Author

Satyam Kumar
21f2000243
21f2000243@ds.study.iitm.ac.in

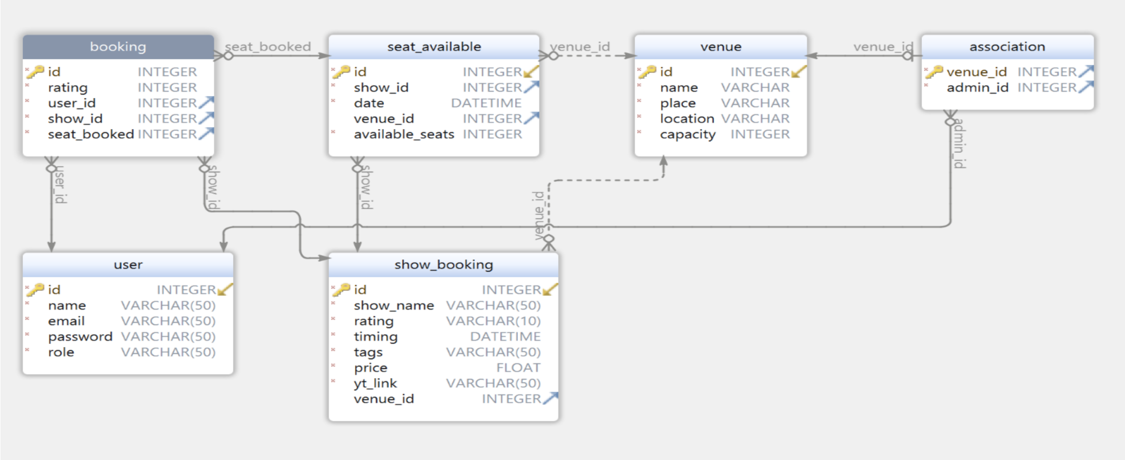
Description

This project allows hosts to manage shows and venues, and users to book tickets. It has a search function with filters, and displays shows based on their timeframe. When seats are full, it displays that the show is houseful.

Technologies used

- Flask: A web framework for Python that is used to develop the backend of the application.
- Flask_SQLAlchemy: An extension that is used to facilitate communication between Flask and the database.
- Flask_RESTful: An extension that is used to implement RESTful APIs for the application.
- Flask_Login: An extension that provides user authentication functionality.
- Flask_CORS: An extension that enables Cross-Origin Resource Sharing (CORS) for the application.
- Flask_Swagger_UI: An extension that is used to create API documentation.
- Matplotlib: A library that is used to create data visualisations. It is required for the visualisation.
- Bootstrap for styling .
- Sqlite for data storage.

DB Schema Design



A Venue can host multiple ShowBookings, while a ShowBooking can only have one Venue. A User can make many Bookings, and a Booking can only belong to one User. A User can have multiple roles, and a Venue can have multiple Users with different roles, which is implemented through the Association class. A ShowBooking can have one SeatAvailable, and a SeatAvailable can belong to one ShowBooking. Finally, a Booking can be associated with one SeatAvailable, and a SeatAvailable can have many Bookings.

API Design

The API supports CRUD for shows and venues, with two Flask-RESTful resources: VenueResource and ShowBookingResource. VenueResource supports GET, POST, PUT, and DELETE requests for venues, while ShowBookingResource supports the same for shows, along with creating a new seat available object. Both resources use Flask-RESTful's reqparse and fields modules for data validation and serialization, and the abort function for raising HTTP exceptions when necessary.

Architecture and Features

The TicketShow project is organised into several folders and files, including a root directory with database files named instance, template and static folder. The static folder contains static files like images, CSS, yml and CSV files, while the templates folder contains HTML templates for various application pages. The model.py file contains database models, while the app.py file contains the Flask application and controllers. The resources.py file contains resources for APIS . The project implements essential booking app features using various technologies and programming languages.

Core features:

- Admin login and User login through flask-login
- Venues & Shows Management
- Booking of show tickets.
- Search for shows/venues . Also , user can search shows /venues based on tags ,location ,ratings etc

Additional features:

- Different price for each venues
- Export venue/show engagement (number of tickets booked,venue performance)
- Dynamic Pricing for top 5 shows which have a high number of bookings and time is less than 6 hrs.
- Rating - The rating feature is available for users who have booked tickets to a show, and the show time has already passed.

Video

https://drive.google.com/file/d/1eMgGVEgHolJj8hiG5_xUhnX9G6-7p-4p/view?usp=sharing