

Author

Name: Bhargav Sharma

Roll Number: 21f1002607

Student Email: 21f1002607@student.onlinedegree.iitm.ac.in

"The way I view myself is as an outstanding candidate for managing multiple priorities with a positive intention and a geek with a strong sense of leadership"

Description:

This project aims to create a web-based application that lets users create blogs that others can read along with some engagement features and also follow and unfollow fellow users so you can watch what they post.

Technologies used:

HTML: used to build a website's structure, --Version:HTML5

Css(Bootstrap): used to style the website, --Version: -Bootstrap (5.3)

Javascript: is used to make web pages interactive

Flask: For managing the backend such as controllers, APIs, and databases and

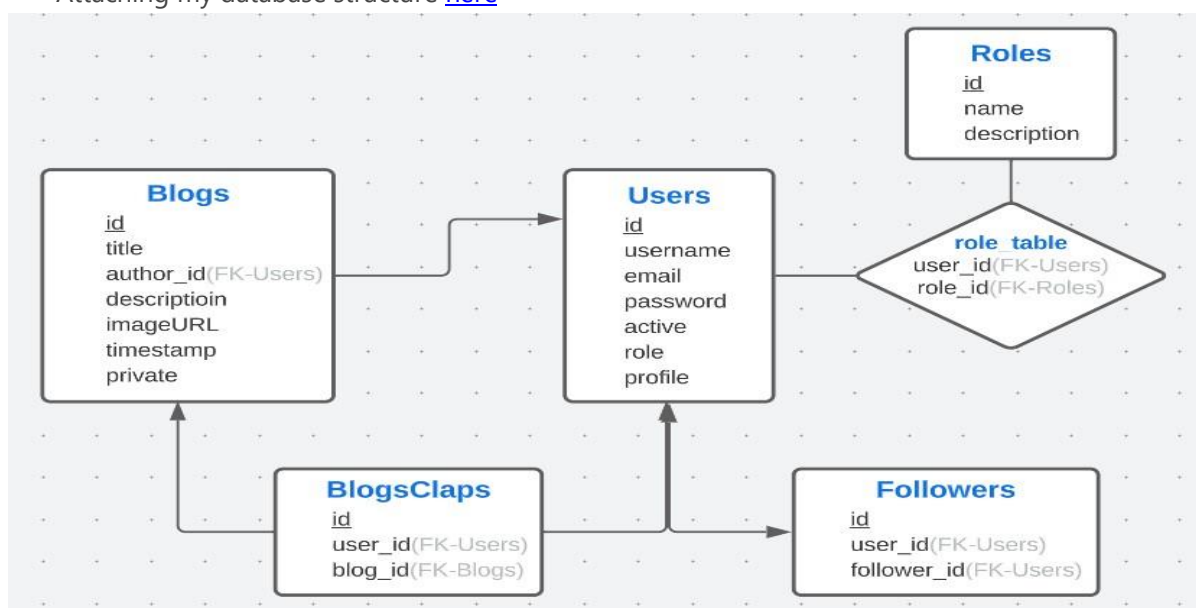
The following are some modules that I specifically used:

bcrypt, email-validator, Flask, Flask-Cors, Flask-Login, Flask-RESTful, Flask-Security, Flask-SQLAlchemy, Jinja2, ,Requests ,SQLAlchemy ,Werkzeug ,WTForms.

Sqlite: SQLite is used as a temporary dataset

DB Schema Design:

Attaching my database structure [here](#)



This database structure allows me to reduce the amount of work I have to do. I can easily find who liked or clapped a blog using the **BlogClaps** table, and I can find which users follow which blogs using the **Followers** table. The **Users** and **Blogs** tables contain enough information to make querying them easy.

API Design

The **flask_restful** module was used to create the **BlogApi** and **UsersApi**, both of which contain a get function that retrieves blogs using **blog_id** and users using **usernames**. I have also implemented functions in API such as `get_username_by_id()`, `get_id_by_username()`, `get_blog_by_id()`, etc. `get_username_by_id()`, `get_id_by_username()`, `get_blog_by_id()`, etc.

Architecture

In the root, there are six directories starting with the application directory consisting of `controllers.py` (routes), `api.py`, `config.py` (configuring the config variables), `db_init.py` (initializing DB), `flaskSecurity.py` (flask_Security configuration), `models.py` (consists of all schemas). Second, we have a database directory which consists of `sqlite3` file, Third is a static directory which has further three directories CSS (CSS files), javascript(js files), and media(will consist of all files uploaded by the user), the fourth directory is of templates(consist all HTML files), the fifth directory is of the virtual environment and last directory of YML files after directories we have three files by `app.py`(main file), `readme.md` .

Features

1. Added private and public blogging features that allow users to hide their blogs from other users by adding private attributes to Blogs tables.
2. A notification section was implemented for when any user starts following you. It is done by adding the user's id in an attribute called notification in the user table.
3. In addition to images, users are allowed to include videos in their blogs.
4. The drag-and-drop images/videos feature was created using javascript events such as `ondrag` and `ondragover` when adding or editing a blog.
5. The application supports login and register functionality which is done using flask_security.

Video

Link of video:

https://drive.google.com/file/d/1ofnzVAOA5x5O5Tg_FlI3WR8Ba1EsN-wgP/view?usp=share_link