Author

Kanishkar K
21f3001666
21f3001666@ds.study.iitm.ac.in

Myself Kanishkar K im currently pursuing 2nd year BTech.Computer Science (Cybersecurity & Blockchain Tech) at SASTRA University,Thanjavur. I am passionate towards Animation and Game dev have some experience in those fields.

Description

The project is about creation of an web application which could be used for booking of tickets to movies which are posted on venues by admins who has access to create,delete or update those venues and shows

Technologies used

Flask, Flask_sqlalchemy, Flask_login, Werkzeug security, Datetime
Flask and extension is used to create a web application. Sqlalchemy makes database interaction easier. Flask_login makes the authentication process simpler and secure. For password security Werkzeug is used. Datetime is used to retrieve the current date for a specific application.

DB Schema Design

The model User has the following: id which is a primary key to the table.it is used as an unique identifier for each user object. Username is stored in the name column. Isadmin is a boolean value to know whether the user is an admin or not,it is set to True if the person is admin. Password is used for authentication and logging in of the user.

The model Venue has the following: Id an unique identifier for each venue. Admin_id a foreignKey to the admin table to indicate which admin created the venue. Name is the name of the venue,City is the city where the venue is located.

The model Show has: Id unique identifier for each show. Venue_id to represent in which venue the show is performed. Name, desc ,time and date are details of the show. Price is the price of a single ticket and capacity is the number of seats available. It has a rating which denotes the show's rating.

The model booking has: Id an unique identifier to represent the booking id. User_id to represent which user has booked it, show_id to represent for which show. Tick is the number of tickets the user has booked.

The model Tag has: Used to represent the tags a movie has and to fetch based on the tag provided. Show_id to like it to the show which has the tag. Tag is the tag which the show has is been stored in the table for easy fetching

API Design

API's used for CRUD of Venue & Shows, finding the user logged in, booking of shows.

For Venues and Shows this takes in input through an interface which is later stored in Database respectively.

The user which has been logged in is found through the current_user which is available in the login manager.

Booking shows again takes the input through the interface and checks for availability and on re entering the number of tickets this will book the tickets.

Architecture and Features

The Project is organized through multiple html forms each for specific purposes like for user signup,user login, admin signup and login, home pages for admin and users,,bookings venue and show posting etc.

The architecture starts with a page containing 2 buttons to determine user or admin each redirecting to respecting pages.

The pages has a login and signup forms for both the admin and user.Which on successful login redirects to their respective homepages.

Admin home page has a navbar button to create a venue which redirects to a html form for creation of the venue. Then each venue can be updated and deleted with confirmation.

Each venue can have any number of shows which can be added through the add button in the home page.

Adding the show send to a form where the required details are gathered and the show is created.

Then the user side.

The user home page has a search bar which has a city date tag and a text field.

These fields operate on searching the specific needed shows which are preferred by the user.

If the city field is empty then it throws an error message because we cannot search a show without specifying the city, however even if the city is empty and the text area has a venue name it returns the venue name with a button to see its details.

The date if left empty takes the default value of the current date.

Tag if empty doesn’t do anything but when it is not empty filters the showa accordingly.

The movie name can also be searched but with a city name.

Features:

CRUD of Show , Venue.

Searching for venues and shows based on preferences.

Booking tickets in needed numbers and posting housefull.

Displays Shows which are after the current time for today's date if not provided.

Video

The video explains the working and features in the project:

https://drive.google.com/file/d/1ptL3YZpo0R8C5BI-tHtNqW0_IMuAlzeV/view?usp=share_link