Author

Dona Mary Jose
21f1006996
21f1006996@student.onlinedegree.iitm.ac.in

I am a Computer Science Engineering student, currently in my fourth year, with an interest in Data Science and Web Development.

Description

TicketFlix is a Ticket Show Management Application project with admin and user dashboards where admins have the functionality to create venues, movies and shows while users can view, book the showings and download the tickets, which they have booked.

Technologies used

- Flask – used to facilitate web development in python
- Vue.js – framework used to develop the frontend and UI
- Flask-SQLAlchemy – to support the use of SQLAlchemy in the application
- Flask-security - to ensure authentication of the users
  Flask-Restful – used to create REST API
- sqlite3 – used for the database
- Celery - for tasks and job management
- Redis - for caching

DB Schema Design

- User Table

-	user_id	Integer	Primary Key, Auto Increment
-	email	String	Not Null, Unique
-	name	String	Not Null
-	password	String	Not Null

- Role Table

-	role_id	Integer	Primary Key, Auto Increment
-	name	String	Not Null
-	description	String	

- Venue Table

-	venue_id	Integer	Primary Key, Auto Increment
-	name	String	Not Null
-	location	String	Not Null
-	capacity	Integer	Not Null

- Movie Table

-	movie_id	Integer	Primary Key, Auto Increment
-	name	String	Not Null
-	poster	String	Not Null
-	description	String	Not Null
-	duration	Integer	Not Null
-	rating	Float	

-	tags	String	

- Show Table

-	show_id	Integer	Primary Key, Auto Increment
-	venue_id	Integer	Not Null, Foreign Key
-	movie_id	Integer	Not Null, Foreign Key
-	datetime	String	Not Null
-	seats_available	Integer	Not Null
-	price	Float	Not Null

- Booking Table

-	booking_id	Integer	Primary Key, Auto Increment
-	user_id	Integer	Not Null, Foreign Key
-	show_id	Integer	Not Null, Foreign Key
-	datetime	String	Not Null
-	quantity	Integer	Not Null
-	total_price	Float	Not Null
-	status	String	

API Design

API has be defined for User, Venue, Movie, Show and Booking. The GET, PUT, POST and DELETE functions have been defined for each resource. GET returns the details of the item by taking in the required arguments, PUT allows the user to update the item by taking in the required arguments, DELETE removes the item from the database, and POST is used to create a new item by taking in the required arguments.

In case of any invalid entries, the appropriate responses/error messages will be sent along with its code and description. The required error codes have been specified and added on to the API along with their appropriate error messages.

Architecture and Features

The project is structured such that the frontend and the backend have a clear separation. The frontend files are stored entirely on the client side. Vue.js framework is used to develop the frontend. It is connected with the backend through API calls. The main view files are stored in the views folder while the Vue components are stored in the components folder. Static files are stored in the assets folder.

The backend includes the database, the configuration, as well as the API definitions, which are accessed by the main application. The asynchronous tasks will be running on the backend using Celery. Redis database is used for caching.

The admin can login to the application using the authorized credentials and will be redirected to the admin dashboard, where they can create, update and delete new venues, movies and shows. The number of tickets for each show will be equal to the capacity of its venue. Deletion of a venue or a movie will lead to deletion of all of its shows.

The users can create an account using unique email IDs and login, after which they will be led to the main dashboard. The users can view the available showings and book tickets, which they will be able to edit, delete or confirm. Once they confirm a booking, a confirmation mail will be sent along with the E-ticket. Users can also utilize the search bar which they can use to find shows according to their preference. The users can log out of their account once they are done.

Video

https://drive.google.com/file/d/1TUuGO5JUjSdjhtDi1mAFymU0ooBzPFVL/view?usp=sharing