Project Report : MAD-2 Bloglite

Author:

Name : Katta Midhil
Roll Number : 21f1006129
Email : 21f1006129@ds.study.iitb.ac.in

Brief Introduction : I am a full-time student at IITM BS Degree program, this is like a golden opportunity for me to pursue and finish this degree. I hope this opens up many fruitful opputunies for me.

Description:

The Blog lite application would be a platform where users can register themselves and registered users can make a blog post to share it on the platform and in turn they are able to follow other users to view their posts in ones feed and can also search for these users, and also able to update, delete and download their own posts.

Technologies used:

flask : Flask, redirect, url_for, render_template, request, make_response
flask_login : UserMixin, login_user, LoginManager, login_required, logout_user, current_user
jinja2: Template
flask_sqlalchemy : SQLAlchemy
email.mime.text :MIMEText
email.mime.multipart :MIMEMultipart
datetime : datetime
smtplib

Flask library was used to render HTML pages, redirect to Urls ,make requests from running server and send response from api functions.

Flask_sqlalchemy was used to generate the database class elements, make queries from the database, add to the database update the database and also delete entries from the database.

Flask_restful was used to construct the UserApis and PostApis. These apis perform CRUD operations on the users and post.

Flask_login was used to handle the user login and authentication for home page access.

Miscellaneous libraries above were used to aid the application.

DB Schema Design:

Database Scheme:

Tablename: 'user'

Columns: user_id(Integer,primary key), username(String, unique),password(String)

Tablename: ‘post’

Columns: post_id(integer,primary key), username(String), title(String), caption(String), imageurl(String), likes(Integer)

Tablename: ‘friendship’

Columns: friendship_id(integer,primary key), host_id(integer), friend_id(integer)

Tablename: ‘likes’

Columns: likes_id(integer,primary key), post_id(integer), friend_id(integer)

Tablename: ‘lastvisit

Columns: visit_id(integer,primary key), user_id(integer), timestamp(String)

User table’s primary objective was to serve as an authentication method during login and also during several functions where queries are made to the table for retrieving information and checking legitimacy of the user request.

Post table’s primary objective was to store the posts and the username column is added to link each post to its user and queries are made to the table for retrieving the posts for updation, deletion and display.

Friendship table’s primary objective is to store relations among users and queries are made to retrieve information on a user’s following and followers. Also used to generate information for the users feed of posts.

Likes table is used to store the data of post likes, Lastvisit data is used to store users last visit timestamp used in the logic for sending daily remainder mail

API Design:

PostAPI, Search API, FollowAPI, InteractAPI were used. All the Apis are used to do a fetch request in the javascript files for retrieving, modifying the data as and when required.

Architecture and Features:

All the code is in a single file names “main.py”, all the html files are in folder named “templates”,all the javascript files are in “Static” folder in “Js” folder, database is in the file “blogbase2.sqlite3”.

The main features were user login, user signup, home page display, profile page display, friend profile display, followers display, following display, search for users, add post , update post, delete post, like post,follow user, unfollow user, download posts, user logout.

All the User interaction is done with Vue js running in the Html files and python app routing is used to switch pages. Flask_login is used to authenticate the user. SQLAlchemy was used to query the information, processed to get the required result and rendered with the respective html page via javascript. Flask restful was used to create the Api for required Fetch call operations on user ,post objects in the database

Video link:

https://drive.google.com/file/d/1KAOXLft_kvewd3QJ-XW9A_M9bgK5g-pH/view?usp=sharing