Author

Amisha Mishra
22F1000938
22f1000938@ds.study.iitm.ac.in

My name is Amisha Mishra, currently in the 1st term of Diploma with MAD1, DBMS and MAD1 Project.

Description

We were supposed to make a multi-app which lets users book their choice of show tickets online through our app. We were supposed to make two separate login systems, one for user and one for admin and add corresponding sections to them.

Technologies used

- Flask -for application code
- flask_sqlalchemy –to work with databases
- flask_restful –for creating API
- flask_cors –To enable Cross Origin Resource Sharing (CORS) in flask app
- matplotlib.pyplot –for creating bar graph
- SQLite – for data storage
- Html- for making the html pages
- Css, bootstrap- for styling
- Jinja2- for creating templates

DB Schema Design

Class Venue

Columns: Venue_id (Integer, Primary_key)
Venue_name (String, not null)
Place (String, not null)
Capacity (Integer, not null)
Rating (Integer)
rel_show (many to many relationship)

Class Show

Columns: Show_id (Integer, Primary_key)
Show_name (String, not null)
Rating (Integer)
Tags (String)
Date (String)
Time (String)
TicketPrice (Integer, nullable=False)
Language (String, nullable=False)
Type (String, nullable= False)
Venue_id (Integer, ForeignKey)
Show_Capacity (Integer)

Class Link

Columns: Venue_id (Integer, ForeignKey)
Show_id (Integer, ForeignKey)

Class User_data

Columns: Email_Address (String, Not null, unique)
Username (String, Primary_key)
Password (String, not null)

Class Admin_data

Columns: Username (String, Primary_key)
Email_Address (String, not null, unique)
Password (String, not null)

Class Booking

Columns: ID (Integer, primary_key)
Show_id (Integer)
Show_name (String)
Venue_id (Integer)
Venue_name (String)
Username (String)
email (String)
phone (Integer)
Number_of_Seats (Integer)

I have created 6 tables for my project, "Show" and "Venue" for keeping the Show and the Venue data respectively. "Link" table created which is the association between the "Show" and the "Venue" tables. Then, I created "User_data" and "Admin_data" to keep the User and Admin information. At last, "Booking" for keeping the data or shows booked by users.

API Design

I have created the CRUD operations on "Show" and "Venue" table, i.e. admin can CREATE, READ, UPDATE and DELETE shows and venues using the corresponding API endpoints. I have created 2 classes, one each for show and venue and in each class, I have defined 4 functions, post(for create), get(for read), put(for update), delete(for delete), and have assigned corresponding endpoints for them.

Architecture and Features

All my application code, including models, controllers and API are present in "app.py" file and all my html pages are present in "templates" folder, which is present in the same directory as the app.py. My images and bar graphs are present in "static" folder, also in the same directory. I have created a virtual environment wherein all the needed modules are installed (one just needs to activate it every time they want to run their app) and I have stored all the installed modules in requirements.txt.

I have implemented various features including: ,User can book many tickets for many movies, Admin can create venues and shows, Every venue can run a number of shows, System will automatically show the latest added shows

Base requirements:

- Admin login and User login
- Venue and Show Management(creating , updating, deleting shows and venue)
- Booking show tickets based on availability
- Search for shows

Ability to book multiple tickets for a show at a given venue. Bar graph showing Shows and number of seats available. Bar graph showing Venues and their Capacity.

Video: https://drive.google.com/file/d/16PsKYLAUiQN4oerF_jVEjjspZsKwzd_X/view?usp=share_link