

Author

Chaitanya Gaur

21F1003275

21f1003275@ds.study.iitm.ac.in

Data Science enthusiast with interest in domains like Machine Learning and Artificial Intelligence.

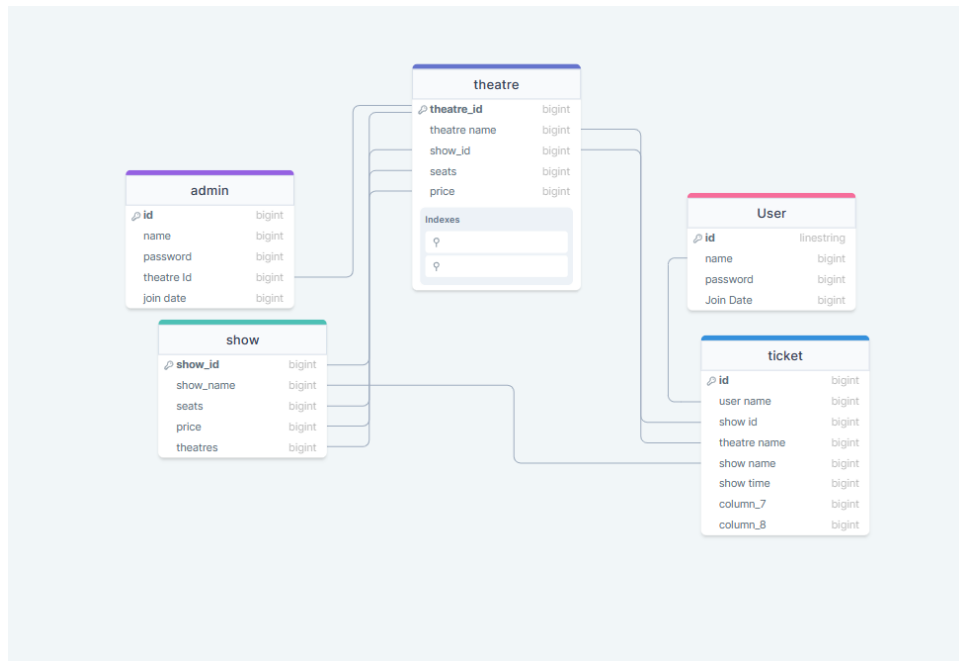
Description

Movie Ticket Booking Web Application project based on Python based backend framework Flask, using various Libraries and framework related to Flask, like Flask-SQLAlchemy for Database integration (SQLite based Database) and Flask Login for Login interfaces.

Technologies used

- **Flask**: used for small to medium-sized applications, as it provides a minimalistic approach to web development, allowing developers to build applications quickly and with flexibility.
- **SQLAlchemy**: SQLAlchemy is an open-source SQL toolkit and ORM (Object-Relational Mapping) library for Python that provides a set of high-level API for connecting to SQL databases, executing SQL queries, and managing database transactions.
- **Flask-Login**: Flask-Login provides a simple and secure way to manage user authentication and session management in Flask applications.
- **Flask-WTF**: Flask extension that provides integration between Flask and the popular Python form library, WTForms. Flask-WTF makes it easy to create and render web forms in Flask applications and provides a range of features that make it easier to handle form submissions, validation, and error handling.
- **WTforms**: WTForms is a powerful and flexible library that makes it easy to handle web forms in Python web applications, providing a range of features that make it easier to handle form submissions, validation, and error handling.
- **Flask bcrypt: Flask** extension that provides bcrypt hashing utilities for password hashing in Flask applications. Bcrypt is a popular password-hashing algorithm that is known for its security and is widely used in web applications.
- **Jinja templating**: Jinja templating is a powerful and flexible tool that makes it easy to generate dynamic content in Python web applications, providing a range of features that make it easier to create reusable and modular templates.

DB Schema Design



Architecture and Features

The above Flask app uses the Model-View-Controller (MVC) architectural pattern. The models are represented by the classes defined in the code (e.g., user, admin, theatre, and show). The views are the HTML templates that are rendered by Flask (e.g., render_template). The controllers are the functions that handle the incoming requests and route them to the appropriate models and views (e.g., load_user, register, login, and logout). The Key features of MVC based architectural systems are:

- **Reusability:** The modularity of the MVC pattern allows for components to be reused across different parts of the application, which can save development time and reduce errors.
- **Flexibility:** The MVC pattern allows for greater flexibility in design and implementation, as the various components can be modified or replaced without affecting the rest of the application.

Video

Google Drive: https://drive.google.com/file/d/1U-4MEwgZSFRAF6trSIRsxv9Vi83r011F/view?usp=share_link

YouTube Link: <https://www.youtube.com/watch?v=iUm9qXIP1VA>