AUTHOR

Name: Nandini Kushwah

Roll No: 22f1001148

Student's Email: 22f1001148@ds.study.iitm.ac.in

Greetings everyone I am Nandini Kushwah, from Gwalior, Madhya Pradesh. Write now I am 22 years old and perusing BS in Data Science and Programming from IIT Madras. I did my intermediate with Mathematics from Miss Hill H.S School, Gwalior. Before this degree I was doing Hotel Management from IHM Pusa. But unfortunately, I dropped it in my 2nd year. But now I am glad to be the part of one of the prestigious institutes of India. And loved to learn new things with exciting challenges.

DESCRIPTION

In this ticket show application, we supposed to create data driven flask web app with SQLite for database management that book tickets for a particular user, where user can search for shows and venue, and create a CRUD interface for admin, who can create, update and delete venues and show.

TECHNOLOGIES USED

Flask: Flask is a micro-web framework with little to no dependencies to external libraries. This makes the framework light with little dependency to update and watch for security bugs, which is best suited for beginners. When we install flask some default extensions of flask also get installed in the machine, for example;

- **Click:** It's the "Command Line Interface Creation Kit". It is a python package for creating beautiful command line interfaces in a composable way with as little code as necessary.
- **ItsDangerous:** It helps in security, when some data is sent to untrusted environments, then get it back later. To do this safely, the data must be signed to detect changes. We can assign a key which helps to get the data without any changes.
- **Jinja2**: Jinja2 is extensible templating engine which have a very similar coding syntax like python, which is very easy to work with. And its logical and arithmetical operations are very useful to manage the data for a particular page.
- **MarkupSafe:** MarkupSafe escapes characters so text is safe to use in HTML and XML. Characters that have special meanings are replaced so that they display as the actual characters. This mitigates injection attacks, meaning untrusted user input can safely be displayed on a page.
- **Werkzeug:** It is a collection of libraries which is used to create web server gateway interface. It helps in requests processing and routing.

Flask-SQLAlchemy: Flask-SQLAlchemy is an extension for Flask that adds support for SQLAlchemy to flask application. It simplifies using SQLAlchemy with Flask by setting up common objects and patterns for using those objects, such as a session tied to each web request, models, and engines. In this application we are going to use,

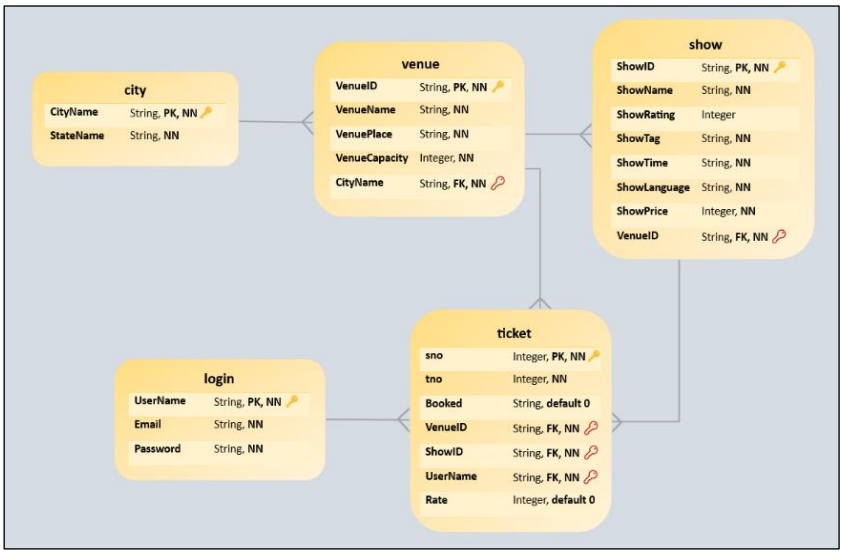
- **SQLite:** It is a self-contained full featured embedded database in file format, with no client-server model which doesn't need separate process to run the server, by which we can run it in the application itself. And as it is a file, we can delete, copy or retrieve this anywhere.

Bootstrap: It is used for making the body of the application, because bootstrap provide inbuilt JavaScript plugins and default CSS, which lessen the efforts to make a good-looking responsive website.

DB SCHEMA DESIGN

For this web application I have made 6 entities; **login, city, venue, show, ticket** and **update.**

In this schema I have made one to many relationship, where city can have multiple venues, venues can have multiple shows, shows can have multiple tickets, venue can have multiple tickets and a user in login table can book multiple tickets. update is not written in this schema because it is for "UPDATE" functionality, it stores all the attributes of the above entities temporarily when update process called by admin. All the attributes can be null for this entity, except sno.



ARCHITECTURE AND FEATURES

In my project folder named as "22f1001148", there are 2 folders "templates" and "static" and 4 files "app.py", "database.sqlite3", "ReadMe.txt" and "requirements.txt".

A. **"templates"** folder contains 12 HTML files;

1. **"base.html":** It have the code for navbar. And it will be extended for every other page.
2. **"index.html":** This is the first page, where it shows all the venues in a city and three shows of that city (I set it to three for good appearance). In this page we can select a city from nav bar also. And it takes default user as "default" and CityName as "Gwalior".
3. **"admin.html":** When admin login, this page will be rendered where shows all the cities are available. And admin can add and delete any city and can go to update page and venue page for a city. When a city is deleted all the venue, shows and tickets will also be deleted.
4. **"venue.html":** This page is similar to admin page, with functionalities for venues and from here you can go to the show page for a particular venue.
5. **"show.html":** This page is also same as admin page, with functionalities for show. Here it has a "seat" button which lands you on seating arrangement of the show.
6. **"updatecity.html":** This is for updating city.
7. **"updatevenue.html":** This is for updating venue.
8. **"updateshow.html":** This is for updating show.
9. **"search.html":** When we search venue or show with venue name or place, show name, tag or time, this page will be rendered with the searches.
10. **"venue_show.html":** We had buttons in "index.html" for venues, when any button is clicked, this page will be rendered with shows of a particular venue.
11. **"seating.html":** This page will be open when a user tries to book tickets. It has the seating arrangement for a particular show.
12. **"booked.html":** When a user booked their tickets, he/she will be rendered to this page which show their ticket and make them able to rate this movie.

B. **"static"** folder contains a "style.css" file and an image folder that contains all the images used in the web app.
C. **"database.sqlite3"** is an SQLite 3 file for the data where the data will be stored and retrieved.
D. **"ReadMe.txt"** is made for the users who want run the code in their machine with detailed explanation to 'How to run this web app?'.
E. **"requirements.txt"** contains all packages details that need to be installed of import.

VIDEO

For More Details Click Here → Link