

**Refrea, Mar John S.**

**4IT – B**

### **Script.js**

// Shopping Cart System for BrewTong Coffee Shop

```
$(document).ready(function() {
```

```
    // Cart data structure
```

```
    let cart = [];
```

```
    let cartTotal = 0;
```

```
    let orderHistory = [];
```

```
    let favorites = [];
```

```
    let allProducts = [
```

```
        { id: 1, name: "Americano", price: 120, category: "Espresso", image: "images/americano.jpg" },
```

```
        { id: 2, name: "Espresso", price: 80, category: "Espresso", image: "images/espresso.jpg" },
```

```
        { id: 3, name: "Cappuccino", price: 140, category: "Milk-based", image: "images/cappuccino.jpg" },
```

```
        { id: 4, name: "Latte", price: 150, category: "Milk-based", image: "images/latte.jpg" },
```

```
        { id: 5, name: "Flat White", price: 160, category: "Milk-based", image: "images/flatwhite.jpg" },
```

```
        { id: 6, name: "Mocha", price: 170, category: "Specialty", image: "images/mocha.jpg" },
```

```
        { id: 7, name: "Macchiato", price: 130, category: "Espresso", image: "images/machiato.jpg" },
```

```
        { id: 8, name: "Frappe", price: 180, category: "Cold Drinks", image: "images/frappe.jpg" }
```

```
    ];
```

```
// Initialize data from localStorage if available
```

```
function initData() {
```

```
    const savedCart = localStorage.getItem('brewtongCart');
```

```
    const savedOrders = localStorage.getItem('brewtongOrders');
```

```
    const savedFavorites = localStorage.getItem('brewtongFavorites');
```

```
    if (savedCart) {  
        cart = JSON.parse(savedCart);  
    }  
    if (savedOrders) {  
        orderHistory = JSON.parse(savedOrders);  
    }  
    if (savedFavorites) {  
        favorites = JSON.parse(savedFavorites);  
    }  
  
    updateCartDisplay();  
    updateFavoritesDisplay();  
    updateOrderHistoryDisplay();  
}  
  
// Save data to localStorage  
function saveData() {  
    localStorage.setItem('brewtongCart', JSON.stringify(cart));  
    localStorage.setItem('brewtongOrders', JSON.stringify(orderHistory));  
    localStorage.setItem('brewtongFavorites', JSON.stringify(favorites));  
}  
  
// Update cart count display  
function updateCartCount() {  
    const totalItems = cart.reduce((sum, item) => sum + item.quantity, 0);  
    $('#cartCount').text(totalItems);  
  
    // Add animation when items are added  
    if (totalItems > 0) {
```

```
$('#cartCount').addClass('pulse');

setTimeout(() => {

    $('#cartCount').removeClass('pulse');

}, 1000);

}

}
```

// Update cart modal display

```
function updateCartDisplay() {

    const cartItems = $('#cartItems');

    const cartTotalElement = $('#cartTotal');

    if (cart.length === 0) {

        cartItems.html(`

            <div class="empty-cart">

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

                <p>Your cart is empty</p>

                <p>Add some delicious coffee to get started!</p>

            </div>

        `);

        cartTotalElement.text('0');

        $('#checkoutBtn').prop('disabled', true);

    } else {

        let cartHTML = "";

        cartTotal = 0;

        cart.forEach((item, index) => {

            cartTotal += item.price * item.quantity;

            cartHTML += `
```

```

<div class="cart-item" data-index="${index}">
  <div class="cart-item-info">
    <div class="cart-item-name">${item.name}</div>
    <div class="cart-item-price">₹${item.price}</div>
  </div>
  <div class="cart-item-quantity">
    <button class="quantity-btn" onclick="decreaseQuantity(${index})">-</button>
    <span class="quantity-display">${item.quantity}</span>
    <button class="quantity-btn" onclick="increaseQuantity(${index})">+</button>
  </div>
  <button class="remove-item" onclick="removeFromCart(${index})">
    <i class="fas fa-trash"></i>
  </button>
</div>
`;
});

cartItems.html(cartHTML);
cartTotalElement.text(cartTotal.toLocaleString());
$('#checkoutBtn').prop('disabled', false);
}

updateCartCount();
saveData();
}

// Add item to cart
function addToCart(name, price) {
  const existingItem = cart.find(item => item.name === name);

```

```
    if (existingItem) {
        existingItem.quantity += 1;
    } else {
        cart.push({
            name: name,
            price: parseInt(price),
            quantity: 1
        });
    }

    updateCartDisplay();
    showNotification(`${name} added to cart!`);
}
```

```
// Increase quantity
window.increaseQuantity = function(index) {
    cart[index].quantity += 1;
    updateCartDisplay();
};
```

```
// Decrease quantity
window.decreaseQuantity = function(index) {
    if (cart[index].quantity > 1) {
        cart[index].quantity -= 1;
    } else {
        removeFromCart(index);
    }

    updateCartDisplay();
}
```

```
};
```

```
// Remove item from cart
```

```
window.removeFromCart = function(index) {  
    const itemName = cart[index].name;  
    cart.splice(index, 1);  
    updateCartDisplay();  
    showNotification(`${itemName} removed from cart`);  
};
```

```
// Add/Remove from favorites
```

```
window.toggleFavorite = function(productId) {  
    const product = allProducts.find(p => p.id === productId);  
    const existingIndex = favorites.findIndex(f => f.id === productId);  
  
    if (existingIndex >= 0) {  
        favorites.splice(existingIndex, 1);  
        showNotification(`${product.name} removed from favorites`);  
    } else {  
        favorites.push(product);  
        showNotification(`${product.name} added to favorites`);  
    }  
  
    updateFavoritesDisplay();  
    saveData();  
};
```

```
// Update favorites display
```

```
function updateFavoritesDisplay() {
```

```

const favoritesContainer = $('#favoritesContainer');

if (favorites.length === 0) {
  favoritesContainer.html(`
    <div class="empty-favorites">
      <i class="fas fa-heart"></i>
      <p>No favorites yet</p>
      <p>Add drinks to your favorites!</p>
    </div>
  `);
} else {
  let favoritesHTML = "";
  favorites.forEach(product => {
    favoritesHTML += `
      <div class="favorite-item">
        
        <div class="favorite-info">
          <h4>${product.name}</h4>
          <p class="favorite-category">${product.category}</p>
          <p class="favorite-price">P$${product.price}</p>
        </div>
        <div class="favorite-actions">
          <button class="btn btn-primary btn-sm" onclick="addToCart('${product.name}',
${product.price})">
            <i class="fas fa-plus"></i> Add to Cart
          </button>
          <button class="btn btn-outline-danger btn-sm" onclick="toggleFavorite(${product.id})">
            <i class="fas fa-heart-broken"></i> Remove
          </button>
        </div>
      </div>
    `;
  });
}

```

```

        </div>
    </div>
    `;
});
favoritesContainer.html(favoritesHTML);
}
}

```

```
// Update order history display
```

```

function updateOrderHistoryDisplay() {
    const orderHistoryContainer = $('#orderHistoryContainer');

    if (orderHistory.length === 0) {
        orderHistoryContainer.html(`
            <div class="empty-orders">
                <i class="fas fa-receipt"></i>
                <p>No orders yet</p>
                <p>Place your first order to see it here!</p>
            </div>
        `);
    } else {
        let ordersHTML = "";
        orderHistory.forEach((order, index) => {
            const orderDate = new Date(order.date).toLocaleDateString('en-PH', {
                year: 'numeric',
                month: 'long',
                day: 'numeric',
                hour: '2-digit',
                minute: '2-digit'
            });

```



```
});
```

```
let itemsHTML = '';
```

```
order.items.forEach(item => {
```

```
  itemsHTML += `
```

```
    <div class="order-item">
```

```
      <span>${item.name} x${item.quantity}</span>
```

```
      <span>₹${(item.price * item.quantity).toLocaleString()}</span>
```

```
    </div>
```

```
  `;
```

```
});
```

```
ordersHTML += `
```

```
  <div class="order-card">
```

```
    <div class="order-header">
```

```
      <h5>Order #${orderHistory.length - index}</h5>
```

```
      <span class="order-date">${orderDate}</span>
```

```
    </div>
```

```
    <div class="order-items">
```

```
      ${itemsHTML}
```

```
    </div>
```

```
    <div class="order-total">
```

```
      <strong>Total: ₹${order.total.toLocaleString()}</strong>
```

```
    </div>
```

```
    <div class="order-status">
```

```
      <span class="status-badge status-${order.status.toLowerCase()}">${order.status}</span>
```

```
    </div>
```

```
  </div>
```

```
`;
```

```
});  
orderHistoryContainer.html(ordersHTML);  
}  
}
```

// Search and filter functionality

```
function filterProducts() {  
    const searchTerm = $('#searchInput').val().toLowerCase();  
    const categoryFilter = $('#categoryFilter').val();  
    const priceFilter = $('#priceFilter').val();  
  
    let filteredProducts = allProducts.filter(product => {  
        const matchesSearch = product.name.toLowerCase().includes(searchTerm);  
        const matchesCategory = categoryFilter === '' || product.category === categoryFilter;  
        const matchesPrice = priceFilter === '' ||  
            (priceFilter === 'low' && product.price <= 100) ||  
            (priceFilter === 'medium' && product.price > 100 && product.price <= 150) ||  
            (priceFilter === 'high' && product.price > 150);  
  
        return matchesSearch && matchesCategory && matchesPrice;  
    });  
  
    displayFilteredProducts(filteredProducts);  
}
```

// Display filtered products

```
function displayFilteredProducts(products) {  
    const productsContainer = $('#productsContainer');
```

```

if (products.length === 0) {
  productsContainer.html(`
    <div class="no-results">
      <i class="fas fa-search"></i>
      <p>No products found</p>
      <p>Try adjusting your search or filters</p>
    </div>
  `);
} else {
  let productsHTML = "";
  products.forEach(product => {
    const isFavorite = favorites.some(f => f.id === product.id);
    productsHTML += `
      <div class="product-card">
        <div class="product-image">
          
          <button class="favorite-btn ${isFavorite ? 'active' : ''}"
onlick="toggleFavorite(${product.id})">
            <i class="fas fa-heart"></i>
          </button>
        </div>
        <div class="product-info">
          <h3>${product.name}</h3>
          <p class="product-category">${product.category}</p>
          <p class="product-price">P$${product.price}</p>
          <button class="add-to-cart" data-name="${product.name}" data-
price="${product.price}">
            <i class="fas fa-plus"></i> Add to Cart
          </button>
        </div>
      </div>
    `;
  });
}

```

```

        </div>
    </div>
    `;
});
productsContainer.html(productsHTML);
}
}

```

// Show notification

```

function showNotification(message) {
    const notification = $(`
        <div class="notification">
            <i class="fas fa-check-circle"></i>
            <span>${message}</span>
        </div>
    `);

```

```

    $('body').append(notification);

```

// Animate in

```

notification.css({
    position: 'fixed',
    top: '20px',
    right: '20px',
    background: 'linear-gradient(135deg, #8B4513 0%, #654321 100%)',
    color: '#f4e4c1',
    padding: '1rem 1.5rem',
    borderRadius: '10px',
    boxShadow: '0 5px 15px rgba(0,0,0,0.3)',

```

```
    zIndex: '3000',  
    transform: 'translateX(100%)',  
    transition: 'transform 0.3s ease'  
  });
```

```
  setTimeout(() => {  
    notification.css('transform', 'translateX(0)');  
  }, 100);
```

```
// Remove after 3 seconds
```

```
  setTimeout(() => {  
    notification.css('transform', 'translateX(100%)');  
    setTimeout(() => {  
      notification.remove();  
    }, 300);  
  }, 3000);  
}
```

```
// Cart modal functionality
```

```
$('#cartIcon').click(function() {  
  $('#cartModal').fadeIn(300);  
  $('body').css('overflow', 'hidden');  
});
```

```
$('#closeCart').click(function() {  
  $('#cartModal').fadeOut(300);  
  $('body').css('overflow', 'auto');  
});
```

```
// Close modal when clicking outside
```

```
$('#cartModal').click(function(e) {  
  if (e.target === this) {  
    $('#cartModal').fadeOut(300);  
    $('body').css('overflow', 'auto');  
  }  
});
```

```
// Checkout functionality
```

```
$('#checkoutBtn').click(function() {  
  if (cart.length === 0) {  
    showNotification('Your cart is empty!');  
    return;  
  }  
}
```

```
// Create order object
```

```
const order = {  
  id: Date.now(),  
  date: new Date(),  
  items: [...cart],  
  total: cartTotal,  
  status: 'Completed'  
};
```

```
// Add to order history
```

```
orderHistory.unshift(order);
```

```
// Show checkout confirmation
```

```
const orderSummary = cart.map(item =>
```

```

        `${item.name} x${item.quantity} - ₪${(item.price * item.quantity).toLocaleString()}`
    ).join('\n');

    const confirmed = confirm(
        `Order Summary:\n\n${orderSummary}\n\nTotal: ₪${cartTotal.toLocaleString()}\n\nProceed to checkout?`
    );

    if (confirmed) {
        // Clear cart after successful checkout
        cart = [];
        updateCartDisplay();
        updateOrderHistoryDisplay();
        $('#cartModal').fadeOut(300);
        $('body').css('overflow', 'auto');
        showNotification('Order placed successfully! Thank you for choosing BrewTong!');
    }
});

// Add to cart button functionality
$(document).on('click', '.add-to-cart', function() {
    const name = $(this).data('name');
    const price = $(this).data('price');
    addToCart(name, price);

    // Add button animation
    $(this).addClass('added');
    setTimeout(() => {
        $(this).removeClass('added');
    }, 1000);
});

```

```

    }, 1000);
  });

  // Search and filter event listeners
  $('#searchInput').on('input', filterProducts);
  $('#categoryFilter').on('change', filterProducts);
  $('#priceFilter').on('change', filterProducts);

  // Smooth scrolling for navigation links
  $('.nav-link').click(function(e) {
    e.preventDefault();
    const target = $(this).attr('href');
    const offset = $('.header').outerHeight();

    $('html, body').animate({
      scrollTop: $(target).offset().top - offset
    }, 800);
  });

  // Initialize data on page load
  initData();

  // Initialize products display
  displayFilteredProducts(allProducts);

  // Add CSS for new features
  $('<style>')
    .prop('type', 'text/css')
    .html(`

```



```
.add-to-cart.added {  
    background: linear-gradient(135deg, #28a745 0%, #20c997 100%) !important;  
    transform: scale(1.05);  
}
```

```
.notification {  
    display: flex;  
    align-items: center;  
    gap: 0.5rem;  
    font-weight: 500;  
}
```

```
.notification i {  
    font-size: 1.2rem;  
}
```

```
.pulse {  
    animation: pulse 0.5s ease-in-out;  
}
```

```
/* Search and Filter Styles */
```

```
.search-filter-container {  
    background: linear-gradient(135deg, #f8f9fa 0%, #e9ecef 100%);  
    padding: 2rem;  
    border-radius: 15px;  
    margin-bottom: 2rem;  
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
}
```

```
.search-filter-row {  
  display: flex;  
  gap: 1rem;  
  align-items: center;  
  flex-wrap: wrap;  
}
```

```
.search-filter-row input,  
.search-filter-row select {  
  padding: 0.75rem 1rem;  
  border: 2px solid #dee2e6;  
  border-radius: 8px;  
  font-size: 1rem;  
  transition: border-color 0.3s ease;  
}
```

```
.search-filter-row input:focus,  
.search-filter-row select:focus {  
  outline: none;  
  border-color: #8B4513;  
}
```

/\* Product Cards \*/

```
.product-card {  
  background: white;  
  border-radius: 15px;  
  overflow: hidden;  
  box-shadow: 0 5px 15px rgba(0,0,0,0.1);  
  transition: transform 0.3s ease, box-shadow 0.3s ease;
```

```
}
```

```
.product-card:hover {  
    transform: translateY(-5px);  
    box-shadow: 0 10px 25px rgba(0,0,0,0.15);  
}
```

```
.product-image {  
    position: relative;  
    height: 200px;  
    overflow: hidden;  
}
```

```
.product-image img {  
    width: 100%;  
    height: 100%;  
    object-fit: cover;  
}
```

```
.favorite-btn {  
    position: absolute;  
    top: 10px;  
    right: 10px;  
    background: rgba(255,255,255,0.9);  
    border: none;  
    border-radius: 50%;  
    width: 40px;  
    height: 40px;  
    cursor: pointer;
```

```
    transition: all 0.3s ease;
}
```

```
.favorite-btn:hover {
    background: #ff6b6b;
    color: white;
}
```

```
.favorite-btn.active {
    background: #ff6b6b;
    color: white;
}
```

```
.product-info {
    padding: 1.5rem;
    text-align: center;
}
```

```
.product-category {
    color: #6c757d;
    font-size: 0.9rem;
    margin-bottom: 0.5rem;
}
```

```
.product-price {
    font-size: 1.25rem;
    font-weight: bold;
    color: #8B4513;
    margin-bottom: 1rem;
}
```

```
}
```

```
/* Favorites Styles */
```

```
.favorite-item {  
    display: flex;  
    align-items: center;  
    gap: 1rem;  
    padding: 1rem;  
    background: white;  
    border-radius: 10px;  
    margin-bottom: 1rem;  
    box-shadow: 0 2px 8px rgba(0,0,0,0.1);  
}
```

```
.favorite-image {  
    width: 80px;  
    height: 80px;  
    object-fit: cover;  
    border-radius: 8px;  
}
```

```
.favorite-info {  
    flex: 1;  
}
```

```
.favorite-category {  
    color: #6c757d;  
    font-size: 0.9rem;  
    margin: 0;
```

```
}
```

```
.favorite-price {  
  font-weight: bold;  
  color: #8B4513;  
  margin: 0;  
}
```

```
.favorite-actions {  
  display: flex;  
  flex-direction: column;  
  gap: 0.5rem;  
}
```

```
.empty-favorites {  
  text-align: center;  
  padding: 3rem;  
  color: #6c757d;  
}
```

```
.empty-favorites i {  
  font-size: 3rem;  
  margin-bottom: 1rem;  
  color: #dee2e6;  
}
```

```
/* Order History Styles */
```

```
.order-card {  
  background: white;
```

```
border-radius: 15px;
padding: 1.5rem;
margin-bottom: 1rem;
box-shadow: 0 3px 10px rgba(0,0,0,0.1);
}
```

```
.order-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 1rem;
  padding-bottom: 1rem;
  border-bottom: 2px solid #f8f9fa;
}
```

```
.order-date {
  color: #6c757d;
  font-size: 0.9rem;
}
```

```
.order-items {
  margin-bottom: 1rem;
}
```

```
.order-item {
  display: flex;
  justify-content: space-between;
  padding: 0.5rem 0;
  border-bottom: 1px solid #f8f9fa;
```

```
}
```

```
.order-total {  
  text-align: right;  
  font-size: 1.1rem;  
  margin-bottom: 1rem;  
  color: #8B4513;  
}
```

```
.order-status {  
  text-align: center;  
}
```

```
.status-badge {  
  padding: 0.5rem 1rem;  
  border-radius: 20px;  
  font-size: 0.9rem;  
  font-weight: 500;  
}
```

```
.status-completed {  
  background: #d4edda;  
  color: #155724;  
}
```

```
.empty-orders {  
  text-align: center;  
  padding: 3rem;  
  color: #6c757d;
```



```
}
```

```
.empty-orders i {  
  font-size: 3rem;  
  margin-bottom: 1rem;  
  color: #dee2e6;  
}
```

```
/* No Results */  
.no-results {  
  text-align: center;  
  padding: 3rem;  
  color: #6c757d;  
}
```

```
.no-results i {  
  font-size: 3rem;  
  margin-bottom: 1rem;  
  color: #dee2e6;  
}
```

```
/* Responsive Design */  
@media (max-width: 768px) {  
  .search-filter-row {  
    flex-direction: column;  
  }  
}
```

```
.search-filter-row input,  
.search-filter-row select {
```

```

        width: 100%;
    }

    .favorite-item {
        flex-direction: column;
        text-align: center;
    }

    .favorite-actions {
        flex-direction: row;
        justify-content: center;
    }
}

`)
.appendTo('head');
});

// Additional utility functions
function clearCart() {
    cart = [];
    updateCartDisplay();
    showNotification('Cart cleared!');
}

function clearFavorites() {
    favorites = [];
    updateFavoritesDisplay();
    saveData();
    showNotification('Favorites cleared!');
}

```

```
}
```

```
function clearOrderHistory() {  
  orderHistory = [];  
  updateOrderHistoryDisplay();  
  saveData();  
  showNotification('Order history cleared!');  
}
```

```
// Export functions for global access  
window.clearCart = clearCart;  
window.clearFavorites = clearFavorites;  
window.clearOrderHistory = clearOrderHistory;
```