Author:

Name : D.ANIRUDH NARAYAN

Roll no: 21f1001877

Email_id : 21f1001877@ds.study.iitm.ac.in

I am a student who has an extreme interest in Web development and Data Science. I enjoyed doing this project.

Description:

This project deals with the development of a Ticket Show Application similar to “Book My Show” which enables you to create theatres, apply CRUD operations on it and to book theatres. Reminders are also made available for the User.

Technologies Used :

1) Advanced version of Vue-js is used which is Vue-CLI.
2) Bootstraps-for decorating the components of Navbar,Theatre and for styling purposes.
3) Flask-used as backend : used this for defining end-points and api.
4) Flask CORS: to allow smooth functioning between frontend and backend and to handle Cross Origin Requests without error.
5) Impleted RBAC for the entire application. Two different interfaces for User and Admin.
6) Used Flask for Token-based-authentication and for the hashing purposes used in that.
7) Flask SQL-Alchemy to define the database properly.
8) Used Sqlite3 to open the database and to check all the contents in that.
9) Redis : Used as a message broker for Celery.
10) Caching using Redis as a performance measurement.

DB Schema Design:

1) There are three tables named User ,Admin,Theatre,Shows.
2) The table User contained id (integer),emailid (string 80) ,password (string 80).
3) The table Admin has id (integer),name (string 80) password (string 80).
4) The table Theatre has id , venue name (string 80) ,place (string 100) ,location (string 100),capacity (integer).
5) The table Shows has id ,show_name (string 80) , ratings (float),timings ,price (float) ,tags (string 255),

API DESIGN:

1) Authetication is done for every login and returns a token.

2) Every requests (GET,POST,PUT,DELETE) needs the token for the process to take place.

3) Definied the following end points :

a) /signup: which takes name,emailid,password and stores it in the database.

b) /admin_login: which authenticates whether the credentials are matching with those in the database and also returns a token to the frontend.

c) /get_theatre: retrieved the theatre data from the backend.

d)/shows/<int:show_id> : which retrieves the show corresponding to the user and gives it to the frontend through a ‘GET’ request.

e)/delete_show/<int:show_id> which deletes the shows corresponding to the show_id through ‘DELETE’ method.

f) Other endpoints are defined the in app.py file present in the backend folder.