Author

Full name : **SACHAPARA NIL VIJAYBHAI**
Roll No. : **22f2001418**
Email id : **22f2001418@ds.study.iitm.ac.in**

I am a student currently pursuing my Bachelor of Engineering degree in Information Technology. I am a dedicated and passionate learner who is always eager to explore new technologies and apply them to solve real-world problems. I have a keen interest in web development and database management systems.

Description

The project is a web application that allows users to book tickets for shows at various venues. The application has a user registration and authentication system, with different levels of access for users and administrators. Users can view available shows at venues, book tickets, and view their booking history. Administrators can manage venues, shows, and user bookings.

Technologies used

- Python
- Flask framework
- Flask extensions: Flask-RESTful,Flask-SQLAlchemy
- SQLAlchemy
- SQLite database

The purpose behind using these technologies was to create a scalable and maintainable web application with a secure user authentication system and a RESTful API.

DB Schema Design

- User table: id (primary key), username (unique), email (unique), mobile (unique), password
- Venue table: id (primary key), name (unique), address, capacity
- Show table: id (primary key), name (unique), description, timing, venue_id (foreign key referencing Venue.id)
- Tickets table: id (primary key), user_id (foreign key referencing User.id), show_id (foreign key referencing Show.id), venue_id (foreign key referencing Venue.id), number_of_tickets
- Enroll table : enrolment_id (primary key), evenue_id (foreign key referencing Venue.venue_id), eshow_id (foreign key referencing Show.show_id.

The schema was designed in this way to ensure data consistency and to prevent duplication of data. Foreign keys were used to enforce data integrity.

API Design

The API was designed using Flask-RESTful and provides endpoints for retrieving and booking tickets. The endpoints are:

- POST /user/register: register a new user
- POST /login: login a user
- POST /admin/login : login a admin
- GET /venue: retrieve all venues
- GET /venue/{venue_id}: retrieve a specific venue by ID
- GET /user/me: retrieve information about the currently logged-in user
- POST /book/{venue_name}/{show_name}: book tickets for a show at a venue
- GET /book/{venue_name}/{show_name}: retrieve information about available tickets for a show at a venue
- GET /bookings: retrieve a list of bookings for the currently logged-in user

The API was implemented by creating Flask-RESTful resources for each endpoint and defining the necessary methods (GET, POST, etc.) for each resource.

Architecture and Features

The project is organized according to the Model-View-Controller (MVC) architecture. The controllers are defined in the main.py file, and the templates are stored in the templates folder. The API resources are defined in the api.py file.

The default features of the project include user registration and authentication, venue and show management, ticket booking, and booking history. Additional features include a RESTful

Video

https://drive.google.com/file/d/13ef0x3ymvn3cM0TfyYYLHDaqs5Sd4yqn/view?usp=share_link