## Author

**Name:** Rohit Kumar
**Roll No:** 21F1001368
**Email ID:** 21f1001368@ds.study.iitm.ac.in

My name is Rohit Kumar. I am a student of Diploma level at Indian Institute of Technology, Madras. Apart from this, I have completed my graduation in Chemical Engineering from Dr S.S.B. University Institute of Chemical Engineering and Technology, Panjab University, Chandigarh. I have keen interest in Machine Learning, Deep Learning and Artificial Intelligence.

## Description

ShowSpot, is a ticket booking application. In this application, users can book tickets for shows happening in their locality. Also, show administrators can post new shows, venues for the user. Multiple users and Multiple administrators can work on ShowSpot applications simultaneously.

## Technologies used

**flask:** Flask is a web framework which is used as backend server for this application
**flask_login :** flask_login module is used to control user access to application through proper login framework
**flask_SQLAlchemy**: flask_SQLAlchemy is a flask extension that uses SQLAlchemy to connect to the database server
**flask_restful**: flask_restful is extension of flask that helps to add support for APIs
**flask_bcrypt**: Flask-Bcrypt is a flask extension that provides hashing functionality to our applications
**Matplotlib:** is a comprehensive plotting library. We are using this library to plot bar graphs for insights.
**HTML/CSS/JINJA/BOOTSTRAP**: Bricks for this application. We are using these to create our frontend for our application.

## DB Schema Design

Database used for this application has 5 tables - **admin, booking, show, user, venue**. Each table is used to store information as their name suggests.

- admin

| | | |
|---|---|---|
| user_id | Text | NOT NULL, UNIQUE, PRIMARY KEY |
| password | Text | NOT NULL |
| email_id | Text | |
| phone_no | Integer | |
| name | Text | |

- booking

| | | |
|---|---|---|
| booking_id | Integer | NOT NULL, UNIQUE, PRIMARY KEY, AUTO-INCREMENT |
| user_id | Text | NOT NULL, ForeignKey references user(user_id) |
| show_id | Integer | NOT NULL ForeignKey references shows(show_id) |
| tickets_booked | Integer | NOT NULL |
| price_paid | NUMERIC | NOT NULL |

- show

| | | | |
|---|---|---|---|
| show_id | Integer, Primary Key | ticket_price | Integer, Not Null |
| show_name | Text, Not Null | admin_id | Text, Not Null |
| description | Text,Not Null | timing | Text, Not Null |
| tags | Text | rating | Integer, Not Null |
| venue_id | Integer, Not Null, Foreign Key | rseats | Integer |

- venue

| | | | |
|---|---|---|---|
| venue_id | Integer, Primary Key | admin_id | Text, Foreign Key |
| venue_name | Text | venue_img_url | Text |
| location | Text | | |
| address | Text, | | |
| capacity | Integer, Not Null | | |

- user

| | | | |
|---|---|---|---|
| user_id | Integer, Primary Key | address | Text, |
| password | Text | | |
| name | Text | | |
| email_id | Text, | | |
| phone_no | Integer | | |

## API Design

In API design, This application has 5 request points which includes

- GET request to fetch user details with specific userID
- PUT request to update user details with specific userID
- DELETE request to update user with specific userID
- GET request to fetch all the booking of user with specific userID
- GET request to fetch all the venues hosted by particular admin with adminID
- GET request to fetch all the shows hosted by particular admin with specific adminID

All these API are implemented with flask_restful framework by providing various access points.

## Architecture and Features

This project has a python file named “main.py” that launches the applications. Flask Blueprint is used to host two applications, where one is dedicated to admin functionalities. The admin app has a separate folder structure within the “admin” folder, where all the controllers and login functionality are located. Admin functionalities include creating new venues, new shows across multiple venues, and tracking user transactions.

The main Flask app is responsible for user functionalities and all relevant python files such as user login, booking shows, cancelling tickets and viewing previous transactions are placed in the “application” folder located in the root directory . Overall the project is structured in a way that separates admin and user functionalities, allowing for easier maintenance and modularity.

## Video

**Video Link:** https://drive.google.com/drive/u/1/folders/15igPgbL_-eWShJqWD0YIqTYF6GW1Ilax