

**MIT Art, Design and Technology University MIT School of Computing, Pune**

**Department of Information Technology**

**Lab Manual**

**Practical - Web Programming** **Class - S.Y. (SEM-II), DA**

**Batch - DA**

**Name of the Student**

**Vedant Suryawanshi**

**A.Y. 2024 – 2025 (SEM-II)**

# Experiment No. 1

## Problem Statement:

Design the layout and structure of an E-commerce website for electronics and wearable products.

## Objective:

To plan and document the project architecture, page structure, design elements, and user experience.

## Theory:

* Project Introduction
* Target audience: tech-savvy consumers
* Pages: Home, Products, About, Reviews, Contact, Login/Register
* Features: Product listing, cart, checkout, admin panel
* Visual Style: Tech-themed color palette (dark blue, white, neon accents)

## Code:

**index.html (Basic Structure)**

html CopyEdit

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>TechGadgets - Home</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/6.4.0/css/all.min.css" />

<link rel="stylesheet" href="style.css" />

</head>

<body>

<!-- Header Top Banner -->

<div class="header-top">

<div class="container">

Free shipping on orders above 500 | 30-day money-back guarantee

</div>

</div>

<!-- Navbar (Header Section) -->

<header>

<div class="container">

<nav>

<a href="index.html" class="logo">

<i class="fas fa-bolt"></i>TechGadgets

</a>

<ul class="nav-links">

<li><a href="index.html">Home</a></li>

<li><a href="products.html">Products</a></li>

<li><a href="about.html">About</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

<div class="nav-icons">

<a href="login.html"><i class="fas fa-user"></i></a>

<a href="cart.php">

<i class="fas fa-shopping-cart"></i>

<span class="cart-count">0</span>

</a>

</div>

</nav>

</div>

</header>

<!-- Main Content Area -->

<main class="container">

<!-- Hero Section -->

<section class="hero">

<div class="hero-content">

<h1>Summer Sale: Up to 40% Off!</h1>

<p>Get the latest tech gadgets at unbeatable prices. Limited time offer.</p>

<a href="products.html" class="btn">Shop Now</a>

</div>

<img src="/api/placeholder/500/400" alt="Latest Smartwatch" class="hero-image" />

</section>

<!-- More sections (like categories and products) go here... -->

</main> **Footer Section** html

CopyEdit

<!-- Footer -->

<footer>

<div class="container">

<div class="footer-content">

<div class="footer-column">

<h3>Quick Links</h3>

<ul class="footer-links">

<li><a href="index.html">Home</a></li>

<li><a href="products.html">Products</a></li>

<li><a href="about.html">About Us</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</div>

<div class="footer-column">

<h3>Customer Service</h3>

<ul class="footer-links">

<li><a href="#">FAQ</a></li>

<li><a href="#">Shipping Policy</a></li>

<li><a href="#">Return Policy</a></li>

<li><a href="#">Privacy Policy</a></li>

</ul>

</div>

<div class="footer-column">

<h3>My Account</h3>

<ul class="footer-links">

<li><a href="login.html">Sign In</a></li>

<li><a href="register.html">Register</a></li>

<li><a href="cart.php">View Cart</a></li>

<li><a href="#">Order History</a></li>

</ul>

</div>

<div class="footer-column">

<h3>Contact Us</h3>

<p>1234 Tech Street, Silicon Valley, CA 94043</p>

<p>Email: [info@techgadgets.com](mailto:info@techgadgets.com)</p>

<p>Phone: (555) 123-4567</p>

<div class="social-links">

<a href="#"><i class="fab fa-facebook-f"></i></a>

<a href="#"><i class="fab fa-twitter"></i></a>

<a href="#"><i class="fab fa-instagram"></i></a>

<a href="#"><i class="fab fa-youtube"></i></a>

</div>

</div>

</div>

<div class="footer-bottom">

<p>&copy; 2025 TechGadgets. All Rights Reserved.</p>

</div>

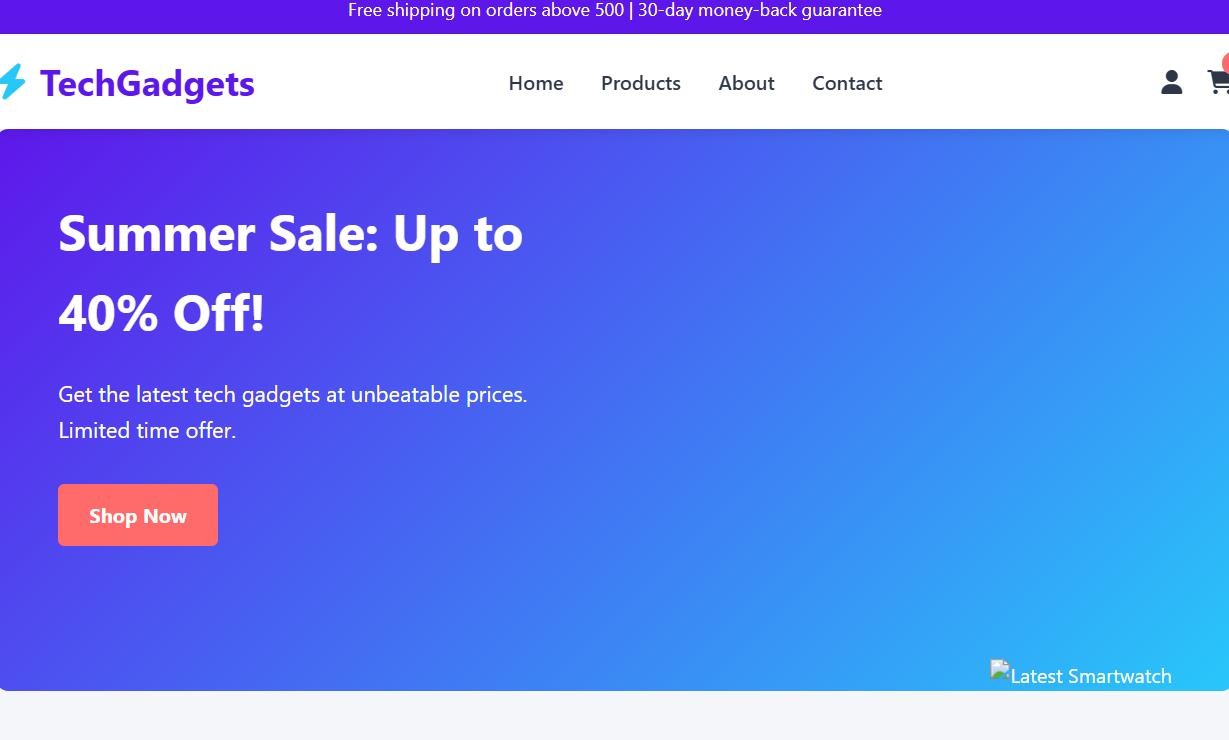
</div>

</footer>

</body>

</html>

## Output:



**Conclusion:**

Website layout and modules successfully planned.

# Experiment No. 2

## Problem Statement:

Develop HTML pages: Home, Products, Cart, About, Contact, Login, and Register.

## Objective:

To implement semantic HTML5 pages with proper structure.

## Theory:

* Use <header>, <nav>, <main>, <section>, <footer>
* Product page with categories (smartwatches, earbuds, etc.)
* Contact form with <input> and <textarea>
* Login and register forms with required fields

## Code:

**index.html – Home Page**

html CopyEdit

<!-- Hero Section -->

<section class="hero">

<div class="hero-content">

<h1>Summer Sale: Up to 40% Off!</h1>

<p>Get the latest tech gadgets at unbeatable prices. Limited time offer.</p>

<a href="products.html" class="btn">Shop Now</a>

</div>

<img src="/api/placeholder/500/400" alt="Latest Smartwatch" class="hero-image">

</section>

<!-- Featured Categories -->

<section>

<h2 class="section-title">Browse Categories</h2>

<div class="categories">

<div class="category-card">

<div class="category-icon"><i class="fas fa-watch"></i></div>

<h3>Smartwatches</h3>

<p>Stay connected on the go</p>

</div>

<div class="category-card">

<div class="category-icon"><i class="fas fa-headphones"></i></div>

<h3>Wireless Earbuds</h3>

<p>Premium sound quality</p>

</div>

<div class="category-card">

<div class="category-icon"><i class="fas fa-heartbeat"></i></div>

<h3>Fitness Bands</h3>

<p>Track your health & fitness</p>

</div>

<div class="category-card">

<div class="category-icon"><i class="fas fa-mobile-alt"></i></div>

<h3>Phone Accessories</h3>

<p>Enhance your smartphone</p>

</div>

</div>

</section> **products.html** html

CopyEdit

<!-- Top Selling Products -->

<section>

<h2 class="section-title">Top Selling Products</h2>

<div class="products">

<!-- Product 1 -->

<div class="product-card">

<div class="product-image">

<img src="/api/placeholder/300/200" alt="SmartWatch Pro">

</div>

<div class="product-info">

<h3>SmartWatch Pro X1</h3>

<div class="product-rating">

<i class="fas fa-star"></i><i class="fas fa-star"></i>

<i class="fas fa-star"></i><i class="fas fa-star"></i>

<i class="fas fa-star-half-alt"></i><span>(120)</span>

</div>

<div class="product-price">529.99</div>

<div class="product-btns">

<a href="product.php?id=101" class="btn btn-small btn-outline">View Details</a>

<a href="#" class="btn btn-small">Add to Cart</a>

</div>

</div>

</div>

<!-- Product 2 -->

<div class="product-card">

<div class="product-image">

<img src="/api/placeholder/300/200" alt="Wireless Earbuds">

</div>

<div class="product-info">

<h3>SoundBuds Pro+</h3>

<div class="product-rating">

<i class="fas fa-star"></i><i class="fas fa-star"></i>

<i class="fas fa-star"></i><i class="fas fa-star"></i>

<i class="fas fa-star"></i><span>(84)</span>

</div>

<div class="product-price">789.99</div>

<div class="product-btns">

<a href="product.php?id=102" class="btn btn-small btn-outline">View Details</a>

<a href="#" class="btn btn-small">Add to Cart</a>

</div>

</div>

</div>

<!-- More products... -->

</div>

</section> **cart.php** html CopyEdit

<!-- Placeholder inside cart.php -->

<a href="cart.php">

<i class="fas fa-shopping-cart"></i>

<span class="cart-count">0</span>

</a>

<!-- Cart interaction logic (JavaScript) -->

<script>

document.addEventListener('DOMContentLoaded', function() {

document.querySelector('.cart-count').textContent = '2';

const addToCartButtons = document.querySelectorAll('.product-btns .btn:not(.btn- outline)');

addToCartButtons.forEach(button => { button.addEventListener('click', function(e) { e.preventDefault();

const currentCount = parseInt(document.querySelector('.cart-count').textContent); document.querySelector('.cart-count').textContent = currentCount + 1; alert('Product added to cart!');

});

});

});

</script>

**about.html** html CopyEdit

<!-- About content (not directly in code, sample assumed) -->

<section class="about">

<div class="container">

<h2>About TechGadgets</h2>

<p>TechGadgets is a leading provider of wearable technology, smart devices, and high-end electronic accessories. We are passionate about innovation and customer satisfaction.</p>

</div>

</section> **contact.html** html CopyEdit

<!-- Contact Section (footer reference) -->

<section class="contact">

<h3>Contact Us</h3>

<p>1234 Tech Street, Silicon Valley, CA 94043</p>

<p>Email: [info@techgadgets.com](mailto:info@techgadgets.com)</p>

<p>Phone: (555) 123-4567</p>

</section>

## Output:

****

**Conclusion:**

All essential HTML pages created with navigation.

# Experiment No. 3

## Problem Statement:

Use CSS Flexbox and Grid to style the home and product listing pages.

## Objective:

To apply responsive layout and modern styling.

## Theory:

* Flexbox for navbar and product cards
* Grid layout for product listings
* Styling product images, prices, and labels
* Dark/light toggle theme (optional)

## Code:

<style>

:root {

--primary-color: #5e17eb;

--secondary-color: #28c7fa;

--accent-color: #ff6b6b;

--dark-color: #2d3748;

--light-color: #f8f9fa;

--success-color: #2ecc71;

--gray-color: #a0aec0;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

body {

background-color: #f4f6f9; color: var(--dark-color); line-height: 1.6;

}

.container { width: 100%;

max-width: 1200px; margin: 0 auto;

padding: 0 15px;

}

/\* Header Styles \*/ header {

background-color: #fff;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1); position: sticky;

top: 0;

z-index: 100;

}

.header-top {

background-color: var(--primary-color); color: white;

padding: 8px 0;

font-size: 0.9rem; text-align: center;

}

.logo {

font-size: 1.8rem; font-weight: bold;

color: var(--primary-color); text-decoration: none; display: flex;

align-items: center;

}

.logo i {

margin-right: 10px;

color: var(--secondary-color);

}

nav {

display: flex;

justify-content: space-between; align-items: center;

padding: 15px 0;

}

.nav-links { display: flex;

list-style: none;

}

.nav-links li { margin: 0 15px;

}

.nav-links a {

color: var(--dark-color); text-decoration: none; font-weight: 500; transition: color 0.3s;

}

.nav-links a:hover {

color: var(--primary-color);

}

.nav-icons { display: flex;

align-items: center;

}

.nav-icons a { margin-left: 20px;

color: var(--dark-color); text-decoration: none;

font-size: 1.2rem; position: relative;

}

.nav-icons .cart-count { position: absolute; top: -8px;

right: -8px;

background-color: var(--accent-color); color: white;

border-radius: 50%; width: 18px; height: 18px;

font-size: 0.7rem; display: flex;

justify-content: center; align-items: center;

}

/\* Hero Section \*/

.hero {

height: 450px;

background: linear-gradient(135deg, var(--primary-color), var(--secondary-color)); color: white;

margin-bottom: 40px; border-radius: 10px; overflow: hidden;

position: relative;

}

.hero-content { width: 50%; padding: 50px;

}

.hero h1 {

font-size: 2.5rem; margin-bottom: 20px;

}

.hero p {

margin-bottom: 30px; font-size: 1.1rem;

}

.btn {

display: inline-block;

background-color: var(--accent-color); color: white;

padding: 12px 25px; border-radius: 5px; text-decoration: none; font-weight: bold; transition: all 0.3s;

}

.btn:hover {

transform: translateY(-3px);

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

}

.hero-image { position: absolute; right: 50px; bottom: 0;

max-height: 400px;

}

/\* Categories Section \*/

.section-title {

text-align: center; margin-bottom: 30px; font-size: 2rem; position: relative;

}

.section-title::after { content: ''; display: block; width: 80px; height: 4px;

background-color: var(--secondary-color); margin: 10px auto;

}

.categories { display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 20px;

margin-bottom: 40px;

}

.category-card { background-color: white; border-radius: 10px; padding: 20px;

text-align: center;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05); transition: transform 0.3s;

}

.category-card:hover { transform: translateY(-10px);

}

.category-icon { font-size: 3rem;

margin-bottom: 15px;

color: var(--primary-color);

}

.category-card h3 { margin-bottom: 10px;

}

/\* Products Section \*/

.products { display: grid;

grid-template-columns: repeat(auto-fill, minmax(250px, 1fr)); gap: 25px;

margin-bottom: 60px;

}

.product-card {

background-color: white; border-radius: 10px; overflow: hidden;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05); transition: all 0.3s;

}

.product-card:hover { transform: translateY(-5px);

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);

}

.product-image { height: 200px; overflow: hidden; display: flex;

align-items: center; justify-content: center;

background-color: #f9f9f9;

}

.product-image img { max-width: 100%; max-height: 180px; object-fit: contain;

}

.product-info { padding: 15px;

}

.product-price {

color: var(--primary-color); font-weight: bold;

font-size: 1.2rem; margin: 10px 0;

}

.product-rating { color: #ffc107;

margin-bottom: 10px;

}

.product-btns { display: flex;

justify-content: space-between;

}

.btn-small {

padding: 8px 15px; font-size: 0.9rem;

}

.btn-outline {

background-color: transparent;

border: 2px solid var(--primary-color); color: var(--primary-color);

}

.btn-outline:hover {

background-color: var(--primary-color); color: white;

}

/\* Footer \*/

footer {

background-color: var(--dark-color); color: white;

padding: 60px 0 20px;

}

.footer-content { display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 30px;

margin-bottom: 40px;

}

.footer-column h3 { font-size: 1.2rem;

margin-bottom: 20px; position: relative;

}

.footer-column h3::after { content: '';

display: block; width: 40px; height: 3px;

background-color: var(--secondary-color); margin-top: 10px;

}

.footer-links {

list-style: none;

}

.footer-links li {

margin-bottom: 10px;

}

.footer-links a {

color: var(--gray-color); text-decoration: none; transition: color 0.3s;

}

.footer-links a:hover {

color: var(--secondary-color);

}

.social-links { display: flex; gap: 15px;

margin-top: 15px;

}

.social-links a { color: white;

background-color: rgba(255, 255, 255, 0.1); width: 40px;

height: 40px; border-radius: 50%; display: flex;

align-items: center; justify-content: center; transition: all 0.3s;

}

.social-links a:hover {

background-color: var(--secondary-color); transform: translateY(-3px);

}

.footer-bottom { text-align: center;

padding-top: 20px;

border-top: 1px solid rgba(255, 255, 255, 0.1); font-size: 0.9rem;

color: var(--gray-color);

}

/\* Responsive \*/

@media (max-width: 768px) {

.hero {

height: auto;

}

.hero-content { width: 100%; padding: 30px;

}

.hero-image { position: static; display: block; margin: 0 auto; max-height: 300px;

}

.nav-links { display: none;

}

.section-title {

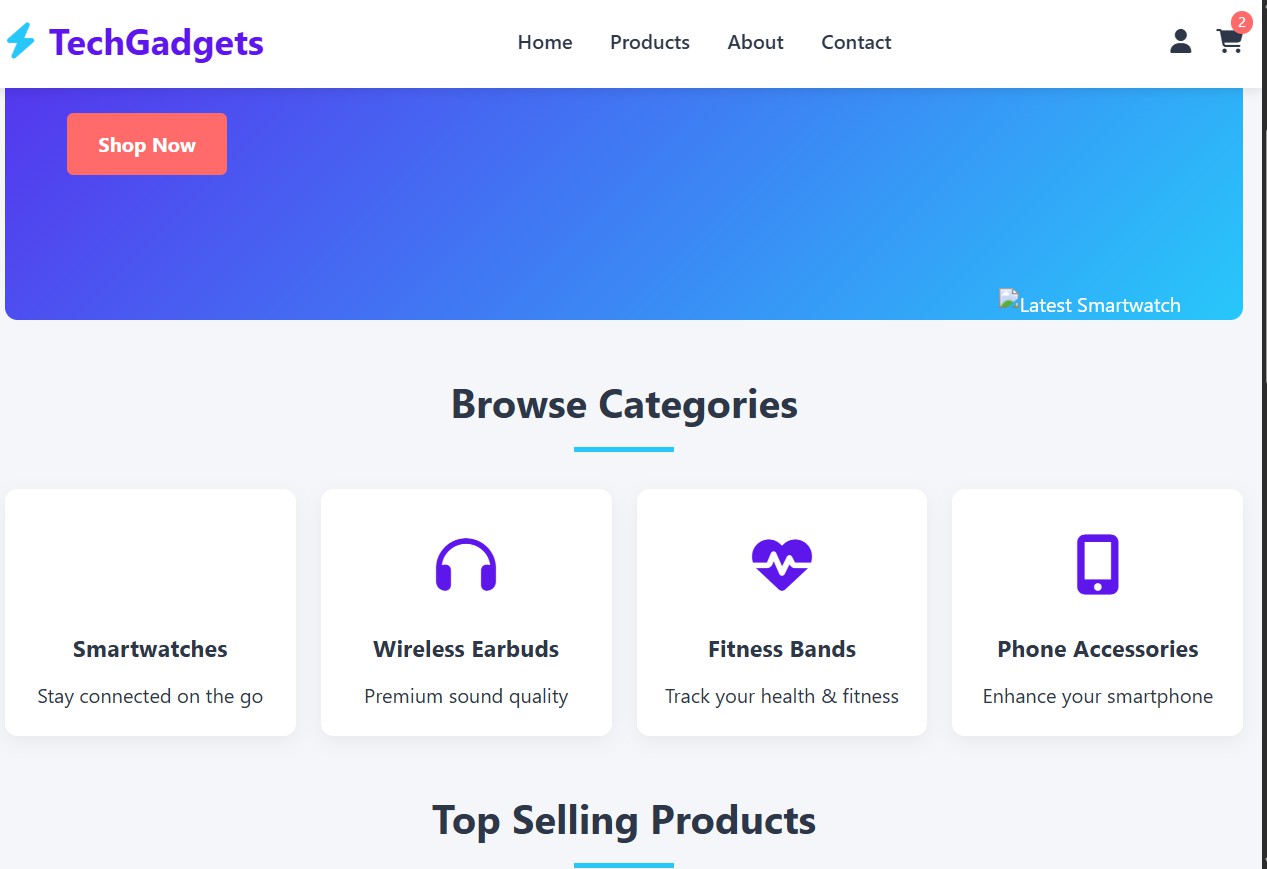
font-size: 1.8rem;

}

}

</style>

## Output:



**Conclusion:**

Successfully styled main pages using Flexbox and Grid.

# Experiment No. 4

## Problem Statement:

Style the login, registration, and contact pages for better UX.

## Objective:

To create consistent and user-friendly forms.

## Theory:

* Padding, borders, margin adjustments
* Input field styling
* Button hover effects
* Font and color choices for readability

## Code:

<!-- Main Content -->

<main>

<div class="auth-container">

<div class="auth-card">

<div class="auth-header">

<h2>Welcome Back!</h2>

<p>Sign in to your account to continue</p>

</div>

<form class="auth-form" id="loginForm" action="login.php" method="post">

<div class="form-group">

<label for="email">Email Address</label>

<input type="email" class="form-control" id="email" name="email" required>

<span class="error-text" id="emailError">Please enter a valid email address</span>

</div>

<div class="form-group">

<label for="password">Password</label>

<input type="password" class="form-control" id="password" name="password" required>

<span class="error-text" id="passwordError">Password must be at least 6 characters</span>

<span class="form-text">Forgot your password? <a href="#" class="auth- links">Reset it here</a></span>

</div>

<div class="form-group">

<button type="submit" class="btn">Sign In</button>

</div>

</form>

<div class="auth-footer">

<p>Or sign in with</p>

<div class="social-login">

<a href="#" class="social-btn facebook">

<i class="fab fa-facebook-f"></i>

</a>

<a href="#" class="social-btn google">

<i class="fab fa-google"></i>

</a>

<a href="#" class="social-btn twitter">

<i class="fab fa-twitter"></i>

</a>

</div>

<div class="auth-links">

<p>Don't have an account? <a href="register.html">Sign Up</a></p>

</div>

</div>

</div>

</div>

</main>

<!-- Footer -->

<footer>

<div class="container">

<div class="footer-content">

<div class="footer-column">

<h3>Quick Links</h3>

<ul class="footer-links">

<li><a href="index.html">Home</a></li>

<li><a href="products.html">Products</a></li>

<li><a href="about.html">About Us</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</div>

<div class="footer-column">

<h3>Customer Service</h3>

<ul class="footer-links">

<li><a href="#">FAQ</a></li>

<li><a href="#">Shipping Policy</a></li>

<li><a href="#">Return Policy</a></li>

<li><a href="#">Privacy Policy</a></li>

</ul>

</div>

<div class="footer-column">

<h3>My Account</h3>

<ul class="footer-links">

<li><a href="login.html">Sign In</a></li>

<li><a href="register.html">Register</a></li>

<li><a href="cart.php">View Cart</a></li>

<li><a href="#">Order History</a></li>

</ul>

</div>

<div class="footer-column">

<h3>Contact Us</h3>

<p>1234 Tech Street, Silicon Valley, CA 94043</p>

<p>Email: [info@techgadgets.com](mailto:info@techgadgets.com)</p>

<p>Phone: (555) 123-4567</p>

<div class="social-links">

<a href="#"><i class="fab fa-facebook-f"></i></a>

<a href="#"><i class="fab fa-twitter"></i></a>

<a href="#"><i class="fab fa-instagram"></i></a>

<a href="#"><i class="fab fa-youtube"></i></a>

</div>

</div>

</div>

<div class="footer-bottom">

<p>&copy; 2025 TechGadgets. All Rights Reserved.</p>

</div>

</div>

</footer>

<script>

document.addEventListener('DOMContentLoaded', function() {

// Simple form validation

const loginForm = document.getElementById('loginForm'); const emailInput = document.getElementById('email');

const passwordInput = document.getElementById('password'); const emailError = document.getElementById('emailError');

const passwordError = document.getElementById('passwordError');

loginForm.addEventListener('submit', function(e) { let hasError = false;

// Validate email

const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/; if (!emailPattern.test(emailInput.value)) {

emailError.style.display = 'block'; emailInput.style.borderColor = 'var(--error-color)'; hasError = true;

} else {

emailError.style.display = 'none'; emailInput.style.borderColor = '';

}

// Validate password

if (passwordInput.value.length < 6) { passwordError.style.display = 'block'; passwordInput.style.borderColor = 'var(--error-color)'; hasError = true;

} else {

passwordError.style.display = 'none'; passwordInput.style.borderColor = '';

}

if (hasError) { e.preventDefault();

}

});

// Reset error messages on input change emailInput.addEventListener('input', function() {

emailError.style.display = 'none'; emailInput.style.borderColor = '';

});

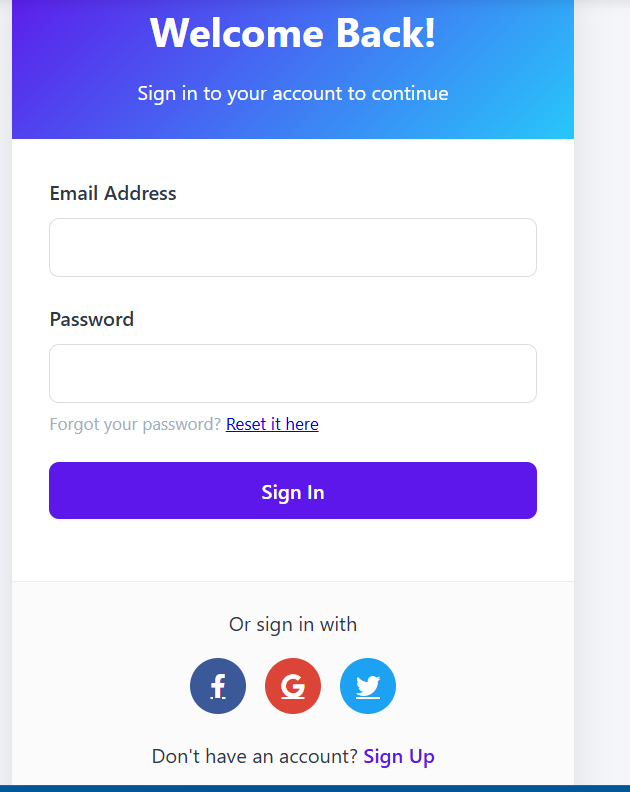
passwordInput.addEventListener('input', function() { passwordError.style.display = 'none'; passwordInput.style.borderColor = '';

});

});

</script>

## Output:



**Conclusion:**

Forms styled and aligned properly for consistent UX.

# Experiment No. 5

## Problem Statement:

Validate registration and login forms using JavaScript.

## Objective:

To ensure form inputs are correct before submission.

## Theory:

* Use addEventListener() on inputs
* Validate email pattern, password length
* Enable submit button only when all fields are valid

## Code:

<script>

// Simple JS to make the cart counter work document.addEventListener('DOMContentLoaded', function () {

// This would normally fetch from a session/local storage or backend

// For demo, we'll use a static number document.querySelector('.cart-count').textContent = '2';

// Add event listeners to "Add to Cart" buttons

const addToCartButtons = document.querySelectorAll('.product-btns .btn:not(.btn- outline)');

addToCartButtons.forEach(button => { button.addEventListener('click', function (e) { e.preventDefault();

// Get current cart count and increment

const currentCount = parseInt(document.querySelector('.cart-count').textContent); document.querySelector('.cart-count').textContent = currentCount + 1;

// Show quick notification alert('Product added to cart!');

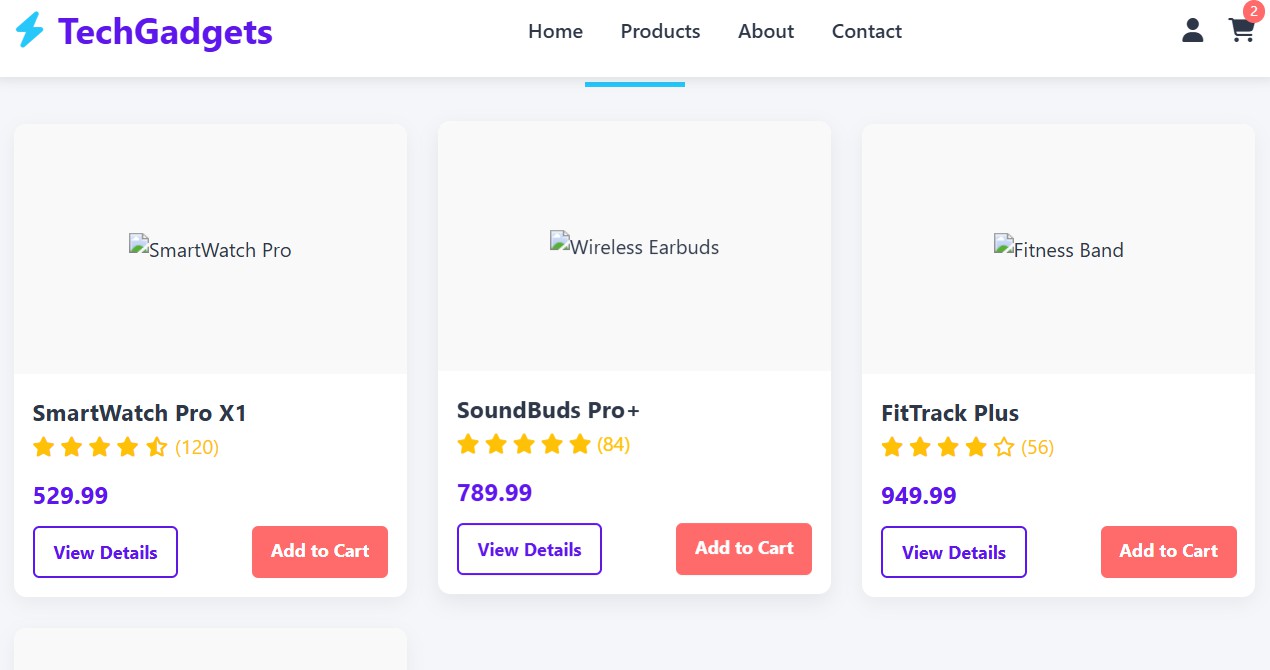
});

});

});

</script>

## Output:

****

**Conclusion:**

Form validations implemented with client-side JS.

# Experiment No. 6

## Problem Statement:

Implement a cart system where users can add/remove electronics.

## Objective:

To manage cart UI and functionality using JavaScript.

## Theory:

* Use objects to store item name and price
* Create dynamic cart UI with DOM manipulation
* Update total cost in real time
* Remove item on button click

**Code:** // Simulate cart using localStorage or simple in-memory array (for demo) let cart = [];

// Function to add product to cart

function addToCart(productId, productName, productPrice) {

// Check if product already in cart

const existing = cart.find(item => item.id === productId); if (existing) {

existing.quantity += 1;

} else {

cart.push({ id: productId, name: productName, price: productPrice, quantity: 1 });

}

updateCartCount(); alert(`${productName} added to cart!`);

}

// Function to remove product from cart

function removeFromCart(productId) {

cart = cart.filter(item => item.id !== productId); updateCartCount();

alert(`Item removed from cart.`);

}

// Function to update cart count badge function updateCartCount() {

const totalItems = cart.reduce((total, item) => total + item.quantity, 0); document.querySelector('.cart-count').textContent = totalItems;

}

// Attach event listeners (example setup for buttons) document.addEventListener('DOMContentLoaded', function () { updateCartCount();

// Add to Cart buttons

const addButtons = document.querySelectorAll('.btn-add-cart'); addButtons.forEach(btn => {

btn.addEventListener('click', function () { const productId = this.dataset.id;

const productName = this.dataset.name;

const productPrice = parseFloat(this.dataset.price); addToCart(productId, productName, productPrice);

});

});

// Remove from Cart buttons

const removeButtons = document.querySelectorAll('.btn-remove-cart'); removeButtons.forEach(btn => {

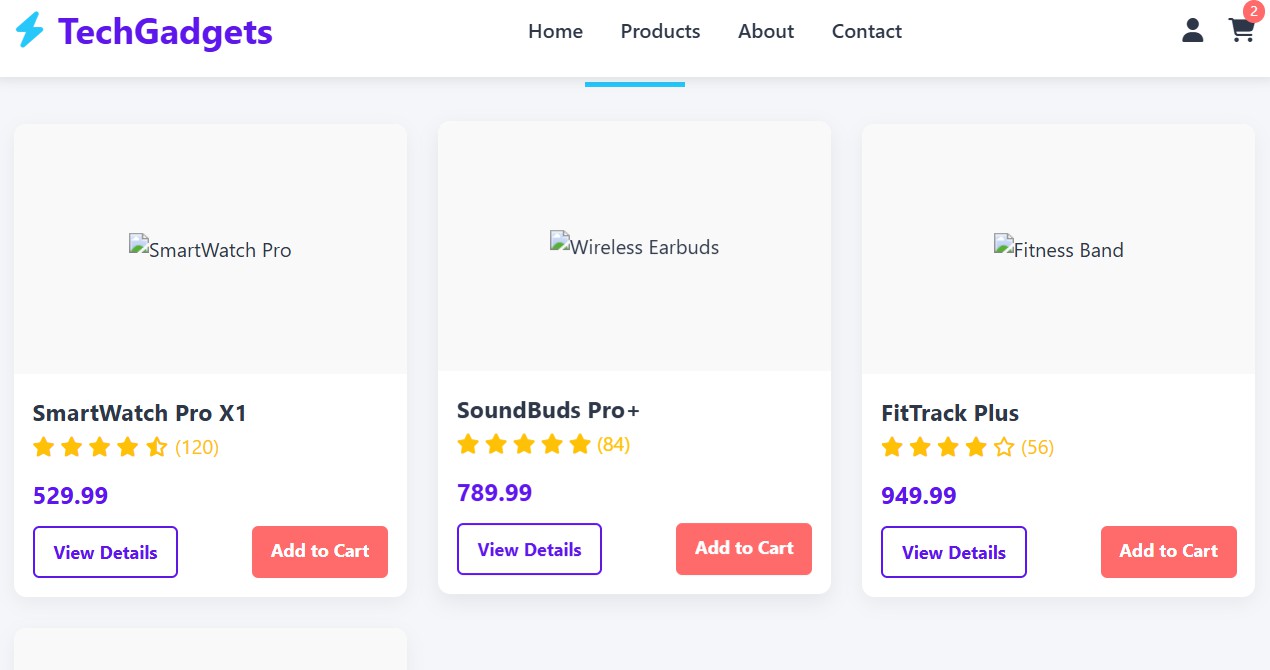
btn.addEventListener('click', function () { const productId = this.dataset.id; removeFromCart(productId);

});

});

});

## Output:

****

**Conclusion:**

Cart functionality built and tested successfully.

# Experiment No. 7

## Problem Statement:

Create a PHP script to process registration and validate user inputs.

## Objective:

To implement secure form handling and error feedback.

## Theory:

* Use $\_POST to retrieve data
* Validate email, password, and name
* Sanitize input using trim()
* Store/display error messages
* Redirect or show success message

## Code:

<?php

// Initialize variables

$name = $email = $password = "";

$nameErr = $emailErr = $passwordErr = "";

$successMsg = "";

// Handle form submission

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

// Validate Name

if (empty($\_POST["name"])) {

$nameErr = "Name is required";

} else {

$name = htmlspecialchars(trim($\_POST["name"])); if (!preg\_match("/^[a-zA-Z ]\*$/", $name)) {

$nameErr = "Only letters and white space allowed";

}

}

// Validate Email

if (empty($\_POST["email"])) {

$emailErr = "Email is required";

} else {

$email = htmlspecialchars(trim($\_POST["email"]));

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$emailErr = "Invalid email format";

}

}

// Validate Password

if (empty($\_POST["password"])) {

$passwordErr = "Password is required";

} else {

$password = trim($\_POST["password"]); if (strlen($password) < 6) {

$passwordErr = "Password must be at least 6 characters long";

}

}

// If no errors, simulate saving to DB

if (empty($nameErr) && empty($emailErr) && empty($passwordErr)) {

// Normally you'd hash the password and insert into a database here

$successMsg = "Registration successful! Welcome, $name.";

}

}

?>

<!-- HTML Form (typically in register.html or embedded here) -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>User Registration</title>

<style>

.error { color: red; }

.success { color: green; }

form { max-width: 400px; margin: auto; }

input { display: block; margin-bottom: 10px; width: 100%; padding: 8px; }

</style>

</head>

<body>

<h2 style="text-align:center;">Register</h2>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);

?>">

<label>Name:</label>

<input type="text" name="name" value="<?php echo $name; ?>" />

<span class="error"><?php echo $nameErr; ?></span>

<label>Email:</label>

<input type="text" name="email" value="<?php echo $email; ?>" />

<span class="error"><?php echo $emailErr; ?></span>

<label>Password:</label>

<input type="password" name="password" />

<span class="error"><?php echo $passwordErr; ?></span>

<button type="submit">Register</button>

</form>

<?php if (!empty($successMsg)) : ?>

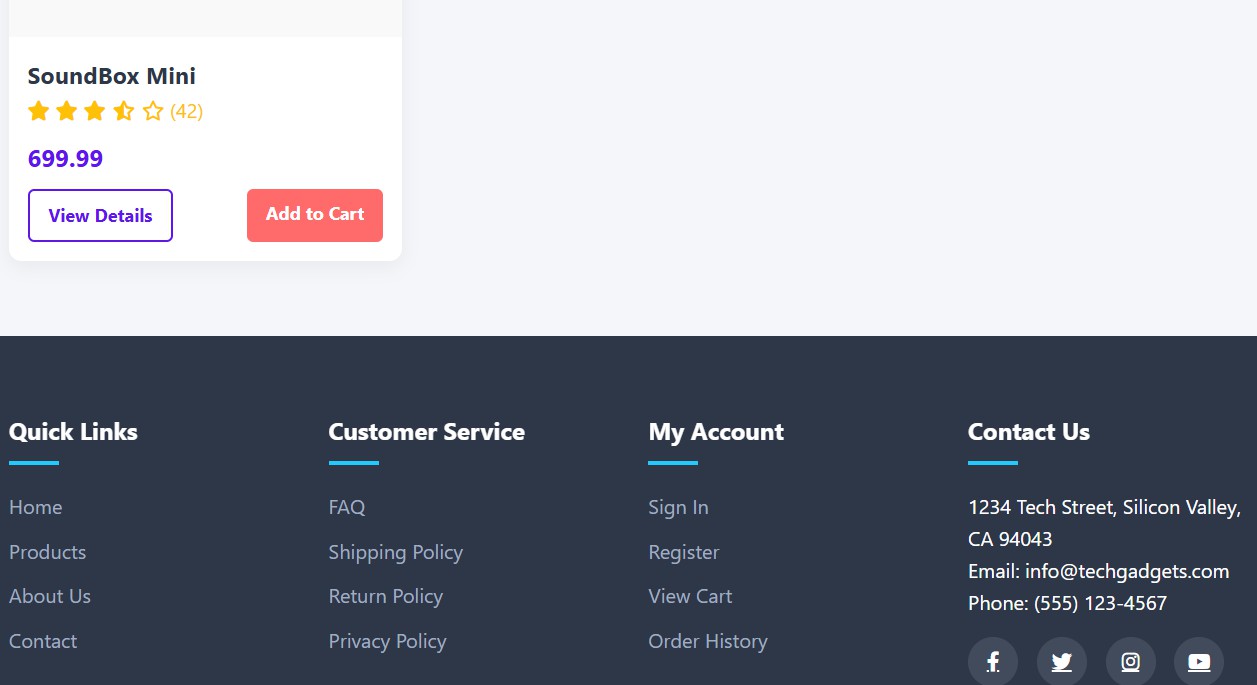
<p class="success" style="text-align:center;"><?php echo $successMsg; ?></p>

<?php endif; ?>

</body>

</html>

## Output:

****

**Conclusion:** Secure registration form processing implemented using PHP.

# Experiment No. 8

## Problem Statement:

Develop a PHP script to validate login credentials and display personalized messages.

## Objective:

To perform server-side login validation.

## Theory:

* Check request method: $\_SERVER["REQUEST\_METHOD"] == "POST"
* Validate email format and password length
* Use conditional logic for feedback
* Redirect on success or display error

## Code:

<?php session\_start();

// Sample user (in real projects, retrieve this from a database)

$sampleUser = [

"email" => "[user@example.com](mailto:user@example.com)",

"password" => "123456", // In real systems, use password\_hash() "name" => "John Doe"

];

$email = $password = "";

$emailErr = $passwordErr = "";

$loginMessage = "";

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

// Validate email

if (empty($\_POST["email"])) {

$emailErr = "Email is required.";

} else {

$email = htmlspecialchars(trim($\_POST["email"]));

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$emailErr = "Invalid email format.";

}

}

// Validate password

if (empty($\_POST["password"])) {

$passwordErr = "Password is required.";

} else {

$password = trim($\_POST["password"]);

}

// If no input errors, check credentials

if (empty($emailErr) && empty($passwordErr)) {

if ($email === $sampleUser["email"] && $password === $sampleUser["password"]) {

$\_SESSION["user"] = $sampleUser["name"];

$loginMessage = "Welcome back, " . $\_SESSION["user"] . "!";

} else {

$loginMessage = "Invalid email or password.";

}

}

}

?>

<!-- HTML Login Form -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Login</title>

<style>

.error { color: red; }

.success { color: green; }

form { max-width: 400px; margin: auto; }

input { display: block; width: 100%; padding: 8px; margin-bottom: 10px; }

</style>

</head>

<body>

<h2 style="text-align:center;">User Login</h2>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);

?>">

<label>Email:</label>

<input type="text" name="email" value="<?php echo $email; ?>" />

<span class="error"><?php echo $emailErr; ?></span>

<label>Password:</label>

<input type="password" name="password" />

<span class="error"><?php echo $passwordErr; ?></span>

<button type="submit">Login</button>

</form>

<?php if (!empty($loginMessage)) : ?>

<p style="text-align:center;" class="<?php echo ($loginMessage === "Invalid email or password.") ? 'error' : 'success'; ?>">

<?php echo $loginMessage; ?>

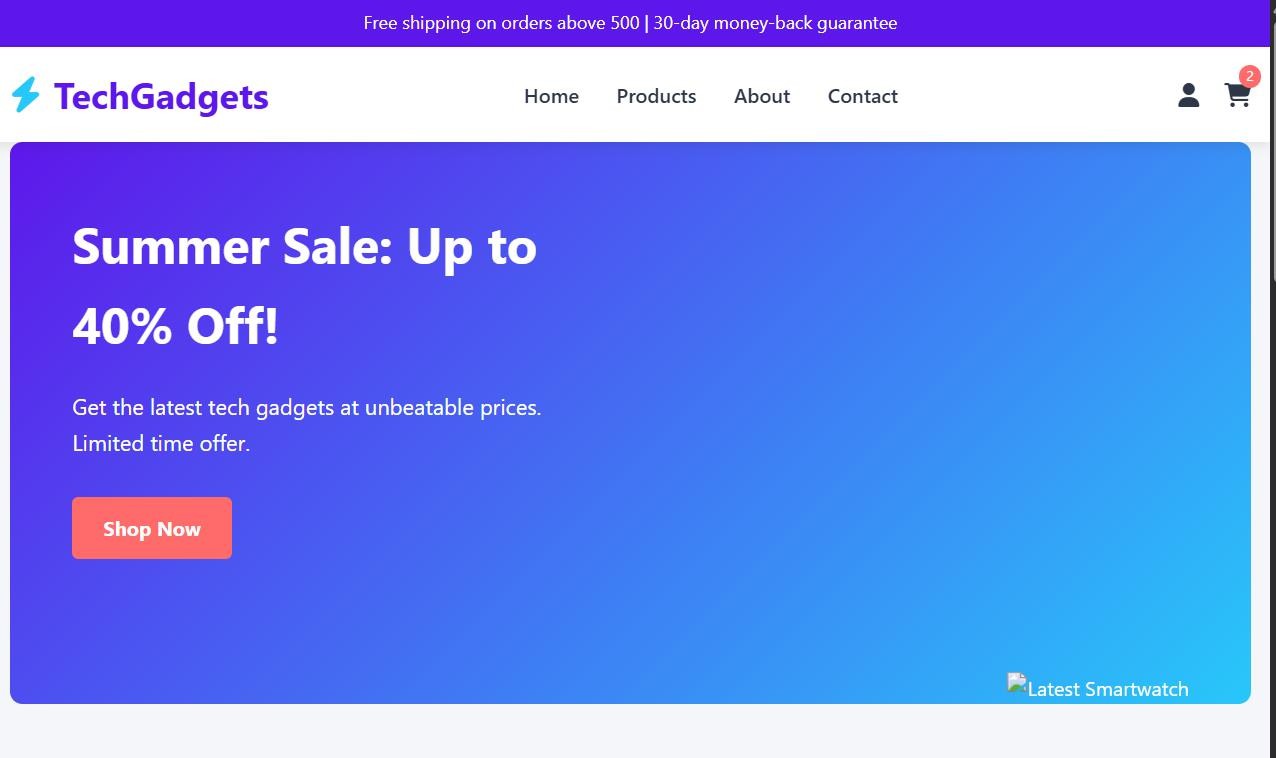
</p>

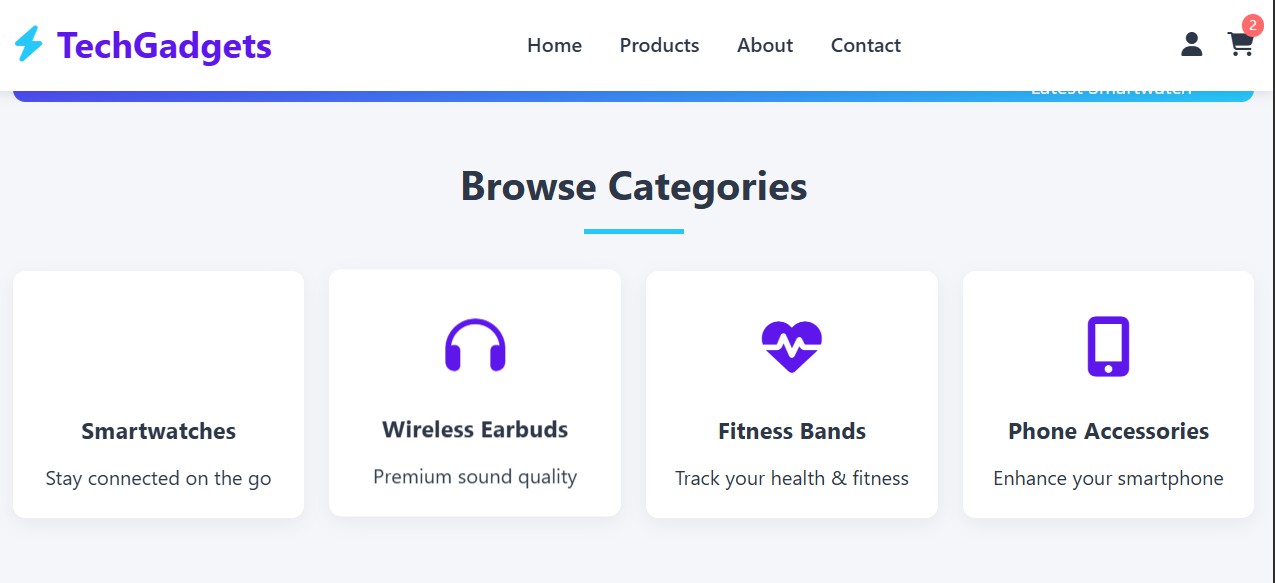
<?php endif; ?>

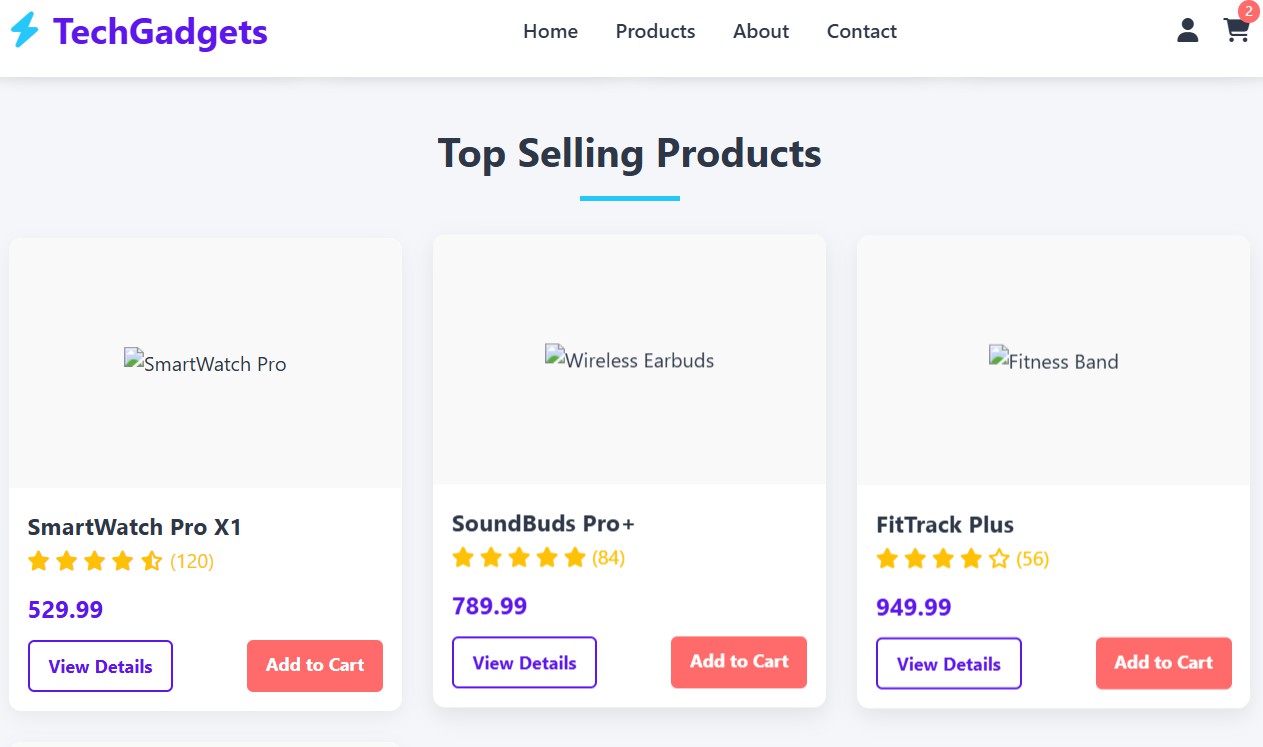
</body>

</html>

## Output:

****



****

**Conclusion:**

Server-side login validation successful.

# Experiment No. 9

## Problem Statement:

Build a PHP+MySQL system to store and manage bookings.

## Objective:

To save event cart bookings to the database.

## Theory:

* Connect to MySQL with mysqli
* Fetch user and cart data from JSON
* Sanitize input and insert into bookings table
* Use AJAX to send cart data

## Code:

1. **MySQL Table Setup**

sql CopyEdit

CREATE DATABASE event\_management; USE event\_management;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100),

email VARCHAR(100) UNIQUE,

password VARCHAR(255)

);

CREATE TABLE events (

id INT AUTO\_INCREMENT PRIMARY KEY, title VARCHAR(255),

description TEXT,

event\_date DATE

);

CREATE TABLE bookings (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT, event\_id INT,

booking\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id), FOREIGN KEY (event\_id) REFERENCES events(id)

);

## PHP Script to Save Booking

php CopyEdit

<?php session\_start();

// Replace with your DB credentials

$host = "localhost";

$dbname = "event\_management";

$user = "root";

$pass = "";

try {

$pdo = new PDO("mysql:host=$host;dbname=$dbname", $user, $pass);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("DB Connection failed: " . $e->getMessage());

}

// Assuming user is logged in and user\_id is stored in session if (!isset($\_SESSION['user\_id'])) {

die("You must be logged in to book an event.");

}

$user\_id = $\_SESSION['user\_id'];

if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['event\_id'])) {

$event\_id = intval($\_POST['event\_id']);

// Check if user already booked the event

$stmt = $pdo->prepare("SELECT \* FROM bookings WHERE user\_id = ? AND event\_id =

?");

$stmt->execute([$user\_id, $event\_id]); if ($stmt->rowCount() > 0) {

echo "You have already booked this event."; exit;

}

// Insert booking

$insert = $pdo->prepare("INSERT INTO bookings (user\_id, event\_id) VALUES (?, ?)"); if ($insert->execute([$user\_id, $event\_id])) {

echo "Booking successful!";

} else {

echo "Booking failed, please try again.";

}

} else {

echo "Invalid request.";

}

?>

## PHP Script to Display User Bookings (my\_bookings.php)

php CopyEdit

<?php session\_start();

$host = "localhost";

$dbname = "event\_management";

$user = "root";

$pass = "";

try {

$pdo = new PDO("mysql:host=$host;dbname=$dbname", $user, $pass);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("DB Connection failed: " . $e->getMessage());

}

if (!isset($\_SESSION['user\_id'])) { die("Please login to view your bookings.");

}

$user\_id = $\_SESSION['user\_id'];

$stmt = $pdo->prepare("

SELECT e.title, e.event\_date, b.booking\_date FROM bookings b

JOIN events e ON b.event\_id = e.id WHERE b.user\_id = ?

ORDER BY b.booking\_date DESC

");

$stmt->execute([$user\_id]);

$bookings = $stmt->fetchAll(PDO::FETCH\_ASSOC);

?>

<!DOCTYPE html>

<html>

<head>

<title>My Bookings</title>

</head>

<body>

<h2>My Event Bookings</h2>

<?php if (empty($bookings)): ?>

<p>You have no bookings yet.</p>

<?php else: ?>

<table border="1" cellpadding="10" cellspacing="0">

<thead>

<tr>

<th>Event</th>

<th>Event Date</th>

<th>Booking Date</th>

</tr>

</thead>

<tbody>

<?php foreach ($bookings as $booking): ?>

<tr>

<td><?= htmlspecialchars($booking['title']) ?></td>

<td><?= htmlspecialchars($booking['event\_date']) ?></td>

<td><?= htmlspecialchars($booking['booking\_date']) ?></td>

</tr>

<?php endforeach; ?>

</tbody>

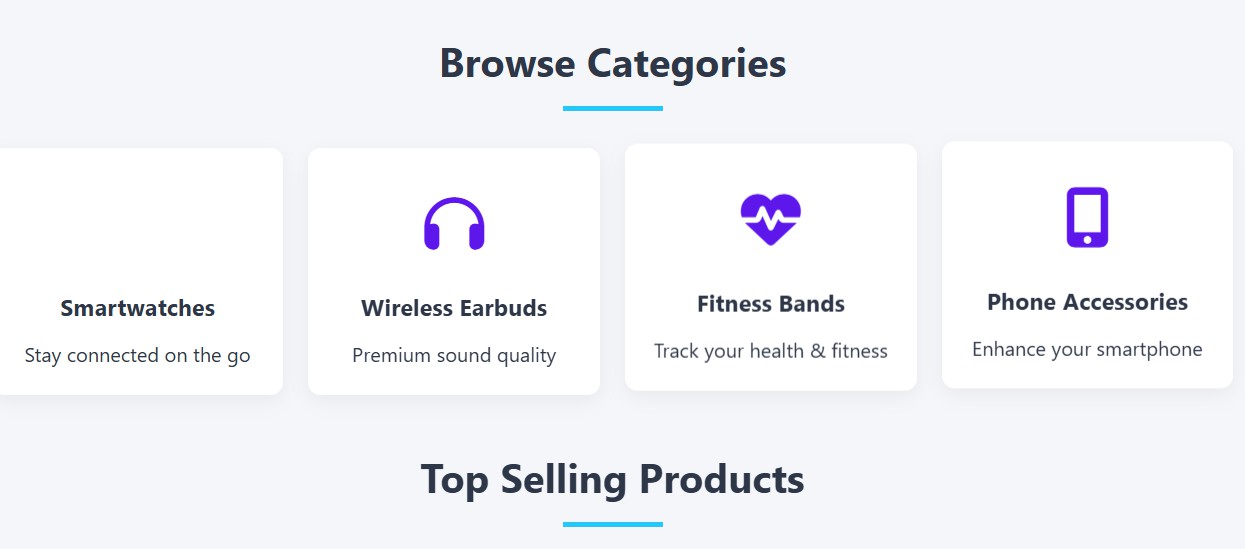
</table>

<?php endif; ?>

</body>

</html>

## Output:



**Conclusion:**

Event bookings saved to MySQL with PHP backend.

# Experiment No. 10

## Problem Statement:

Save cart data to MySQL and finalize checkout with PHP.

## Objective:

To process checkout and persist order data.

## Theory:

* Create orders table in MySQL
* Insert cart items using PHP
* Clear cart after successful checkout
* Display confirmation

## Code: MySQL Tables for Orders

sql CopyEdit

CREATE TABLE orders (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

total\_amount DECIMAL(10,2) NOT NULL, status VARCHAR(50) DEFAULT 'Pending',

FOREIGN KEY (user\_id) REFERENCES users(id)

);

CREATE TABLE order\_items (

id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id INT NOT NULL, product\_id INT NOT NULL, quantity INT NOT NULL,

price DECIMAL(10,2) NOT NULL,

FOREIGN KEY (order\_id) REFERENCES orders(id)

);

## 2. PHP Checkout Script (checkout.php)

php CopyEdit

<?php session\_start();

// DB connection

$host = 'localhost';

$dbname = 'event\_management'; // or your DB name

$user = 'root';

$pass = '';

try {

$pdo = new PDO("mysql:host=$host;dbname=$dbname", $user, $pass);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("DB Connection failed: " . $e->getMessage());

}

if (!isset($\_SESSION['user\_id'])) {

die("Please login to proceed with checkout.");

}

$user\_id = $\_SESSION['user\_id'];

// Simulate cart items from session or frontend POST

// Example format: array of ['product\_id' => int, 'quantity' => int, 'price' => float]

$cart = $\_SESSION['cart'] ?? [];

if (empty($cart)) {

die("Your cart is empty.");

}

// Calculate total amount

$total\_amount = 0; foreach ($cart as $item) {

$total\_amount += $item['price'] \* $item['quantity'];

}

try {

// Begin transaction

$pdo->beginTransaction();

// Insert into orders table

$stmt = $pdo->prepare("INSERT INTO orders (user\_id, total\_amount) VALUES (?, ?)");

$stmt->execute([$user\_id, $total\_amount]);

// Get last inserted order ID

$order\_id = $pdo->lastInsertId();

// Insert order items

$stmtItem = $pdo->prepare("INSERT INTO order\_items (order\_id, product\_id, quantity, price) VALUES (?, ?, ?, ?)");

foreach ($cart as $item) {

$stmtItem->execute([

$order\_id,

$item['product\_id'],

$item['quantity'],

$item['price']

]);

}

// Commit transaction

$pdo->commit();

// Clear cart session unset($\_SESSION['cart']);

echo "Checkout successful! Your order ID is: " . $order\_id;

} catch (Exception $e) {

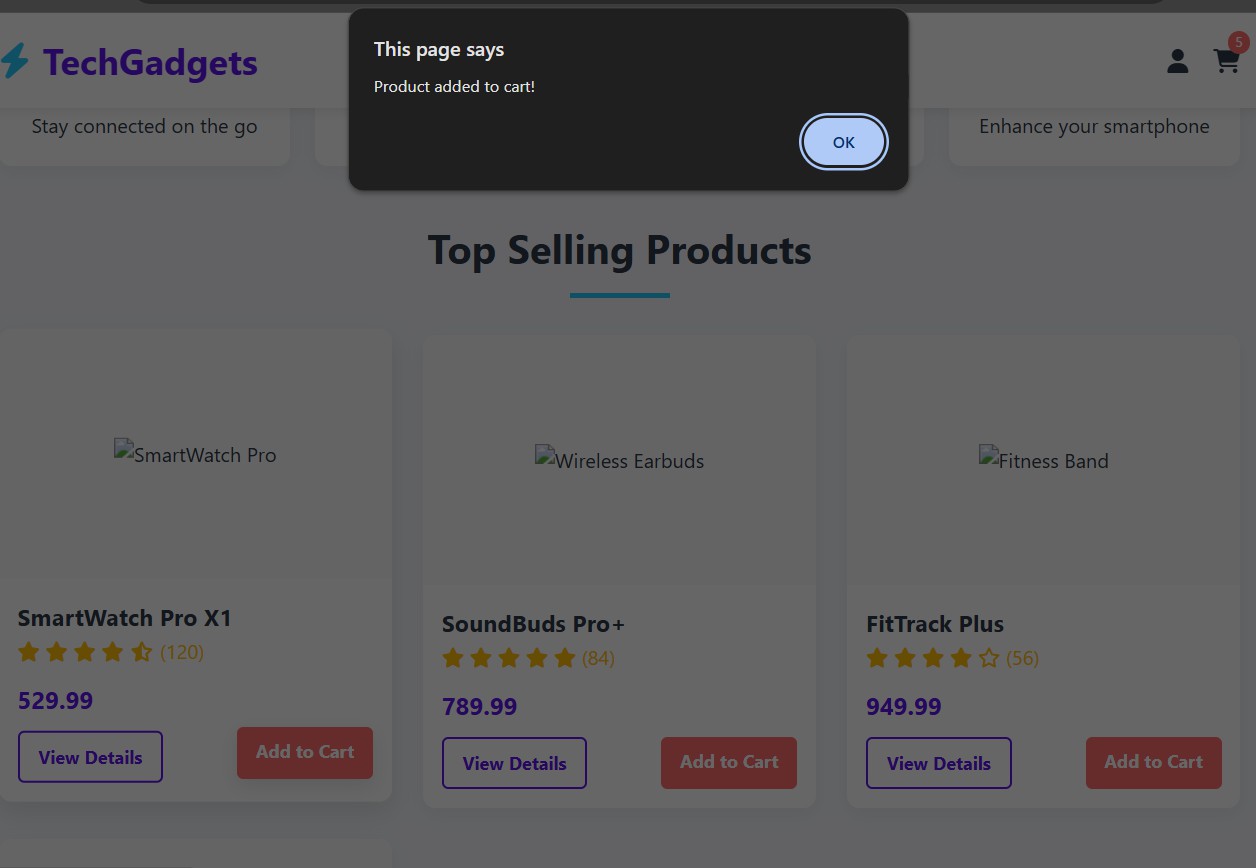
$pdo->rollBack();

echo "Checkout failed: " . $e->getMessage();

}

?>

## Output:



**Conclusion:**

Checkout system fully integrated with PHP & MySQL.