AMA UNIVERSITY

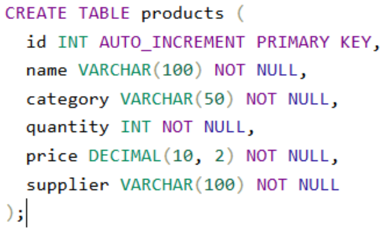
College of Computer Studies
Web App Development 1

Name	Allen Jheru Santiago
USN	22013022410
Course Code	IT6314

SIMPLE INVENTORY TRACKER

Project Overview:
The "Sari-Sari Store Inventory Tracker" is a web-based CRUD application developed to help a small sari-sari store owner manage their product inventory. The goal was to digitize the tracking of items, including categories, quantity in stock, prices, and suppliers. The system provides a visual alert when inventory levels are low and allows the store owner to add, update, and delete product records through an intuitive interface.

Database Schema: SQL used to create your tables



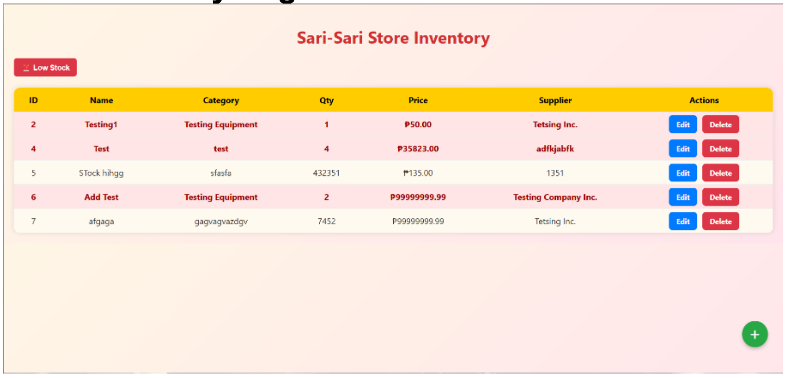
```
CREATE TABLE products (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  category VARCHAR(50) NOT NULL,
  quantity INT NOT NULL,
  price DECIMAL(10, 2) NOT NULL,
  supplier VARCHAR(100) NOT NULL
);
```

CRUD Feature Walkthrough: Describe how each operation (Create, Read, Update, Delete) was implemented

- **Create (Add Product)**
  1. Implemented in add.php
  2. HTML form collects product details.
  3. PHP uses prepared statements with named parameters to insert into the products table.
- **Read (View Inventory)**
  1. Implemented in index.php
  2. Uses a SELECT query to fetch all product rows from the database.
  3. Results are displayed in a styled HTML table with dynamic quantity alerts.
  4. Includes a modal triggered by a button to show products with quantity less than 5.
- **Update (Edit Product)**
  1. Implemented in edit.php
  2. Product ID is passed via URL, record is fetched in a form.
  3. On submission, an UPDATE query is executed using named parameters.
- **Delete (Remove Product)**
  1. Implemented in delete.php
  2. Product ID is passed via URL and deleted using a prepared DELETE statement.
  3. A confirmation prompt is triggered before deletion.

Screenshots:

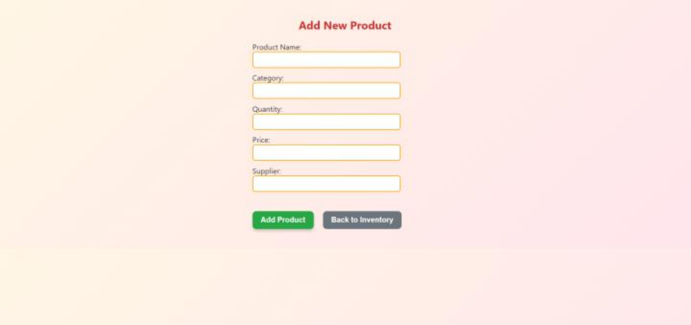
1. Main Inventory Page:



Sari-Sari Store Inventory

ID	Name	Category	Qty	Price	Supplier	Actions
2	Testing1	Testing Equipment	1	₱50.00	Tetsing Inc.	Edit Delete
4	Test	test	4	₱35823.00	adfkjabfk	Edit Delete
5	STock hihgg	sfasfa	432351	₱135.00	1351	Edit Delete
6	Add Test	Testing Equipment	2	₱99999999.99	Testing Company Inc.	Edit Delete
7	afgaga	gagvagvazdgv	7452	₱99999999.99	Tetsing Inc.	Edit Delete

2. Add Product Form:



Add New Product

Product Name:

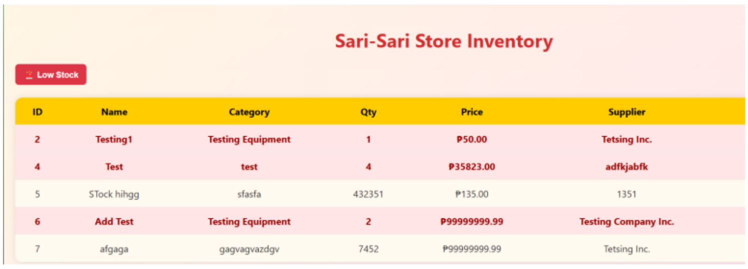
Category:

Quantity:

Price:

Supplier:

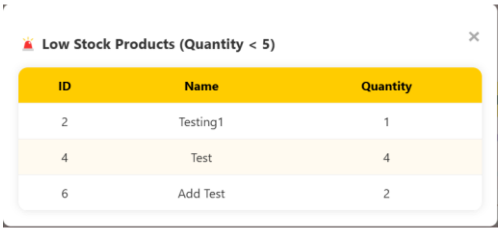
3. Low Stock Alert:



Sari-Sari Store Inventory

ID	Name	Category	Qty	Price	Supplier
2	Testing1	Testing Equipment	1	₱50.00	Tetsing Inc.
4	Test	test	4	₱35823.00	adfkjabfk
5	STock hihgg	sfasfa	432351	₱135.00	1351
6	Add Test	Testing Equipment	2	₱99999999.99	Testing Company Inc.
7	afgaga	gagvagvazdgv	7452	₱99999999.99	Tetsing Inc.

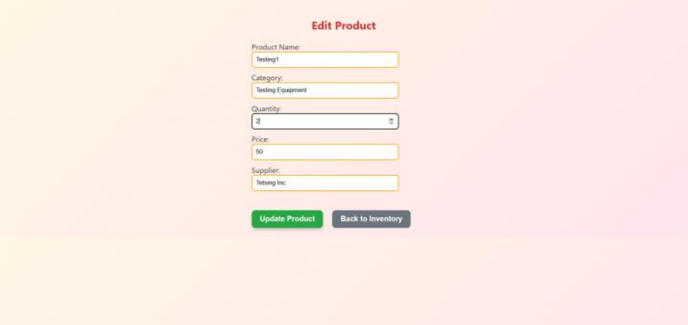
4. Low Stock Modal Form:



Low Stock Products (Quantity < 5)

ID	Name	Quantity
2	Testing1	1
4	Test	4
6	Add Test	2

5. Edit Product Form:



Edit Product

Product Name: Testing1

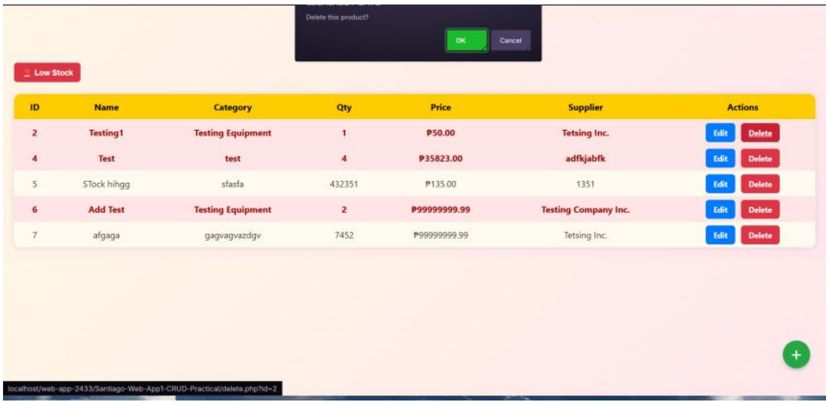
Category: Testing Equipment

Quantity: 2

Price: 50

Supplier: Tetsing Inc.

6. **Delete Product Form:**



ID	Name	Category	Qty	Price	Supplier	Actions
2	Testing1	Testing Equipment	1	₱50.00	Tetsing Inc.	Edit Delete
4	Test	test	4	₱35823.00	adfkjabfk	Edit Delete
5	STock hihgg	sfasfa	432351	₱135.00	1351	Edit Delete
6	Add Test	Testing Equipment	2	₱99999999.99	Testing Company Inc.	Edit Delete
7	afgaga	gagvagvazdgv	7452	₱99999999.99	Tetsing Inc.	Edit Delete

localhost/web-app-2433/Santiago-Web-App1-CRUD-Practical/delete.php?id=2

Challenges Encountered and How You Solved Them:

- **CSS Not Applying:** The style.css file link was incorrectly placed or misnamed. This was fixed by verifying the <link> tag and folder structure.
- **JavaScript Alert Not Working:** The script.js file was not being loaded due to incorrect file linking. This was corrected, and DOMContentLoaded was used to ensure DOM readiness.
- **XSS Security:** htmlspecialchars() was used in all echo statements to prevent injection of malicious HTML/JS code.
- **Database Field Mismatches:** Ensured consistent renaming (e.g., product_name to name) across all PHP files.

Learnings/Reflections:

Through this project, I learned how to implement full CRUD functionality using PHP and MySQL with security best practices like prepared statements and output escaping. I also learned how to dynamically manipulate the DOM using JavaScript, particularly for UI feedback like quantity alerts and modals. One major takeaway is the importance of debugging and tracing file linkage issues early on, as even minor path errors can prevent entire features from working. One major takeaway is the importance of debugging and tracing file linkage issues early on, as even minor path errors can prevent entire features from working.