shows and book tickets. It facilitates users to search the shows based on genre and search theaters based on location. It also allows the user to voice their opinions on shows by providing reviews.

Technologies Used:

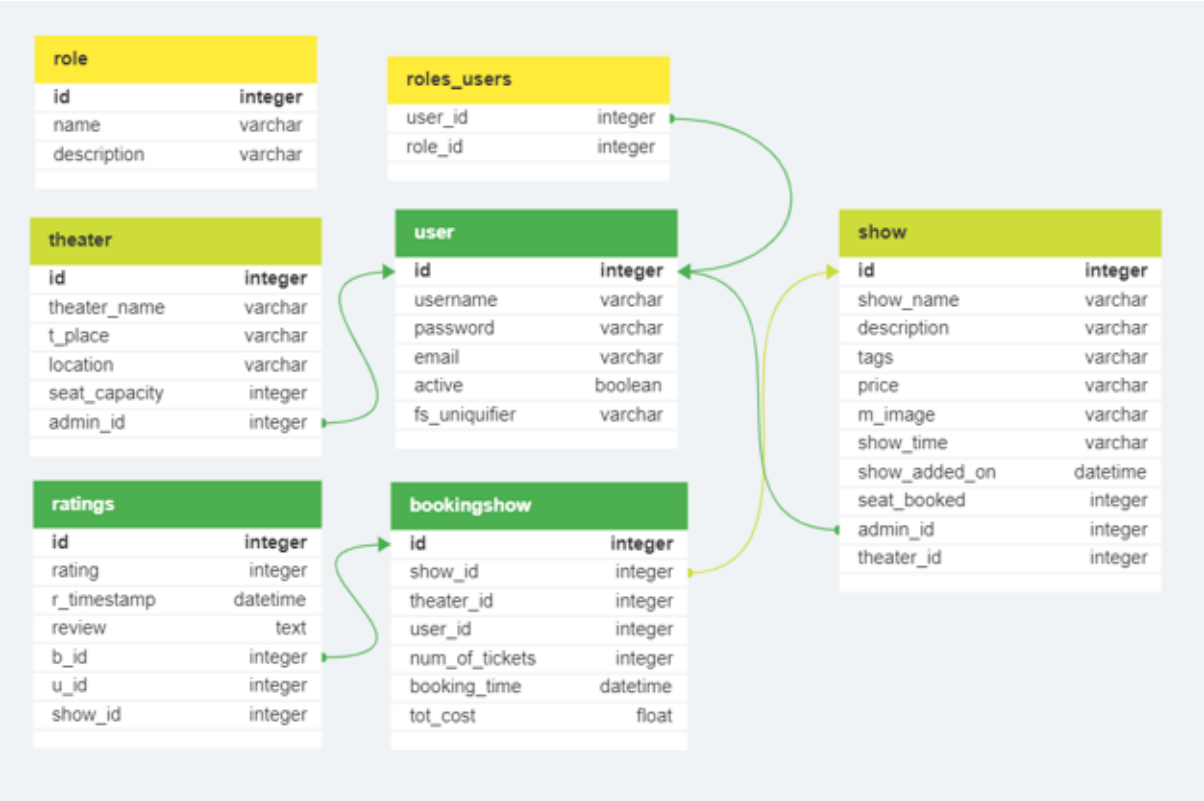
- **Python:** responsible for developing the controllers and serving as the host programming language for the app.
- **Vue:** used to develop the front-end of the app
- **CSS:** responsible for styling the web-pages
- **Bootstrap:** used to make the front-end appealing and easy to navigate
- **SQLite:** serves as the database for the app
- **Flask:** serves as the web-framework for the app
- **Flask-Security:** used for authentication and role based access control
- **Flask-Restful:** used to develop the RESTful API for the app
- **Flask-SQLAlchemy:** used to access and modify the SQLite database
- **Flask-Caching:** used for caching API outputs and increasing performance
- **Flask-Celery:** used for async background jobs at the backend
- **Redis:** used as an in-memory database for the API cache and as a message broker for celery
- **MailHog:** used for mailing purpose when the background job is triggered.

Database schema:

The schema of the database consists of seven tables. **user, show, theatre, ratings, bookingshow** are the main tables.

Relationships between the tables are defined as:

- Theater <1-n> show
- Show <1-n> ratings
- Show <1-n> bookings



API Design:

A REST API for Ticket show app was made with Flask-RESTful module. It has:

- Endpoints for basic CRUD operations on theaters, shows only for admin
- Endpoints for ratings, bookings for users
- Other endpoints for frontend rendering

Authentication tokens and roles are used for specific requests that requires them. These tokens are obtained as json data with query param of include_auth_token in flask-security. The role authentication is implicitly handled by Flask-security.

Architecture and Features:

The architecture of Ticket-Show follows client-server model, where Vue servers as the frontend framework and Flask as the backend framework. Vue handles the presentation layer

and manages the user/admin interactions, while Flask handles the server-side logic, such as HTTP requests and responses, async tasks and database interaction.

The features are as follows:

- **User authentication:** Signup and Login
- **Admin authentication:** Admin Login

ADMIN

- **Show management:** Create, view, update and delete shows
- **Theater management:** Create, view, update and delete theaters
- **Data Export:** Download theater's details and stats as CSV file from mail

USER

- **Explore shows:** View shows for booking based on search results of genre
- **View theaters:** View theaters based on location searches
- **Voice Opinions:** Users can provide review and ratings for the shows booked
- **Users profile:** User details and reviews provided by them are shown.
- **Reminders:** Receive daily reminders to book shows
- **Monthly entertainment report:** Receive a report as an email summarizing the bookings for the month

Video

Click here to watch the video

https://drive.google.com/file/d/1xcmQNzma9pCFoYb141zcmY20We5Azwcc/view?usp=sharing