

## LAB 3 – WEB DEVELOPMENT APPLICATION

Huỳnh Tuấn Anh – ITITIU23003

### Task 1.1: Form validation

Source code:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Form Validator</title>
  <style>
    .form-container {
      max-width: 400px;
      margin: 50px auto;
      padding: 30px;
      border: 1px solid #ddd;
      border-radius: 8px;
      font-family: Arial, sans-serif;
    }

    .form-group {
      margin-bottom: 20px;
    }

    label {
      display: block;
      margin-bottom: 5px;
      font-weight: bold;
    }

    input {
      width: 100%;
      padding: 10px;
      border: 2px solid #ddd;
      border-radius: 4px;
      font-size: 14px;
    }

    input.valid {
      border-color: #28a745;
    }

    input.invalid {
      border-color: #dc3545;
    }

    .error-message {
      color: #dc3545;
      font-size: 12px;
      margin-top: 5px;
    }
  </style>
</head>
```

```

        display: none;
    }

    .error-message.show {
        display: block;
    }

    button[type="submit"] {
        width: 100%;
        padding: 12px;
        background: #007bff;
        color: white;
        border: none;
        border-radius: 4px;
        font-size: 16px;
        cursor: pointer;
    }

    button[type="submit"]:disabled {
        background: #ccc;
        cursor: not-allowed;
    }
}
</style>
</head>

<body>
    <div class="form-container">
        <h2>Sign Up Form</h2>
        <form id="signupForm">
            <div class="form-group">
                <label for="username">Username</label>
                <input type="text" id="username" placeholder="Enter username">
                <div class="error-message" id="usernameError"></div>
            </div>

            <!-- TODO: Add email field -->
            <div class="form-group">
                <label for="email">Email</label>
                <input type="email" id="email" placeholder="Enter email">
                <div class="error-message" id="emailError"></div>
            </div>

            <!-- TODO: Add password field -->
            <div class="form-group">
                <label for="password">Password</label>
                <input type="password" id="password" placeholder="Enter password">
                <div class="error-message" id="passwordError"></div>
            </div>

            <!-- TODO: Add confirm password field -->
            <div class="form-group">

```

```

        <label for="confirmPassword">Confirm Password</label>
        <input type="password" id="confirmPassword" placeholder="Confirm
password">

        <div class="error-message" id="confirmPasswordError"></div>
    </div>

    <button type="submit" id="submitBtn" disabled>Sign Up</button>
</form>
</div>

<script>
    // TODO: Implement validation functions
    const usernameField = document.getElementById('username');
    const emailField = document.getElementById('email');
    const passwordField = document.getElementById('password');
    const confirmPasswordField = document.getElementById('confirmPassword');
    const submitBtn = document.getElementById('submitBtn');

    function validateUsername(username) {
        // TODO: Check length (4-20) and alphanumeric
        const usernameRegex = /^[a-zA-Z0-9]{4,20}$/;
        return usernameRegex.test(username);
    }

    function validateEmail(email) {
        // TODO: Use regex for email validation
        const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
        return emailRegex.test(email);
    }

    function validatePassword(password) {
        // TODO: Check length >= 8, has uppercase, has number
        const passwordRegex = /^(?=.*[A-Z])(?=.*\d)[A-Za-z\d]{8,}$/;
        return passwordRegex.test(password);
    }

    function validatePasswordMatch(pass1, pass2) {
        // TODO: Display error message
        return pass1 === pass2 && pass1 !== '';
    }

    function showError(fieldId, message) {
        // TODO: Display error message
        const input = document.getElementById(fieldId);
        const errorDiv = document.getElementById(fieldId + 'Error');
        input.classList.remove('valid');
        input.classList.add('invalid');
        errorDiv.textContent = message;
        errorDiv.classList.add('show');
    }

```

```

function clearError(fieldId) {
    // TODO: Clear error message
    const input = document.getElementById(fieldId);
    const errorDiv = document.getElementById(fieldId + 'Error');
    input.classList.remove('invalid');
    input.classList.add('valid');
    errorDiv.textContent = '';
    errorDiv.classList.remove('show');
}

function validateForm() {
    // TODO: Validate all fields and enable/disable submit
    const username = usernameField.value.trim();
    const email = emailField.value.trim();
    const password = passwordField.value.trim();
    const confirmPassword = confirmPasswordField.value.trim();
    let isValid = true;

    if (!validateUsername(username)) {
        showError('username', 'Username must be 4-20 characters (letters or
numbers only)');
        isValid = false;
    } else {
        clearError('username');
    }

    if (!validateEmail(email)) {
        showError('email', 'Please enter a valid email address');
        isValid = false;
    } else {
        clearError('email');
    }

    if (!validatePassword(password)) {
        showError('password', 'Password must be greater than or equal 8
chars, include uppercase & number');
        isValid = false;
    } else {
        clearError('password');
    }

    if (!validatePasswordMatch(password, confirmPassword)) {
        showError('confirmPassword', 'Passwords do not match');
        isValid = false;
    } else {
        clearError('confirmPassword');
    }

    submitBtn.disabled = !isValid;
}

// TODO: Add event listeners for real-time validation

```

```

usernameField.addEventListener('input', validateForm);
emailField.addEventListener('input', validateForm);
passwordField.addEventListener('input', validateForm);
confirmPasswordField.addEventListener('input', validateForm);

document.getElementById('signupForm').addEventListener('submit', function
(e) {
    e.preventDefault();
    // TODO: Handle form submission
    if (!submitBtn.disabled) {
        alert('Form submitted successfully!');
        e.target.reset();
        submitBtn.disabled = true;
        usernameField.classList.remove('valid', 'invalid');
        emailField.classList.remove('valid', 'invalid');
        passwordField.classList.remove('valid', 'invalid');
        confirmPasswordField.classList.remove('valid', 'invalid');
    }
});
</script>
</body>
</html>
Result:

```

**Sign Up Form**

**Username**

Username must be 4-20 characters (letters or numbers only)

**Email**

Please enter a valid email address

**Password**

Password must be greater than or equal 8 chars, include uppercase & number

**Confirm Password**

Passwords do not match

Sign Up

There are message errors from input field and the sign up button is disabled

### Email

Please enter a valid email address

There is a message error for incorrect type of email

### Password

### Confirm Password

Passwords do not match

Message for incorrect type of password

## Sign Up Form

**Username**

**Email**

**Password**

**Confirm Password**

Sign Up

The sign up button is enabled if and only if all field are in the correct form

127.0.0.1:5500 cho biết

Form submitted successfully!



The alert window pop up to announce that submit successfully

Checklist:

- All fields validate on input

```

function validateUsername(username) {
    // TODO: Check length (4-20) and alphanumeric
    const usernameRegex = /^[a-zA-Z0-9]{4,20}$/;
    return usernameRegex.test(username);
}

function validateEmail(email) {
    // TODO: Use regex for email validation
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return emailRegex.test(email);
}

function validatePassword(password) {
    // TODO: Check length >= 8, has uppercase, has number
    const passwordRegex = /^(?=.*[A-Z])(?=.*\d)[A-Za-z\d]{8,}$/;
    return passwordRegex.test(password);
}

function validatePasswordMatch(pass1, pass2) {
    // TODO: Display error message
    return pass1 === pass2 && pass1 !== '';
}

```

- Error messages appear/disappear correctly

```

if (!validateUsername(username)) {
    showError('username', 'Username must be 4-20 characters (letters or numbers only)');
    isValid = false;
} else {
    clearError('username');
}

if (!validateEmail(email)) {
    showError('email', 'Please enter a valid email address');
    isValid = false;
} else {
    clearError('email');
}

if (!validatePassword(password)) {
    showError('password', 'Password must be greater than or equal 8 chars, include uppercase & number');
    isValid = false;
} else {
    clearError('password');
}

if (!validatePasswordMatch(password, confirmPassword)) {
    showError('confirmPassword', 'Passwords do not match');
    isValid = false;
} else {
    clearError('confirmPassword');
}

```

- Border colors change (green/red): See in the result
- Submit button enables only when all valid: See in the result
- Form submission works: There is an alert window.

## Task 1.2: Shopping cart

Source code:

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Shopping Cart</title>
    <style>

```

```

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: Arial, sans-serif;
  background: #f5f5f5;
}

.header {
  background: #333;
  color: white;
  padding: 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.cart-icon {
  position: relative;
  cursor: pointer;
}

.cart-count {
  position: absolute;
  top: -10px;
  right: -10px;
  background: red;
  color: white;
  border-radius: 50%;
  width: 24px;
  height: 24px;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 12px;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}

.products-grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
  gap: 20px;
  margin-bottom: 40px;
}

```



```

}

.product-card {
  background: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.product-image {
  width: 100%;
  height: 200px;
  background: #ddd;
  border-radius: 4px;
  margin-bottom: 10px;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 80px;
}

.product-name {
  font-size: 18px;
  font-weight: bold;
  margin-bottom: 10px;
}

.product-price {
  color: #28a745;
  font-size: 20px;
  margin-bottom: 15px;
}

.add-to-cart-btn {
  width: 100%;
  padding: 10px;
  background: #007bff;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
}

.add-to-cart-btn:hover {
  background: #0056b3;
}

.cart-section {
  background: white;
  padding: 20px;
}

```

```

        border-radius: 8px;
    }

    .cart-item {
        display: flex;
        justify-content: space-between;
        align-items: center;
        padding: 15px;
        border-bottom: 1px solid #eee;
    }

    .quantity-controls {
        display: flex;
        gap: 10px;
        align-items: center;
    }

    .quantity-controls button {
        width: 30px;
        height: 30px;
        border: 1px solid #ddd;
        background: white;
        cursor: pointer;
        border-radius: 4px;
    }

    .cart-total {
        text-align: right;
        font-size: 24px;
        font-weight: bold;
        padding: 20px;
        border-top: 2px solid #333;
    }
</style>
</head>

<body>
    <div class="header">
        <h1>My Shop</h1>
        <div class="cart-icon" onclick="toggleCart()">
            🛒 Cart
            <span class="cart-count" id="cartCount">0</span>
        </div>
    </div>

    <div class="container">
        <h2>Products</h2>
        <div class="products-grid" id="productsGrid"></div>

        <div class="cart-section" id="cartSection" style="display: none;">
            <h2>Shopping Cart</h2>

```

```

    <div id="cartItems"></div>
    <div class="cart-total">
      Total: $<span id="cartTotal">0.00</span>
    </div>
  </div>
</div>

<script>
  // Product data
  const products = [
    { id: 1, name: 'Laptop', price: 999.99, image: '💻' },
    { id: 2, name: 'Smartphone', price: 699.99, image: '📱' },
    { id: 3, name: 'Headphones', price: 199.99, image: '🎧' },
    { id: 4, name: 'Smartwatch', price: 299.99, image: '⌚' }
  ];

  // Cart array
  let cart = [];

  // TODO: Implement functions

  function addToCart(productId) {
    // TODO: Add product to cart or increase quantity
    let item = cart.find(x => x.id === productId);

    if (item) {
      item.quantity++;
    } else {
      let p = products.find(x => x.id === productId);
      cart.push({ id: p.id, name: p.name, price: p.price, image: p.image,
quantity: 1 });
    }
    renderCart();
  }

  function removeFromCart(itemId) {
    // TODO: Remove item from cart
    cart = cart.filter(item => item.id !== itemId);
    renderCart();
  }

  function updateQuantity(productId, change) {
    // TODO: Update item quantity (change is +1 or -1)
    let item = cart.find(i => i.id === productId);
    if (item) {
      item.quantity += change;
      if (item.quantity <= 0) {
        removeFromCart(productId);
      }
    }
    renderCart();
  }

```

```

    }

    function calculateTotal() {
        // TODO: Calculate total price
        let total = 0;
        for (let item of cart) {
            total += item.price * item.quantity;
        }
        return total;
    }

    function renderProducts() {
        // TODO: Display products
        const productsGrid = document.getElementById('productsGrid');
        productsGrid.innerHTML = '';

        for (let product of products) {
            productsGrid.innerHTML += `
                <div class="product-card">
                    <div class="product-image">${product.image}</div>
                    <div class="product-name">${product.name}</div>
                    <div class="product-
price">${product.price.toFixed(2)}</div>
                    <button class="add-to-cart-btn"
onclick="addToCart(${product.id})">Add to Cart</button>
                </div>
            `;
        }
    }

    function renderCart() {
        // TODO: Display cart items
        const cartItems = document.getElementById('cartItems');
        const cartCount = document.getElementById('cartCount');
        const cartTotal = document.getElementById('cartTotal');

        cartItems.innerHTML = '';

        if (cart.length === 0) {
            cartItems.innerHTML = '<p>Your cart is empty.</p>';
        } else {
            for (let item of cart) {
                cartItems.innerHTML += `
                    <div class="cart-item">
                        <div>${item.name} - ${item.price.toFixed(2)}</div>
                        <div class="quantity-controls">
                            <button onclick="updateQuantity(${item.id}, -1)">-
</button>

                            <span>${item.quantity}</span>
                            <button onclick="updateQuantity(${item.id},
1)">+</button>
                `;
            }
        }
    }

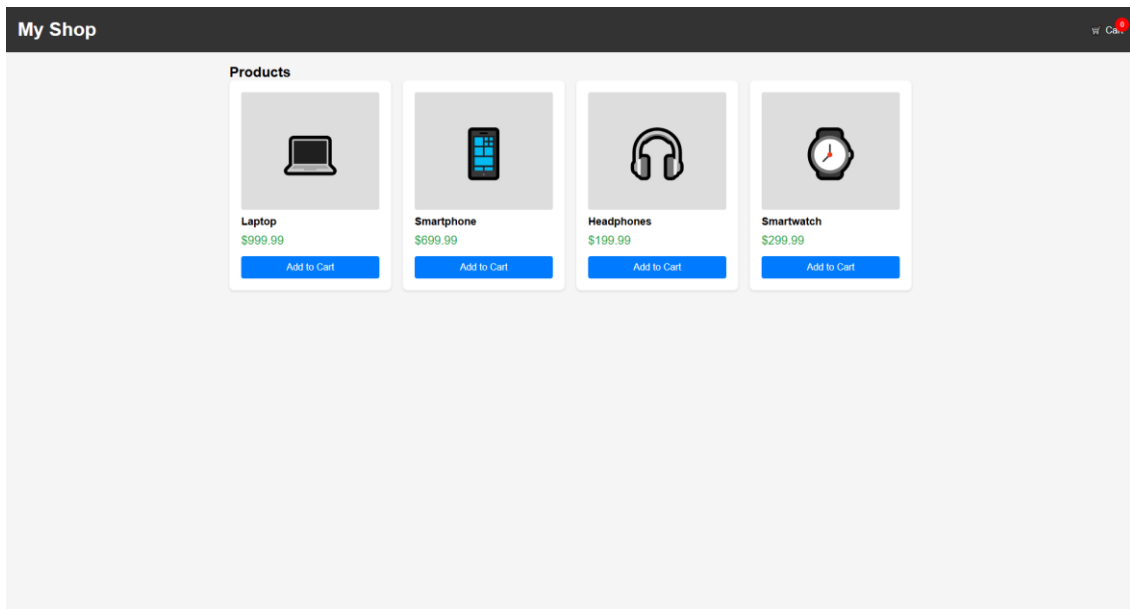
```

```

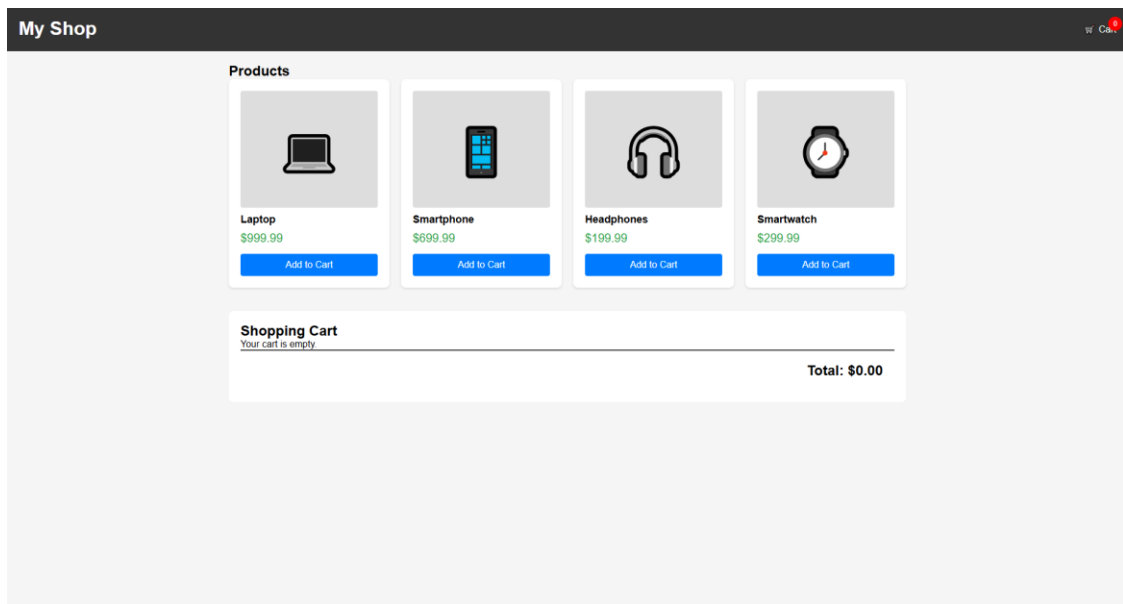
        <button
onclick="removeFromCart(${item.id})">Remove</button>
        </div>
    </div>
    `;
    }
}
cartTotal.textContent = calculateTotal().toFixed(2);
let count = 0;
for (let i of cart) {
    count += i.quantity;
}
cartCount.textContent = count;
}
function toggleCart() {
    // TODO: Show/hide cart section
    const cartSection = document.getElementById('cartSection');
    if (cartSection.style.display === 'none') {
        cartSection.style.display = 'block';
    } else {
        cartSection.style.display = 'none';
    }
}
// Initialize
renderProducts();
</script>
</body>
</html>

```

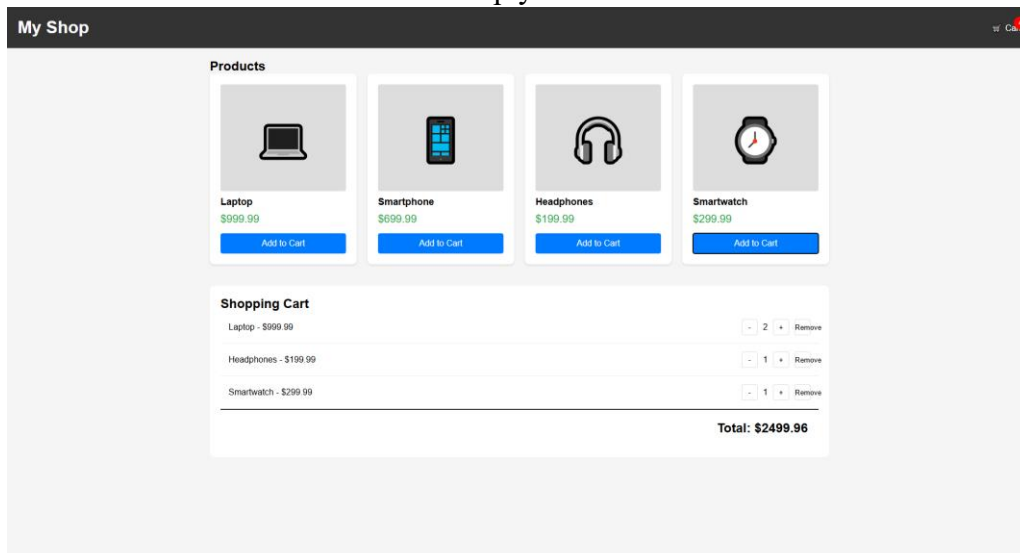
Result:



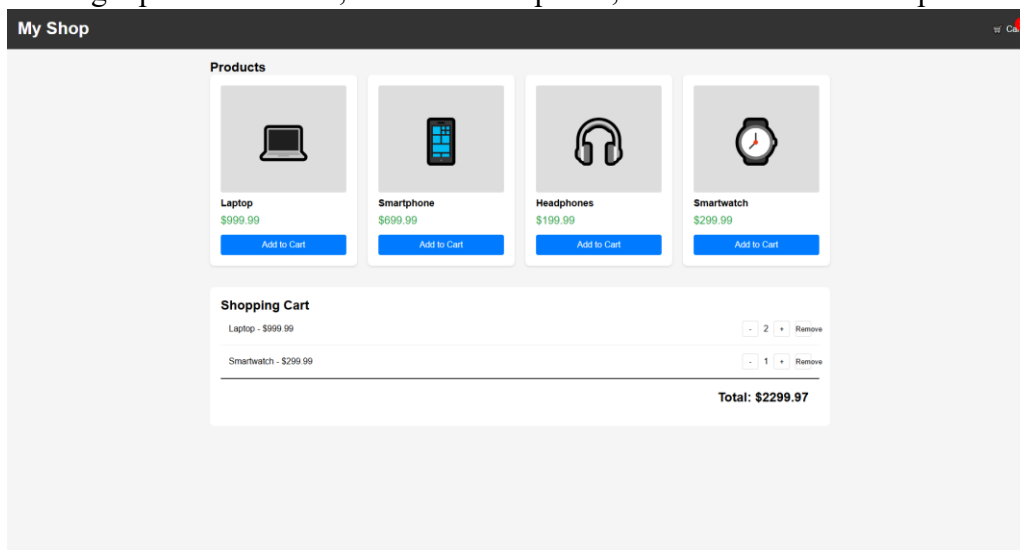
Homepage



Empty cart



Adding 4 products to cart, have the total prices, and see the number of products



Remove headphones product, updating the number of products, and calculating the new total prices  
Checklist:

- Products display correctly:

```
function renderProducts() {
  // TODO: Display products
  const productsGrid = document.getElementById('productsGrid');
  productsGrid.innerHTML = '';
  for (let product of products) {
    productsGrid.innerHTML += `
    <div class="product-card">
      <div class="product-image">${product.image}</div>
      <div class="product-name">${product.name}</div>
      <div class="product-price">${product.price.toFixed(2)}</div>
      <button class="add-to-cart-btn" onclick="addToCart(${product.id})">Add to Cart</button>
    </div>
  `;
  }
}
```

- + There is a loop to go through products in cart. For each product, it has a card to display correctly its description such as image, name, price, and the number of products. After that, that card will be added to the productGrid to display on the webpage.

- "Add to Cart" adds items:

```
function addToCart(productId) {
  // TODO: Add product to cart or increase quantity
  let item = cart.find(x => x.id === productId);

  if (item) {
    item.quantity++;
  } else {
    let p = products.find(x => x.id === productId);
    cart.push({ id: p.id, name: p.name, price: p.price, image: p.image, quantity: 1 });
  }
  renderCart();
}
```

- + The addToCart function will find is the product in the cart or not. If it exists, the function will add one to the quantity when the user click the button “Add to cart”. If it does not exist, the function will insert it into the array with the quantity equal to 1.

- Cart count badge updates:

```
function updateQuantity(productId, change) {
  // TODO: Update item quantity (change is +1 or -1)
  let item = cart.find(i => i.id === productId);
  if (item) {
    item.quantity += change;
    if (item.quantity <= 0) {
      removeFromCart(productId);
    }
  }
  renderCart();
}
```

- + The function will search to find the product in cart. If the item exists, the quantity will add with the change (plus one or minus one). If the item quantity is less than or equal to 0, it will be deleted from the array cart.
- Quantity +/- buttons work
- Remove button works: See images in the result
- Total calculates correctly: See images in the result
- Toggle cart shows/hides