# Ticket Tales (Mad-2 Project Report)

## Author


- **Name:** Vishesh Phutela
- **Roll No.:** 21f3001040
- **Email:** 21f3001040@ds.study.iitm.ac.in
- **About Me:** I love to Design Stuff, and this passion for designing also made me a graphic designer during my journey as a Data Scientist I am also an avid gamer. I love to play competitive strategy games and first-person shooters. Lastly, I love learning. Every day I push myself to learn something new, whether that be about machine learning, software engineering, or miscellaneous facts about the universe.


## Description

Ticket Tales is a single-page web application that allows you to book and manage tickets online for various venues and theatres, it uses Vue.js as its main frontend component to provide a user-friendly interface while leveraging the backend powers of the Flask also using Celery and Redis for various backend jobs, tasks, and caching with sqlite3 as a database.

## Technologies Used

- **Python:** Python is used as the base programming language for this project
- **Front-end:** Vue.js, Html and css
- **Database:** SQLite3 is used as a database
- **Mail-Hog:** SMTP Server Simulator for testing e-mails
- **Redis:** It is used for celery jobs queuing and caching

| Python Package | Role |
|---|---|
| **Flask** | Flask is used as the base backend for this application |
| **Flask-SQLAlchemy** | It is used for creating db models and ORM |
| **Flask-Restful** | It is used to create API's |
| **Flask-CORS** | It used to solve Cors issues |
| **JWT-Extended** | It is used to create token-based Authentication |
| **Celery-Redis** | Celery is used to create scheduled jobs and Redis as a task queue for celery |
| **Flask-Caching** | It is used for backend caching with Redis Cache |
| **Jinja 2** | It is used in passing data for generating pdfs from html |
| **Pandas** | Pandas is used for generating csv files |
| **Pdfkit** | It is used to generated pdfs from html, this library uses wkhtmltopdf to generate pdfs |

These are some of the major libraries used in this project while all the libraries used are listed in requirements.txt file.

## API Design

The Ticket Tales API follows RESTful principles and supports CRUD operations for the following resources:

Users, Admins, Theatre, Shows, Bookings, Search.

The API uses the HTTP methods GET, POST, PUT, and DELETE to perform the CRUD operations. The endpoints are designed to be self-explanatory and follow the standard RESTful conventions with proper authentication for users and admins for the functioning of app.

## Architecture and Features

```
├── app.py
├── application/
│   ├── config.py
│   ├── controllers/
│   │   ├── api --> Contains all the code for APIs
│   │   ├── controllers.py
│   │   └── validate.py
│   ├── marshal.py
│   ├── models.py
│   ├── tasks.py
│   └── workers.py
├── instance/
│   └── TicketTalesV2.sqlite3
├── requirements.txt
├── static/
│   ├── css --> Cantains all the css
│   ├── img --> Conatins all the required images
│   └── js --> Conatins all the js files
├── templates/
│   ├── index.html
│   ├── monthly_report.html
│   └── theater_pdf.html
└── venv
```

This is the overall Architecture/Structure of the app.

**Features:**

All the required features of the problem statement is incorporated and are working properly.

**Extra Features:**

- Invoice generation of the booking for users with a valid QR code
- Aesthetic UI, I've not used bootstrap but rather wrote manual css for better design and Aesthetics.
- Separate dashboards for User and Admins.

Fig. Architecture of TicketTales

## Database Schema Design

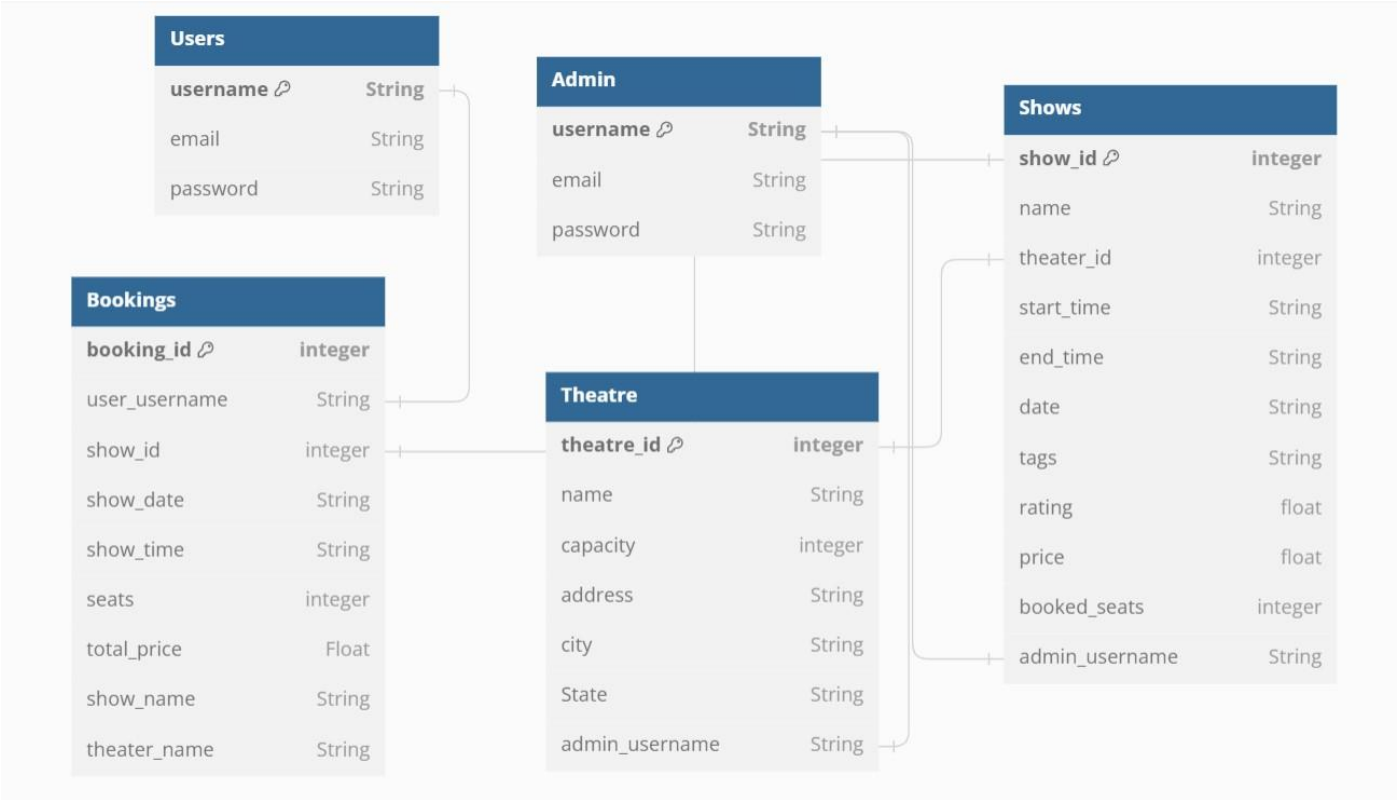


Fig. Database schema of TicketTales

## Video : Link