Author

Bhavesh Choudhary
22f1000718
22f1000718@student.onlinedegree.iitm.ac.in

I am Full Stack developer knows MERN Stack and Flask. I am also learning Data Science from IITM online B.S Degree in Data Science and Applications (9.5 CGPA) with offline government college CTAE (3rd Year, 8.3 OGPA), Udaipur in CSE. I have good knowledge in Data Structure and Algorithms and knows C/C++, Java, Python, Flask, JavaScript.

Description

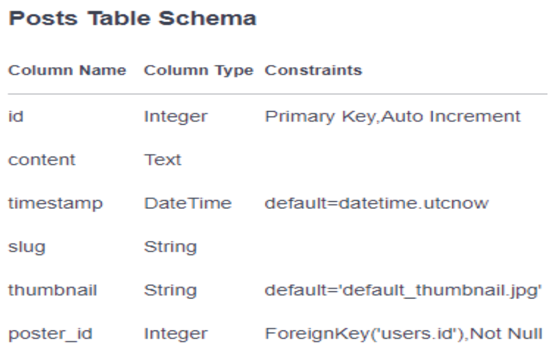
In BlogLite a user can create account then login and create Blog Post. Can connect others by following them. Users feed will have all of his posts with posts of users he follows ordered by timestamp and can like, comment them. Edit and Delete User account and posts details.

Technologies used

Sqlite DB, Flask, Flask-Login, Flask-RESTful, Flask-SQLAlchemy, Flask-WTF, Jinja2, Werkzeug, WTForms, email-validator and Jinja2 templates + Bootstrap for HTML generation and styling

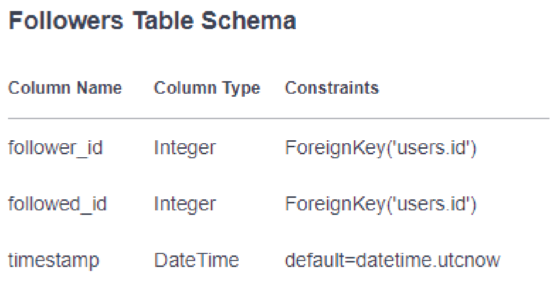
Flask is used to build web app based on python, Flask-Login for cookie based login support with Remember me feature, Flask-RESTful for API, Flask-SQLAlchemy for implementation Sqlite database models and Object oriented query support, Flask-WTF for advanced flask form protected by secret key, Werkzeug for securing filenames and creating and Storing Password as Hash to improve security and verifying email-validator to validate email.

DB Schema Design



Posts Table Schema

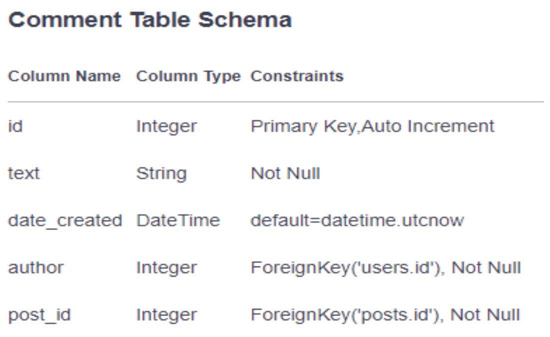
Column Name	Column Type	Constraints
id	Integer	Primary Key,Auto Increment
content	Text	
timestamp	DateTime	default=datetime.utcnow
slug	String	
thumbnail	String	default='default_thumbnail.jpg'
poster_id	Integer	ForeignKey('users.id'),Not Null



Followers Table Schema

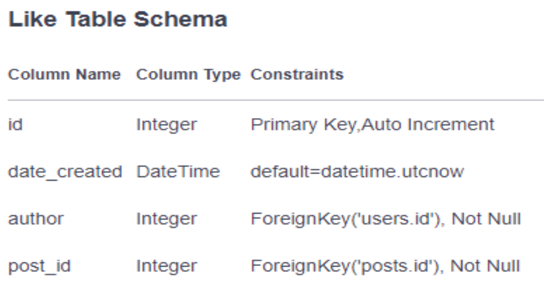
Column Name	Column Type	Constraints
follower_id	Integer	ForeignKey('users.id')
followed_id	Integer	ForeignKey('users.id')
timestamp	DateTime	default=datetime.utcnow

Reason: User can write many posts, comment many posts, like many posts they are one to many relationships with on delete cascade option so that if a user is deleted all relevant posts, like and comments are deleted automatically. Similarly, a post can have many comments and many Users table is linked with followers table with many to many relationships.



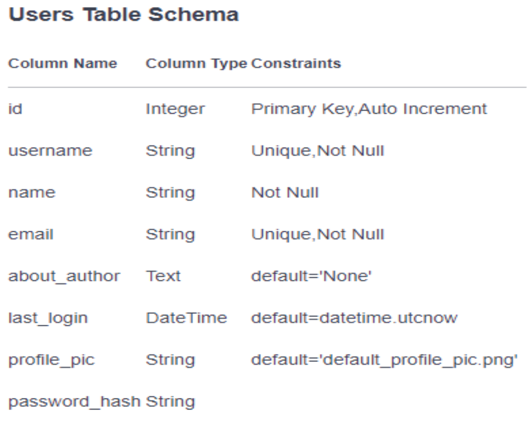
Comment Table Schema

Column Name	Column Type	Constraints
id	Integer	Primary Key,Auto Increment
text	String	Not Null
date_created	DateTime	default=datetime.utcnow
author	Integer	ForeignKey('users.id'), Not Null
post_id	Integer	ForeignKey('posts.id'), Not Null



Like Table Schema

Column Name	Column Type	Constraints
id	Integer	Primary Key,Auto Increment
date_created	DateTime	default=datetime.utcnow
author	Integer	ForeignKey('users.id'), Not Null
post_id	Integer	ForeignKey('posts.id'), Not Null



Users Table Schema

Column Name	Column Type	Constraints
id	Integer	Primary Key,Auto Increment
username	String	Unique,Not Null
name	String	Not Null
email	String	Unique,Not Null
about_author	Text	default='None'
last_login	DateTime	default=datetime.utcnow
profile_pic	String	default='default_profile_pic.png'
password_hash	String	

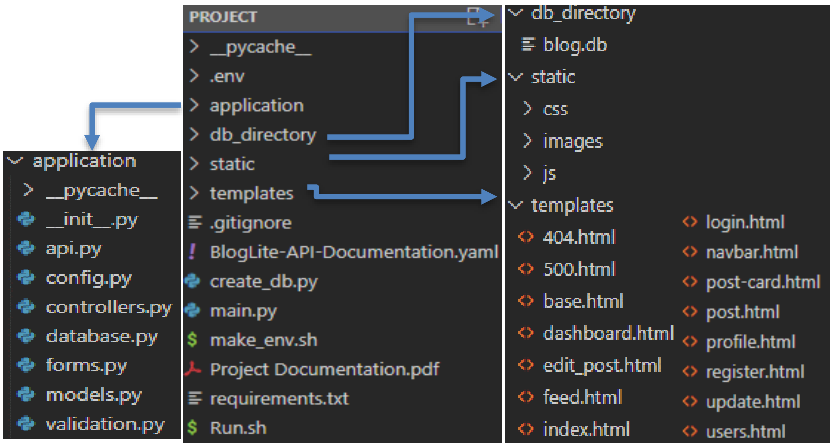
API Design BlogLite-API-Documentation.yaml

User - Operations about user **GET** *\/api/user/{username}* Get User Detail **PUT** */api/user/{username}* **DELETE** */api/user/{username}* Delete a User **GET** */api/user* Get All Users as List **POST** */api/user* Register New User	Post - Operations about post **GET** */api/post/{postid}* Get Post Details **PUT** */api/post/{postid}* **DELETE** */api/post/{postid}* Delete a Post **GET** */api/user/{username}/post* All posts **POST** */api/user/{username}/post* Add Post
Comment - Operations about comment **GET** */api/comments/post/{postid}* Get Post's all comments Likes - Operations about likes **GET** - */api/likes/post/{postid}* No. of likes	follower Info - Get No. of followers and following **GET** */api/user/{username}/followers* Get no of followers **GET** */api/user/{username}/following* Get no of following

For user api I used query return user with username parameter passed in api URL and return user by converting object into JSON with marshal with decorator. Similarly, for post api I used posts relationship to get post for user with username. And for comment, likes and followers count no of objects by count() function.

Architecture and Features

Project is organised as python package. Templated in Template Folder, application/ controllers.py as controllers.



```
PROJECT
> __pycache__
> .env
> application
> db_directory
> static
> templates
.gitignore
BlogLite-API-Documentation.yaml
create_db.py
main.py
make_env.sh
Project Documentation.pdf
requirements.txt
Run.sh

application
  > __pycache__
  __init__.py
  api.py
  config.py
  controllers.py
  database.py
  forms.py
  models.py
  validation.py

db_directory
  blog.db
static
  > css
  > images
  > js
templates
  404.html         login.html
  500.html         navbar.html
  base.html        post-card.html
  dashboard.html   post.html
  edit_post.html   profile.html
  feed.html        register.html
  index.html       update.html
                   users.html
```

1. **Login/Sign Up page**: backend support with login framework, flask-login, flask-wtf and sessions.
2. **APIs** for interaction with users and blogs: CRUD for user and Post.
3. Searching for a user, feed and My Blogs.
4. **Feed**: System will automatically show the blogs from the user follow in a particular sequence based on the timestamp.
5. User profile view with basic stats.
6. Search and Follow / Unfollow Others.
7. **Blog management**: Create, Edit and Remove Blog.
8. **Validation**: All form inputs fields - text, numbers, dates etc. with suitable messages. Backend validation before storing / selecting from database.
9. **Engagement on Blogs/Posts**: Ability to **like** or add **comments** on a blog.
10. **Styling** and Aesthetics as per latest trend.
11. **Fully Responsive** on any device.
12. **CKEditor:** Advanced editor to write and style post content easily.
13. **Pagination:** A single window contains only 9 posts and can change to next page by Next Page Button.
14. **Profile Picture Upload:** Upload profile picture.
15. **Post video upload:** Upload Post Picture.
16. **View my followers and following user details:** Get details of followers and following.
17. **Delete account with confirmation:** Dialog box for account deletion.
18. **Posts as Cards**

Video

https://drive.google.com/file/d/1F_IJjsw1puD_E1F-OytaUYOEmwCKHStP/view?usp=share_link