Author

Himadri Dixit

21f1006310

21f1006310@ds.study.iitm.ac.in

Description

Build a multi-user(admin/user) ticket booking application with different functionality for admin and users. Users can book tickets, rate and search for shows whereas admin can apply CRUD on both venues and shows.

Technologies used

1. Flask: Web framework
2. VueJS: Rendering web pages
3. Jinja2: Template Engine
4. Bootstrap: Layout/Styling
5. SQLite: Database
6. SQLAlchemy: ORM
7. Flask-Restful: REST APIs
8. Flask-Security: Authentication
9. Flask-werkzeug: Security
10. YAML: API Spec
11. Python: Language
12. WeasyPrint: Converts HTML reports to PDF format
13. Redis: Asynchronous batch jobs/caching
14. Celery: Asynchronous batch jobs
15. FakeSMTP: to receive emails
16. Pandas: Export data as CSV files

DB Schema Design

1. Table for:
    1. Venue: Details like name, place, capacity, shows running.
    2. Show: Details like name, venue, rating, tags, price, timing, tickets remaining.
    3. User: Details like email, password.
    4. Role: Available roles in the database.
    5. Role_users: Roles assigned to users.
    6. Booking: Details of booking like user, show, number of tickets, rating.
2. One to Many relationship between show and venue.
3. One to Many relationship between show and booking.
4. Roles are updated in the role_users table automatically at the time of registering.

API Design

I implemented:

1. 9 GET endpoints.
2. 5 POST endpoints.
3. 2 DELETE endpoints.
4. 2 PUT endpoints.
5. 2 PATCH endpoints

I can

1. CRUD on venues and shows (admin-only)
2. Create and read user and admin
3. Get history of booking for a particular user
4. Stop taking bookings if the capacity is reached for that show.
5. Display shows in latest added order for each venue.
6. Search for shows on the basis of name, tags, rating and location of the venue.
7. Get a summary for each venue (admin-only)
8. Rate a show (user-only)
9. Change preference for monthly report (user-only)

Architecture and Features

The backend folder consists of business logic – Flask routes, cache description, asynchronous jobs, APIs and such. It also contains the report format and the project database along with the YAML file for API specifications.

The frontend folder contains different views and JS/.VUE files for running the application.

The application has a different register and login page for user and admin. After logging in they're redirected to their respective dashboard. User dashboard shows a list of shows under a venue where they can book the tickets. It also includes a detailed page for each venue. User can see their past booking in bookings history page and rate them also. There is functionality for daily reminders and monthly reports sent in PDF/HTML format via email.

Admin dashboard has similar UI but with admin specific features like creating/deleting/editing shows and venues. Admin can also see stats about each venue. Admin can export these stats in a .csv format.

Video

https://drive.google.com/file/d/1lMI9gbuD_EQz3IVFf22JGcmzKnT5u0Lr/view?usp=sharing