Ticket-Show Report

Author:

- **Name:** Muskan Sindhu
- **Roll No:** 21f1003710
- **Email:** 21f1003710@ds.study.iitm.ac.in
- **About me:** I graduated with a Bachelor of Vocation in Web Designing from Kalindi College, Delhi University in 2022. Currently, I am pursuing BS Degree in Data Science and Applications from IIT-Madras and I am at the diploma level.

Description:

Ticket Show is a Flask-based web application allowing users to book multiple shows at different venues. The app has a feature where users can search for different shows and have a look at the bookings made by them on a separate page. This app allows the admin to perform CRUD on venues and shows and to view a summary chart for the number of bookings made for different shows.

Technologies used:

- **Flask:** a web framework for building the web application
- **Flask-SQLAlchemy:** used for streamlined management of databases
- **SQLite:** database for storing admin, user, venues, shows and bookings
- **Jinja:** used for separation of logic and dynamic HTML rendering in the Flask app
- **HTML:** for developing the web page
- **Bootstrap**: To make the frontend appealing and easily navigate
- **Matplotlib:** To create a chart to view the most booked show

DB Schema Design:

The database has five tables and the schema is as follows:

Admin Table	User Table
• **ID** (Integer): Primary Key, Auto Increment • **Username** (String): Unique, Not Null • **Password** (String): Unique, Not Null	• **ID** (Integer): Primary Key, Auto Increment • **Username** (String): Unique, Not Null • **Password** (String): Unique, Not Null • **Seats_booked** (Integer): Unique, Not Null

Venue Table	Show Table
• **ID** (Integer): Primary Key, Auto Increment • **Venue_name** (String): Not Null • **Place** (String): Not Null • **Location** (String): Not Null • **Capacity**(Integer): Not Null	• **ID** (Integer): Primary Key, Auto Increment • **Show_name** (String): Not Null • **Venue ID**(Integer): Foreign key (Venue), Not Null • **Tags** (String): Not Null • **Time** (String): Not Null • **Rating**(Integer): Not Null • **Price**(Integer): Not Null • **Poster_path**(String): Not Null • **Seats_booked**(Integer): Not Null

Booking Table
• **ID** (Integer): Primary Key, Auto Increment • **Show ID**(Integer): Foreign key (Show), Not Null • **User ID**(Integer): Foreign key (User), Not Null • **Seats_booked**(Integer): Not Null • **Total_price**(Integer): Not Null

API Design:

The CRUD API was created for performing operations such as adding the venue, and editing and deleting the venue. A similar CRUD API was created for adding the show, editing and deleting the show.

Architecture and Features:

The application follows the standard MVC architecture. The View of the application is created using HTML and Bootstrap. The Controller is created using Python and Flask. The Model is created using SQLite.
The features of the application are as follows:

- Separate Sign-Up and Login pages for admin and user
- Ability to view venues and shows
- Ability to search for shows
- Ability to make bookings
- Ability to view the number of show booking summary chart
- Create, View, Edit and Delete the venue
- Create, View, Edit and Delete show

Video:

For the video, click here!