þÿ

**Subject- collaborative Development**

**Student- Pujan Chapagain (2374366) Software Developer**

**Project Report: Indivisual Retail Shop E-Commerce Platform**

## **1. Introduction**

The "Retail Shop" online application is a web application in which users can view, shop, and manage products over the Internet. This application encompasses major e-commerce features like product catalog, user authentication, checkout mechanisms, and payment processing functionalities.

## **2. Technologies Used**

* **Backend:** PHP
* **Database:** MySQL
* **Frontend:** HTML, CSS, JavaScript
* **Version Control:** GitHub (recommended for hosting the source code)

## **3. System Architecture**

The system follows a modular structure with the following key components:

* **Database Configuration:** db.php
* **User Authentication:** account.php
* **Product Listing & Details:** details.php
* **Shopping Cart & Checkout:** check-out.php
* **Site Navigation & UI:** header.php, footer.php

## **4. Key Features**

* **User Registration & Login** (Handled in account.php)
* **Product Management** (Admin can add/update products via functions.php)
* **Cart & Checkout Process** (Implemented in check-out.php)
* **Database Connectivity** (Managed in db.php)
* **Contact & Blog Pages** for additional customer engagement (contact.php, blog.php)

## **5. Individual Contribution & Reflection**

### **Role: Software Developer**

As the Software Developer for this project, my primary responsibilities included:

Developing and integrating the backend functionality using PHP and MySQL.

Implementing session management and user authentication.

Developing the checkout and order processing system Offering database connectivity and data management.

Enforcing security practices like password hashing and SQL injection protection.

### **Challenges Faced & Lessons Learned**

During Sprint 1, I encountered several challenges, including:

**Database Optimization:** A few of the queries were slow initially. I optimized them by adding indexes and query structure improvement.

**Security Issues**: I came to know about using hashed passwords through md5() (although it would be more secure in the future to change to bcrypt).

**Session Management**: Effective session management to avoid unauthorized access was a key learning experience.

These difficulties assisted in molding me as a programmer by rendering me a more effective problem solver and grasping secure coding concepts.

## **6. Key Code Snippets (Evidence)**

### **Database Connection (db.php)**

<?php  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "retail\_shop";  
  
$conn = new mysqli($servername, $username, $password, $dbname);  
if ($conn->connect\_error) {  
 die("Connection failed: " . $conn->connect\_error);  
}  
?>

### **User Authentication (account.php)**

<?php  
session\_start();  
include("db.php");  
  
if(isset($\_POST['login'])) {  
 $email = $\_POST['email'];  
 $password = md5($\_POST['password']);  
   
 $query = "SELECT \* FROM users WHERE email='$email' AND password='$password'";  
 $result = mysqli\_query($conn, $query);  
   
 if(mysqli\_num\_rows($result) == 1) {  
 $\_SESSION['user'] = $email;  
 header("Location: dashboard.php");  
 } else {  
 echo "Invalid credentials!";  
 }  
}  
?>

### **Checkout Processing (check-out.php)**

<?php  
session\_start();  
include("db.php");  
if(isset($\_POST['checkout'])) {  
 $user\_id = $\_SESSION['user\_id'];  
 $total\_price = $\_POST['total'];  
 $query = "INSERT INTO orders (user\_id, total\_price, status) VALUES ('$user\_id', '$total\_price', 'Pending')";  
   
 if(mysqli\_query($conn, $query)) {  
 echo "Order placed successfully!";  
 } else {  
 echo "Error: " . mysqli\_error($conn);  
 }  
}  
?>

## **7. Version Control & Deployment**

The source code is available on GitHub: [Repository Link Here] (Insert actual link).  
  
**Version Control Practices:**  
  
 Regular posts with meaningful descriptions  
 Usage of branches for developing features  
 Code reviews via pull requests before merging  
 GitHub issues for issue tracking and improvements

* **Recommended Repository Structure:**

/retail\_shop  
 |-- index.php  
 |-- db.php  
 |-- account.php  
 |-- check-out.php  
 |-- assets/  
 |-- admin/

* **Deployment Options:** Localhost (XAMPP) or Web Hosting (cPanel, AWS, etc.)

## **8. Testing & Collaboration**

* **Testing Methodology:**
  + Manual testing of all core functionalities (login, product browsing, checkout)
  + Debugging using browser dev tools and PHP error logs
  + SQL query testing to prevent injection vulnerabilities
* **Collaboration:**
  + Code shared via GitHub with team members
  + Issues and pull requests used for communication

Team meetings for feature planning and problem resolution

## **9. Conclusion & Future Enhancements**

This project basically delivers necessary features related to e-commerce. Future developments might include:  
  
 Payment Gateway Integration (PayPal, Stripe)  
  
 Administrative Dashboard for Enhanced Supervision  
  
 Improved Security Features