Author

Name: Ishita Saxena
Roll no. : 21f006334
Email id: 21f006334@ds.study.iitm.ac.in
Economics major from University of Delhi with a minor in Commerce, currently pursuing BS Data Science and Application at Indian Institute of Technology, Madras.

Description

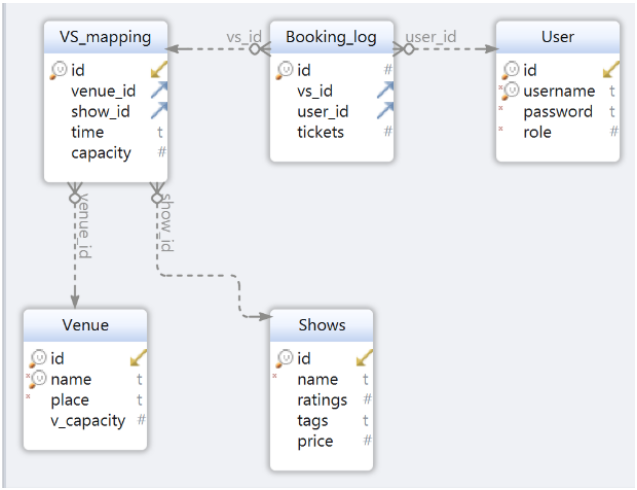
Ticket Show is a multi-user app in which users can book tickets, search for movies based on location of the venue, movie name, genres, etc. This app also has an admin interface where the admin can perform CRUD operations on shows and venues.

Technologies used:

Following technologies have been used in this project.

- Python: Primary Programming Language used in the project
- Flask: Its python web framework used to support the development of app.
- Flask SQL Alchemy: Extension of flask which is a SQL-Alchemy wrapper. It adds support for SQL-Alchemy.
- SQL-Alchemy: Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL. It is used to access and manipulate the database via python.
- Jinja-2: This is a templating engine that is utilized to render data sent by APIs and controllers onto an HTML document.
- SQLite: It's used for data storage.
- DB-Browser: It's a GUI app, used to create and interact with database.

DB Schema Design



The database has following tables:

1. **Users**: which includes basic user information like username and password, the role column make sures whether the user is a customer or an admin. Value of role :1 corresponds to admin and 0 corresponds to customer
2. **Shows**: Consists of basic information about shows, as shown in the diagram above
3. **Venues**: Consists of basic information about the venue as shown in the diagram above

4. **VS_mapping**: This table maps the relation between shows and venue, a venue can have multiple shows. It has two foreign keys venue_id and show_id which refer to the primary key of Venues and Shows table respectively.
5. **Booking_log:** This table keep tracks of the bookings made by user, it has one foreign key "vs_id" which references to primary key of VS_mapping table.

API Design

The project consists of two API end points

- VenueApi : which supports CRUD operations on Venue, Can also create a show at a particular venue.
- VenueShowApi: which helps in deleting a show from a venue or making changes to it.
- DashboardApi: which displays all the shows and venues at the users' feed.

Architecture and Features

The project consists of two folders and one main.py file in the root directory. The main.py file has all the configurations and consists of functions to start the app, controllers, etc. The folder "database" contains the database that is used in the project. The application folder consists of api.py," which has all the API functions,
database.py," which creates an SQL-Alchemy object, and models.py," where all the models are defined, which correspond to different tables in the database.

Following Features have been implemented;

1. Login and registration pages for users and administrators use the role value in the table to determine whether the person attempting to log in is a user or an administrator.
2. The user dashboard displays all the shows at all the venues.
3. The user can book multiple tickets at different venues.
4. The booking function gets disabled if there are no seats available.
5. The user can search for shows based on names, location of the venue, genres, and ratings.
6. The admin dashboard displays all the shows at all venues, and all of them have edit, delete, and add buttons.
7. Admins can delete, create, and update a venue.
8. Admins can create a show and add it to multiple venues.
9. Admins can delete a show at a particular venue, change its timing, name, price, etc.

Video

https://drive.google.com/file/d/14UNKTZ1xQr8UsMAULpTe23nO97d0AthS/view?usp=share_link