Author

Shailesh Kumar
21f1004597
21f1004597@ds.study.iitm.ac.in
Data Science Enthusiast

Description

This project involves building a Vue.js web application for managing venues, and movies. The application has two user roles: administrators and users. Admin can add, edit, and delete venue details and movies through modals. Users can view and book venues and movies.

Technologies used

- **Vue.js**: A JavaScript framework for building user interfaces. It allows for creating dynamic and responsive web applications with its component-based architecture.
- **BootstrapVue**: A library that integrates Bootstrap components with Vue.js, providing a convenient way to create visually appealing and responsive user interfaces.
- **Vuex**: A state management pattern and library for Vue.js applications. It centralizes and manages the state of the application, making it easier to share data between components.
- **Vue Router**: A routing library for Vue.js applications. It enables navigation between different views and components in a single-page application.
- **Axios**: A popular HTTP client for making asynchronous HTTP requests. It's used to communicate with the backend API to fetch and send data.
- **Flask**: A lightweight and flexible Python web framework. It's used to build the backend API that handles data storage, retrieval, and manipulation.
- **Flask-RESTful**: An extension for Flask that simplifies the creation of RESTful APIs. It helps in defining API resources and their endpoints.
- **Flask-CORS**: An extension for handling Cross-Origin Resource Sharing (CORS) in Flask applications. It allows the frontend to make requests to the backend from a different domain.
- **SQLAlchemy**: A powerful SQL toolkit and Object-Relational Mapping (ORM) library for Python. It's used to interact with the database and manage database operations.
- **SQLite**: A lightweight, serverless, and self-contained database engine. It's used as the database system to store and manage application data.
- **JWT (JSON Web Tokens)**: A compact and self-contained way for securely transmitting information between parties as a JSON object. It's used for user authentication and authorization.
- **Python**: The programming language used for developing the Flask backend and handling server-side logic.

DB Schema Design

db schema design link

API Design

The Flask-Restful library for Python was used to create a RESTful API adhering to the OpenAPI Specifications. All database tables have CRUD operations available through the API. Authentication tokens are used for specific requests that require them. These tokens can only be obtained from the user's account page when signed in. For further details, please –refer to the openapi.yaml file.

Architecture and Features

The architecture of Ticket-Booking follows a client-server model, where Vue.js serves as the front-end framework and Python-Flask as the back-end framework. Vue.js handles the presentation layer and manages user interactions through its MVVM architecture, while Python-Flask handles the server-side logic, such as HTTP requests and responses, asynchronous tasks, and database interactions.

The features of the application are as follows:

- User authentication: Signup and Login
- User profile: View own movie booking
- User-specific API tokens: Generate tokens to use user-specific requests
- Data export: Download user's booking and analytics as a CSV f
- Content management: Create, view, edit, and delete show
- Account management: Create, view, edit, and delete user accounts
- RESTful API: API available for venue, users and movie
- Export booking: Export booking show as PDFs
- Reminders: Receive daily reminders to book show
- Monthly engagement report: Receive a report as an email or PDF summarising engagement for the month

Video

video link