Author

Name-Mahi Mudgal
Roll no.-21f3002602
Student email address-21f3002602@onlinedegree.iitm.ac.in
Hello everyone, myself mahi currently pursuing IITM Bs degree along with that I am also pursuing BSc in physical science with cs from Maharaja agrasen college Delhi University. I am always curious to learn about new technology.

Description

Bloglite is a multi user app where each user can post their blogs multiple times ,Hence for making it we need to allow user performing all CRUD operation for both User and Blogs .User can follow other user as well so we need to make follower table and on each Blog there can be multiple comments so we need to make comment table as well.

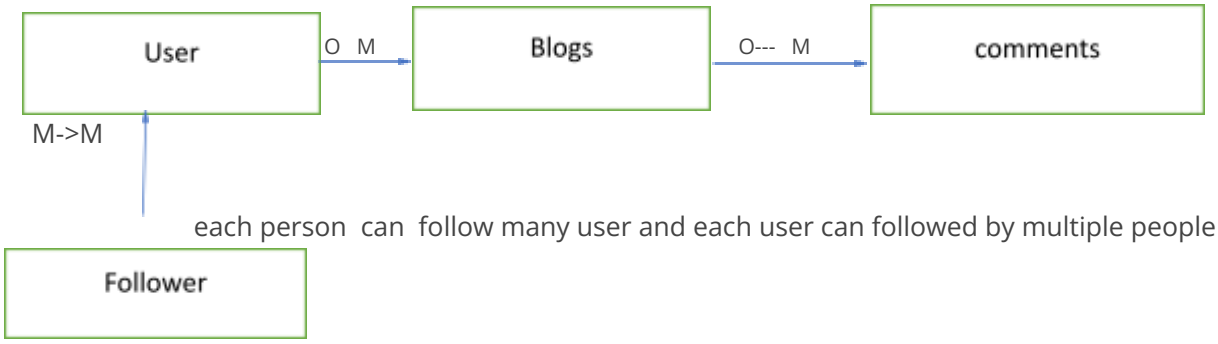
Technologies used

1. HTML and CSS for structuring and styling the front-end of the web application
2. Bootstrap for frontend templates
3. Jinja2 for embedding Python code in HTML templates
4. SQLite and database libraries for storing and interacting with data
5. Flask-Login: adds user session management to Flask
6. Flask-SQLAlchemy: adds support for the SQLAlchemy ORM (Object Relational Mapper) to Flask

DB Schema Design

Table:-Columns
1)User -Username, Password, id
2)Blogs-Blog id, Blog title, Blog Description ,Blog blog-pic ,likes ,date-created,User_id
3)Comments-Blog id, comment ,c.id
4)Followers-follower id , followed id

Relationship
One User has many blog posts each Blog has many comments



Reason for choosing the database:

One to many:

As we can see each user can post multiple times and for identifying users uniquely we have user_id which can connect with blog's serial no which is unique in Blogs, Hence Blog's user id is the foreign key in Blog's table which corresponds to id in user table.

Same goes for Blog and Comments here Blog_id in comments table is the foreign key in Comment table which corresponds to the Blog s.no each s.no can have multiple comments.

Many to Many:

We choose a self referential many to many relationship for storing followers and followed id in table . Follower id and followed _id are the foreign key in Follower table which correspond to primary key column in user table ie. User id .

Whenever a current user follows some other user, the follower id stores the value of current_user_id and followed id stores the value of other user id whom we are following .

For showing the relationship in the user table we use primary and secondary joins for the left and right side of the Followers table respectively.

Architecture and Features

Project Contains ,a website folder and an app.py file. Inside the website folder there is a static folder in which images are stored .
Template Folder in which all frontend templates are stored using jinja templates .There are three .py files 1)for initializing everything .__init__.py file,2)views.py for mostly blogs related routing this is the blueprint file registered in init.py file ,3)auth.py is also a blueprint used to do mostly User related work.
app.py file basically import app from website and run it
Basic Roadmap of App:-

login/Create Account ->Add Blog->See Blog in Profile section ->open blog to view in full screen ->Available Option in Navbar-

1)search:- For searching other users open their profile, follow /unfollow them.

2) Home:- Here you can see everyone blog with author name written on them you can open the profile of that particular user as well .You can also able to view ,like/ comment on everyone posts ,whereas delete and update option visible to your post only

3)User Profile:-Here you can see your own posts, no .of follower, no. of people following you ,no of posts .

3)Feeds:- Here your's and people whom you follow posts is visible

4)User-update :-here we can update our profile password and username

5)Logout :-If you want to change your account and sign in to a different account you can logout .

6)like/comments:-you can see no. of like in profile page and on clicking view comment you can see an popup where you can see comments on your post

Video:- project vedio