Description :

This project is a platform for sharing images where users can register, login, create and manage posts, like and comment on other users' posts, follow and unfollow other users, and search for other users using their usernames.

Technologies used :

1. Flask - for application code
2. Jinja2 templates and Bootstrap for HTML generation and styling.
3. SQLite and SQLAlchemy for data storage.

DB schema design :

a. The database has several models/tables created: User, Post, Like, Comment, and Follow. Each table has different attributes and helper functions.

b. The database is designed to store user information, post details and their images, like, comments and followers details and relationships between them for the smooth functioning of the application.

c. Structure and details of the columns :

1. User: This table stores information about the users of the application, including their unique id, username, password_hash, profile_image, followers, following, follower_count, following_count.
2. Post: This table stores information about the posts created by users, including the post id, title, timestamp, status, caption, image, user_id, user, likes, comments.
3. Like: This table stores information about the likes given by users to the post, it includes the like id, user_id, post_id, timestamp.
4. Comment: This table stores information about the comments given by the users on the post, it includes the comment id, timestamp, content, user_id, post_id.
5. Follow: This table stores the relationship between the followers and the following, it includes the id, follower_id, followed_id, timestamp and a unique constraint is added to this table to make sure that a user can follow another user only once.

API design :

I have not implemented the API functionality (part of the recommended section and not core).

Architecture and Features :

a. The project is organized using the Model-View-Controller (MVC) architecture, with the controllers handling logic and routing, templates for displaying views, and models for interacting with the database.

b. Features implemented include :

1. User registration and login (using username and password)

2. User's profile :
   - ☑ Basic profile view of the user
   - ☑ Ability to view the total number of pots created
   - ☑ Ability to view the number of followers and people you follow.
   - ☑ Ability to view the list of posts created.

3. Post creation and management (blog/post management)
   - ☑ Create a new post
   - ☑ Update / Edit a post
   - ☑ Delete / Remove a post

4. Searching other users using usernames.

5. Following / unfollowing users

6. User's Feed
   - ☑ Show the posts created by other users
   - ☑ Navigate to the user's profile on clicking the username on the blog or post.

Additional features :

1. Liking / unliking posts

2. Commenting posts :
   - ☑ Add comments to posts
   - ☑ Update / edit existing comments
   - ☑ Delete/ remove existing comments.

3. Validation :
   - ☑ Client-side validation with HTML
   - ☑ Server-side validation with python / WT forms

These features are implemented using helper functions and forms created separately for each functionality.