**Exercise 1: Sum of Two Numbers**

**Task**: Write a JavaScript function sumTwoNumbers(a, b) that takes two numbers as arguments and returns their sum. Call the function with different sets of numbers and log the result.

**Example**:

function sumTwoNumbers(a, b) {

return a + b;

}

console.log(sumTwoNumbers(5, 10)); // 15

console.log(sumTwoNumbers(3, 7)); // 10

**Exercise 2: Factorial Function**

**Task**: Write a function factorial(n) that takes a number n as an argument and returns its factorial. Use recursion to solve this.

**Example**:

function factorial(n) {

if (n === 0 || n === 1) {

return 1;

}

return n \* factorial(n - 1);

}

console.log(factorial(5)); // 120

console.log(factorial(0)); // 1

**Exercise 3: Check if a String is a Palindrome**

**Task**: Write a function isPalindrome(str) that checks if a given string is a palindrome (reads the same forward and backward).

**Example**:

function isPalindrome(str) {

let cleaned = str.replace(/\W/g, '').toLowerCase();

let reversed = cleaned.split('').reverse().join('');

return cleaned === reversed;

}

console.log(isPalindrome('madam')); // true

console.log(isPalindrome('hello')); // false

**Exercise 4: Find the Maximum Number in an Array**

**Task**: Write a function findMax(arr) that takes an array of numbers and returns the maximum value.

**Example**:

function findMax(arr) {

let max = arr[0];

for (let i = 1; i < arr.length; i++) {

if (arr[i] > max) {

max = arr[i];

}

}

return max;

}

console.log(findMax([2, 5, 8, 3, 9])); // 9

console.log(findMax([10, 20, 30, 5])); // 30

**Exercise 5: Convert Temperature from Celsius to Fahrenheit**

**Task**: Write a function convertToFahrenheit(celsius) that takes a temperature in Celsius and converts it to Fahrenheit using the formula: F = C \* 9/5 + 32.

**Example**:

function convertToFahrenheit(celsius) {

return (celsius \* 9/5) + 32;

}

console.log(convertToFahrenheit(0)); // 32

console.log(convertToFahrenheit(25)); // 77

**Exercise 6: Reverse a String**

**Task**: Write a function reverseString(str) that takes a string and returns it reversed.

**Example**:

function reverseString(str) {

return str.split('').reverse().join('');

}

console.log(reverseString('hello')); // 'olleh'

console.log(reverseString('world')); // 'dlrow'

**Exercise 7: Sum of Array Elements**

**Task**: Write a function sumArray(arr) that takes an array of numbers and returns the sum of all its elements.

**Example**:

function sumArray(arr) {

let sum = 0;

for (let i = 0; i < arr.length; i++) {

sum += arr[i];

}

return sum;

}

console.log(sumArray([1, 2, 3, 4, 5])); // 15

console.log(sumArray([10, 20, 30])); // 60

**Exercise 8: Check Prime Number**

**Task**: Write a function isPrime(num) that takes a number and returns true if the number is prime, and false otherwise.

**Example**:

function isPrime(num) {

if (num <= 1) return false;

for (let i = 2; i < num; i++) {

if (num % i === 0) {

return false;

}

}

return true;

}

console.log(isPrime(7)); // true

console.log(isPrime(4)); // false

**Exercise 9: Count Vowels in a String**

**Task**: Write a function countVowels(str) that takes a string and returns the number of vowels in the string.

**Example**:

function countVowels(str) {

const vowels = 'aeiouAEIOU';

let count = 0;

for (let i = 0; i < str.length; i++) {

if (vowels.includes(str[i])) {

count++;

}

}

return count;

}

console.log(countVowels('hello')); // 2

console.log(countVowels('world')); // 1

**Exercise 10: Remove Duplicates from an Array**

**Task**: Write a function removeDuplicates(arr) that removes duplicate values from an array and returns a new array.

**Example**:

function removeDuplicates(arr) {

return [...new Set(arr)];

}

console.log(removeDuplicates([1, 2, 3, 4, 3, 2, 1])); // [1, 2, 3, 4]

console.log(removeDuplicates([10, 20, 30, 20])); // [10, 20, 30]

**Exercise 11: Find the Longest Word in a Sentence**

**Task**: Write a function longestWord(sentence) that takes a string (sentence) and returns the longest word in it.

**Example**:

function longestWord(sentence) {

let words = sentence.split(' ');

let longest = words[0];

for (let i = 1; i < words.length; i++) {

if (words[i].length > longest.length) {

longest = words[i];

}

}

return longest;

}

console.log(longestWord('The quick brown fox jumps over the lazy dog')); // 'jumps'

**Exercise 12: Calculate the Area of a Circle**

**Task**: Write a function areaOfCircle(radius) that takes the radius of a circle and returns its area using the formula: Area = π \* r^2.

**Example**:

function areaOfCircle(radius) {

return Math.PI \* Math.pow(radius, 2);

}

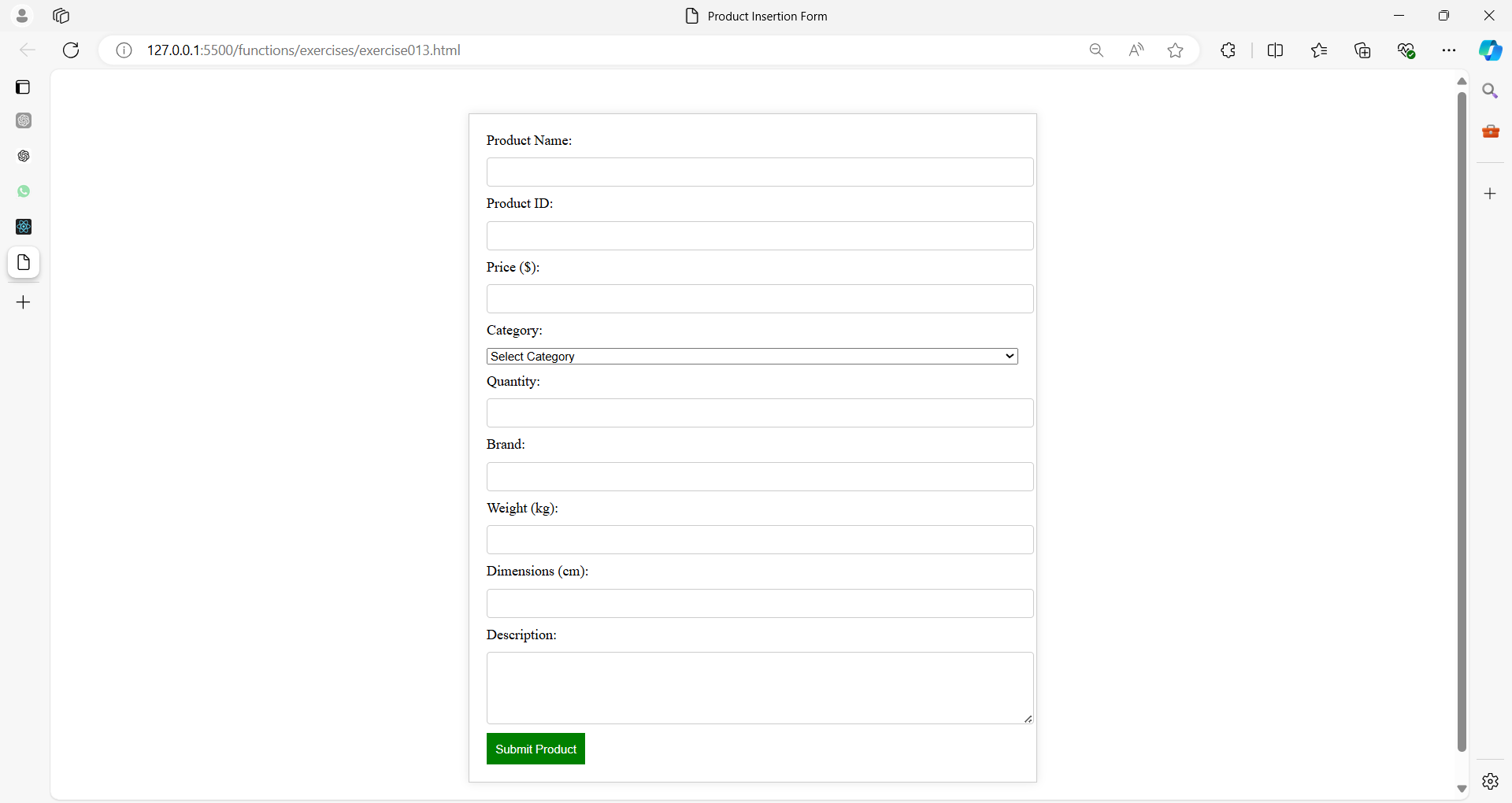
console.log(areaOfCircle(5)); // 78.5398...

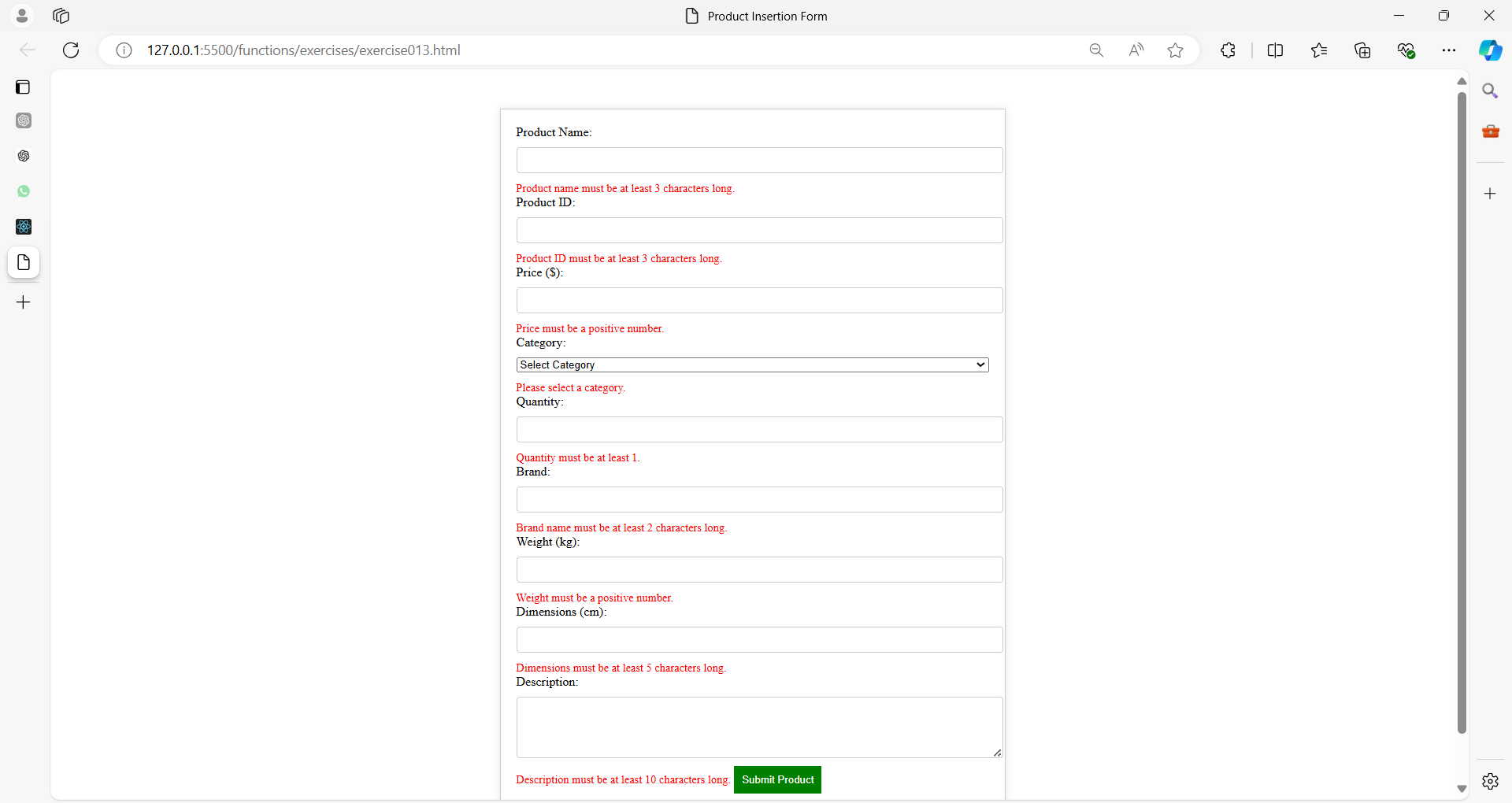
**Exercise 13: Product Insertion Form**

**Objective**: Create a product insertion form with various fields such as:

1. Product Name
2. Product ID
3. Price
4. Category
5. Quantity
6. Brand
7. Weight
8. Dimensions
9. Description

Use JavaScript to validate each field and display error messages in red when necessary.





**HTML**:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Product Insertion Form</title>

<style>

form {

width: 600px;

margin: 50px auto;

padding: 20px;

border: 1px solid #ccc;

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

}

label, input, textarea, select {

display: block;

margin-bottom: 10px;

width: 100%;

}

input[type="text"], input[type="number"], textarea {

padding: 8px;

border: 1px solid #ccc;

border-radius: 4px;

}

.error {

color: red;

font-size: 0.9em;

}

button {

padding: 10px;

background-color: green;

color: white;

border: none;

cursor: pointer;

}

button:hover {

background-color: darkgreen;

}

</style>

</head>

<body>

<form id="productForm">

<label for="productName">Product Name:</label>

<input type="text" id="productName" name="productName">

<span id="productNameError" class="error"></span>

<label for="productID">Product ID:</label>

<input type="text" id="productID" name="productID">

<span id="productIDError" class="error"></span>

<label for="price">Price ($):</label>

<input type="number" id="price" name="price">

<span id="priceError" class="error"></span>

<label for="category">Category:</label>

<select id="category" name="category">

<option value="">Select Category</option>

<option value="electronics">Electronics</option>

<option value="apparel">Apparel</option>

<option value="furniture">Furniture</option>

<option value="books">Books</option>

</select>

<span id="categoryError" class="error"></span>

<label for="quantity">Quantity:</label>

<input type="number" id="quantity" name="quantity">

<span id="quantityError" class="error"></span>

<label for="brand">Brand:</label>

<input type="text" id="brand" name="brand">

<span id="brandError" class="error"></span>

<label for="weight">Weight (kg):</label>

<input type="number" id="weight" name="weight">

<span id="weightError" class="error"></span>

<label for="dimensions">Dimensions (cm):</label>

<input type="text" id="dimensions" name="dimensions">

<span id="dimensionsError" class="error"></span>

<label for="description">Description:</label>

<textarea id="description" name="description" rows="4"></textarea>

<span id="descriptionError" class="error"></span>

<button type="submit">Submit Product</button>

</form>

<script>

document.getElementById('productForm').addEventListener('submit', function(event) {

event.preventDefault();

validateProductForm();

});

function validateProductForm() {

let valid = true;

// Clear previous error messages

document.querySelectorAll('.error').forEach(error => error.innerText = '');

// Get values from form fields

const productName = document.getElementById('productName').value;

const productID = document.getElementById('productID').value;

const price = document.getElementById('price').value;

const category = document.getElementById('category').value;

const quantity = document.getElementById('quantity').value;

const brand = document.getElementById('brand').value;

const weight = document.getElementById('weight').value;

const dimensions = document.getElementById('dimensions').value;

const description = document.getElementById('description').value;

// Product Name Validation

if (productName.length < 3) {

document.getElementById('productNameError').innerText = 'Product name must be at least 3 characters long.';

valid = false;

}

// Product ID Validation

if (productID.length < 3) {

document.getElementById('productIDError').innerText = 'Product ID must be at least 3 characters long.';

valid = false;

}

// Price Validation

if (price <= 0) {

document.getElementById('priceError').innerText = 'Price must be a positive number.';

valid = false;

}

// Category Validation

if (category === "") {

document.getElementById('categoryError').innerText = 'Please select a category.';

valid = false;

}

// Quantity Validation

if (quantity <= 0) {

document.getElementById('quantityError').innerText = 'Quantity must be at least 1.';

valid = false;

}

// Brand Validation

if (brand.length < 2) {

document.getElementById('brandError').innerText = 'Brand name must be at least 2 characters long.';

valid = false;

}

// Weight Validation

if (weight <= 0) {

document.getElementById('weightError').innerText = 'Weight must be a positive number.';

valid = false;

}

// Dimensions Validation

if (dimensions.length < 5) {

document.getElementById('dimensionsError').innerText = 'Dimensions must be at least 5 characters long.';

valid = false;

}

// Description Validation

if (description.length < 10) {

document.getElementById('descriptionError').innerText = 'Description must be at least 10 characters long.';

valid = false;

}

if (valid) {

alert('Product submitted successfully!');

}

}

</script>

</body>

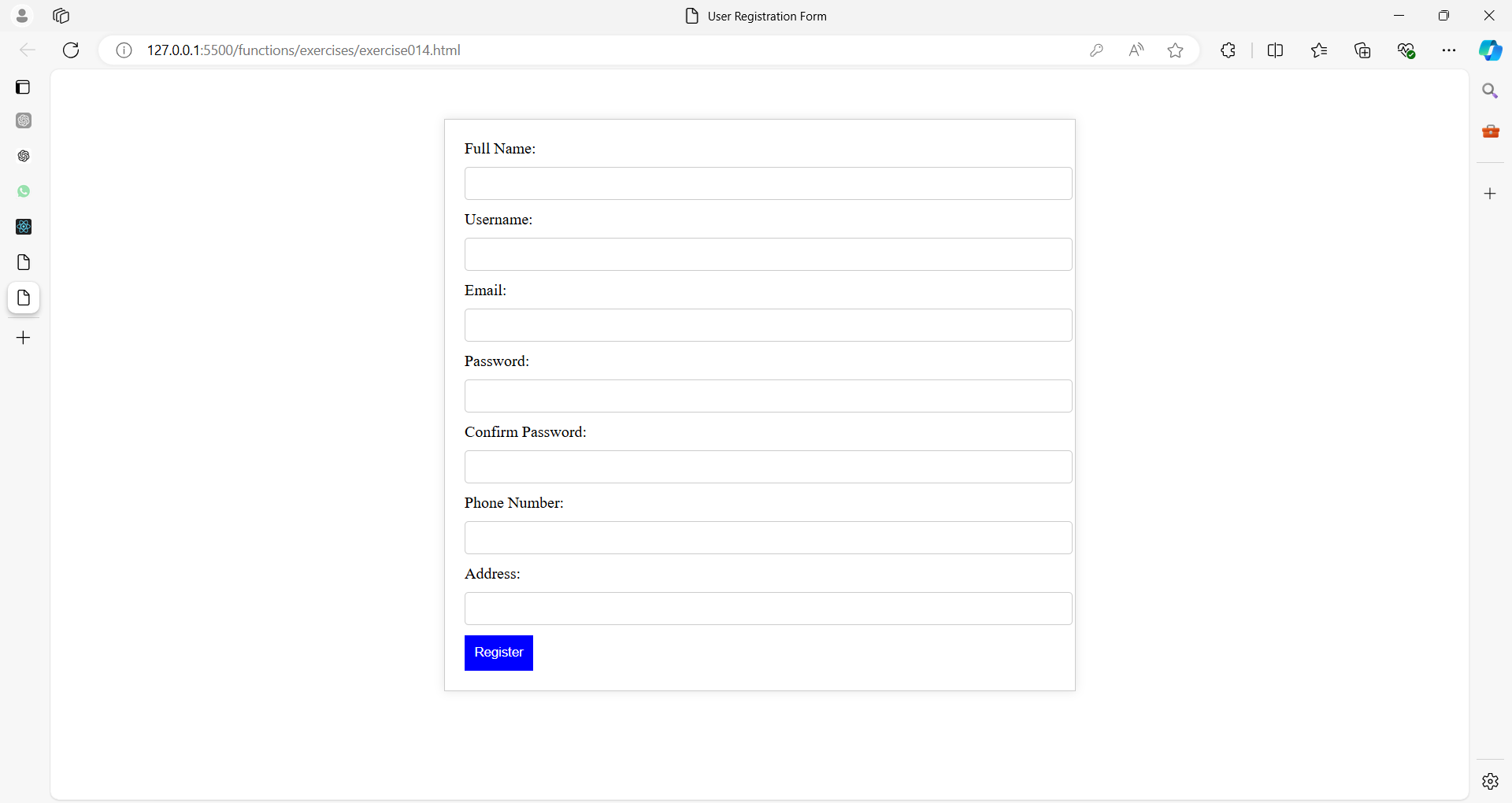
</html>

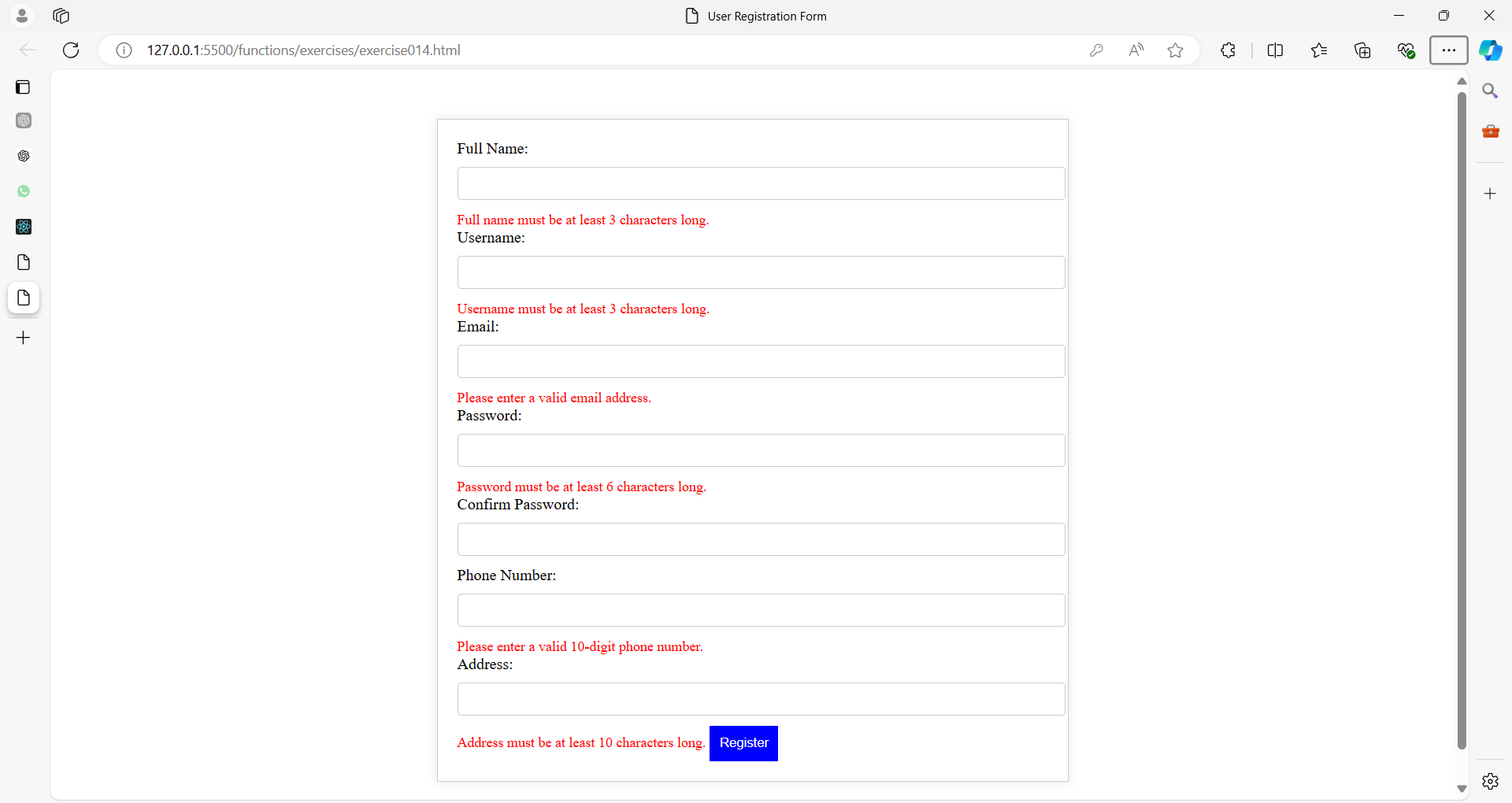
**Exercise 14: User Registration Form**

**Objective**: Create a user registration form with the following fields:

1. Full Name
2. Username
3. Email
4. Password
5. Confirm Password
6. Phone Number
7. Address

Validate each field using JavaScript and display error messages in red.





**HTML**:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>User Registration Form</title>

<style>

form {

width: 600px;

margin: 50px auto;

padding: 20px;

border: 1px solid #ccc;

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

}

label, input {

display: block;

margin-bottom: 10px;

width: 100%;

}

input[type="text"], input[type="email"], input[type="password"], input[type="tel"] {

padding: 8px;

border: 1px solid #ccc;

border-radius: 4px;

}

.error {

color: red;

font-size: 0.9em;

}

button {

padding: 10px;

background-color: blue;

color: white;

border: none;

cursor: pointer;

}

button:hover {

background-color: darkblue;

}

</style>

</head>

<body>

<form id="registrationForm">

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName">

<span id="fullNameError" class="error"></span>

<label for="username">Username:</label>

<input type="text" id="username" name="username">

<span id="usernameError" class="error"></span>

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

<input type="email" id="email" name="email">

<span id="emailError" class="error"></span>

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

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

<span id="passwordError" class="error"></span>

<label for="confirmPassword">Confirm Password:</label>

<input type="password" id="confirmPassword" name="confirmPassword">

<span id="confirmPasswordError" class="error"></span>

<label for="phone">Phone Number:</label>

<input type="tel" id="phone" name="phone">

<span id="phoneError" class="error"></span>

<label for="address">Address:</label>

<input type="text" id="address" name="address">

<span id="addressError" class="error"></span>

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

</form>

<script>

document.getElementById('registrationForm').addEventListener('submit', function(event) {

event.preventDefault();

validateRegistrationForm();

});

function validateRegistrationForm() {

let valid = true;

// Clear previous error messages

document.querySelectorAll('.error').forEach(error => error.innerText = '');

// Get values from form fields

const fullName = document.getElementById('fullName').value;

const username = document.getElementById('username').value;

const email = document.getElementById('email').value;

const password = document.getElementById('password').value;

const confirmPassword = document.getElementById('confirmPassword').value;

const phone = document.getElementById('phone').value;

const address = document.getElementById('address').value;

// Full Name Validation

if (fullName.length < 3) {

document.getElementById('fullNameError').innerText = 'Full name must be at least 3 characters long.';

valid = false;

}

// Username Validation

if (username.length < 3) {

document.getElementById('usernameError').innerText = 'Username must be at least 3 characters long.';

valid = false;

}

// Email Validation

const emailPattern = /^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$/;

if (!emailPattern.test(email)) {

document.getElementById('emailError').innerText = 'Please enter a valid email address.';

valid = false;

}

// Password Validation

if (password.length < 6) {

document.getElementById('passwordError').innerText = 'Password must be at least 6 characters long.';

valid = false;

}

// Confirm Password Validation

if (password !== confirmPassword) {

document.getElementById('confirmPasswordError').innerText = 'Passwords do not match.';

valid = false;

}

// Phone Number Validation

const phonePattern = /^[0-9]{10}$/;

if (!phonePattern.test(phone)) {

document.getElementById('phoneError').innerText = 'Please enter a valid 10-digit phone number.';

valid = false;

}

// Address Validation

if (address.length < 10) {

document.getElementById('addressError').innerText = 'Address must be at least 10 characters long.';

valid = false;

}

if (valid) {

alert('Registration successful!');

}

}

</script>

</body>

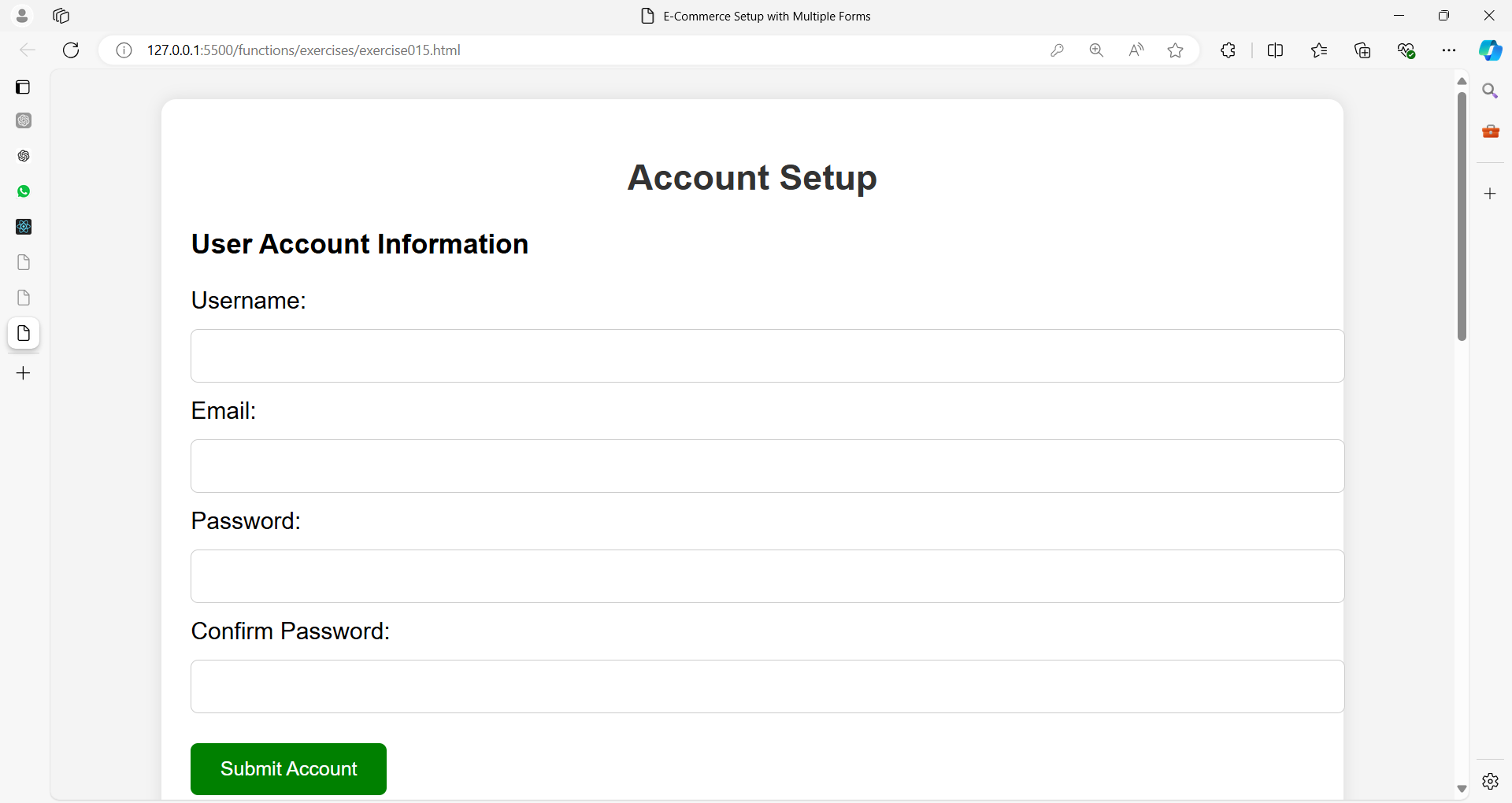
</html>

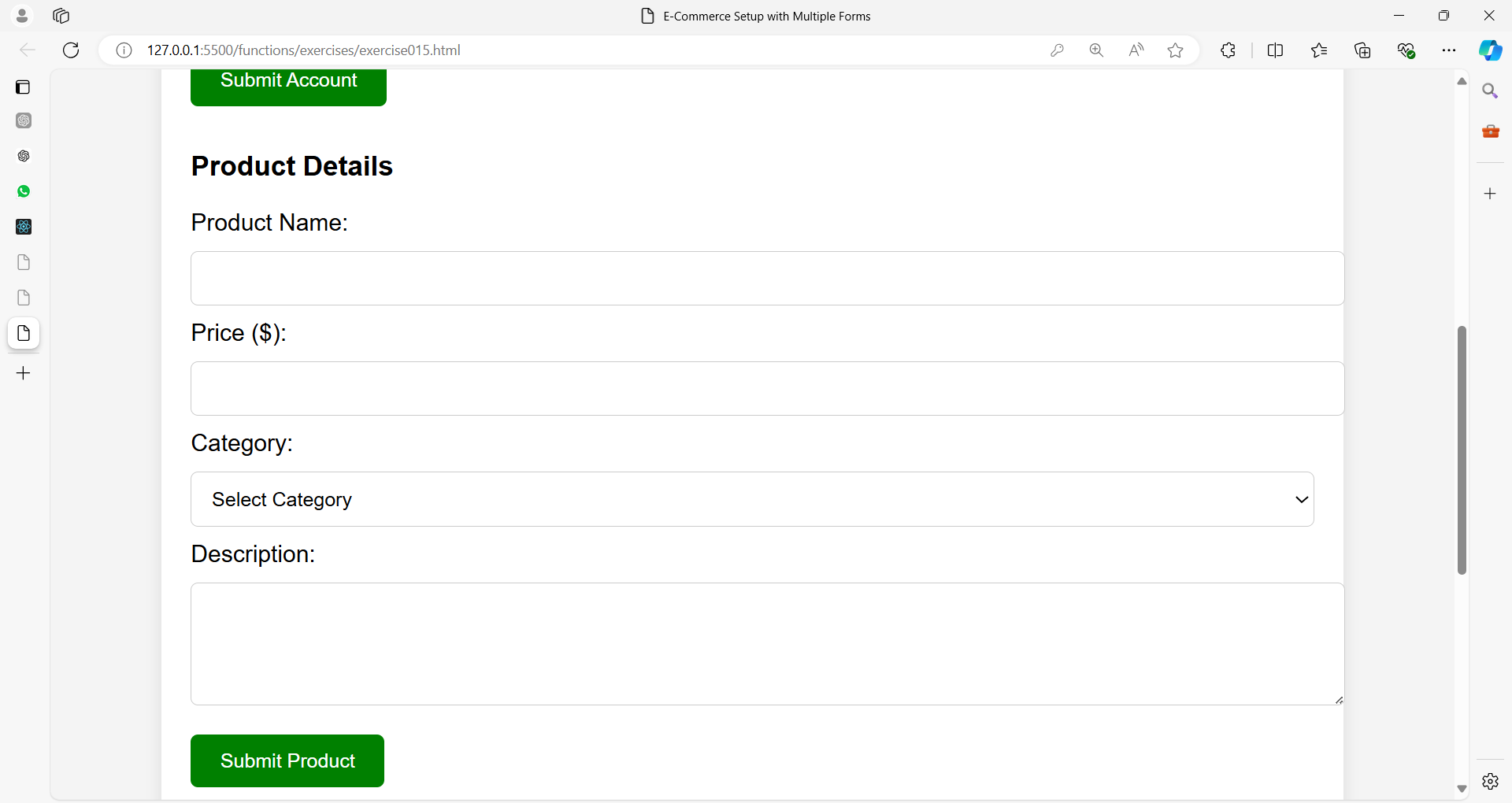
**Exercise 15: E-commerce Setup with Multiple Forms**

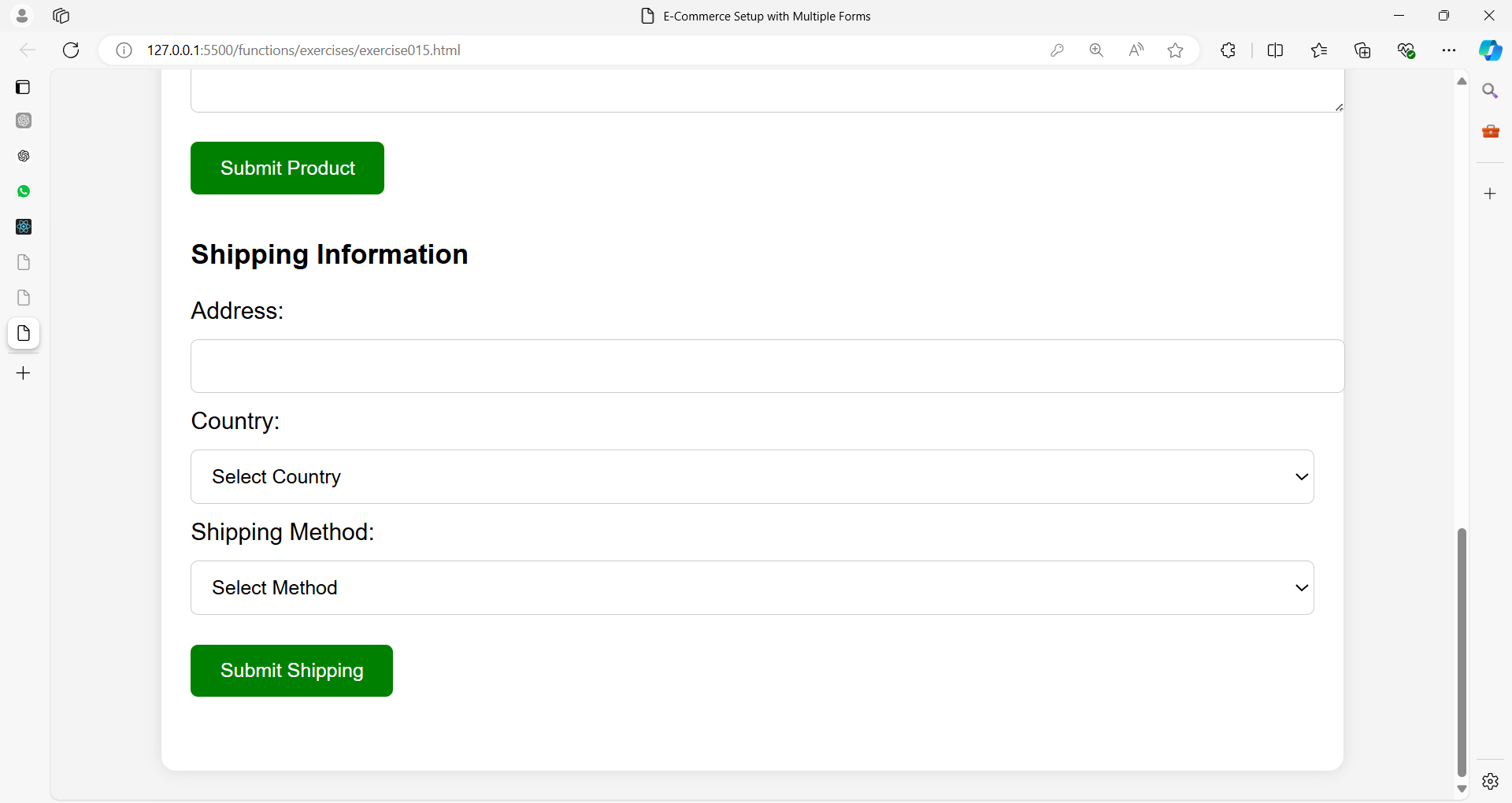
The system consists of three forms:

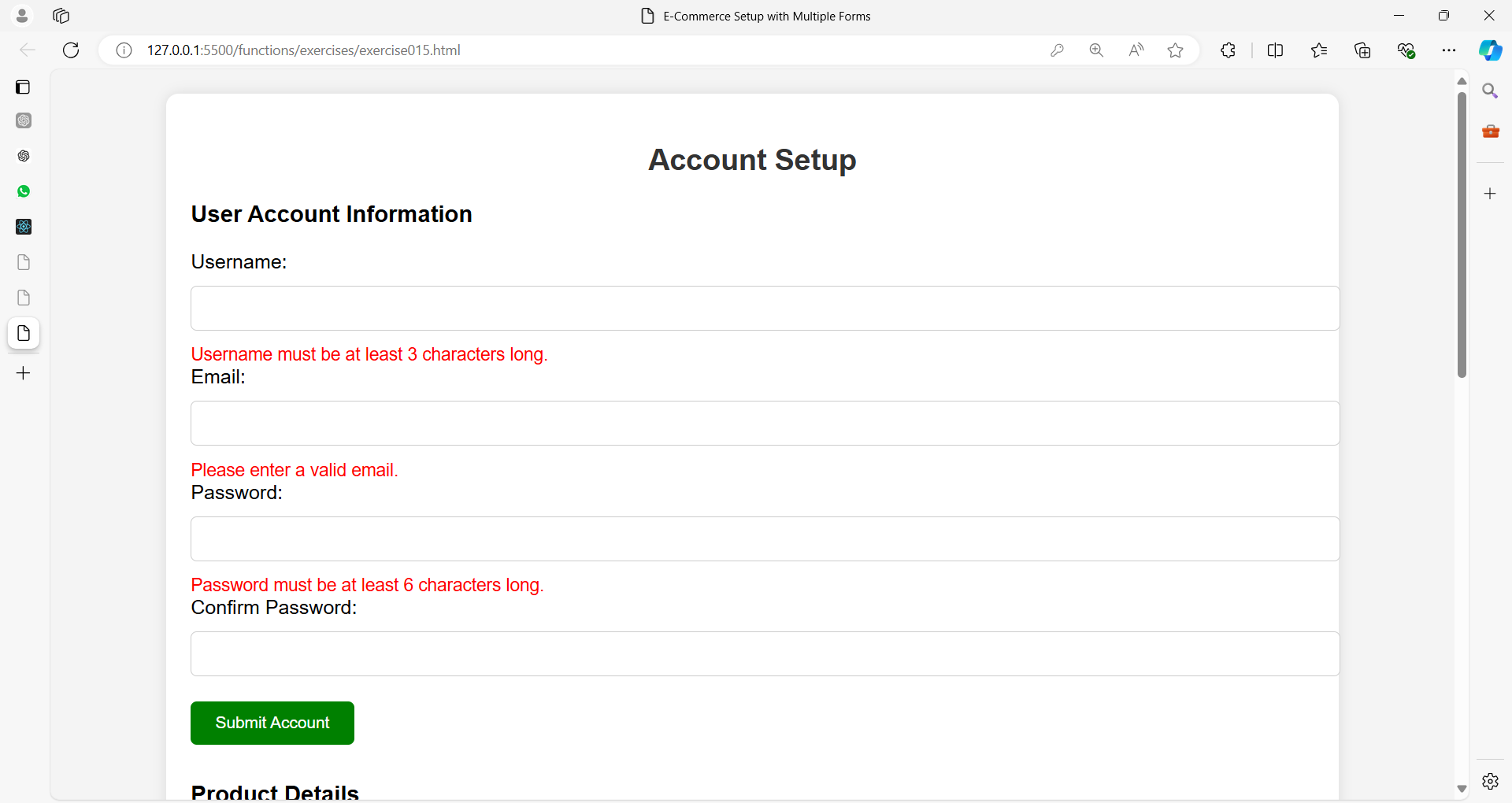
1. **User Account Form**: Collects basic user details like name, email, and password.
2. **Product Details Form**: Gathers information on the product, including name, price, category, and description.
3. **Shipping Information Form**: Collects address, country, and shipping method information.

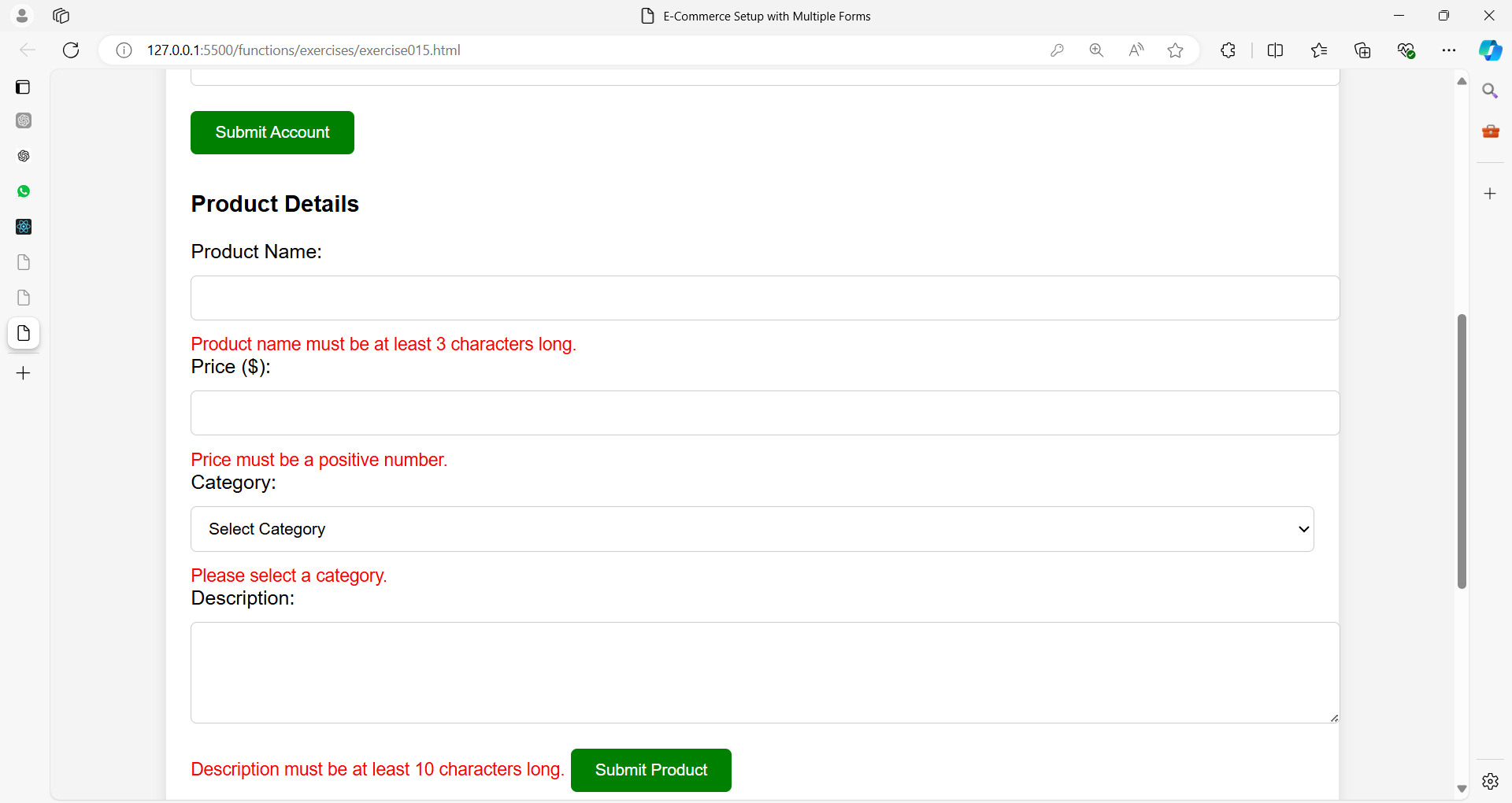
Each form has its own validation, and all forms are validated separately.

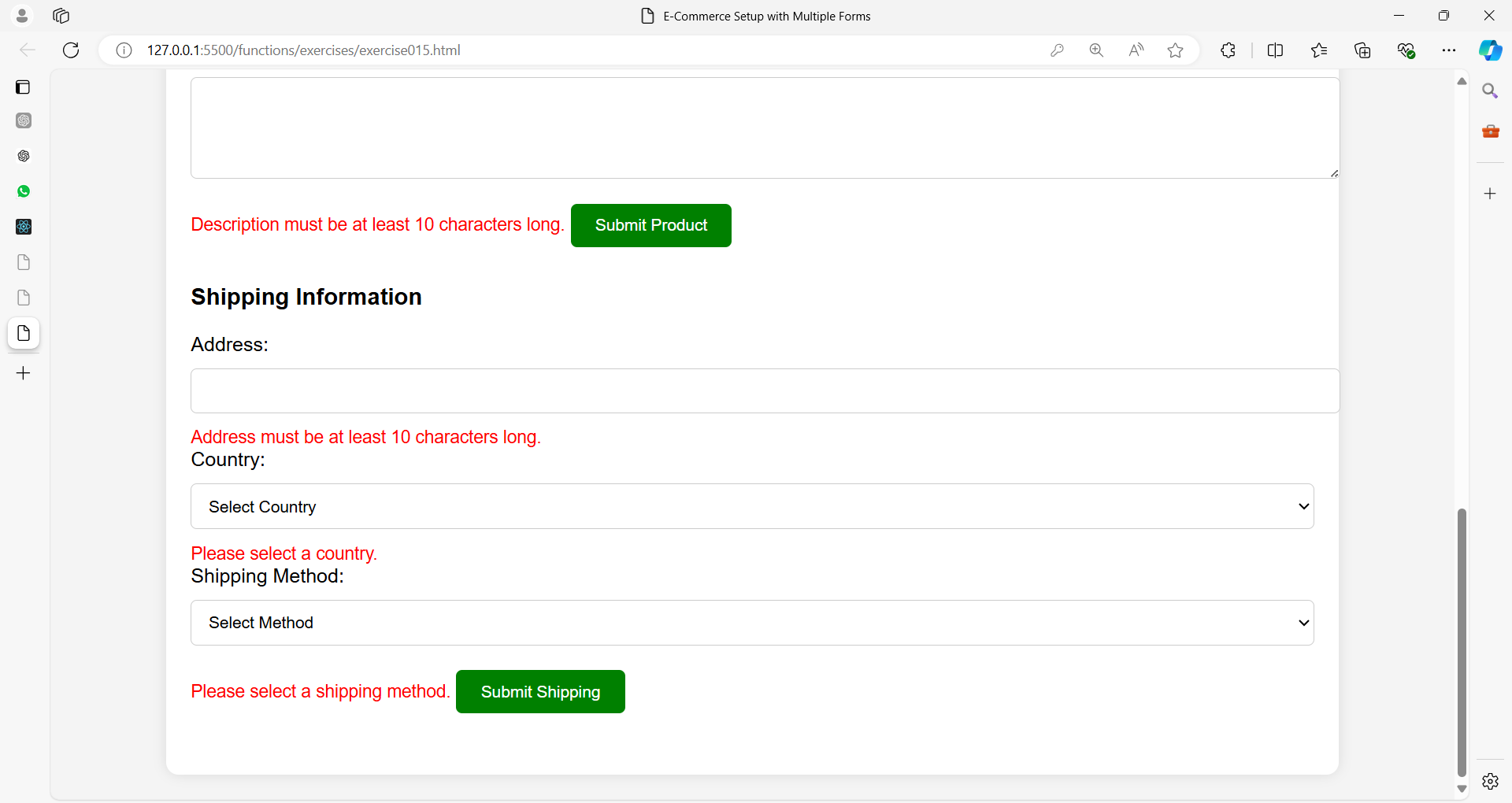












**Full Code Example**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>E-Commerce Setup with Multiple Forms</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.container {

width: 80%;

margin: 20px auto;

padding: 20px;

background-color: white;

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

border-radius: 10px;

}

h2 {

text-align: center;

color: #333;

}

form {

margin-bottom: 30px;

}

label, input, textarea, select {

display: block;

margin-bottom: 10px;

width: 100%;

}

input[type="text"], input[type="password"], input[type="email"], input[type="number"], textarea, select {

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

.error {

color: red;

font-size: 0.9em;

}

button {

padding: 10px 20px;

background-color: green;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

margin-top: 10px;

}

button:hover {

background-color: darkgreen;

}

</style>

</head>

<body>

<div class="container">

<h2>Account Setup</h2>

<!-- Form 1: User Account Information -->

<form id="accountForm">

<h3>User Account Information</h3>

<label for="username">Username:</label>

<input type="text" id="username" name="username">

<span id="usernameError" class="error"></span>

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

<input type="email" id="email" name="email">

<span id="emailError" class="error"></span>

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

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

<span id="passwordError" class="error"></span>

<label for="confirmPassword">Confirm Password:</label>

<input type="password" id="confirmPassword" name="confirmPassword">

<span id="confirmPasswordError" class="error"></span>

<button type="button" onclick="validateAccountForm()">Submit Account</button>

</form>

<!-- Form 2: Product Details -->

<form id="productForm">

<h3>Product Details</h3>

<label for="productName">Product Name:</label>

<input type="text" id="productName" name="productName">

<span id="productNameError" class="error"></span>

<label for="price">Price ($):</label>

<input type="number" id="price" name="price">

<span id="priceError" class="error"></span>

<label for="category">Category:</label>

<select id="category" name="category">

<option value="">Select Category</option>

<option value="electronics">Electronics</option>

<option value="clothing">Clothing</option>

<option value="home">Home</option>

</select>

<span id="categoryError" class="error"></span>

<label for="description">Description:</label>

<textarea id="description" name="description" rows="4"></textarea>

<span id="descriptionError" class="error"></span>

<button type="button" onclick="validateProductForm()">Submit Product</button>

</form>

<!-- Form 3: Shipping Information -->

<form id="shippingForm">

<h3>Shipping Information</h3>

<label for="address">Address:</label>

<input type="text" id="address" name="address">

<span id="addressError" class="error"></span>

<label for="country">Country:</label>

<select id="country" name="country">

<option value="">Select Country</option>

<option value="USA">USA</option>

<option value="Canada">Canada</option>

<option value="UK">UK</option>

</select>

<span id="countryError" class="error"></span>

<label for="shippingMethod">Shipping Method:</label>

<select id="shippingMethod" name="shippingMethod">

<option value="">Select Method</option>

<option value="standard">Standard Shipping</option>

<option value="express">Express Shipping</option>

</select>

<span id="shippingMethodError" class="error"></span>

<button type="button" onclick="validateShippingForm()">Submit Shipping</button>

</form>

</div>

<script>

// Form 1: Validate User Account Form

function validateAccountForm() {

let valid = true;

const username = document.getElementById('username').value;

const email = document.getElementById('email').value;

const password = document.getElementById('password').value;

const confirmPassword = document.getElementById('confirmPassword').value;

// Clear error messages

document.getElementById('usernameError').innerText = '';

document.getElementById('emailError').innerText = '';

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

document.getElementById('confirmPasswordError').innerText = '';

// Username validation

if (username.length < 3) {

document.getElementById('usernameError').innerText = 'Username must be at least 3 characters long.';

valid = false;

}

// Email validation

const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailPattern.test(email)) {

document.getElementById('emailError').innerText = 'Please enter a valid email.';

valid = false;

}

// Password validation

if (password.length < 6) {

document.getElementById('passwordError').innerText = 'Password must be at least 6 characters long.';

valid = false;

}

// Confirm Password validation

if (password !== confirmPassword) {

document.getElementById('confirmPasswordError').innerText = 'Passwords do not match.';

valid = false;

}

if (valid) {

alert('User account form submitted successfully!');

}

}

// Form 2: Validate Product Form

function validateProductForm() {

let valid = true;

const productName = document.getElementById('productName').value;

const price = document.getElementById('price').value;

const category = document.getElementById('category').value;

const description = document.getElementById('description').value;

// Clear error messages

document.getElementById('productNameError').innerText = '';

document.getElementById('priceError').innerText = '';

document.getElementById('categoryError').innerText = '';

document.getElementById('descriptionError').innerText = '';

// Product name validation

if (productName.length < 3) {

document.getElementById('productNameError').innerText = 'Product name must be at least 3 characters long.';

valid = false;

}

// Price validation

if (price <= 0) {

document.getElementById('priceError').innerText = 'Price must be a positive number.';

valid = false;

}

// Category validation

if (category === "") {

document.getElementById('categoryError').innerText = 'Please select a category.';

valid = false;

}

// Description validation

if (description.length < 10) {

document.getElementById('descriptionError').innerText = 'Description must be at least 10 characters long.';

valid = false;

}

if (valid) {

alert('Product form submitted successfully!');

}

}

// Form 3: Validate Shipping Information Form

function validateShippingForm() {

let valid = true;

const address = document.getElementById('address').value;

const country = document.getElementById('country').value;

const shippingMethod = document.getElementById('shippingMethod').value;

// Clear error messages

document.getElementById('addressError').innerText = '';

document.getElementById('countryError').innerText = '';

document.getElementById('shippingMethodError').innerText = '';

// Address validation

if (address.length < 10) {

document.getElementById('addressError').innerText = 'Address must be at least 10 characters long.';

valid = false;

}

// Country validation

if (country === "") {

document.getElementById('countryError').innerText = 'Please select a country.';

valid = false;

}

// Shipping method validation

if (shippingMethod === "") {

document.getElementById('shippingMethodError').innerText = 'Please select a shipping method.';

valid = false;

}

if (valid) {

alert('Shipping form submitted successfully!');

}

}

</script>

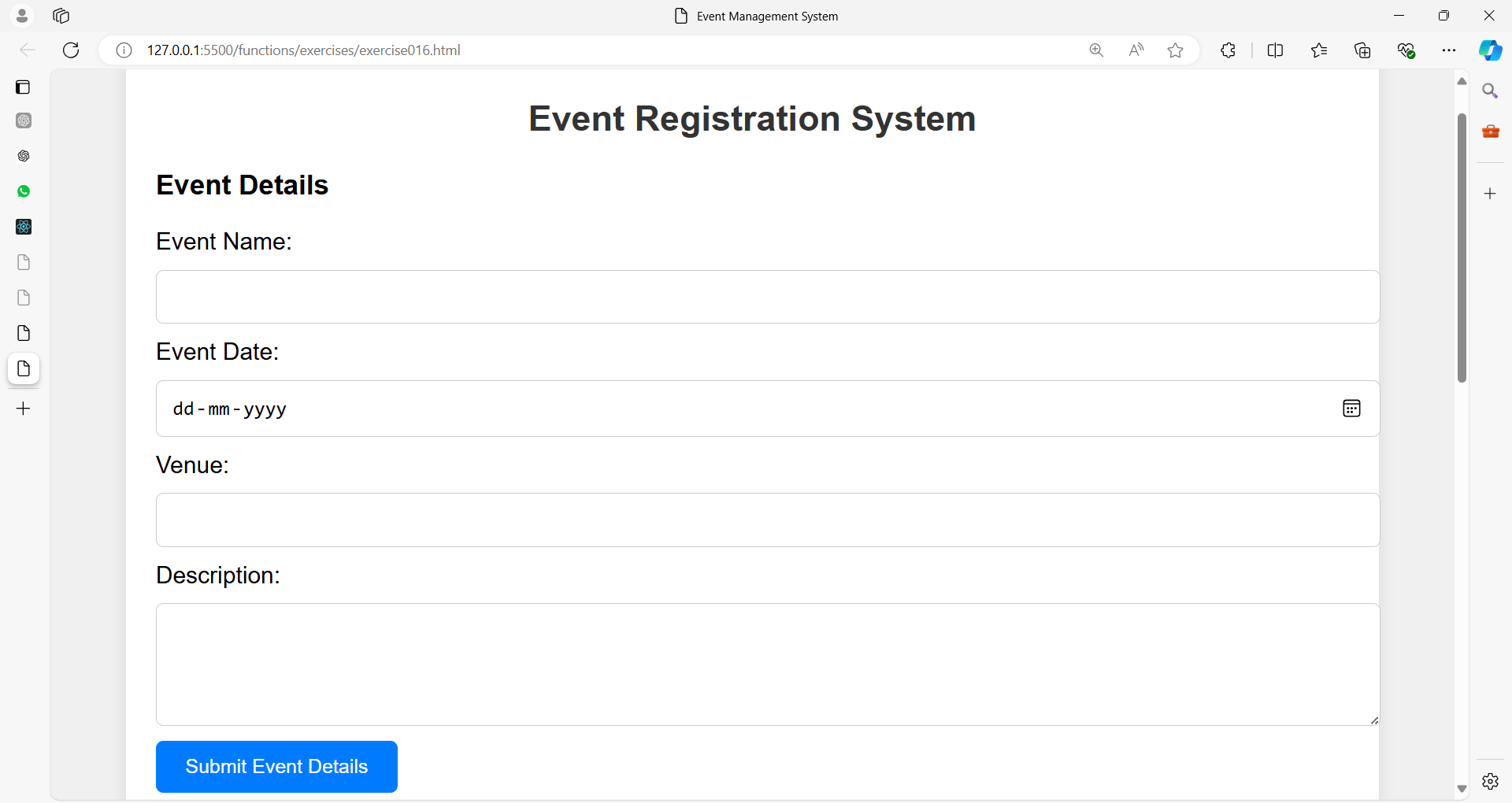
</body>

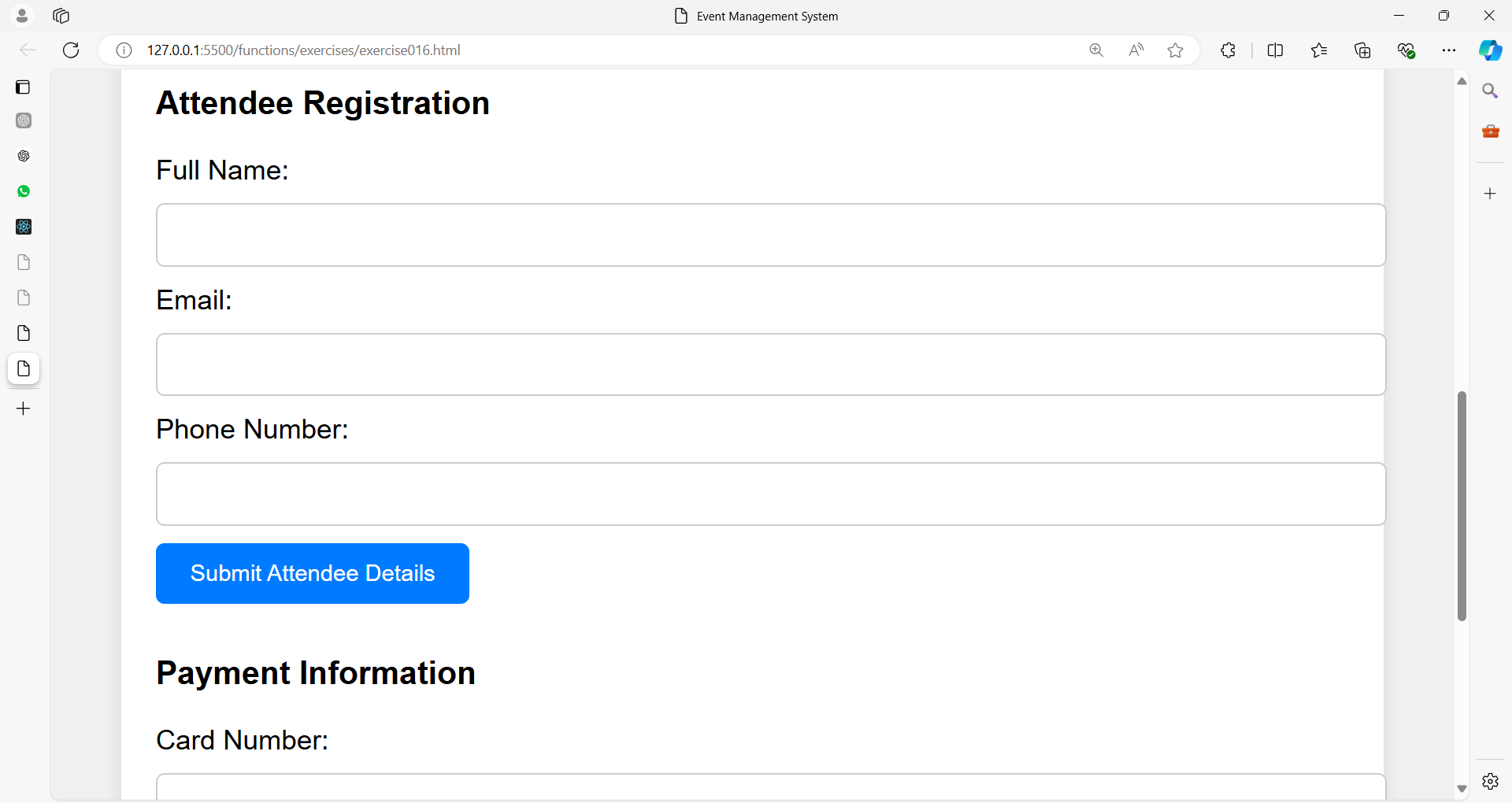
</html>

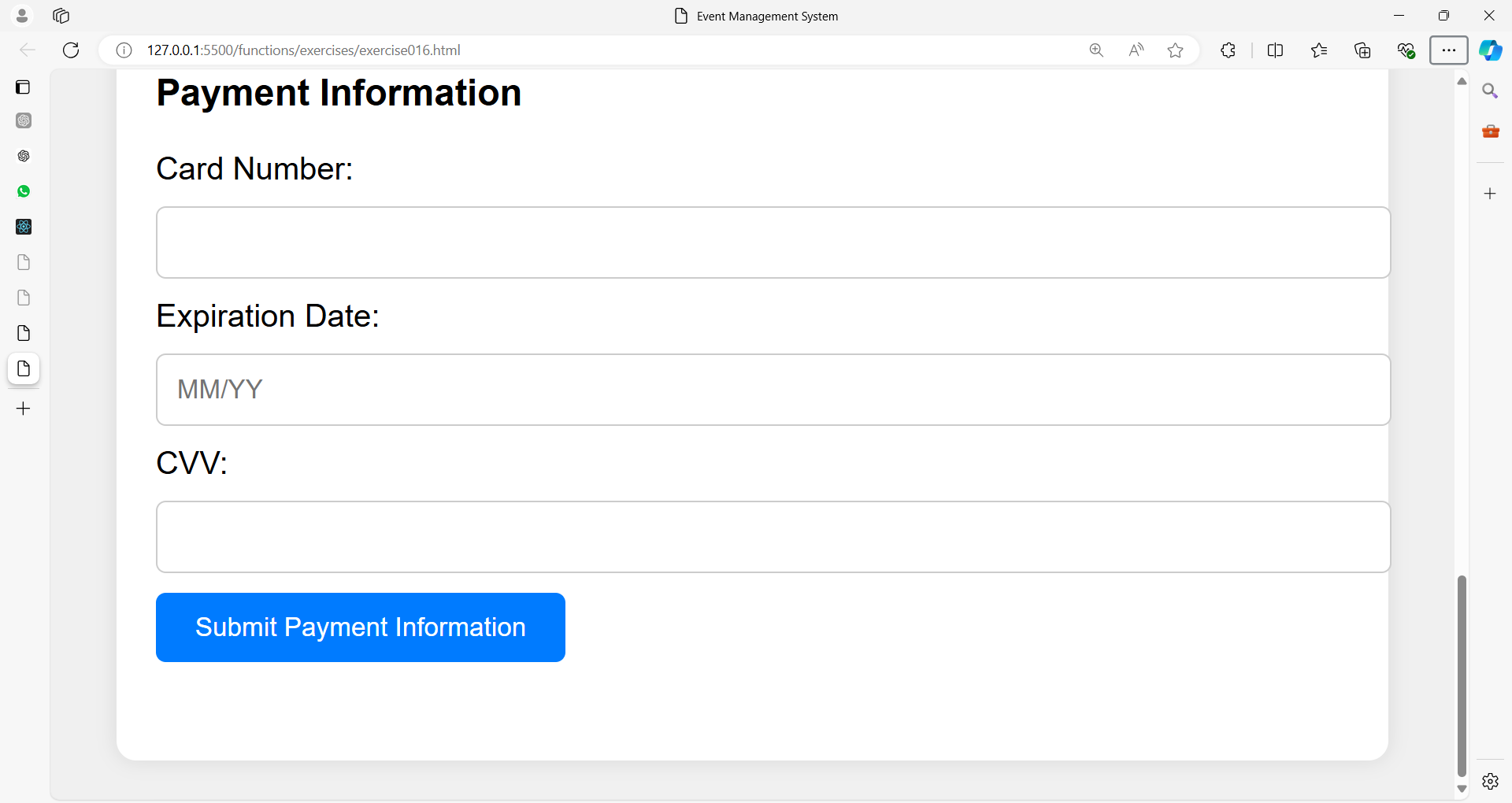
**Exercise 16: Event Registration System with Multiple Forms**

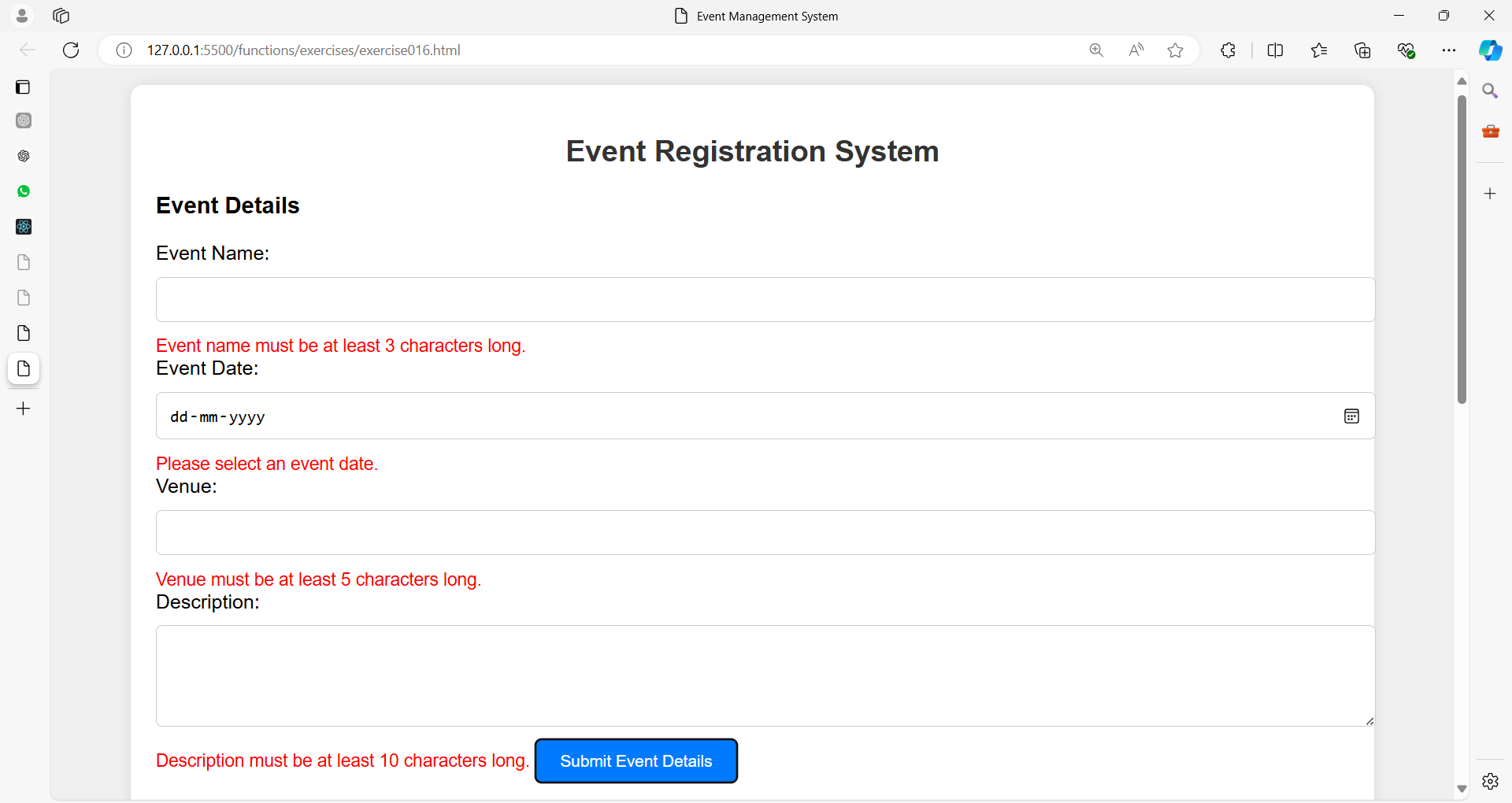
The system consists of:

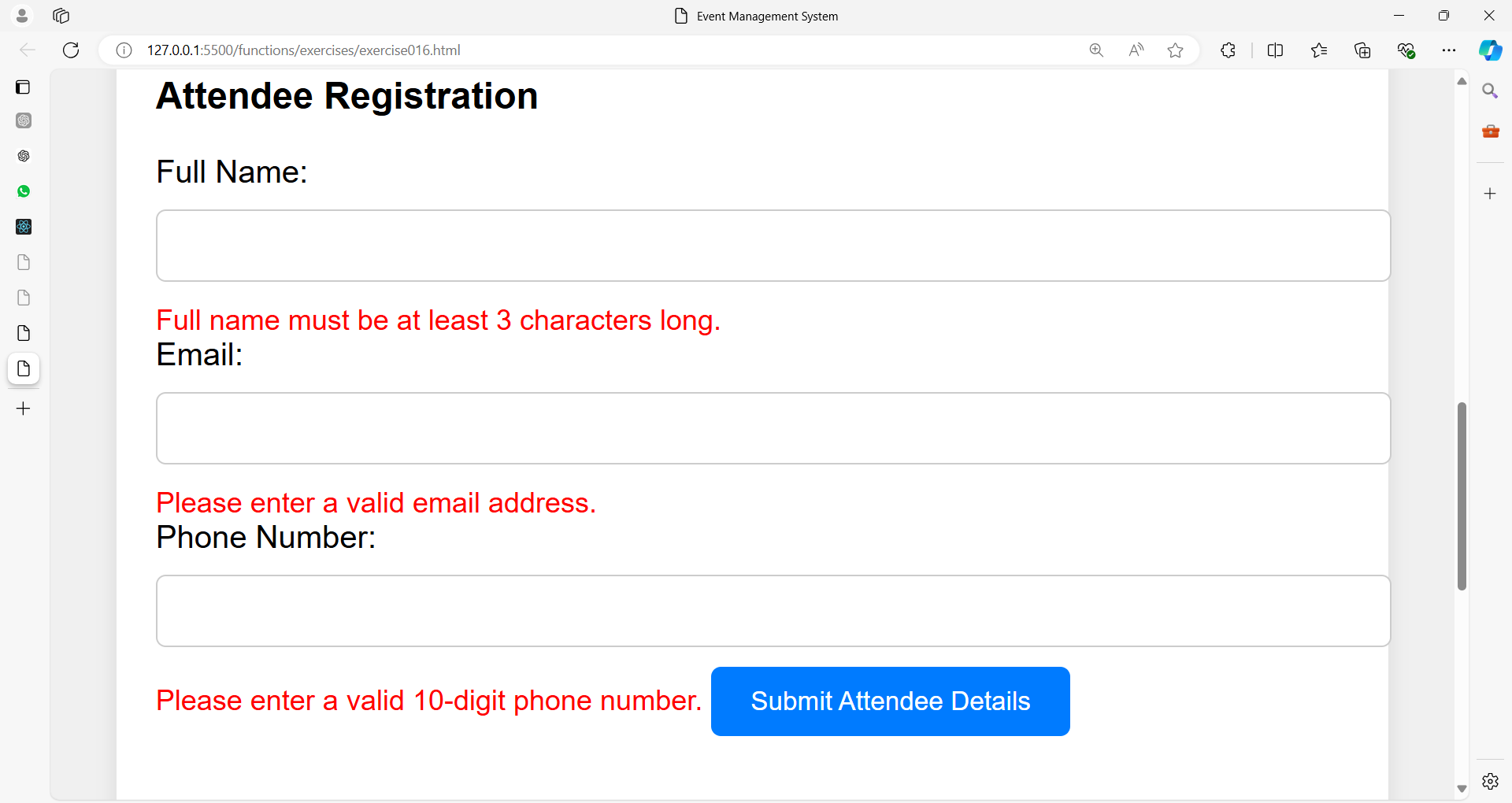
1. **Event Details Form**: Gathers event information, such as event name, date, venue, and description.
2. **Attendee Registration Form**: Collects attendee's personal information, like full name, email, and phone number.
3. **Payment Information Form**: Collects credit card details, including card number, expiration date, and CVV.

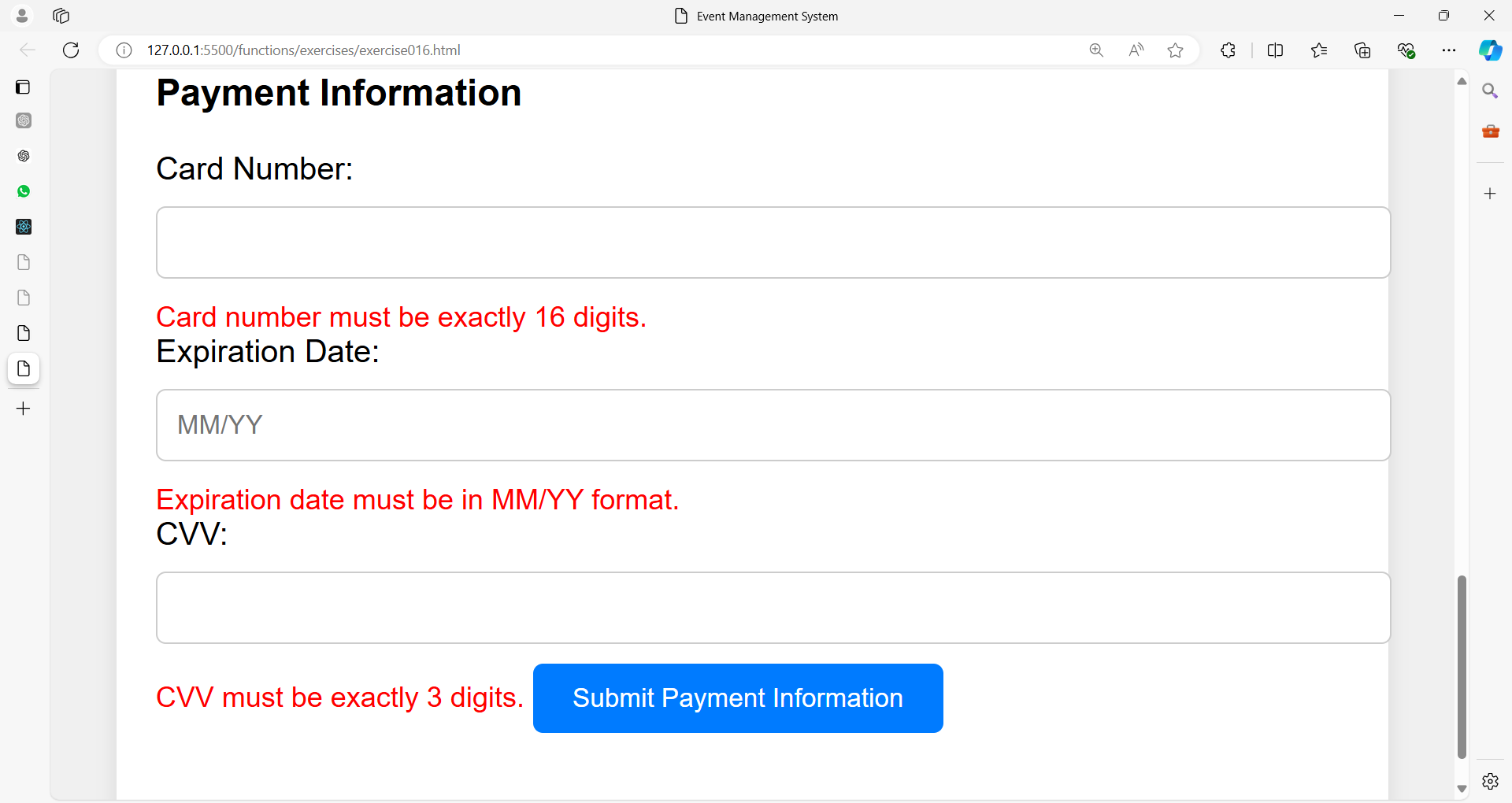












**Full Code Example**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Event Management System</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

padding: 0;

margin: 0;

}

.container {

width: 85%;

margin: 20px auto;

background-color: #fff;

padding: 20px;

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

border-radius: 10px;

}

h2 {

text-align: center;

color: #333;

}

form {

margin-bottom: 30px;

}

label, input, textarea, select {

display: block;

margin-bottom: 10px;

width: 100%;

}

input[type="text"], input[type="email"], input[type="number"], input[type="date"], input[type="tel"], textarea, select {

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

.error {

color: red;

font-size: 0.9em;

}

button {

padding: 10px 20px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<div class="container">

<h2>Event Registration System</h2>

<!-- Form 1: Event Details -->

<form id="eventForm">

<h3>Event Details</h3>

<label for="eventName">Event Name:</label>

<input type="text" id="eventName" name="eventName">

<span id="eventNameError" class="error"></span>

<label for="eventDate">Event Date:</label>

<input type="date" id="eventDate" name="eventDate">

<span id="eventDateError" class="error"></span>

<label for="venue">Venue:</label>

<input type="text" id="venue" name="venue">

<span id="venueError" class="error"></span>

<label for="description">Description:</label>

<textarea id="description" name="description" rows="4"></textarea>

<span id="descriptionError" class="error"></span>

<button type="button" onclick="validateEventForm()">Submit Event Details</button>

</form>

<!-- Form 2: Attendee Registration -->

<form id="attendeeForm">

<h3>Attendee Registration</h3>

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName">

<span id="fullNameError" class="error"></span>

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

<input type="email" id="email" name="email">

<span id="emailError" class="error"></span>

<label for="phone">Phone Number:</label>

<input type="tel" id="phone" name="phone">

<span id="phoneError" class="error"></span>

<button type="button" onclick="validateAttendeeForm()">Submit Attendee Details</button>

</form>

<!-- Form 3: Payment Information -->

<form id="paymentForm">

<h3>Payment Information</h3>

<label for="cardNumber">Card Number:</label>

<input type="number" id="cardNumber" name="cardNumber">

<span id="cardNumberError" class="error"></span>

<label for="expDate">Expiration Date:</label>

<input type="text" id="expDate" name="expDate" placeholder="MM/YY">

<span id="expDateError" class="error"></span>

<label for="cvv">CVV:</label>

<input type="number" id="cvv" name="cvv">

<span id="cvvError" class="error"></span>

<button type="button" onclick="validatePaymentForm()">Submit Payment Information</button>

</form>

</div>

<script>

// Form 1: Validate Event Form

function validateEventForm() {

let valid = true;

const eventName = document.getElementById('eventName').value;

const eventDate = document.getElementById('eventDate').value;

const venue = document.getElementById('venue').value;

const description = document.getElementById('description').value;

// Clear error messages

document.getElementById('eventNameError').innerText = '';

document.getElementById('eventDateError').innerText = '';

document.getElementById('venueError').innerText = '';

document.getElementById('descriptionError').innerText = '';

// Event name validation

if (eventName.length < 3) {

document.getElementById('eventNameError').innerText = 'Event name must be at least 3 characters long.';

valid = false;

}

// Event date validation

if (!eventDate) {

document.getElementById('eventDateError').innerText = 'Please select an event date.';

valid = false;

}

// Venue validation

if (venue.length < 5) {

document.getElementById('venueError').innerText = 'Venue must be at least 5 characters long.';

valid = false;

}

// Description validation

if (description.length < 10) {

document.getElementById('descriptionError').innerText = 'Description must be at least 10 characters long.';

valid = false;

}

if (valid) {

alert('Event details submitted successfully!');

}

}

// Form 2: Validate Attendee Form

function validateAttendeeForm() {

let valid = true;

const fullName = document.getElementById('fullName').value;

const email = document.getElementById('email').value;

const phone = document.getElementById('phone').value;

// Clear error messages

document.getElementById('fullNameError').innerText = '';

document.getElementById('emailError').innerText = '';

document.getElementById('phoneError').innerText = '';

// Full name validation

if (fullName.length < 3) {

document.getElementById('fullNameError').innerText = 'Full name must be at least 3 characters long.';

valid = false;

}

// Email validation

const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailPattern.test(email)) {

document.getElementById('emailError').innerText = 'Please enter a valid email address.';

valid = false;

}

// Phone number validation

const phonePattern = /^[0-9]{10}$/;

if (!phonePattern.test(phone)) {

document.getElementById('phoneError').innerText = 'Please enter a valid 10-digit phone number.';

valid = false;

}

if (valid) {

alert('Attendee details submitted successfully!');

}

}

// Form 3: Validate Payment Form

function validatePaymentForm() {

let valid = true;

const cardNumber = document.getElementById('cardNumber').value;

const expDate = document.getElementById('expDate').value;

const cvv = document.getElementById('cvv').value;

// Clear error messages

document.getElementById('cardNumberError').innerText = '';

document.getElementById('expDateError').innerText = '';

document.getElementById('cvvError').innerText = '';

// Card number validation

if (cardNumber.length !== 16) {

document.getElementById('cardNumberError').innerText = 'Card number must be exactly 16 digits.';

valid = false;

}

// Expiration date validation

const expDatePattern = /^(0[1-9]|1[0-2])\/([0-9]{2})$/;

if (!expDatePattern.test(expDate)) {

document.getElementById('expDateError').innerText = 'Expiration date must be in MM/YY format.';

valid = false;

}

// CVV validation

if (cvv.length !== 3) {

document.getElementById('cvvError').innerText = 'CVV must be exactly 3 digits.';

valid = false;

}

if (valid) {

alert('Payment information submitted successfully!');

}

}

</script>

</body>

</html>

**Exercise 17: Multi-Step Job Application Form**

**Problem Statement:**

You are tasked with creating a **multi-step job application form**.

The form should be broken down into three steps:

1. **Personal Information**
   * Full Name
   * Email
   * Phone Number
2. **Educational Background**
   * Highest Degree
   * University/College
   * Graduation Year
3. **Work Experience**
   * Company Name
   * Job Title
   * Years of Experience

**Requirements:**

1. **Form Layout**:
   * Use the <fieldset> and <legend> tags to group related form controls.
   * Each section should be displayed as a separate form that appears one after the other.
   * Include "Next" and "Previous" buttons to navigate between the sections.
   * The forms should be hidden and revealed as the user progresses through the steps.
2. **Validation**:
   * Perform validation for each form before allowing the user to proceed to the next section:
     + **Full Name** must be at least 3 characters long.
     + **Email** must follow a valid email format.
     + **Phone Number** must be exactly 10 digits long.
     + **Highest Degree** and **Company Name** should be at least 2 characters long.
     + **Graduation Year** should be between 1900 and the current year.
     + **Years of Experience** should be a number between 0 and 50.
3. **JavaScript**:
   * Write JavaScript code to handle the validation of each form.
   * Use error messages displayed in red if the input is invalid.
   * Implement the logic to show/hide forms based on user navigation (Next/Previous).
4. **Styling**:
   * Style the form elements using CSS. The form should be visually appealing and responsive.
   * Error messages should be displayed in red under each invalid input field.
   * Ensure that the form is laid out clearly and is easy to use.

**Additional Requirements:**

* Ensure that all three forms are submitted as a single entity after validation.
* Display a success message using alert() or on the page itself when the form is submitted successfully.

**Example of Inputs:**

**Step 1**: Personal Information

Full Name: John Doe

Email: johndoe@example.com

Phone Number: 1234567890

**Step 2**: Educational Background

Highest Degree: Bachelor of Science

University/College: XYZ University

Graduation Year: 2020

**Step 3**: Work Experience

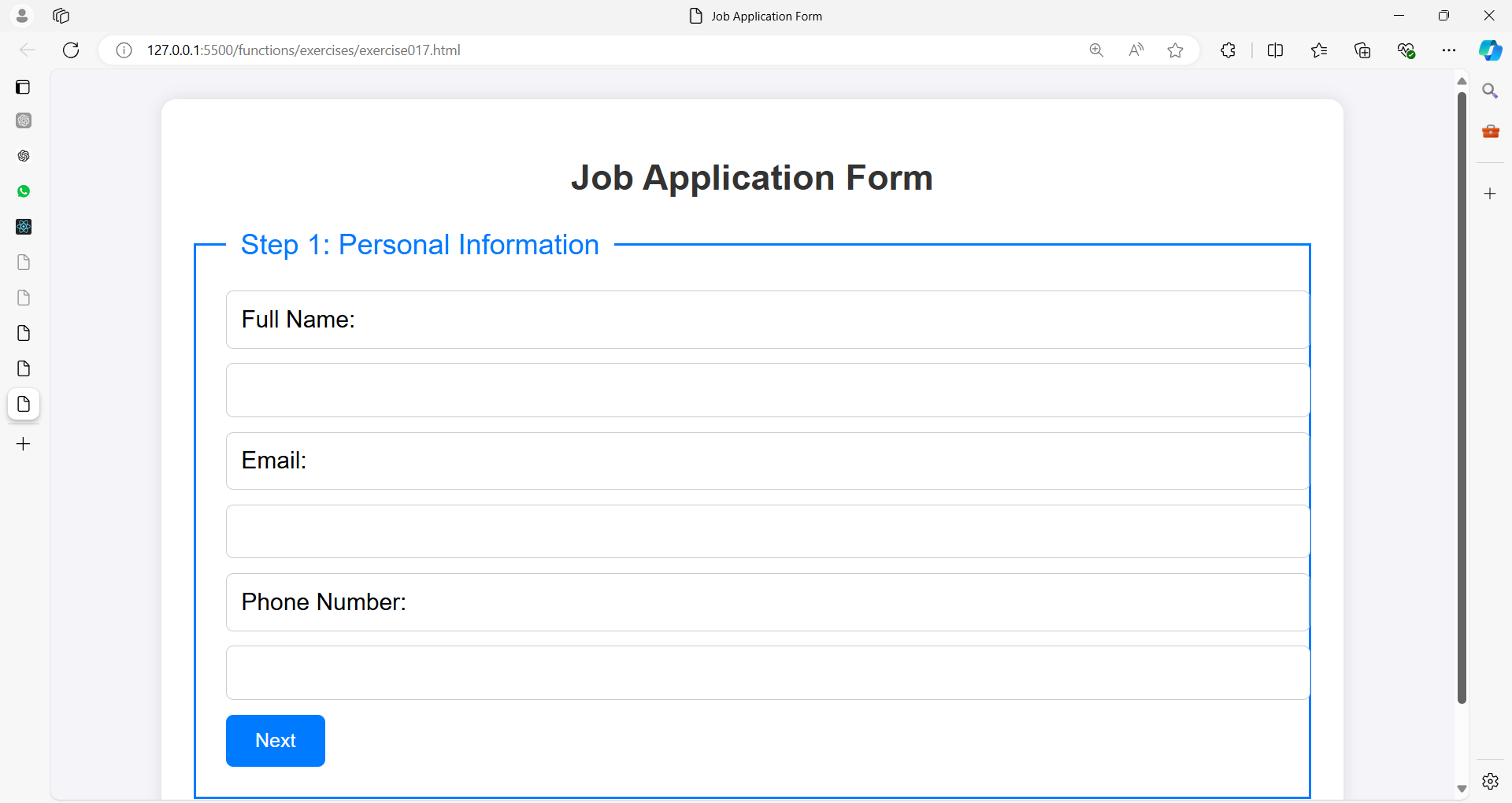
Company Name: ABC Corp

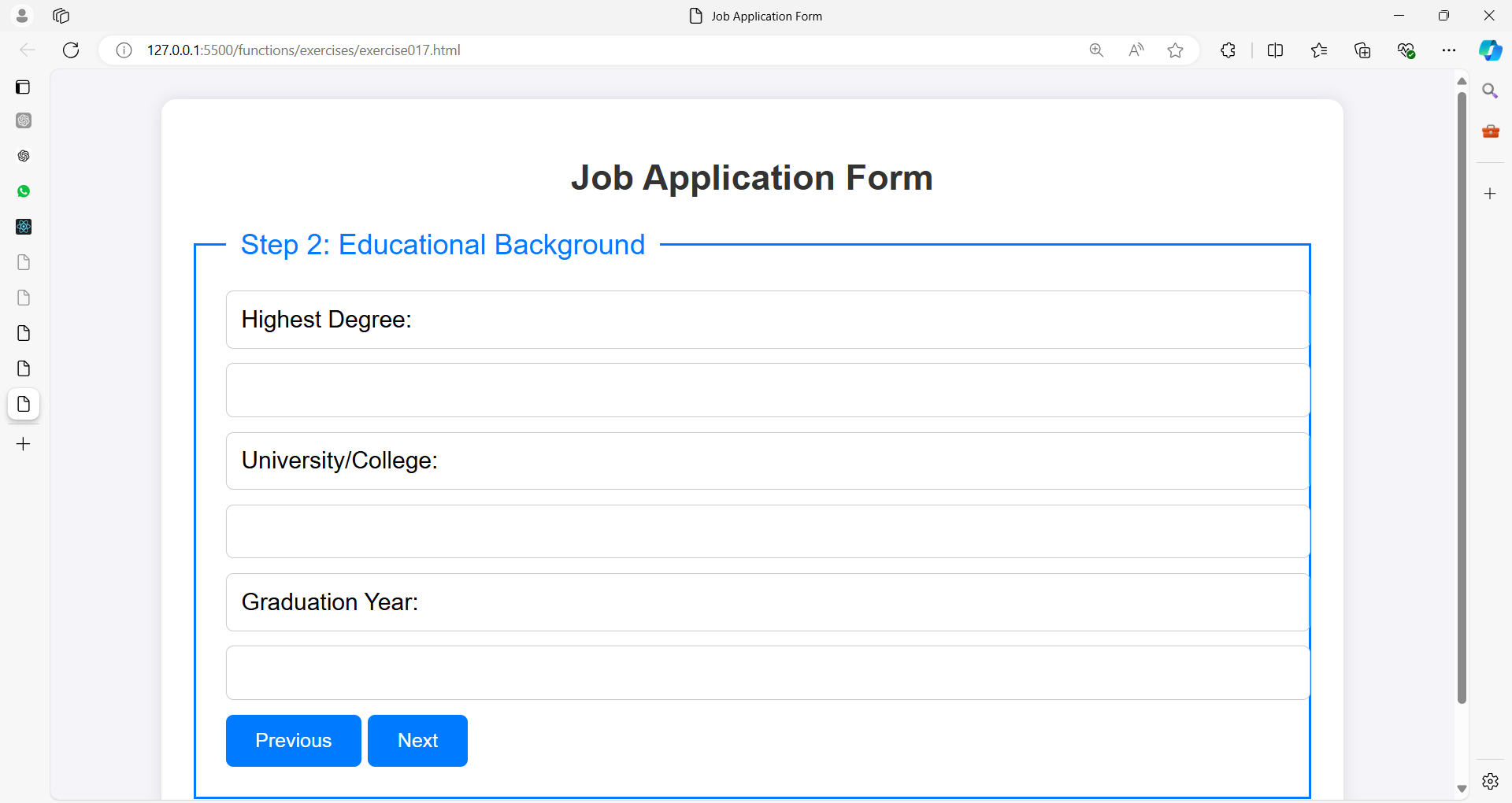
Job Title: Software Engineer

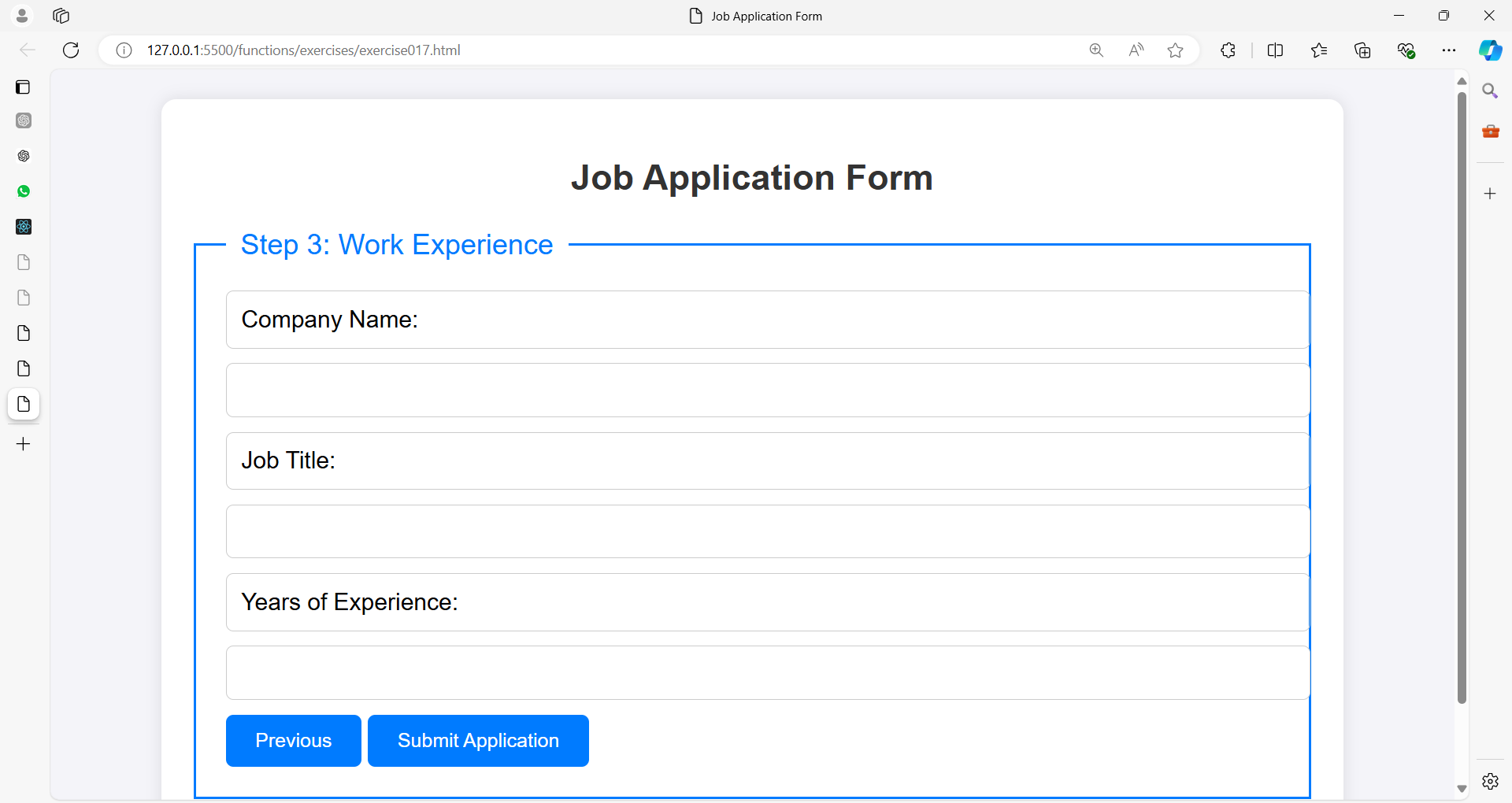
Years of Experience: 5

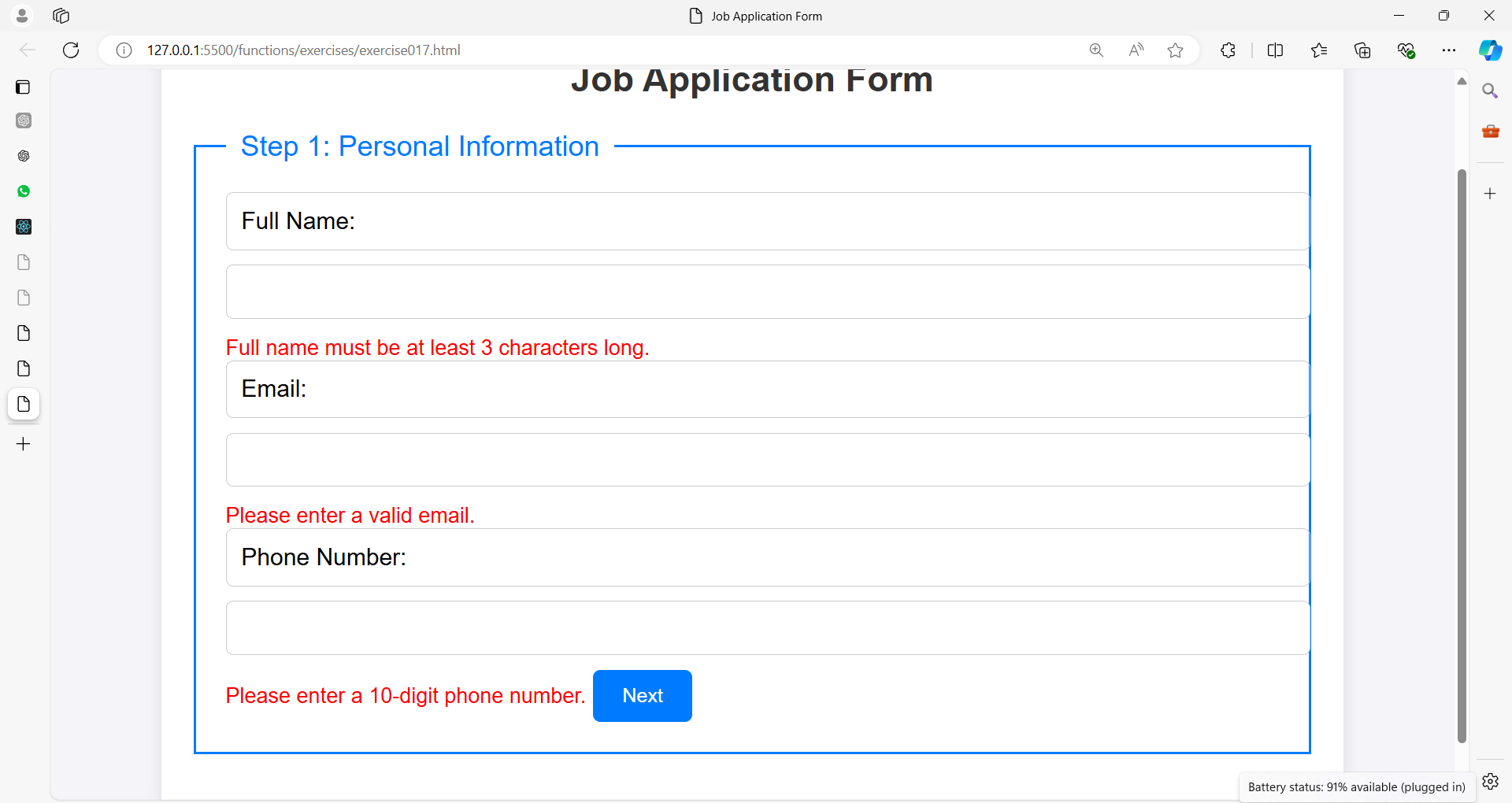
**Deliverables:**

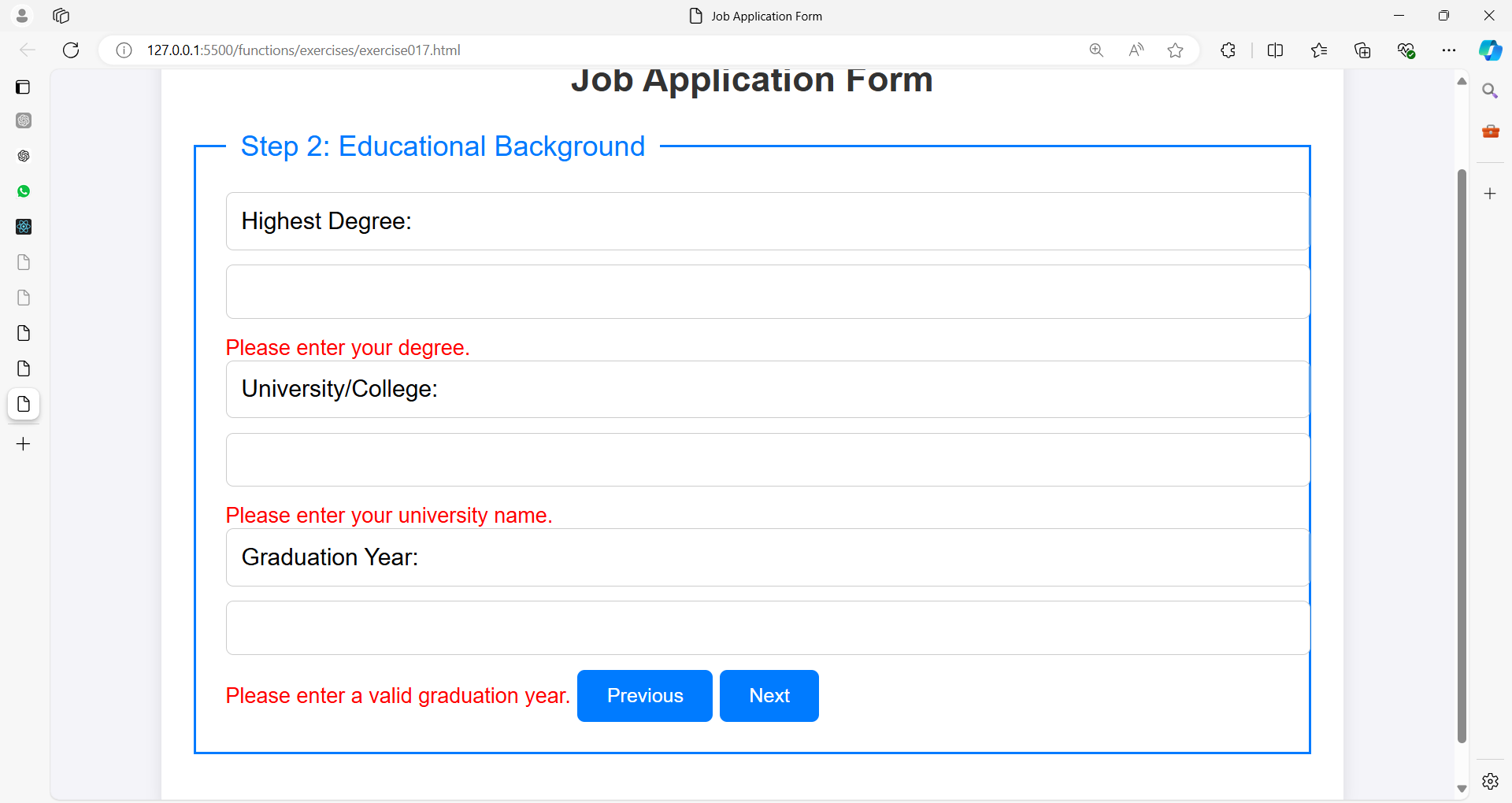
1. HTML code for the job application form.
2. CSS code to style the form and error messages.
3. JavaScript code for validation and form navigation.

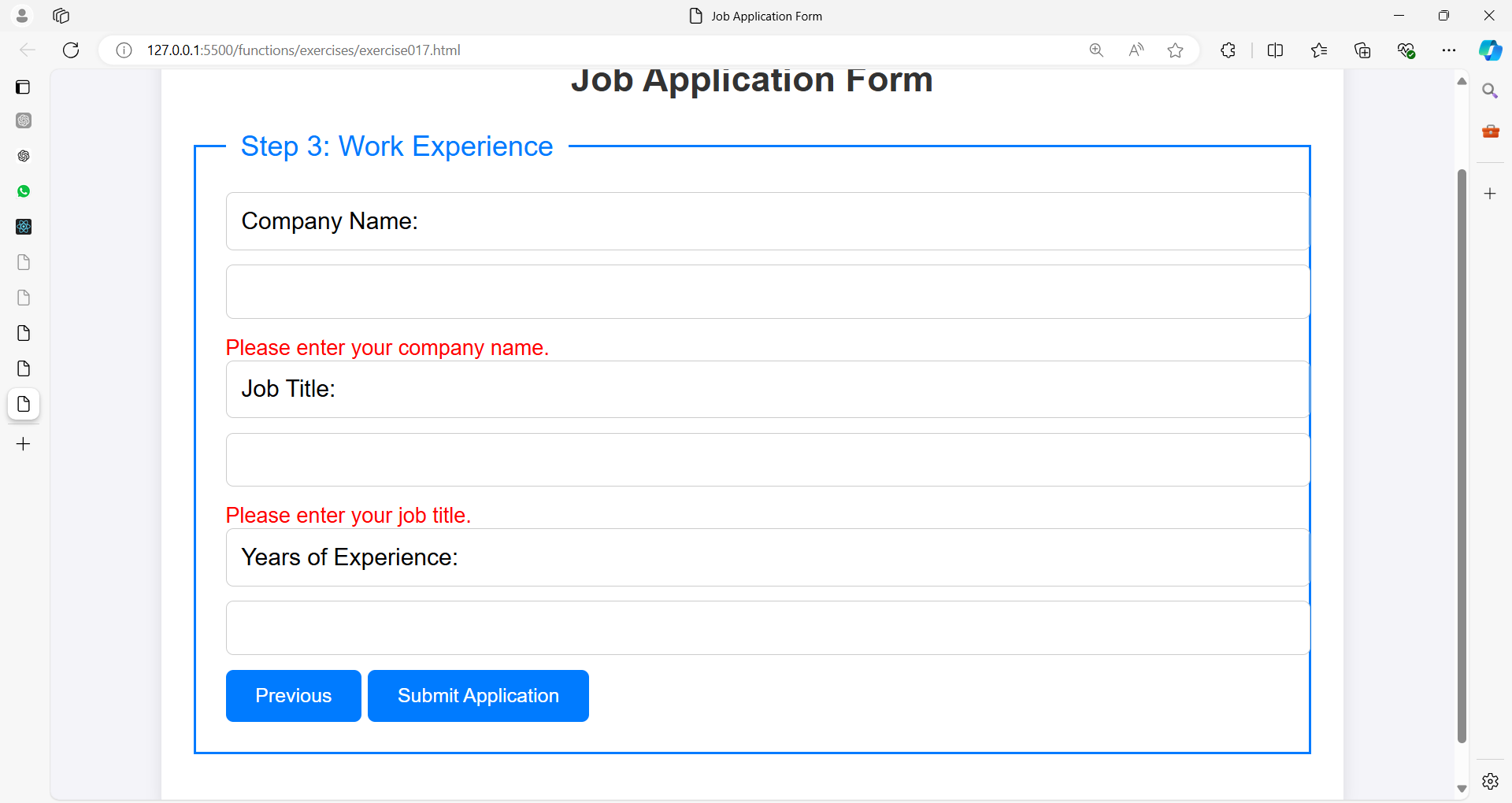












<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Job Application Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f9;

margin: 0;

padding: 0;

}

.container {

width: 80%;

margin: 20px auto;

background-color: #fff;

padding: 20px;

border-radius: 10px;

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

}

h2 {

text-align: center;

color: #333;

}

fieldset {

margin-bottom: 20px;

border: 2px solid #007bff;

padding: 20px;

}

legend {

font-size: 1.2em;

color: #007bff;

padding: 0 10px;

}

label, input, select, textarea {

display: block;

width: 100%;

margin-bottom: 10px;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

.error {

color: red;

font-size: 0.9em;

}

button {

padding: 10px 20px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

.hidden {

display: none;

}

</style>

</head>

<body>

<div class="container">

<h2>Job Application Form</h2>

<!-- Form 1: Personal Information -->

<form id="personalInfoForm">

<fieldset>

<legend>Step 1: Personal Information</legend>

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName">

<span id="fullNameError" class="error"></span>

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

<input type="email" id="email" name="email">

<span id="emailError" class="error"></span>

<label for="phone">Phone Number:</label>

<input type="tel" id="phone" name="phone">

<span id="phoneError" class="error"></span>

<button type="button" onclick="nextStep('personalInfoForm', 'educationForm')">Next</button>

</fieldset>

</form>

<!-- Form 2: Educational Background -->

<form id="educationForm" class="hidden">

<fieldset>

<legend>Step 2: Educational Background</legend>

<label for="highestDegree">Highest Degree:</label>

<input type="text" id="highestDegree" name="highestDegree">

<span id="degreeError" class="error"></span>

<label for="university">University/College:</label>

<input type="text" id="university" name="university">

<span id="universityError" class="error"></span>

<label for="gradYear">Graduation Year:</label>

<input type="number" id="gradYear" name="gradYear" min="1900" max="2100">

<span id="gradYearError" class="error"></span>

<button type="button" onclick="prevStep('educationForm', 'personalInfoForm')">Previous</button>

<button type="button" onclick="nextStep('educationForm', 'experienceForm')">Next</button>

</fieldset>

</form>

<!-- Form 3: Work Experience -->

<form id="experienceForm" class="hidden">

<fieldset>

<legend>Step 3: Work Experience</legend>

<label for="company">Company Name:</label>

<input type="text" id="company" name="company">

<span id="companyError" class="error"></span>

<label for="jobTitle">Job Title:</label>

<input type="text" id="jobTitle" name="jobTitle">

<span id="jobTitleError" class="error"></span>

<label for="yearsExp">Years of Experience:</label>

<input type="number" id="yearsExp" name="yearsExp" min="0" max="50">

<span id="yearsExpError" class="error"></span>

<button type="button" onclick="prevStep('experienceForm', 'educationForm')">Previous</button>

<button type="button" onclick="validateAllForms()">Submit Application</button>

</fieldset>

</form>

</div>

<script>

// Move to the next form

function nextStep(currentForm, nextForm) {

document.getElementById(currentForm).classList.add('hidden');

document.getElementById(nextForm).classList.remove('hidden');

}

// Move to the previous form

function prevStep(currentForm, prevForm) {

document.getElementById(currentForm).classList.add('hidden');

document.getElementById(prevForm).classList.remove('hidden');

}

// Validate all forms

function validateAllForms() {

let valid = true;

// Personal Information Form Validation

const fullName = document.getElementById('fullName').value;

const email = document.getElementById('email').value;

const phone = document.getElementById('phone').value;

document.getElementById('fullNameError').innerText = '';

document.getElementById('emailError').innerText = '';

document.getElementById('phoneError').innerText = '';

if (fullName.length < 3) {

document.getElementById('fullNameError').innerText = 'Full name must be at least 3 characters long.';

valid = false;

}

const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailPattern.test(email)) {

document.getElementById('emailError').innerText = 'Please enter a valid email.';

valid = false;

}

const phonePattern = /^[0-9]{10}$/;

if (!phonePattern.test(phone)) {

document.getElementById('phoneError').innerText = 'Please enter a 10-digit phone number.';

valid = false;

}

// Educational Background Form Validation

const highestDegree = document.getElementById('highestDegree').value;

const university = document.getElementById('university').value;

const gradYear = document.getElementById('gradYear').value;

document.getElementById('degreeError').innerText = '';

document.getElementById('universityError').innerText = '';

document.getElementById('gradYearError').innerText = '';

if (highestDegree.length < 2) {

document.getElementById('degreeError').innerText = 'Please enter your degree.';

valid = false;

}

if (university.length < 3) {

document.getElementById('universityError').innerText = 'Please enter your university name.';

valid = false;

}

if (gradYear < 1900 || gradYear > new Date().getFullYear()) {

document.getElementById('gradYearError').innerText = 'Please enter a valid graduation year.';

valid = false;

}

// Work Experience Form Validation

const company = document.getElementById('company').value;

const jobTitle = document.getElementById('jobTitle').value;

const yearsExp = document.getElementById('yearsExp').value;

document.getElementById('companyError').innerText = '';

document.getElementById('jobTitleError').innerText = '';

document.getElementById('yearsExpError').innerText = '';

if (company.length < 2) {

document.getElementById('companyError').innerText = 'Please enter your company name.';

valid = false;

}

if (jobTitle.length < 2) {

document.getElementById('jobTitleError').innerText = 'Please enter your job title.';

valid = false;

}

if (yearsExp < 0 || yearsExp > 50) {

document.getElementById('yearsExpError').innerText = 'Please enter valid years of experience.';

valid = false;

}

// If valid, show success message

if (valid) {

alert('Job application submitted successfully!');

}

}

</script>

</body>

</html>

**Exercise 18: Multi-Step Product Order Form**

**Problem Statement:**

Create a **multi-step product order form** that allows users to place an order for a product. The form should be divided into three steps:

1. **Product Selection**
   * Product Name
   * Category (Dropdown: Electronics, Furniture, Clothing, etc.)
   * Quantity
2. **Shipping Information**
   * Full Name
   * Address
   * City
   * Postal Code
   * Country (Dropdown with at least 5 countries)
3. **Payment Information**
   * Cardholder's Name
   * Credit Card Number
   * Expiration Date
   * CVV

**Requirements:**

1. **Form Layout**:
   * Use the <fieldset> and <legend> tags to clearly group form fields.
   * Display one form section at a time (Product Selection, Shipping Information, Payment Information).
   * Implement "Next" and "Previous" buttons to navigate between the steps.
   * Each step should be validated before moving to the next section.
2. **Validation**:
   * Implement validation checks in JavaScript:
     + **Product Name** must be at least 3 characters.
     + **Quantity** must be a number greater than 0.
     + **Full Name** must be at least 3 characters.
     + **Postal Code** must be a 5-digit number.
     + **Credit Card Number** must be 16 digits long.
     + **Expiration Date** must be a valid future date.
     + **CVV** must be exactly 3 digits.
3. **JavaScript**:
   * Write JavaScript to validate each form step independently.
   * Display error messages in red if validation fails.
   * Use JavaScript to show or hide each form step based on user interaction.
4. **CSS**:
   * Style the form fields, buttons, and error messages.
   * The form should be user-friendly and visually appealing.
   * Error messages should appear below the relevant input fields in red.
   * Ensure the form layout is responsive.
5. **Submission**:
   * Once the final step is completed and all fields are valid, display a message indicating successful order submission.
   * Use either an alert() or display the success message on the page.

**Example Inputs:**

**Step 1**: Product Selection

Product Name: Laptop

Category: Electronics

Quantity: 2

**Step 2**: Shipping Information

Full Name: Jane Doe

Address: 123 Main St

City: New York

Postal Code: 12345

Country: United States

**Step 3**: Payment Information

Cardholder's Name: Jane Doe

Credit Card Number: 1234567812345678

Expiration Date: 12/25

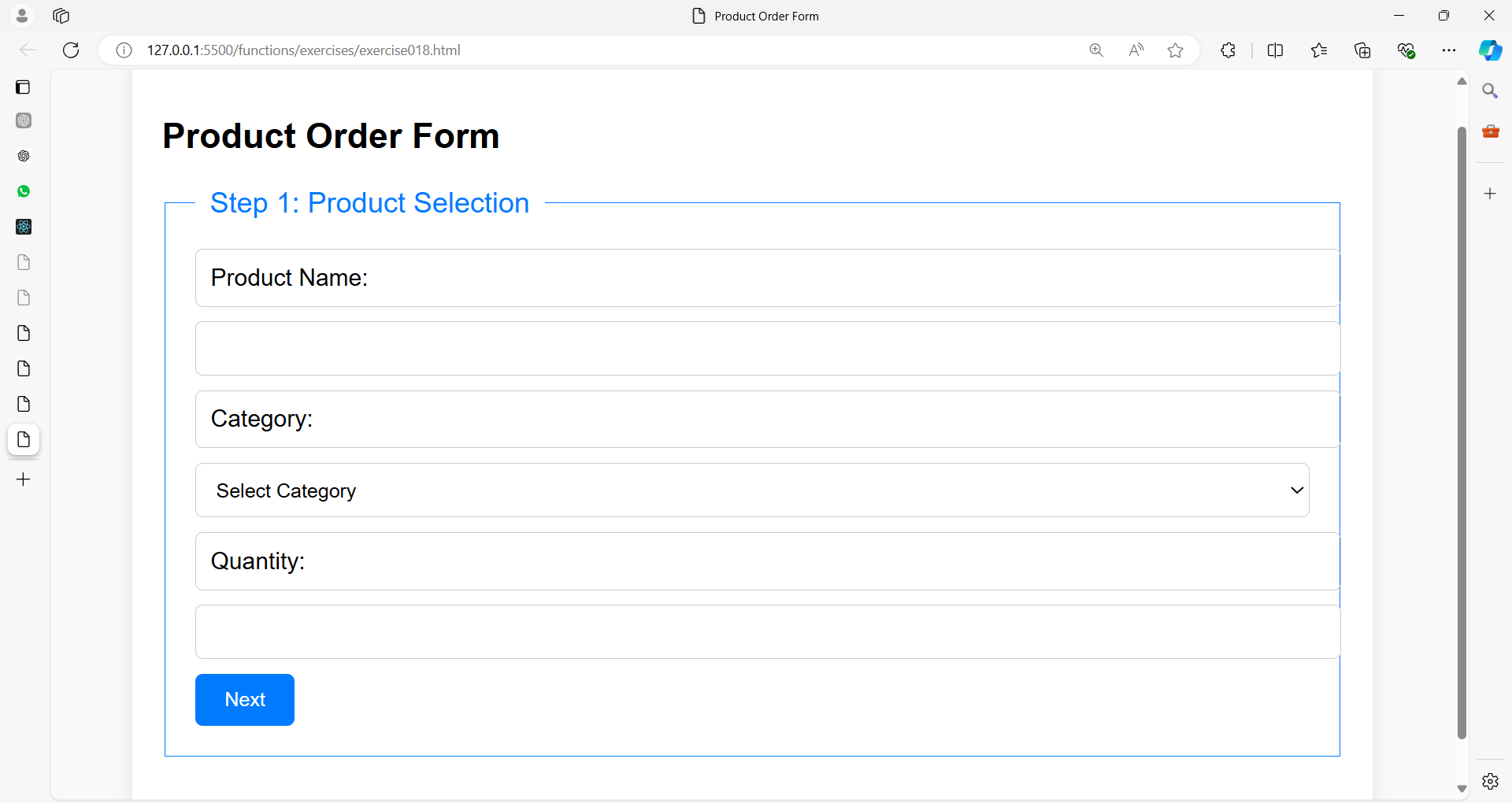
CVV: 123

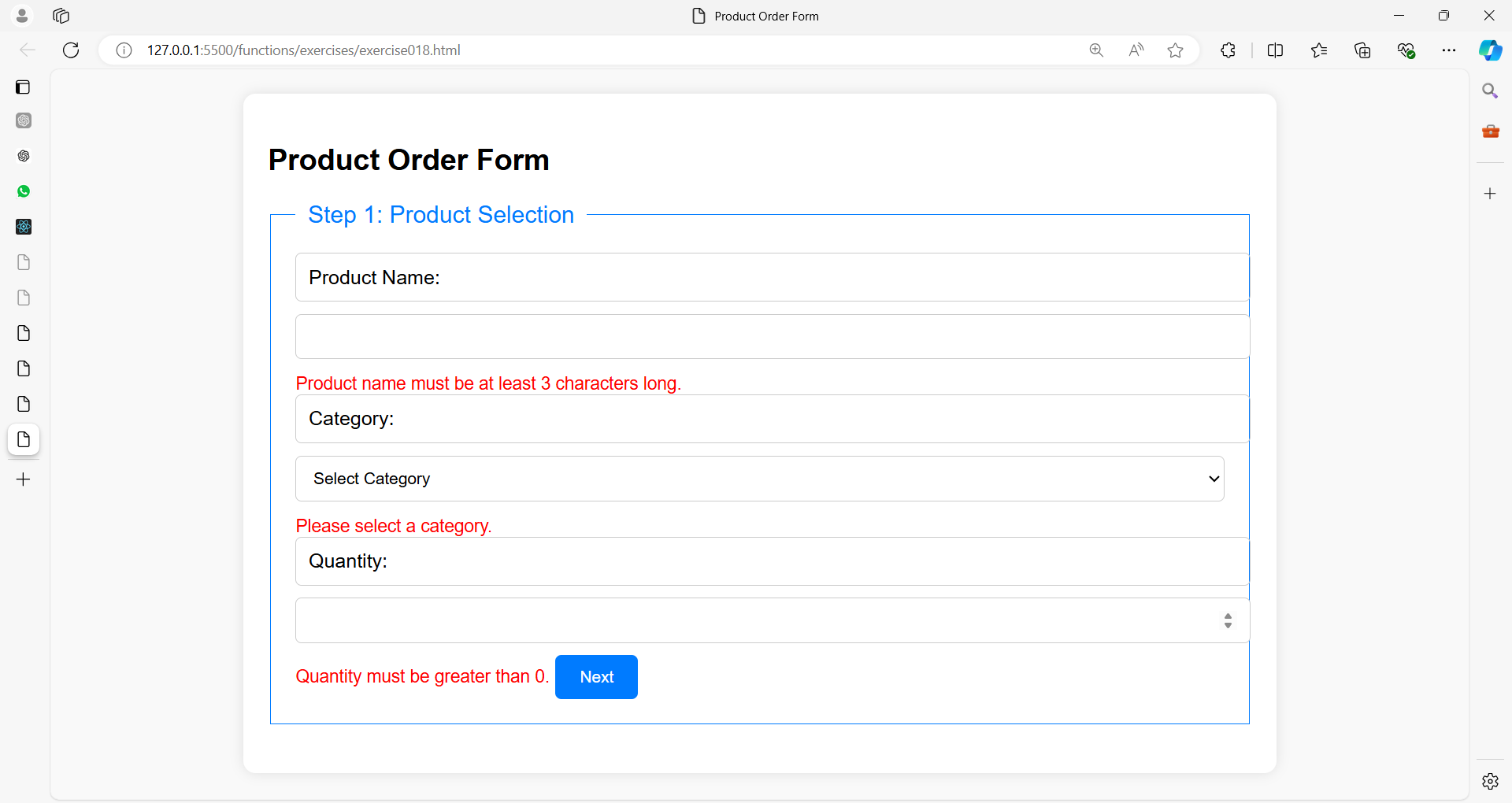
**Deliverables:**

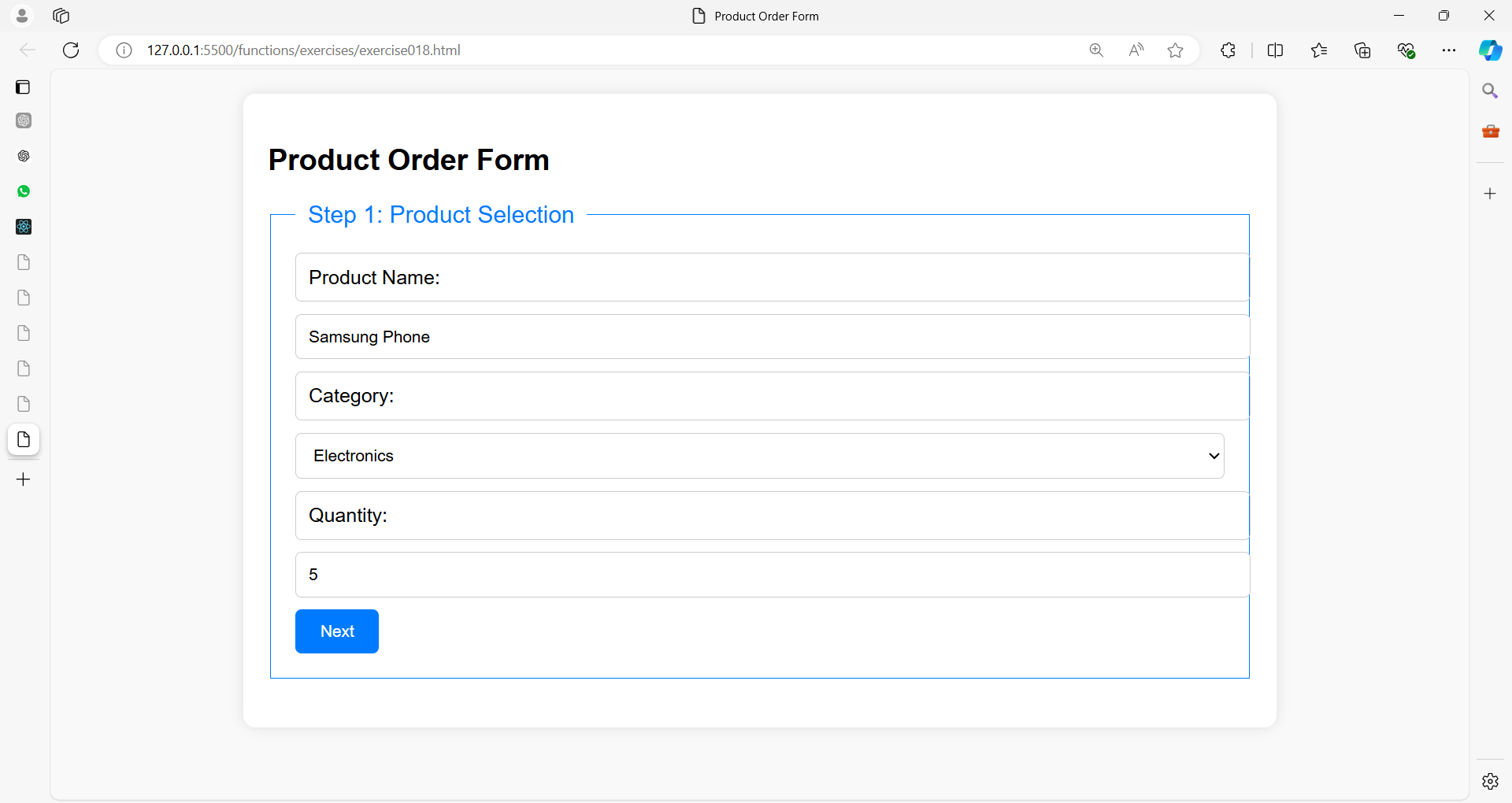
1. HTML code for the multi-step product order form.
2. CSS to style the form and error messages.
3. JavaScript for form validation and step navigation.

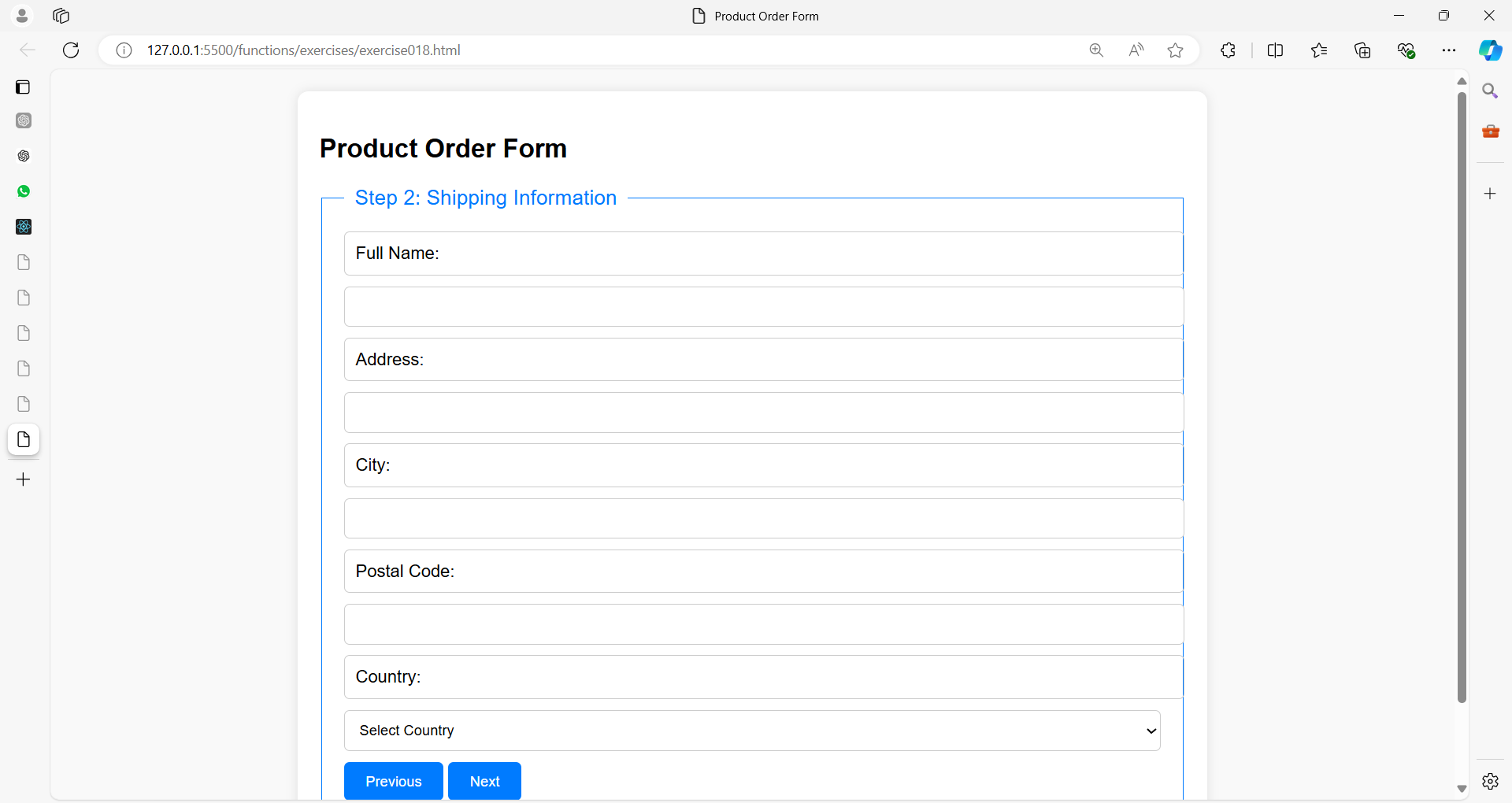
**Bonus Challenge:**

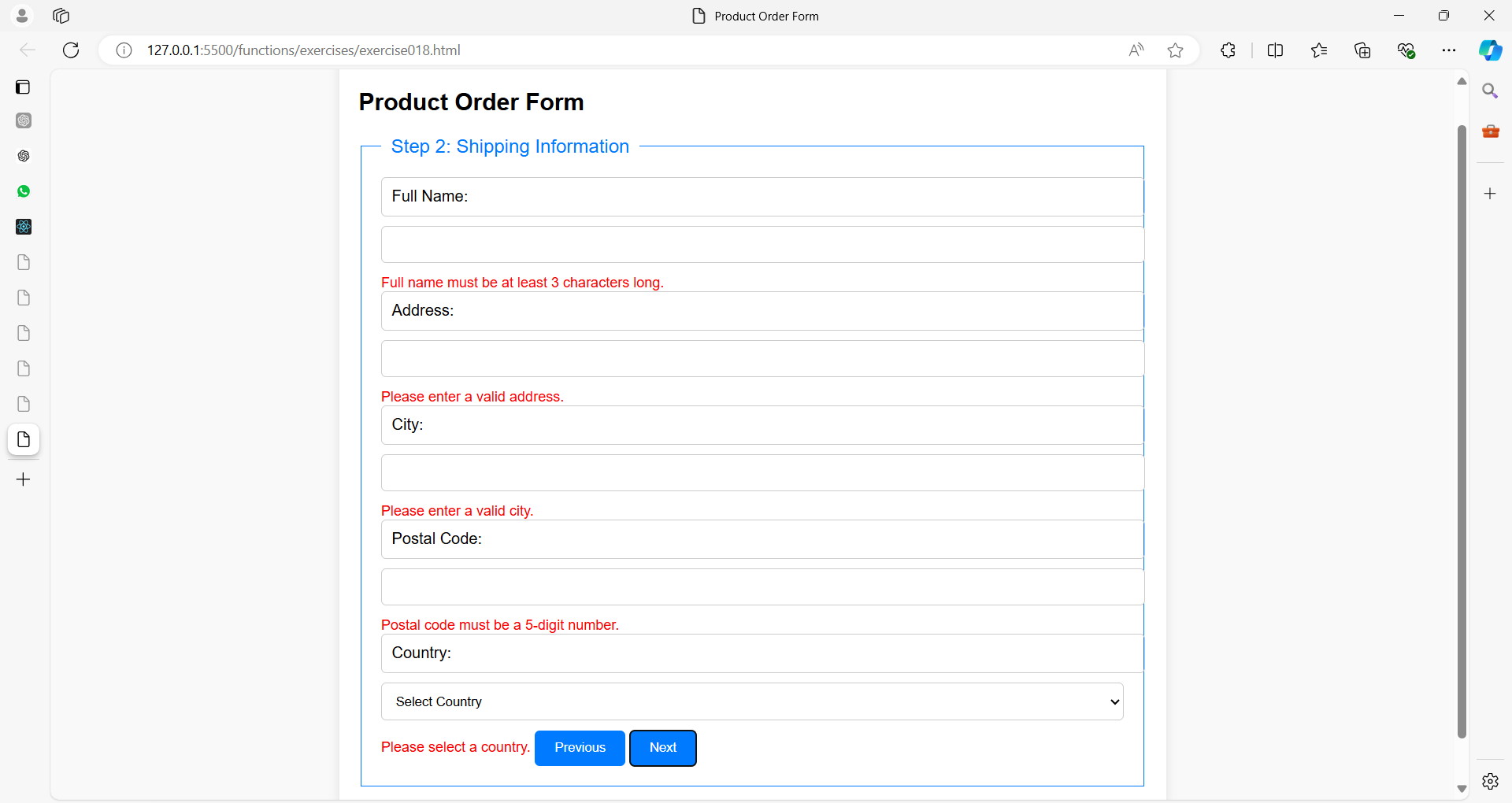
* Allow users to see a summary of their order (product and shipping details) before submitting the payment.
* Include a "Confirm Order" button after showing the summary.

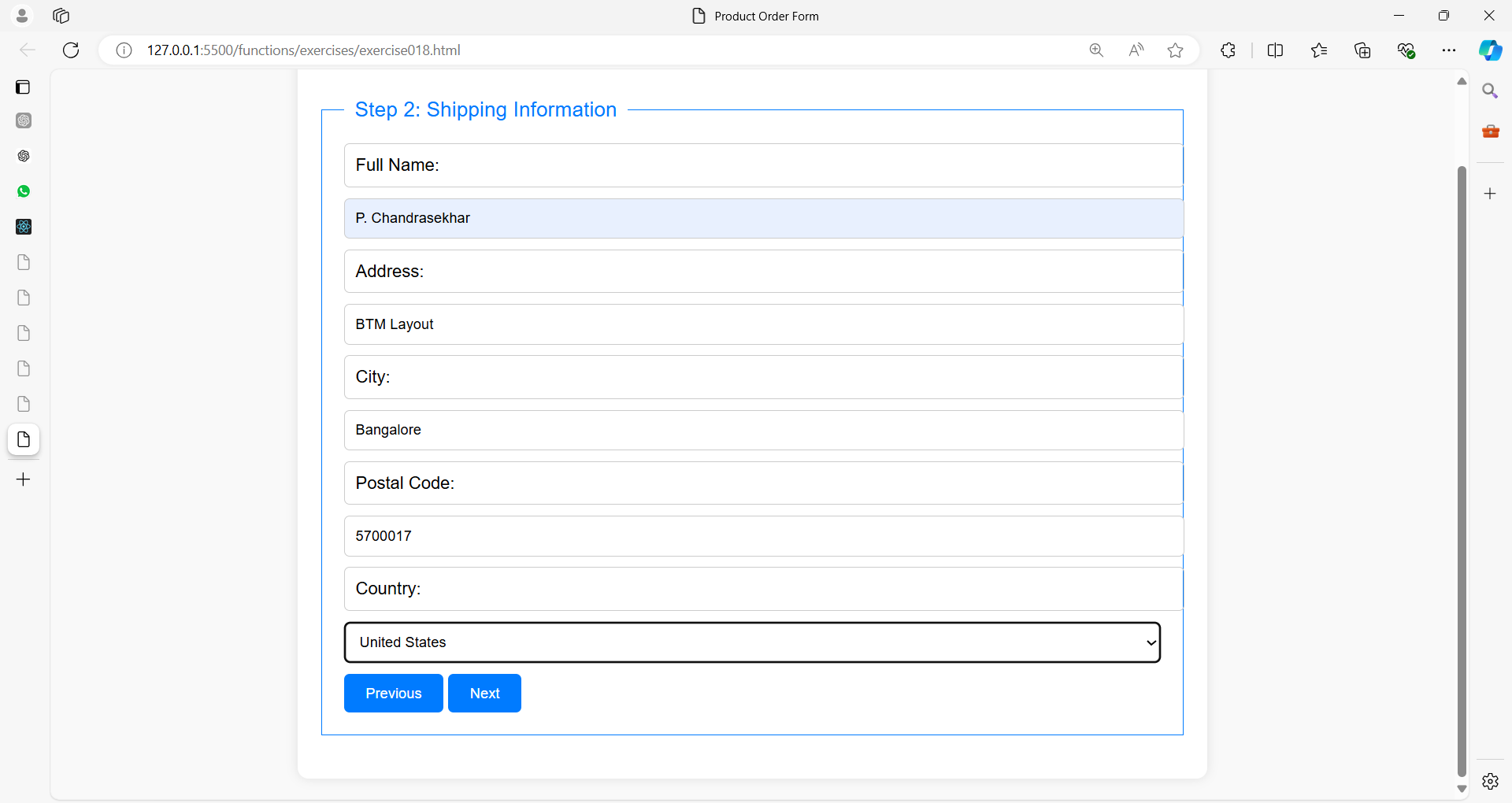


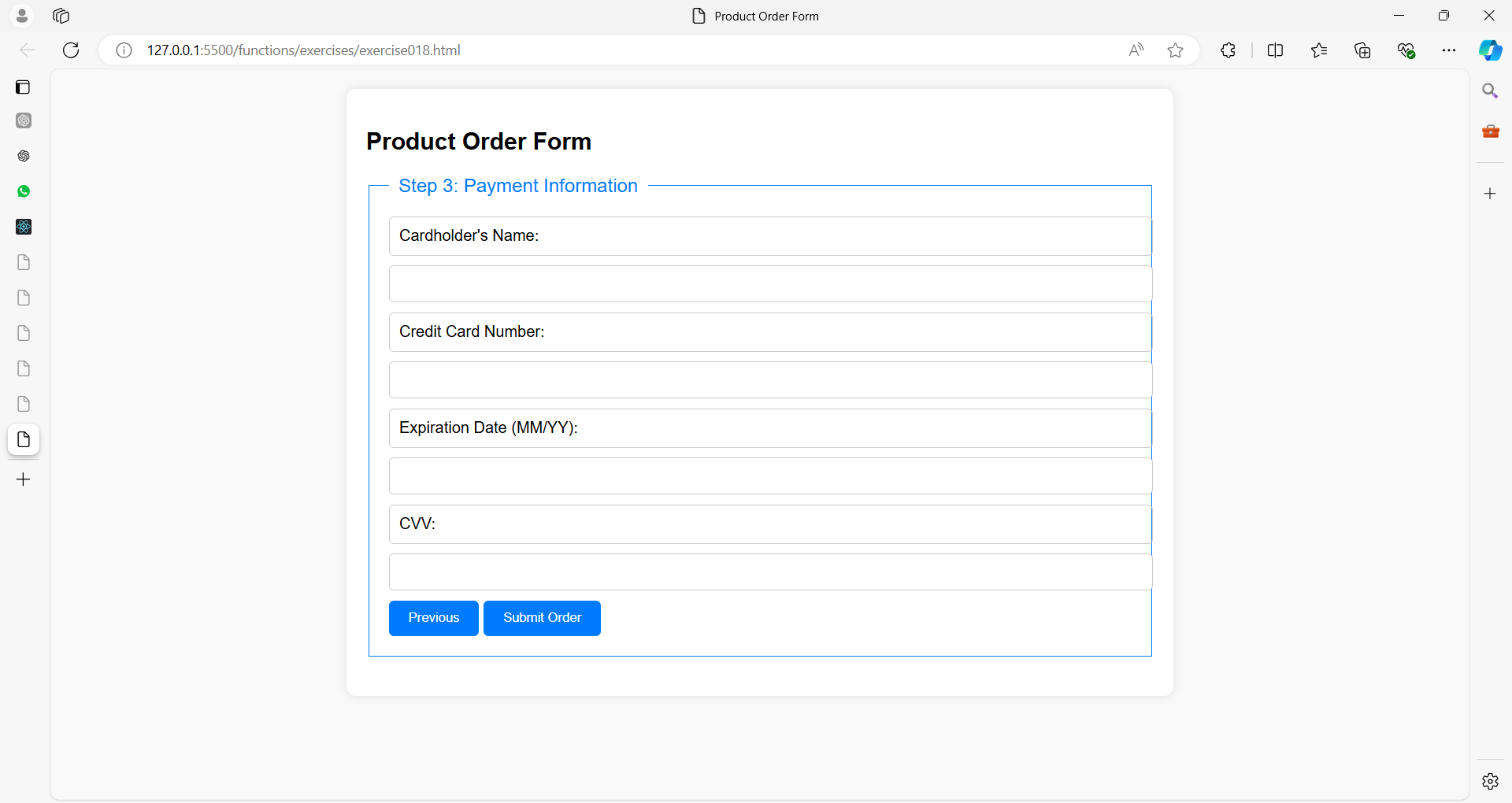


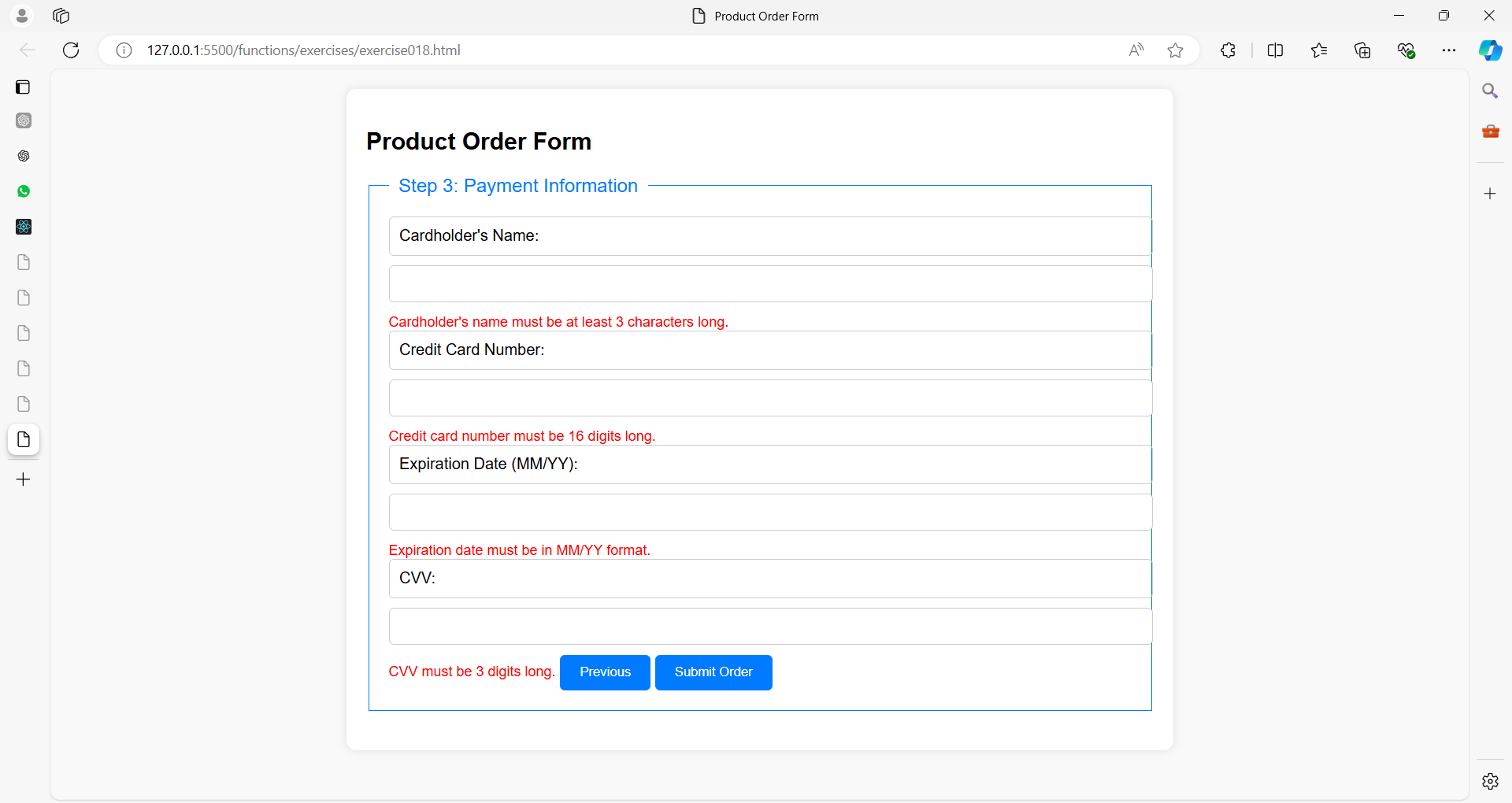


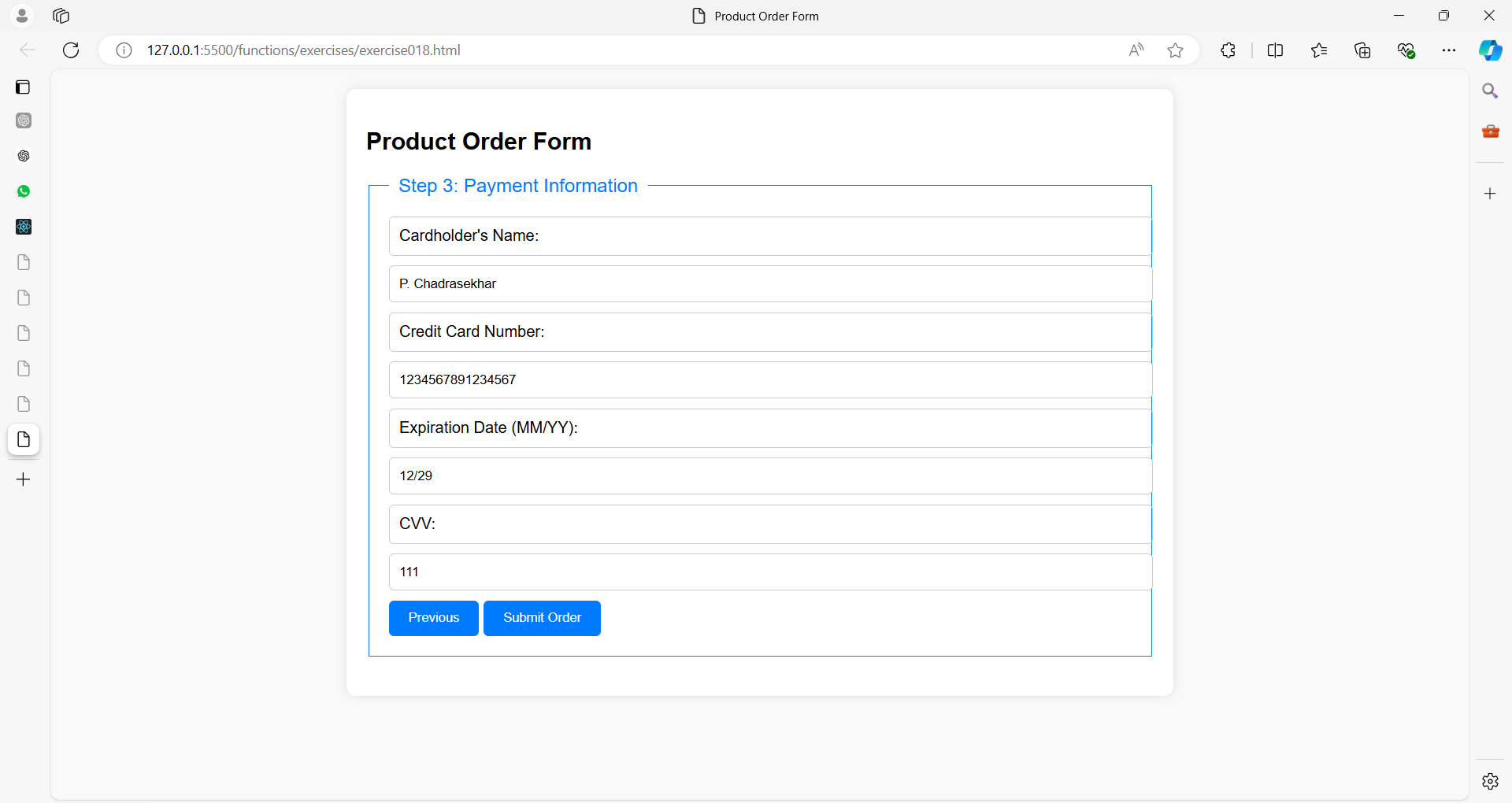












**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Product Order Form</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 20px;

}

.container {

max-width: 800px;

margin: auto;

background-color: white;

padding: 20px;

border-radius: 10px;

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

}

fieldset {

border: 1px solid #007bff;

padding: 20px;

margin-bottom: 20px;

}

legend {

font-size: 1.2em;

color: #007bff;

padding: 0 10px;

}

label, input, select {

display: block;

width: 100%;

margin-bottom: 10px;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

.error {

color: red;

font-size: 0.9em;

margin-bottom: 10px;

}

button {

padding: 10px 20px;

background-color: #007bff;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

.hidden {

display: none;

}

</style>

</head>

<body>

<div class="container">

<h2>Product Order Form</h2>

<!-- Form 1: Product Selection -->

<form id="productForm">

<fieldset>

<legend>Step 1: Product Selection</legend>

<label for="productName">Product Name:</label>

<input type="text" id="productName" name="productName">

<span id="productError" class="error"></span>

<label for="category">Category:</label>

<select id="category" name="category">

<option value="">Select Category</option>

<option value="Electronics">Electronics</option>

<option value="Furniture">Furniture</option>

<option value="Clothing">Clothing</option>

</select>

<span id="categoryError" class="error"></span>

<label for="quantity">Quantity:</label>

<input type="number" id="quantity" name="quantity" min="1">

<span id="quantityError" class="error"></span>

<button type="button" onclick="nextStep('productForm', 'shippingForm')">Next</button>

</fieldset>

</form>

<!-- Form 2: Shipping Information -->

<form id="shippingForm" class="hidden">

<fieldset>

<legend>Step 2: Shipping Information</legend>

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName">

<span id="fullNameError" class="error"></span>

<label for="address">Address:</label>

<input type="text" id="address" name="address">

<span id="addressError" class="error"></span>

<label for="city">City:</label>

<input type="text" id="city" name="city">

<span id="cityError" class="error"></span>

<label for="postalCode">Postal Code:</label>

<input type="text" id="postalCode" name="postalCode">

<span id="postalCodeError" class="error"></span>

<label for="country">Country:</label>

<select id="country" name="country">

<option value="">Select Country</option>

<option value="United States">United States</option>

<option value="Canada">Canada</option>

<option value="United Kingdom">United Kingdom</option>

<option value="Australia">Australia</option>

<option value="Germany">Germany</option>

</select>

<span id="countryError" class="error"></span>

<button type="button" onclick="prevStep('shippingForm', 'productForm')">Previous</button>

<button type="button" onclick="nextStep('shippingForm', 'paymentForm')">Next</button>

</fieldset>

</form>

<!-- Form 3: Payment Information -->

<form id="paymentForm" class="hidden">

<fieldset>

<legend>Step 3: Payment Information</legend>

<label for="cardName">Cardholder's Name:</label>

<input type="text" id="cardName" name="cardName">

<span id="cardNameError" class="error"></span>

<label for="cardNumber">Credit Card Number:</label>

<input type="text" id="cardNumber" name="cardNumber">

<span id="cardNumberError" class="error"></span>

<label for="expirationDate">Expiration Date (MM/YY):</label>

<input type="text" id="expirationDate" name="expirationDate">

<span id="expirationDateError" class="error"></span>

<label for="cvv">CVV:</label>

<input type="text" id="cvv" name="cvv">

<span id="cvvError" class="error"></span>

<button type="button" onclick="prevStep('paymentForm', 'shippingForm')">Previous</button>

<button type="button" onclick="validateAllForms()">Submit Order</button>

</fieldset>

</form>

</div>

<script>

// Move to the next form step

function nextStep(currentForm, nextForm) {

if (validateForm(currentForm)) {

document.getElementById(currentForm).classList.add('hidden');

document.getElementById(nextForm).classList.remove('hidden');

}

}

// Move to the previous form step

function prevStep(currentForm, prevForm) {

document.getElementById(currentForm).classList.add('hidden');

document.getElementById(prevForm).classList.remove('hidden');

}

// Validate current form

function validateForm(formId) {

let valid = true;

// Product Selection validation

if (formId === 'productForm') {

const productName = document.getElementById('productName').value;

const category = document.getElementById('category').value;

const quantity = document.getElementById('quantity').value;

document.getElementById('productError').innerText = '';

document.getElementById('categoryError').innerText = '';

document.getElementById('quantityError').innerText = '';

if (productName.length < 3) {

document.getElementById('productError').innerText = 'Product name must be at least 3 characters long.';

valid = false;

}

if (category === '') {

document.getElementById('categoryError').innerText = 'Please select a category.';

valid = false;

}

if (quantity <= 0) {

document.getElementById('quantityError').innerText = 'Quantity must be greater than 0.';

valid = false;

}

}

// Shipping Information validation

if (formId === 'shippingForm') {

const fullName = document.getElementById('fullName').value;

const address = document.getElementById('address').value;

const city = document.getElementById('city').value;

const postalCode = document.getElementById('postalCode').value;

const country = document.getElementById('country').value;

document.getElementById('fullNameError').innerText = '';

document.getElementById('addressError').innerText = '';

document.getElementById('cityError').innerText = '';

document.getElementById('postalCodeError').innerText = '';

document.getElementById('countryError').innerText = '';

if (fullName.length < 3) {

document.getElementById('fullNameError').innerText = 'Full name must be at least 3 characters long.';

valid = false;

}

if (address.length < 5) {

document.getElementById('addressError').innerText = 'Please enter a valid address.';

valid = false;

}

if (city.length < 3) {

document.getElementById('cityError').innerText = 'Please enter a valid city.';

valid = false;

}

if (!/^\d{5}$/.test(postalCode)) {

document.getElementById('postalCodeError').innerText = 'Postal code must be a 5-digit number.';

valid = false;

}

if (country === '') {

document.getElementById('countryError').innerText = 'Please select a country.';

valid = false;

}

}

// Payment Information validation

if (formId === 'paymentForm') {

const cardName = document.getElementById('cardName').value;

const cardNumber = document.getElementById('cardNumber').value;

const expirationDate = document.getElementById('expirationDate').value;

const cvv = document.getElementById('cvv').value;

document.getElementById('cardNameError').innerText = '';

document.getElementById('cardNumberError').innerText = '';

document.getElementById('expirationDateError').innerText = '';

document.getElementById('cvvError').innerText = '';

if (cardName.length < 3) {

document.getElementById('cardNameError').innerText = 'Cardholder\'s name must be at least 3 characters long.';

valid = false;

}

if (!/^\d{16}$/.test(cardNumber)) {

document.getElementById('cardNumberError').innerText = 'Credit card number must be 16 digits long.';

valid = false;

}

if (!/^(0[1-9]|1[0-2])\/\d{2}$/.test(expirationDate)) {

document.getElementById('expirationDateError').innerText = 'Expiration date must be in MM/YY format.';

valid = false;

}

if (!/^\d{3}$/.test(cvv)) {

document.getElementById('cvvError').innerText = 'CVV must be 3 digits long.';

valid = false;

}

}

return valid;

}

// Validate all forms and submit

function validateAllForms() {

if (validateForm('productForm') && validateForm('shippingForm') && validateForm('paymentForm')) {

alert('Order submitted successfully!');

// Here you could also send data to a server

}

}

</script>

</body>

</html>