Author

Name: Priyadarsh Singh

Roll No: 21f1005104

Student Email: 21f1005104@student.onlinedegree.iitm.ac.in

About Me: Myself Priyadarsh Singh, apart from being a Diploma Level student at IIT Madras BS Degree, I'm a graduate in Statistics and currently working as a DataScience Intern at Culinda. My keen interest lies in the domain of ML, AI and computer vision and I am looking forward to working in the respective domain.

Description

For this BlogLite project, we firstly need to create the database, which will consist of three tables: for users, for blogs posted and final table making a mapping of the user and its respective blog posted. Now, with the help of HTML, Jinja2, flask and Bootstrap we will create webpages for the respective functionality of the application.

Technologies used

HTML - Used to create webpages for the application BlogLite.

CSS - Used to give background to the webpages of the application.

BootStrap - A CSS framework which is used to give graphics and styling to the html webpage.

SQLite - In order to store and fetch the data related to the BlogLite application.

Jinja - Used to make the webpage dynamic and render the data from the database.

flask - In order to develop web applications like BlogLite when using Python.

render_template - A flask function used to generate output from the template file like html pages etc.

session - It is an extension of flask and is the time between a user login into the server and the user logout from the server.

redirect - A flask function which redirects the control to the specified path.

SQLAlchemy - Provides a way to interact with the database engine like SQLite as well as allows to make queries and handle data using Python.

DateTime - Used to store the date and time when a blog was created by a user.

DB Schema Design

Three tables : user, blog, relation were created -

TABLE user("user_id” INTEGER, "username" TEXT UNIQUE, "password" TEXT, "followers" INTEGER, "posts" INTEGER, "photo" TEXT, PRIMARY KEY("user_id" AUTOINCREMENT))

Purpose - Stores the data related to each and every user who has signed up in the application.

TABLE blog ("post_id" INTEGER, "title" TEXT NOT NULL, "caption" TEXT, "image" NUMERIC NOT NULL, "timestamp" TEXT, "user_id" INTEGER NOT NULL DEFAULT 1, "username" INTEGER NOT NULL, PRIMARY KEY("post_id" AUTOINCREMENT), FOREIGN KEY("username") REFERENCES "user"("username"), FOREIGN KEY("user_id") REFERENCES "user"("user_id"))

Purpose - Stores the data related to the blog posted by users.

TABLE relation("relation_id" INTEGER NOT NULL, "id_follower" INTEGER NOT NULL, "id_following" INTEGER NOT NULL, "username_follower" TEXT NOT NULL, "username_following" TEXT NOT NULL, PRIMARY KEY("relation_id" AUTOINCREMENT))

Purpose - Used to store the mapping of the blogs posted by the respective user.

Architecture and Features

#Architecture:
Structure of the project consists of three folders namely ‘static’, ‘templates’, & ‘directory’ as well as a controller file named main.py, a README file and a requirement.txt file.

- static - this folder contains another folder named imageuploads where the images uploaded by the user while uploading a post as well as their profile images are stored. It also contains a CSS file which is used for giving background to the html templates.
- templates - this folder contains the html files of the web application BlogLite such as Login, Signup, Homepage, Userprofile etc.
- directory - this folder contains the database file(SQLite) of the BlogLite application.
- main.py - the main controller file of the application where all the respective models etc are defined.
- .README - file which contains the instructions on how to execute the application.
- requirements.txt - file which contains the packages which need to be installed for smooth functioning of the application.

#Features:

- All the CRUD(Create, Read, Update, Delete) operations for the account created by the user are applicable. Users can create an account, update information relating to their account, see their account statistics as well as can delete their account from the application.
- All the CRUD(Create, Read, Update, Delete) operations for the Blog posted by the user are applicable. Users can create a new blog, update that blog, can see their posted blogs in the myprofile section as well as can delete the blog posted. Users also have the choice to make a particular blog public/private to other users who are following him/her.
- Users can search for other users using their username and see their profile and follow/unfollow them.
- Users have the privilege to see their followers and their profile, as well as have the power to remove them from his/her followers list. Users also can see the profile of those users whom he is following.
- Users will be able to see the blogs posted by other users whom he/she is following, in the order of the latest blog posted by them.

Video Link :

CLICK TO VIEW VIDEO