

Student Name – Lalit Kamar

Roll Number - 21f3001013

Email : 21f3001013@ds.study.iitm.ac.in

Brief Description : I am a serving Air Force officer pursuing BSc Data Science degree from IIT Madras

PROJECT REPORT **TICKETSHOW (WEB APPLICATION FOR BOOKING TICKETS)**

1. This project is to develop a web application using Flask 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. Software used are listed in requirements.txt to help in installation from requirements.txt in virtual environment. I

2. **Models** There are four models defined in models.py for the project. UserMixin from flask_login alongwith the database has been used for defining these models. The sqlite3 database is made in database directory with the same models . Models are as following :-

(a) **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).

(b) **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.

(c) **Show.** The Show model is designed to store information about shows 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.

(d) **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.

3. **Forms.** Following forms have been designed in forms.py to fetch data from templates for respective functionalities.

For User login, For Admin login, For creating Venue Object, For creating Show Object, For creating Booking Object, For creating Registration Object .

4. **Configuration.** Configuration has been defined in config.py. base directory has been given there and LocalDevelopmentConfig function has been defined in config.py.

5. **Login.py** The Login.py contains routes for login . Blueprint has been used to define login route. Routes for 'login', 'user_login', 'admin_login', 'register' 'register_approval' and 'logout' have been created here. Two functions user_required and admin_required have been defined and have been used as decorators. Appropriate templates using Jinja2 have been defined in template folder.

6. Controller.py Controller.py contains main logical functions and here route 'routes' defined using blueprint. Anyone can search for shows, however for booking tickets user login is mandatory. In controller.py, functions to create, edit and delete venues and shows have been defined with decorator 'admin_required'. 'create_booking', 'my_booking' and 'delete_booking' have been defined for creating a booking, display of bookings of a user and deleting bookings by user with decorator 'user_required'.

7. Main.py main.py contains the function to create_app. Database imported and initialized inside the app. After creation of app, Blueprint routes are registered with the app, and then all routes are imported from controllers.py.

8. Functionality of the app. Following Base requirements are functional in the app:

- a. Admin login and User login, Venue Management, Show Management
- b. Booking show tickets, Search for shows/venues, Form for username and password for user, Separate form for admin login, proper login framework
- c. Suitable model for user,
 - Core - Venue Management (Only for Admin)
 - i. Create a new venue
 - ii. Edit a venue
 - iii. Change title/caption
 - iv. Remove a venue
 - v. With a confirmation from the admin
 - d. Core - Show Management (Only for Admin)
 - i. Create a new show
 - ii. Edit a show
 - iii. Change title/caption or image
 - iv. Remove a show
 - v. With a confirmation from the admin
 - vi. Allocate venues while creating shows
 - vii. Different Pricing for each venue
- e. Core - Search for Shows/Venues
- f. Ability to book tickets for a show at a given venue

Video Link : https://drive.google.com/file/d/1OM5sRUcdFWXIGaz-mmsSyUk-5JWtIBPS/view?usp=share_link