Name: Shagun Niranjan

Roll Number: 22dp2000121

Email: 22dp2000121@ds.study.iitm.ac.in

Introduction:

Myself Shagun Niranjan current Pursuing Diploma in Programming from IIT Madras and also Pursuing MCA from VIT BHOPAL. I possess technical skills in HTML, CSS, Java and Python.

DESCRIPTION: A working model of Ticket Booking Application which allows admin to do CRUD operations (create, read, update, delete) in Venues and Events. And user to book an event and view his bookings.

Technologies used:

Python: To Develop clear, simple and reusable code.

Flask: Framework for the application.

SQLite: It is used to create and handle database.

Bootstrap: To style the application.

HTML & CSS: The basic Structure of the Application.

DB Schema Design:

Users Table

Id (Primary Key)	Integer	To Identify every user uniquely
userType	String (100)	To choose the user type between user and admin
email	String (100)	To shore the User email
name	String (50)	To store the User name
password	String (200)	To store the user password

Venue Table

Id (Primary Key)	Integer	To Identify every Venue uniquely
v_venueName	String (200)	To store Venue Name
v_place	String (60)	To store Venue Address
v_location	String (60)	To store Venue City
v_capacity	Integer	To store the capacity of Venue
v_organizer	String (60)	To store the email of the user whose userType is Admin
admin_id (Foreign Key)	Integer	To store the admin id who created the particular venue

Event Table

Id (Primary Key)	Integer	To Identify every Event uniquely
e_title	String (200)	To store the Event name
e_time	String (50)	To store the Time of event
e_tags	String (400)	To store all the tags of event
e_price	Integer	To store the price of one ticket
e_location	String (60)	Location of event
e_venue	Integer	To store the venue name in which event is created
Venue_id (Foreign Key)	Integer	To store the venue id

Booking Table

Id (Primary Key)	Integer	To Identify every Booking uniquely
b_name	String (100)	To store the user name who booked the show
b_email	String (100)	To store the email of user who is booking the event
b_eventName	String (200)	To store the booked event name
b_price	Integer	To store the total price user have to pay
user_id (Foreign Key)	Integer	To store the user id who booked the show
event_id (Foreign Key)	Integer	To store the event id
venue_id (Foreign Key)	Integer	To store the id of venue where the booked event is happening

API Design:

I have created CRUD (get, post, put, delete) endpoints for User, Venue and Event. I have used flask-restful along with API, @marshal_with, Resource, fields, reqparse and abort.

Architecture and Features:

- instance
    - TicketStow.sqlite: - SQLite3 file
- static
    - bootstraps.css: - bootstrap file to add style in the application.
    - logo.png: - Logo of the application in png format.
    - style.css: - all the Custom styles.
- templates
    - adminhome.html: - Admin home page.
    - allBookings.html: - All the tickets which user buys will show in this page.
    - base.html: - This is the base page for all the html files. It contains the navbar.
    - book.html: - User can select the count of ticket he has to buy in this page.
    - createEvent.html: - When admin click the create Event button then this page will render.
    - createVenue.html: - When admin click the create Venue button this page will render.
    - login.html: - The first page which will render.
    - signup.html: - Signup page for the use Users.
    - successBooking.html: - This page will render when user select the count of tickets. This will display the total amount of money he has to pay.
    - updateEvent.html: - for admin to update Event
    - updateVenue.html: - For admin to Update Venue
    - userhome.html: - This will render when the userType is user.
- app.py

Video: https://drive.google.com/file/d/1P3EmU32_O73PUyOsha0_Dpps08euHf3_/view?usp=share_link