

Student Name – Lalit Kumar
Roll Number - 21f3001013
Email : 21f3001013@ds.study.iitm.ac.in

I am a serving Air Force officer pursuing BSc Data Science degree from IIT Madras
PROJECT REPORT TICKETSHOW (WEB APPLICATION FOR BOOKING TICKETS)

Description: This project is to develop a web application 'TICKETSHOW' using Flask for backend and vueJS for frontend to book movie tickets with CRUD operations for admin on venues and shows, and give functionality to the user to register, login, book tickets, see my_bookings and delete his bookings.

Techonolgies Used

- (a) SQL Lite - Used as the main database to implement all the models for this project.
- (b) Flask used for backend API to serve JSON objects to the front end.
- (c) Flask SQL Alchemy - Used Python to make changes directly into the SQL Lite database.
- (d) VueJS - Used for the front end, displaying, and making calls to the back end.
- (e) Redis - Used for message broking and caching.
- (f) Celery - Used to schedule async tasks and periodic jobs.

Database Schema Design

There are four models defined in models.py for the project. Models are as following :-

User The model is designed to store information about users including admin) and the stored information is user.id, user.username, user.password, role is_admin(default =0 , means user is not admin), role (default is 0, meaning user is a user, 1 for admin), approved (default is 0, 1 for approved users).

Venue The Venue model is designed to store information about venues in terms of id, name, place and capacity. Id is the primary key and venue has relationship many to many with Show as shows can have many venues and a venue can have many shows.

Show The Show model is designed to store information about showss in terms of id, name, rating, tags, image_path, ticket_price and venue_id(foreign key is Venue.id) . Id is the primary key and one to one relationship defined for bookings.

Bookings The booking model is designed to store information about bookings containing id, quantity, user_id, show_id and venue_id. Id is the primary key. Show_id and venue_id foreign keys are show.id and venue.id respectively.

API Design All the APIs have been defined in api.py with APIs for registration, login, CRUD operations by Admin on all models and CRUD operations by the User on bookings. API specifications have been included in api.yaml in root folder.

Architecture and Features.

Following is the structure of the backend and the frontend of the app:-

Backend The backend app is organised with application folder (with api.py containing all APIs, config.py with app configurations, database.py with database configurations, login.py for login functions, controllers.py with all CRUD functions of admin and user, models.py with data models, workers.py for celery configuration and emailcontrollers.py for scheduling celery tasks) , db_directory folder(with the database), static folder (with folder uploads with images), templates (with various HTML templates for login and CRUD ops methods), api.yaml with API specifications, requirements.txt containing all software dependencies and the main.py.

Frontend The frontend app is organised with the src folder containing Vue.js source code files (assets, components, views, routers, Vue.vue and main.js). The asset folder holds static assets like images. The component folder contains all the reusable UI components and the views folder contains larger components representing different views of the app. The Vue.vue is the root Vue component and main.js initialises the Vue app.

Core Functionalities

Following core functionalities have been implemented in the app:-

Basic login/signup for the user is implemented and the JSON Token-based authentication is used to authenticate users.

Home page contains option either to login or register. Login for both admin and user is implemented and on successful login a token is generated, which is used for authentication by the frontend. Homepage also has search functionality for venues and shows.

On the Admin dash board, Admin can view all venues and shows. On viewing them, Admin gets an option to either delete or edit them. Admin also has features of creating a new venue, new show. It also has a feature for approval of newly registered users by the Admin.

On the User dash board, user has options to make new bookings, see all his bookings and edit/delete a particular booking. User also has the option to export his bookings as csv. They are sent monthly progress reports via mail and a daily reminder to visit the app.

Video Link : https://drive.google.com/file/d/18TOEUDzLd4lhC1E3qQVgw-I_Eet5_I7l/view?usp=drive_link