Author

Name : Ashrey
Roll number : 21F2000448
Email id : 21f2000448@ds.study.iitm.ac.in
I love to do maths as well as programming, which data science provides me in the perfect proportion. I'm currently pursuing B.Sc. Mathematics (Hons).

Description

The main aim of this project is to create a blog application using Flask, HTML, SQLite, etc. CRUD on blogs and user profiles along with alert messages and the ability to follow/unfollow other users, proper login system for improved security has been implemented in the project. The user will get daily reminder in the email provided and a detailed monthly report of his/her status of blogs. A CSV file will be sent to the email, if the user wants to export the data on the application.

Technologies Used

Here's are the technologies I used in this project. Python, HTML/CSS/JS, Jinja, Bootstrap, Flask, Flask-Login, SQLite, Flask-SQLAlchemy, Flask-caching, Flask-restful, weasyprint, celery, Extensions (like: request, os).

- Python is the core programming language used.
- Flask is the main framework used for the Web-app.
- Flask-Login is used for managing multiple user login and keeping a session alive.
- Flask-SQLAlchemy is the SQL toolkit used to connect with the database file.
- Flask-caching for making cache and improving performance
- Flask-restful for creating REST architecture based API
- Weasyprint for creating PDF's from HTML
- Celery for back-end asynchronous jobs

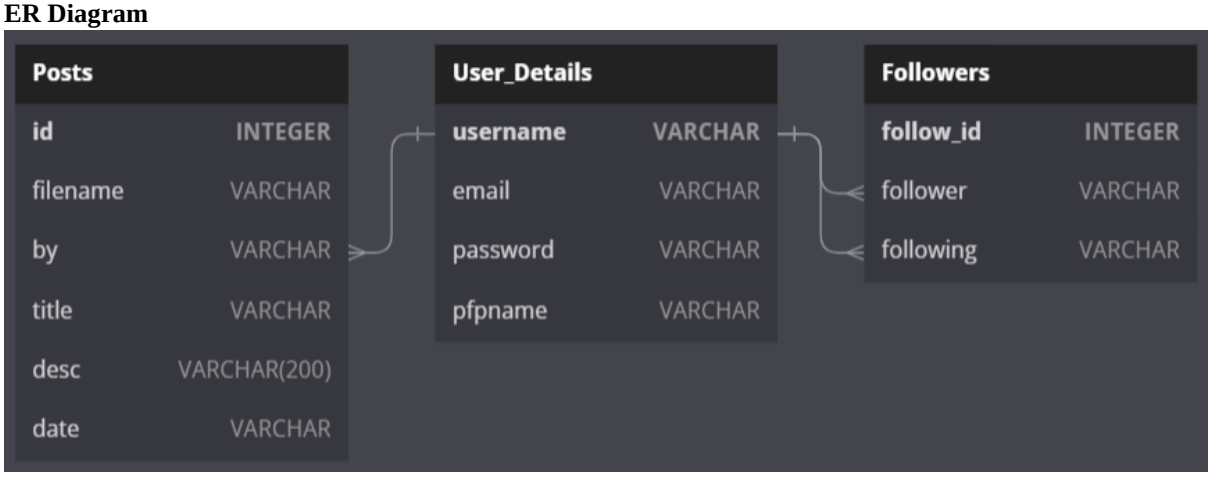
API Design

I have implemented Get, Put, Delete and Post for Users table as Users_api class, Posts table as Posts_api class, Follower table as Followers_api class. Username has been set as primary key in the Users table.

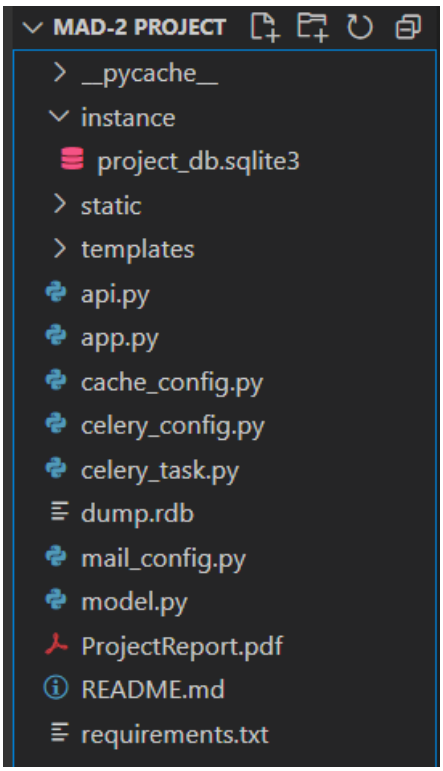
DB Design Schema

Table Name	Columns	Description	Constraints
User_Details	username	Username of the user	String, Unique, Primary_key
	email	Email ID of the user	String
	password	Password of the user	String
	pfpname	Name by which the user's profile pic will be stored	String, Unique
Posts	id	Unique id for each blog	Integer, Primary_key, Autocrement
	filename	Name by which each blog image will be stored	String, Unique
	by	Username of the user who created the post	String, Foreign_key
	title	Title of the post	String
	desc	Description of the post	String
	date	Date & time when the post was created	String
Followers	follow_id	Unique id for each follow/following	Integer, Primary_key, Autocrement
	follower	Current user who is following other users	String, Foreign_key
	following	User who is being followed by current user	String, Foreign_key

ER Diagram



Architecture



- Here, I have 2 folders:
  - static - which holds the JS, CSS and image files
  - templates - which holds all the HTML files.
- Then, I have made 7 python files:
  - api.py file has all the code related to APIs
  - app.py file has the code for the main code to start the Web App
  - models.py file has the all the code related to the different models.
  - cache_config.py, celery_config.py, mail_config.py files have the configurations for the respective parts
  - celery_task.py file has all the asynchronous tasks to be done by celery
- README.md has the instructions on how to start the Flask Web App.
- project_db.sqlite3 is the database file.
- Requirement.txt has the required packages name

Features

Here's a list of features:

- Multiple users can use the Web App at the same time.
- Interactive page which shows an alert when we delete a blog or user or if the password given does not match the re-typed password during new registration.
- CRUD operations on profile - Create, Read, Update, Delete.
- CRUD operations on blogs - Create, Read, Update, Delete.
- Daily reminder and monthly report will be sent to the user's email-id.
- Sending an email to the user's email-id containing a csv file that consists of the user's posts.

Video

Link to my Video:
https://drive.google.com/file/d/1KEXZmsjdIPGQNxeNpkyHURTXBvCMNz05/view?usp=share_link