KENYATTA UNIVERSITY

SCO 207

WEB DEVELOPMENT TECHNOLOGIES

CAT II

NAME	REG NO.	SIGNATURE
MUSYOKA JAMES	I163/1071/2021	
BARAKA BRAVE	I163/5842/2021	
WANJIRU LYNN	I163/4691/2021	
YANO FELIX	I163/5849/2021	
KITIVO ERIC	I163/5819/2021	
ARANI ISAAC	I163/5860/2021	
HELLEN MULATYA	I163/5826/2021	
MBUGUA CHRISTINE	I163/1720/2021	

Contents

# INVENTORY MANAGEMENT SYSTEM DOCUMENTATION

Introduction

Inventory management is a crucial aspect of any business that involves handling physical goods. The efficiency of managing inventory directly impacts various aspects of business operations, including cost management, customer satisfaction, and overall profitability. An effective inventory management system streamlines the process of tracking, storing, and managing inventory items, ensuring optimal stock levels while minimizing overhead costs.

The Inventory Management System website described in this documentation offers a comprehensive solution for businesses to manage their inventory efficiently. Through a user-friendly web interface, users can perform various tasks, including user authentication, registration, viewing inventory items, adding new products, editing existing products, and deleting products.
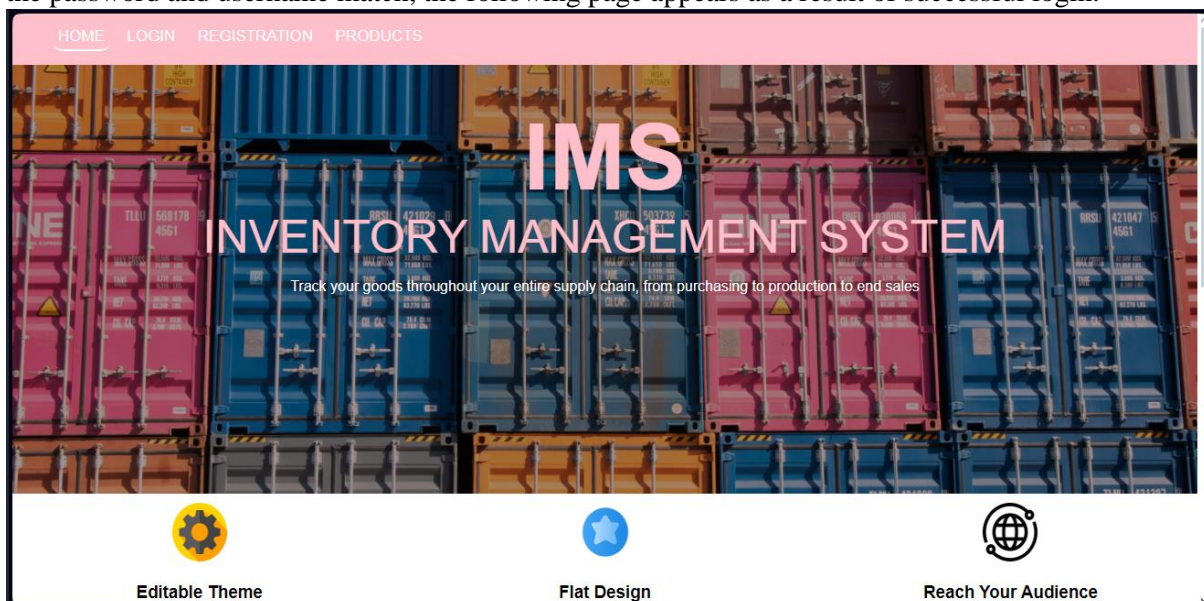
1. Login Page
Purpose:
      The Login page serves as the primary entry point for registered users to access the system securely. It verifies the identity of users by requiring them to input their credentials, typically a username and password, before granting access to the system's functionalities. See the figure below:
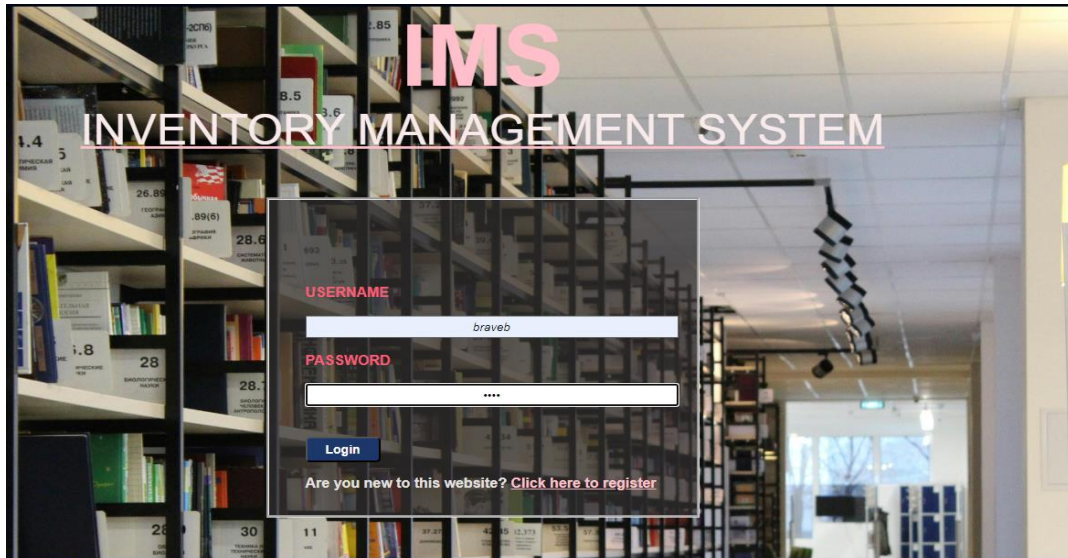

Functionalities:
User Authentication:
      The Login page verifies the authenticity of user-provided credentials against the information stored in the system's database. Upon successful authentication, users are granted access to the system. If the password and username match, the following page appears as a result of successful login:
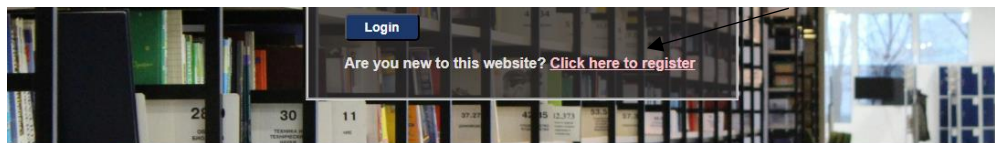
Error Handling:

In cases where users provide incorrect or invalid credentials, the Login page displays appropriate error messages, guiding users to rectify their input and attempt login again.
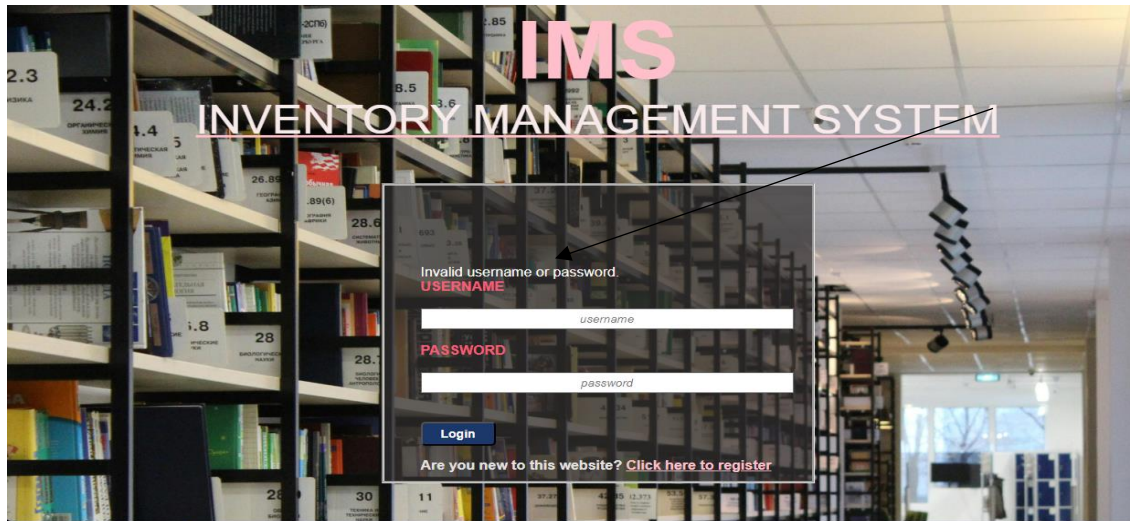
Also, if the user does not have an account, the interface allows the user to quickly register by clicking this link:

Database Connection:

The Login page interacts with the database to validate user credentials securely. It performs a query to retrieve the user's stored information based on the provided username and then compares the hashed password stored in the database with the hashed version of the password provided by the user during login. This ensures that sensitive user data remains protected even in the event of a security breach. The figure below illustrates how this was achieved:

```php
<?php
// Function to handle login logic
function handleLogin() {
    // Assuming you have a database connection
    $conn = new mysqli("localhost", "root","", "inventory");

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // Sanitize user input
    $username = $conn->real_escape_string($_POST["username"]);
    $password = $conn->real_escape_string($_POST["password"]);

    // Query to check user credentials
    $sql = "SELECT * FROM login WHERE username='$username' AND password='$password'";
    $result = $conn->query($sql);

    // Check if a matching user is found
    if ($result->num_rows > 0) {
        header("Location: homepage.php");
        echo "Login successful!";
        exit();
    } else {
        echo "Invalid username or password.";
    }

    // Close database connection
    $conn->close();
}
?>
```
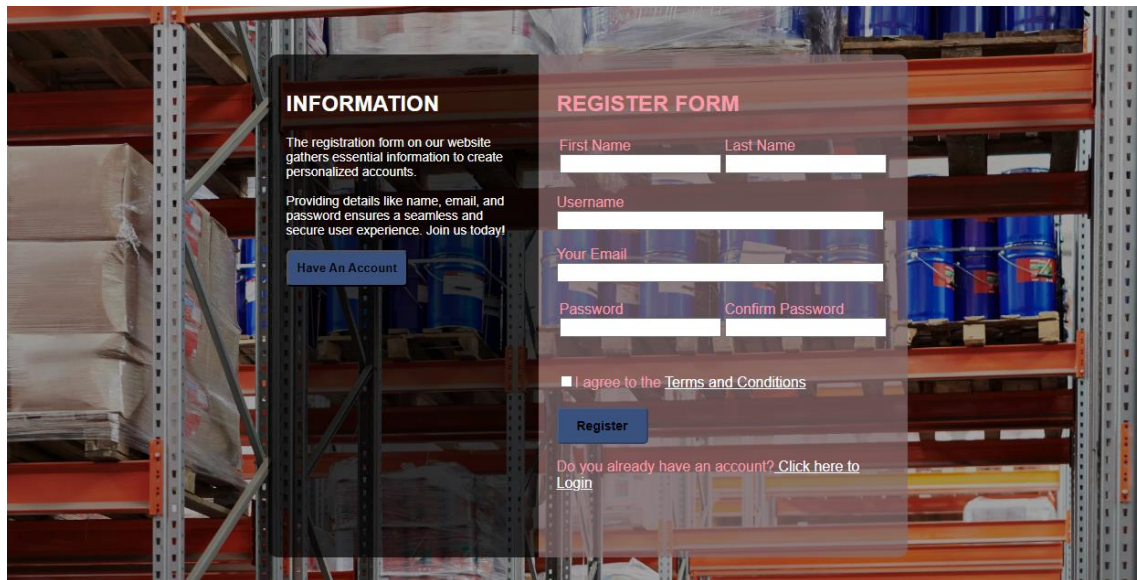
2. Registration Page

Purpose:

      The Registration page facilitates the creation of new user accounts within the system. It allows individuals who do not yet have an account to sign up and gain access to the system's features and functionalities. The figure below is an outlook of the Registration page:
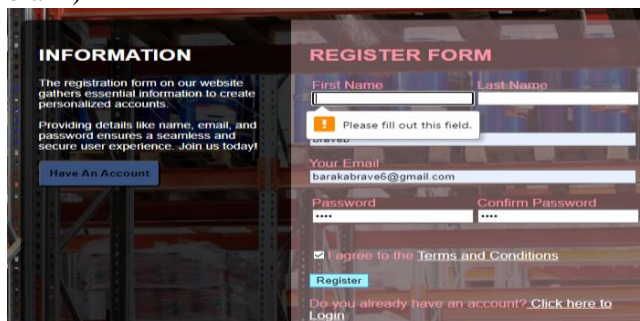

Functionalities:

User Registration Form:

      The Registration page presents users with a form where they can input their details, including First Name, Last Name, unique Username, email address, and Password.

Data Validation:

    Upon submission of the registration form, the system validates the entered data to ensure that all required fields are filled, and the provided information meets specific criteria:

i)   All required fields are entered (in this case, the 'First name' which is a required field was left blank)

ii)   The 'I Agree to the Terms and Conditions' checkbox is checked : The 'Register' button is deactivated, i.e the user cannot register is the checkbox is unchecked.

iii)   The values entered in the 'Password' textbox and the 'Confirm Password' textbox should match, otherwise and error message will be displayed and the registration will not be complete. See the figure below for illustration:

Account Creation:

Once the user-provided data passes validation checks, the Registration page adds the new user's details to the system's database, creating a new user account.
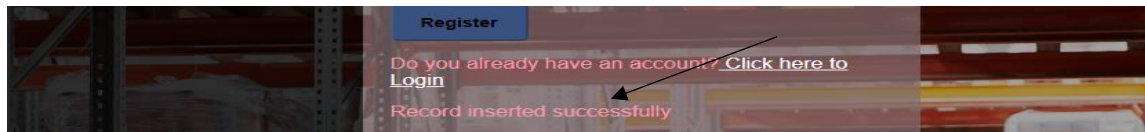
Database Connection:

The Registration page interacts with the database to store newly created user accounts securely. It performs an insertion operation, adding the user's information to the appropriate database table while ensuring data integrity and security. The following figure illustrates how this was achieved:

```php
</script>
            <?php
$servername = "localhost";  // Change to your server name if different
$username = "root";         // Change to your MySQL username
$password = "";             // Change to your MySQL password if set
$dbname = "inventory";  // Change to your database name

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $firstname = $conn->real_escape_string($_POST['firstname']);
    $lastname = $conn->real_escape_string($_POST['lastname']);
    $username = $conn->real_escape_string($_POST['username']);
    $email = $conn->real_escape_string($_POST['email']);
    $password = $conn->real_escape_string($_POST['password']);
    $confirm_password= $conn->real_escape_string($_POST["confirm_password"]);
    if ($password === $confirm_password){
    $sql = "INSERT INTO login (f_name, l_name, username, email, password) VALUES ('$firstname', '$lastname', '$username','$email', '$password')";

    if ($conn->query($sql) === TRUE) {
        echo "Record inserted successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
else{
    echo "Password do not match";
    $conn->close();
    return;
}}

// Close connection
$conn->close();
?>
```
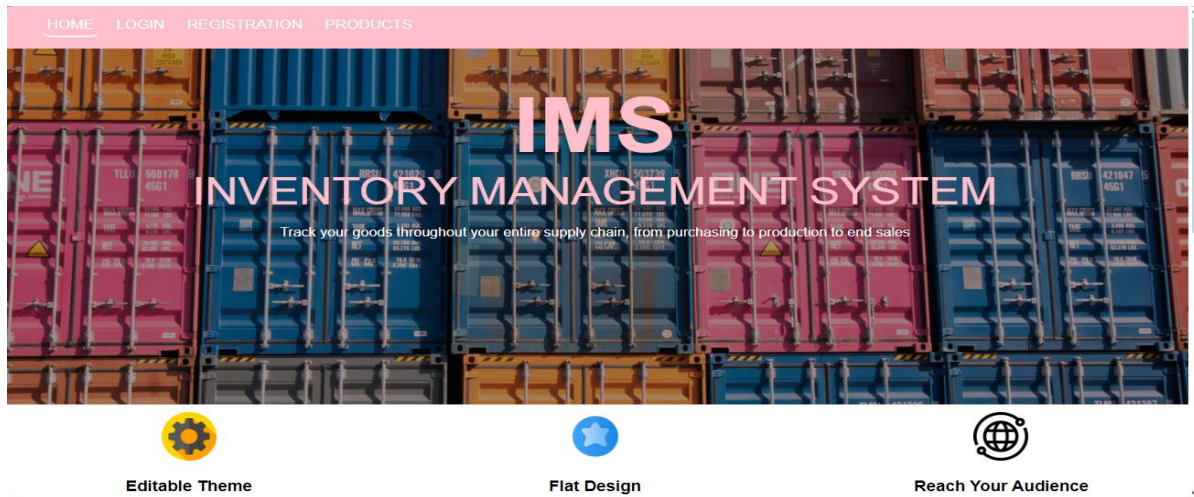
3. Homepage

Purpose:

The Homepage serves as the central hub of the Inventory Management System, providing users with an overview of their inventory-related activities and facilitating navigation to various sections of the system.


Functionalities:

Navigation Menu:

A navigation menu or sidebar allows users to navigate seamlessly between different sections of the system, such as viewing products, home and login, and accessing account settings.

Recent Activity Feed:

The Homepage may include a section that highlights recent changes or updates to the inventory, providing users with real-time insights into inventory movements, such as newly added products, updated stock levels, or fulfilled orders.
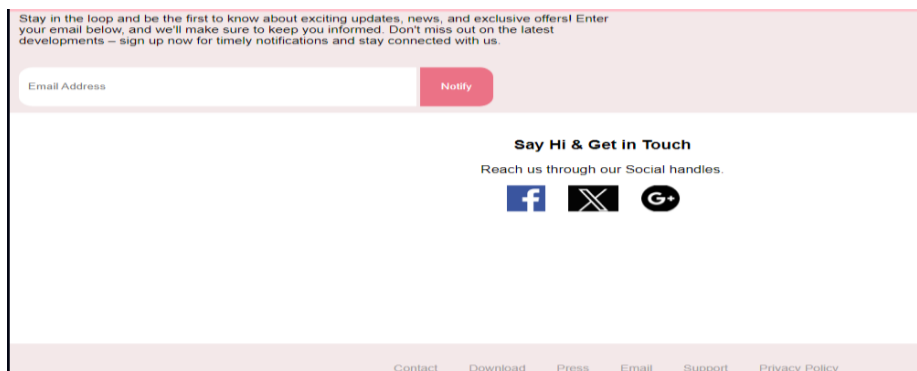
4. Products Page
Purpose:
       The Products page allows users to view, manage, and manipulate inventory items within the system. It serves as a central repository for storing product information and enables users to perform actions such as adding new products, editing existing ones, and deleting obsolete items.

Functionalities:
Product Listing:
       The Products page displays a comprehensive list of inventory items, including details such as product name, description, quantity in stock, unit price, and any relevant attributes or categories.
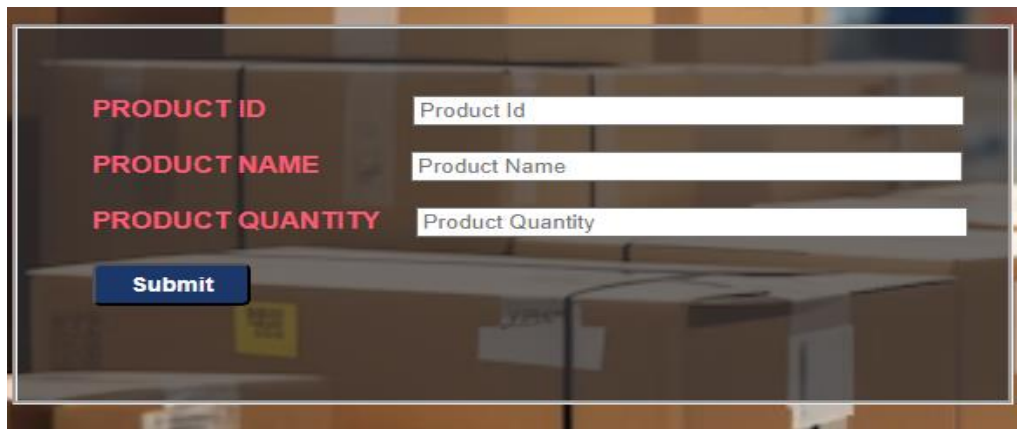

Add Product Form:
       Users can utilize a form to add new products to the inventory, providing necessary information such as product id, product name and quantity.

Database Connection:

The Products page interacts extensively with the system's database to retrieve, update, and delete product information. It executes database queries to fetch product data for display, insert new product entries, update existing records, and delete products based on user actions.

```php
$username = "root";        // Change to your MySQL username
$password = "";            // Change to your MySQL password if set
$dbname = "inventory";     // Change to your database name

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $product_id = $conn->real_escape_string($_POST['product_id']);
    $product_name = $conn->real_escape_string($_POST['product_name']);
    $quantity = $conn->real_escape_string($_POST['quantity']);

    $sql = "INSERT INTO product (product_id, product_name, quantity) VALUES ('$product_id', '$product_name', '$quantity')";

    if ($conn->query($sql) === TRUE) {
        echo "Record inserted successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
```

Conclusion

The Inventory Management System website offers a robust solution for businesses to streamline their inventory management processes effectively. By providing a user-friendly interface and seamless integration with a backend database, the system empowers users to manage their inventory efficiently, track stock levels accurately, and make informed decisions to optimize inventory operations and maximize profitability.