

Author

Dhruv Sanan

21f1004012

21f1004102@student.onlinedegree.iitm.ac.in

My name is Dhruv Sanan and I live in Amritsar, Punjab. Apart from being an IIT Madras student, I am also a regular student at Punjab Technical University studying Btech CSE in the 5th semester. I am also fascinated with the working of my brain, so I also study neuroscience and psychology. And I am planning to combine these subjects with AI in my master's.

Description

I started with a lot of confidence because of MAD 1 project since I aced in that project, but this is my first project handling frontend, backend, celery worker, mailhog, redis and pwa, so I could work with limited functionalities, so i had to open myself up to new technologies. ScreenCasts, YouTubers and official documents helped me a lot in terms of debugging errors and adding specific functionalities. I started with creating a frontend for my mad 1 project, building core functionalities and working up to recommended ones. I even tried recommended ones as well. I kept Instagram, chat gpt and youtube as a model in my mind to implement functionalities.

Technologies used

Sqlite: for the database storage engine

Celery: to create backend jobs

Redis: for caching and backend jobs

Vue-cli: to create frontend

Vue-pws: to create PWA

Flask: for the creation of application code

Flask-Caching: for caching

Flask-Cors: to create Cross Connection

Flask-JWT-Extended: to create tokens

Sqlalchemy: for the query on the database

Vscode: as my code editor

Fpdf: to export data in pdf format

Json: to create a response in JSON format

Csv: to export data in CSV format

Os: to interact with files and folder

Bootstrap: for HTML generation and styling

Werkzeug.security: to generate and check password hash

Werkzeug.utils: to get a secure version of the filename

Flask_restful: for building REST APIs

Postman: to test the created APIs

DB Schema Design

View (**id** integer primary key, **post_id** integer Foreign Key(post.id), **author** integer Foreign Key(user.id) nullable=False, **date_created** DateTime default=func.now())

Follow (**id** integer primary key, **user_id** integer Foreign Key(user.id), **followed_user_id** integer ForeignKey(user.id), **date_created** DateTime default=func.now())

User (**id** integer primary key, **email** string, **url** string, **username** string, **password** string, **date_created** DateTime default=func.now(), **daily_hour** integer, **daily_minute** integer, **monthly_hour** integer, **monthly_minute** integer)

Post (**id** integer primary key, **title** string, **text** string, **url** string, **date_created** DateTime default=func.now(), **author** integer Foreign Key(user id) nullable=False)

Comment (**id** integer primary key, **text** string, **date_created** DateTime default=func.now(), **author** integer Foreign Key(user.id) nullable=False, **post_id** integer Foreign Key(user.id) nullable=False)

Like (**id** integer primary key, **date_created** DateTime default=func.now(), **author** integer Foreign Key(user.id) nullable=False, **post_id** integer Foreign Key(user.id) nullable=False)

The database is designed keeping in mind to implement functionalities like adding comments, likes, and views. Users should also be able to follow so the follow table was created. And the user should be able to post so the Post table is created with a foreign key of user id. The id of every table is kept as the primary key in order to uniquely identify each row.

API Design

I have created 8 paths(endpoints) for CRUD operations on User and Post (4 for each) and a path to get posts of the home feed page. I have used similar codes on API as I have used for controllers with necessary exceptions.

Architecture and Features

The code which just runs the application (app.py), a website folder, and an instance folder where the database is stored, are all present in the Blog Website folder. All the codes and templates are stored in the website folder. I have my vue project named Auth. Inside the Auth folder, there is a component folder where components that I have used are stored. All models, controllers, APIs, basically everything is in __init__.py file. I could not create separate .py files because, for some unknown reason, I couldn't import classes from a different file. Users can signup and log in, profile view with basic stats, Blog Post Management, Search and Follow / Unfollow Others, own Feed, CRUD operations on users, posts and comments, like/unlike, search posts, create async job to export data of posts in csv, see all posts of some user and their dashboard. CRUD operations in user and post using API, home page feed of a user using API, daily job for reminder, monthly job to send monthly report in pdf or html, caching to increase performance and option to create a number of posts using csv file.

Video

[Link](#)