

# Author

Aryaman Gupta

21f3001884

21f3001884@student.onlinedegree.iitm.ac.in

I am a college student and a *creative technologist* with front-end development and design experience. Looking to upscale my skills in the programming field, I am doing this degree to gain exposure to various other technologies and frameworks, such as Flask.

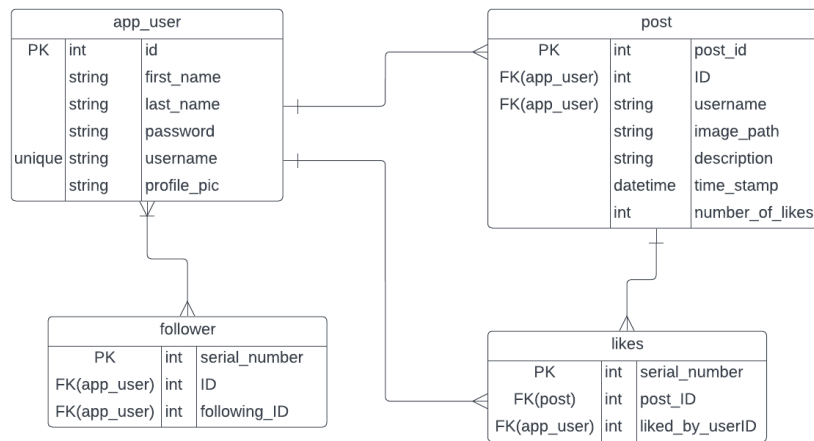
## Description

In this project we were supposed to build a blogging web application using flask (a microweb framework) and connect it to a database using sqlalchemy and sqlite. The web app must contain basic functionalities such as signing up a user, logging in, seeing another user's profile, seeing your own profile, editing your own profile, adding posts, following/unfollowing other users, interaction with other user's posts et cetera.

## Technologies used

- **os**  
Copy images from anywhere in the computer to 'static' folder of my project when uploading a blog or profile picture.
- flask\_sqlalchemy -> **SQLAlchemy**  
Integrates SQLAlchemy with Flask. This handles setting up one or more engines, associating tables and models with specific engines, and cleaning up connections and sessions after each request.
- **flask\_login**  
Using login\_manager to maintain user login sessions, logout and allowing access to URLs only when the user is logged in.
- **bcrypt**  
Hashing the passwords and then storing them into the database in order to maintain security. Once hashed, the passwords cannot be unhashed, but only be compared, like when a user enters a password to login, the entered password is compared with hashed password to verify credentials. This prevents access to passwords even to the database admin.
- **datetime**  
To store timestamps of post creation in the Posts table in the database.
- werkzeug.utils -> **secure\_filename**  
Secure\_filename module is used to return a secure filename of the file uploaded by the user. The problem is that there is that principle called "never trust user input". This is also true for the filename of an uploaded file.
- **uuid**  
Generates a unique filename based on existing filename and date/time of upload.
- pillow -> **Image**  
Used to resize image to 406px \* 406px so that all the cards in the feed have the same size and look uniform.

# DB Schema Design



- **App\_user** - consists details of all the users currently registered in the database
- **Post** - consists of all the posts existing in the uploaded by users in app\_user table
- **Follower** - consists information of who follows whom
- **Likes** - consists information of which post is liked by which user

## Architecture and Features

1. The main flask app resides in '*app.py*', which is just inside the main project folder BLOGGER
2. The BLOGGER folder consists of a file named '*requirements.txt*' which contains all the libraries used in the project.
3. The BLOGGER folder consists of 3 subfolders-
  - a. *instance* - Consists of the sqlite database named 'database.sqlite3'
  - b. *static* - Consists of pictures uploaded by users when they create posts. It also consists of a subfolder named 'images' which contains user profile pictures.
  - c. *templates* - Consists of various html files used to render output

### Features-

- A. Ability to sign up, consists of **password validation**, upload profile picture, basic validation checking **whether all fields are filled or not**.
- B. Successful signup redirects to the login page which shows **an alert** whether your signup was successful or not.
- C. The feed shows posts uploaded by the users you follow in an **ascending order** of their time of upload.
  - a. You can interact with the posts by **liking** it.
  - b. You can also click the **read more button** to read the whole description of the post
  - c. You can click on the name of the user on the post to go to their **profile**
- D. Your profile
  - a. shows the number of followers, following and posts you have.
  - b. You can click the followers or following text to see the **list of followers/following**.
  - c. You can see all the posts posted by you, delete button is there to **delete a post**
  - d. **'Add Post'** button is there to add a new post
  - e. **'Edit Profile'** button is there from where you can change your name, password and profile picture.
- E. Finally a **search button** is present on the navigation bar which opens a search bar. It supports **suggestive searching** where you can type few words and it shows all the users starting with the words by autocompleting it.

## Video

[https://drive.google.com/file/d/1ST0TXel7PZBmf5rimCkRvKWEbPGQj4I/\\_view?usp=sharing](https://drive.google.com/file/d/1ST0TXel7PZBmf5rimCkRvKWEbPGQj4I/_view?usp=sharing)