

Author

Richik Majumder

21F1000460

21F1000460@ds.study.iitm.ac.in

I am from West Bengal, Kolkata. I have completed B.Tech in Mechanical Engineering in 2019 and currently pursuing IITM BS in Data Science and Applications as a standalone course

Description

A ticket booking app should provide a convenient, user-friendly, and efficient way for users to purchase tickets to their favourite shows. It is used to simplify the process of booking show tickets.

Technologies used

Flask is used for rendering templates, url redirection, sessions.

Flask_session is used for user session management and authentication.

Flask_wtf and wtform is used for handling and validating login and signup forms.

Flask_bcrypt is used for hashing passwords using bcrypt.

Flask_sqlalchemy is used for creating database tables.

DB Schema Design

The database consists of six tables: **Venue**, **Show**, **Link**, **User**, **Booking**, and **Admin**. Each table has a unique primary key, auto-incrementing integer ID column, except for the User table, which has a unique string username column as its primary key.

The **Venue** table has columns for the name, place, location, capacity, and a relationship with the Link and Booking tables. The **Link** table has foreign key relationships with the Venue and Show tables, as well as columns for the show time, ticket price, and tickets left.

The **Show** table has columns for the name, rating, tags, and a relationship with the Link and Booking tables. The **Booking** table has foreign key relationships with the Show, User, and Venue tables, as well as columns for the number of tickets booked, the ticket price, and the user's rating of the show.

Finally, the **User** table has columns for the username, password, and whether the user is an admin. The **Admin** table has columns for the username, password, and a boolean flag indicating whether the user is an admin.

The tables have constraints such as primary keys and foreign keys, which ensure data integrity and consistency. The Link table's foreign keys ensure that every link is associated with a venue

and a show, while the Booking table's foreign keys ensure that every booking is associated with a user, a venue, and a show.

Architecture and Features

All the controllers are located inside "main.py" and the code for the database is located inside "database.py" and all the templates are located inside the "templates" folder.

Features like signup for users/admin , login for users/admin, logout for users/admin, create/update/delete venues for admin based login, create/update/delete shows for admin based login, linking/delinking/updating shows to venues for admin, booking show tickets, booking multiple show tickets up to the number of tickets available, stop accepting bookings in case of a housefull, searching shows based on tags/show name/rating (in case of rating greater than or equal to logic is considered while showing search results), booking tickets based on location based search on venues, allowing users to check their booking history inside the user dashboard and rate shows (once rated, the ratings for that particular show will automatically be updated) are the default features that are implemented.

Apart from the default features dynamic pricing is implemented for show ticket booking, which results in a 10% increase of the ticket price for every 10% of total seats booked capped at Rs 1000 per ticket per show for a particular show (i.e 10% increase in price or Rs 1000 whichever is lower)

Video

<https://drive.google.com/file/d/1Y1kMRzXs2ajsZcP7SoxzMy2O-gXWCKZc/view?usp=sharing>