Author

Shourjya Bhattacharjee
Roll No: 21f1000757
Email: 21f1000757@student.onlinedegree.iitm.ac.in

I'm a graduate in Physics and currently in the Diploma level of the IITM BS Degree program. This is my first experience in developing a web application!

Description

A multi-user blog application is required to be developed as a part of this project.
The application must have core functionalities such as – User signup and login; User feed; User profile; Create, Edit and Delete blog posts; Search and Follow other users; and a few additional functionalities such as – Like and Comment on posts; and CRUD API's for users and posts. A suitable database structure needs to be created to implement the above functionalities, along with styled web pages for user interaction.

Technologies used

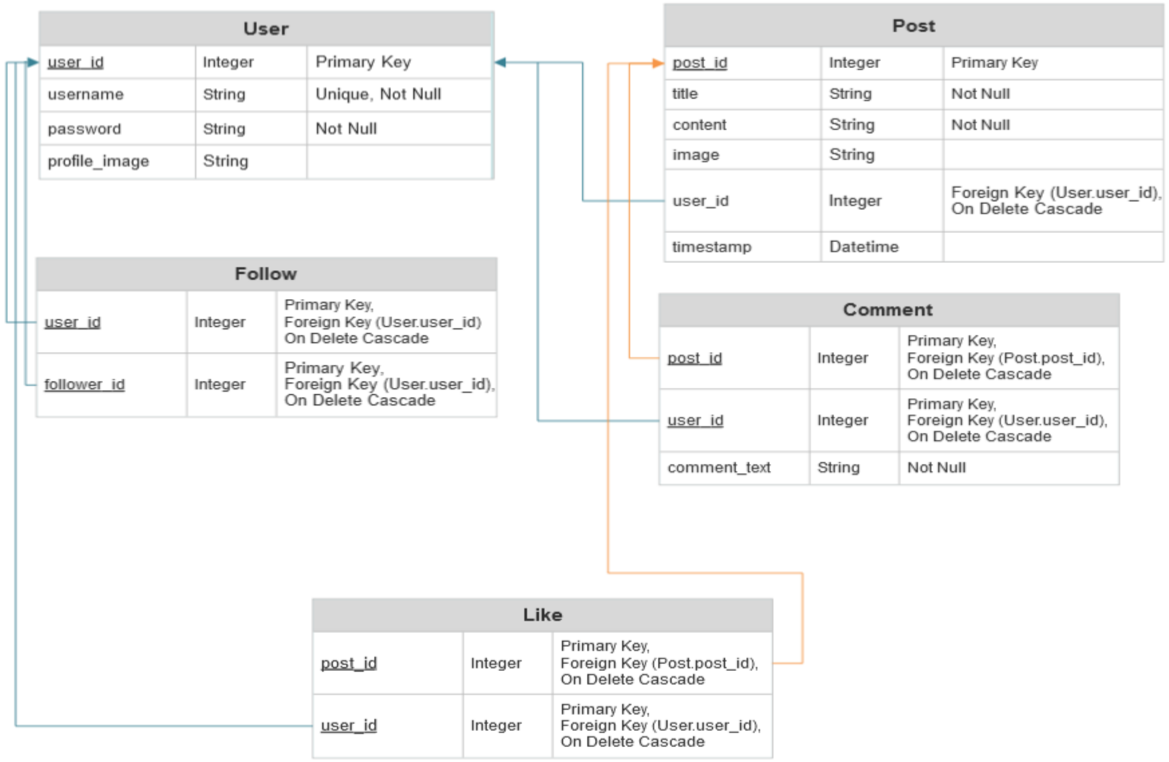
HTML, Jinja2 and Bootstrap – For web page structure and design
SQLite – For data storage
Flask – For application code
Flask-SQLAlchemy – For database interaction with Python
Flask-Restful – For API development

Architecture

The application files are organized in the project folder in the following manner:

- 'static' folder containing Images for user profiles and posts
- 'templates' folder containing HTML files to be rendered
- Database file 'blogdb.sqlite3' consisting data used by the app
- Text file 'requirements.txt' enclosing packages to be installed for running the app
- Python file 'controllers.py' containing controllers for the app
- Python file 'models.py' containing database models for the app
- Python file 'api.py' containing API's for the app
- Python file 'validation.py' containing validation classes used by the API's
- Python file 'main.py' - used for integrating all the components and running the app
- YAML file 'blog_lite_api.yaml' containing documentation for the API's developed
- 'Blog_App_Report.pdf' – Final Report of the project

DB Schema Design



User

user_id	Integer	Primary Key
username	String	Unique, Not Null
password	String	Not Null
profile_image	String	

Post

post_id	Integer	Primary Key
title	String	Not Null
content	String	Not Null
image	String	
user_id	Integer	Foreign Key (User.user_id), On Delete Cascade
timestamp	Datetime	

Follow

user_id	Integer	Primary Key, Foreign Key (User.user_id) On Delete Cascade
follower_id	Integer	Primary Key, Foreign Key (User.user_id), On Delete Cascade

Comment

post_id	Integer	Primary Key, Foreign Key (Post.post_id), On Delete Cascade
user_id	Integer	Primary Key, Foreign Key (User.user_id), On Delete Cascade
comment_text	String	Not Null

Like

post_id	Integer	Primary Key, Foreign Key (Post.post_id), On Delete Cascade
user_id	Integer	Primary Key, Foreign Key (User.user_id), On Delete Cascade

User model contains fields related to the user
Post model contains fields related to a particular post
Follow, Like and Comment models were created to implement related features

API Design

CRUD API’s were created for User and Post models using Flask-Restful and integrated with the app

Features

- User Signup and Login – with backend validation using Python
- User Feed – posts show number of likes and comments
- User Profile – with options to update and delete profile
- Create, Edit and Delete post
- Search and Follow/Unfollow other users
- Add Likes and Comments to posts
- View Post – view a particular post, post page shows comments added

Video

Link:
https://drive.google.com/file/d/1X2EOV0M5TKgS5eKWTcYu6yqeqcvm1e_F/view?usp=share_link