Author

Sandeep Kumar
Roll number: 21f1000195
21f1000195@student.onlinedegree.iitm.ac.in

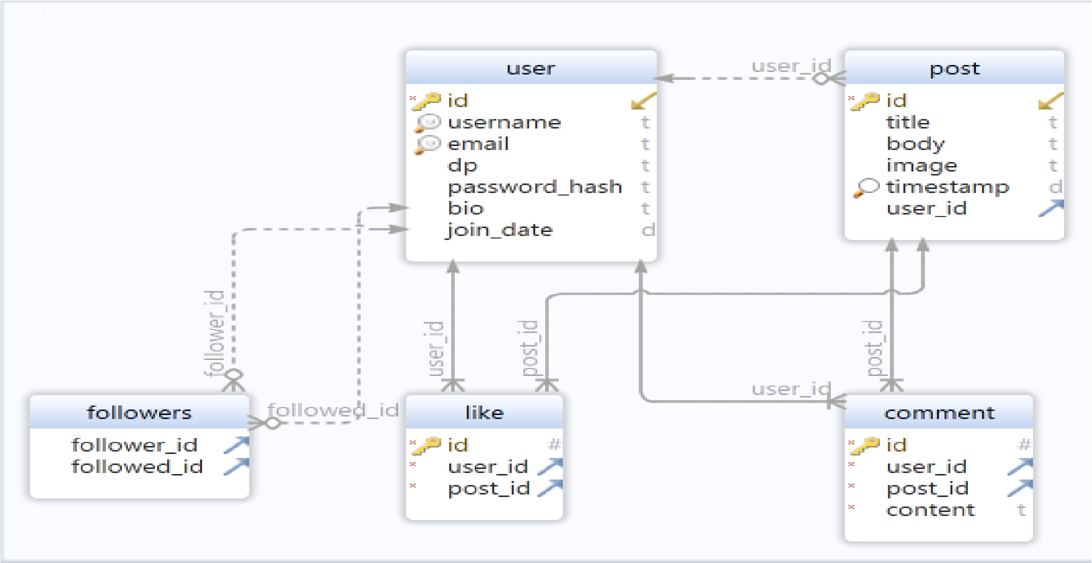
Description

It’s a lightweight multipurpose blogging app that allows users to create posts, like and comment on posts, follow other users, and search for other users. And all these have to be implemented using Flask, jinja, Bootstrap and Sqlite.

Technologies used

Flask: used this as the web framework for building this application.
Flask-Login: used it for session management. It also helps in handling login, and logout.
Flask-RESTful: used it for creating RESTful APIs for the app.
Flask-Reuploaded: This library helped in handling file uploads.
Flask-SQLAlchemy: This library is used to add support for SQLAlchemy. It makes it easy to interact with databases in your application.
Flask-WTF: used it for handling forms in the app.
Jinja2: used it as the template engine of the app.
SQLAlchemy: It’s an ORM tool that helps to interact with SQLite databases in Python.
Bootstrap: CSS framework, helped in design and styles of the app.

DB Schema Design



Blue arrow represents foreign key, t means varchar, d means datetime and # means integer.

The followers table is a many-to-many relationship table between users. It stores the follower_id and followed_id of the relationship.

The User table has several columns such as id, username, email, password_hash, bio and join_date. It also has relationships with other tables such as Post, Like, Comment and User (self-referential)

The Post class represents a post made by a user. It has a one-to-many relationship with Like and Comment and has a foreign key to the user who made the post.

The Like class represents a like on a post made by a user. It has a foreign key to the user who made the like and to the post that is liked.

The Comment class represents a comment on a post made by a user. It has a foreign key to the user who made the comment and to the post that is commented on.

API Design

Project has an API for User and Post. Through this API, CRUD operations can be performed on both user and post data. The API was implemented using the Flask-RESTful extension. It’s not integrated with the app, it is independent from the main app logic. Each resource (user and post) has its own endpoint, and the API supports standard HTTP methods such as GET, POST, PUT, and DELETE for performing the CRUD operations.

Architecture and Features

The project is organized with the app.py file serving as the main entry point for the application. This file contains all the routing information, controllers and handle the different URLs that the app can handle. The forms.py file contains the form classes used for handling user inputs. The models.py file contains the data models for the application. Templates are stored in a separate "templates" directory and are rendered using the Jinja2 template engine.

The application allows users to create new posts, like or comment on other users' posts, follow other users, and view profiles. Users can also search for other users through the search bar. The profile of each user displays stats such as the number of followers, following, and number of posts. The navigation bar includes an explore link, which allows users to view the latest news (implemented through newsapi.org) and popular posts. Users can also perform CRUD operations on both posts and users. The explorelink and the ability to like and comment on posts are additional features that have been implemented.

Video

presentation video