

Project Report: BlogLook

Name: Aditi Krishana

Roll: 21f1004270

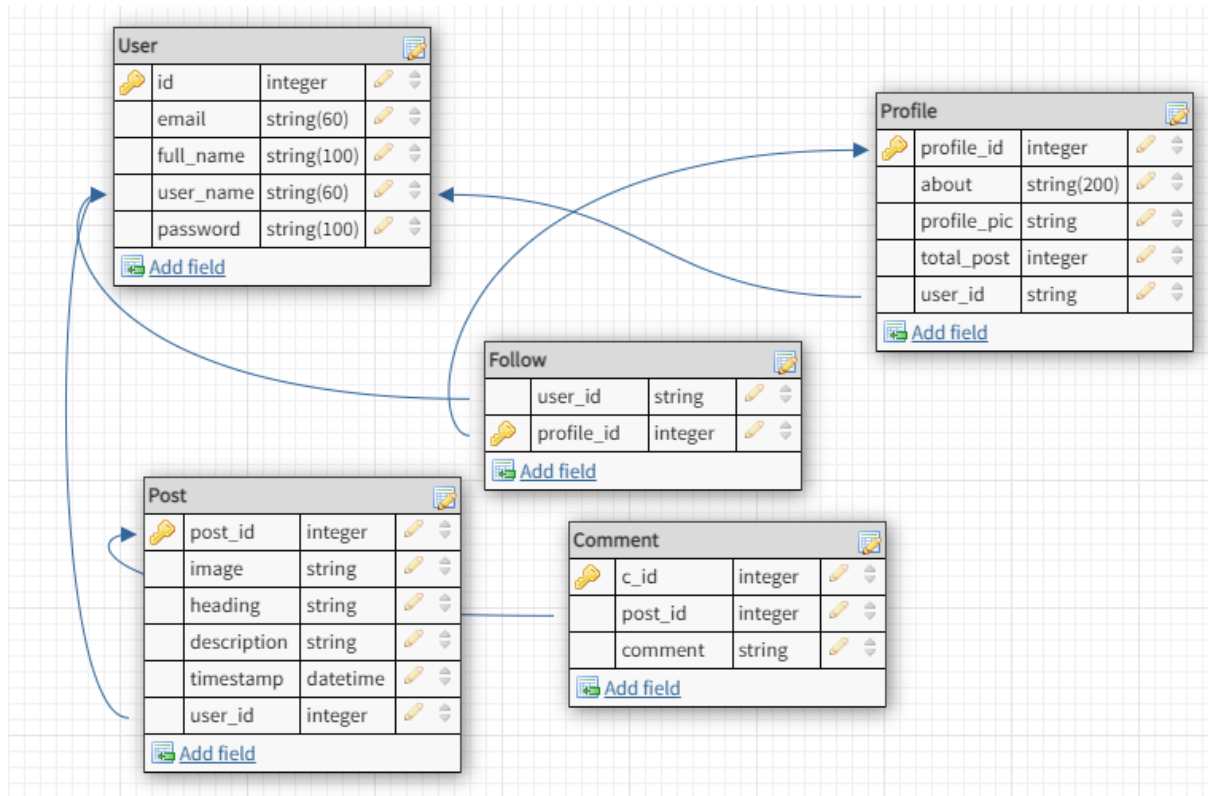
Email: 21f1004270@student.onlinedegree.iitm.ac.in

I am Aditi, a Postgraduate student of Chemistry, trying to gain the basics of data science.

Description

I have used the Flask framework to create the BlogLook app, in which different routes for different purposes and renders different templates, developed using HTML and styling fulfilled using CSS as well as Bootstrap, based on the user's click. It presents the engagement graph of the user in posting the blogs, generated using the matplotlib library of python. All different rendering is based on the Jinja2 library of python.

DB Schema Design



It is a web-app-based project for public use in which multiple users can log in to create their own profile and add their own posts, follow and unfollow other users, and add comments on each post. The project requirement requires login for the registered users which is aimed using the signup page, user will register and then login into their account. The sign-up page takes the user_name and email to be unique, the id is the primary key and the full name and password is taken in the string format from the User table. After the user enters their account, the sidebar can go to the profile page and edit their profile, user_id is the foreign key and profile_id is the primary key, total_post is an integer, about and profile_pic is taken as the string in the Profile table. The third option in the sidebar, add_blogs, allows the user to post the blog, in which post_id is taken as the primary key and user_id as the foreign key, image, heading, and description taken in string format in the Post table. The search option enables

the user to find other usernames and follow or unfollow them, where user_id and profile_id are taken as the foreign key in string and integer format respectively. Users can add comments on the post of the other user, in whichc_id is the primary key and post_id is the foreign key and the comment is taken in string format.

Technologies used

- **os**: For giving the path of the database and folders containing images
- **flask**: **Flask**: For creating the application
- **render_template**: For rendering the HTML pages on different clicks
- **request**: To get data and methods from forms
- **flask_sqlalchemy**- **SQLAlchemy**: For creating models in the database for the app
- **HTML**: For creating web pages
- **CSS**: To add styling effects
- **Bootstrap**: To align web pages as required and create selections
- **matplotlib**: For creating graphs
- **DateTime**: To use dates as data

Architecture and Features

The code for the BlogLook app is organised based on the usage and update required. The code for the app can be viewed inside the project folder with the name app.py python file. The app named BlogLook, on starting gives a home page where we have a link for the signup and login, after signing up it redirects to the login page which requires the correct combination of user_name and password to log in, which further takes us to the feed where the latest blogs of other users are been shown who is been followed. Further, then in the sidebar of the web page, the profile button redirects to the user's profile, where the stats of the user i.e. total no. of posts, followers and following is been shown as well as the edit profile button are made available for updating the profile image, about and the full name of the user, post engagement button is made available for showing the graph of the post by the user, below the posts by the user is also shown which can be edited, deleted and comments can be added into it. The Add_Blog option in the sidebar gives the user the ability to post the blog with a heading, description and image inside it. The Search button enables the user to search other users with their user_name, in which the option of follow and unfollow is provided. In the sidebar, the following and followers have a link which redirects to show the no. of followers and following of the particular user. Deleting the user's account option, deleting the user's account and redirecting it to the home page, and the Logout option just redirects the user to the login page. The user has the ability to delete the post and account using the buttons available which on clicking opens up the confirmation page for confirming the delete action. All the web pages which are rendered on different clicks are saved under the templates folder inside the BLOG - LITE folder. The images are saved inside the static folder. The python file models.py develops the database having 5 tables: User, Profile, Post, Comment, Follow tables for presenting data of use on screen. The app.py consists of all controllers used to access the model and modify the view of the app. The app is being served at <http://127.0.0.1:5000>

Video

[Project Video Link](#)