Author

Aniket Dash
21f1006102
21f1006102@student.onlinedegree.iitm.ac.in

Description

The project required to develop a multi-user blog where users can create accounts, update their account information which includes changing of the username and the profile picture and delete their accounts and all the information linked to it.
It also required the functionality for a user to create, update and delete posts and the functionality for a user to follow another user.

Technologies Used

For server side operations Flask has been used. The extensions of Flask used here are:

- Flask-SQLAlchemy - Uses SQLAlchemy in providing users with Object-relational mapping (ORM)
- Flask-WTF & WTForms-Components - Enables and eases out the process of designing forms in the application.
- Flask-Login - Provides user session management for the application. It handles the common tasks of logging in, logging out, and remembering users' sessions over extended periods of time.
- Flask-Bcrypt - Provides hashing utilities for the application.

Database Schema Design

The database consists of 5 tables, which are as follows:

- User - Stores user information with the following columns.
    - id - Integer, primary key
    - username - String, unique, not nullable
    - password - String, not nullable
    - propic - String, not nullable, with a default value
    - comments - Relationship with the 'Comment' table with backreference to 'User' with the keyword 'author'
    - likes - Relationship with the 'Like' table with backreference to 'User' with the keyword 'author'
    - followed - Relationship with the 'followers' table
        - In addition to the columns or objects of the class 'User', we've got 4 functions as follows:
            - follow(user) - adds the mapping to the 'followers' table
            - unfollow(user) - removes the mapping from the 'followers' table

- is_following(user) - checks if a user is following the passed-in user.
- Followed_posts - gives us a union feed of the current user and the accounts followed by the current user.

- Post - Stores information about posts created by users.
    - Id - Integer, primary key
    - Title - String, not nullable
    - Desc - Text, not nullable
    - imageUrl - String, nullable
    - timeStamp - Datetime, with a default value of current time
    - User_id - Foreign Key mapping to id column of the ‘User’ table
    - Comments - Relationship with the ‘Comment’ table with a backreference to the ‘Post’ table with the keyword ‘post’
    - Likes - Relationship with the ‘Like’ table with a backreference to the ‘Post’ table with a keyword ‘post
- Comment - Stores information about comments on a post by a user
    - Id - Integer, primary key
    - Text - Text, not nullable
    - commentTime - Datetime, with a default value of current time
    - User_id - Foreign Key mapping to the id column of the ‘User’ table
    - post _id - Foreign Key mapping to the id column of the ‘Post’ table
- Like - Stores information about Likes on a post by a user
    - Id - Integer, primary key
    - User_id - Foreign Key mapping to the id column of the ‘User’ table
    - Post_id - Foreign Key mapping to the id column of the ‘Post’ table
- followers - stores mappings of who follows whom.
    - follower_id - Foreign Key mapping to the id column of the ‘User’ table
    - followed_id - Foreign Key mapping to the id column of the ‘User’ table

This design seemed like a suitable model to hold the information for the smooth functioning of the application.
The images related to a user like the profile picture or related to the post are stored with a different name (random hex generated while saving the images with a function) in a folder called ‘images’ and ‘profilePics’ inside the ‘static’ folder. These names are stored in the sqlite database and are accessed from the ‘static’ folder.

The passwords are hashed using a hashing function and brcypt.

Architecture and Features

Main.py - This is the file that runs the app. It lies in the root directory.

/application

This directory consists of the following files and folders:

- App.db - the sqlite database of the application
- Controllers.py - holds all the controllers necessary
- Forms.py - holds all the forms. Forms include
    - registration form,
    - login form,
    - Post form for posting to the user feed,
    - updateAccountForm for updating account information of the current logged in user,
    - FollowForm for following/unfollowing another user. It is just a submit button
- Models.py - holds all the Flask-SQLAlchemy models
- __init__.py - all the controllers had to be imported to this file. The Flask application object had to be in this file.

/application/templates

- Base.html
- Formbase.html
- Login.html
- Register.html
- Make_post.html
- Post.html
- Account.html
- Profile.html
- home.html

This directory holds all the templates required to render the whole application. Most of the templates inherit from the base.html file or the formbase.html file

/application/static

Contains css files of the corresponding templates

- Account.css
- Home.css
- Main.css
- profile.css

/application/static/images & /application/static/profilePics

Contain all the images that are saved with the help of a function called savingPicture(image) and savingProPicture(image)

This application uses jinja templating as the default feature.
The app allows Create, Update, Read and Delete operations on a user account, and on posts.
In addition to CRUD on posts, a user can comment and like on a given post as a form of post engagement.
User's activity is tracked based on the number of posts posted by the user.
Users have the ability to follow other people and view their feed.
Users can delete the comments only if they are the author of the comment or the post on which the comment is posted.
When a user follows another user, the follower's feed is updated to contain both the users' posts.
Deleting a post or a user requires the current user to login and confirm a modal before deleting.
Deleting a user deletes all the posts, likes and the comments generated by the user.
Feed is color coded. User's posts are displayed without a red border. Followed user posts are displayed with a red border.

Video

https://drive.google.com/file/d/14UoYQDdb_My5_c-sjB1AMxyhbzjxSJ3T/view?usp=sharing