

Blog Lite Application V2 Documentation

Author

Sai Vineeth Karampuri

21f1002775

21f1002775@student.onlinedegree.iitm.ac.in

I'm a 4th year CSE student from KU College of Engineering & Technology. I am a Data Science enthusiast and looking forward to learn more in this IITM BS degree.

Description

Blog Lite is a multi-user app in which the user can upload blogs with the images and share them with their followers. User can search for other users and follow them based on their interest. The user can also post their comments on the post.

Technologies used

flask – for API.

flask-SQLAlchemy – for connecting the SQLite database as a server with the flask application.

flask_restful – to implement CRUD operations using flask API.

Flask-Security – for Token Based Authentication

VueJS – for frontend UI

Vue CLI – for advanced VueJS implementation

VueX – for state management

Bootstrap Vue – for CSS in VueJS

Redis and Celery – for user triggered and scheduled Jobs

Jinja2 templates – for celery scheduled jobs

DB Schema Design

User table schema		
Column Name	Column Type	Constraints
id	Integer	Primary Key, Unique, Not Null
fullname	String	Not Null
username	String	Not Null, Unique
password	String	Not Null
email	String	Not Null, Unique
mobile_number	Integer	Not Null, Unique
created_date	String	Not Null
updated_date	String	Not Null
active	Integer	Not Null
fs_uniquifier	String	Not Null

Roles table schema		
Column Name	Column Type	Constraints
id	Integer	Primary Key, Not Null, Unique
name	String	Not Null, Unique
description	String	--

Following_list table schema		
Column Name	Column Type	Constraints
id	Integer	Primary Key, Not Null, Unique
user_id	Integer	Foreign Key ("User. id")
following_id	Integer	Not Null

Followers_list table schema		
Column Name	Column Type	Constraints
id	Integer	Primary Key, Not Null, Unique
user_id	Integer	Foreign Key ("User. id")
following_id	Integer	Not Null

UserPosts table schema		
Column Name	Column Type	Constraints
post_id	Integer	Primary Key, Unique
user_id	Integer	Foreign Key ("User. id")
username	String	Not Null
post_img	Blob	Not Null
img_filename	String	Not Null
mimetype	String	Not Null
post_name	String	Not Null
post_caption	String	--
created_date	String	Not Null
updated_date	String	Not Null
isArchive	Integer	Not Null, Default 0

Comments table schema		
Column Name	Column Type	Constraints
comment_id	Integer	Primary Key, Unique
user_id	Integer	Foreign Key ("User. id")
post_id	Integer	Foreign Key ("UserPosts.post_id")
text	String	Not Null
created_date	String	Not Null

API Design

API was designed for interaction with users, blogs, comments & likes using CRUD operations.

User has GET (fetches all the details of a user), POST (creates a new user), DELETE (deletes the existing user), PUT (updates the existing user).

UserProfile has GET (fetches the profile data of user)

Connections has GET (fetches the list of following & followers of a user)

Feed has GET (fetches the feed of a user which is to be displayed in the home/feed page)

Blog has GET (fetches the details of blog), POST (creates a new blog), DELETE (deletes the existing blog), PUT (updates the existing blog).

Like has PUT (updates the like on the blog)

Comment has POST (adds a new comment on a blog), DELETE (deletes a comment on a blog)

ArchiveBlog has GET (fetches the archived posts of a user), PUT (update a blog to archive)

FollowUnfollowUser has POST (follow a new user), PUT (unfollow a user), DELETE (removes a user from followers)

Search has POST (search for a user)

ExportBlogs has GET (fetches the data which needs to be exported)

YAML file for the API documentation (attached in root directory).

Architecture and Features

In the root directory, there are folders as frontend and backend.

Backend folder contains application which contains (`__init__.py`, `api.py`, `cache.py`, `config.py`, `controllers.py`, `dataaccess.py`, `database.py`, `models.py`, `myfunctions.py`, `tasks.py`, `validation.py` & `workers.py`), `db_directory` which has `bloglitedb.sqlite3` database file, `static` which contains (bootstrap, images & CSS files), `templates` which contains all html files and other files such as `local_run.sh`, `local_setup.sh`, `local_beat.sh`, `local_workers.sh`, `main.py`, `readme.md` & `requirements.txt`.

Frontend folder contains `node_modules`, `public`, `src` that contains assets, components that we use for vuejs, plugins, router, store, views, `App.vue`, `main.js` and other js files.

In Blog Lite app, the new user can sign up and the existing user can login with his credentials and view the feed from the users the user is following. User can upload the blog with an image and caption and share it with other users, edit the blog & delete the blog. The profile page has basic stats about user. User can edit his profile or delete the profile from the application.

It has features like commenting on the blogs of the users we follow and liking the blogs. It shows the engagement on each post graphically. The user can also hide the post by archiving the post so that it is not visible in others feed. The user can search for a particular user and follow or unfollow the user or remove the user from followers. Export option gives the feature of exporting the data in csv format. Daily reminder if the user does not visit/upload the blog will be sent to the mail. Monthly reports will be generated and sent to the mail with a pdf attachment of report.

Video

https://drive.google.com/drive/folders/16wdHzo1LUzL_rJKyGyN3j8TOYPsf0lqX?usp=share_link