Author

Sarthak Komal Akkarbote
21f2000114
21f2000114@ds.study.iitm.ac.in
TY B.Tech student at VIT Pune and diploma student at IITM. Interested in data science, machine learning and software development.

Description

The project involves designing and implementing a web application for booking tickets for events at different venues. The application should allow users to view venues and events, book tickets, and manage their bookings. It requires designing and implementing database models, user authentication and authorization, forms, and views using Flask and SQLAlchemy.

Technologies used

1. Flask
2. SQLalchemy
3. Flask-WTF (extension for Flask)
4. Bootstrap (front-end framework)
5. Jinja2 (template engine for Flask)
6. SQLite (database management system)
7. Bcrypt for hashing password

Flask: Flask is a micro web framework written in Python. It is used in this project to build the web application, handle routing, and to define views.
Flask-WTF: Flask-WTF is a Flask extension that provides an integration with the WTForms package. It is used in this project to handle form submissions and form validation.
SQLAlchemy: It is used in this project to interact with the database.
Bootstrap : Bootstrap is a free and open-source CSS framework. It is used in this project to create the front-end of the web application.
Jinja2 : To build expressive and extensible templates. Easy to handle dynamic data.

DB Schema Design

1. user_login :
   user_id(INTEGER, Primary Key), username(STRING(50)), password(STRING(80))(hashed)
2. venue:
   venueid(INTEGER, Primary Key), venuename(STRING(50)), place(STRING(80)), capacity(INTEGER), contact(INTEGER), shows(Relation)
3. show
   showid(INTEGER, Primary Key), showname(STRING(50)), showtime(STRING(40)), showtime(DATE), rating(INTEGER), tags(Text), price(INTEGER), Venueid(INTEGER,

ForeignKey), booked(INTEGER)

4. Bookings
   username(STRING(50),ForeignKey,PrimaryKey), showid(INTEGER, ForeignKey), venueid(INTEGER, Primary Key), numoftickets(INTEGER)

- user_login: stores user login information, including the user_id, username, and password. The password is hashed for security.
- venue: stores information about venues, including the venueid, venuename, place, capacity, contact, and a relation to shows.
- show: stores information about shows, including the showid, showname, showtime (as both a string and a date), rating, tags, price, a foreign key to venueid, and the number of tickets that have been booked for that show.
- Bookings: stores information about bookings, including the username, showid, venueid, and numoftickets

API Design

API not Implemented

Architecture and Features

The project contains a python package named ‘application’ which has forms.py, controller.py, models.py, login.py and templates folder which has all the templates. Database instances are created in the instance folder. app.py is used to run the project, setup.py to import all the models and create tables in the database. Virtual environment is required to run the project. Requirements.txt contains all required packages and libraries with version.

Nearly all the core features are Implemented. Separate user login and admin login. Admin can add multiple venues and multiple shows at every venue. Users are able to see all the venues and respective shows and can book multiple tickets for the same. Users can not book tickets if the booking capacity is full.These features are implemented using various Flask extensions like Flask-WTF, Flask-Login etc.The templates are created using Jinja2, HTML and Bootstrap for CSS. The data is stored in a SQLite database using SQLAlchemy.

Video

https://drive.google.com/file/d/1Iwk_fY3yUuu6iNi5UiOjpNoWpmM-0DqC/view?usp=sharing