Author

Ashish Gaba
Roll Number – 21f1003118
Email ID – 21f1003118@ds.study.iitm.ac.in

Description

The project is a social media app that is used by users to create and share blogs. The users have the ability to search other users, see their profiles, and follow them to see their blogs on the feed. It provides support for Celery jobs, that can be triggered to be run in the background. It also supports caching with the help of redis-server.

Technologies Used

- Flask: To run the application on a server
- SQLite: For data storage
- Celery: To run triggered and scheduled jobs
- Redis: For in-memory caching of data
- Extensions:
  - JWT: To create, encode and decode JSON web tokens for authentication.
- Front-end technologies:
  - Jinja2
  - Bootstrap

DB Schema Design:

The following are the DB tables with columns and constraints. Each table has a unique identifier (id as primary key).

1. User
   Columns:
   - Username: Username used for logging. (Unique and Non-Nullable)
   - Password: Password used for logging (Non-Nullable)
   - Name: Display name of the user (Non-Nullable)
2. Follow: Each follow relationship is stored in this table.
   Columns:
   - Follower_id: User id of the follower. (Foreign Key, Non-Nullable)
   - Following_id: User id of the user being followed. (Integer, Foreign Key, Non-Nullable)
3. Blog: The user created blog post
   Columns:
   - Title: Title of the blog. (Non-Nullable)
   - Description: The description of the blog post. (Non-Nullable)
   - Creator_user_id: User id of the person who created the blog (Foreign Key, Non-Nullable)
   - Created_timestamp: timestamp of when the blog was created (Non-Nullable)
   - Last_updated_timestamp: timestamp of when the blog was last updated. (Non-Nullable)

API Design:

APIS have been for the following:

- CRUD operations for blog

- READ APIs on the following tables:
    - Blog – Used for showing feed
    - User – For searching for a particular user based on username
    - Follow – For showing the followers and following for a user

Architecture and Features:

Architecture:

The python code for the project is organized into 3 files –

- App.py: It contains the startup code and the logic for all APIs implemented
- Models.py: It defines all the tables that need to be created for the project.
- Celery_worker.py: It configures the celery worker for running background jobs

The static module contains 2 sub-modules:

- Js: It contains the JS files containing routes and different webpages.
- Uploads: contains the media files (blog images and profile pictures)

Features implemented:

1. User registration and login, through JWT token based authentication.
2. Creation, updating and deletion of blogs.
3. Users have the ability to edit their own profile and add profile images. They can also see the number of blogs they have posted, list of blogs posted, and count and list of people following them and who they are following.
4. Users can search for other users, see their profiles and follow/unfollow them.
5. Chronologically sorted blog feed with the option to navigate to the blog writer's profile.
6. Export job (using Celery) to export users blog history data to a CSV file.
7. Monthly report of blogs created sent to users via celery.
8. Daily email reminder to user if they haven't posted in a day, via celery.
9. Caching of data like feed and user data for better performance.

Video Link:

https://drive.google.com/drive/folders/1bNakWPuyziPd1ViJ-lhOjjttreVjRJo2?usp=sharing