

Author

Indranil Patra

22f1001196

22f1001196@student.onlinedegree.iitm.ac.in

I am a Diploma level BS student from Kolkata.

Description

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	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_login	User session management - logging in, logging out, remembering sessions
flask_wtf	Designing forms required in the application
wtforms	Provides various input fields and the validators for the form
flask_restful	Building the REST API
flask_jwt_extended	Authorization for the API using JSON Web Tokens
datetime	Handling the timestamp of the posts
flask_bcrypt	Hashing passwords
werkzeug.utils	Provides the secure_filename function used while storing images as files

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

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
content	String	Not Null
action	String	
seen	Boolean	Default (False)

API Design

Elements the API has been created for: SignUp, AccessToken, User, Post

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

The API was designed using the flask_restful package where every API inherits the properties of the Resource class provided by the flask_restful package.

Architecture and Features

The root folder contains the main.py which consists of all the routes, __init_app.py which initializes the Flask application, models.py which contains the database tables' designs, auth.py contains the SignUp, Login and Edit Profile form designs along with user verifications, api.py contains the elements of the API, config.py contains the application configuration by the developer, run.sh is the bash script for running the application. The styles, scripts and images are present in the static folder, HTML templates in the templates folder, database in the instance folder

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

APIs: CRUD on users, CRUD on blogs

Validation: All form validations using flask_wtforms

Engagement on Blogs: Ability to like, comment on the blog

Archive: Ability to archive(hide) a blog

Notifications: Notifications for engagement with your blog, or follow operation

Video

https://drive.google.com/file/d/1U5aTZEW-lygEfMOMt33_BLMgx2vpi8Y/view?usp=sharing