



Individual Project Requirement

User Authentication

Register

Feature Acceptance Criteria

Login

Feature Acceptance Criteria

Verification

Feature Acceptance Criteria

Reset Password / Forgot Password (*Bonus)

Feature Acceptance Criteria

User Profiling

User Info / Detail

Feature Acceptance Criteria

User Posts and Likes (*Bonus)

Feature Acceptance Criteria

Content Upload / Edit

Create / Post Content

Feature Acceptance Criteria

Edit Content

Feature Acceptance Criteria

Delete Content

Feature Acceptance Criteria

Share Content (SSR) (*Bonus)

Feature Acceptance Criteria

Content Interaction

Fetching Posts

Feature Acceptance Criteria

Post Detail

Feature Acceptance Criteria

Comment

Feature Acceptance Criteria

Like / Unlike

Feature Acceptance Criteria

User Authentication

Register

Users should be able to register themselves into the web app by submitting an **email, username, and password**. After a user has submitted their registration form, an email should be automatically sent to verify their account.

Feature Acceptance Criteria

1. User can input email, username, password, and repeat password.
2. Emails should be unique and its format should be validated (proper email format).
3. Usernames should be unique.
4. Passwords should contain at least 8 characters including an uppercase letter, a symbol, and a number.
5. Newly registered users will have a status of **unverified**, they will need to verify themselves through a link sent to their email.
6. Throttle registration requests by disabling the register button until a response is received.
7. Display proper error messages for cases such as weak password, used email or username, and other errors.
8. This page shouldn't be accessible by users if they are logged in

Login

Users should be able to login using credentials that the user has registered with. **Verified users** should be able to login using their **username or email**, and **password**. **Non-verified users** should still be able to log in, but they **cannot interact with the web app's features** other than resending a verification link.

Feature Acceptance Criteria

1. User can input their username or email (one field) and password.
2. Password visibility can be toggled.
3. Automatically redirect to home page upon successful login.
4. Display proper error messages when password is incorrect.

5. Throttle registration requests by disabling the login button until a response is received.

Verification

Users should be able to verify their account by clicking a link sent to the users' email upon finishing the registration process.

Feature Acceptance Criteria

1. The verification link should be in the format of `http://{domain}/{path}/{token}`.
2. Each user's verification link should be unique.
3. Users should be automatically verified upon landing on the verification page.
4. A user may be able to resend verification emails. So when a user requests a new verification link, previously sent tokens should be invalidated.
5. Use an email template.

Reset Password / Forgot Password (*Bonus)

Create a page for users to reset their password. Users should be able to input their email, and upon submitting their request, the user should receive an email containing a link to a reset password page.

Feature Acceptance Criteria

1. The reset password link should be in the format of `http://{domain}/{path}/{token}`.
2. Each user's reset password link should be unique.
3. User can input their new password.
4. Password visibility can be toggled.
5. A user may be able to resend reset password emails. So when a user requests a new reset password link, previously sent tokens should be invalidated.

User Profiling

User Info / Detail

Create a page to display and edit the user's information. Users can complete their personal information such as full name, bio, username, and profile picture. If a user hasn't verified their email, they can request to resend a verification link in this page.

Feature Acceptance Criteria

1. Display user's information
 - a. Full name
 - b. Bio
 - c. Username
 - d. Email
 - e. Profile Picture
2. Edit user's information
 - a. Full name
 - b. Bio
 - c. Username
 - i. Username should be unique, check username uniqueness before saving new value.
 - d. Profile Picture
 - i. Upload a profile picture.
 - ii. **DO NOT** let the user input a link to an image.



Note that a user's email cannot be changed.

3. Send verification email button when user hasn't verified their email. Hide the button when a user is verified

User Posts and Likes (*Bonus)

Users can also see their own posts here, other than that, they can also view posts other people's post that the user has liked. Preferably, you should create 2 tabs; 1 for the user's posts, the other for the posts the user has liked.

Feature Acceptance Criteria

1. Display the users' own posts
2. Display the users' liked posts

Content Upload / Edit

Create / Post Content

Users should be able to create or post their own content (images). The content can later be viewed by other users.

Feature Acceptance Criteria

1. The content should have the following data:
 - a. Media (image)
 - b. Caption
 - c. Created Date
 - d. Number of likes
2. The media of the content should be uploaded to your server/database. Do not use other sites' media URL as your content. Use your own URL which will point to your own server and media files.

Edit Content

After a post has been created, the user should still be able to edit its caption.

Feature Acceptance Criteria

1. The user is able to edit the post's caption
2. The user is only able to edit their own posts

Delete Content

After a post has been created, the user should be able to delete the post

Feature Acceptance Criteria

1. The user is able to delete the post

2. The user is only able to delete their own posts

Share Content (SSR) (*Bonus)

Create a feature where users can automatically share a post to Facebook, Whatsapp, and/or Twitter. You can use a library called [react-share](#) to achieve this functionality. Don't forget to use the correct meta tags so that the shared URLs can display the proper information when embedded in social media

Feature Acceptance Criteria

1. There should be 4 share buttons
 - a. Whatsapp
 - b. Twitter
 - c. Facebook
 - d. Copy Link
2. Each share button should open a dialog for their respective social medias except for the *copy link* button which will only copy the post's URL.
3. Use SSR and proper meta tags so that the proper information can be displayed when posts are embedded.

Content Interaction

Fetching Posts

Create a page dedicated for posts to be displayed. Since there isn't any "followers / following" mechanism in this web app, posts created by all users should be displayed in this page. Paginate the posts list using infinite scroll.

Feature Acceptance Criteria

1. Display the following data in each post
 - a. Username
 - b. Media/Image
 - c. Created Date
 - d. Number of likes

2. Use infinite scrolling to paginate the posts timeline
3. The pagination should be done in the back end, not the front end. Do not paginate using local states

Post Detail

Create a page dedicated for a post's detail. Display the post's complete information such as the user's username, post created date, the post's media/image, number of likes, and most recent comments.

Feature Acceptance Criteria

1. Display the following data in the page
 - a. Username
 - b. Media/Image
 - c. Created Date
 - d. Number of likes
 - e. 5 most recent comments
2. Use a unique URL for each post

Comment

Any user should be able to comment on any post. Create an input for users to submit their comments. Show the user's username and the comment's created date when displaying comments in a post.

Feature Acceptance Criteria

1. The user is able to submit a text as a comment in a post
2. Paginate the comment section by using a "see more" button.
3. Limit each comment to be only 300 characters max
4. Display the user's username on each comment in the comment section

Like / Unlike

Add a like button for users to like each other's posts (even their own posts). The like button should act as a toggle. Meaning that when a post is liked, the like button

should automatically turn into an “unlike” button which can be used to undo the user’s action.

Feature Acceptance Criteria

1. The user is able to like a post
2. The like button should show if the user has liked the post or not. For example, in Instagram, the like button turns red when the post has been liked by you (the user).
3. The like button should be a toggle. After a user has liked a post, the like button becomes an “unlike” button.
4. When a user likes a post, the number of likes in a post should automatically increment.