

NAME: Navelgas, Quenzzy j.

SECTION: UCOS 3-2

ONE TO MANY CRUD IN A SINGLE PAGE

ACTIVITY.

From our last lecture, I demonstrated how we can create a Create Read Update Delete (CRUD) application with one to many relationship **IN A SINGLE PAGE**. For one post, we can have many comments. Your goal for this activity is to do the same.

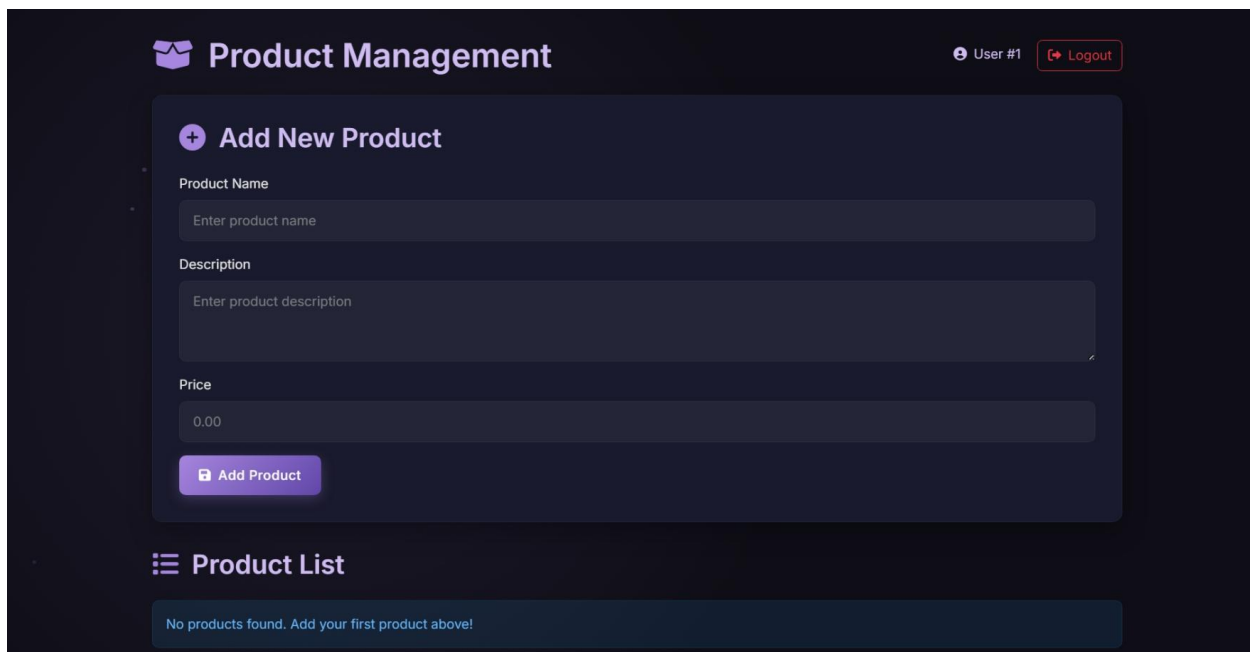
Think first of your ideal one to many relationship (one post having many comments, for example) and write your answer below the question. Please keep in mind that we should be able to identify the user **who inserted the record** and **who updated** it the last, so in that case, PHP should be involved now.

The **CRUD ASPECT** of the application should be created **IN A SINGLE PAGE**, so it's expected that we'll only have three templates in the project (**index.php, login.php, and register.php**).

WHAT'S YOUR ONE TO MANY RELATIONSHIP EXAMPLE?

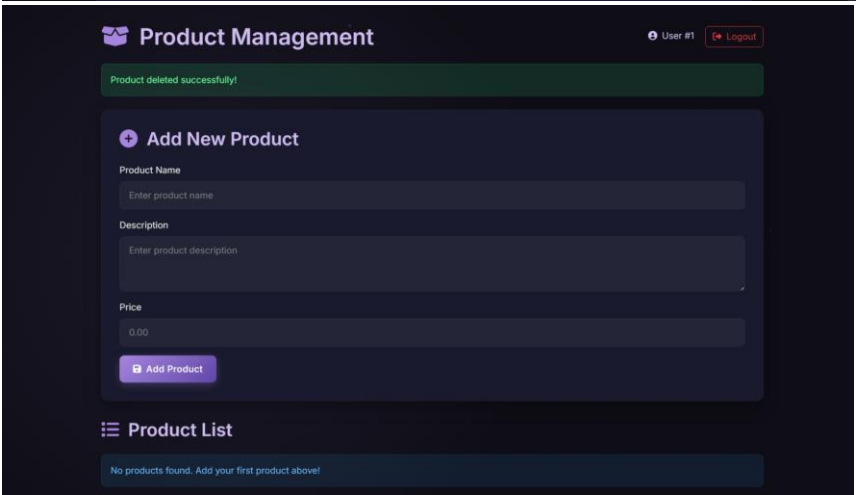
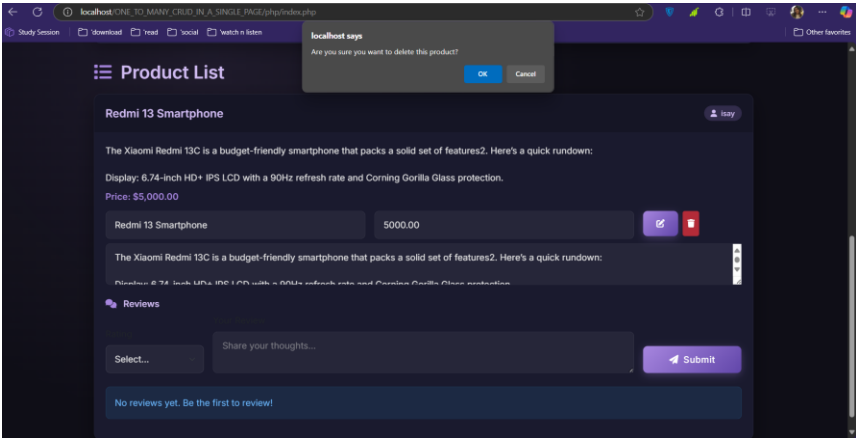
A one-to-many relationship has been established between users and products where one user has created several products through the foreign key created_by in products to one user in the users table. As such, a one product is likely to have numerous reviews given by various users as another instance of a one-to-many relationship has been observed for products to reviews.

HOMEPAGE OF YOUR APPLICATION

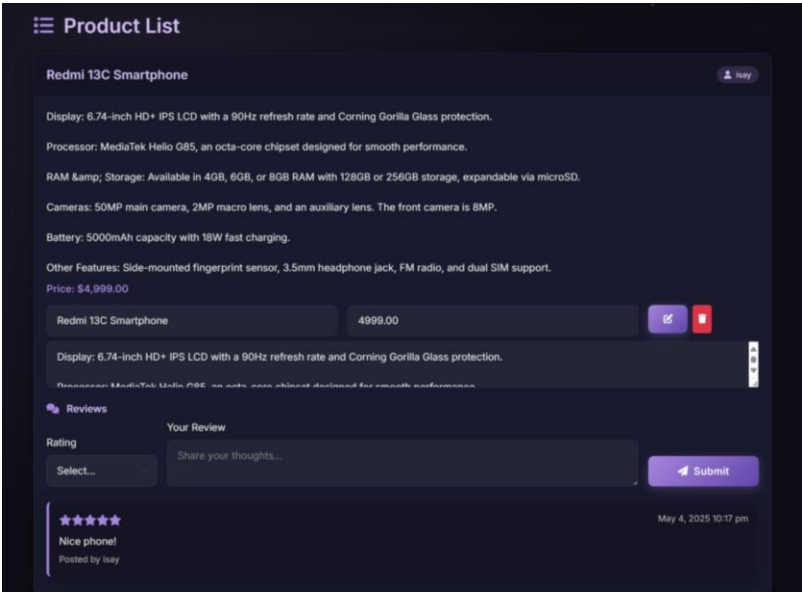


The screenshot shows a web application titled "Product Management" with a dark theme. In the top right corner, it displays "User #1" and a "Logout" button. The main content area is divided into two sections. The first section, "Add New Product", features a form with three input fields: "Product Name" (placeholder: "Enter product name"), "Description" (placeholder: "Enter product description"), and "Price" (placeholder: "0.00"). Below these fields is a purple "Add Product" button. The second section, "Product List", shows a message: "No products found. Add your first product above!".

DELETING A RECORD ON THE APPLICATION



WHO ADDED IT?



Product List

Redmi 13C Smartphone

Quenzzy

Display: 6.74-inch HD+ IPS LCD with a 90Hz refresh rate and Corning Gorilla Glass protection.

Processor: MediaTek Helio G85, an octa-core chipset designed for smooth performance.

RAM & Storage: Available in 4GB, 6GB, or 8GB RAM with 128GB or 256GB storage, expandable via microSD.

Price: \$7,999.00

Redmi 13C Smartphone

7999.00

Display: 6.74-inch HD+ IPS LCD with a 90Hz refresh rate and Corning Gorilla Glass protection.

Processor: MediaTek Helio G85, an octa-core chipset designed for smooth performance.

Reviews

Rating

Select...

Your Review

Share your thoughts...

Submit

★★★★☆

Nice phone :))

Posted by Quenzzy

May 4, 2025 10:19 pm

WHO UPDATED IT?

Redmi 13C Smartphone

Isay

Display: 6.74-inch HD+ IPS LCD with a 90Hz refresh rate and Corning Gorilla Glass protection.

Processor: MediaTek Helio G85, an octa-core chipset designed for smooth performance.

RAM & Storage: Available in 4GB, 6GB, or 8GB RAM with 128GB or 256GB storage, expandable via microSD.

Cameras: 50MP main camera, 2MP macro lens, and an auxiliary lens. The front camera is 8MP.

Battery: 5000mAh capacity with 18W fast charging.

Other Features: Side-mounted fingerprint sensor, 3.5mm headphone jack, FM radio, and dual SIM support.

Price: \$4,999.00

Reviews

Rating

Select...

Your Review

Share your thoughts...

Submit

★★★★☆

Must buy!

Posted by Quenzzy

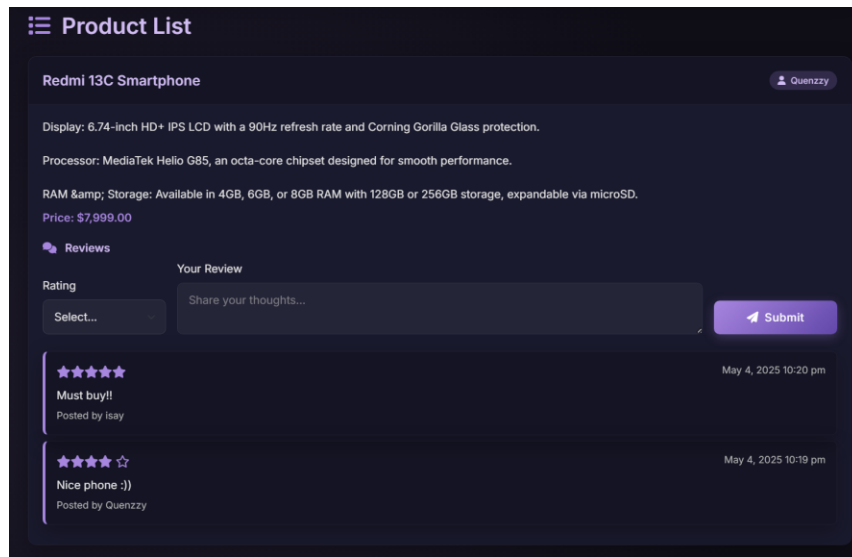
May 4, 2025 10:20 pm

★★★★★

Nice phone!

Posted by Isay

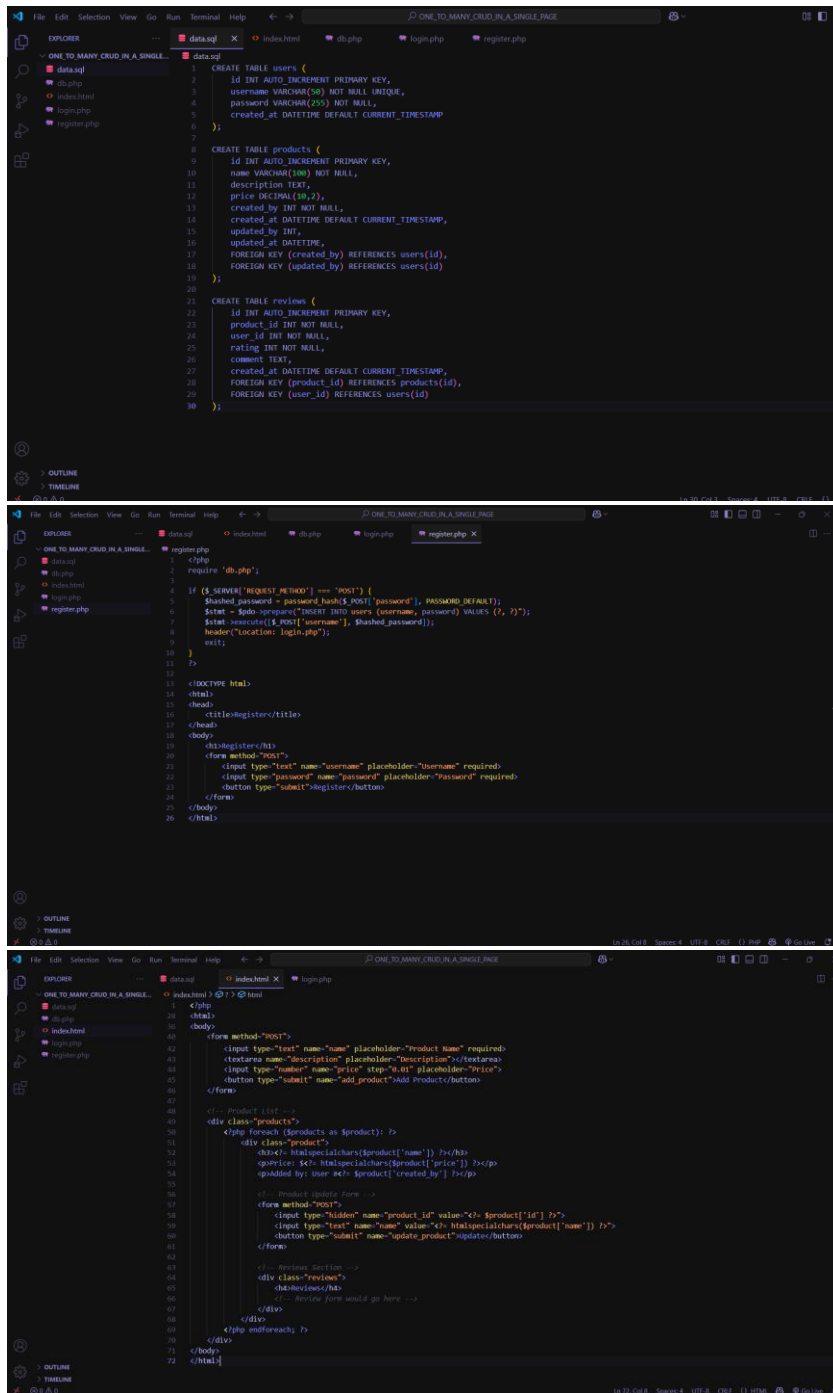
May 4, 2025 10:17 pm



GITHUB REPO LINK:

[isayasi/ONE TO MANY CRUD IN A SINGLE PAGE](https://github.com/isayasi/ONE_TO_MANY_CRUD_IN_A_SINGLE_PAGE)

PROGRESS REPORT (May 3, 2025)



The image displays three sequential screenshots of a code editor, likely Visual Studio Code, showing the development of a web application. The editor's interface includes a sidebar with an Explorer pane, a main code editor, and a bottom status bar.

Top Screenshot: The Explorer pane shows a project named "ONE-TO-MANY CRUD IN A SINGLE PAGE" with files: `data.sql`, `db.php`, `index.html`, `login.php`, and `register.php`. The `data.sql` file is open, showing SQL code for creating three tables: `users`, `products`, and `reviews`. The `users` table has columns `id` (auto-increment primary key), `username` (VARCHAR(50), unique), `password` (VARCHAR(255), not null), and `created_at` (DATETIME, default current timestamp). The `products` table has columns `id` (auto-increment primary key), `name` (VARCHAR(100), not null), `description` (TEXT), `price` (DECIMAL(10,2)), `created_by` (INT, not null), `created_at` (DATETIME, default current timestamp), `updated_by` (INT), `updated_at` (DATETIME), and foreign keys to `users` (`created_by` and `updated_by`). The `reviews` table has columns `id` (auto-increment primary key), `product_id` (INT, not null), `user_id` (INT, not null), `rating` (INT, not null), `comment` (TEXT), `created_at` (DATETIME, default current timestamp), and foreign keys to `products` (`product_id`) and `users` (`user_id`).

Middle Screenshot: The `register.php` file is open. It shows PHP code for handling a registration request. It includes a session start, a prepared statement to insert a new user into the `users` table, and a header redirect to `login.php` upon successful registration. Below the PHP code is an HTML form with fields for `username` and `password`, and a `register` button.

Bottom Screenshot: The `index.html` file is open. It shows an HTML page with a `products` section and a `reviews` section. The `products` section uses PHP to loop through a list of products, displaying their name, price, and a form to add or update them. The `reviews` section is currently empty, with a comment indicating where a review form would go.