Modern Application Development -I

Ticket Show Booking Application

Author

Shashwat Kumar Trivedi

21F3001174

21f3001174@ds.study.iitm.ac.in

Link to presentation:

https://drive.google.com/file/d/1GGr-9QeGc9WrUaiXCtg7_ptB5N8zQkuC/view?usp=sharing

Architecture, Features and Technologies used:

A brief description about the general structure of the application:

- **Application**: This directory controls the overall functioning of the website. It has all the necessary python files needed for proper functioning of the backend server.
- **Static**: This directory contains the static components of any webpage which include CSS and image files.
- **Templates**: This folder has all the necessary html files, which give the basic structure to the website.
- **main.py**: This python file imports necessary function, class, or variable from the module application. It is also responsible for importing *flask* and running the basic server required for our application inside of a virtual environment.

This application uses basic JavaScript, HTML and CSS for frontend while using flask and flask-SQLAlchemy for the backend functioning.

Details about **application** directory/module:

The application module holds the initializing empty python file *"__init__.py"* , *config.py* (holds the basic configuration details of our application), *controllers.py* (describes all the endpoints), *database.py* and *models.py*.

- **controllers.py:** In short, this includes all the endpoints of the web application. It uses libraries like *SQLAlchemy*, *flask*, *application.models*, *statistics* and *matplotlib.* The names of the endpoints are self-explanatory for their respective function.

DB SCHEMA DESIGN

- **models.py:** In total there are five tables defined in this python file which imports the database from application module. The Tables and Relationship models defined are as follow:
    - **SHOWS:** This table has the columns ID, name, rating, tags, price, time, VID, rem_cap, rated, tot_cap, img, user_rating.
      The column *ID* is Integer, primary key, set to autoincrement and non-nullable.
      The column *name* stores names of all the shows in the form of string.
      Column *rating* has the rating given by the admin to that show in float format.

Column *tags* contain all the tags provided by the admin in string formats.
Price has the float value provide by the admin.
Start_time has the starting time of a show stored in the python datetime.time format.
End_time has the ending time of a show stored in the python datetime.time format.
Date has the date of the show entered by the admin.
VID acts as the foreign key to the primary key '*ID'* in table *'venues*'.
Rem_cap holds the remaining capacity of the show.
Tot_cap holds the total capacity of the show entered by the admin.
Img holds the URL of the image that the admin wants to display on the user dashboard. In case this link is not provided, a default link of and image will be stored.
User_rating holds the rating provided by the users; it takes the mean of the ratings.
The admin rating could have been included in this, as when we were taking mean for the user_rating entries from the UserShowRate, we could have added the rating of the admin in the numerator and added 1 to the denominator in the formula.
Rated holds the value '1' if that show is rated and '0' by default i.e., the show has not yet been rated by any user present in our database.

- **USERS:** This table has the following columns with their properties assigned as follows:
  The column *ID* is Integer, primary key, set to autoincrement and non-nullable.
  The column *name* stores names of all the users in the form of string.
  The column *username* stores the unique username for all the users which is crucial factory when it comes to logging in.
  Password column holds the password in string format for the respective user.
  Visits is the primary reference defined for the relationship in *flask-SQLAlchemy* on *SHOWS* with back reference '*mob*' and uses the intermediary table *USER_SHOWS* already defined.

- **VENUES:** This table has the following columns with their properties assigned:
  The column *ID* is Integer, primary key, set to autoincrement and non-nullable.
  The column *name* stores names of all the venues in the form of string.
  The column place stores the place or location in the form of string.
  Shows is the relationship between shows and venues, backreference is 'venue'. It defines the foreign key behavior for *VID* field in *SHOWS*.

- **UserShowRate:** This table has two columns *users_id* and *shows_id* which act as foreign keys for users.ID and *shows.ID* respectively. The combination of these two fields acts as a *unique ID* that states that a show is rated by a user or not, and if it is rated it stores the rating of the show for that user.
  Seats holds the number of seats booked by that user for that show.

- **USERS_SHOWS**: It is not a model rather a table defined inside of out model.py, this is a simple intermediary table that helps establish many-to-many relationship between users and shows set to not nullable.
  The field '*users_id'* is the foreign key to the primary key *ID* field in *'users' table.*
  The field *'shows_id*' is the foreign key to the primary key *ID* field in '*shows 'table.*