Author

Kshitij Roodkee
21f1001631
21f1001631@student.onlinedegree.iitm.ac.in

I am a web enthusiast and open source contirubutor who is excited to learn coding.

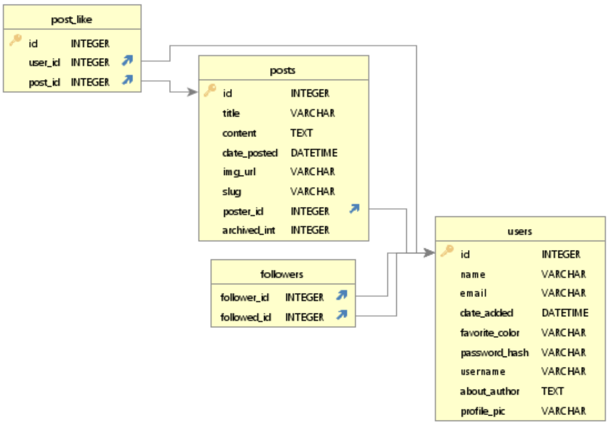
Description

This project gives us hand on experience on the working of flask and databases to create a web app the keypoints to learn was using queries , passing data in flask .

Technologies used

1. **Jinja2 :** It is a tempelating engine which gives us power to use python in html
2. **WTForms :** it is used to validate and build forms that are more secure and easy to use.
3. **Werkzeug :** It is used to enhance login security and save the user password as hashes
4. **SQLAlchemy :** It is used to work with queries on a sqlite database
5. **flask_ckeditor :** To give a rich editing experience while writing the blog
6. **Flask_login :** It is used to implement login and logout feature.
7. **Flask :** It is a web framework .

DB Schema Design



post_like	
id	INTEGER
user_id	INTEGER
post_id	INTEGER

posts	
id	INTEGER
title	VARCHAR
content	TEXT
date_posted	DATETIME
img_url	VARCHAR
slug	VARCHAR
poster_id	INTEGER
archived_int	INTEGER

followers	
follower_id	INTEGER
followed_id	INTEGER

users	
id	INTEGER
name	VARCHAR
email	VARCHAR
date_added	DATETIME
favorite_color	VARCHAR
password_hash	VARCHAR
username	VARCHAR
about_author	TEXT
profile_pic	VARCHAR

Architecture and Features

The project architecture could be divided into

1. **Templates** : It contains all the views/html files
2. **Static Folder** : It contains Css and images folder (user profile picture)
3. **Routes :** Their are mainly
    i. "/posts" route: which shows all the post available in the database

    ii. "/feed/<username>" route which shows specific post depending upon if user follows that author or not
    iii. "/add/posts" route: for adding blog post
    iv. "/user/<username>" route: shows user profile and stats
    v. "/follower/<username>" route: shows user followers list
    vi. "/following/<username>" route : shows user following list
    vii. "/user/dashboard" route: shows user setting can be used to update user profile setting here
    viii. "/posts/edit/<id>" route: for editing posts
    ix. "/posts/delete/<id> route: for removing posts

4. **Classes** :Their are three classes or model implemented
    i. Posts : to maintain Posts
    ii. Users : to maintain Users
    iii. LikePost : to maintain likes on the post
5. **Forms :** The forms are main feature to get the user input: The different forms implemented are:
    i. LoginForm : To be used for login
    ii. SearchForm : To implement Search Functionality
    iii. FollowForm: To submit follow request

Features Implemented :

1. **Users Customised Feed :** A user is able to create his/her own feed by determining whom to follow . This feature is implemented using by creating a relationship between users Many to Many and querying on this relationship.
2. **Likes :** A user can determine the credibility of a post bhy seeing the likes . This feature is implemented by creating a db model that keeps in count the no. of likes on that post
3. **Archive :** A user can archive a post or unarchive a post depending upon his/her needs. This feature is implemented using a check of 0 and 1 count associated with the post where 1 means archived and 0 means unarchived.
4. **Login and Logout:** Access to pages are maintained . A user can only see global posts if he/she is not logged in . Rest all pages need to be logged in . It is implemented using flask_login module.
5. **Search :** A user is able to search another user and also able to search post content. This is implemented using simple query on db
6. **Follow and Unfollow:** A user is able to follow unfollow another user . This is implemented using function call and a relationship between users .

Video

https://drive.google.com/drive/folders/1CfNK3qzkHpdFWBgI8ct3AqXIebDG8gl6?usp=sharing