Author

Muhammed Aslam A
Roll number: 21f1005889
Student email: 21f1005889@student.onlinedegree.iitm.ac.in

I am a computer science student with a passion for web development and building scalable, efficient applications.

Description

The project is a simple Blog Lite web application that allows users to create posts, follow other users, and interact with their posts by liking and commenting on them.The application also includes a REST API that allows external applications to access user engagement data.

Technologies used

- Flask: web framework for building the web application
- Flask-SQLAlchemy: ORM for connecting to the database and performing CRUD operations
- Flask-RESTful: extension for building RESTful APIs
- Flask-Login: extension for user authentication and authorization
- Werkzeug for password hashing
- SQLite: database for storing user, post, and engagement data

DB Schema Design

The database used in this project is a SQLite database. The following tables were created:

- Users: Stores information about users, including their name, email, and password hash.

- Posts: Stores information about posts, including the text of the post and the user who created it.
- Comments: Stores information about comments, including the text of the comment and the user who created it.
- Likes: Stores information about likes, including the user who liked the post and the post that was liked.
- Followers: Stores information about followers, including the user who is following and the user who is being followed.

The reason behind designing the database in this way was to ensure that all data is stored in a logical and organized manner. The Users table stores information about the users of the application, and the Posts, Comments, Likes, and Followers tables store information about the various activities that take place within the application. The relationships between these tables are as follows:

- Users has many Posts, Comments, and Likes
- Posts has many Comments and Likes
- Comments has many Likes
- Followers has many Users

This design allows for easy retrieval of information, such as all of a user's posts or all of the comments on a particular post. Additionally, it allows for easy tracking of likes and followers for each user. Overall, this database design was chosen to best support the functionality of the application and make it easy to manage and maintain the data.

API Design

The API was designed to retrieve information about a user's engagement on the platform, such as the number of posts, comments, likes, followers, and followed users. The API includes a single endpoint that takes a user ID as a parameter and returns a JSON object containing the engagement data.

Architecture and Features

The project is organized using the Model-View-Controller (MVC) pattern. In the application package,the views module contains the controllers for handling requests and rendering templates. The authentication module contains the controllers for handling user authentication and authorization. The models module contains the classes for the database tables. The templates folder contains the HTML templates for the views.The application includes features such as user registration and login, posting, commenting, liking, and editing or deleting post. The user also has the ability to change their username and password. Additionally, users can delete their comments as well. The application also includes an API for retrieving user engagement data and a feature for following and unfollowing other users.

Video

- https://drive.google.com/file/d/1afWeaaw4J1JTw7k_Msul5ADwTTVKHI_G/view?usp=sharing