# Movie N More Documentation

## Introduction

This documentation provides instructions for using the Flask app, which allows users to book movie tickets at different theatres and venues. The app uses SQLite3 and has five database models/tables: User, Movie, Theatre, Venue, and Order. Frameworks used are Flask , Jinja2, Bootstrap( at many places), CSS and Javascript(some parts).

## Installation

1) Unzip the folder.
2) Create a virtual environment for the app using command: ‘python3 -m venv env’
3) Activate using : ‘source env/bin/activate’. Install required dependency using: ‘pip install -r requirements.txt’
4) Run using: ‘python main.py’ . Then go to : ‘http://localhost:5000/’ - register page of Movie N More, from there you can navigate to other pages. Database named ‘database1.db’ is also created if it does not exist.

## User Management

Users can sign up for an account by filling out the registration form at ‘/ ‘. User information is stored in the User table in the database, which has columns for id, username, email, and password. Users can log in to their account by visiting ‘/login’. After logging in, users can choose a city (by default it is Mumbai) and based on that view and book movie tickets at different theatres and for different movies at a given Venue, decided and set by Admin. Booked Venue (past or upcoming) is stored in Orders table. User can see after login.

## Movie Information

The Flask app includes a Movie table in the database, which has columns for ‘id’, ‘name’, ‘actors’, ‘image’, ‘duration’, ‘about’, ‘imdb_rating’, ‘release_date’, and ‘director_name’. The Movie table stores information about movies, including their cast, duration, release date, and director. Users can view this information on the app's ‘/searchbymovie’ pages. Image is stored in BLOB format in table, duration- Time, imdb_rating between 0.0 to 10.0, release_date in Date, actors in JSON array and others in String or Integer. All these data can be changed by admin whenever required.

## Theatre and Venue Information

The Flask app includes two tables in the database for storing information about theatres and venues. The Theatre table has columns for id, name, capacity, and city, id is primary key capacity is int and rest is String. Venue table has columns for id, movie_id, theatre_id, date, time, and cost. Id is primary key, movie_id-foreign key for table Movie(id), theatre_id-foreign key for table Theatre(id). date is Date, time is Time, cost is int but should be greater than 80.

The Theatre table stores information about theatres, including their capacity and location, while the Venue table stores information about specific showings of movies at different theatres

## User  And Other Pages

The Flask app includes several user pages that allow users to view and book movie tickets. These pages are built on top of the base2.html template processed by Jinja2, which provides a consistent layout and design for the app's user-facing pages. User specific html pages are home, order, searchbymovie, searchbytheatre and book. User cannot visit admins page as there is custom decorator @admin_required on every admin url, it will show 'Unauthorized' if tried.

## Admin Pages

The Flask app includes several admin pages that allow admins to manage the app's content and user accounts. These pages are built on top of the base4.html template, which provides a consistent layout and design for the app's admin-facing pages. These pages are addtheatre, admin, createvenue, editmovie, edittheatre, editvenue, newmovie, theatre, venue.

## Jinja Filters and Functions

Jinja filter named base64_encode has been used to convert BLOB to base64 for displaying in html pages. Another filter named seatsbook has been used which returns seats booked in venue by taking venue id as input. Almost 14 other python function are present in views.py:

seatsbooked – takes venue id as input and return total seats that are booked into it.

returntheatres and returnmovies – takes city name and return theatres and movies in it.

booktickets – venue id and ticket and add a row in Order table for logged in user(book)

venuebymovie and venuebytheatre – take theatre is and venue id and returns future venues

updateimage- admin uses to update image of a movie.

Allmovie, alltheatre, allvenue- just return all movies, theatres and venues respectively.

Returndate, returntime – take date in str and return in date format(YYYY-MM-DD), takes time in str and returns in time format (HH:MM) format.

Userobject – takes user id and returns respective user object.

## Conclusion –

This basic overview of the Flask app's functionality and database structure. For more information, please refer to the code. Video Link (no voice): https://drive.google.com/file/d/1okw9xoDr7pmr_IZQhDUx30LVAvYsug8k/view?usp=sharing