

Author

Manas Salunke

21f1002150

21f1002150@ds.study.iitm.ac.in

I am passionate about my work and love cycling and running marathons.

Description

A ticket show app in which registered users can book tickets, new users can register and admin can add, edit, delete shows and venues. Users will automatically get emails if they don't book tickets for 24 hrs or more. Anyone can search for shows and venues based on venue location, show rating, and show tags etc. Api functionality is also provided.

Technologies used

Python language, modules flask (Flask, render_template, url_for, redirect, request), flask_sqlalchemy (SQLAlchemy), etc.

DB Schema Design

Class Venue

Columns: id (integer, primary key)
name (string)
place (string)
location (string)
capacity (integer)
shows (relationship of venue with show)

Class Show

Columns: id (integer, primary key)
name (string)
rating (string)
timing (string)
ticket_price (integer)
tags(string)
venue_id (integer)
booking (relationship of show with venue)

Class User

Columns: id (integer, primary key)
email (string, unique because we don't want multiple users with the same email)
password (string)

Class Admin

Columns: id (integer, primary key)
email (string, unique because we don't want multiple admins with the same email)
password (string)

Class Association

Columns: show_id (integer, primary key, foreign key(show.id))

user_id (integer, primary key, foreign key(user.id))

tickets(integer)

timestamp(Datetime)

(user and show have a many to many relationship, hence this association table)

API Design

Apis for crud on venues and crud on shows

<http://127.0.0.1:5000/user/new>

<http://127.0.0.1:5000/user/login>

<http://127.0.0.1:5000/user/venue>

<http://127.0.0.1:5000/user/book/{id}>

<http://127.0.0.1:5000/user/bookings>

<http://127.0.0.1:5000/admin/login>

<http://127.0.0.1:5000/admin/venue>

http://127.0.0.1:5000/admin/add_venue

http://127.0.0.1:5000/admin/edit_venue/{id}

<http://127.0.0.1:5000/admin/venue/delete/{id}>

http://127.0.0.1:5000/admin/add_show/{vid}

http://127.0.0.1:5000/admin/edit_show/{id}

<http://127.0.0.1:5000/admin/show/delete/{id}>

<http://127.0.0.1:5000/search>

<http://127.0.0.1:5000/venue/export/{id}>

<http://127.0.0.1:5000/status/{id}>

Architecture and Features

The file app.py has a first section of sql models, then the configuration. Next section is controllers and it has user, admin and search controllers. Search section is sub divided into search by venue and search by show.

The templates are the in the templates folder.

The search features has search by venue location and search by show tags, ratings. We can search multiple tags at the same time.(tags are to be entered separated by commas and no space)

https://drive.google.com/file/d/1_frceb4ziDnF3l2Y8o-l0-Q9-JjiUhRA/view?usp=sharing