Author
Suyamoon Pathak
Roll no: 21f3002588
21f3002588@student.onlinedegree.iitm.ac.in
From Gandhinagar, Gujarat
I am actually from Nepal studying BTech in Computer Science and Engineering at Pandit Deendayal Energy University, Gandhinagar, Gujarat.

Description:
I have developed a blog application which can have multiple users. They can upload blogs which have a title, content and/or an image. Users can post any number of blogs. Posts can be liked and commented on. Users can follow each other and feed is generated on that basis. Only the posts from followed users will appear on the feed. Each user has its own dashboard, where he can see his followers, his followed users, and his posts.

Technologies used:

1. **flask: Flask**(for running the app), **Blueprint**(for organizing the logic into subdirectories), **render_template**(to generate output from a template file based on Jinja2 engine), **redirect**(to redirect the user to target location), **url_for**(to redirect user to a source url), **request**(to accept user input from form), **flash**(to display flash messages), **jsonify**(to convert JSON to response object), **current_app**(to access configuration of the current running app)
2. **flask_sqlalchemy: SQLAlchemy**(to access the database's SQL functionalities)
3. **os: Path**(to get the root of Flask application)
4. **flask_login: login_user**(to create a session for a user that will persist as the user stays logged in), **logout_user**(to end the session), **login_required**(to restrict views to logged in users), **current_user**(to use functionalities of the logged in user)
5. **werkzeug.security: secure_filename**(to return a secure version of the filename), **generate_password_hash**(to generate the hash value for user's password), **check_password_hash**(to check if the user's input password matches the hash value stored in the database)
6. **sqlalchemy.sql: func**(produces registered function instances, here it is used with now() to save datetime)
7. **uuid: uuid1**(to generate unique filenames while storing in the database)

DB Schema Design

1. User
   - Id(Integer,Primary_Key)
   - Email(String(150),Unique)
   - Username(String(150),Unique)
   - Password(String(150))
   - Date_created(Datetime(timezone=true),default=func.now())
2. Post
   - Id(Integer,Primary_Key)
   - Text(String(150),Unique, nullable=False)
   - Title(String(150),Unique, nullable=False)
   - Filename(String(150),Unique, nullable=True)
   - Date_created(Datetime(timezone=true),default=func.now())
   - Author(Integer,Foreign_Key=user.id,ondelete="cascade", nullable=False)
3. Comment
   - Id(Integer,Primary_Key)
   - Text(String(200),Unique)
   - Date_created(Datetime(timezone=true),default=func.now())
   - Author(Integer,Foreign_Key=user.id,ondelete="cascade", nullable=False)
   - Post_id(Integer,Foreign_Key=post.id,ondelete="cascade", nullable=False)
4. Like
   - Id(Integer,Primary_Key)
   - Date_created(Datetime(timezone=true),default=func.now())
   - Author(Integer,Foreign_Key=user.id,ondelete="cascade", nullable=False)
   - Post_id(Integer,Foreign_Key=post.id,ondelete="cascade", nullable=False)
5. Followers
   - Follower_id(Integer, Foreign_Key=user.id)
   - Followed_id(Integer, Foreign_Key=user.id)

App Architecture:

- App.py
- Sitefiles
  - Static
    - Images
      - Uploads
        - (post images)
    - Index.js
    - Styles.css
  - Templates
    - Base.html
    - Create_post.html
    - Edit_post.html
    - Followed.html
    - Followers.html
    - Home.html
    - Login.html
    - Messages.html
    - Posts_div.html
    - Profile.html
    - Search.html
    - Signup.html
  - __init__.py
  - Authentication.py
  - Models.py
  - Views.py
- Instance
  - Database.db

Description:

- App.py is my main flask app, Instance folder has the database, and all other files are stored in SiteFiles folder
- Static folder has css and js files. It also stores the images uploaded by users in their blogs under Uploads folder which is in Images folder.
- Templates folder has the html files.
  - Base.html has nav bar, which is extended in all other html files except login and signup
  - Create_post.html is for creating post by the user
  - Edit_post.html is for editing the post by the user
  - Followed.html shows the users followed by the current user
  - Followers.html shows the users who follow the current user
  - Home.html is where Feed is shown
  - Login.html is where user can enter his credentials and login to the app
  - Messages.html has different categories of flash messages defined that are being shown in the app.
  - Posts_div.html has the posts which are to be shown in the feed
  - Profile.html has the profile view of the users
  - Search.html has the search results obtained from the search box in the nav bar
  - Signup.html has the form for creating new user
- __init__.py file allows SiteFiles folder to be imported as a package. All routes and models have been defined in the my_app function. Database and path for storing blog images have also been declared in this file.
- Authentication.py routes for user authentication – login, logout, signup.
- Database models for User, Post, Comment and Like are present in models.py
- Views.py has all other routes used in the app.

Video Link:

https://drive.google.com/file/d/1bjyZL8AMSE9jGG2Qzop6Xo5VC6WsAdNm/view?usp=share_link