

# **Jenny Store**

## **Developer guide**

## **Group member**

Student 1636545	Laiba Khurram
Student 1636180	Noor Fatima Tanveer
Student 1639611	Anjalla Saleem
Student 1636185	Eman Shujaat

**CURRICULUM: ACCP AI**

**SEMESTER: DISM**

**FACULTY: Sir M. Tasbeel Jawed**

# **Certificate of Recognition**

This certificate is proudly presented to

Noor Fatima Tanveer

Laiba Khurram

Anjalla Saleem

Emaan Shujaat

for the successful development of the project titled  
“Jenny Store” during APTECH Semester 2.

This recognition acknowledges the team’s dedication, creativity, and technical skills demonstrated in designing and developing the Jenny Store project. The project reflects a strong understanding of practical concepts, effective teamwork, and a professional approach toward project implementation.

The efforts and commitment shown by the team throughout the development process are commendable and represent a valuable contribution to their academic and professional growth.

This certificate stands as a testament to their hard work,  
innovation, and determination.

Presented On: 23 January 2026

Authorized By Sir M. Tasbeel Jawed

# **Acknowledgement**

We would like to express our sincere gratitude to “*Sir Tasbeel Jawed*”, our esteemed mentor, for his invaluable guidance, continuous support, and insightful feedback throughout the development of the Jenny care project. His expertise and encouragement played a pivotal role in shaping both the technical and conceptual aspects of this work. We extend our heartfelt thanks to Aptech Institution, whose academic resources, infrastructure, and training environment provided a strong foundation for successfully completing this project.

# **Synopsis**

Jenny Store is a web-based e-commerce system developed for a small home-based business dealing in cosmetics and imitation jewelry. Initially, the business was managed manually using physical catalogues and handwritten records, which became inefficient as the number of customers and orders increased.

The proposed system provides an online platform where customers can browse products, search by product name, add items to a shopping cart, specify quantities, and place orders by entering required personal and order details. The system maintains accurate digital records of customers, products, and orders, reducing manual effort and improving efficiency.

An administrator module allows management of product categories and products, updating product information, maintaining the website, managing database records, taking backups, and generating sales and customer reports. The Jenny Store system improves accessibility, enhances business operations, and supports better decision-making through digital automation.

## **Categories:**

The Jenny Store system organizes products into well-defined categories such as cosmetics and imitation jewelry to ensure easy navigation and better product management. Each category contains multiple products with detailed information including product name, price, and description.

## **Distribution:**

Products are distributed across categories in a structured manner, allowing customers to browse efficiently and search for desired items quickly. The category-based distribution also helps the administrator manage inventory, update products, and analyze sales performance effectively. This organized structure improves user experience and supports smooth operation of the e-commerce system.

## **Problem Statement:**

Jenny runs a small home-based business dealing in cosmetics and imitation jewelry. Initially, she managed her business by personally contacting friends and relatives, informing them about available products and prices. Product details were maintained in a physical catalogue, while customer contact details and orders were recorded manually in a diary. As the business grew and the number of customers increased significantly, managing products, clients, and orders manually became inefficient and time-consuming. To handle the growing demand and improve business operations, Jenny now requires a web-based e-commerce website that displays the complete product catalogue and allows customers to place orders online.

The proposed system will maintain complete and accurate records of customers, products, and orders, thereby reducing manual effort and improving accessibility and efficiency.



# **Requirement Specification:**

The development of the Jenny Store e-commerce system requires basic hardware and software resources to ensure smooth operation. The system should run on a computer or laptop with a stable internet connection. A modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge is required to access the website. For development purposes, standard web development tools such as a code editor and local server environment may be used.

The Jenny Store system is a web-based application designed to manage online product display and ordering. The system allows customers to browse product categories, search products, add items to a shopping cart, specify quantities, and place orders by entering mandatory personal and order-related information. The administrator is responsible for managing product categories and products, updating information, maintaining the website, managing database records, taking backups, and generating sales and customer reports. The system is designed to improve efficiency, accuracy, and accessibility through digital automation.

# GUIDE:

```
<?php
require_once "../includes/auth_admin.php";
require_once "../config/db.php";

$message = "";
$error = "";

/* Handle Backup */
if ($_SERVER["REQUEST_METHOD"] === "POST" && isset($_POST['backup'])) {

    if (!isset($_SESSION['admin_id'], $_SESSION['admin_role'])) {
        $error = "Unauthorized access.";
    } else {
        $admin_name = $_SESSION['admin_role'] . " (ID: " . $_SESSION['admin_id'] . ")";

        $backupFile = "backup_" . date("Y-m-d_H-i-s") . ".sql";

        $stmt = $conn->prepare("
            INSERT INTO backup_logs (backup_by, remarks)
            VALUES (?, ?)
        ");
        $stmt->execute([$admin_name, "Manual backup created: $backupFile"]);

        $message = "Database backup created successfully.";
    }
}

$stmt = $conn->query("SELECT * FROM backup_logs ORDER BY backup_date DESC");
$backups = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Database Backup | Admin Panel</title>

<!-- Bootstrap -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
```

# Admin (categories):

```
admin > 🐞 categories.php
1  <?php
2  require_once "../includes/auth_admin.php";
3  require_once "../config/db.php";
4
5  $success = "";
6  $error = "";
7
8  /* ADD CATEGORY */
9  if ($_SERVER["REQUEST_METHOD"] === "POST" && isset($_POST['add_category'])) {
10
11      $category_name = trim($_POST['category_name']);
12      $description = trim($_POST['description']);
13
14      if (empty($category_name)) {
15          $error = "Category name is required.";
16      } else {
17          $stmt = $conn->prepare(
18              "INSERT INTO categories (category_name, description) VALUES (?, ?)"
19          );
20          $stmt->execute([$category_name, $description]);
21
22          $success = "Category added successfully.";
23      }
24  }
25
26  /* DELETE CATEGORY */
27  if (isset($_GET['delete'])) {
28      $id = (int) $_GET['delete'];
29
30      $stmt = $conn->prepare("DELETE FROM categories WHERE category_id = ?");
31      $stmt->execute([$id]);
32
33      $success = "Category deleted successfully.";
34  }
35
36  /* FETCH CATEGORIES */
37  $stmt = $conn->query("SELECT * FROM categories ORDER BY created_at DESC");
38  $categories = $stmt->fetchAll(PDO::FETCH_ASSOC);
39  ?>
```

# Admin (customers):

```
admin > customers.php
1  <?php
2  require_once "../includes/auth_admin.php";
3  require_once "../config/db.php";
4
5  $success = "";
6
7  /* DELETE CUSTOMER */
8  if (isset($_GET['delete'])) {
9      $id = (int) $_GET['delete'];
10
11      $stmt = $conn->prepare("DELETE FROM customers WHERE customer_id = ?");
12      $stmt->execute([$id]);
13
14      $success = "Customer deleted successfully.";
15  }
16
17  /* FETCH ALL CUSTOMERS */
18  $stmt = $conn->query("SELECT * FROM customers ORDER BY created_at DESC");
19  $customers = $stmt->fetchAll(PDO::FETCH_ASSOC);
20  ?>
21
22  <!DOCTYPE html>
23  <html lang="en">
24  <head>
25  <meta charset="UTF-8">
26  <title>Manage Customers | Admin Panel</title>
27
28  <!-- Bootstrap -->
29  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
30
31  <!-- Google Font -->
32  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">
33
34  <style>
35      body {
36          font-family: 'Poppins', sans-serif;
37          background: #f4f7f5;
38          color: #2c2c2c;
39      }
```

# Admin (dashboard):

```
<body>

<?php include "../includes/navbar.php"; ?>

<div class="dashboard-wrapper">

    <div class="dashboard-header text-center">
        <h2>Admin Dashboard</h2>
        <p>Overview & management controls for your jewelry store</p>
    </div>

    <!-- Statistics -->
    <div class="row g-4">
        <div class="col-md-3 col-sm-6">
            <div class="stat-card">
                <div class="stat-value"><?= (int)$totalCategories ?></div>
                <div class="stat-label">Categories</div>
            </div>
        </div>
        <div class="col-md-3 col-sm-6">
            <div class="stat-card">
                <div class="stat-value"><?= (int)$totalProducts ?></div>
                <div class="stat-label">Products</div>
            </div>
        </div>
        <div class="col-md-3 col-sm-6">
            <div class="stat-card">
                <div class="stat-value"><?= (int)$totalCustomers ?></div>
                <div class="stat-label">Customers</div>
            </div>
        </div>
        <div class="col-md-3 col-sm-6">
            <div class="stat-card">
                <div class="stat-value"><?= (int)$totalOrders ?></div>
                <div class="stat-label">Orders</div>
            </div>
        </div>
    </div>
</div>
```

## Admin (login):

```
129
130 <div class="login-wrapper">
131
132     <div class="brand-title">Address Jewelers</div>
133     <div class="brand-subtitle">Admin Control Panel</div>
134
135     <?php if (!empty($error)): ?>
136         <div class="error-msg"><?=$error ?></div>
137     <?php endif; ?>
138
139     <form method="POST">
140         <div class="mb-3">
141             <label class="form-label">Username</label>
142             <input type="text" name="username" class="form-control" required>
143         </div>
144
145         <div class="mb-4">
146             <label class="form-label">Password</label>
147             <input type="password" name="password" class="form-control" required>
148         </div>
149
150         <button type="submit" class="btn btn-login w-100">
151             Secure Login
152         </button>
153     </form>
154
155     <div class="footer-text">
156         &copy; 2025 Address Jewelers • Authorized Access Only
157     </div>
158
159 </div>
160
```

## Admin (logout):

```
admin > 🐘 logout.php
1  <?php
2  session_start();
3  session_destroy();
4  header("Location: ../public/index.php");
5  exit;
6  |
```

## Admin (orders):

```
1  <?php
2  require_once "../includes/auth_admin.php";
3  require_once "../config/db.php";
4
5  /* UPDATE ORDER STATUS */
6  if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['update_status'])) {
7      $order_id = intval($_POST['order_id']);
8      $status = $_POST['status'];
9
10     $stmt = $conn->prepare("UPDATE orders SET status = ? WHERE order_id = ?");
11     $stmt->execute([$status, $order_id]);
12
13     header("Location: orders.php");
14     exit;
15 }
16
17 /* FETCH ORDERS WITH CUSTOMER INFO */
18 $stmt = $conn->query("
19     SELECT o.*, c.name, c.email, c.cell_phone
20     FROM orders o
21     JOIN customers c ON o.customer_id = c.customer_id
22     ORDER BY o.order_date DESC
23 ");
24 $orders = $stmt->fetchAll(PDO::FETCH_ASSOC);
25 ?>
26
27 <!DOCTYPE html>
28 <html lang="en">
29 <head>
30 <meta charset="UTF-8">
31 <title>Manage Orders | Admin</title>
32
33 <!-- Bootstrap -->
34 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
35
36 <!-- Google Font (same as other admin page for navbar consistency) -->
37 <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">
```

## Admin (products):

```
1  <?php
2  require_once "../includes/auth_admin.php";
3  require_once "../config/db.php";
4
5  /* FETCH CATEGORIES */
6  $categories = $conn->query("
7  | SELECT * FROM categories ORDER BY category_name
8  |")->fetchAll(PDO::FETCH_ASSOC);
9
10 /* EDIT MODE FETCH */
11 $editProduct = null;
12 if (isset($_GET['edit'])) {
13     $id = (int)$_GET['edit'];
14     $stmt = $conn->prepare("SELECT * FROM products WHERE product_id = ?");
15     $stmt->execute([$id]);
16     $editProduct = $stmt->fetch(PDO::FETCH_ASSOC);
17 }
18
19 /* ADD PRODUCT */
20 if ($_SERVER["REQUEST_METHOD"] === "POST" && isset($_POST['add_product'])) {
21     $category_id = $_POST['category_id'];
22     $name = trim($_POST['product_name']);
23     $desc = trim($_POST['product_description']);
24     $price = $_POST['price'];
25     $stock = $_POST['stock_quantity'];
26     $status = $_POST['status'];
27
28     /* VALIDATIONS */
29     $errors = [];
30
31     if ($price < 0) {
32         $errors[] = "Price cannot be negative.";
33     }
34
35     if ($stock < 0) {
36         $errors[] = "Stock quantity cannot be negative.";
37     }
38 }
```



## Admin (reports):

```
154 <!-- TOP CUSTOMERS -->
155 <div class="report-box table-responsive">
156   <h2>Top 10 Customers by Purchases</h2>
157
158   <table class="table align-middle">
159     <thead>
160       <tr>
161         <th>Rank</th>
162         <th>Customer Name</th>
163         <th>Total Amount Spent</th>
164       </tr>
165     </thead>
166     <tbody>
167       <?php $rank = 1; foreach ($topCustomers as $c): ?>
168         <tr>
169           <td class="rank">#<?php $rank++ ?></td>
170           <td><?php htmlspecialchars($c['name']) ?></td>
171           <td>R$ <?php number_format($c['total_spent'], 2) ?></td>
172         </tr>
173       <?php endforeach; ?>
174       <?php if (empty($topCustomers)): ?>
175         <tr>
176           <td colspan="3" class="text-center">No customer data available</td>
177         </tr>
178       <?php endif; ?>
179     </tbody>
180   </table>
181 </div>
182
183 </div>
184
185 <?php include "../includes/footer.php"; ?>
```

## Config (db.php):

```
config > db.php > ...
1  <?php
2  $host = "localhost";
3  $dbname = "address_jewelers";
4  $username = "root";
5  $password = "";
6
7  try {
8      $conn = new PDO(
9          "mysql:host=$host;dbname=$dbname;charset=utf8mb4",
10         $username,
11         $password
12     );
13     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
14 } catch (PDOException $e) {
15     die("Database Connection Failed: " . $e->getMessage());
16 }
17 ?>
```

## Includes (auth\_admin.php):

```
includes > auth_admin.php
1  <?php
2  session_start();
3
4  if (!isset($_SESSION['admin_id'])) {
5      header("Location: login.php");
6      exit;
7  }
8  ?>
```

## **Includes** (auth\_customer.php):

```
1  <?php
2  session_start();
3
4  if (!isset($_SESSION['customer_id'])) {
5      header("Location: login.php");
6      exit;
7  }
8  ?>
```

## **Includes** (auth\_function.php):

```
includes > functions.php > ...
1  <?php
2  function sanitize($data) {
3      return htmlspecialchars(trim($data), ENT_QUOTES, 'UTF-8');
4  }
5
6  function isPost() {
7      return $_SERVER['REQUEST_METHOD'] === 'POST';
8  }
```

## Includes (auth\_footer.php):

```
includes / rooter.php / ...
1 <footer class="admin-footer mt-auto py-3">
2   <div class="container-fluid text-center">
3     <p class="mb-1 fw-semibold">
4       Address Jewelers Admin Panel
5     </p>
6     <p class="mb-0 small text-muted">
7       &copy; <? = date('Y'); ?> All Rights Reserved
8     </p>
9   </div>
10 </footer>
11
12 <style>
13 /* Footer Base */
14 .admin-footer {
15   background: linear-gradient(135deg, #0b6b4f, #158f6b);
16   color: #f0f0f0;
17   padding: 15px 0;
18   width: 100%;
19   box-shadow: 0 -4px 15px rgba(0,0,0,0.2);
20   border-top: 1px solid rgba(255,255,255,0.1);
21   font-family: 'Poppins', sans-serif;
22 }
23
24 /* Footer Text */
25 .admin-footer p {
26   margin: 0;
27 }
28
29 /* Footer Link Hover (Optional if you add links) */
30 .admin-footer a {
31   color: #fff;
32   text-decoration: none;
33   transition: color 0.3s ease;
34 }
35
36 .admin-footer a:hover {
37   color: #ffd700;
```

## Includes (auth\_navbar.php):

```
1 <?php
2 // navbar.php
3 if (session_status() === PHP_SESSION_NONE) {
4     session_start();
5 }
6
7 // Get current page for active link
8 $currentPage = basename($_SERVER['PHP_SELF']);
9 ?>
10
11 <nav class="navbar navbar-expand-lg navbar-dark shadow-sm" style="background: linear-gradient(135deg, #0b6b4f, #158f6b);">
12     <div class="container-fluid px-4">
13         <!-- Logo -->
14         <a class="navbar-brand d-flex align-items-center" href="dashboard.php">
15             
16             <span class="fw-bold fs-5 text-white">Jenny Store</span>
17         </a>
18
19         <!-- Toggler for mobile -->
20         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#adminNavbar">
21             <span class="navbar-toggler-icon"></span>
22         </button>
23
24         <!-- Menu Items -->
25         <div class="collapse navbar-collapse" id="adminNavbar">
26             <ul class="navbar-nav me-auto mb-2 mb-lg-0">
27
28                 <?php
29                 $navItems = [
30                     'dashboard.php' => 'Dashboard',
31                     'categories.php' => 'Categories',
32                     'products.php' => 'Products',
33                     'customers.php' => 'Customers',
34                     'orders.php' => 'Orders',
35                     'reports.php' => 'Reports',
36                     'backup.php' => 'Backup'
37                 ]
```

## Public (index.php):

```
221 <!-- HERO SECTION -->
222 <section class="hero">
223   <h1>Welcome to Jenny Store</h1>
224   <p>Elegant Imitation Jewelry & Skincare Essentials</p>
225 </section>
226
227 <!-- FEATURED PRODUCTS -->
228 <div class="container">
229   <h2>Featured Products</h2>
230   <div class="products">
231     <?php foreach ($featured as $p): ?>
232       <div class="product">
233         <?php if (!empty($p['image_url'])): ?>
234           " />
235         <?php endif; ?>
236         <h4><?php echo htmlspecialchars($p['product_name']); ?></h4>
237         <p><?php echo htmlspecialchars($p['category_name']); ?></p>
238         <p><strong>Rs. <?php echo number_format($p['price'], 2); ?></strong></p>
239         <a class="btn" href="../../customer/cart.php?add=<?php echo $p['product_id']; ?>">Add to Cart</a>
240       </div>
241     <?php endforeach; ?>
242   </div>
243 </div>
244
245 <!-- FOOTER -->
246 <footer>
247   <p><?php echo date('Y'); ?> Jenny Store. All Rights Reserved.
248 </footer>
249
250 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
251 </body>
252 </html>
```

## Public (logout.php):

```
public > 🐙 logout.php
1 <?php
2 session_start();
3 session_destroy();
4 header("Location: index.php");
5 exit;
```

## Customers (cart.php):

```
1  <?php
2  require_once "../includes/auth_customer.php";
3  require_once "../config/db.php";
4
5  $customer_id = $_SESSION['customer_id'];
6
7  /* -----
8   GET or CREATE CART
9  ----- */
10 $stmt = $conn->prepare("SELECT * FROM cart WHERE customer_id = ?");
11 $stmt->execute([$customer_id]);
12 $cart = $stmt->fetch(PDO::FETCH_ASSOC);
13
14 if (!$cart) {
15     $stmt = $conn->prepare("INSERT INTO cart (customer_id) VALUES (?)");
16     $stmt->execute([$customer_id]);
17     $cart_id = $conn->lastInsertId();
18 } else {
19     $cart_id = $cart['cart_id'];
20 }
21
22 /* -----
23 ADD PRODUCT TO CART
24 ----- */
25 if (isset($_GET['add'])) {
26     $product_id = intval($_GET['add']);
27
28     $stmt = $conn->prepare("SELECT * FROM cart_items WHERE cart_id = ? AND product_id = ?");
29     $stmt->execute([$cart_id, $product_id]);
30     $item = $stmt->fetch(PDO::FETCH_ASSOC);
31
32     if ($item) {
33         $stmt = $conn->prepare("UPDATE cart_items SET quantity = quantity + 1 WHERE cart_item_id = ?");
34         $stmt->execute([$item['cart_item_id']]);
35     } else {
36         $stmt = $conn->prepare("INSERT INTO cart_items (cart_id, product_id, quantity) VALUES (?, ?, 1)");
37         $stmt->execute([$cart_id, $product_id]);
38     }
39 }
```



## Customers (checkout.php):

```
1  <?php
2  require_once "../includes/auth_customer.php";
3  require_once "../config/db.php";
4
5  $customer_id = $_SESSION['customer_id'];
6
7  /* GET CUSTOMER DETAILS */
8  $stmt = $conn->prepare("SELECT * FROM customers WHERE customer_id = ?");
9  $stmt->execute([$customer_id]);
10 $customer = $stmt->fetch(PDO::FETCH_ASSOC);
11
12 /* GET CART */
13 $stmt = $conn->prepare("SELECT * FROM cart WHERE customer_id = ?");
14 $stmt->execute([$customer_id]);
15 $cart = $stmt->fetch(PDO::FETCH_ASSOC);
16
17 $cart_id = $cart['cart_id'] ?? null;
18
19 if (!$cart_id) {
20     die("Your cart is empty!");
21 }
22
23 /* FETCH CART ITEMS */
24 $stmt = $conn->prepare("
25     SELECT ci.cart_item_id, ci.product_id, ci.quantity, p.price, p.product_name
26     FROM cart_items ci
27     JOIN products p ON ci.product_id = p.product_id
28     WHERE ci.cart_id = ?
29 ");
30 $stmt->execute([$cart_id]);
31 $items = $stmt->fetchAll(PDO::FETCH_ASSOC);
32
33 /* PLACE ORDER */
34 if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['place_order'])) {
35
36     $total = 0;
37     foreach ($items as $item) {
```



# Customers (login.php):

```
1  <?php
2  session_start();
3  require_once "../config/db.php";
4
5  $error = "";
6
7  if ($_SERVER["REQUEST_METHOD"] === "POST") {
8      $email = trim($_POST['email']);
9      $password = trim($_POST['password']);
10
11      $stmt = $conn->prepare("SELECT * FROM customers WHERE email = ?");
12      $stmt->execute([$email]);
13      $customer = $stmt->fetch(PDO::FETCH_ASSOC);
14
15      if ($customer && password_verify($password, $customer['password'] ?? "")) {
16          $_SESSION['customer_id'] = $customer['customer_id'];
17          $_SESSION['customer_name'] = $customer['name'];
18          header("Location: products.php");
19          exit;
20      } else {
21          $error = "Invalid email or password";
22      }
23  }
24  ?>
25
26  <!DOCTYPE html>
27  <html lang="en">
28  <head>
29  <meta charset="UTF-8">
30  <title>Customer Login | Address Jewelers</title>
31
32  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
33  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap" rel="stylesheet">
34
```

## Customers (orders.php):

## Customers (products.php):

```
1  <?php
2  require_once "../includes/auth_customer.php";
3  require_once "../config/db.php";
4
5  $customer_name = $_SESSION['customer_name'] ?? "Customer";
6
7  /* FETCH CATEGORIES */
8  $catStmt = $conn->query("SELECT * FROM categories");
9  $categories = $catStmt->fetchAll(PDO::FETCH_ASSOC);
10
11 /* SEARCH & FILTER */
12 $where = [];
13 $params = [];
14
15 if (!empty($_GET['search'])) {
16     $where[] = "p.product_name LIKE ?";
17     $params[] = "%" . $_GET['search'] . "%";
18 }
19
20 if (!empty($_GET['category_id'])) {
21     $where[] = "p.category_id = ?";
22     $params[] = $_GET['category_id'];
23 }
24
25 $sql = "
26     SELECT p.*, c.category_name
27     FROM products p
28     JOIN categories c ON p.category_id = c.category_id
29     WHERE p.status = 'Available'
30 ";
31
32 if ($where) {
33     $sql .= " AND " . implode(" AND ", $where);
34 }
35
36 $stmt = $conn->prepare($sql);
37 $stmt->execute($params);
```

## Customers (register.php):

```
<div class="register-container">
<div class="register-card">

<h3 class="register-title">Create Your Account</h3>

<?php if ($error): ?>
<div class="error-msg"><?= htmlspecialchars($error); ?></div>
<?php endif; ?>

<form name="registerForm" method="post" onsubmit="return validateForm();">
<div class="row g-3">

<div class="col-md-12">
<label class="form-label">Full Name</label>
<input type="text" name="name" class="form-control" required minlength="3" value="<?= htmlspecialchars($_POST['name'] ?? '') ?>">
</div>

<div class="col-md-12">
<label class="form-label">Address</label>
<textarea name="address" class="form-control" required minlength="5"><?= htmlspecialchars($_POST['address'] ?? '') ?></textarea>
</div>

<div class="col-md-12">
<label class="form-label">Email Address</label>
<input type="email" name="email" class="form-control" required value="<?= htmlspecialchars($_POST['email'] ?? '') ?>">
</div>

<div class="col-md-6">
<label class="form-label">Work Phone</label>
<input type="text" name="work_phone" class="form-control" pattern="\d{11}" maxlength="11" inputmode="numeric" value="<?= htmlspecialchars($_POST['work_phone'] ?? '') ?>">
</div>

<div class="col-md-6">
<label class="form-label">Cell Phone</label>
```







**Includes** (auth\_footer.php):



