### Author

RITUPARNA DAS

21f1003954

21f1003954@ds.study.iitm.ac.in

I have completed my graduation in Mathematics. This is the first App development project for me and I have tried do this as per my understanding and knowledge.

### Description

A basic ticket booking app with venue and show management by admin where admin can create , update and remove venues and shows. User can book tickets for the shows in particular venues. Also user can search for shows and venues and give ratings.

### Technologies used

The project uses Flask, a Python web framework, as the main technology for implementing the web application.

Flask==2.2.3, Flask-SqLalchemy==3.0.3, jinja2==3.1.2,SQLAlchemy==2.0.6,matplotlib==3.7.1, numpy==1.24.2

* Flask, SQLAlchemy and Flask-SQLAlchemy are used to build the models and database.
* Jinja2 is used for templating. Matplotlib and numpy are used for plotting graphs.

### DB Schema Design

* The database for my ticket booking app, consists six tables : User, Admin, Venue, Show, Association, Booking and Rating
* **User table:** user\_id(primary\_key), user\_name, user\_email, and user\_password.
* **Admin table**: admin\_id(primary\_key), admin\_username, and admin\_password.
* **Venue table**: venue\_id(primary\_key), venue\_name, venue\_place, venue\_city, venue\_capacity. The shows column is a relationship with the Show table through an Association table.
* **Show table** : show\_id(primary\_key), show\_name, show\_genre, show\_rating, show\_price, start\_time, end\_time, venue\_id(foreign\_key), and available\_seats.
* **Association table** : venue\_id(foreign\_key) and show\_id(foreign\_key).
* **Rating table**: rating\_id(primary\_id), venue\_id(foreign\_key), show\_id(foreign\_key), user\_id(foreign\_key).
* **Booking table** : booking\_id(primary\_key), venue\_id(foreign\_key), user\_id(foreign\_key), show\_id(foreign\_key), date, and number\_of\_seats.

1. Only admin can perform CRUD on venues and shows
2. Each show and venue has unique show and venue ids respectively. Shows are listed under the venues, user can book multiple tickets for one or multiple
3. shows. User can search for shows or venues, user can give ratings for shows and venues.
4. I have tried to follow the wireframe and project guidelines(problem definition)

### API Design

Not Applicable

### Architecture and Features

The project is organized using the Model-View-Controller (MVC) architecture pattern. The models are defined using SQLAlchemy ORM and are stored in the app.py file. The controllers, which handle the business logic, are implemented using Flask framework and are located in the app.py file. The controllers act as an interface between the user and the models, which means that they receive user input, process it, retrieve data from the models and render the appropriate templates. The templates, which define the user interface, are located in the templates folder and are written using Jinja2 template engine. The static files such as CSS and images are stored in the static folder. Overall, the project is well-organized and follows standard practices for Flask web application development.

* The project includes several features implemented to provide a smooth user experience. The default features includes admin login, logout, user registration, user login, and user logout. Once logged in the admin can view the list of venues and shows and can perform CRUD on shows and venues. Once logged in, users can view the list of available shows and venues, book tickets for a particular show, rate a venue or show. Also search for a venue or show. The admin dashboard is protected by a login page that can only be accessed by admin. The additional feature is in admin dashboard admin can see the analysis or graphs of available seats for a show or rating chart for venues and shows.

### Video

https://drive.google.com/file/d/1gvX5zxXxN9aMjLpyAH6khr\_aZojf8Le9/view?usp=sharing