SHOWDEKHO APP REPORT

Author

RAJ SIRUVANI
21f1005673
21f1005673@ds.study.iitm.ac.in
A Graduate in Aerospace Engineering, Interested in Coding, Web Development and Blockchain Technology and Web3 Gaming

Description

Basically, Design & Develop a Web2 App, that Allows users to Book Tickets for the Shows created by the admin, Allowing users to Search on Movies and Shows based on Filters like Director/Ratings/Location, And Admin has CRUD Rights on Movies, Theates,Shows.

Technologies used

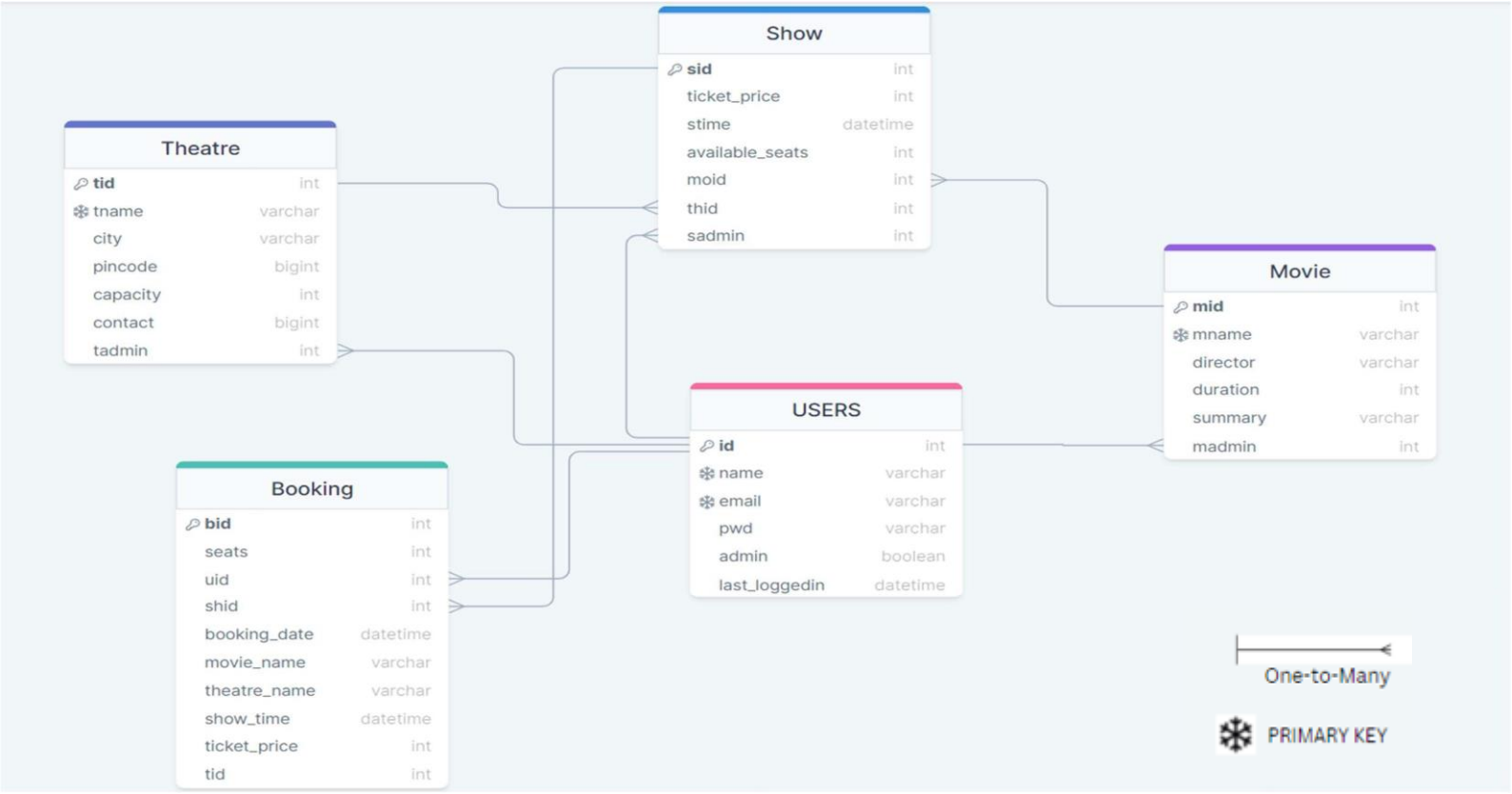
Flask – For Backend, Models, API(Heart of the Project)
Flask_SQLALCHEMY – DB Models, Connecting with backend Server
Flask-RestFul – For CRUD on DB Models, API
FLASK-JWT – For JSON Web Token based Authentication of users
VUE CLI – For UI
CELERY(Worker & Beat) – For Triggering Asynchronous (backend) jobs, Scheduling jobs
Redis – For Caching
SMTPLIB – For Emailing the Users
PDFKIT – Converting HTML Templates to PDFs

API Design

1. /logi – USER LOGIN
2. /usersignup – USER SIGNUP
3. /home – Lists All the Movies Available
4. /movie – CUD(Create, Update & Delete for Movies) GET for Receiving Movie data by id
5. /theatre – CRUD Operation for theatres by Admin
6. /theatre/:id – Getting a Theatre info by theatre_id
7. /shows – CRUD Operation for Shows by Admin
8. /shows/:id – Receiving Shows for a particular Movie(User Only)
9. /show/:id – Getting A Show info by show id(ADMIN ONLY)
10. /booking – User Booking a Ticket(USER ONLY)
11. /report/:id – Admin exporting a Theatre Report

3,4 -> Movies l 5,6 -> Theatres | 7,8,9 -> Shows

DB Schema Design



The Booking table is like a cold storage, when the admin deletes the theatres/Shows/Movies we want to preserve that data somewhere if a user has booked it, so that we could use that data while sending a mail report or Also while theatres exporting csv, we would like to keep a track of movies/shows when deleted by admin.

Architecture and Features

In the Root Directory, we will be having app.py, celery_worker.py & local db, Inside the app.py I have everything related to backend, from db models, to controllers to API and CELERY schedules, etc..,

I then have a dedicated folder for frontend, Inside the frontend Folder ./src, I have organised into 3 different folders, for routers, views and components for UI design.

Features: Different Views for User & Admin, CRUD Operations for Movies, Theatres and Shows by Admin for(ShowDekhoApp), When a Show is HouseFull, I will stop taking any more bookings, Different Users can login and use the app, Have provided an export csv feature for theatres that the Admin could use to see the theatre report, Daily reminders to all the Users if they haven't logged in the app, Every User will get a Monthly Maintenance report about the bookings made by them, Expenses by them, etc.. I have used Redis for Caching.

Video

https://drive.google.com/file/d/1zM7l2ljgXJYG6aZ-dt-p40VgBGYm3tO0/view?usp=sharing