App Development 2 Project

1 Author

- Name: Janani D

- Roll no: 310621243022

- Student Email: 21f3002138@ds.study.iitm.ac.in

About me

Greetings!, this is Janani D currently pursuing BS in Data Science And Applications from IIT Madras. I have cleared my App Development -2 theory course in last term. My current project in app development-2 has given me valuable insights into frontend technologies, igniting a desire to continuously learn and improve in this field.I am excited about my journey ahead, eager to explore new opportunities and make a positive impact in the world of data science and application development.
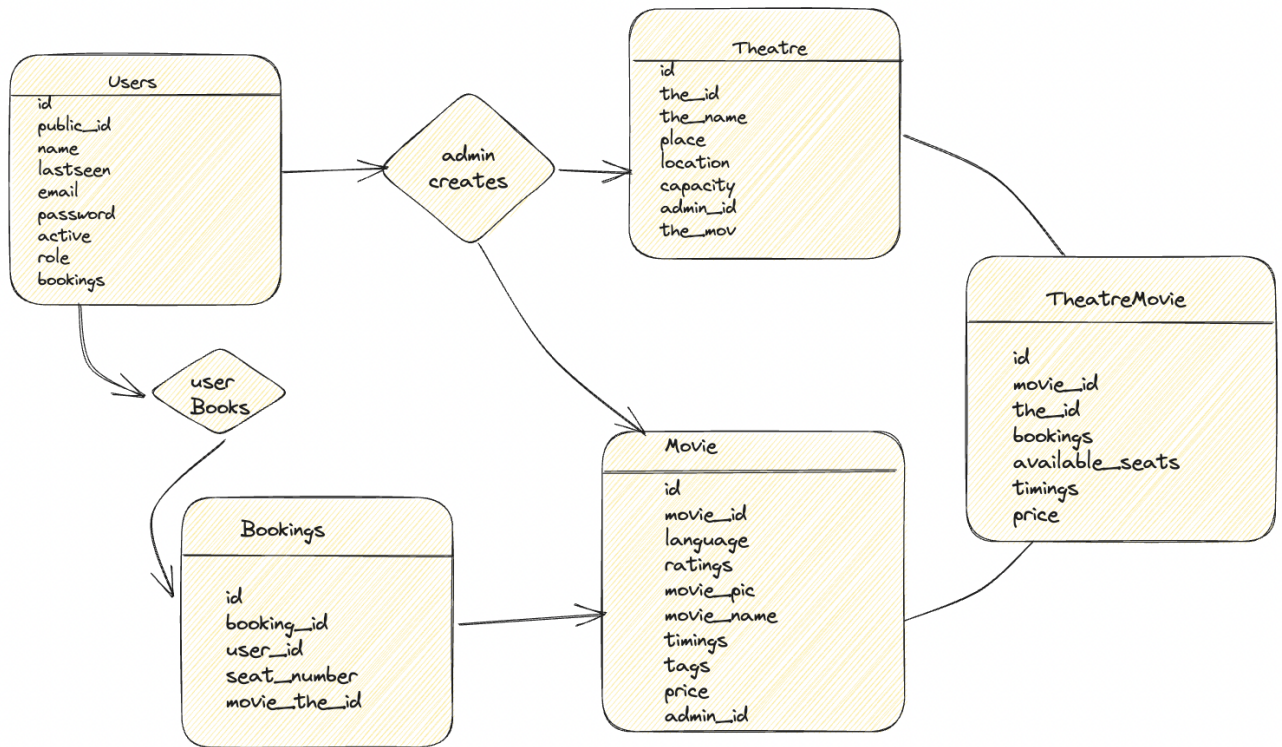
2 Description

This project is based upon developing a ticket booking application for movies. This application has both user and admin logins. The admin can create theatres, movies and add movies into the theatres. The users can book tickets for movies in desired theatres.

3 Technologies Used

- **Flask** Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2.

- **SQL-Alchemy** SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL. It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access.

- **Vue js** It is a popular and progressive JavaScript framework used for building user interfaces and single-page applications.

- **Celery** Celery is a distributed task queue system for Python that allows you to execute tasks asynchronously in the background.

- **SMTP** It is used to send emails.

- **Python 3.11**

- **SQLite3**

- **Redis** Redis is an open-source, in-memory data structure store known for its exceptional performance and versatility.

- **JWT** It is a compact and self-contained method for securely transmitting information between parties as a JSON object.

4   DB Schema

The Users table (id, public_id, name, lastseen, email, password, active, role, bookings) connects via "admin creates" to the Theatre table (id, the_id, the_name, place, location, capacity, admin_id, the_mov) and via "user Books" to the Bookings table (id, booking_id, user_id, seat_number, movie_the_id). The Movie table contains (id, movie_id, language, ratings, movie_pic, movie_name, timings, tags, price, admin_id). The TheatreMovie table contains (id, movie_id, the_id, bookings, available_seats, timings, price).

5   API Design

There are CRUD operations for Theatres, Movies, Bookings, Users where I can perform GET, PUT, POST and DELETE operations. These are fetched by frontend from backend.

6   Architecture and features

The architecture is simple. In this application the admin can create movies and theatres. The admin can add the movies to the desired theatres with timings. The user can book ticket to any movie in his desired theatre. The admin can create,delete and update both theatres and movies. Emails are send to user every month and after every login as engagements.

7   Video

Video