Ecommerce Lab

***Note: To run all the program: install xampp on your system or any php related package and when php command recognized by system use php -S 0.0.0.0:8080 and run in browser as localhost:8080***

**1. Write a program to create products category list and add items to the created categories in an e-commerce store. Display items with their categories. You can write program in any of the language as per your comfort. The design is not required. Database is optional,you can also work using array or any other data structure as per your comfort. Though if you opt to use database,you can.**

-> Solution:

| *<?php*  *// ecommerce.php*  *// This defines a class (a blueprint) for our e-commerce store*  *class EcommerceStore {*  *// Private variable to hold categories; only this class can access it directly*  *private $categories = [];*  *// This is the constructor, it runs when we create a new store object*  *public function \_\_construct() {*  *// Check if the session doesn't already have categories stored*  *if (!isset($\_SESSION['categories'])) {*  *// If not, start with an empty array in the session to store categories*  *$\_SESSION['categories'] = [];*  *}*  *// Link the class's categories to the session's categories so they stay in sync*  *$this->categories = &$\_SESSION['categories'];*  *}*  *// Function to create a new category*  *public function createCategory($categoryName) {*  *// Check if the category doesn’t already exist*  *if (!isset($this->categories[$categoryName])) {*  *// If it doesn’t exist, create it as an empty array to hold items*  *$this->categories[$categoryName] = [];*  *// Return true to show it worked*  *return true;*  *}*  *// If it already exists, return false (no duplicates allowed)*  *return false;*  *}*  *// Function to add an item to a category*  *public function addItem($categoryName, $item) {*  *// Check if the category doesn’t exist yet*  *if (!isset($this->categories[$categoryName])) {*  *// If it doesn’t, create it first*  *$this->createCategory($categoryName);*  *}*  *// Create an array with the item’s details*  *$itemDetails = [*  *// Use the current time as a unique ID for the item*  *'id' => time(),*  *// Store the item’s name from the input*  *'name' => $item['name'],*  *// Convert the price to a float (number with decimals)*  *'price' => floatval($item['price']),*  *// Convert the stock to an integer (whole number)*  *'stock' => intval($item['stock'])*  *];*  *// Add the item details to the category’s array*  *$this->categories[$categoryName][] = $itemDetails;*  *// Return true to show it worked*  *return true;*  *}*  *// Function to get all categories and their items*  *public function getCategories() {*  *// Simply return the categories array*  *return $this->categories;*  *}*  *}*  *// Start the session to keep data (like categories) between page loads*  *session\_start();*  *// Create a new store object using our class*  *$store = new EcommerceStore();*  *// Check if the page got data from a form submission (POST request)*  *if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {*  *// Make sure an action is specified in the form data*  *if (isset($\_POST['action'])) {*  *// Decide what to do based on the action value*  *switch ($\_POST['action']) {*  *// If the action is to add a category*  *case 'add\_category':*  *// Get the category name from the form and remove extra spaces*  *$categoryName = trim($\_POST['category\_name']);*  *// Check if the name isn’t empty*  *if (!empty($categoryName)) {*  *// Add the new category to the store*  *$store->createCategory($categoryName);*  *}*  *// Stop here for this action*  *break;*  *// If the action is to add an item*  *case 'add\_item':*  *// Get the category name for the item and remove extra spaces*  *$categoryName = trim($\_POST['item\_category']);*  *// Create an array with the item’s details from the form*  *$item = [*  *// Get the item name and remove extra spaces*  *'name' => trim($\_POST['item\_name']),*  *// Convert the price to a float*  *'price' => floatval($\_POST['item\_price']),*  *// Convert the stock to an integer*  *'stock' => intval($\_POST['item\_stock'])*  *];*  *// Check if the item has a name and a positive price*  *if (!empty($item['name']) && $item['price'] > 0) {*  *// Add the item to the specified category*  *$store->addItem($categoryName, $item);*  *}*  *// Stop here for this action*  *break;*  *}*  *}*  *}*  *?>*  *<!DOCTYPE html>*  *<!-- Start of the HTML part, which makes the webpage -->*  *<html>*  *<head>*  *<!-- Set the page title shown in the browser tab -->*  *<title>E-commerce Store</title>*  *<!-- Add some basic styling to make the page look nice -->*  *<style>*  */\* Style the whole page with a font and some spacing \*/*  *body { font-family: Arial, sans-serif; margin: 20px; }*  */\* Limit the width of the main content area and center it \*/*  *.container { max-width: 800px; margin: auto; }*  */\* Add spacing between sections \*/*  *.section { margin: 20px 0; }*  */\* Style each category box with a border and padding \*/*  *.category { border: 1px solid #ddd; padding: 10px; margin: 10px 0; }*  */\* Add spacing and indent to items inside categories \*/*  *.item { margin: 5px 0 5px 20px; }*  */\* Space out the forms \*/*  *form { margin: 10px 0; }*  */\* Add spacing between form inputs \*/*  *input { margin: 5px; }*  *</style>*  *</head>*  *<body>*  *<!-- Main content area -->*  *<div class="container">*  *<!-- Big title at the top -->*  *<h1>E-commerce Store Catalog</h1>*  *<!-- Section for adding a new category -->*  *<div class="section">*  *<!-- Smaller title for this section -->*  *<h2>Add New Category</h2>*  *<!-- Form to submit a new category -->*  *<form method="POST">*  *<!-- Hidden field to tell PHP this is an add\_category action -->*  *<input type="hidden" name="action" value="add\_category">*  *<!-- Text box for the category name -->*  *<input type="text" name="category\_name" placeholder="Category Name" required>*  *<!-- Button to submit the form -->*  *<input type="submit" value="Add Category">*  *</form>*  *</div>*  *<!-- Section for adding a new item -->*  *<div class="section">*  *<!-- Smaller title for this section -->*  *<h2>Add New Item</h2>*  *<!-- Form to submit a new item -->*  *<form method="POST">*  *<!-- Hidden field to tell PHP this is an add\_item action -->*  *<input type="hidden" name="action" value="add\_item">*  *<!-- Dropdown to pick a category -->*  *<select name="item\_category" required>*  *<!-- Default option prompting the user to select -->*  *<option value="">Select Category</option>*  *<!-- Loop through all existing categories -->*  *<?php foreach (array\_keys($store->getCategories()) as $cat): ?>*  *<!-- Add each category as an option in the dropdown -->*  *<option value="<?php echo htmlspecialchars($cat); ?>">*  *<?php echo htmlspecialchars($cat); ?>*  *</option>*  *<?php endforeach; ?>*  *</select>*  *<!-- Text box for the item name -->*  *<input type="text" name="item\_name" placeholder="Item Name" required>*  *<!-- Number input for the price, allowing decimals -->*  *<input type="number" name="item\_price" placeholder="Price" step="0.01" min="0" required>*  *<!-- Number input for the stock, whole numbers only -->*  *<input type="number" name="item\_stock" placeholder="Stock" min="0" required>*  *<!-- Button to submit the form -->*  *<input type="submit" value="Add Item">*  *</form>*  *</div>*  *<!-- Section to display the catalog -->*  *<div class="section">*  *<!-- Smaller title for this section -->*  *<h2>Catalog</h2>*  *<!-- Get all categories and items from the store -->*  *<?php $categories = $store->getCategories(); ?>*  *<!-- Check if there are no categories yet -->*  *<?php if (empty($categories)): ?>*  *<!-- Show a message if the catalog is empty -->*  *<p>No categories or items yet.</p>*  *<?php else: ?>*  *<!-- Loop through each category -->*  *<?php foreach ($categories as $categoryName => $items): ?>*  *<!-- Display the category in a styled box -->*  *<div class="category">*  *<!-- Show the category name and how many items it has -->*  *<h3><?php echo htmlspecialchars($categoryName); ?>*  *(<?php echo count($items); ?> items)</h3>*  *<!-- Loop through each item in the category -->*  *<?php foreach ($items as $item): ?>*  *<!-- Display the item details -->*  *<div class="item">*  *<!-- Show the item name in bold -->*  *<strong><?php echo htmlspecialchars($item['name']); ?></strong>*  *<!-- Show the item’s unique ID -->*  *(ID: <?php echo $item['id']; ?>)<br>*  *<!-- Show the price, formatted with 2 decimals -->*  *Price: Rs.<?php echo number\_format($item['price'], 2); ?><br>*  *<!-- Show the stock amount -->*  *Stock: <?php echo $item['stock']; ?> units*  *</div>*  *<?php endforeach; ?>*  *</div>*  *<?php endforeach; ?>*  *<?php endif; ?>*  *</div>*  *</div>*  *</body>*  *</html>* |
| --- |

**Lab 2: Write a program to create product list and add to cart feature for the products in an e-commerce store and display cart items. You can write program in any of the language as per your comfort. The design is not required. Database is optional,you can also work using array or any other data structure as per your comfort. Though if you opt to use database,you can.**

-> Solution:

| *<?php*  *// Start the session to store products and cart data across page loads*  *session\_start();*  *// Check if products array doesn't exist in session, initialize it as empty*  *if (!isset($\_SESSION['products'])) {*  *$\_SESSION['products'] = [];*  *}*  *// Check if cart array doesn't exist in session, initialize it as empty*  *if (!isset($\_SESSION['cart'])) {*  *$\_SESSION['cart'] = [];*  *}*  *// Class to manage our e-commerce store*  *class EcommerceStore {*  *// Private variables to hold references to products and cart*  *private $products;*  *private $cart;*  *// Constructor runs when we create a new store object*  *public function \_\_construct() {*  *// Link the class variables to session data so they stay updated*  *$this->products = &$\_SESSION['products'];*  *$this->cart = &$\_SESSION['cart'];*  *}*  *// Function to add a new product to the product list*  *public function addProduct($name, $price) {*  *// Create a new product array with a unique ID (using timestamp for simplicity)*  *$product = [*  *'id' => time(), // Simple unique ID (not ideal for production)*  *'name' => $name,*  *'price' => floatval($price) // Ensure price is a float*  *];*  *// Add the product to the products array*  *$this->products[] = $product;*  *// Return true to show it worked*  *return true;*  *}*  *// Function to get all products*  *public function getProducts() {*  *// Return the products array*  *return $this->products;*  *}*  *// Function to add a product to the cart*  *public function addToCart($productId, $quantity) {*  *// Loop through products to find the one with matching ID*  *foreach ($this->products as $product) {*  *if ($product['id'] == $productId) {*  *// Check if this product is already in the cart*  *if (isset($this->cart[$productId])) {*  *// If yes, increase the quantity in the cart*  *$this->cart[$productId]['quantity'] += $quantity;*  *} else {*  *// If not, add it with the quantity*  *$this->cart[$productId] = [*  *'name' => $product['name'],*  *'price' => $product['price'],*  *'quantity' => $quantity*  *];*  *}*  *// Return true to show it worked*  *return true;*  *}*  *}*  *// Return false if product ID wasn’t found*  *return false;*  *}*  *// Function to get all items in the cart*  *public function getCart() {*  *// Return the cart array*  *return $this->cart;*  *}*  *// Function to clear the cart*  *public function clearCart() {*  *// Empty the cart array*  *$this->cart = [];*  *}*  *}*  *// Create a new store object*  *$store = new EcommerceStore();*  *// Handle form submissions (POST requests)*  *if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {*  *// Check if the action is to add a product*  *if (isset($\_POST['action']) && $\_POST['action'] === 'add\_product') {*  *// Get product details from the form*  *$name = trim($\_POST['product\_name']);*  *$price = floatval($\_POST['product\_price']);*  *// Check if inputs are valid*  *if (!empty($name) && $price > 0) {*  *// Add the product to the store*  *$store->addProduct($name, $price);*  *$message = "Product '$name' added!";*  *} else {*  *$message = "Invalid product details.";*  *}*  *}*  *// Check if the action is to add to cart*  *if (isset($\_POST['action']) && $\_POST['action'] === 'add\_to\_cart') {*  *// Get the product ID and quantity from the form*  *$productId = intval($\_POST['product\_id']);*  *$quantity = intval($\_POST['quantity']);*  *// Make sure quantity is positive*  *if ($quantity > 0) {*  *// Try to add the item to the cart*  *$result = $store->addToCart($productId, $quantity);*  *// Set a message based on the result*  *if ($result) {*  *$message = "Item added to cart!";*  *} else {*  *$message = "Failed to add item (invalid ID).";*  *}*  *} else {*  *$message = "Quantity must be greater than 0.";*  *}*  *}*  *// Check if the action is to clear the cart*  *if (isset($\_POST['action']) && $\_POST['action'] === 'clear\_cart') {*  *// Clear the cart*  *$store->clearCart();*  *$message = "Cart cleared!";*  *}*  *}*  *?>*  *<!-- Start of the HTML part -->*  *<!DOCTYPE html>*  *<html>*  *<head>*  *<!-- Set the page title -->*  *<title>E-commerce Store</title>*  *</head>*  *<body>*  *<!-- Main heading -->*  *<h1>E-commerce Store</h1>*  *<!-- Display any messages (like "Product added!" or "Item added to cart!") -->*  *<?php if (isset($message)): ?>*  *<p><?php echo $message; ?></p>*  *<?php endif; ?>*  *<!-- Add Product Section -->*  *<h2>Add New Product</h2>*  *<!-- Form to add a new product -->*  *<form method="POST">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="add\_product">*  *<!-- Input for product name -->*  *<input type="text" name="product\_name" placeholder="Product Name" required>*  *<!-- Input for price (decimal allowed) -->*  *<input type="number" name="product\_price" placeholder="Price" step="0.01" min="0" required>*  *<!-- Submit button -->*  *<input type="submit" value="Add Product">*  *</form>*  *<!-- Product List Section -->*  *<h2>Products</h2>*  *<!-- Get all products -->*  *<?php $products = $store->getProducts(); ?>*  *<!-- Check if there are no products -->*  *<?php if (empty($products)): ?>*  *<p>No products available.</p>*  *<?php else: ?>*  *<!-- Loop through all products -->*  *<?php foreach ($products as $product): ?>*  *<!-- Show product details -->*  *<div>*  *<!-- Product name and price with Rs. symbol -->*  *<strong><?php echo $product['name']; ?></strong> -*  *Rs. <?php echo number\_format($product['price'], 2); ?>*  *<!-- Form to add this product to cart -->*  *<form method="POST" style="display:inline;">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="add\_to\_cart">*  *<!-- Hidden field for the product ID -->*  *<input type="hidden" name="product\_id" value="<?php echo $product['id']; ?>">*  *<!-- Input for quantity -->*  *<input type="number" name="quantity" min="1" value="1" style="width:50px;">*  *<!-- Submit button -->*  *<input type="submit" value="Add to Cart">*  *</form>*  *</div>*  *<?php endforeach; ?>*  *<?php endif; ?>*  *<!-- Cart Section -->*  *<h2>Cart</h2>*  *<!-- Get the cart items -->*  *<?php $cartItems = $store->getCart(); ?>*  *<!-- Check if the cart is empty -->*  *<?php if (empty($cartItems)): ?>*  *<!-- Show message if no items -->*  *<p>Your cart is empty.</p>*  *<?php else: ?>*  *<!-- Loop through cart items -->*  *<?php foreach ($cartItems as $productId => $item): ?>*  *<!-- Show cart item details with Rs. symbol -->*  *<div>*  *<!-- Item name, quantity, and total price -->*  *<?php echo $item['name']; ?> -*  *Quantity: <?php echo $item['quantity']; ?> -*  *Total: Rs. <?php echo number\_format($item['price'] \* $item['quantity'], 2); ?>*  *</div>*  *<?php endforeach; ?>*  *<!-- Form to clear the cart -->*  *<form method="POST">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="clear\_cart">*  *<!-- Submit button to clear cart -->*  *<input type="submit" value="Clear Cart">*  *</form>*  *<?php endif; ?>*  *</body>*  *</html>* |
| --- |

**Lab 3: Write a program to implement add to cart and checkout feature of e-commerce shopping store and display the ordered list of items. You can write program in any of the language as per your comfort. The design is not required. database is optional,you can also work using array or any other data structure as**

**per your comfort. Though if you opt to use database,you can.**

-> Solution:

| *<?php*  *// Start the session to store cart and order data across page loads*  *session\_start();*  *// Check if products array doesn't exist in session, initialize it with sample products*  *if (!isset($\_SESSION['products'])) {*  *// Predefined products with id, name, and price (no stock)*  *$\_SESSION['products'] = [*  *['id' => 1, 'name' => 'Laptop', 'price' => 45000.00],*  *['id' => 2, 'name' => 'Phone', 'price' => 20000.00],*  *['id' => 3, 'name' => 'Headphones', 'price' => 1500.00]*  *];*  *}*  *// Check if cart array doesn't exist in session, initialize it as empty*  *if (!isset($\_SESSION['cart'])) {*  *$\_SESSION['cart'] = [];*  *}*  *// Check if orders array doesn't exist in session, initialize it as empty*  *if (!isset($\_SESSION['orders'])) {*  *$\_SESSION['orders'] = [];*  *}*  *// Class to manage our e-commerce store*  *class EcommerceStore {*  *// Private variables to hold references to products, cart, and orders*  *private $products;*  *private $cart;*  *private $orders;*  *// Constructor runs when we create a new store object*  *public function \_\_construct() {*  *// Link the class variables to session data so they stay updated*  *$this->products = &$\_SESSION['products'];*  *$this->cart = &$\_SESSION['cart'];*  *$this->orders = &$\_SESSION['orders'];*  *}*  *// Function to get all products*  *public function getProducts() {*  *// Return the products array*  *return $this->products;*  *}*  *// Function to add a product to the cart*  *public function addToCart($productId, $quantity) {*  *// Loop through products to find the one with matching ID*  *foreach ($this->products as $product) {*  *if ($product['id'] == $productId) {*  *// Check if this product is already in the cart*  *if (isset($this->cart[$productId])) {*  *// If yes, increase the quantity*  *$this->cart[$productId]['quantity'] += $quantity;*  *} else {*  *// If not, add it with the quantity*  *$this->cart[$productId] = [*  *'name' => $product['name'],*  *'price' => $product['price'],*  *'quantity' => $quantity*  *];*  *}*  *// Return true to show it worked*  *return true;*  *}*  *}*  *// Return false if product ID wasn’t found*  *return false;*  *}*  *// Function to get all items in the cart*  *public function getCart() {*  *// Return the cart array*  *return $this->cart;*  *}*  *// Function to checkout (move cart items to orders and clear cart)*  *public function checkout() {*  *// Check if the cart is not empty*  *if (!empty($this->cart)) {*  *// Create an order array with a timestamp and the cart items*  *$order = [*  *'order\_id' => time(), // Use timestamp as a simple order ID*  *'items' => $this->cart,*  *'timestamp' => date('Y-m-d H:i:s') // Record the date and time*  *];*  *// Add the order to the orders array*  *$this->orders[] = $order;*  *// Clear the cart after checkout*  *$this->cart = [];*  *// Return true to show it worked*  *return true;*  *}*  *// Return false if cart was empty*  *return false;*  *}*  *// Function to get all orders*  *public function getOrders() {*  *// Return the orders array*  *return $this->orders;*  *}*  *}*  *// Create a new store object*  *$store = new EcommerceStore();*  *// Handle form submissions (POST requests)*  *if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {*  *// Check if the action is to add to cart*  *if (isset($\_POST['action']) && $\_POST['action'] === 'add\_to\_cart') {*  *// Get the product ID and quantity from the form*  *$productId = intval($\_POST['product\_id']);*  *$quantity = intval($\_POST['quantity']);*  *// Make sure quantity is positive*  *if ($quantity > 0) {*  *// Try to add the item to the cart*  *$result = $store->addToCart($productId, $quantity);*  *// Set a message based on the result*  *if ($result) {*  *$message = "Item added to cart!";*  *} else {*  *$message = "Failed to add item (invalid ID).";*  *}*  *} else {*  *$message = "Quantity must be greater than 0.";*  *}*  *}*  *// Check if the action is to checkout*  *if (isset($\_POST['action']) && $\_POST['action'] === 'checkout') {*  *// Try to process the checkout*  *$result = $store->checkout();*  *// Set a message based on the result*  *if ($result) {*  *$message = "Checkout successful! Order placed.";*  *} else {*  *$message = "Cart is empty. Add items before checkout.";*  *}*  *}*  *}*  *?>*  *<!-- Start of the HTML part -->*  *<!DOCTYPE html>*  *<html>*  *<head>*  *<!-- Set the page title -->*  *<title>E-commerce Store</title>*  *</head>*  *<body>*  *<!-- Main heading -->*  *<h1>E-commerce Store</h1>*  *<!-- Display any messages (like "Item added to cart!" or "Checkout successful!") -->*  *<?php if (isset($message)): ?>*  *<p><?php echo $message; ?></p>*  *<?php endif; ?>*  *<!-- Product List Section -->*  *<h2>Products</h2>*  *<!-- Loop through all products -->*  *<?php foreach ($store->getProducts() as $product): ?>*  *<!-- Show product details -->*  *<div>*  *<!-- Product name and price with Rs. symbol -->*  *<strong><?php echo $product['name']; ?></strong> -*  *Rs. <?php echo number\_format($product['price'], 2); ?>*  *<!-- Form to add this product to cart -->*  *<form method="POST" style="display:inline;">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="add\_to\_cart">*  *<!-- Hidden field for the product ID -->*  *<input type="hidden" name="product\_id" value="<?php echo $product['id']; ?>">*  *<!-- Input for quantity -->*  *<input type="number" name="quantity" min="1" value="1" style="width:50px;">*  *<!-- Submit button -->*  *<input type="submit" value="Add to Cart">*  *</form>*  *</div>*  *<?php endforeach; ?>*  *<!-- Cart Section -->*  *<h2>Cart</h2>*  *<!-- Get the cart items -->*  *<?php $cartItems = $store->getCart(); ?>*  *<!-- Check if the cart is empty -->*  *<?php if (empty($cartItems)): ?>*  *<!-- Show message if no items -->*  *<p>Your cart is empty.</p>*  *<?php else: ?>*  *<!-- Loop through cart items -->*  *<?php foreach ($cartItems as $productId => $item): ?>*  *<!-- Show cart item details with Rs. symbol -->*  *<div>*  *<!-- Item name, quantity, and total price -->*  *<?php echo $item['name']; ?> -*  *Quantity: <?php echo $item['quantity']; ?> -*  *Total: Rs. <?php echo number\_format($item['price'] \* $item['quantity'], 2); ?>*  *</div>*  *<?php endforeach; ?>*  *<!-- Form to checkout -->*  *<form method="POST">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="checkout">*  *<!-- Submit button to checkout -->*  *<input type="submit" value="Checkout">*  *</form>*  *<?php endif; ?>*  *<!-- Orders Section -->*  *<h2>Ordered Items</h2>*  *<!-- Get all orders -->*  *<?php $orders = $store->getOrders(); ?>*  *<!-- Check if there are no orders -->*  *<?php if (empty($orders)): ?>*  *<!-- Show message if no orders -->*  *<p>No orders placed yet.</p>*  *<?php else: ?>*  *<!-- Loop through all orders -->*  *<?php foreach ($orders as $order): ?>*  *<!-- Show order details -->*  *<div>*  *<!-- Order ID and timestamp -->*  *<strong>Order ID: <?php echo $order['order\_id']; ?></strong> -*  *Placed on: <?php echo $order['timestamp']; ?>*  *<!-- Loop through items in this order -->*  *<?php foreach ($order['items'] as $productId => $item): ?>*  *<!-- Show item details with Rs. symbol -->*  *<div>*  *<?php echo $item['name']; ?> -*  *Quantity: <?php echo $item['quantity']; ?> -*  *Total: Rs. <?php echo number\_format($item['price'] \* $item['quantity'], 2); ?>*  *</div>*  *<?php endforeach; ?>*  *</div>*  *<hr> <!-- Horizontal line to separate orders -->*  *<?php endforeach; ?>*  *<?php endif; ?>*  *</body>*  *</html>* |
| --- |

**Lab 4: Write a program to create product list and add to wish list feature for the products in ecommerce store and display the wish list items. You can write program in any of the language as per your comfort. The design is not required. Database is optional, you can also work using array or any other data structure as per your comfort. Though if you opt to use database, you can.**

-> Solution:

| *<?php*  *// Start the session to store products and wishlist data across page loads*  *session\_start();*  *// Check if products array doesn't exist in session, initialize it as empty*  *if (!isset($\_SESSION['products'])) {*  *$\_SESSION['products'] = [];*  *}*  *// Check if wishlist array doesn't exist in session, initialize it as empty*  *if (!isset($\_SESSION['wishlist'])) {*  *$\_SESSION['wishlist'] = [];*  *}*  *// Class to manage our e-commerce store*  *class EcommerceStore {*  *// Private variables to hold references to products and wishlist*  *private $products;*  *private $wishlist;*  *// Constructor runs when we create a new store object*  *public function \_\_construct() {*  *// Link the class variables to session data so they stay updated*  *$this->products = &$\_SESSION['products'];*  *$this->wishlist = &$\_SESSION['wishlist'];*  *}*  *// Function to add a new product to the product list*  *public function addProduct($name, $price) {*  *// Create a new product array with a unique ID (using timestamp for simplicity)*  *$product = [*  *'id' => time(), // Simple unique ID (not ideal for production)*  *'name' => $name,*  *'price' => floatval($price) // Ensure price is a float*  *];*  *// Add the product to the products array*  *$this->products[] = $product;*  *// Return true to show it worked*  *return true;*  *}*  *// Function to get all products*  *public function getProducts() {*  *// Return the products array*  *return $this->products;*  *}*  *// Function to add a product to the wishlist*  *public function addToWishlist($productId) {*  *// Loop through products to find the one with matching ID*  *foreach ($this->products as $product) {*  *if ($product['id'] == $productId) {*  *// Check if this product is already in the wishlist*  *if (!isset($this->wishlist[$productId])) {*  *// If not, add it to the wishlist (no quantity, just the product details)*  *$this->wishlist[$productId] = [*  *'name' => $product['name'],*  *'price' => $product['price']*  *];*  *}*  *// Return true to show it worked*  *return true;*  *}*  *}*  *// Return false if product ID wasn’t found*  *return false;*  *}*  *// Function to get all items in the wishlist*  *public function getWishlist() {*  *// Return the wishlist array*  *return $this->wishlist;*  *}*  *// Function to remove an item from the wishlist*  *public function removeFromWishlist($productId) {*  *// Check if the product exists in the wishlist*  *if (isset($this->wishlist[$productId])) {*  *// Remove it from the wishlist*  *unset($this->wishlist[$productId]);*  *// Return true to show it worked*  *return true;*  *}*  *// Return false if product wasn’t in the wishlist*  *return false;*  *}*  *}*  *// Create a new store object*  *$store = new EcommerceStore();*  *// Handle form submissions (POST requests)*  *if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {*  *// Check if the action is to add a product*  *if (isset($\_POST['action']) && $\_POST['action'] === 'add\_product') {*  *// Get product details from the form*  *$name = trim($\_POST['product\_name']);*  *$price = floatval($\_POST['product\_price']);*  *// Check if inputs are valid*  *if (!empty($name) && $price > 0) {*  *// Add the product to the store*  *$store->addProduct($name, $price);*  *$message = "Product '$name' added!";*  *} else {*  *$message = "Invalid product details.";*  *}*  *}*  *// Check if the action is to add to wishlist*  *if (isset($\_POST['action']) && $\_POST['action'] === 'add\_to\_wishlist') {*  *// Get the product ID from the form*  *$productId = intval($\_POST['product\_id']);*  *// Try to add the item to the wishlist*  *$result = $store->addToWishlist($productId);*  *// Set a message based on the result*  *if ($result) {*  *$message = "Item added to wishlist!";*  *} else {*  *$message = "Failed to add item (invalid ID).";*  *}*  *}*  *// Check if the action is to remove from wishlist*  *if (isset($\_POST['action']) && $\_POST['action'] === 'remove\_from\_wishlist') {*  *// Get the product ID from the form*  *$productId = intval($\_POST['product\_id']);*  *// Try to remove the item from the wishlist*  *$result = $store->removeFromWishlist($productId);*  *// Set a message based on the result*  *if ($result) {*  *$message = "Item removed from wishlist!";*  *} else {*  *$message = "Item not found in wishlist.";*  *}*  *}*  *}*  *?>*  *<!-- Start of the HTML part -->*  *<!DOCTYPE html>*  *<html>*  *<head>*  *<!-- Set the page title -->*  *<title>E-commerce Store</title>*  *</head>*  *<body>*  *<!-- Main heading -->*  *<h1>E-commerce Store</h1>*  *<!-- Display any messages (like "Product added!" or "Item added to wishlist!") -->*  *<?php if (isset($message)): ?>*  *<p><?php echo $message; ?></p>*  *<?php endif; ?>*  *<!-- Add Product Section -->*  *<h2>Add New Product</h2>*  *<!-- Form to add a new product -->*  *<form method="POST">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="add\_product">*  *<!-- Input for product name -->*  *<input type="text" name="product\_name" placeholder="Product Name" required>*  *<!-- Input for price (decimal allowed) -->*  *<input type="number" name="product\_price" placeholder="Price" step="0.01" min="0" required>*  *<!-- Submit button -->*  *<input type="submit" value="Add Product">*  *</form>*  *<!-- Product List Section -->*  *<h2>Products</h2>*  *<!-- Get all products -->*  *<?php $products = $store->getProducts(); ?>*  *<!-- Check if there are no products -->*  *<?php if (empty($products)): ?>*  *<p>No products available.</p>*  *<?php else: ?>*  *<!-- Loop through all products -->*  *<?php foreach ($products as $product): ?>*  *<!-- Show product details -->*  *<div>*  *<!-- Product name and price with Rs. symbol -->*  *<strong><?php echo $product['name']; ?></strong> -*  *Rs. <?php echo number\_format($product['price'], 2); ?>*  *<!-- Form to add this product to wishlist -->*  *<form method="POST" style="display:inline;">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="add\_to\_wishlist">*  *<!-- Hidden field for the product ID -->*  *<input type="hidden" name="product\_id" value="<?php echo $product['id']; ?>">*  *<!-- Submit button -->*  *<input type="submit" value="Add to Wishlist">*  *</form>*  *</div>*  *<?php endforeach; ?>*  *<?php endif; ?>*  *<!-- Wishlist Section -->*  *<h2>Wishlist</h2>*  *<!-- Get the wishlist items -->*  *<?php $wishlistItems = $store->getWishlist(); ?>*  *<!-- Check if the wishlist is empty -->*  *<?php if (empty($wishlistItems)): ?>*  *<!-- Show message if no items -->*  *<p>Your wishlist is empty.</p>*  *<?php else: ?>*  *<!-- Loop through wishlist items -->*  *<?php foreach ($wishlistItems as $productId => $item): ?>*  *<!-- Show wishlist item details with Rs. symbol -->*  *<div>*  *<!-- Item name and price -->*  *<?php echo $item['name']; ?> -*  *Rs. <?php echo number\_format($item['price'], 2); ?>*  *<!-- Form to remove this item from wishlist -->*  *<form method="POST" style="display:inline;">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="remove\_from\_wishlist">*  *<!-- Hidden field for the product ID -->*  *<input type="hidden" name="product\_id" value="<?php echo $productId; ?>">*  *<!-- Submit button -->*  *<input type="submit" value="Remove">*  *</form>*  *</div>*  *<?php endforeach; ?>*  *<?php endif; ?>*  *</body>*  *</html>* |
| --- |

**Lab 5: Write a program to integrate payment gateway of your choice in an ecommerce page.**

-> Solution (Using ESEWA):

NOTE : Use Below Demo Credential for Esewa  
**eSewa ID:** 9806800001/2/3/4/5

**Password:** Nepal@123 **MPIN:** 1122 **Token:**123456

| *<?php*  *// Start the session to store cart data across page loads*  *session\_start();*  *// Check if cart array doesn't exist in session, initialize it as empty*  *if (!isset($\_SESSION['cart'])) {*  *$\_SESSION['cart'] = [];*  *}*  *// Predefined product list (in a real app, this could come from a database)*  *$products = [*  *['id' => 1, 'name' => 'Laptop', 'price' => 45000.00],*  *['id' => 2, 'name' => 'Phone', 'price' => 20000.00],*  *['id' => 3, 'name' => 'Headphones', 'price' => 1500.00]*  *];*  *// Function to generate HMAC-SHA256 signature as required by eSewa*  *function generateSignature($data, $secretKey) {*  *// Create the HMAC-SHA256 hash using the data and secret key*  *$hmac = hash\_hmac('sha256', $data, $secretKey, true);*  *// Convert the hash to base64 format (eSewa requires this)*  *return base64\_encode($hmac);*  *}*  *// Handle form submissions (POST requests)*  *if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {*  *// Check if the action is to add to cart*  *if (isset($\_POST['action']) && $\_POST['action'] === 'add\_to\_cart') {*  *// Get the product ID and quantity from the form*  *$productId = intval($\_POST['product\_id']);*  *$quantity = intval($\_POST['quantity']);*  *// Make sure quantity is positive*  *if ($quantity > 0) {*  *// Find the product in the product list*  *foreach ($products as $product) {*  *if ($product['id'] == $productId) {*  *// Check if this product is already in the cart*  *if (isset($\_SESSION['cart'][$productId])) {*  *// If yes, increase the quantity*  *$\_SESSION['cart'][$productId]['quantity'] += $quantity;*  *} else {*  *// If not, add it with the quantity*  *$\_SESSION['cart'][$productId] = [*  *'name' => $product['name'],*  *'price' => $product['price'],*  *'quantity' => $quantity*  *];*  *}*  *$message = "Item added to cart!";*  *break;*  *}*  *}*  *} else {*  *$message = "Quantity must be greater than 0.";*  *}*  *}*  *// Check if the action is to initiate payment (checkout)*  *if (isset($\_POST['action']) && $\_POST['action'] === 'checkout') {*  *// Check if the cart is not empty*  *if (!empty($\_SESSION['cart'])) {*  *// Calculate the total amount from the cart*  *$totalAmount = 0;*  *foreach ($\_SESSION['cart'] as $item) {*  *$totalAmount += $item['price'] \* $item['quantity'];*  *}*  *// eSewa configuration (test credentials from the reference link)*  *$esewaUrl = "https://rc-epay.esewa.com.np/api/epay/main/v2/form"; // Test server URL*  *$merchantCode = "EPAYTEST"; // Test merchant code*  *$secretKey = "8gBm/:&EnhH.1/q"; // Test secret key*  *$successUrl = "http://localhost/esewa-success.php"; // Success redirect URL*  *$failureUrl = "http://localhost/esewa-failure.php"; // Failure redirect URL*  *$transactionUuid = uniqid(); // Unique transaction ID (using PHP's uniqid)*  *// Prepare the data string for signature (order must match signed\_field\_names)*  *$data = "total\_amount={$totalAmount},transaction\_uuid={$transactionUuid},product\_code={$merchantCode}";*  *// Generate the signature using the secret key*  *$signature = generateSignature($data, $secretKey);*  *// Redirect to eSewa payment form (this will be output in HTML)*  *$esewaForm = "*  *<form id='esewa\_form' action='{$esewaUrl}' method='POST'>*  *<input type='hidden' name='amount' value='{$totalAmount}'>*  *<input type='hidden' name='tax\_amount' value='0'>*  *<input type='hidden' name='total\_amount' value='{$totalAmount}'>*  *<input type='hidden' name='transaction\_uuid' value='{$transactionUuid}'>*  *<input type='hidden' name='product\_code' value='{$merchantCode}'>*  *<input type='hidden' name='product\_service\_charge' value='0'>*  *<input type='hidden' name='product\_delivery\_charge' value='0'>*  *<input type='hidden' name='success\_url' value='{$successUrl}'>*  *<input type='hidden' name='failure\_url' value='{$failureUrl}'>*  *<input type='hidden' name='signed\_field\_names' value='total\_amount,transaction\_uuid,product\_code'>*  *<input type='hidden' name='signature' value='{$signature}'>*  *</form>*  *<script>document.getElementById('esewa\_form').submit();</script>*  *";*  *// Output the form and exit to redirect immediately*  *echo $esewaForm;*  *exit;*  *} else {*  *$message = "Cart is empty. Add items before checkout.";*  *}*  *}*  *}*  *?>*  *<!-- Start of the HTML part -->*  *<!DOCTYPE html>*  *<html>*  *<head>*  *<!-- Set the page title -->*  *<title>E-commerce Store with eSewa</title>*  *</head>*  *<body>*  *<!-- Main heading -->*  *<h1>E-commerce Store</h1>*  *<!-- Display any messages (like "Item added to cart!") -->*  *<?php if (isset($message)): ?>*  *<p><?php echo $message; ?></p>*  *<?php endif; ?>*  *<!-- Product List Section -->*  *<h2>Products</h2>*  *<!-- Loop through all products -->*  *<?php foreach ($products as $product): ?>*  *<!-- Show product details -->*  *<div>*  *<!-- Product name and price with Rs. symbol -->*  *<strong><?php echo $product['name']; ?></strong> -*  *Rs. <?php echo number\_format($product['price'], 2); ?>*  *<!-- Form to add this product to cart -->*  *<form method="POST" style="display:inline;">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="add\_to\_cart">*  *<!-- Hidden field for the product ID -->*  *<input type="hidden" name="product\_id" value="<?php echo $product['id']; ?>">*  *<!-- Input for quantity -->*  *<input type="number" name="quantity" min="1" value="1" style="width:50px;">*  *<!-- Submit button -->*  *<input type="submit" value="Add to Cart">*  *</form>*  *</div>*  *<?php endforeach; ?>*  *<!-- Cart Section -->*  *<h2>Cart</h2>*  *<!-- Get the cart items -->*  *<?php $cartItems = $\_SESSION['cart']; ?>*  *<!-- Check if the cart is empty -->*  *<?php if (empty($cartItems)): ?>*  *<!-- Show message if no items -->*  *<p>Your cart is empty.</p>*  *<?php else: ?>*  *<!-- Loop through cart items -->*  *<?php foreach ($cartItems as $productId => $item): ?>*  *<!-- Show cart item details with Rs. symbol -->*  *<div>*  *<!-- Item name, quantity, and total price -->*  *<?php echo $item['name']; ?> -*  *Quantity: <?php echo $item['quantity']; ?> -*  *Total: Rs. <?php echo number\_format($item['price'] \* $item['quantity'], 2); ?>*  *</div>*  *<?php endforeach; ?>*  *<!-- Form to initiate checkout with eSewa -->*  *<form method="POST">*  *<!-- Hidden field for the action -->*  *<input type="hidden" name="action" value="checkout">*  *<!-- Submit button to checkout -->*  *<input type="submit" value="Pay with eSewa">*  *</form>*  *<?php endif; ?>*  *</body>*  *</html>* |
| --- |