Author

DINESH KUMAR KUMAWAT
21F1001956
21f1001956@ds.study.iitm.ac.in
I am Dinesh Kumar, Currently I am pursuing Diploma level Courses.

Description

Admin can manage shows and venues, including creating, editing, and deleting them. They can also view the summary of rated shows.
Users can view all available shows and venues, and search for specific ones using a search bar.

Technologies used

Flask:- web framework oof python to develop web applications.
flash:- to display temporary messages to user and admin for incorrect actions
render_template:- to render HTML templates from template folder
Request:- to handle HTTP requests. It provides methods to access data sent by the client in the request.
redirect:- to redirect the user to a different URL
url_for:- to generate URLs for a specific function
datetime, time:- to work with dates and times.
matplotlib.pyplot:- to create chart between show and rating
pytz:- to working with time zones (like converting UTC to IST time zone)
re:- regular expressions for validation of string pattern
SQLAlchemy:- Object-Relational Mapping (ORM) library for storing data of application

DB Schema Design

Table Admin:-
admin_id → Integer, primary_key
username → String, unique, Not Null
password → String, Not Null

Table Showvenue :-
show_id → Integer,ForeignKey,primary_key
venue_id → Integer,ForeignKey,primary_key
n_seat → Integer
d_price → Integer

Table Venue
venue_id →Integer, primary_key
venue_name → String, unique, Not Null
Place → String,Not Null
Location → String, Not Null

Capacity → Integer,Not Null
admin_ → Integer, ForeignKey('admin.admin_id')

Table Show:-
show_id → Integer, primary_key
show_name → String, unique, Not Null
rating → String, Not Null
start_time → Time, Not Null
end_time → Time, Not Null
tags → String, Not Null
price → Integer, Not Null

Table User:
user_id → Integer, primary_key
username → String, unique, Not Null
password → String, Not Null

Table Booking:
booking_id → Integer, primary_key
Number → Integer, Not Null
total → Integer, Not Null
show_id → Integer, Not Null
venue_id → Integer, Not Null
user_ → Integer, ForeignKey('user.user_id'))

Relationship:-
Admin -> Venue :- one to many → Admin can create multiple venues
Venue -> Show :- many to many → One Venue can have multiple shows and vise versa through Showvenue table
User -> Booking :- one to many → One user can book multiple tickets

Architecture and Features

All Controllers are in main.py file , Tables in database.py file and html files in template folder and chart and image are in static folder.

Features implemented:-

- Admin login and User login :- using a simple HTML form with username and password
- Venue Management :- Admin can create, edit, and remove the venues
- Show Management :- Admin can create, edit, and remove the shows (Dynamic Price)
- Booking show tickets :- User can book multiple tickets for a show at a given venue Search for shows/venues :- User can search for venues and shows and also see for a given timeframe (Venue Home View)
- Validation :- All form inputs fields are validated through python

Video

https://drive.google.com/file/d/1RImWztX1XlzJo7ClzZQ-0__v7U_cqqVz/view?usp=sharing