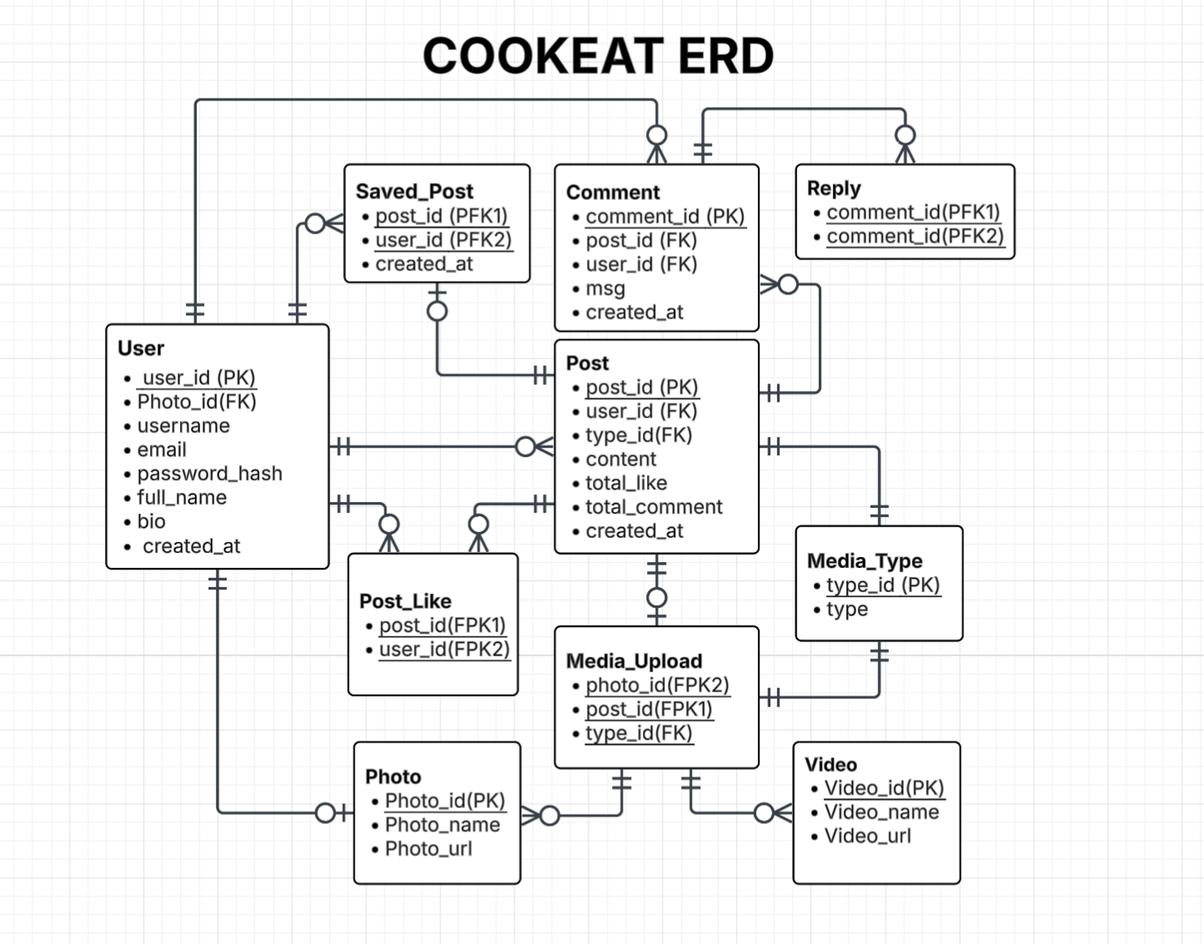
**Made by :**

**Muhammad Ashraf Malabali, Jefferson Ruben Sulaiman, Muhammad Dzaki Abrar**

“Our project aims to create a social media website that took inspiration from Instagram, but instead of posting pictures of our life, it’s pictures of food and recipes.”

**1. Database Table**

**Table information:**

User (user\_id, username, email, full\_name, password, profile\_picture, bio)

  - Stores information about each user.

  - user\_id is the primary key.

  - username and email must be unique.

  - profile\_picture and bio are optional user profile details.

Post (post\_id, user\_id, description, media\_url, media\_type\_id, created\_at)

  - Stores posts made by users.

  - post\_id is the primary key.

  - user\_id is a foreign key referencing User(user\_id).

  - media\_url points to the uploaded image or video.

  - media\_type\_id references Media\_Type(type\_id) to validate media type.

  - created\_at stores the timestamp of the post.

Media\_Type (type\_id, type)

  - Stores the allowed media types for posts.

  - type\_id is the primary key (e.g., 1 for image, 2 for video).

  - type is a descriptive string (e.g., 'image', 'video').

Comment (comment\_id, post\_id, user\_id, msg, created\_at)

  - Stores comments made by users on posts.

  - comment\_id is the primary key.

  - post\_id references the post being commented on.

  - user\_id references the user who made the comment.

  - created\_at stores when the comment was made.

Reply (reply\_id, comment\_id, user\_id, msg, created\_at)

  - Stores replies to comments (not to other replies).

  - reply\_id is the primary key.

  - comment\_id is a foreign key referencing the parent comment.

  - user\_id is a foreign key referencing the replier.

  - msg is the reply content.

  - created\_at tracks when the reply was made.

Like (post\_id, user\_id)

  - Represents users liking posts.

  - Composite primary key of post\_id and user\_id.

  - post\_id references the liked post.

  - user\_id references the liking user.

  - Each user can like a post only once.

Comment\_Like (comment\_id, user\_id)

  - Represents users liking comments.

  - Composite primary key of comment\_id and user\_id.

  - comment\_id references the liked comment.

  - user\_id references the user who liked the comment.

Saved (post\_id, user\_id)

  - Represents users saving posts.

  - Composite primary key of post\_id and user\_id.

  - post\_id references the saved post.

  - user\_id references the saving user.

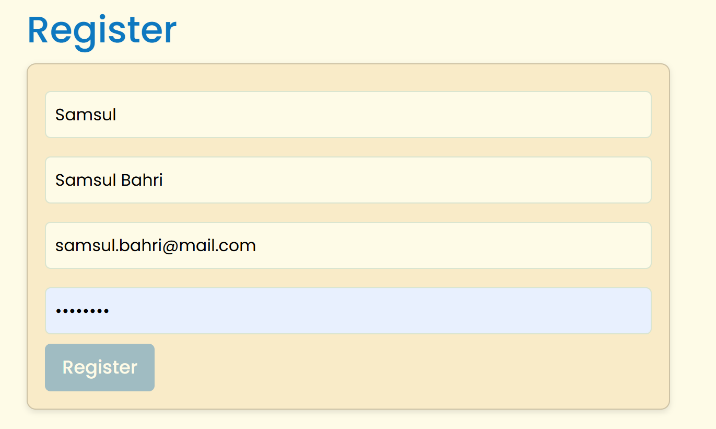
  - Each user can save a post only once.

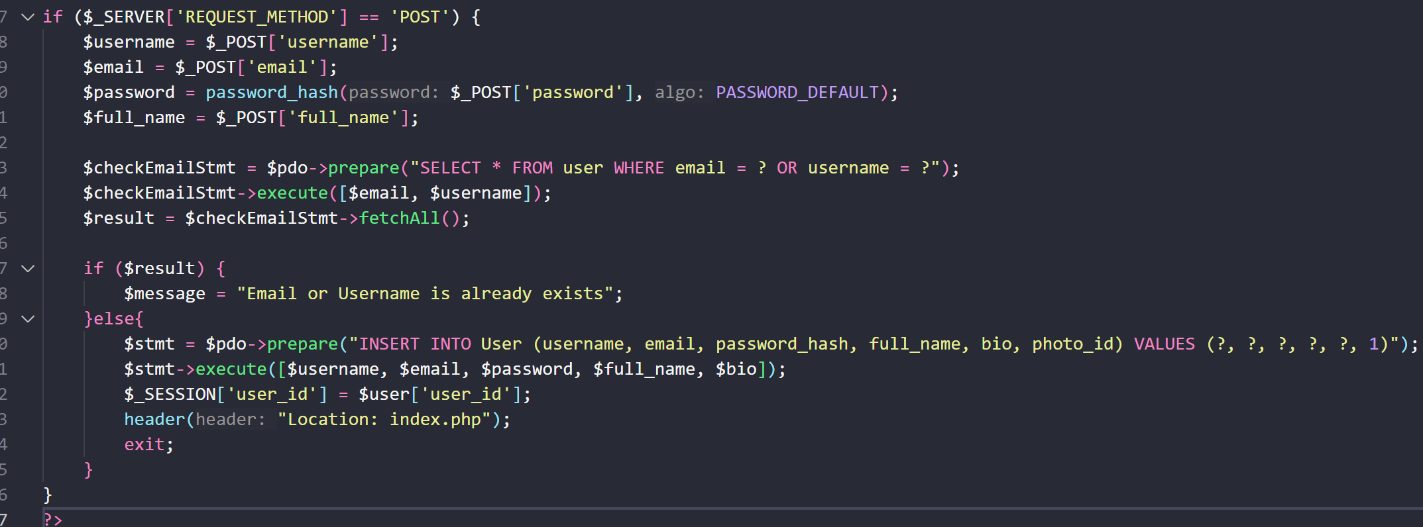
**2. CRUD Implementation**

1. **Create**

Here we have several Create features, such creating a new account, creating posts, and commenting.

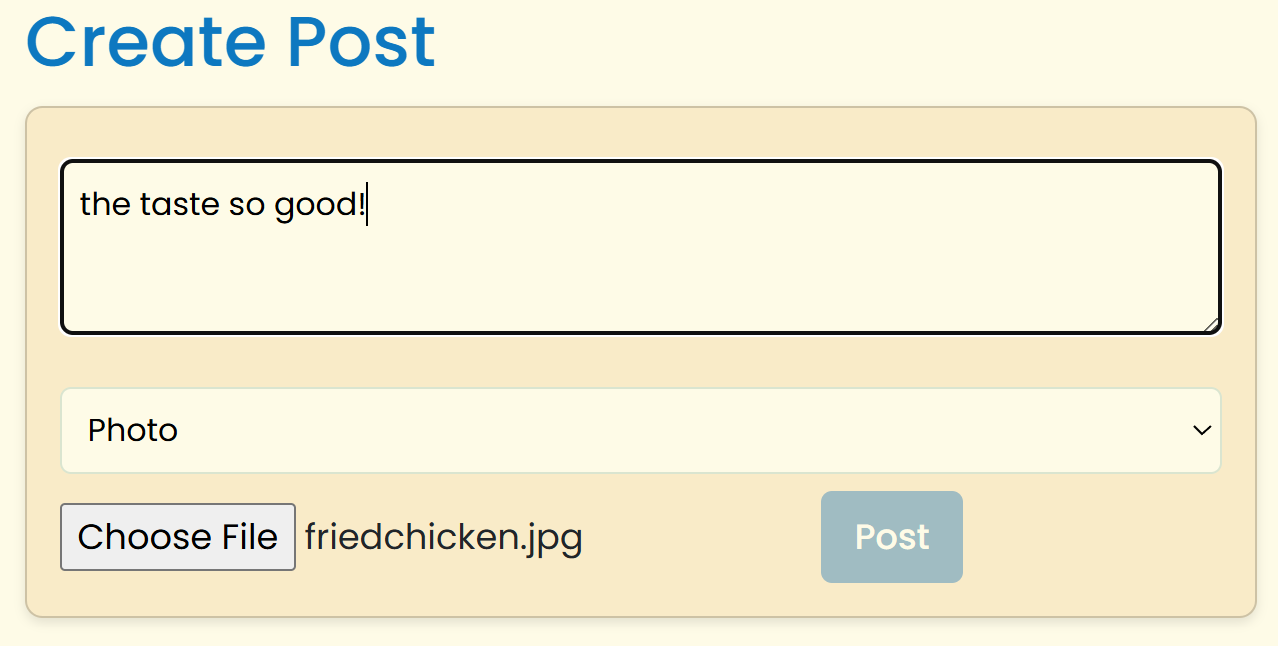
* **Create new account (register.php)**

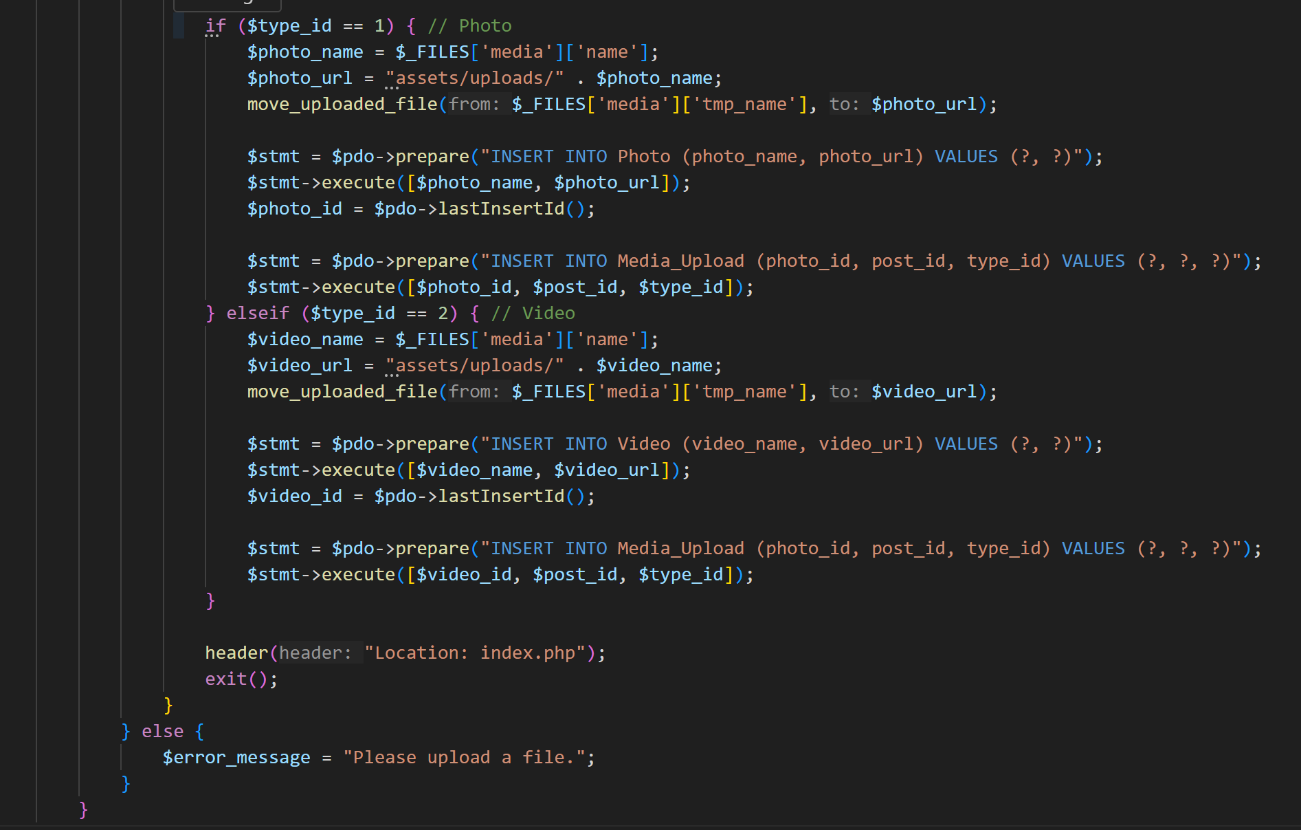


****

process of creating a new account is done by entering data such as username, full\_name, email, password, and photo\_id into the User table, and using hashed for the password.

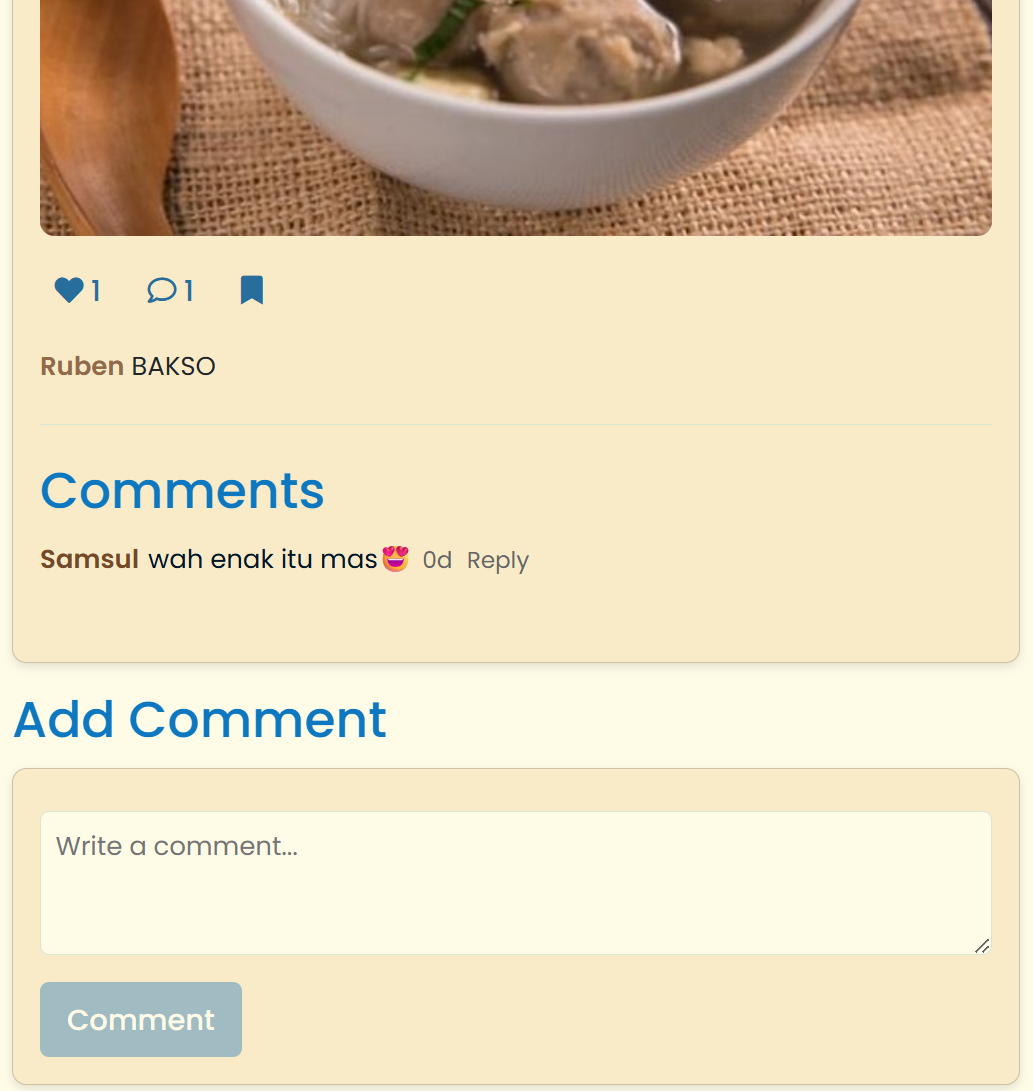
* **Create post (create\_post.php)**

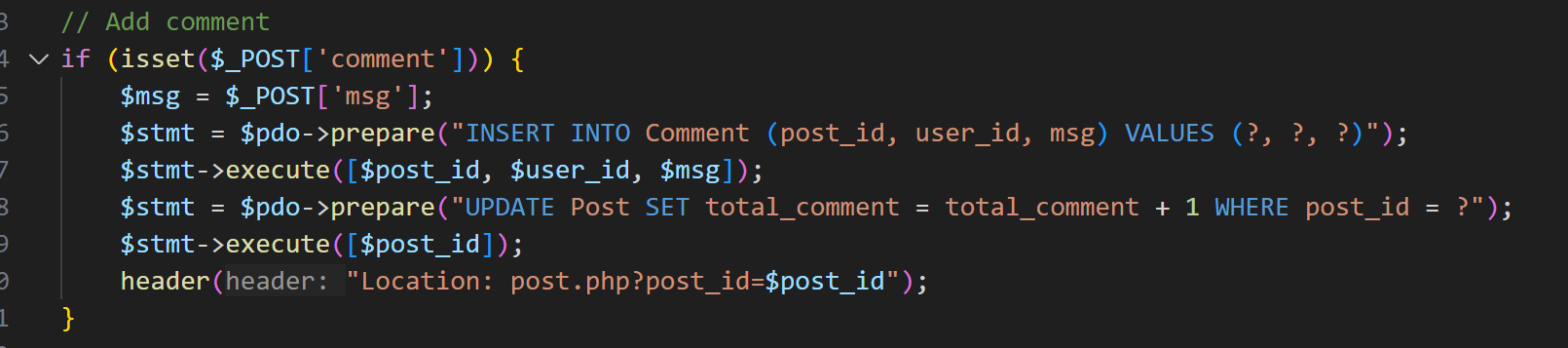
****

****

Create a new post is involve saving data such as content, user\_id, type\_id for photo/video, and media to the Post and media\_upload tables, with validation to ensure content is not empty and media files only .jpg/.mp4

* **Comment (post.php)**

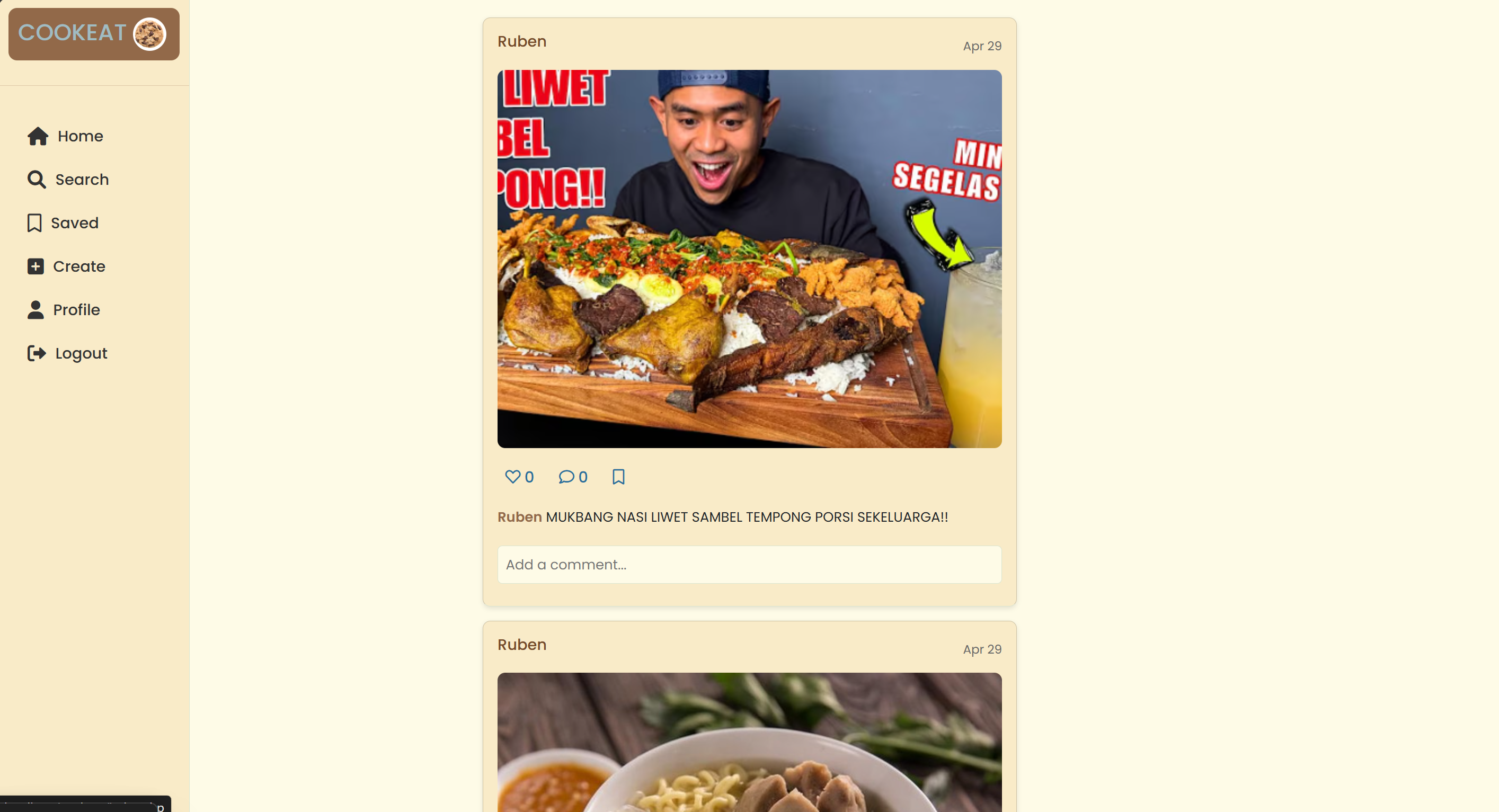
****

****

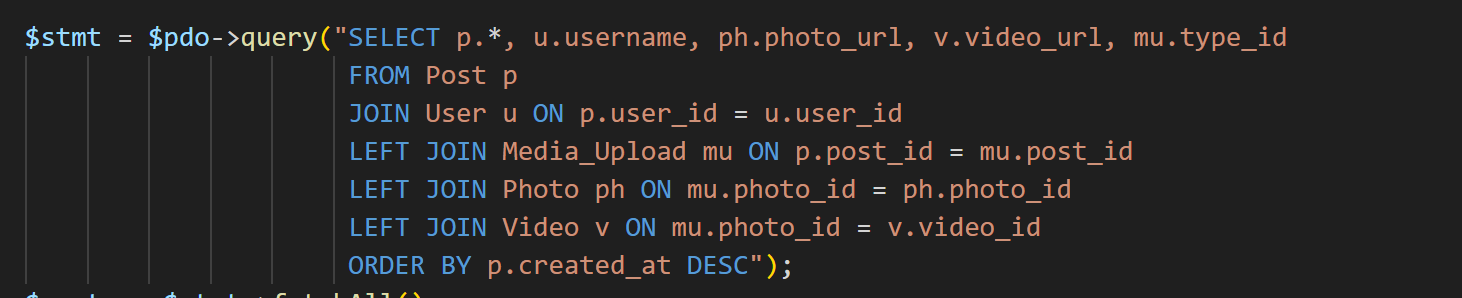
The comment creation process saves the content, user\_id, and post\_id to the comments table, with validation to ensure the comment is not empty and the post exists.

1. **Read**

* **Show post feed (index.php)**

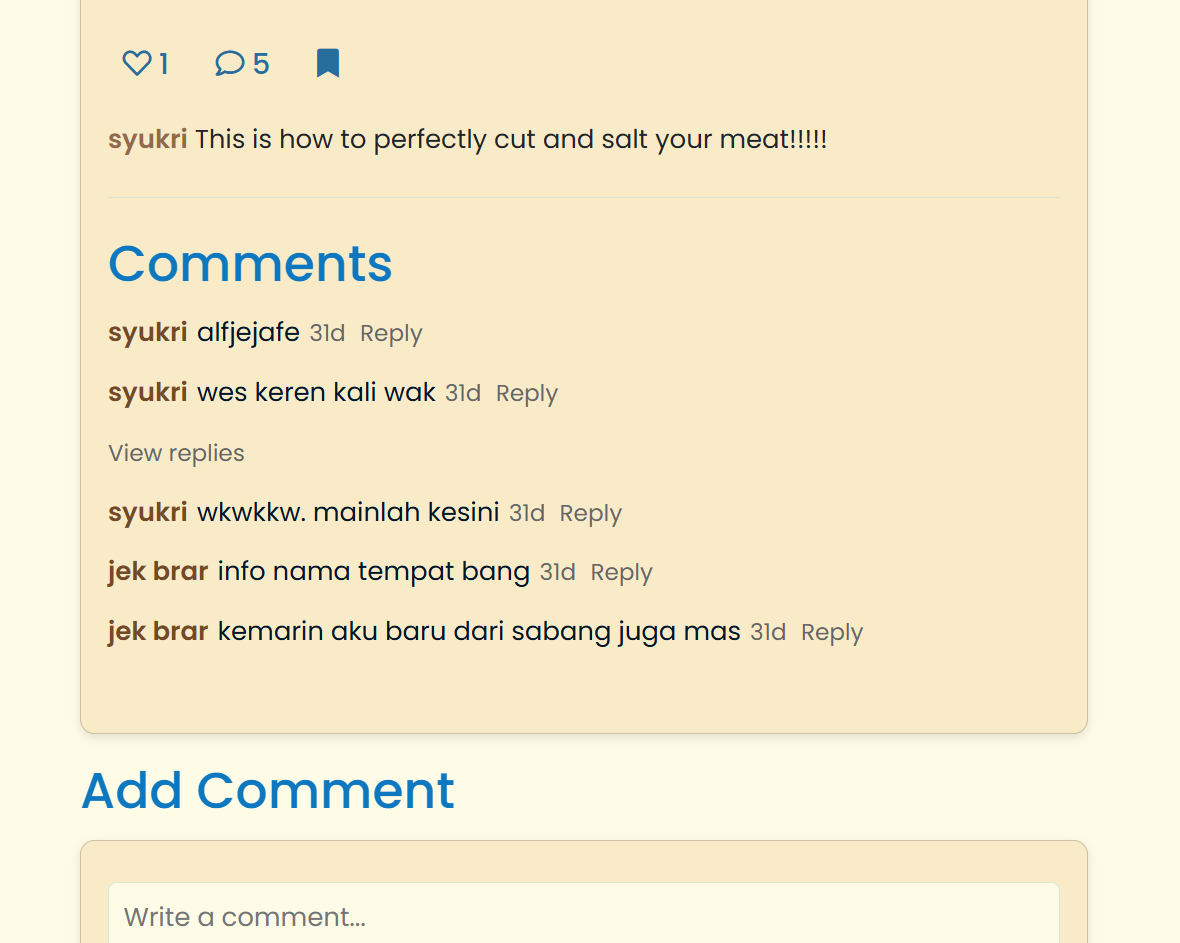
****

Displays a feed of recent posts from all users, get from the ‘Post’ table with information (`User`) and media (`Photo` or `Video`) information via `Media\_Upload`. Posts are sorted by creation time `created\_at` in descending order.



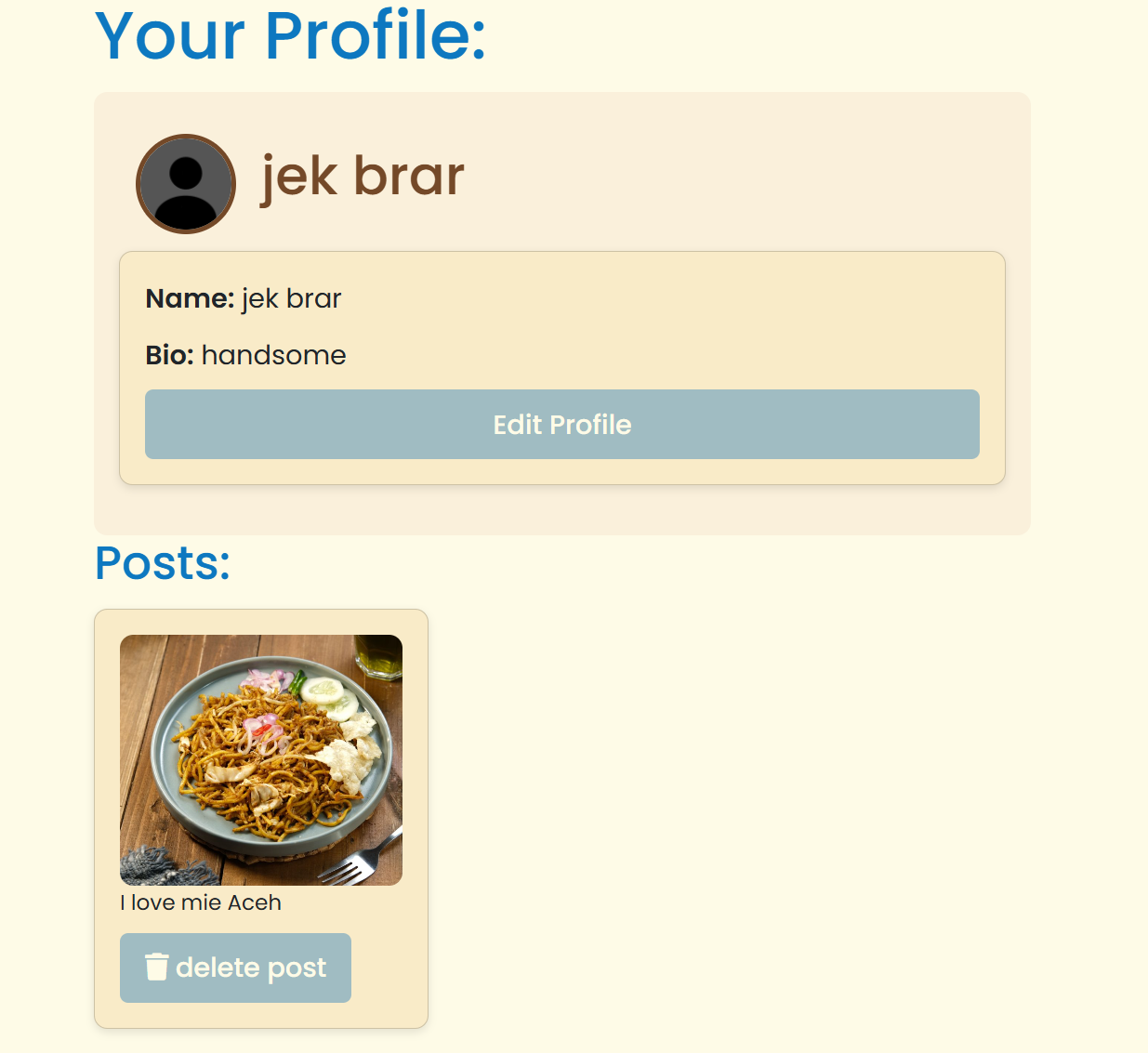


* **Show comment**

****

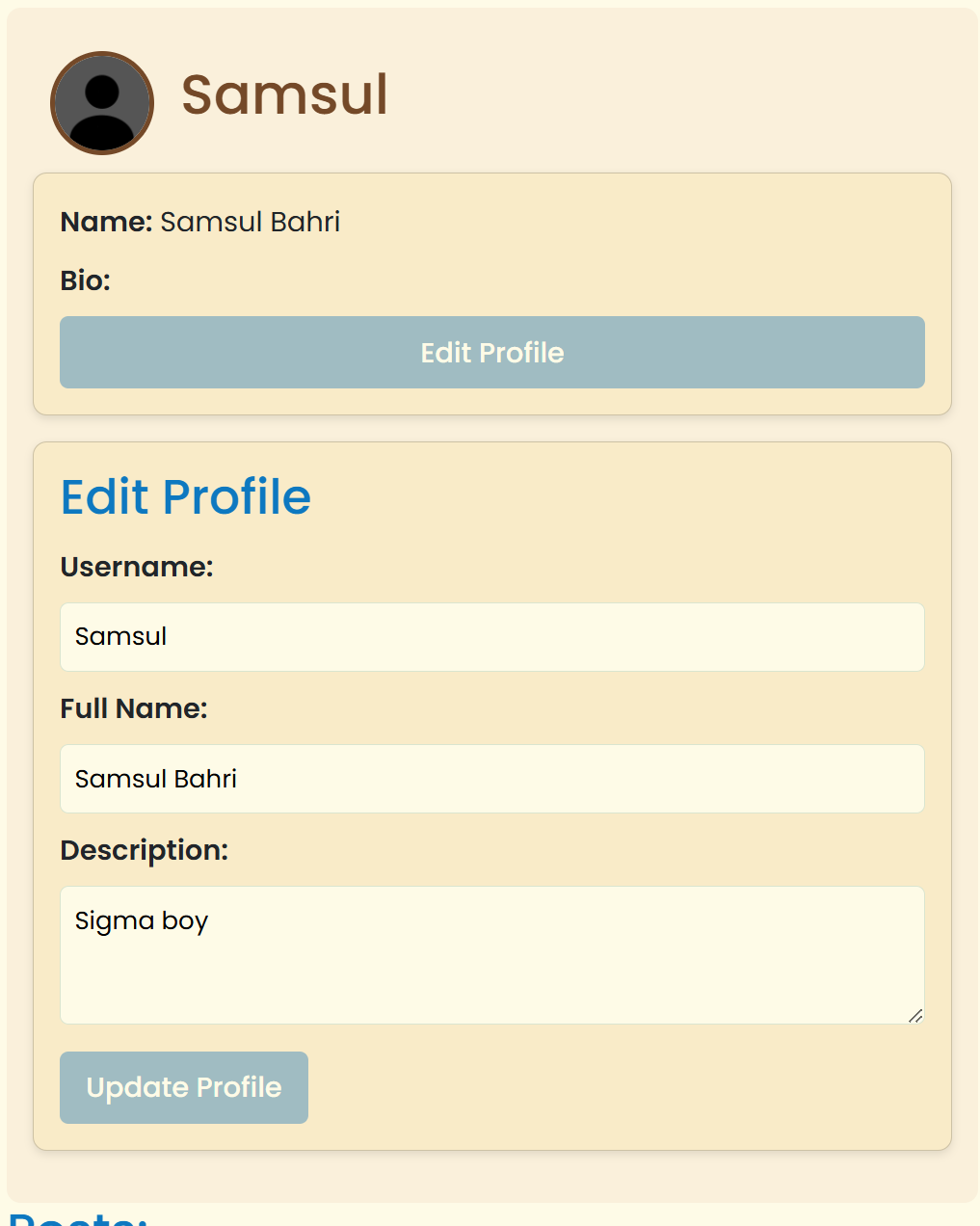
This allows all users to view existing comments. this will display a list of comments along with their usernames, sorted by the time they were created.

* **User profile (profile.php) etc.**

****

1. **Update**

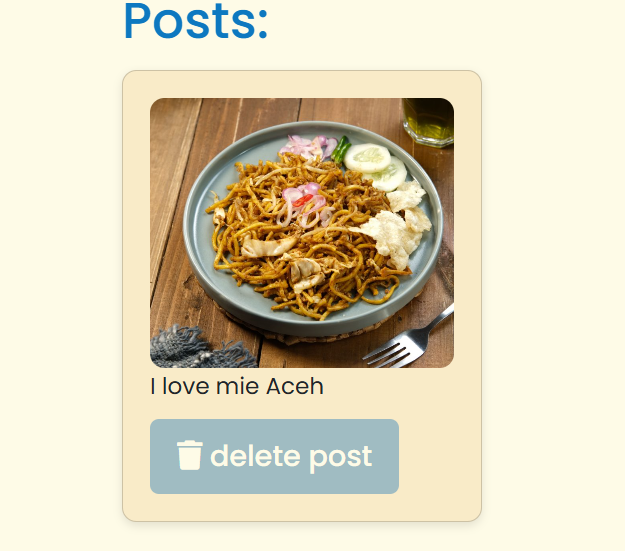
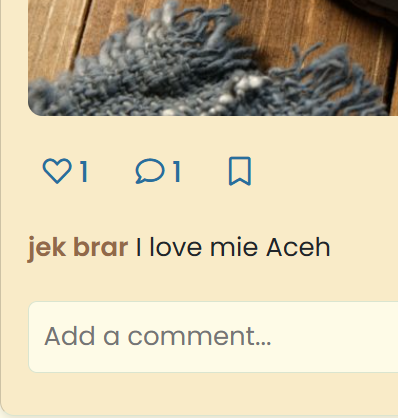
* **Update user profile (profile.php)**

****

Users are given the ability to edit their profile, including name, description and so on, with validation to ensure the username is not empty and unique, using prepared statement

1. **Delete**

* **Delete post, unlike, unsave**

**** 

**Delete post** process involves verifying ownership based on user\_id, then deleting entries from the comments, media\_upload, saved\_post, likes, and Post tables sequentially to maintain foreign key integrity. and also javascript confirmation.

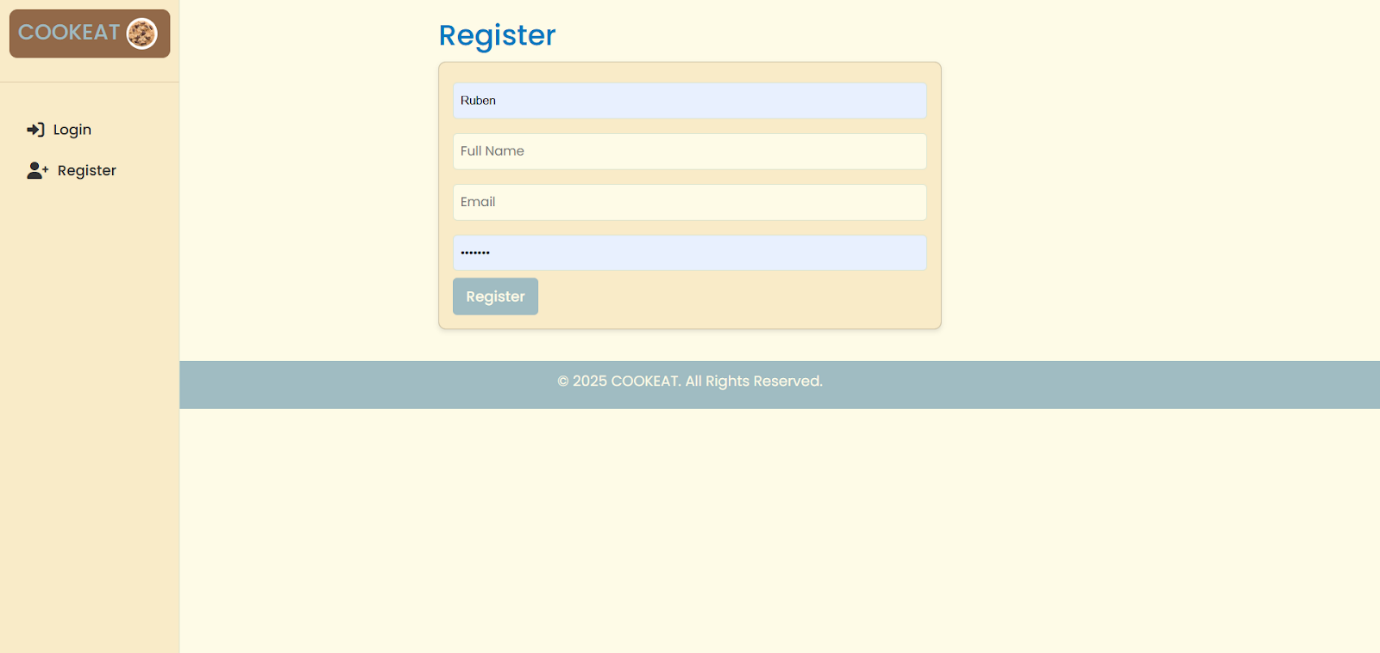
**unlike** process is done by deleting entries from the likes table based on user\_id and post\_id, then reducing the like\_count in the Post table using GREATEST to prevent negative values.

as well as **unsave** by deleting entries from the saved\_post table based on user\_id and post\_id.

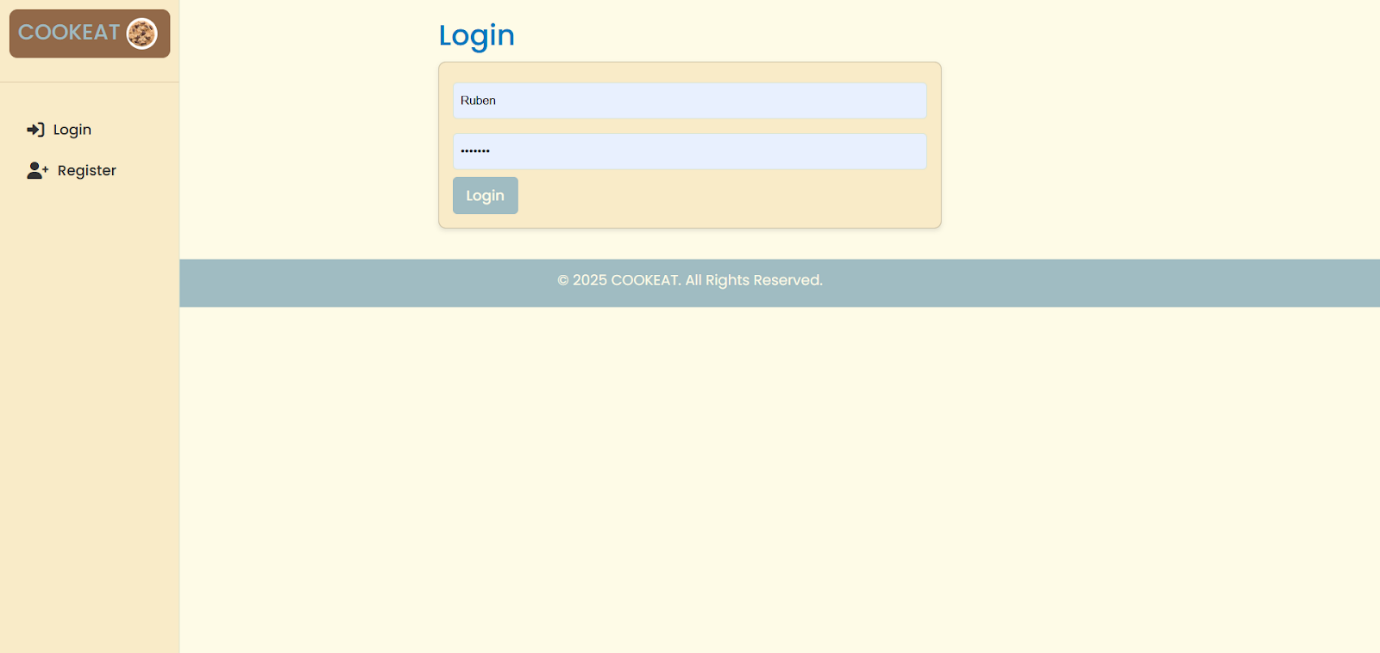
**3. Flow of Application**

The website will first ask for us to login, but if we don’t have an account, there is a “Register” button.

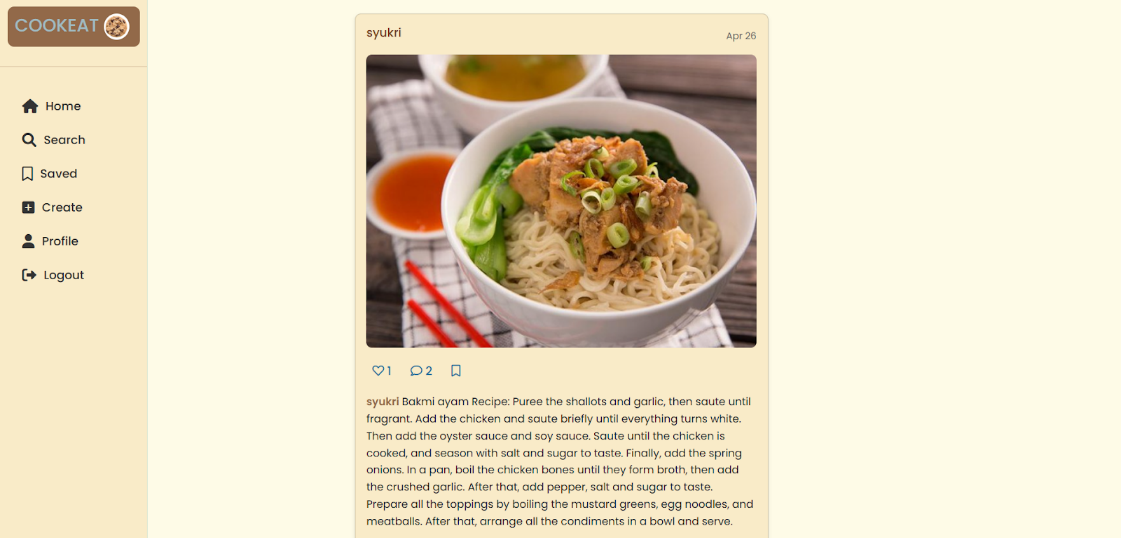
****

****

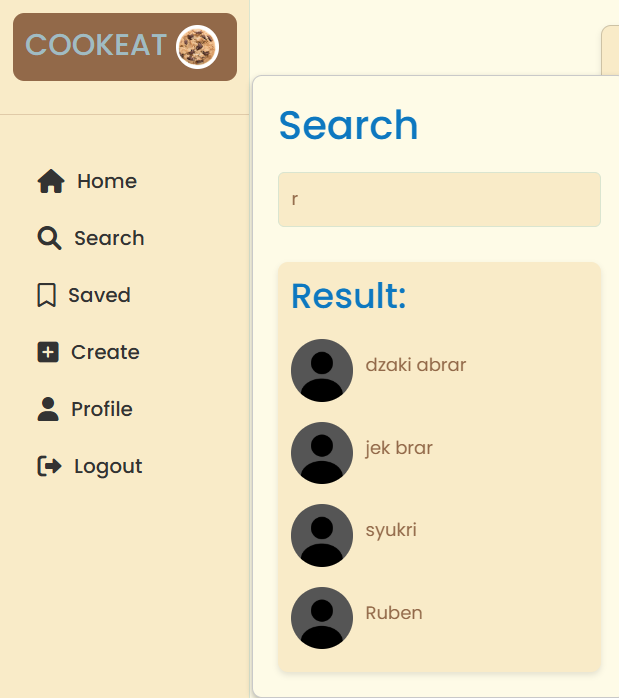
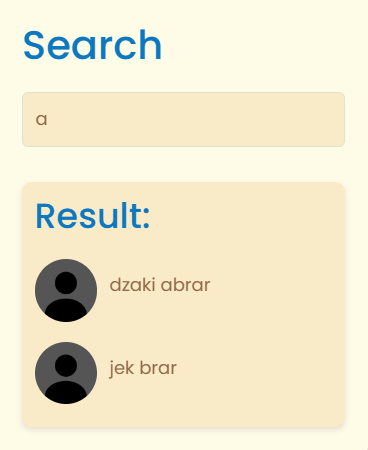
When the Register button is clicked, it goes to another page where you can fill in all of your credentials such as user name, email, full name, and lastly the password.

****

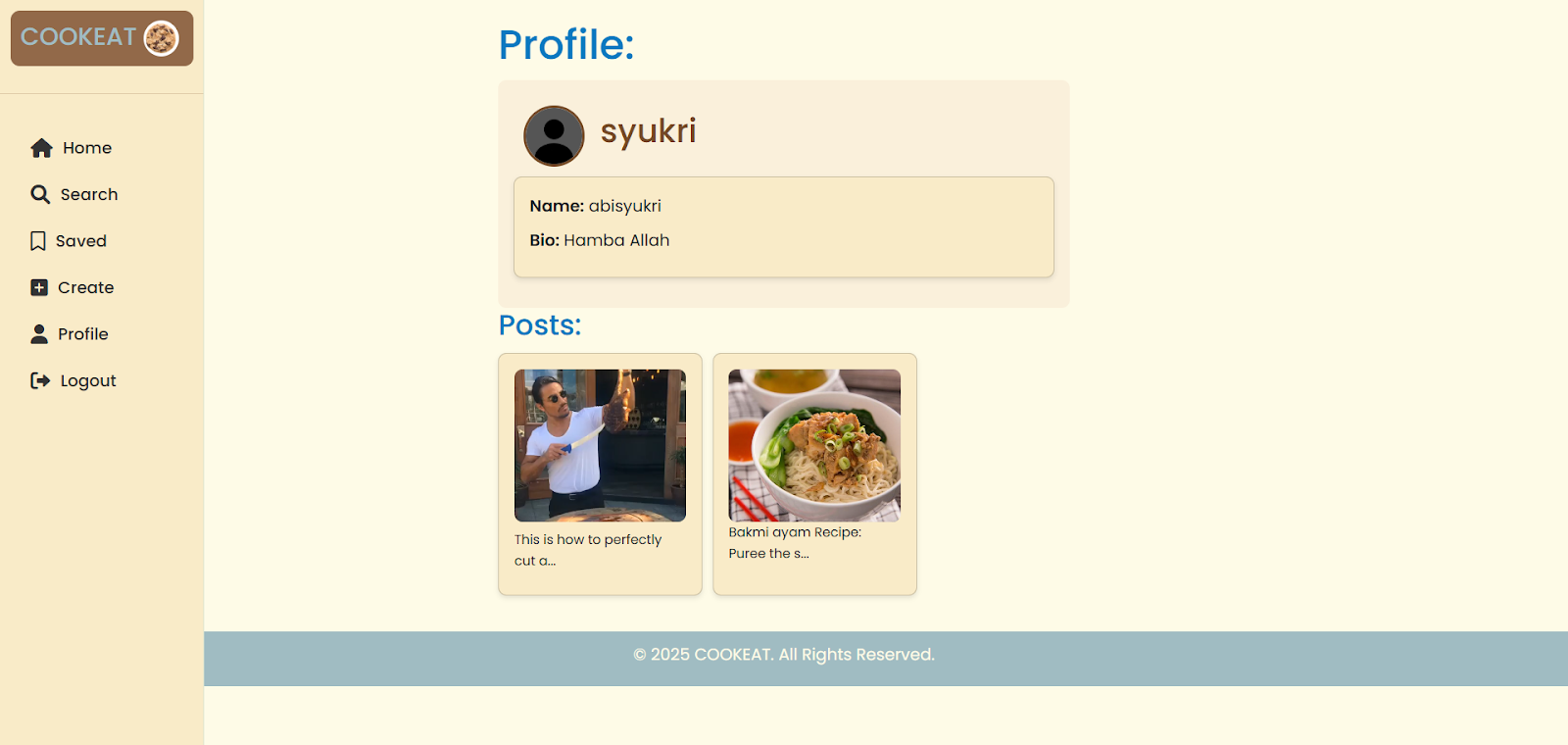
After signing up, the page then goes back to the login page and asks you to login using the email and password that you entered. Once entered, click the login button and it will direct you to the dashboard.

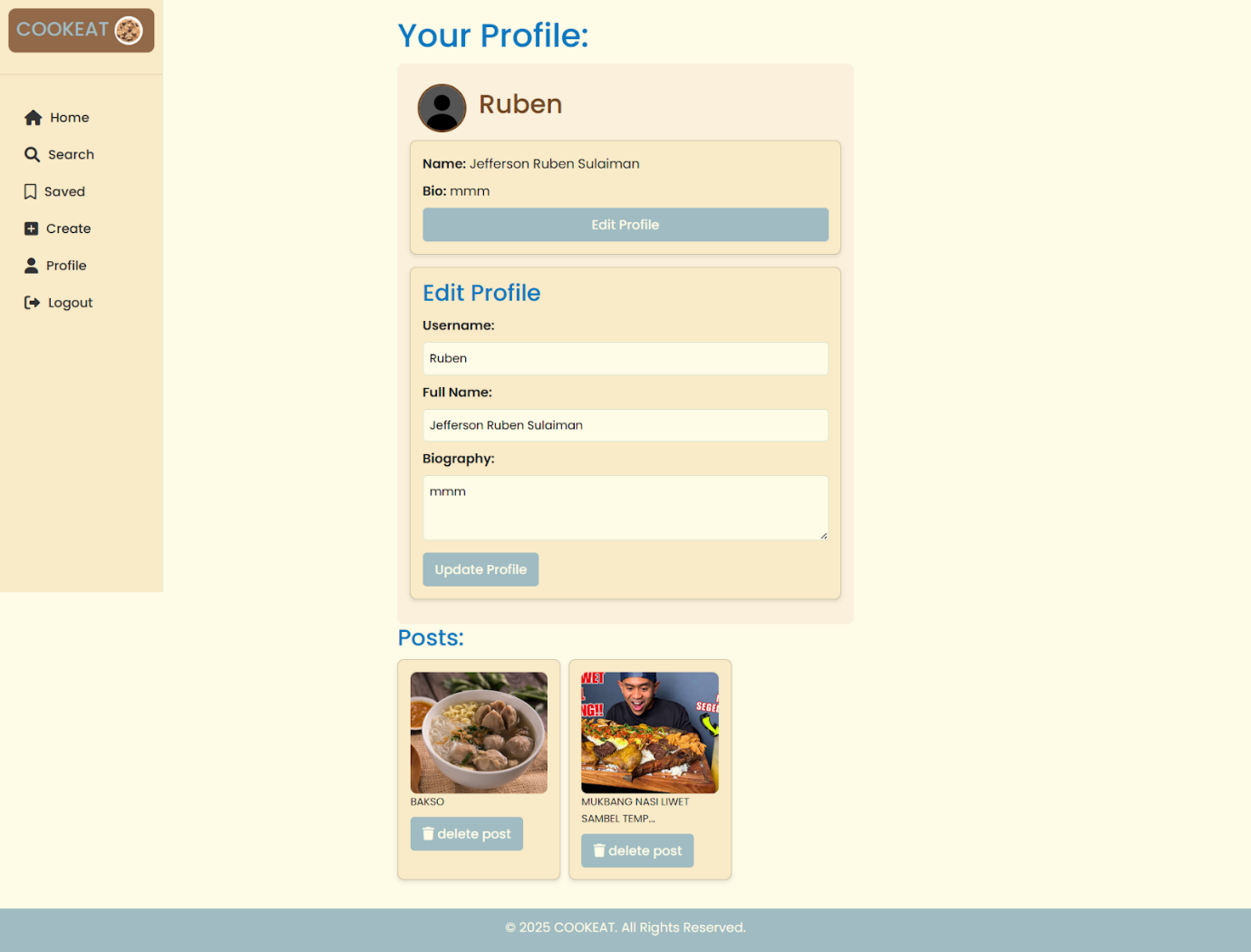


In the dashboard, users can navigate the website using the navigation bar on the left side of the screen. This includes buttons for Home, Search, Saved, Create, Profile, and Logout. The first page users will see is the Home page. The Home page displays the posts, which includes the post content itself, a comment section, a like button, and a save button. Posts can contain either a photo or a video.

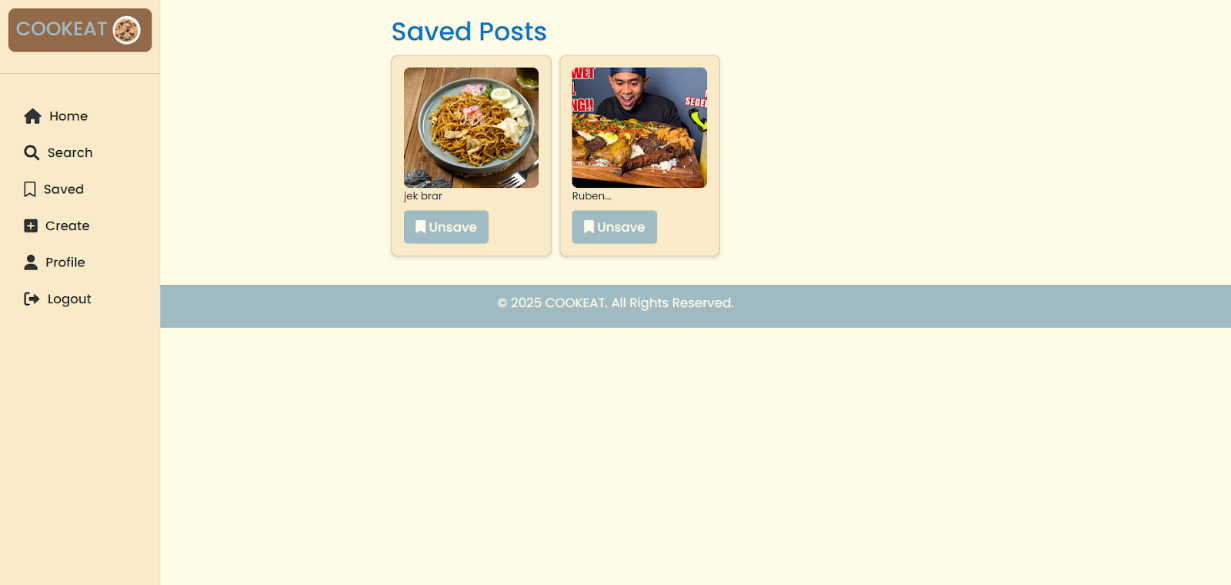
 

The search button which is located on the nav bar can be pressed to search for a specific user that is in the database. When pressed, the user is redirected to the clicked user’s profile page. the profile page displays the user’s profile picture, username, full name, biography, and their posts.

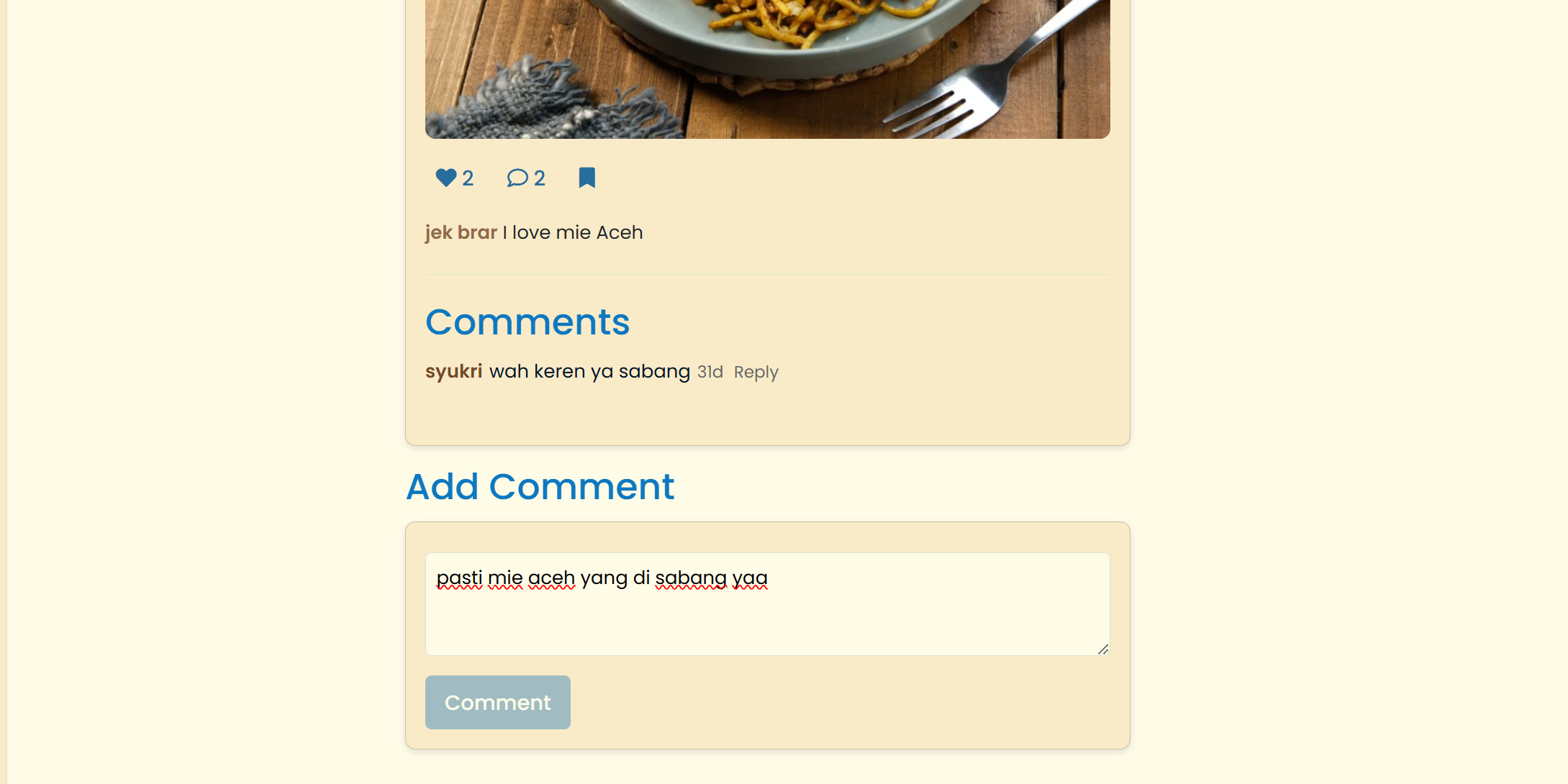




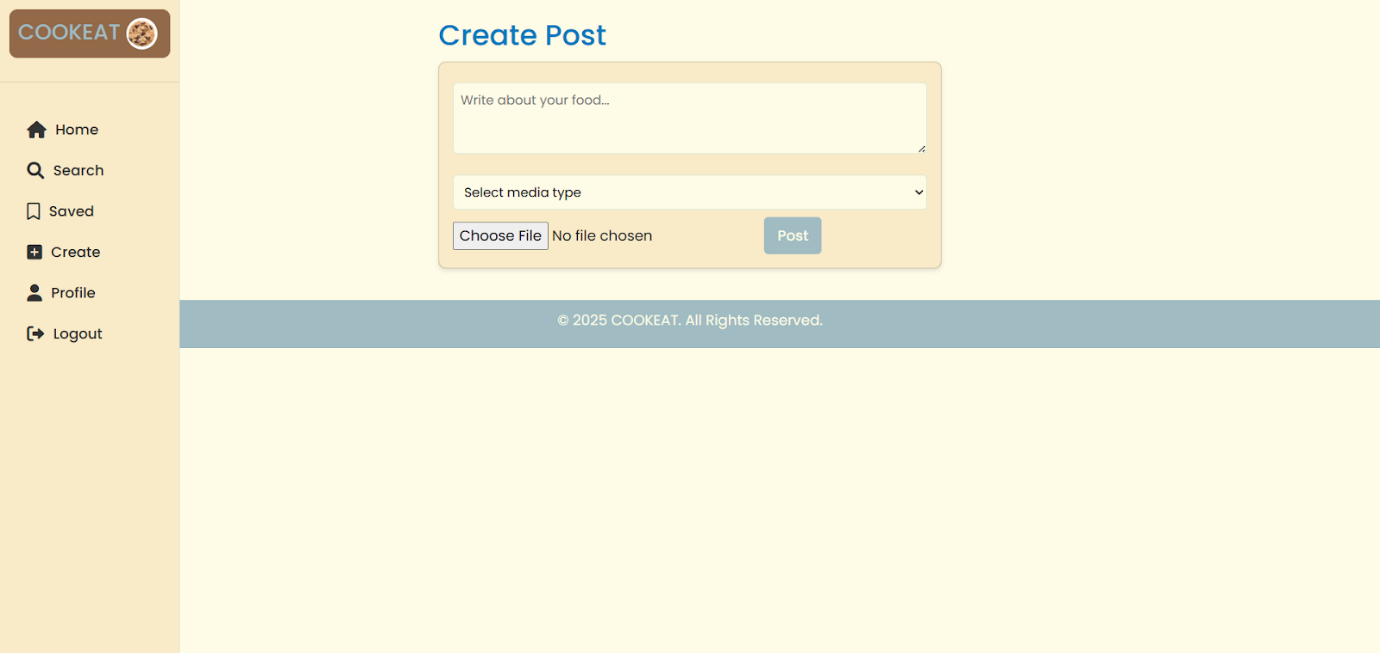
In the profile tab. It will show your name and bio, and also your posts which have a button that can delete them. There is also an edit profile button which is used to edit the username, full name, and the biography.



The saved tab is where all your bookmarked posts are located. If you ever need to check them for later. To unsave the posts, a dedicated button is available to remove the posts from the saved tab.



And we can comment on and even reply to other comments in all posts on the web.



The Posting tab lets users upload a photo or video and add a description. Once done, the post can be shared with others on the platform.

At the very bottom is the log out button which destroys the session and logs you out.