



De La Salle University
2401 Taft Avenue, Manila Philippines 1004
Gokongwei College of Engineering
1st Trimester, AY 2024-2025

StarVault Inventory Management System

Submitted to:

Engr. Ronnel Agulto

Date Submitted:

Dec 5, 2024

Submitted by:

Eleydo, Russell Ian E.
Gatchalian, Richmond
Laher, Toby Andrei

I. Abstract

The Starvault Inventory Management System addresses the challenges of managing inventory effectively in a business environment. Traditional inventory systems often lack the scalability, security, and user-friendliness required to meet the needs of modern organizations. This project aims to bridge that gap by providing a comprehensive, web-based solution tailored for efficient inventory tracking and management. The system leverages a user-friendly interface, built using HTML, CSS, and JavaScript, to ensure seamless navigation and ease of use. Its back end, developed in PHP, processes user inputs, manages requests, and securely interacts with a MySQL database. This database stores critical data, including user credentials, product details, and supplier information, enabling a streamlined workflow. The integration of secure practices such as password hashing and SQL prepared statements ensures data integrity and user security. The primary objective of Starvault is to simplify inventory operations for businesses by providing a scalable and reliable system. The solution is designed to enhance productivity by reducing manual effort, minimizing errors, and providing real-time insights into inventory status. By incorporating modular components and robust database design, the application not only solves the immediate problem of inventory management but also offers the scalability required for future business growth. This project demonstrates the potential of technology-driven solutions in addressing operational inefficiencies and driving business success.

II. Introduction

Problem Statement / Objective

In today's fast-paced business environment, managing inventory efficiently is a significant challenge, particularly for small and medium-sized enterprises. Traditional inventory management methods, such as manual tracking or spreadsheets, are prone to human error, lack scalability, and fail to provide real-time data access. These inefficiencies often lead to overstocking, stockouts, or mismanagement, adversely impacting business operations and profitability.

The **Starvault Inventory Management System** is a web-based solution designed to streamline inventory operations. By integrating a user-friendly interface, secure back-end logic, and robust database management, the system ensures accurate tracking, simplified workflows, and enhanced productivity. It offers features such as supplier and product management, user authentication, and real-time updates, making it a comprehensive tool for addressing inventory-related challenges.

Target Audience

The primary target users of this application are **business owners**, particularly those managing small to medium enterprises. Additionally, **inventory managers** and operational staff can benefit from the system's user-friendly and efficient design. This solution is also ideal for companies aiming to transition from manual inventory processes to a digital platform.

Type of Application

The **Starvault Inventory Management System** is a **web application** accessible through standard web browsers. It is designed to function on local and hosted servers, ensuring flexibility in deployment and ease of access for users across devices.

III. System Architecture

The project uses html, css, javascript, bootstrap for frontend development, while php and mysql is used for backend development. In the program, the frontend and backend interact through php because it connects the functionalities of the frontend to the existing database.

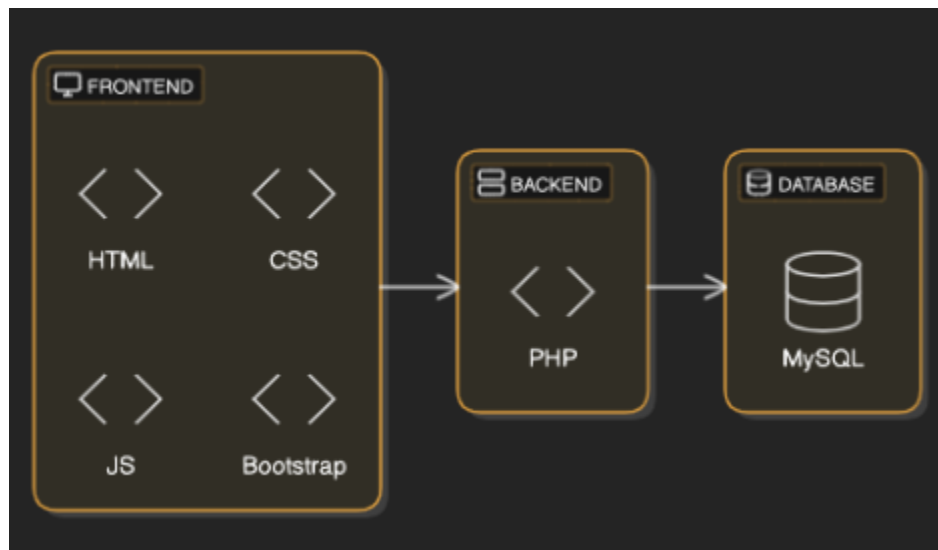


Diagram 1.

The *Diagram 1* shows how the frontend, backend, and database work together. The frontend uses HTML, CSS, JavaScript, and Bootstrap to create the user interface where users can interact. It sends user requests to the backend, built with PHP, which processes the requests and handles the logic. Then, the backend connects to the database, powered by MySQL, to store, retrieve, or update data. Once the backend gets the needed data, it sends the results back to the frontend for the features' functionalities.

IV. Methodology

A. Features & Functionality:

a. Registration System

The registration page allows new users to sign up and create an account. This ensures that only authorized users can access the system's features.

b. Login System

The login page is used for user authentication. Existing users must enter their correct credentials (username and password) to access the main dashboard securely.

c. Logout System

The logout functionality securely logs users out of the system and redirects them to the registration or login page, ensuring their session is properly terminated.

d. Main Dashboard

The main dashboard is where the users can navigate the system's features freely. It includes functionalities like minimizing the sidebar for better screen utilization and displaying the logged-in user's name for personalization.

e. Data Analytics

Data analytics provide insights into the system's information. For instance, the dashboard visualizes product counts for each supplier using graphs, offering a clear and concise overview of inventory and supplier performance.

f. Products Page

The products page enables users to manage their inventory effectively. Users can add new products with details such as an image, product name, supplier, and additional attributes. The page also includes a product viewing feature for tracking and managing inventory with ease.

g. Suppliers Page

The suppliers page allows users to manage supplier information. Users can add new suppliers to the system to keep track of products sourced from them. A "View Supplier" feature helps users efficiently manage supplier-related data and inventory.

B. Database Design:

- a. The program uses MySQL database with the main database named “inventorymanagement” it has the following tables and values:

i. Users

- id INT AUTO_INCREMENT PRIMARY KEY
- username VARCHAR(255)
- password VARCHAR(255)
- email VARCHAR(100)

ii. Suppliers

- id INT AUTO_INCREMENT PRIMARY KEY
- supplier_name VARCHAR(255)
- supplier_location VARCHAR(255)
- email VARCHAR(255)
- created_by INT
- created_at DATETIME
- updated_at DATETIME
- FOREIGN KEY (created_by) REFERENCES Users(id)

iii. Products

- id INT AUTO_INCREMENT PRIMARY KEY
- product_name VARCHAR(255)
- description VARCHAR(255)
- image VARCHAR(255)
- created_by INT
- created_at DATETIME
- updated_at DATETIME
- FOREIGN KEY (created_by) REFERENCES Users(id)

iv. Products_suppliers

- id INT AUTO_INCREMENT PRIMARY KEY
- supplier INT
- product INT
- updated_at DATETIME
- created_at DATETIME
- FOREIGN KEY (supplier) REFERENCES suppliers(id);
- FOREIGN KEY (product) REFERENCES products(id);

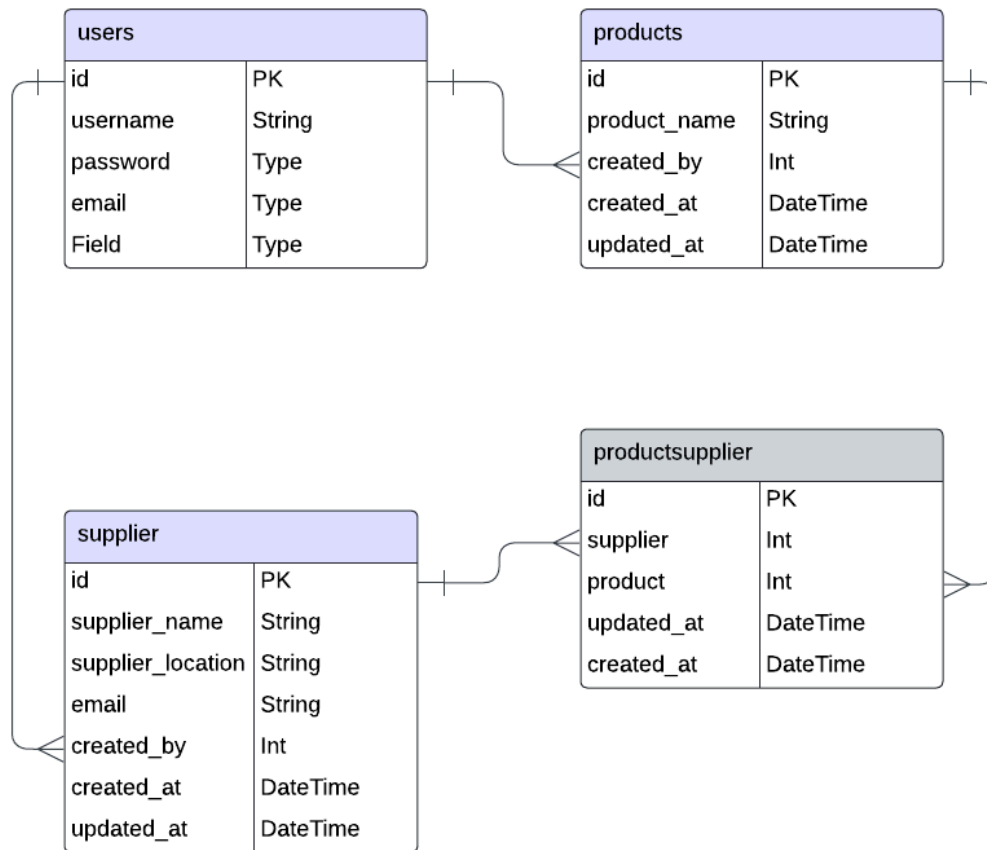


Diagram 2.

The entity-relationship diagram illustrates a system where users, products, and suppliers are interconnected. Each user can create and manage multiple products and suppliers, establishing a one-to-many relationship between the users table and both the products and supplier tables through the *created_by* attribute. Products and suppliers are connected through a many-to-many relationship, facilitated by the *productsupplier* table. This join table associates a product with one or more suppliers and a supplier with one or more products. Each entry in the *productsupplier* table links a product from the products table to a supplier from the supplier table.

V. Simulation and Results

- **System Demonstration:**

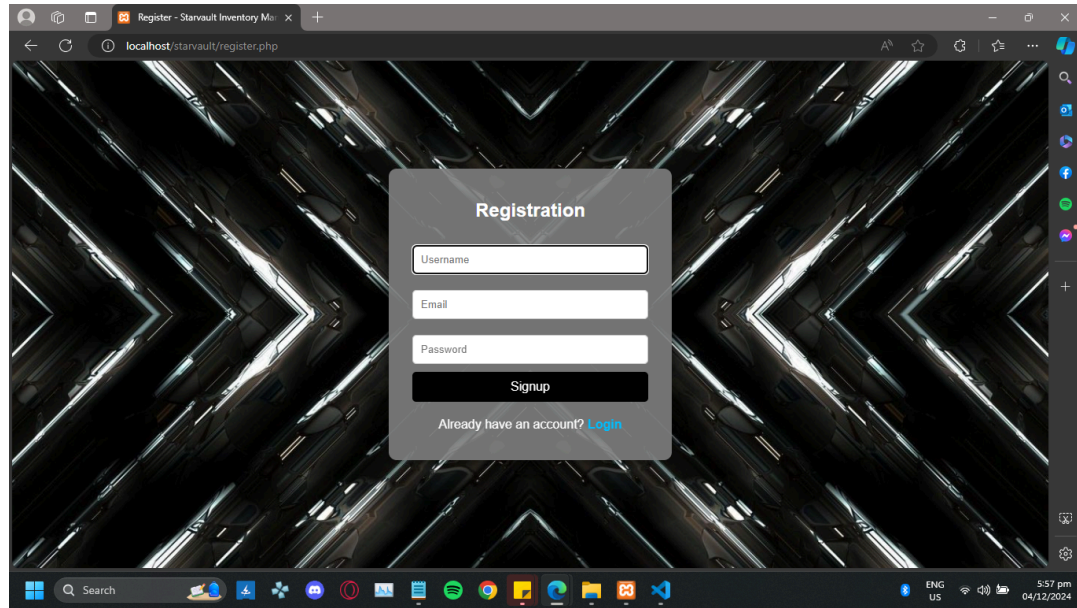


Figure 1: Registration Page

The registration page provides an intuitive form for new users to create an account. It collects essential details like username, email, and password. The form includes client-side and server-side validation to ensure data integrity. Upon successful registration, the user is redirected to the login page.

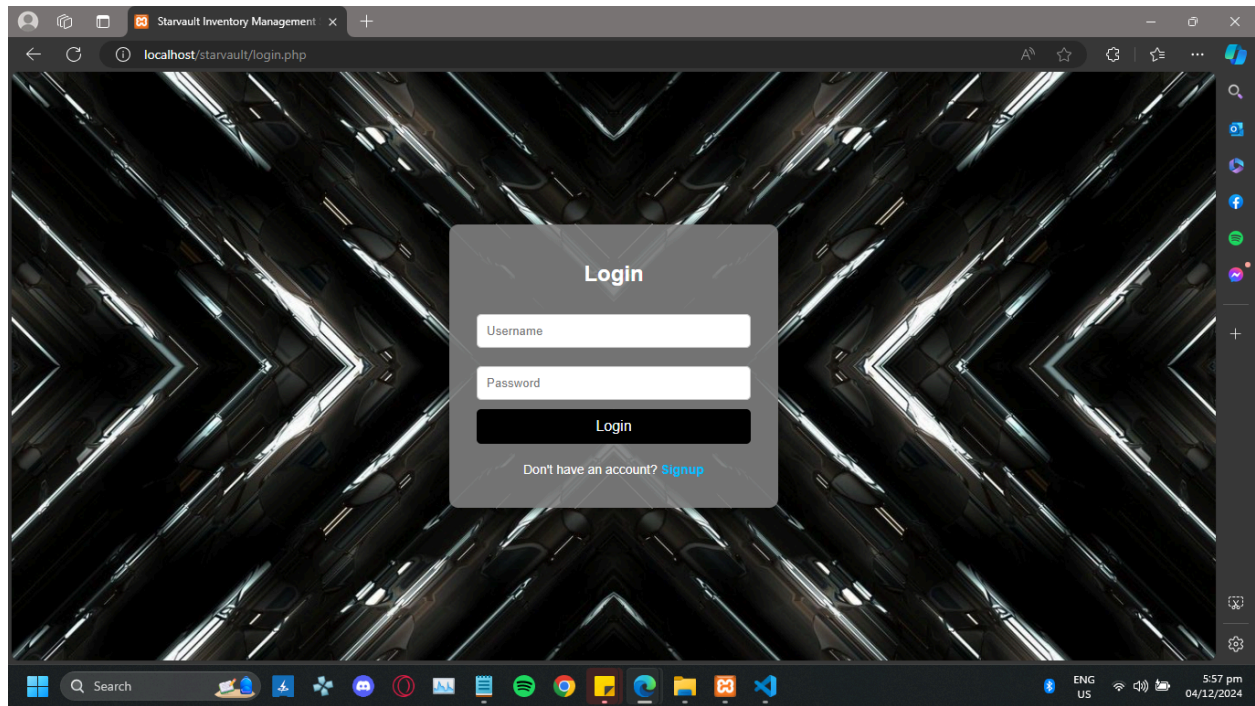


Figure 2: Login Page

The login page allows registered users to access their accounts. It features input fields for username and password and utilizes secure password hashing for authentication. Invalid credentials prompt an error message, ensuring unauthorized users are denied access.

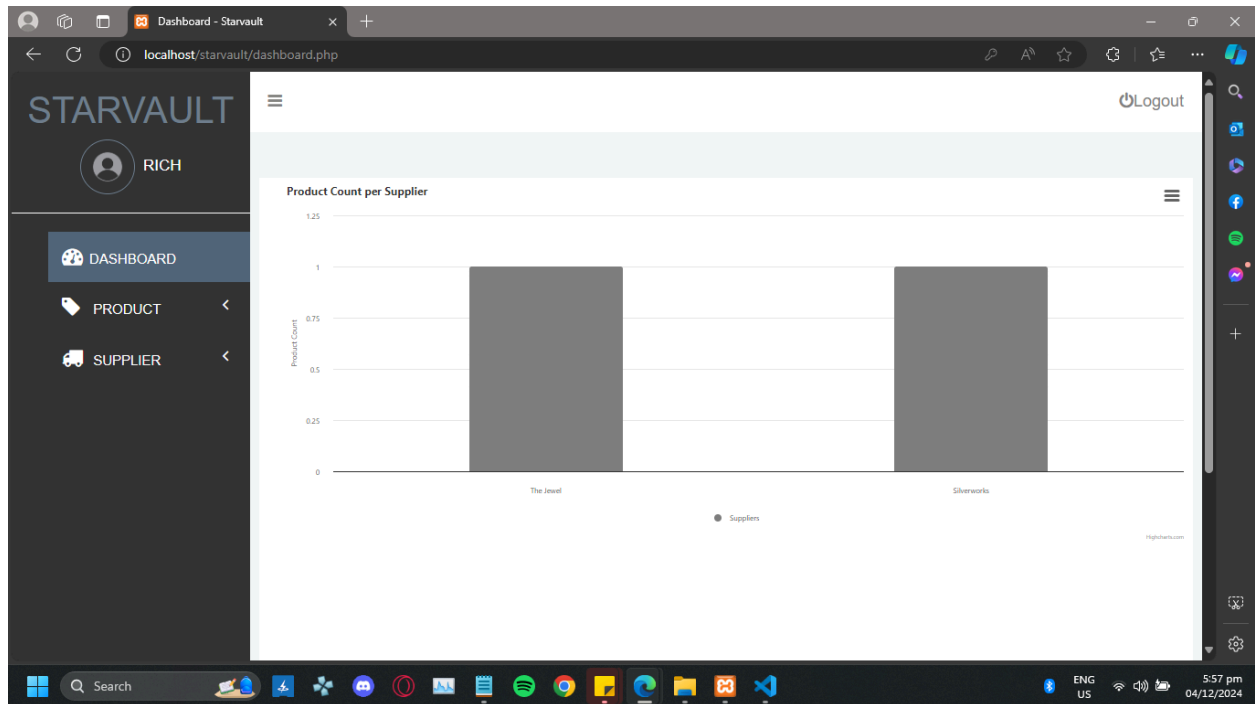


Figure 3: Dashboard Page

The dashboard serves as the central hub, offering an overview of key metrics, such as product counts per supplier. It utilizes dynamic charting to visualize data trends in real-time, enhancing decision-making capabilities. The bar chart is dynamically generated based on real-time data retrieved from the database. This ensures that users always have access to up-to-date information.

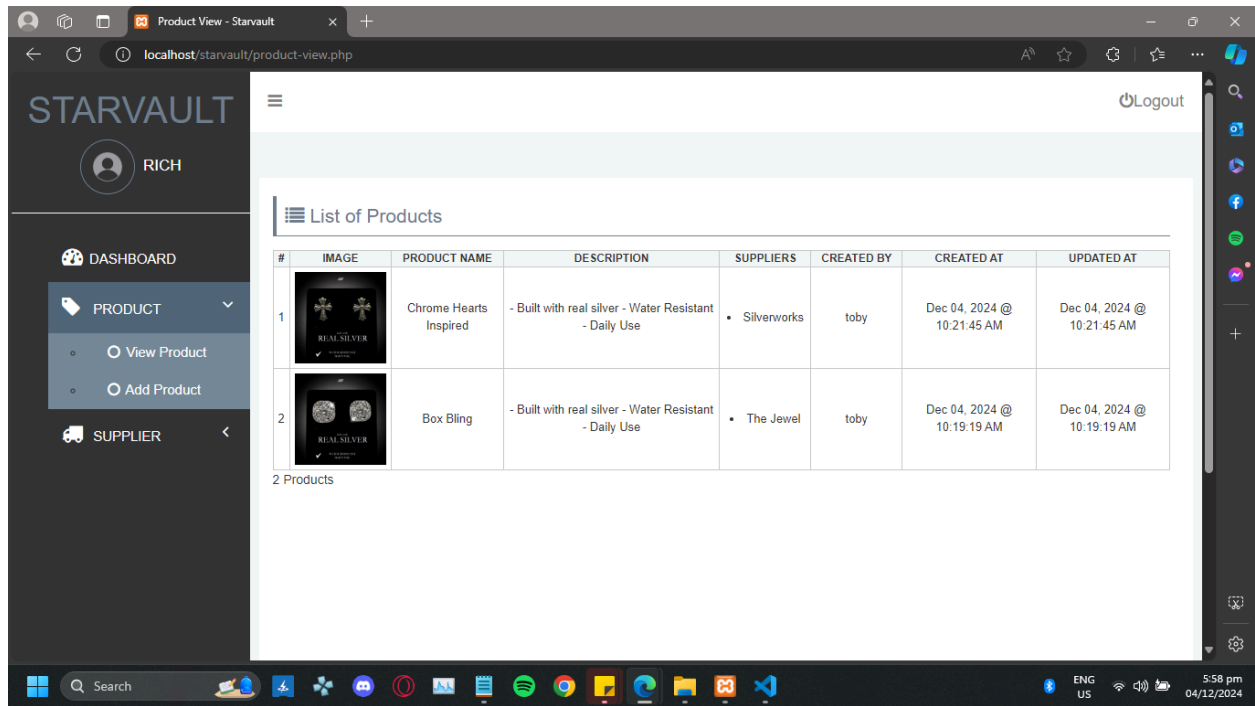


Figure 5: Product List

The product view displays a detailed table of products, including their name, description, associated suppliers, and timestamps. The interface supports pagination and sorting for efficient navigation of large datasets.

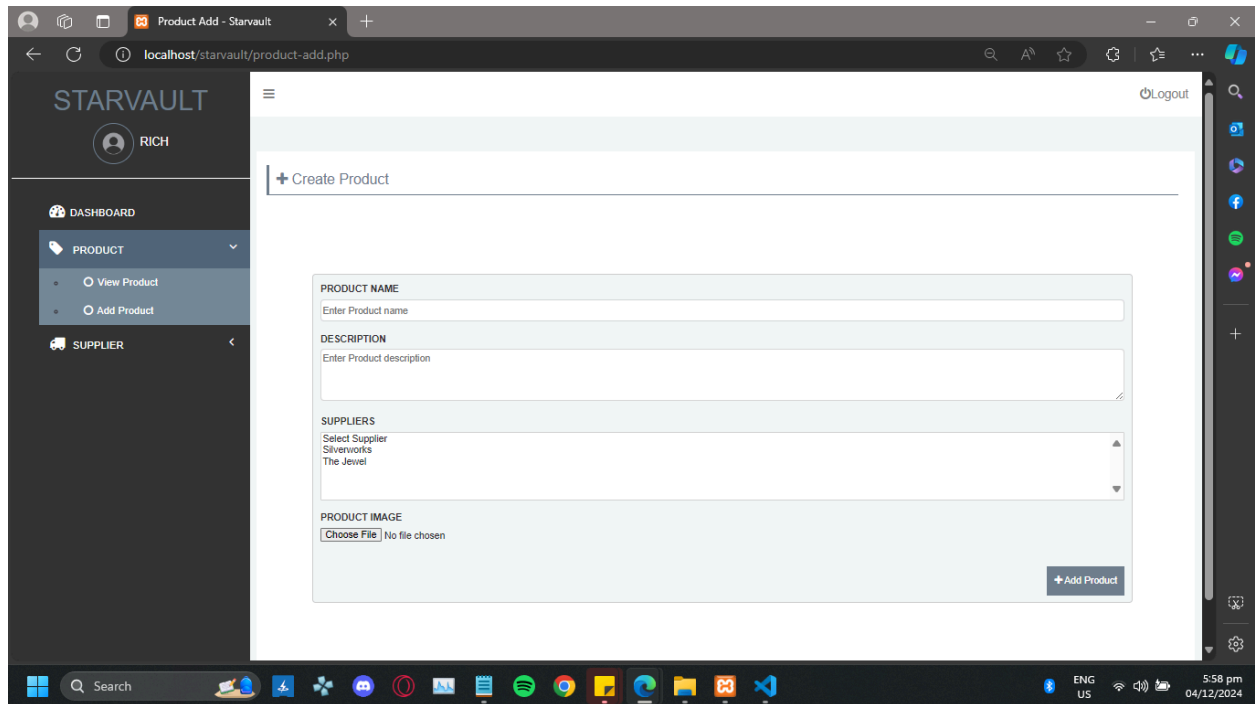


Figure 6: Add Product Page

The add product page allows users to input product details, including name, description, supplier, and image upload. The form ensures mandatory fields are completed before submission and integrates seamlessly with the database.

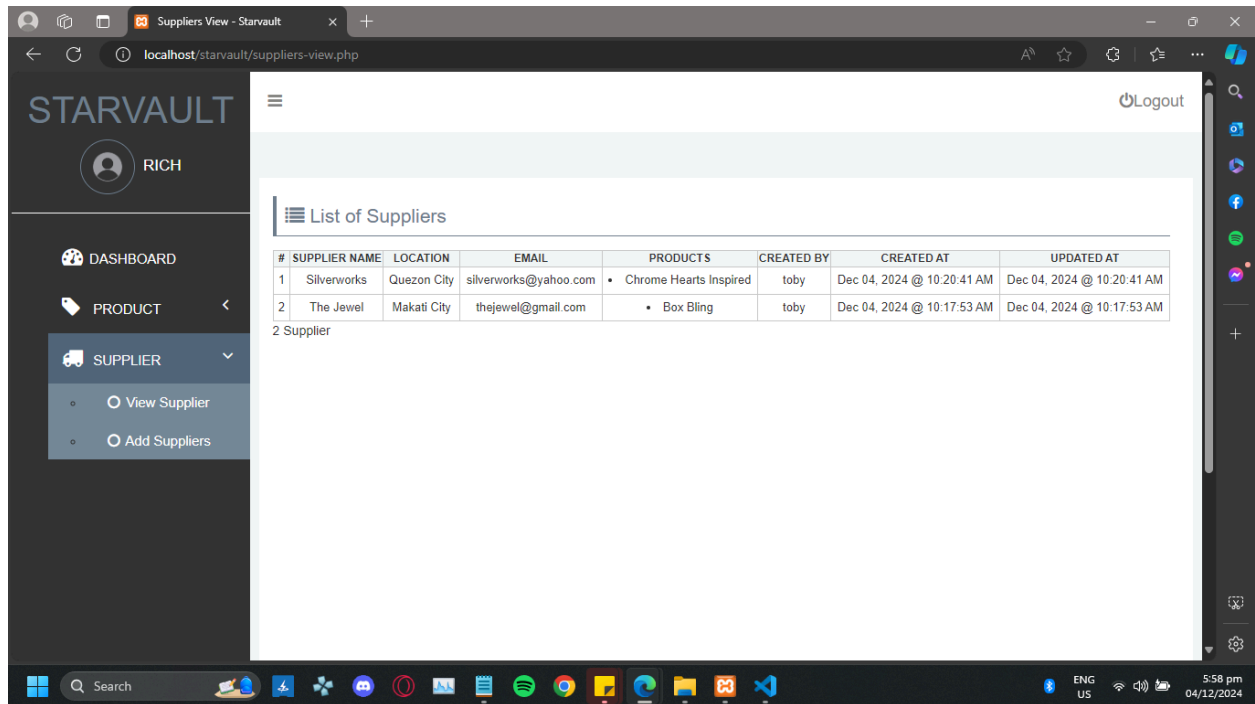


Figure 7: Supplier List

The supplier list provides a tabular view of all suppliers, displaying details like name, location, email, associated products, and timestamps. It supports real-time updates and cross-referencing with the product database.

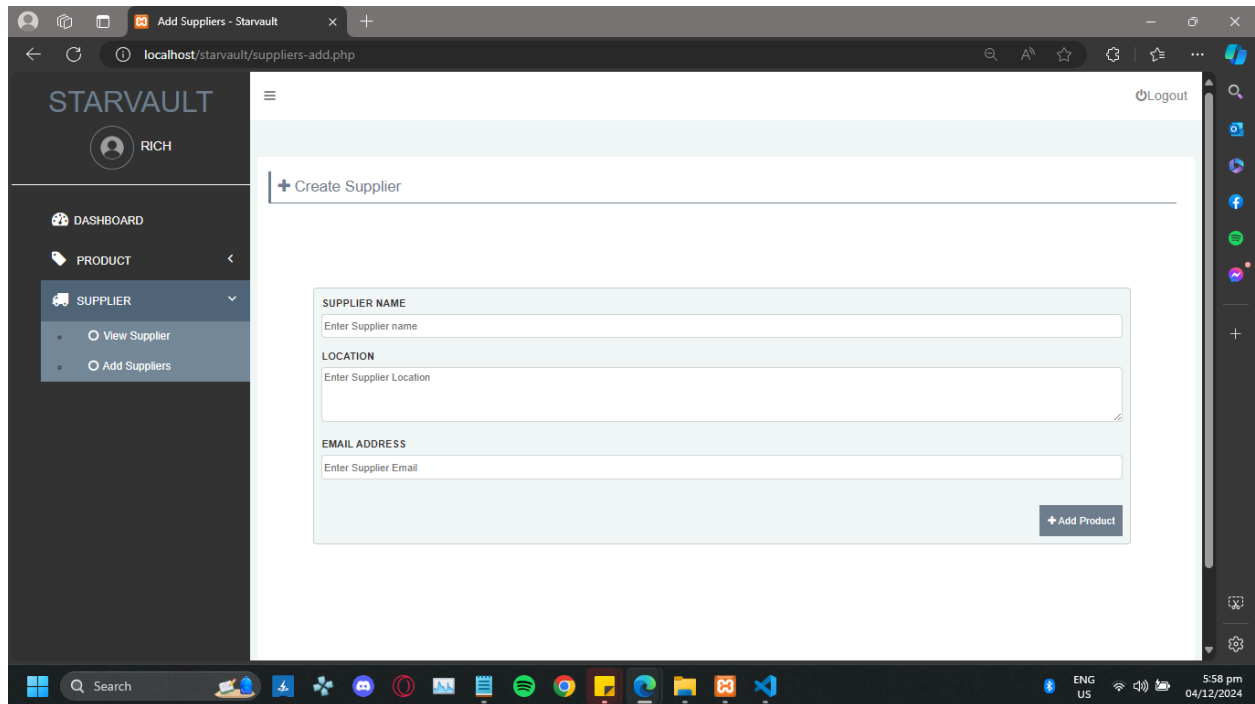


Figure 8: Supplier Page

The add supplier page enables users to input new supplier details, including name, location, and email. The form validates inputs to prevent data inconsistencies and ensures a seamless addition to the database.

Simulation Process

When implemented, the system ensures efficient management of products and suppliers. The user-friendly interface facilitates smooth navigation, while real-time data visualization enhances decision-making. Security features, such as password hashing and prepared statements, protect user data and system integrity. In simulation, the system demonstrated robust performance under various scenarios, proving its readiness for deployment in real-world environments.

VI. References

- G. Booch, "Object-Oriented Analysis and Design," 2nd ed. Redwood City, CA, USA: Benjamin-Cummings, 1994.
- J. Ullman, "Principles of Database Systems," 2nd ed. Rockville, MD, USA: Computer Science Press, 1982.
- T. Connolly and C. Begg, "Database Systems: A Practical Approach to Design, Implementation, and Management," 6th ed. Boston, MA, USA: Pearson Education, 2014.
- A. Moisseiev, "Introduction to PHP and MySQL," presented at the Web DevConf, Boston, MA, USA, Jun. 2019.