Author

Indranil Patra
22f1001196
22f1001196@student.onlinedegree.iitm.ac.in
I am a Diploma level BS student from Kolkata.

Description

This is the version 2 of the Blog Lite Application. It is a multi-user social application for uploading blogs. The application supports engagement with the blogs in the form of likes and comments and every user can follow other users.

Technologies used

Module (Backend)	Purpose
flask	Provides the required configurations with Flask object, routing, request handling functions
flask_sqlalchemy	Handles the database connection provides functions to perform operations on the database
flask_restful	Building the REST API
flask_jwt_extended	Authorization for the API using JSON Web Tokens
redis	Python interface to the Redis key-value store
celery	Run the tasks asynchronously using workers and task queue.
flask _sse	Flask extension to support Server Side Events using Redis
gunicorn	Python HTTP server for UNIX, to implement Server Side Event Handling
flask_caching	To add caching support to the backend
flask_cors	To handle and make cross origin requests possible
boto3	AWS SDK for Python, interface for AWS S3 storage for storing images
pillow	Image processing such as scaling down and cropping
flask_bcrypt	Hashing passwords
werkzeug.utils	Provides the secure_filename function used while storing images as files

Module (Frontend)	Purpose
Vue	For building the user interface
VueX	For state management.
Vue-Router	For routing, ease of user navigation without refreshing pages.
npm	As package manager for the frontend

DB Schema Design

User Table Schema

Column Name	Column Type	Constraints
user_id	Integer	Primary Key, Auto Increment
username	String	Unique, Not Null
password	String	Not Null
name	String	Not Null
email	String	
role	String	Not Null
pic	String	
bio	String	

Post Table Schema:

Column Name	Column Type	Constraints
post_id	Integer	Primary Key, Auto Increment
author	Integer	Foreign Key (user.user_id), Not Null
title	String	Not Null
content	String	Not Null
img	String	Not Null
created_on	DateTime	Not Null
archived	Boolean	Default (False)

Follow Table Schema:

Column Name	Column Type	Constraints
follower	Integer	Primary Key, Foreign Key(user.user_id)
following	Integer	Primary Key, Foreign Key(user.user_id)

Like Table Schema:

Column Name	Column Type	Constraints
post_id	Integer	Primary Key, Foreign Key(post.post_id)
liked_by	Integer	Primary Key, Foreign Key(user.user_id)

Comment Table Schema

Column Name	Column Type	Constraints
comment_id	Integer	Primary Key, Auto Increment
post_id	Integer	Primary Key, Foreign Key(post.post_id)
comment_by	Integer	Primary Key, Foreign Key(user.user_id)
comment	String	Not Null

Notification Table Schema:

Column Name	Column Type	Constraints
notification_id	Integer	Primary Key, Auto Increment

to_notify	Integer	Primary Key, Foreign Key(user.user_id)
timestamp	DateTime	Not Null
type	Number	Not Null
id	String	
username	String	
seen	Boolean	Default(False)

API Design

Elements the API has been created for: SignUp, AccessToken, User, Post, Like, Comment, Notifications
SignUp: For new user registration.
AccessToken: For providing access token required for authentication for CUD operations.
User: CRUD operations on user
Post: CRUD operations on post
Like: Read likes of a post or like action
Comment: Read comments of a post or post a comment

The API was implemented using the flask_restful package where every API inherits the properties of the Resource class provided by the flask_restful package. The authentication for the API is done using flask_jwt_extended, which is a token based authentication mechanism.

Architecture and Features

The root folder is divided into Frontend and Backend. The frontend consists of the required user interfaces whereas the backend consists of the required python files to serve requests from the frontend. Here the frontend sends http requests using Fetch to the backend API, which returns json formatted responses.

Features implemented:
User Signup and Login using flask_login package
User Profile view with follower and following count and users, count and list of posts
Blog Post Management: Create, Edit and Delete blogs
Search: Search for users, Follow and Unfollow users
User's Feed: Displayed in order of timestamp
Engagement on Blogs: Ability to like, comment on the blog
Notifications: Notifications for engagement with your blog, or follow operation
Daily reminders on Google Chat using webhook
Monthly Engagement Report
Export all the blogs as CSV which is notified to the user using SSE and sent to the email
Caching to improve performance with cache expiry.

Video

https://drive.google.com/file/d/1asV_8MIol4H2lJ6_DMq0h1F6azpF-Rct/view?usp=sharing