MAD I Project – September 2022 – Blog Lite

Author

Name (Roll Number): Harasiddh G (21f1002916)
Email: 21f1002916@student.onlinedegree.iitm.ac.in
About Me: I am a third-year student of Economics with an interest in how data can be used to understand our world and hence am pursuing this degree. I'm also interested in the Animal Kingdom, and enjoy watching sports, especially football and cricket.

Description

Based on my understanding, our objective was to create a social media website like several already in existence such as Twitter, Facebook, Instagram, and others. Since I primarily use these sites to share memes with friends, I decided to call my website 'memebook' because I thought it will be amazing to have a single social media website dedicated to sharing memes.

Technologies used:

Flask, Flask-SQLAlchemy, Flask-Restful, sqlalchemy, sqlite3, jinja2 templates (html+css)

DB Schema Design

TABLE 1: USER		
COLUMN NAME	**COLUMN TYPE**	**COLUMN CONSTRAINTS**
id	Integer	Primary Key, Auto-increment
email	String	Unique
first_name	String	-
last_name	String	-
username	String	Unique
password	String	-

TABLE 2: MEME		
COLUMN NAME	**COLUMN TYPE**	**COLUMN CONSTRAINTS**
id	Integer	Primary Key, Auto-increment
title	String	-
caption	String	-
creator_username	String	Foreign Key – user.username, non-null
extension	String	-
likes	Integer	-

TABLE 3: FOLLOWERS		
COLUMN NAME	**COLUMN TYPE**	**COLUMN CONSTRAINTS**
user_username	String	Foreign Key – user.username, Primary Key, non-null
follower_username	String	Foreign Key – user.username, Primary Key, non-null

TABLE 4: MEME_CREATORS		
COLUMN NAME	**COLUMN TYPE**	**COLUMN CONSTRAINTS**
user_id	Integer	Foreign Key – user.id, Primary Key, non-null
meme_id	Integer	Foreign Key – meme.id, Primary Key, non-null

TABLE 5: LIKES		
COLUMN NAME	**COLUMN TYPE**	**COLUMN CONSTRAINTS**
user_id	Integer	Foreign Key – user.id, Primary Key, non-null
meme_id	Integer	Foreign Key – meme.id, Primary Key, non-null

The logic behind using these five different tables seems straightforward. We need to store information regarding Users and Memes, hence tables 1 and 2. The other tables help us keep track of who is following whom, who is creating what meme and who is liking the memes.

API Design

APIs have been implemented for users and memes. User details can be obtained, updated, and deleted based on username. New users can also be created. Both updating and deleting user details requires a password. Similarly, one can obtain, update, and delete memes. One can also create new memes. Passwords are required wherever necessary. However, while the application allows one to work with images, I was unable to implement the same in the API

Architecture and Features

All the logic-based components such as the configuration, the database initialisation, the models, the controllers, the APIs and required validations, are all stored in the applications folder and the filenames are indicative of the contents. The templates are all stored in the templates folder, whereas the static folder contains another folder where the uploaded files can be stored. Apart from this, there is a db_directory file containing the database. There is also a .sql file if one needs to create the database and a .yaml file if one needs to debug and test and the APIs. The main.py file is also present, along with a README file that gives further instructions on how to utilize the application. Please note the presence of the two bash scripts as well, which help in setting up the development environment and running the project locally.

Video

https://drive.google.com/file/d/16RGPkeULAWhIjduPteaKcCTnezVBthwu/view?usp=sharing