Bloglite App 2.0

Project for Modern Application 2, January 2023

AJEET KUMAR

21f1006807@ds.study.iitm.ac.in

Description

Bloglite is a blogging application that allows users to post, edit, save, view, like and comment blog posts. It is a free platform with its unique set of features and capabilities. Bloglite is for anyone looking to start a blog and share their thoughts and ideas with the world. The project at hand entails the development of a web-based Bloglite application utilizing Python Flask as the backend framework and Vue.js as the frontend framework. The uniqueness of this project is that Bloglite is a progressive web application. It utilizes VitePWA for this purpose.

Technologies Used

Backend

- Python-flask, SQLAlchemy, flask-cors, flask-jwt-extended, celery, redis, smtplib

Frontend

- Vue 3, Vite, Pinia, VitePWA Vue-router, dayJS, validateJS, ApexCharts, Bootstrap

Backend Design

Database

Database Schema

The database contains 5 models: user, relations, comments, bloglikes and blog. The number of attributes in these models are 8, 3, 5, 5 and 9 respectively. They are userId, userName, password etc. Blogs and relations are linked to the user directly. Bloglike and comments are directly connected to the blog.

API

The app API is created with a flask blueprint named “api”. The API has 12 parts - userAPI, authenticationAPI, feedsAPI, exportAPI, imageAPI, likes_dislikesAPI, profileAPI, relationsAPI, searchAPI, summaryAPI, commentsAPI and blogCreateAPI. userAPI, relationsAPI, and commentsAPI are used for standard CRUD operations on respective model. authenticationAPI deals with user authentication and token generation, while exportAPI is used for exporting requested content in CSV. OpenAPI specification is attached for deeper outlook.

Asynchronous and Scheduled Jobs

The app uses celery and Redis for asynchronous and scheduled jobs. Exporting blogs is an asynchronous job, and daily/monthly reminder is a scheduled job.

Frontend Design

Frontend for Bloglite app is created with standard Vue3 build tool “Vite”. It can be broken down further in following parts:

- ### State Management

  The frontend has a centralized state manager “counter” created with Pinia store. This store contains user data fetched from the API and access token which are stored in browser cookies.

- ### Router and Views

  The router is created with vue-router. The router is used to deliver the app as an SPA. The router has 5 ‘views’ inside it - LoginView, BlogView, ProfileView, RegisterView and SummaryView which render login page, feeds, home page, registration page and summary page respectively.

- ### Components

  The app has different Vue components for different parts of the app – Blog, Create Blog, Delete User Confirmation, Search User, Update Password and Update Blog.

Architecture and Features

The Bloglite app comprises two fundamental components: Home and Profile View. Each user has his/her own dashboard which shows other users’ feeds and a profile section to view his/her own older posts with a functionality to edit, delete, and export posts. A registered user can also update the password and delete the account permanently (if required). The user can only see the blog posts in his/her feed of those users to whom he/she is following. User has two options (PDF and HTML) to get the monthly report summarizing the number of blogs, no. of followers, likes dislikes, comments on each blog, etc. The summary page graphically describes the likes, dislikes, comments on each blog post. User is also allowed to search any other user with username to view the profile, follow and unfollow.

Please note that this project is composed of two separate folders, the "frontend" and "backend" folders, each containing a corresponding "readme.md" file that provides detailed information on the contents and functionality of that specific folder. It is recommended to review "readme.md" file to fully understand and run the project.

[Presentation Video]

*****