Author

Name: Soumya V Namboodiripad
Roll Number: 21f1004752
Mail id: 21f1004752@student.onlinedegree.iitm.ac.in

I have completed MCA and have 8 years of experience in Mainframe Technology. Currently I am a housewife.

Description

BlogSpot is a multi-user app used for creating Blogs or posts. Users can post Blogs and follow other users. Users can see the posts/Blogs created by the users whom the follow. Users can like, dislike or comment to a post.

Technologies used

Flask – for developing application code
SQLAlchemy – for database connectivity
Flask_login – for login management
werkzeug.security – for password management
HTML – for designing web pages
Flask-restful – for API development
pandas – for exporting to a .csv file

DB Schema Design

The database includes 4 tables.
User details are given in User table. Posts table contains the details of the posts. User and Posts table are connected using the user_id. Followers table contains the user no and the following user no. both the columns together is the primary key, the same combination of user and following should not get repeated. Followers table is connected to User table using user_id. Comments table contains comment details for a post. Post id and the username who created the comment are also given. User and Posts are having one-to-many relationship. Posts and Comments are having One-to-many relationship. User and Followers have may-to-many relationships. Created an Index on username column for faster search.

1. User Table:
    a. Id (Primary Key) – Integer
    b. Username - String (Unique) (Index on username for Search)
    c. Email - String (Unique)
    d. Password - String
    e. Profile_image - String
2. Posts Table:
    a. Post_id (primary key) - Integer
    b. Post_title - String (Unique)
    c. Description - String
    d. Image_url - String
    e. Post_created_ts - DateTime
    f. Likes - Integer

g. Dislikes - Integer
h. Post_user (ForeignKey) - Integer (User.id)
i. Post_updated_ts - DateTime

3. Followers Table:
   a. User_no (PK) (FK) - Integer (User.id)
   b. Following_user (PK) (FK) - Integer (User.id)
4. Comments Table:
   a. Comment_id (PK) - Integer
   b. Comment_text - String
   c. Post_id (FK) - Integer (Posts.post_id)
   d. Username (FK) - Integer (User.username)

API Design

APIs for interaction with users and blogs – implemented using flask-restful Resource, Api

- CRUD on User
- CRUD on Posts
- APIs for getting the blogs/posts to show in feed
- API for adding / deleting comments

Project Architecture:

The ProjectBlog is the main folder for this project. The controller is blog_app.py and the API blog_app_api.py is inside ProjectBlog. All HTML files are present in templates folder inside ProjectBlog, Database file blog.db is present inside ProjectBlog folder. CSS styles are defined in styles.css inside static folder under ProjectBlog. All the images are also available in static folder. The ProjectBlog folder also contains the YAML file BlogAPI.yaml, the requirement.txt file containing the necessary python packages and a README.txt file explaining how to run the code. There is a folder 'exports' which will store the exported posts .csv files

Features:

BlogSpot is a multi-user app used for creating Blogs or posts. Users can post Blogs and follow other users. Home page allow Users to see the posts/Blogs created by the users whom the follow. Users can like, dislike or comment to a post. User can search for other users giving the name and follow them. There is a provision to unfollow also. User can create new post. Users can view, modify or delete the post created by them using Profile page. Profile page also displays number of posts created by the user, number of followers and number of following users. Also from profile page, user will have a facility to modify personal details and delete the user account. Once the user account is deleted, all the posts created by the user, all the comments added by the users will also get deleted.

Additional Features: Implemented proper login using flask_login. No end-point can be accessed directly without logging into the system. All end points are defined using "login_required" property. Passwords are encrypted before storing the database using werkzeug.security. Implemented the feature to export the post details of current user to a .csv file

Video

https://drive.google.com/file/d/1S6zSmgCUi3qBBWWRiAgE8KSwgrQDAvR2/view?usp=share_link