MAD 2 Final Project Report - Blog Lite V2

1. Introduction:

In this project, we were asked to create an application that would be like a social media application (similar to Instagram) where different users can create and share their posts with their followers. Users would be able to see the posts created by users they follow, check their profile information, create their own posts and Update and Delete their posts as well. The first part of this project was to create the model (database structure). In this case, I had 5 tables whose attributes evolved overtime.

The 1st table is users table (contains user specific information) with attributes id (primary key), username, email, password, active, nof (no. of followers), nop (no. of posts), following (no. of users this user is following), last_seen and profilepic.
The 2nd table is user_posts (contains post information) with attributes post_id (primary key), id (from users table), title, description, image_URL, last_update (datetime of when post was created or last edited).
The 3rd table is follow (contains information about which users are being followed by whom) with attributes logid (primary key), follower (user who is following a user), followee (user who is being followed by user mentioned in the corresponding follower).
The 4th table is Role (can be used to give different levels of access to user) with attributes id (primary key, different from id used in users), name (name of role), description (description of role).
The 5th table is role_user (can be used to link users to their respective roles) with attributes user_id (foreign key from users.id) and role_id (foreign key from Role.id)

2. Profile Information Page:

This is the page that sits at the home url (base url + /) which is protected by flask security with @login_required. Once authenticated, the user would gain access to this page and see 3 sections.

The first section is the user's account information. The first part of this section shows the user's profile picture along with the option to update it. If the user has not set a profile picture, a default profile picture is displayed. When the user clicks on the update button, a section pops up where the user can enter the URL for an image or the path for a PNG file and click on Submit to save the changes in the database or click on close to close the profile picture update pop up. Additionally, this section also displays the number of posts the user has created over time, the number of users the current user is following and the number of users the current user is followed by. The following and followers count is hyperlinked. When the hyperlink is clicked a page with the list of users opens up.

The second section ('My Posts' Section) contains a list of posts made by the current user with an option to view the whole post, an option to remove the post and the option to edit. Clicking on the hyperlinked 'view full post' opens up a page with the entire post information containing the post title, description, image along with the name of the user who created the post and last updated date time indicating when the post was created or edited, the full post view page also contains the option to edit and remove the post if the current user is also the author of the post.

The third and final section is the 'Export CSV details' section from where the user can Export list of followers, list of users the current user is following, the current user's feed and the posts created by the current user. If the user does not have any posts and/or followers and/or is not following other users, the respective export buttons are disabled. This page can be accessed from a dropdown available in the navigation bar.

3. My Feed Page:

   On this page, the user can see all the posts created by the users who the current user is following. These posts on the My Feed page are in a shortened format containing the post title, the post description limited to 100 characters as well as the hyperlinked author name and the option to view the full post. Clicking on the hyperlinked post author takes the user to the authors page. This page can be accessed from a dropdown available in the navigation bar.

4. Post View Page:

   Each time the user clicks on the ‘view full post’ hyperlinked text, they are redirected to this page where the current user can view the title, description, image, author name and last updated date-time associated with the post. If the post was created by the current user, they would also see the option edit and the option to remove the post.

5. Add Post Page:

   This page can be accessed via the ‘My Profile’ drop down menu in the navigation bar. On this page the user can fill in the title, description, and image URL, and click on submit to fill in the information into the database. In the image URL field the user can fill in the path for a PNG file or the URL for an image file.

6. Search Page:

   This page can be accessed via the navigation bar. On this page the user can search for all the users by username and the options would populate below the search bar automatically. Next to each username shown is a ‘Go to page’ button that redirects the user to the profile page of the adjacent users.

7. Profile Information Page of other users:

   The profile page for other users can be accessed by searching for usernames on the search page and clicking on the corresponding ‘Go to profile’, by clicking on the hyperlinked author name in the feed or by going to the followers or following page and clicking on their username. Once the current user is directed to the user profile page,They would first see the page title as ‘User Profile:’ followed by the username. They would then be able to see 2 sections, identical to the first two sections on the current user profile information page except below the account information is a small section indicating the ‘Following’ status which is a boolean value indicating if the current user is following the user whose profile is being viewed and a follow or unfollow button based on the follow status.

8. Security:

   This app uses flask security and the @login_required function in almost all endpoints to keep them protected. If a user tries to access these endpoints directly. They are prompted to login via the default flask security login template. The sign-up page and login page are not protected with @login_required. I have also used flask flash to flash messages in case a user is using incorrect credentials to login or trying to create an account with username or email already in the database.

9. Scheduled Jobs:

   The automated jobs use celery workers and crontab function to start an async job sending a daily reminder to users who have not been active and to send a monthly report of overall user activity at the start of the month.