

## Author

Kirupa Krishan G

21f1006352

21f1006352@student.onlinedegree.iitm.ac.in

Nothing exciting about me , I am the kind of person who likes to keep to his work and focus on what is necessary . Take challenging tasks that help me with my future goals

## Description

This project tasks us with the notion of building a social network platform that is able to connect people with each other. To be able to create an account for yourself ,follow your friends, create/upload your photos that you want your friends/followers to see , like them , comment on them .

## Technologies used

**Flask** : Library that forms a structure for the application and acts like a link between the python library and the internet .

**Flask-SQLAlchemy** :Responsible for the interconnection of the database to the app . This particular library is responsible for the CRUD operation on the data we give it. The data can be manipulated based on the schema of the database and also create relationships between tables .This particular aspect of creating relationships between databases helps us to avoid writing the join statements.

**Werkzeug.security** :Used for the purpose of encrypting/hashing the data that is sent back and forth between the app and the internet. To protect the data from unauthorized personnel

**Flask-Login** :Used for creating a session for the given user . To remember if the user is still logged in or not and do tasks based on the login status of the user .

**Jinja2** :Used to create html templates in html files . In order to do basic operations like iteration , conditional statements the jinja library is utilized. The code here is a python script that acts and gives the jinja what actions are to be done. This is in particular necessary for reduction of code.

**Timeago** : This is in particular a simple library that is used to note and format the date and time at any given instance. This is required in order to allot every post and comment a date and time posted

**Os** : This particular library is used for the purpose of interacting with the OS of the system . It helps us to create , save , update and delete files and folders from the system. To do system operations from within the code.

## DB Schema Design

### Tables:

User:

Columns:

Uid -> Integer,primary\_key=True,autoincrement=True  
Email -> String(150),unique=True  
Username ->String(150),unique=True  
Name -> String(150))  
Caption -> String(150),default="Caption"  
Password -> String(150)  
Date\_created -> DateTime(timezone=True),default=func.now()  
image\_file -> String(50),nullable=False

Relationships:

Comment: One to Many relationship of the no. of comments the user has made

Post: One to Many relationship of the no. of posts the user has uploaded.

Follows: Many to Many relationship of people who follow user and people who user

follows

Post:

Columns:

Pid -> Integer, primary\_key=True

Content -> String(200), nullable=True

media -> String(32), nullable=True

Date\_posted -> DateTime, nullable=False, default=func.now()

User\_id -> Integer, ForeignKey('users.uid'), nullable=False

Relationships:

Like: One to Many relationship of the people who like the post.

Comment: One to many relationship of people who commented on the post

Comment:

Columns:

Cid -> Integer, primary\_key=True

Content -> Text, nullable=False

Post\_id -> Integer, ForeignKey('posts.pid'), nullable=False

User\_id -> Integer, ForeignKey('users.uid'), nullable=False

Date\_posted -> DateTime, nullable=False, default=func.now()

Like:

Columns:

Post\_id -> Integer, db.ForeignKey('posts.pid')

User\_id -> Integer, db.ForeignKey('users.uid')

Followers:

Columns:

User\_id -> Integer, db.ForeignKey('users.uid')

Follows\_id -> Integer, db.ForeignKey('users.uid')

User: This table collects all data that is needed from a new user to create an account for him on the platform.

Post: Collects all data required to make a post. The image is saved with its file name. Caption as String.

Comment: Collects data required to make a comment . The data is about who comments on what post and what they comment on it .

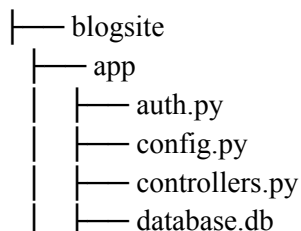
Like: This is used to store the data of who likes what post.

Followers: This is used to store the data of who follows whom.

API Design

No API design implemented .

Architecture and Features



```
|
| |— database.py
| |— __init__.py
| |— models.py
| |— __pycache__
| |— static
| |— templates
| |— views.py
|— main.py
|— Project Documentation.pdf
|— Readme.md
|— requirments.txt
```

Some features include :

Login/Logout: To login and logout the user.

Signup: To create an account for the user.

Create Post/Delete Post: To add a post with a caption and a feature to like and comment on it. Delete the same post.

Explore: To explore all other users present in the platform and check their profile .

Follow/Unfollow: Follow people to look at their posts and unfollow if you don't want to look at their posts.

Like/Unlike: Like/Unlike a post .

Comment: Comment on any users posts that you follow

Chek-Profile: Check the profile of other users on the platform.

View-Post: To look at the comments and the individual posts separately .

Feed-Page: Home page where you can see all posts that are uploaded by people you follow including your own.

Edit/Profile: To edit your profile to update your DP and change your About section , username , name.

Video

[https://drive.google.com/file/d/1ZCIT1\\_7lPODfVZWPRp7SjcCfB0nn3tQA/view?usp=sharing](https://drive.google.com/file/d/1ZCIT1_7lPODfVZWPRp7SjcCfB0nn3tQA/view?usp=sharing)