

## Author

Full Name: Md Farhan Reza

Roll Number: 21f1002213

Email: [21f1002213@ds.study.iitm.ac.in](mailto:21f1002213@ds.study.iitm.ac.in)

## About Me

I am an aspiring Data Scientist, currently in 4th year of NIT Durgapur as well as in the diploma level of IITM BS degree. I am learning subjects like Statistics, Mathematics, Machine Learning, and Application development so that I can create stories from the data.

## Description

In this project we had to create a multi-user app for booking movie tickets. Users can book movie tickets for multiple shows in multiple venues. Admin can add shows and venues in the database after which the user can book.

## Technologies used

**Flask:** A micro web framework for building web applications in Python.

**Flask-Cors:** An extension for Flask that provides cross-origin resource sharing (CORS) support.

**Flask-SQLAlchemy:** An extension for Flask that integrates SQLAlchemy, a popular SQL toolkit, into Flask applications.

**Flask-RESTful:** An extension for Flask that simplifies the creation of RESTful APIs.

**flask\_bcrypt:** An extension for Flask that provides bcrypt hashing utilities for secure password storage.

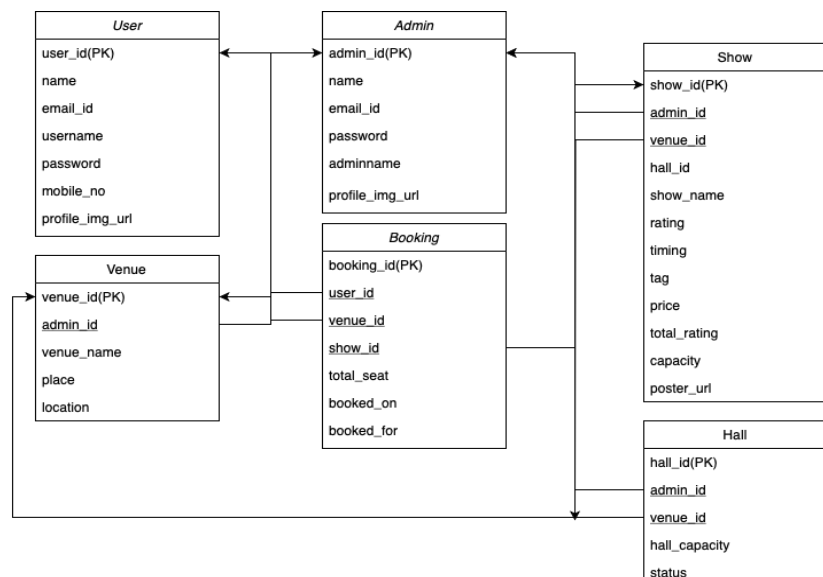
**JWT Tokenization:** A technique for securely transmitting data as JSON web tokens (JWTs) between parties in web applications.

**VueJs:** A progressive JavaScript framework used for building user interfaces and single-page applications.

**Redis:** An in-memory data structure store used as a database, cache, and message broker in web applications.

**Celery:** A distributed task queue used for background job processing and scheduling in web applications.

## DB Schema Design



## API Design

TicketShow is a multi-user app which is used for booking show tickets. My api's five primary categories were User, Admin, Show, Venue and Booking. User primarily comprises user-related operations like user login and user registration. Then, Admin consists of operations related to admin login and admin registration and other things. Show consists of show related things like adding, editing and deleting and same with Venue. Booking comprises booking the show and deleting it.

## Architecture and Features

In my submission file there are:

1. Application folder:
  - config.py-(apps configuration)
  - cache.py-(caching setup)
  - database.py-(database linking with SQLAlchemy)
  - Models.py-(models)
  - operations.py-(api implementation / controller)
  - workers.py-(used for celery)
  - token\_validation.py( for jwt token\_validation in operations.py)
2. db\_directory:
  - blogliteneew.sqlite3-(database)
3. Third is static folder which consists static file like logo
4. Templates:
  - Components
    - i. various templates which will be rendered while running the application.
    - ii. index.html
    - iii. router.js
    - iv. monthly\_report.html
5. main.py
6. documents:
  - user\_operation.py
  - ticket\_show.yaml
  - project report

### Features

1. User signup and login
2. Admin signup and login
3. User profile view
4. Show and venue management
5. Show and venue search
6. Booking the ticket
7. Daily Notification
8. Monthly Notification

**Video :** [Project video link](#)