

Author

Name: Rohit Kumar

Roll No: 21F1001368

Email ID: 21f1001368@ds.study.iitm.ac.in

My name is Rohit Kumar. I am a student of Diploma level at Indian Institute of Technology, Madras. Apart from this, I have completed my graduation in Chemical Engineering from Dr S.S.B. University Institute of Chemical Engineering and Technology, Panjab University, Chandigarh. I have keen interest in Machine Learning, Deep Learning and Artificial Intelligence.

Description

Blog Lite is a platform for blogging where users can publish their own blogs and discover new bloggers. Users can follow and unfollow other bloggers, and multiple users can use the platform simultaneously. To encourage users to post regularly, the application sends daily reminders to those who haven't posted a blog in the past 24 hours. Additionally, monthly engagement reports are generated and sent to users every month. Users also have the option to export their blogs in a csv file and download them to their own system.

Technologies used

flask: Flask is a web framework which is used as backend server for this application

flask_security_tool: flask_security module is used to control user access to applications through proper login framework. In-built default Token based authentication mechanism is used to secure api calls.

flask_SQLAlchemy: flask_SQLAlchemy is a flask extension that uses SQLAlchemy to connect to the database server

flask_restful: flask_restful is extension of flask that helps to add support for APIs

flask_bcrypt: Flask-Bcrypt is a flask extension that provides hashing functionality to our applications

Vue.js: is a frontend framework built on javascript. Vue.js is used for frontend UI design.

redis: redis is an in-memory key-value pair database. In this application, redis is used for caching, web service broker to host async jobs using celery.

Celery: celery is used to handle user triggered async tasks, daily reminders, monthly reminders, sending mails etc.

Flask_caching: flask-caching module is used for caching purposes.

Weasyprint: Weasyprint library is used to generate pdf from html. This is used to generate monthly engagement reports.

DB Schema Design

Database used for this application has 9 tables - **user**, **blogs**, **blog_media**, **comments**, **comment_media**, **followers_list**, **likes**, **role**, **roles_users**. Each table is used to store information as their name suggests.

- user

id	Integer	PRIMARY KEY, AUTO-INCREMENT
username	varchar(255)	NOT NULL
email	varchar	NOT NULL
password	Varchar(255)	
active	Boolean	
fs_uniquifier	Varchar(255)	NOT NULL
fs_token_uniquifier	Varchar(255)	
created	Datetime	NOT NULL
updated	Datetime	

- blogs

blog_id	Integer	PRIMARY KEY, AUTO-INCREMENT
title	Varchar	NOT NULL,
description	Varchar	NOT NULL
user_id	Integer	NOT NULL ForeignKey references user(id)
created	Datetime	NOT NULL
updated	Datetime	

- blog_media

media_id	Integer	Primary Key, Auto Increment
blog_id	Integer	Not Null, ForeignKey references blogs(blog_id)
media_url	Varchar	Not Null

- comment_media

cmedia_id	Integer	Primary Key, AUTO INCREMENT
comment_id	Integer	NOT NULL ForeignKey references comments(comment_id)
cmedia_url	varchar	NOT NULL

- comments

comment_id	Integer	Primary Key, Auto-Increment
blog_id	Integer	Not Null, ForeignKey references blogs(blog_id)
user_id	Integer	NOT Null, ForeignKey references user(id)
comment	Varchar	
created	Datetime,	Not Null
updated	Datetime	

- followers_list

follower_id	Integer	Primary Key, Auto-Increment
user_id	Integer	Not Null, ForeignKey references user(id)
follower	Integer	NOT Null, ForeignKey references user(id)

- likes

likes_id	Integer	Primary Key, Auto-Increment
blog_id	Integer	Not Null, ForeignKey references blogs(blog_id)
user_id	Integer	NOT Null, ForeignKey references user(id)

- role

id	Integer	Primary Key, Auto-Increment
name	Varchar	Not Null
description	Varchar	

- role_users

user_id	Integer	ForeignKey references user(id)
role_id	Integer	ForeignKey references role(id)

API Design

API design is described in `api.yaml`. APIs are created for reading user information, retrieving authentication tokens and signing in to the application, creating new blog posts, editing blog posts and deleting posts. Also, APIs are created to handle export jobs, mailing jobs etc.

Architecture and Features

This project has a python file named "main.py" that launches the applications. Using flask_restfull, APIs are designed that performs various tasks pertaining to CRUD applications and other utility tasks such as sending emails, sending daily reminders and monthly engagement reports. Celery is used to create and execute asynchronous tasks. Redis plays an important role as a message broker and it is also used for caching.

This application behaves as a single page application. Vue.js framework is used to create this dynamic single page application. Vue.js receives the data from the database by making API calls to the server and updating in the frontend.

Flask-Caching is used to cache the various sections of views and data fetching at the backend.

Video

Video Link: <https://drive.google.com/drive/u/1/folders/1GDsQn4QTxvpfz47cBF126nF1Prw0hO4k>