Login and Register System with **CRUD (Create, Read, Update, Delete)** operations for products, specifically for logged-in users. We'll use **JSON files** to store user data (including id, username, email, and password) and product information (including IDs, names, prices, and quantities).

Here's a step-by-step description of how to build this project using **PHP**:

1. **User Registration and Login**:

    - Create a registration form where users can sign up by providing their **username**, **email**, and **password and also make a password confirmation**.
    - Validate the input data (e.g., check if the email is valid, ensure the password meets security requirements).
    - Upon successful registration, store the user data in a **JSON file** (e.g., `users.json`).
    - Create a login form with fields for **username** and **password**.
    - Validate the login credentials against the data in the `users.json` file.
    - If the login is successful, allow the user to access the system.
    - **When user login in the system** give him the ability to **update his profile** like update email username or change password also a button to **delete this his account**
2. **Product Management**:

    - Create a page (e.g., `/products`) where logged-in users can view a list of products.
    - Read product data from a **JSON file** (e.g., `products.json`).
    - Display the product details (ID, name, price, quantity) in a table.
3. **CRUD Operations on Products**:

    - **Create (Add Product)**:
        - Provide a form (e.g., `/products/add`) to add a new product.
        - Validate the input data (e.g., ensure the ID is unique).
        - Append the new product data to the existing data in the `products.json` file.
    - **Read (View Product List)**:
        - Display the list of all products from the `products.json` file.
    - **Update (Edit Product)**:
        - Allow users to edit existing product details (e.g., `/products/edit/:id`).
        - Retrieve the product data based on the specified `:id`.
        - Update the product information in the `products.json` file.
    - **Delete (Remove Product)**:
        - Provide a button to delete a product.
        - Remove the product entry from the `products.json` file.

4. **JSON File Handling**:

   - Read and write data to/from the JSON files using PHP functions (e.g., `file_get_contents`, `file_put_contents`).
   - Ensure proper error handling when reading/writing data.

5. **Security Considerations**:

   - Hash user passwords before storing them in the `users.json` file (use functions like `password_hash`).
   - Implement session management to keep users authenticated during their session.

6. **Front-End Design**:

   - Create user-friendly forms and pages using HTML, CSS, and PHP.
   - Use Bootstrap or other CSS frameworks for styling.

Ahmed Dabour
Back End Engineer