**INVENTORY MANAGEMENT SYSTEM**

****

***Submitted by***

|  |  |
| --- | --- |
| **Name of the Student** | **University Roll No.** |
| SHOUVIK SEN | 12022002003023 |
| SWAPNIL GANGULY | 12022002003020 |

Under the supervision of

Prof. Deepsubhra Guha Roy

***Academic Year: 2024-25***

**REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) OF**

**MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)**

**INSTITUTE OF ENGINEERING AND MANAGEMENT**

**KOLKATA**



*DECLARATION*

We hereby declare that the work recorded in this project report entitled ‘**Inventory Management System’** be accepted in partial fulfilment of the requirements for the degree of BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING(AIML), is a faithful work carried out by **SHOUVIK SEN** and **SWAPNIL GANGULY** under the supervision and guidance of Mr. Deepsubhra Guha Roy.

We would like to acknowledge his assistance and help received during the course of this project.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Head, CSE(AIML)Department Project Guide***

***Institute of Engineering and Management, Kolkata***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***Student Signature : Shouvik Sen Student Signature : Swapnil Ganguly***

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **NO.** | **TITLE** | **PAGE NO.** |
| 1. | Abstract | 04 |
| 2. | Introduction | 05 |
| 3. | Aim and goal of project | 06 |
| 4. | Components | 07-08 |
| 5. | Project Details | 09-22 |
| 6. | Databases | 23 |
| 7. | Conclusion | 24 |

**ABSTRACT**

This paper outlines planning, designing and realization of an Inventory Management System having been developed as part of a Database Management System lab course. The system's primary goal is assisting in the efficient management of product inventory inside a company or organization, specifically related to the structure of products, supplier management, order processing and restocking. In industries which deal with high quantity items of goods or in case of dealing with raw material, a manual maintenance of record can be inefficient besides being prone to errors. This system also prevents erroneous data, reduces the time of processing orders and also helps in getting real time stock management information. We implement a full-stack application development using MySQL as the backend of the database while using PHP for the server side and HTML/CSS for user interface. The work includes building a client-server application. The key concepts of the project includes ER modelling, entity-relationship modeling, relational database design, SQL operations, statelessness and CRUD (Create, Read, Update, Delete) functions along with retrieve and PHP REST API, specifically, in handling data for the front end for retrieval and storage. It provides developers with an application that can manage the current inventory, along with providing for feature enhancements for example checking low stock and future inventory trends. Thus, this work adds to a confirmed case in inventory management handling, providing a significant role for database management in inventory workflow management.

**Introduction**

Inventory management is the backbone of resource usage and stands at the core of maintenance of product availability and satisfaction of customer demand, and it is integral to any organization's operational framework. As an organization grows, so do the inventories and the supply chains that tend to get more complex, calling for digital solutions that can handle large volumes of data accurately and efficiently for such organizations. The IMS developed in this project addresses these challenges by offering a structured approach to tracking and updating product information, supplier details, stock quantities, and order statuses real-time.

Traditional approaches in managing inventory often rely on spreadsheets or hand records, posing inherent limitations in terms of data accuracy and accessibility as well as in updating. These may result in frustrating delays, overstocking, understocking, and even stockouts. An inventory system driven by a database, however, has automated all these tasks while providing ready access to updated information, hence enriching decision-making and allowing for proactive management of the supply chain.

The application utilizes a relational database generated with MySQL to implement and store the product and supplier data. For the server-side data processing, it uses PHP, while the front end was created using HTML and CSS. It allowed the creation of read, update, and delete product records. It displayed the current levels of inventory and requested automated restocking when these reached a critical level. This report details the stage by stage development of this IMS involving requirement analysis, database design, both front-end and back-end development, and testing. Through the integration of data management principles with a focus on user-centered interface, this project provides a thorough yet flexible solution that can easily develop as the organization may necessitate. Ultimately, it serves as a more straightforward practical implementation of core DBMS concepts and shows the real world of how such technology could simplify complex workflows on managing an inventory.

**Aim**

This project aims to design and implement an inventory management system that provides accurate records of products, suppliers, stock levels, and orders.

So that the users can easily add, update and browse the inventory details.

Simplifies monitoring restock requirements and order history.

Provides access and updation with real-time information to support inventory management in the organization.

This is to outline the practical example of database design principles and how a relational database integrates with a web-based interface to better access and usability.

**Goal of the Project**

The prime goal of this IMS is to design an electronic solution that helps optimize the process of inventory management within an organization. Particularly, this system will provide efficient, effective, and reliable methods for product tracking, supplier information management, ordering processing, and restocking control. The IMS develops a centralized and accessible database system to improve the accuracy of data and decrease manual errors, thus enhancing the decision-making process by giving access in real-time to the necessary inventory information. This project not only handles the functional requirements in the area of inventory management but aims to exemplify practical applications of the principles of database management, SQL querying, and CRUD operations within a web-based environment.

**Components**

The Inventory Management System comprises the following key components:

1. **Product Management:**

* Allows users to add, update, and delete product information, including details such as product name, description, price, and current stock level.
* Enables the viewing of a comprehensive product list, providing users with an overview of available inventory and allowing easy access to specific product details.

1. **Order Management:**

* Provides functionality for users to place new orders based on current inventory levels, helping ensure products are always available.
* Tracks order history, including order dates, quantities, and supplier details, allowing the organization to monitor procurement activities and plan accordingly.

1. **Restocking System:**

* Identifies low-stock items based on predefined thresholds and generates restocking prompts for these products.
* Assists in generating restocking requests to suppliers, helping to prevent stockouts and ensuring that inventory remains at optimal levels.

1. **Database:**

* The core of the IMS, built using MySQL, contains multiple relational tables such as \*\*Products\*\*, \*\*Suppliers\*\*, and \*\*Orders\*\*.
* Ensures data integrity and facilitates efficient data retrieval, updates, and storage through normalized structures and SQL operations.

1. **User Interface:**

* Developed using HTML and CSS, the front-end interface provides a user-friendly environment for interacting with the system.
* The interface includes pages for adding products, viewing inventory, placing orders, and managing restocking needs, with navigation links for seamless transitions between tasks.

1. **Backend Processing:**

* Built using PHP, the backend processes user requests, performs CRUD operations on the database, and dynamically generates feedback.
* Ensures secure and efficient data processing, allowing for real-time updates to inventory records and supplier information.

1. **Validation and Error Handling:**

* Includes form validation to ensure data accuracy when users input information into the system.
* Provides feedback for errors (such as missing required fields or invalid input formats) and notifies users upon successful completion of tasks like adding a product or placing an order.

1. **Testing and Validation:**

* A critical component to ensure the system functions correctly and meets the required specifications.
* Testing is conducted to verify data accuracy, system reliability, and ease of use, focusing on verifying database updates, form validations, and proper user feedback.

**Project Details**

1. **HOME PAGE:**

**Description**: This is the main landing page, providing an overview of the Inventory Management System.

**Functionality**:

* Shows summary information such as the total number of products, recent orders, low-stock alerts, and pending restocks.
* Contains navigation links to each of the other pages for easy access to various functionalities within the system.

**Source Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Inventory Management System</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<!-- Navigation Bar -->

<header>

<nav>

<ul>

<li><a href="add\_product.php">Add Product</a></li>

<li><a href="view\_inventory.php">View Inventory</a></li>

<li><a href="place\_order.php">Place Order</a></li>

<li><a href="restock.php">Restock Product</a></li>

</ul>

</nav>

</header>

<!-- Main Content Section -->

<main>

<h1>Welcome to the Inventory Management System</h1>

<p>Select an option from the menu to manage your inventory.</p>

</main>

</div>

</body>

</html>

**Style:**

/\* Reset default styles \*/

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    font-family: Arial, sans-serif;

}

/\* Body styling \*/

body {

    display: flex;

    flex-direction: column;

    align-items: center;

    background-color: #f4f4f9;

    color: #333;

    min-height: 100vh;

    margin: 0;

}

/\* Header styling \*/

header {

    background-color: #4CAF50;

    width: 100%;

    padding: 20px 0;

    text-align: center;

    color: white;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

header h1 {

    font-size: 24px;

    margin-bottom: 10px;

}

nav {

    display: flex;

    justify-content: center;

    gap: 10px;

    margin-top: 10px;

}

nav button {

    background-color: white;

    color: #4CAF50;

    border: 1px solid #4CAF50;

    padding: 10px 20px;

    border-radius: 5px;

    cursor: pointer;

    font-size: 14px;

    transition: all 0.3s;

}

nav button:hover {

    background-color: #45a049;

    color: white;

}

/\* Main content styling \*/

main {

    width: 100%;

    max-width: 600px;

    padding: 20px;

    display: flex;

    flex-direction: column;

    gap: 20px;

}

/\* Section styling \*/

.section {

    display: none;

    background-color: #ffffff;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.section h2 {

    font-size: 20px;

    color: #333;

    margin-bottom: 15px;

}

.section.hidden {

    display: none;

}

.section:not(.hidden) {

    display: block;

}

/\* Form styling \*/

form {

    display: flex;

    flex-direction: column;

    gap: 15px;

}

form input[type="text"],

form input[type="number"],

form input[type="date"],

form button {

    padding: 10px;

    font-size: 16px;

    border: 1px solid #ddd;

    border-radius: 5px;

    transition: all 0.3s;

}

form input:focus {

    border-color: #4CAF50;

    outline: none;

}

form button {

    background-color: #4CAF50;

    color: white;

    border: none;

    cursor: pointer;

    font-size: 16px;

}

form button:hover {

    background-color: #45a049;

}

/\* Result message styling \*/

#addProductResult,

#placeOrderResult,

#restockResult,

#inventoryList {

    margin-top: 15px;

    padding: 10px;

    border-radius: 5px;

    font-weight: bold;

}

#addProductResult.success,

#placeOrderResult.success,

#restockResult.success {

    background-color: #d4edda;

    color: #155724;

    border: 1px solid #c3e6cb;

}

#addProductResult.error,

#placeOrderResult.error,

#restockResult.error {

    background-color: #f8d7da;

    color: #721c24;

    border: 1px solid #f5c6cb;

}

/\* Responsive design \*/

@media (max-width: 600px) {

    nav {

        flex-direction: column;

    }

    nav button {

        width: 100%;

    }

}

/\* Styling for View Inventory \*/

#viewInventory {

    background-color: #fff;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

#viewInventory button {

    background-color: #4CAF50;

    color: white;

    padding: 10px 20px;

    border-radius: 5px;

    cursor: pointer;

    font-size: 16px;

    border: none;

    margin-bottom: 20px;

    transition: background-color 0.3s;

}

#viewInventory button:hover {

    background-color: #45a049;

}

Inventory table container

.inventory-table-container {

    overflow-x: auto;

}

.inventory-table {

    width: 100%;

    border-collapse: collapse;

    margin-top: 20px;

    border: 1px solid #ddd;

}

.inventory-table th, .inventory-table td {

    padding: 12px;

    text-align: left;

    border: 1px solid #ddd;

}

.inventory-table th {

    background-color: #4CAF50;

    color: white;

    font-weight: bold;

}

.inventory-table tr:nth-child(even) {

    background-color: #f9f9f9;

}

.inventory-table tr:hover {

    background-color: #f1f1f1;

}

/\* Styling for inventory empty message \*/

#inventoryList.empty {

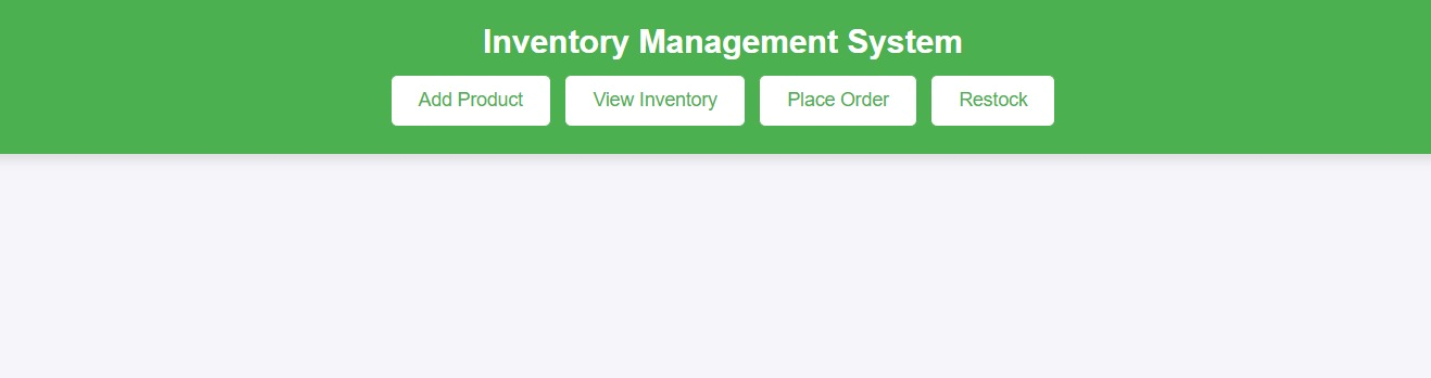
    text-align: center;

    color: #888;

    font-size: 18px;

}

**Screenshot:**



1. **ADD PRODUCT PAGE:**

**Description**: This page allows users to add new products to the inventory database.

**Functionality**:

* Users can input product details such as the product name, description, price, and quantity in stock.
* A form with fields is provided for each of these details, with validation to ensure all required fields are correctly filled.
* Upon submitting, the product information is saved to the database, updating the inventory.

**Source Code:**

<?php

include 'db.php';

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    $name = $\_POST['name'];

    $description = $\_POST['description'];

    $price = $\_POST['price'];

    $quantity = $\_POST['quantity'];

    $sql = "INSERT INTO products (name, description, price, quantity\_in\_stock)

            VALUES ('$name', '$description', '$price', '$quantity')";

    if ($conn->query($sql) === TRUE) {

        echo "New product added successfully";

    } else {

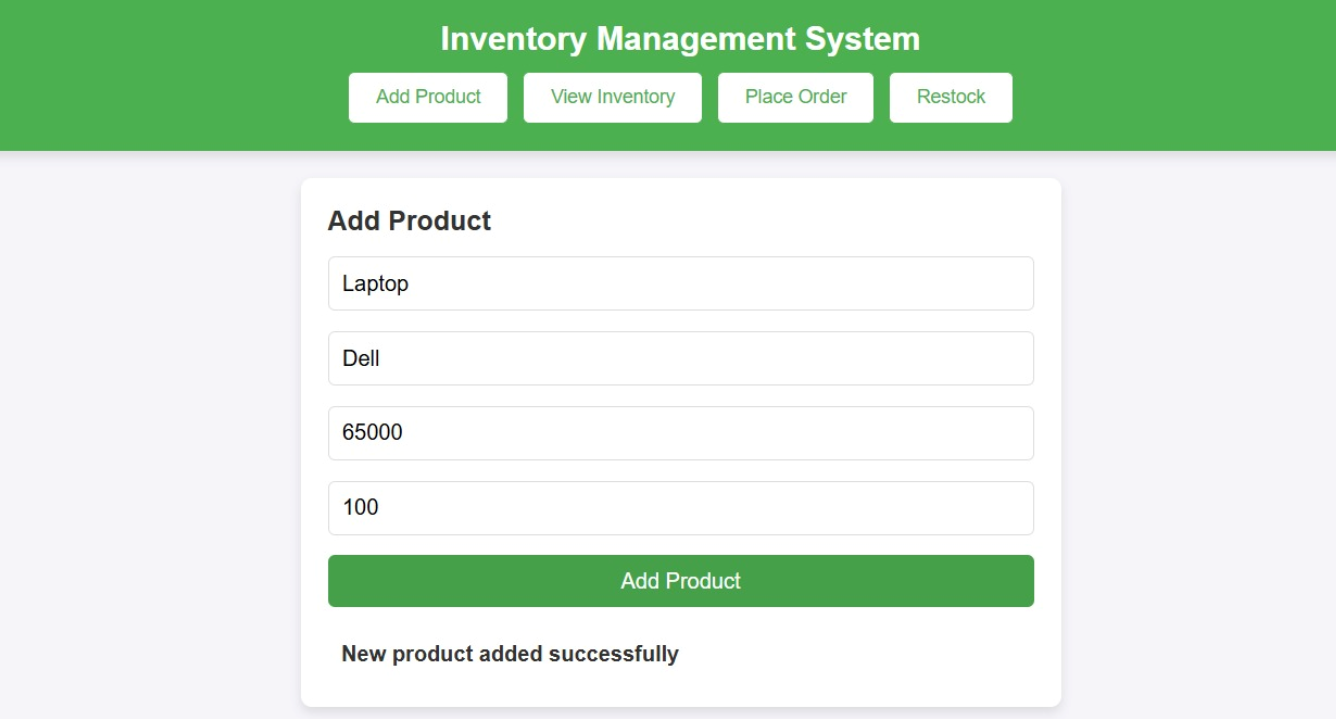
        echo "Error: " . $sql . "<br>" . $conn->error;

    }

}

?>

**Screenshot:**



1. **VIEW INVENTORY PAGE:**

**Description**: This page displays a list of all products currently in the inventory, offering a centralized overview of stock levels and product information.

**Functionality**:

* Users can see details of each product, including name, description, price, and available quantity.
* Provides real-time data on current stock levels, enabling users to assess inventory status at a glance.
* Optional sorting or filtering options may be available for easier viewing and organization of product data.

**Source Code:**

<?php

include 'db.php';

$sql = "SELECT \* FROM products";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

    echo "<table><tr><th>Product ID</th><th>Name</th><th>Description</th><th>Price</th><th>Quantity In Stock</th></tr>";

    while ($row = $result->fetch\_assoc()) {

        echo "<tr><td>" . $row["product\_id"] . "</td><td>" . $row["name"] . "</td><td>" . $row["description"] . "</td><td>" . $row["price"] . "</td><td>" . $row["quantity\_in\_stock"] . "</td></tr>";

    }

    echo "</table>";

} else {

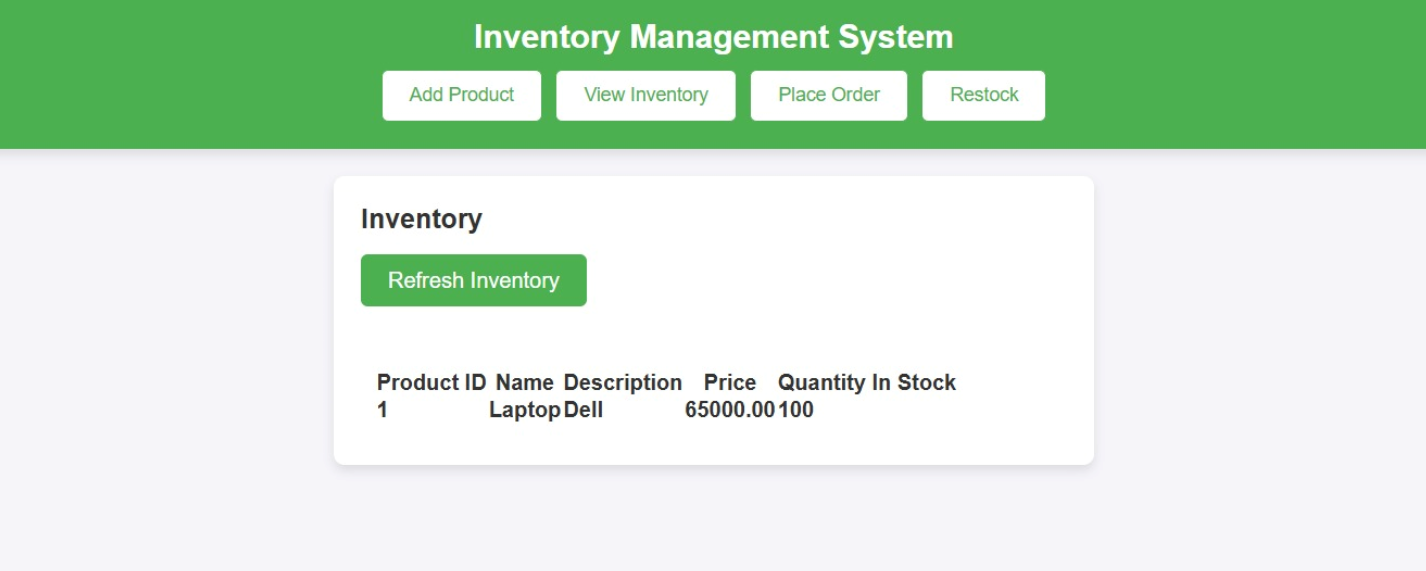
    echo "0 results";

}

$conn->close();

?>

**Screenshot:**



1. **PLACE ORDER PAGE**

Description: This page allows users to create new orders for products that need replenishing.

Functionality:

* Users can select a product and specify the quantity to order, linking the order to a specific supplier if necessary.
* Information such as the product name, supplier, order quantity, and date is input through form fields.
* Upon submission, the order is recorded in the database, helping track procurement activity and manage stock levels.

**Source Code:**

<?php

include 'db.php';

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    $customer\_name = $\_POST['customer\_name'];

    $product\_id = $\_POST['product\_id'];

    $quantity\_ordered = $\_POST['quantity\_ordered'];

    $order\_date = date('Y-m-d');

    // Update product stock

    $update\_stock\_sql = "UPDATE products SET quantity\_in\_stock = quantity\_in\_stock - $quantity\_ordered WHERE product\_id = $product\_id";

    $conn->query($update\_stock\_sql);

    // Insert into orders table

    $sql = "INSERT INTO orders (customer\_name, product\_id, quantity\_ordered, order\_date)

            VALUES ('$customer\_name', '$product\_id', '$quantity\_ordered', '$order\_date')";

    if ($conn->query($sql) === TRUE) {

        echo "Order placed successfully";

    } else {

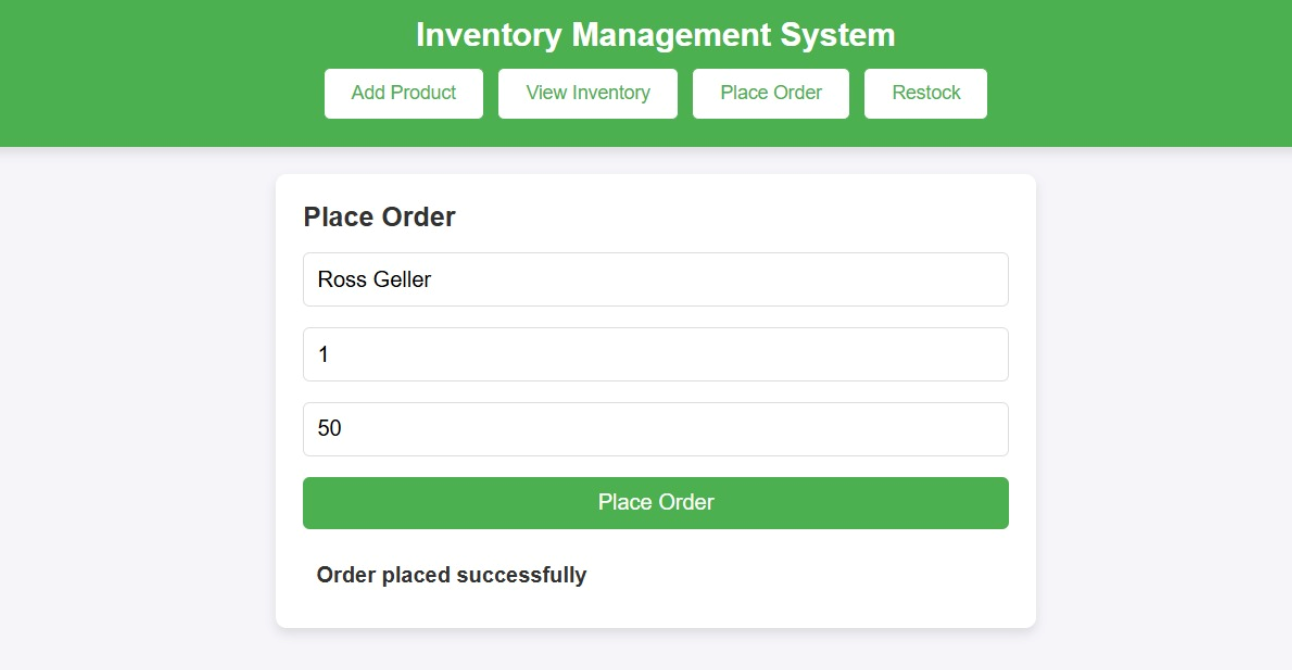
        echo "Error: " . $sql . "<br>" . $conn->error;

    }

}

?>

**Screenshot:**



1. **RESTOCK PRODUCT PAGE**

**Description**: This page is designed to assist users in managing low-stock items and initiating restocking requests.

**Functionality**:

* Displays a list of products with stock levels below a predefined threshold, signaling the need for restocking.
* Allows users to initiate a restock request by specifying the required quantity for each low-stock item.
* Once submitted, these requests are added to the database as pending restocks, aiding the inventory team in replenishing stock in a timely manner.

**Source Code:**

<?php

include 'db.php';

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    $product\_id = $\_POST['product\_id'];

    $quantity = $\_POST['quantity'];

    $restock\_date = date('Y-m-d');

    // Update product stock

    $update\_stock\_sql = "UPDATE products SET quantity\_in\_stock = quantity\_in\_stock + $quantity WHERE product\_id = $product\_id";

    $conn->query($update\_stock\_sql);

    // Insert into restocking table

    $sql = "INSERT INTO restocking (product\_id,  quantity, restock\_date)

          VALUES ('$product\_id',  '$quantity', NOW())";

    if ($conn->query($sql) === TRUE) {

        echo "Product restocked successfully";

    } else {

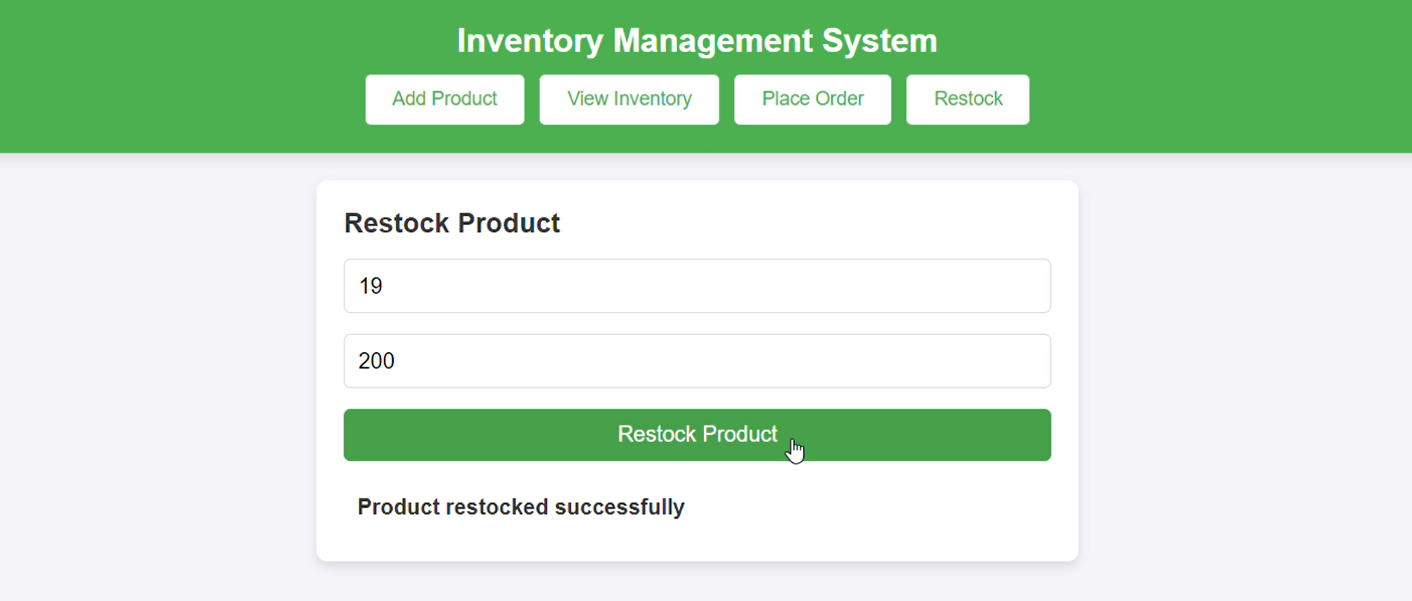
        echo "Error: " . $sql . "<br>" . $conn->error;

    }

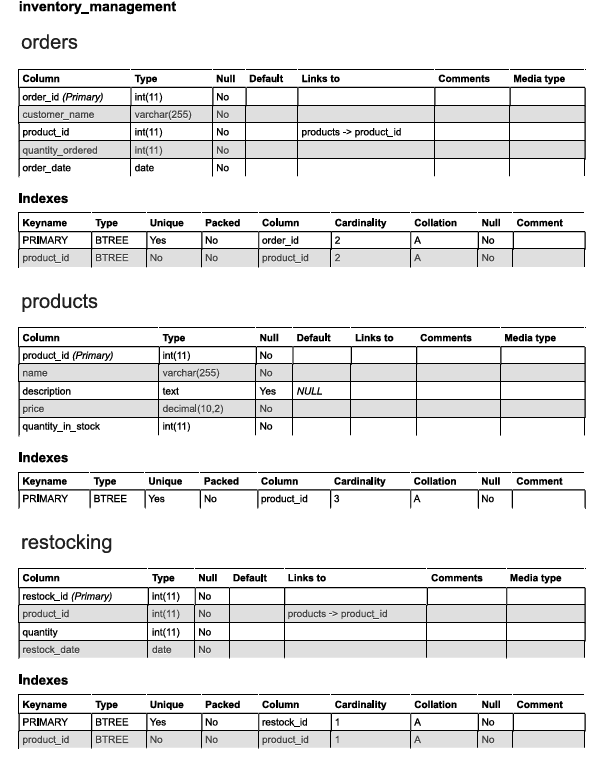
}

?>

**Screenshot:**

****

**DATABASES**



**Video description of the project:** https://drive.google.com/file/d/1N6ff-LEMZH660PtUyv9UV8SV5TZEUrEd/view

**Conclusion**

The Inventory Management System developed in this project demonstrates the effectiveness of a database-driven approach to handling and optimizing inventory processes within an organization. By implementing a MySQL relational database coupled with a PHP-based backend, this system provides a robust solution for real-time inventory tracking, product management, and supplier oversight. This project successfully addresses the inherent challenges of manual inventory systems—such as data inaccuracies, labor-intensive updates, and limited scalability—by providing a platform that automates and centralizes critical inventory information.

Moreover, the IMS project underscores the practical importance of database management concepts and technologies in developing applications that support business operations. The use of relational tables, SQL queries, and web-based interfaces in this system illustrates how core DBMS principles can be applied to create solutions that are both reliable and adaptable. Although the current version of the system meets essential requirements, the potential for expansion is substantial. Future enhancements could include automated notifications for low-stock items, integration with other organizational systems, and advanced data analytics for inventory forecasting.

Through this project, we highlight the role of DBMS in modern inventory management, demonstrating how digital tools can streamline workflows, minimize errors, and enhance organizational responsiveness. The success of this project reinforces the need for continued exploration of database-driven solutions in operational settings, positioning the IMS as a valuable tool for businesses seeking to optimize inventory control and resource allocation.