Author

Vyshakh G Nair
21f1000839
21f1000839@ds.study.iitm.ac.in
My name is Vyshakh G Nair, a Final -year student at NSS College Of Engineering, Palakkad, Kerala. Artificial Intelligence, Machine Learning, and Data Science are my core interests.

Description

This project involves developing a Vue.js-based user interface for a Movie booking management system. The interface allows users to view their bookings, cancel bookings, and rate their experiences.

Technologies used

1. Vue.js: JavaScript framework for dynamic UI components.
2. Vue Router: Client-side routing for seamless navigation.
3. Axios: HTTP library for fetching data from the backend.
4. Flask: Python micro web framework for backend services.
5. Flask-SQLAlchemy: Simplifies database operations.
6. Flask-CORS: Enables secure cross-domain API requests.
7. Chart.js: Creates interactive charts for data visualization.

DB Schema Design

CREATE TABLE booking (bookingid INTEGER NOT NULL, showid INTEGER NOT NULL, venueid INTEGER NOT NULL, userid INTEGER NOT NULL, seats INTEGER NOT NULL, PRIMARY KEY (bookingid), UNIQUE (bookingid), FOREIGN KEY(showid) REFERENCES show (showid), FOREIGN KEY(venueid) REFERENCES venue (venueid), FOREIGN KEY(userid) REFERENCES user (userid))

CREATE TABLE "show" ("showid" INTEGER NOT NULL, "showname" VARCHAR(100) NOT NULL, "showrating" INTEGER NOT NULL, "showtags" TEXT NOT NULL, UNIQUE("showid"), PRIMARY KEY("showid"))

CREATE TABLE "statushistory" ("userid" INTEGER NOT NULL, "lastlogin" TEXT, "status" INTEGER, PRIMARY KEY("userid"))

CREATE TABLE "venue" ("venueid" INTEGER NOT NULL, "venuename" VARCHAR(100) NOT NULL, "venueplace" TEXT NOT NULL, "venuelocation" TEXT NOT NULL, "venuecapacity"

INTEGER NOT NULL, PRIMARY KEY("venueid"), UNIQUE("venueid"))

CREATE TABLE "venueshow" ("venueid" INTEGER NOT NULL, "showid" INTEGER NOT NULL, "timing" TEXT NOT NULL, "price" INTEGER NOT NULL, "showdate" DATE NOT NULL, "venueshowid" INTEGER NOT NULL UNIQUE, "bookedseats" INTEGER NOT NULL DEFAULT 0, FOREIGN KEY("showid") REFERENCES "show"("showid"), FOREIGN KEY("venueid") REFERENCES "venue"("venueid"), PRIMARY KEY("venueshowid" AUTOINCREMENT))

API Design

I have developed a comprehensive Flask API for a movie booking system that facilitates user authentication, venue and show management, ticket booking, analytics, and more. Users can register, log in, and reset passwords. Venues and shows can be created, updated, and deleted. Ticket searches are enabled based on genres, cities, and dates. Users can book and cancel tickets, affecting seat availability. Ratings can be added to bookings, impacting show ratings. Automated reports and notifications are sent via email, and administrators receive CSV reports of show data. JSON Web Tokens and Redis caching ensure security and efficient data handling. Celery handles background tasks like emailing and report generation, enhancing the system's functionality.

Architecture and Features

The project is organized with a main database file named movie.db. The Flask application logic is contained within app.py, where the API endpoints, controllers, and models for the database are defined. The project directory also contains Vue.js components in separate .vue files. The configuration for Flask and other settings are stored in a YAML file within the same directory as app.py. The Celery configuration is integrated directly within app.py to handle background tasks. The Vue.js frontend follows the standard structure of a Vue.js CLI project.

Regarding features, the default functionalities include user registration, authentication, venue and show management, ticket booking, and cancellation. Additionally, the implementation covers advanced features such as automatic daily and monthly job scheduling using Celery, which sends emails to inactive users and generates reports for users and administrators. The system uses JSON Web Tokens (JWT) for user authentication and Redis caching for efficient data storage. Vue.js components enable user interactions and display information. The integration of Celery enhances the system's efficiency by executing tasks asynchronously, ensuring seamless email delivery and report generation.

Video

Link : https://drive.google.com/file/d/1bn1IDbpK-5apo4aGIqVfHbEYVmbyjdai/view?usp=sharing