

## Author

Suyash Srivastav

22f1001215@ds.study.iitm.ac.in

I am Suyash Srivastav currently in the Diploma level of IITM BS degree. I love programming and have a deep interest in AI and Machine Learning. This is my second coursework project.

## Description

Our project merges Vue.js for dynamic frontend interactions with Flask for powerful backend management, resulting in a cohesive Show Booking and Ticket Management System. Users can seamlessly explore shows, book tickets, and receive reminders, while administrators efficiently manage events and venues. The integration offers secure user authentication, smart booking processes, and automated communication for a comprehensive event experience.

## Technologies used

- Frontend: Vue.js
- Backend: Flask (Python web framework)
- Database Management: SQLAlchemy
- User Authentication: JWT (JSON Web Tokens)
- Password Hashing: Flask-Bcrypt
- Background Task Scheduler: Celery
- In-Memory Data Store: Redis
- Cross-Origin Resource Sharing: Flask-CORS

## DB Schema Design

The database comprises four tables tailored for efficient event and user management:

1. **show**: Stores show details like title, genre, date, time, and available tickets, linked to venues through `venue\_id`.
2. **ticket**: Tracks booked tickets with name, date, time, and quantity, associated with both shows (`show\_id`) and users (`user\_id`).
3. **user**: Manages user data such as name, username, password (hashed), and admin status.
4. **venue**: Houses static venue information, including name, location, and city.

Design Rationale:

- **Relationships**: Foreign keys establish efficient relationships between tables for seamless data querying.
- **Normalization**: The design minimizes redundancy, ensuring data integrity and efficiency.
- **User-Centric**: The `user` and `ticket` tables cater to user authentication and ticket management.
- **Event Coordination**: `show` and `venue` tables support organized event scheduling and coordination.
- **Integrity**: Unique and foreign key constraints safeguard data accuracy and consistency.

## API Design

I've developed a comprehensive API using Flask for a Show Booking and Ticket Management System. It covers user authentication, show and venue management, ticket booking, and user bookings. Key endpoints include user registration, show/venue handling, and booking management. The API ensures secure communication between Vue.js frontend and Flask backend, enhancing event management and user experience.

## Architecture and Features

The project has two main folders: backend and frontend. Backend includes Flask components like API routes, database models, and Celery tasks. Frontend is a Vue.js project with components, assets, and routing. Features include user auth, show/venue management, ticket booking, reminders, user dashboards, and responsive design. This modular approach ensures efficient development and a user-friendly Show Booking and Ticket Management System.

## Video

Here is the video link:

[https://drive.google.com/drive/folders/18kEPL\\_rdmDivvWhwiCGfvT0MeU193Jmb?usp=sharing](https://drive.google.com/drive/folders/18kEPL_rdmDivvWhwiCGfvT0MeU193Jmb?usp=sharing)