KATHFORD
INTERNATIONAL COLLEGE OF ENGINEERING & MANAGEMENT | AFFILIATED TO TRIBHUVAN UNIVERSITY

Lab Report 2

Introduction for JavaScript to E-commerce

Course: CSC381 - E-Commerce

Submitted by:

Parakram Kharel
Roll No: 24

Kathford International College of Engineering and Management
Affiliated to Tribhuvan University

August 23, 2025

1. Objective

To implement product price display, simple calculations, and interactive UI elements for an e-commerce web application using JavaScript, building upon the HTML/CSS foundation created in Lab 1.

2. Tools and Technologies Used

Technology	Purpose
HTML5	Structure and semantic markup for product displays
CSS3	Styling, animations, and responsive design
JavaScript	Price calculations and interactive functionality
VS Code	Code editor and development environment
Web Browser Developer Tools	Debugging and performance testing

3. Theory / Background

JavaScript is a powerful scripting language that is used to create dynamic and interactive content on Web pages. Unlike HTML and CSS, which define structure and style, JavaScript enables real-time functionality such as responding to user actions, updating content without reloading the page, and performing calculations in the browser.

In this lab, JavaScript is used to enhance the product catalog by implementing interactive features such as quantity selectors, price formatting, discount calculations, and responsive UI behavior. These dynamic elements improve the overall user experience and bring essential e-Commerce functionality to the web interface.

4. Page Layout Design

4.1 Product Card Structure

Each interactive product card now contains the following:

- Product Image with dynamic badges (NEW, SALE, HOT)
- Product Title and Description
- Enhanced Price Section with:
  - Current price display with proper formatting

  - Original price with strike through for sale items
  - Automatic discount percentage calculation
  - Savings amount display
- Interactive Quantity Section featuring:
  - Increment/decrement buttons with validation
  - Number input field (1–10 range)
  - Real-time total calculation display
- Action buttons with visual feedback

4.2 Interactive Features Implementation

- Real-time price calculations based on quantity selection
- Dynamic discount percentage and savings computations
- Visual feedback animations for button interactions
- Responsive quantity selectors with input validation
- Enhanced search functionality including price-based filtering
- Category filtering with smooth transition effects

4.3 JavaScript Functionality Components

- Price formatting functions for proper currency display
- Mathematical calculation functions for discounts, taxes, and shipping
- Event listeners for user interactions and input validation
- DOM manipulation for dynamic content updates
- Animation control through programmatic CSS class manipulation

5. Code Snippets

5.1 HTML Code: Product Card with Interactive Elements

```
<div class='product-card' data-category='electronics'>
    <div class='product-image'>
        <img src='images/headphone.jpg' alt='Wireless Headphones'>
        <div class='product-badge'>NEW</div>
    </div>
    <div class='product-info'>
        <h3>Premium Headphones</h3>
        <p class='product-desc'>
            High-quality wireless headphones with noise cancellation
        </p>
        <div class='price-section'>
            <div class='price-tag' data-original-price='8999'>NPR
            8,999</div>
            <div class='discount-info' style='display: none;'>
                <span class='discount-percent'></span>
                <span class='savings-amount'></span>
            </div>
        </div>
        <div class='quantity-section'>
            <label>Quantity:</label>
            <div class='quantity-selector'>
                <button class='qty-btn minus' onclick='changeQuantity(this,
                -1)'> - </button>
                <input type='number' class='qty-input' value='1' min='1'
                max='10' onchange='updateProductTotal(this)'>
                <button class='qty-btn plus' onclick='changeQuantity(this,
                1)'>+</button>
            </div>
            <div class='product-total'>
                Total: NPR <span class='total-amount'>8,999</span>
            </div>
        </div>
        <div class='product-actions'>
            <button class='select-btn' data-price='8999'>Select
            Item</button>
            <button class='details-btn'>View Details</button>
        </div>
    </div>
</div>
```

5.2 JavaScript Code: Price Formatting Functions

```
function formatPrice (price) {
    return price.toLocaleString('en-NP');
}

function formatCurrency (amount) {
    return `NPR ${formatPrice(amount)}`;
}
```

5.3 JavaScript Code: Discount Calculation

```
function calculateDiscount(originalPrice, salePrice) {
    const discount = originalPrice - salePrice;
    const discountPercent = Math.round((discount / originalPrice) * 100);
    return {
        percent: discountPercent,
        amount: discount,
        savings: formatCurrency(discount)
    };
}
```

5.4 JavaScript Code: Quantity Management

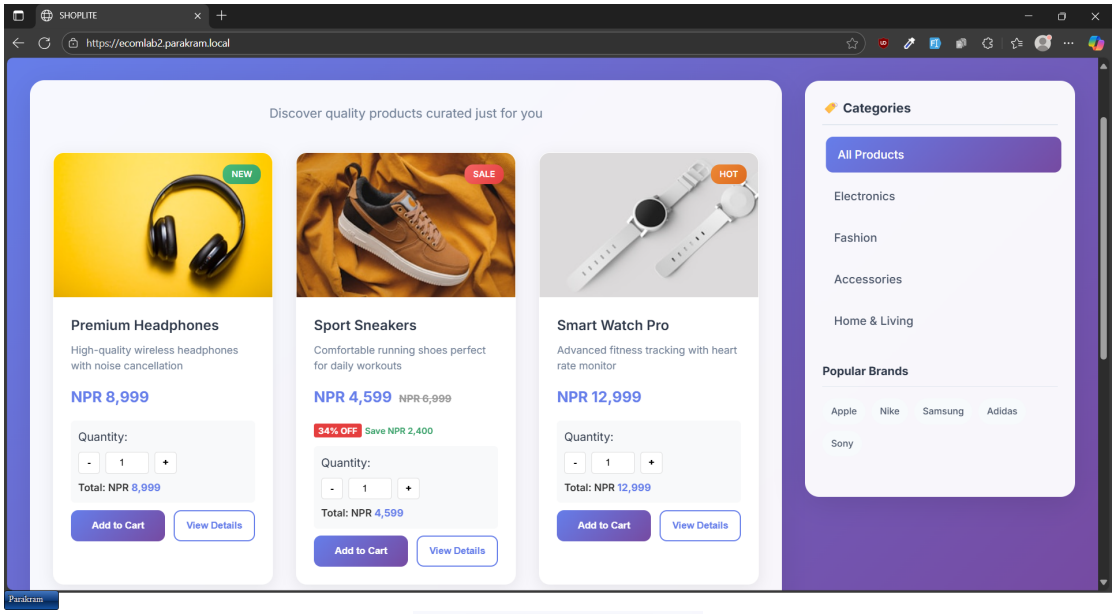
```
function changeQuantity(button, change) {
    const qtyInput = button.parentNode.querySelector('.qty-input');
    const currentValue = parseInt(qtyInput.value);
    const newValue = currentValue + change;

    if (newValue >= 1 && newValue <= 10) {
        qtyInput.value = newValue;
        updateProductTotal(qtyInput);
    }
}

function updateProductTotal(input) {
    const quantity = parseInt(input.value);
    const productCard = input.closest('.product-card');
    const priceTag = productCard.querySelector('.price-tag');
    const originalPrice =
    parseInt(priceTag.getAttribute('data-original-price'));

```

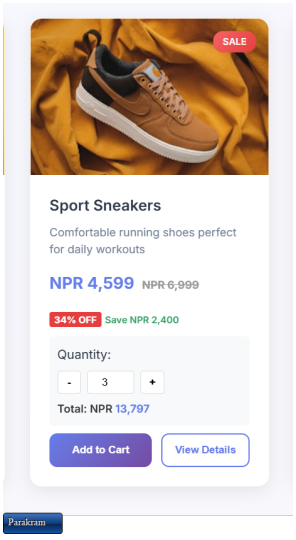
```
    const totalAmount = originalPrice * quantity;
    const totalAmountElement = productCard.querySelector('.total-amount');
    totalAmountElement.textContent = formatPrice(totalAmount);
}
```

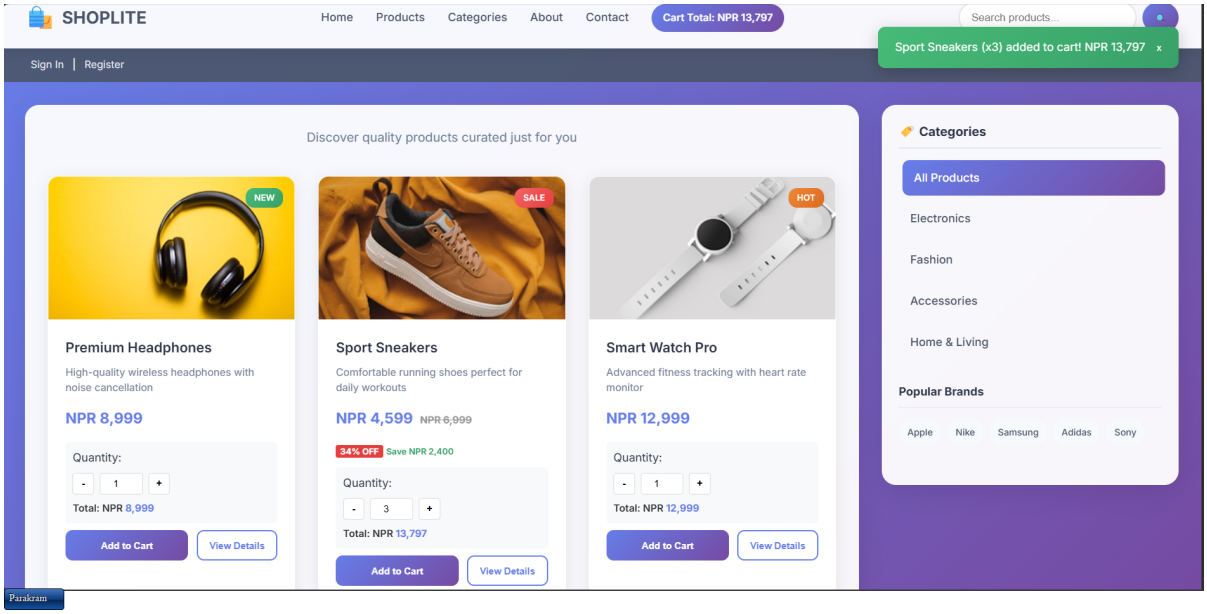
5.5 JavaScript Code: Interactive UI Elements

```
function selectItem(productName, quantity, button) {
    selectedItemsCount++;
    updateItemsDisplay();
    showButtonFeedback(button);

    const quantityText = quantity > 1 ? ` (x${quantity})` : '';
    showNotification(`${productName}${quantityText} selected!`);
}

function showButtonFeedback(button) {
    const originalText = button.textContent;
    button.textContent = '✓ Selected!';
    button.style.background = 'linear-gradient(135deg, #48bb78, #38a169)';

    setTimeout(() => {
        button.textContent = originalText;
        button.style.background = 'linear-gradient(135deg, #667eea,
        #764ba2)';
    }, 1500);
}

```

6. Output / Screenshots







7. Result

The enhanced product catalog features dynamic JavaScript, including real-time price calculations, interactive quantity selectors, automatic discounts, and instant total updates. Prices are in NPR, and visual feedback like animations and color changes improves the user experience. Users can adjust quantities, see totals instantly, receive selection notifications, and filter searches by price.

8. Conclusion

This lab demonstrated how JavaScript integrates with HTML and CSS to build dynamic e-commerce features. Students learned to format prices, create interactive UI components with event handling, and perform real-time calculations for pricing and discounts. It provided a strong foundation in DOM manipulation, event-driven programming, and data validation, showing how static HTML can become responsive, user-driven interfaces.

9. References

- MDN JavaScript Documentation
- The Modern JavaScript Tutorial
- W3Schools JavaScript Tutorial
- MDN NumberFormat API Documentation