Blog-Lite Report

Author:

- **Name**: Sherry Shajimon P T
- **Roll No**: 21f3001449
- **Email**: 21f3001449@student.onlinedegree.iitm.ac.in
- **About Me**: I graduated with a Bachelor of Arts degree in Philosophy from Christ University in 2021 and was also the top of my class. Currently, I am pursuing Bachelor of Science in Data Science from IIT-Madras and am in the diploma level.

Description:

Blog-Lite is a multi-user app where users can post blog-entries (Posts that reflect and discuss opinions) with appropriate images to spread their ideas in the form of text. The app has features to follow and search other users, and also voice opinions through comments on their posts. A Rest API has also been developed, following the OpenAPI Specifications, which can be used to access and modify the database for the app.

Technologies Used:

- **Python**: Develop the controllers and serve as the host programming language for the app
- **HTML**: Develop the required web-pages
- **CSS**: Style the web-pages
- **Bootstrap**: To make the frontend appealing and easy to navigate
- **SQLite**: Serves as the database for the app
- **Flask**: Serves as the web-framework for the app
    - **Flask-Restful**: Used to develop the RESTful API for the app
    - **Flask-SQLAlchemy**: Used to access and modify the app's SQLite database
    - **Flask-CORS**: Used to enable CORS for the app
- **Swagger OpenAPI**: Used to create the documentation for the API developed for the app
- **Matplotlib**: To create the charts to view the clicks on a post.
- **CSV**: To create the CSV files for the user about their posts
- **Secrets**: To create the token for a user
- **Git**: Version Control

Database Schema:

The database has four tables and the schema is as follows:

User Table	Follow Table
● **Roll** (Integer): Primary Key, Auto Increment ● **Username** (String): Unique, Not Null ● **Password** (String): Unique, Not Null ● **Img** (String): Not Null, Default 0	● **Roll** (Integer): Primary Key, Auto Increment ● **Author** (String): Foreign Key (User.roll), Not Null ● **Img** (String): Not Null ● **Text** (String): Not Null ● **Date** (String): Not Null ● **Title** (String): Not Null

Comment Table	Post Table
● **Roll** (Integer): Primary Key, Auto Increment ● **Post** (Integer): Foreign Key (Post.roll), Not Null ● **Author** (String): Foreign Key (User.username), Not Null ● **Comment** (String): Not Null	● **Roll** (Integer): Primary Key, Auto Increment ● **Following** (Integer): Foreign Key (User.username), Not Null ● **Follower** (String): Foreign Key (User.username), Not Null

API Design:

The RESTful API was created using the Flask-Restful library for Python according to the OpenAPI Specifications. All the database tables have CRUD operations available through the API. The API uses tokens for authentication for certain requests that require them. The token for a user can only be obtained from the account page of the user that is signed-in. For more information, please refer to the openapi.yaml file.

Architecture and Features:

The application follows the standard MVC architecture. The View of the application is created using HTML, CSS, and Bootstrap. The Controller is created using Python and Flask. The Model is created using SQLite.

The features of the application are as follows:

- Signup and Login for users
- Ability to view user’s posts, followers, and follows
- Navigate and view other’s posts, followers, and follows
- Generate API tokens to use user specific requests
- See chart showing the clicks on a post
- Download the user’s posts and their data as a CSV file
- Ability to search, follow, and unfollow other users
- User specific feed according to the follows of the user
- Create, View, Edit, and Delete posts
- Create, View, Edit, and Delete user accounts
- Comment to express user’s opinions on posts
- RESTful API for the posts, users, comments, and follows available

Video:

For the video, click here!