Tran Khanh Tai

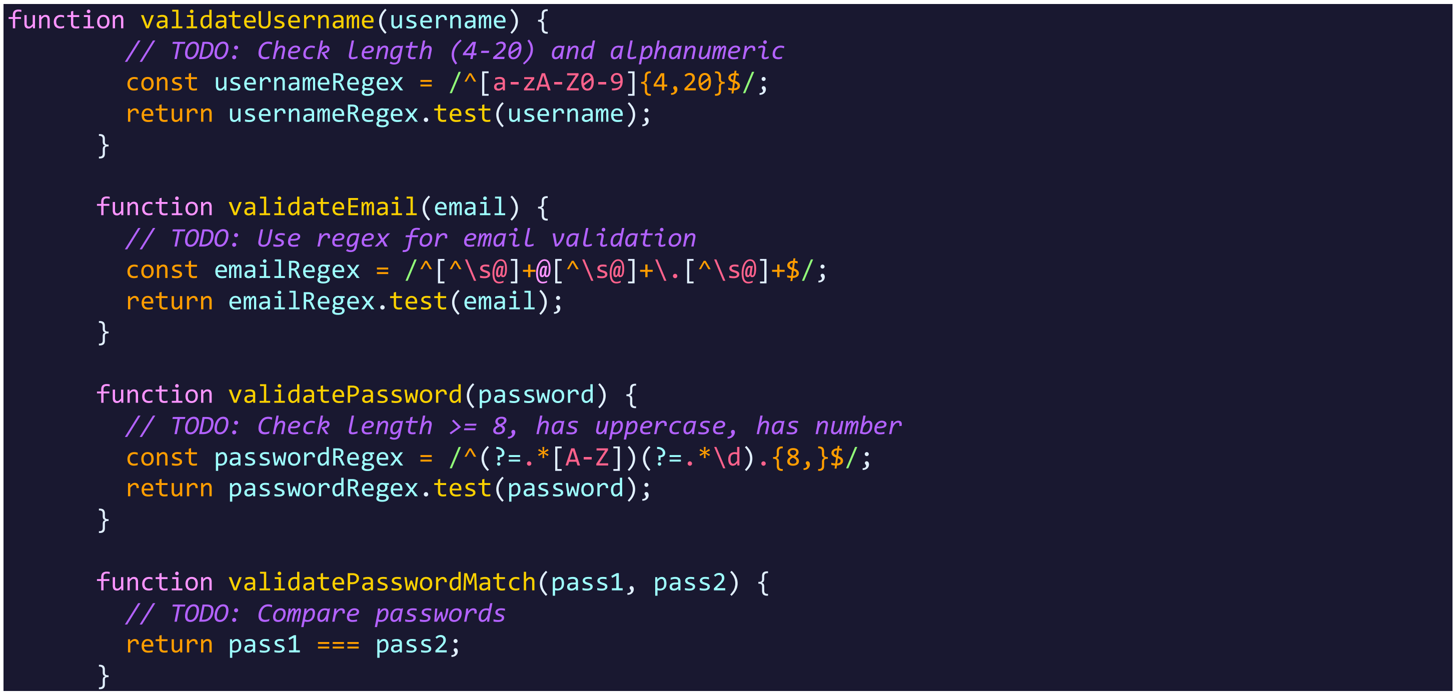
ITITIU21300

Github link: Web-Application-Lab/Lab 3 Practice at main · KhanhTaiTran/Web-Application-Lab

Lab 3 practice

Task 1.1: Interactive Form Validators

Validate the user input function



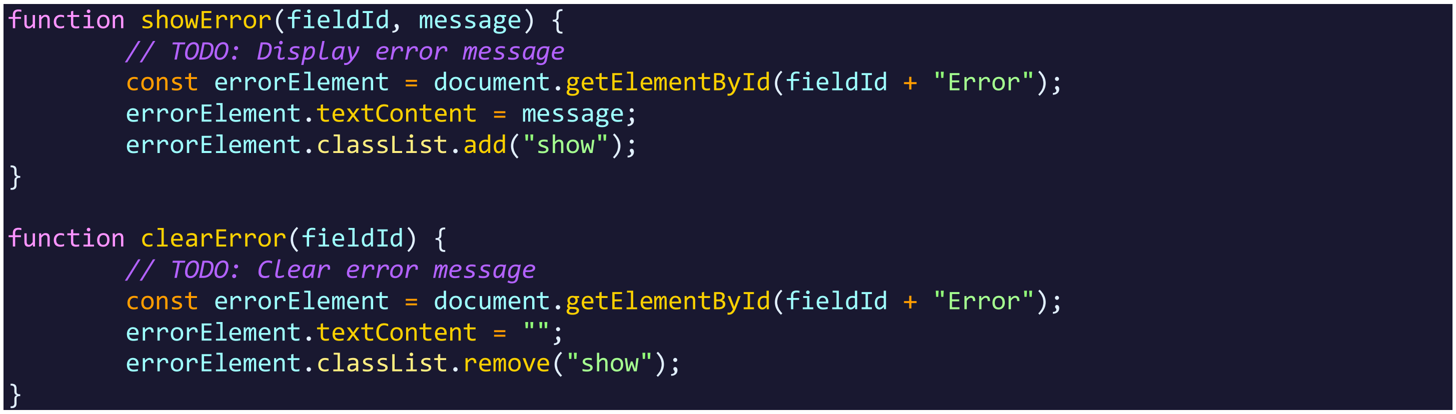
```
function validateUsername(username) {
        // TODO: Check length (4-20) and alphanumeric
        const usernameRegex = /^[a-zA-Z0-9]{4,20}$/;
        return usernameRegex.test(username);
      }

      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).{8,}$/;
        return passwordRegex.test(password);
      }

      function validatePasswordMatch(pass1, pass2) {
        // TODO: Compare passwords
        return pass1 === pass2;
      }
```

Show/clear error

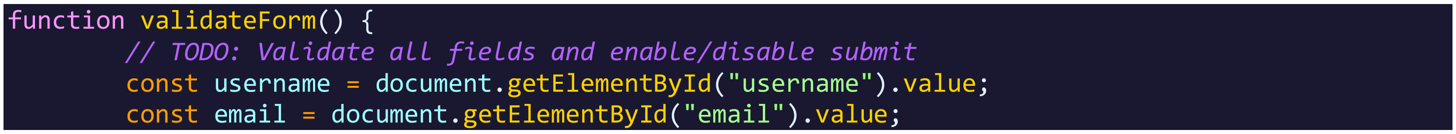


```
function showError(fieldId, message) {
        // TODO: Display error message
        const errorElement = document.getElementById(fieldId + "Error");
        errorElement.textContent = message;
        errorElement.classList.add("show");
}

function clearError(fieldId) {
        // TODO: Clear error message
        const errorElement = document.getElementById(fieldId + "Error");
        errorElement.textContent = "";
        errorElement.classList.remove("show");
}
```

These functions use to display the error message when user input an not valid info to the text box

Main validation logic



```
function validateForm() {
        // TODO: Validate all fields and enable/disable submit
        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 submitBtn = document.getElementById("submitBtn");
        let isValid = true;
        if (username.length === 0) {
          clearError("username");
          const un = document.getElementById("username");
          un.classList.remove("invalid");
          un.classList.remove("valid");
        } else if (!validateUsername(username)) {
          showError(
            "username",
            "Username must be 4-20 characters and alphanumeric"
          );
          isValid = false;
          document.getElementById("username").classList.add("invalid");
        } else {
          clearError("username");
          document.getElementById("username").classList.remove("invalid");
          document.getElementById("username").classList.add("valid");
        }
        if (email.length === 0) {
          //…code…
        } else if (!validateEmail(email)) {
          //…code…
         } else {
          //…code…
        }
        if (password.length === 0) {
          //…code…
        } else if (!validatePassword(password)) {
          //…code…
        } else {
          //…code…
        }
        if (confirmPassword.length === 0) {
          //…code…
        } else if (!validatePasswordMatch(password, confirmPassword)) {
          //…code…
        } else {
          //…code…
        }
        if (
          username.length === 0 ||
          email.length === 0 ||
          password.length === 0 ||
          confirmPassword.length === 0
        ) {
          isValid = false;
        }
        submitBtn.disabled = !isValid;
      }
```

This function use to validate the input

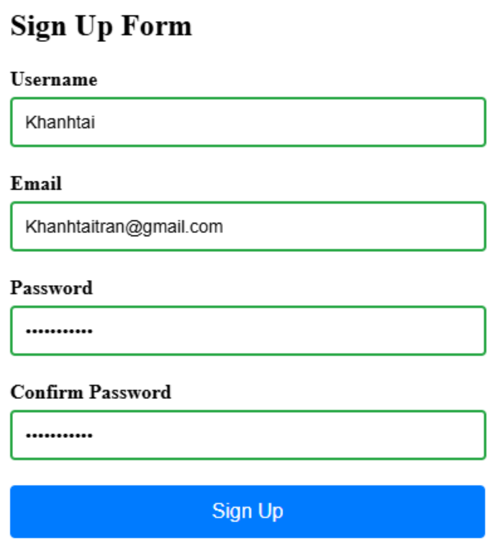
Each field is the same logic

- First, check if the length == 0, that mean the user has been type anything, don't display the error message and the red border.
- Then, check if it not validate, show error message and red border
- Else, clear the error message and green border
- Finally, check if there is any text field that not contain any letter, set isValid = false, and the submit button will disable

Handle form submition

```
document
        .getElementById("signupForm")
        .addEventListener("submit", function (e) {
          e.preventDefault();
          // TODO: Handle form submission
          alert("Form submitted successfully!");
          document.getElementById("signupForm").reset();
          document.getElementById("submitBtn").disabled = true;
          const inputs = document.querySelectorAll("#signupForm input");
          inputs.forEach((input) => {
            input.classList.remove("valid");
            input.classList.remove("invalid");
          });
        });
```

This code use to handle when we click the submit button, the form will reset and each text field will remove the color border, and set the button to disable



Sign Up Form

Username

Khanhtai

Email

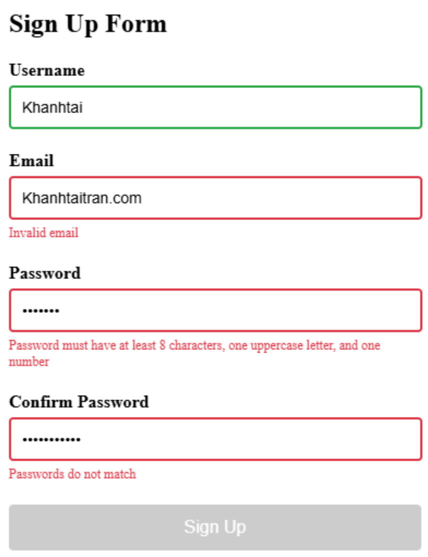
Khanhtaitran@gmail.com

Password

•••••••••••

Confirm Password

•••••••••••



Sign Up Form

Username

Khanhtai

Email

Khanhtaitran.com

Invalid email

Password

•••••••

Password must have at least 8 characters, one uppercase letter, and one number

Confirm Password

•••••••••••

Passwords do not match

Task 1.2: Dynamic Shopping Cart

List of products:

```
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: "⌚" },
      ];
```

Add to cart function:

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

        if (cartItem) {
          cartItem.quantity++;
        } else {
          cart.push({ ...product, quantity: 1 });
        }

        updateCartCount();
        renderCart();
      }
```

This function use to add the products to cart by using products id

- This function use the product id to find the product
- If the product is already exist in cart, it will increase the quantity by 1
- If new, add the product with quantity = 1
- And then update the display

Remove item from cart function:

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

This function use the id to filter item

- Filters out the item from the cart array :
    - Uses filter() to create new array excluding the itam with matching itemId
    - **Reassign cart**: The filtered array becomes the new cart
- Updates the display

Update the quantity function

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

This function use to update the quatity of the poducts by using the id. (This function is called by the [+] and [-] buttons in the cart)

- First, we find cart item
- If the item is in cartItem already, we will change the quantity = change (change is +1 or - 1)
- And if the quantity of item in cartItem <= 0, we will remove it from the Cart
- Otherwise, update the display

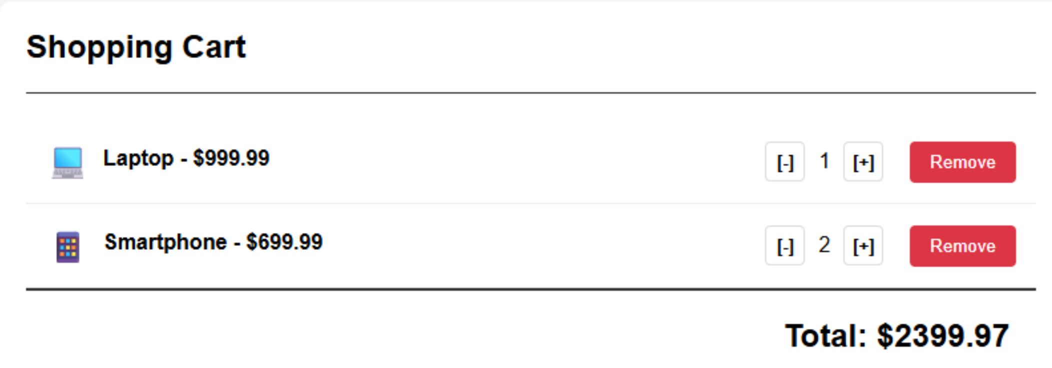
Calculate the total function

```
function calculateTotal() {
        // Calculate total price
        return cart.reduce(
          (total, item) => total + item.price * item.quantity,
          0
        );
      }
```

This function use reduce() method to iterates through entire cart array

- Start with total = 0
- And for each item add (price * quantity) to total

Example: Laptop ($999.99 × 1) + Smartphone ($699.99 × 2) = $2399.97



Shopping Cart

Laptop - $999.99 [-] 1 [+] Remove

Smartphone - $699.99 [-] 2 [+] Remove

Total: $2399.97

Update cart count function

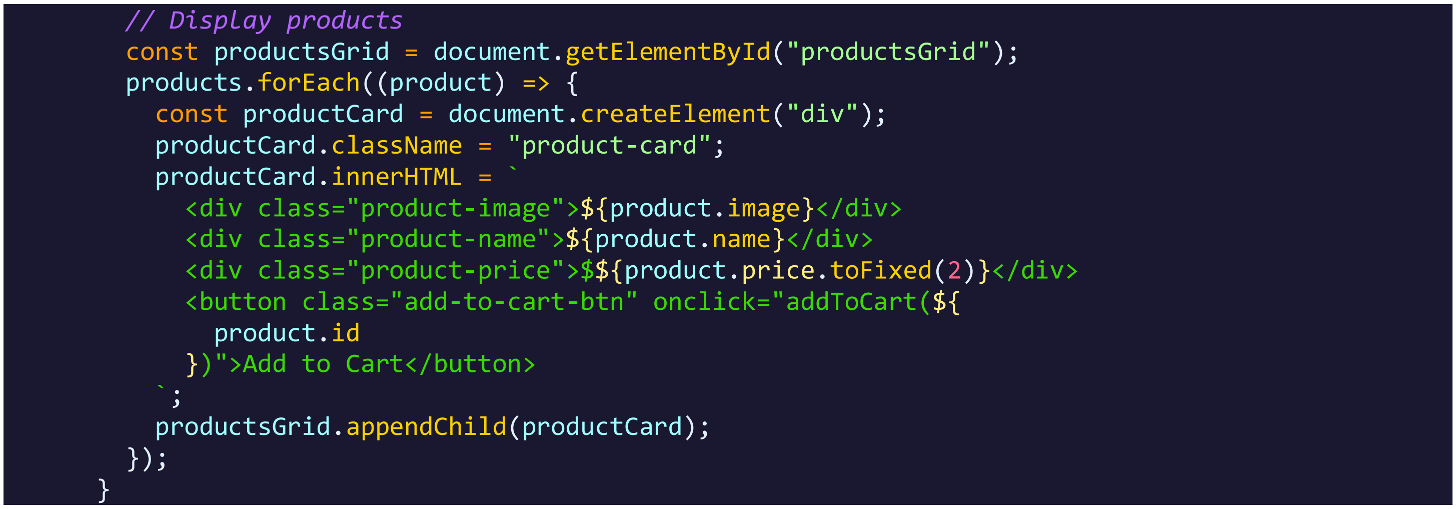
```
function updateCartCount() {
        // Update cart count badge
        const totalItems = cart.reduce(
          (total, item) => total + item.quantity,
          0
        );
        document.getElementById("cartCount").textContent = totalItems;
      }
```

Use reduce() to loop through the cart array and count the number of item in the cart and display it.

Cart (3)

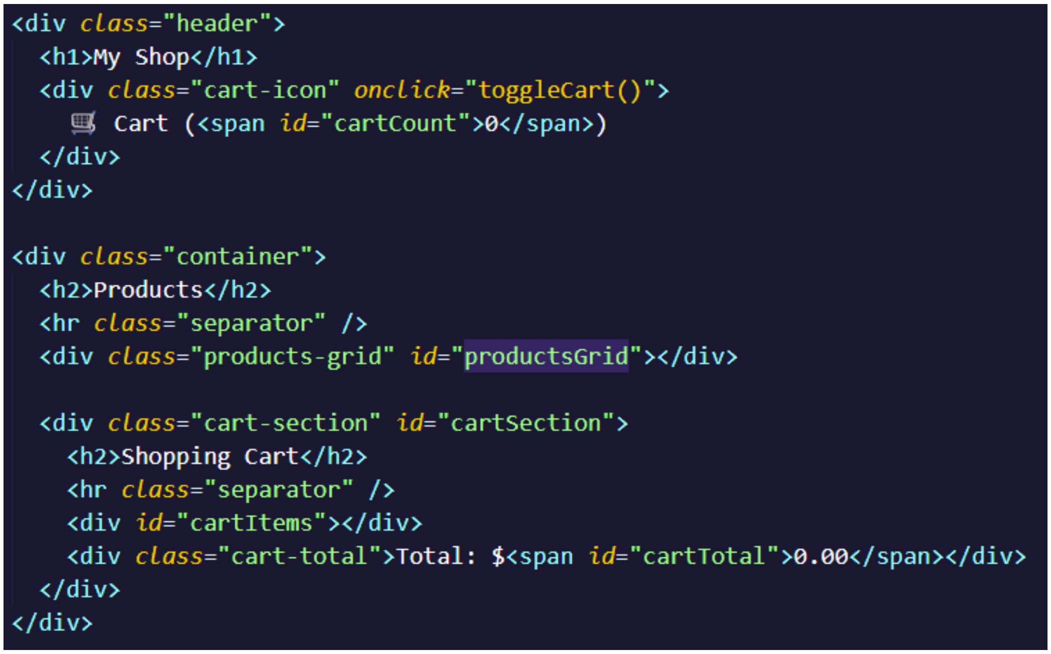
Render products function

```
function renderProducts() {
```



```
        // Display products
        const productsGrid = document.getElementById("productsGrid");
        products.forEach((product) => {
          const productCard = document.createElement("div");
          productCard.className = "product-card";
          productCard.innerHTML = `
            <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>
          `;
          productsGrid.appendChild(productCard);
        });
      }
```

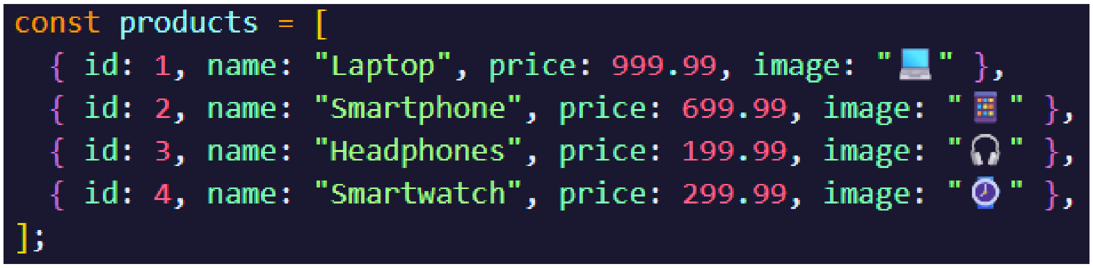
Here is the html section:



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

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

  <div class="cart-section" id="cartSection">
    <h2>Shopping Cart</h2>
    <hr class="separator" />
    <div id="cartItems"></div>
    <div class="cart-total">Total: $<span id="cartTotal">0.00</span></div>
  </div>
</div>
```

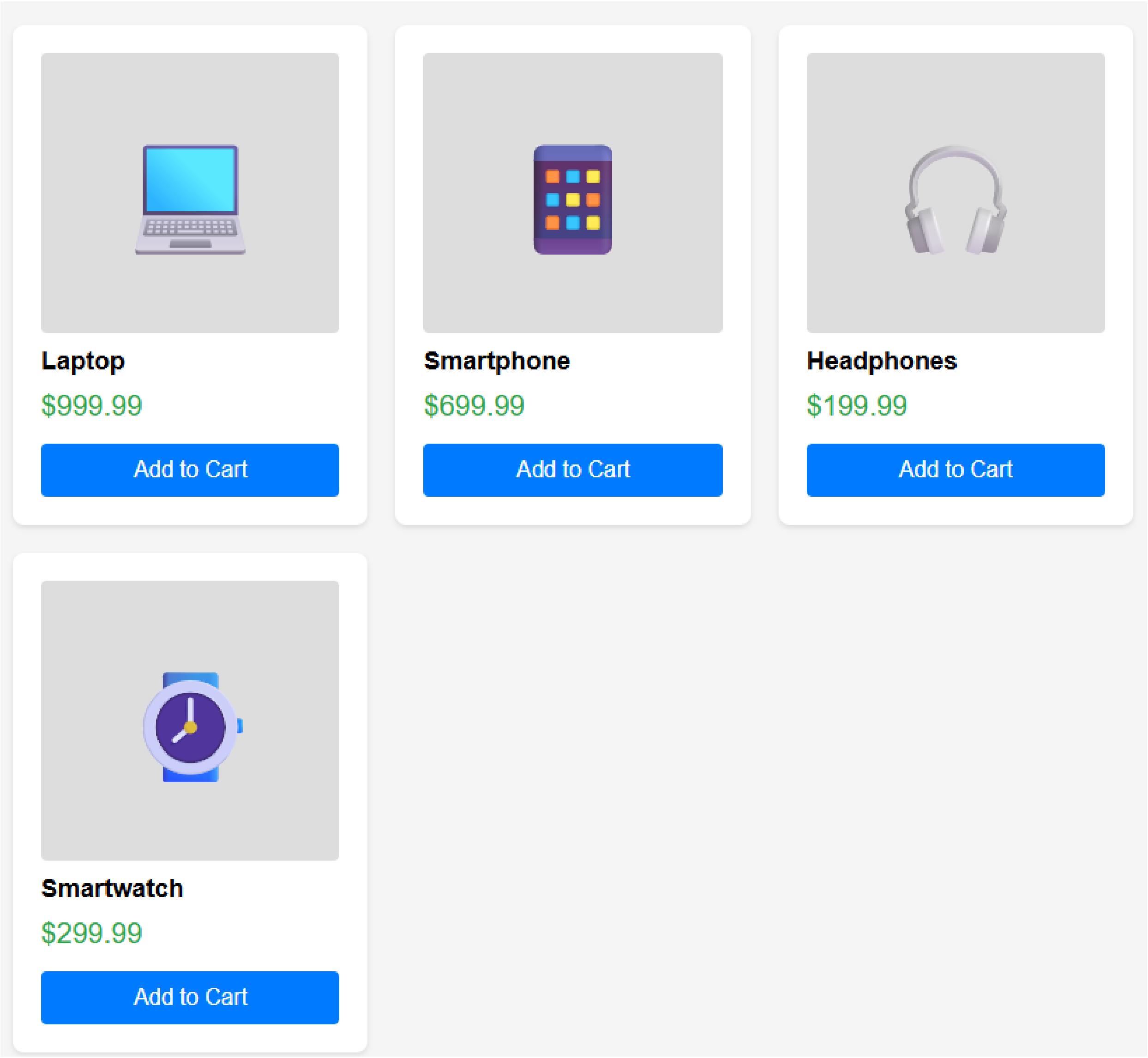
- First, we get the div id = productsGrid
- Then, loop through the products using forEach()
    - 

```
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: "⌚" },
];
```

- Then create card for eaach product:
    - Create a new <div> element
    - Add CSS class = product-card
    - Set innerHTML with product data
- Add to page: appends card to the grid

- Called once time druing the initialization

  Here is 4 cards for 4 item in card array



Laptop

$999.99

Smartphone

$699.99

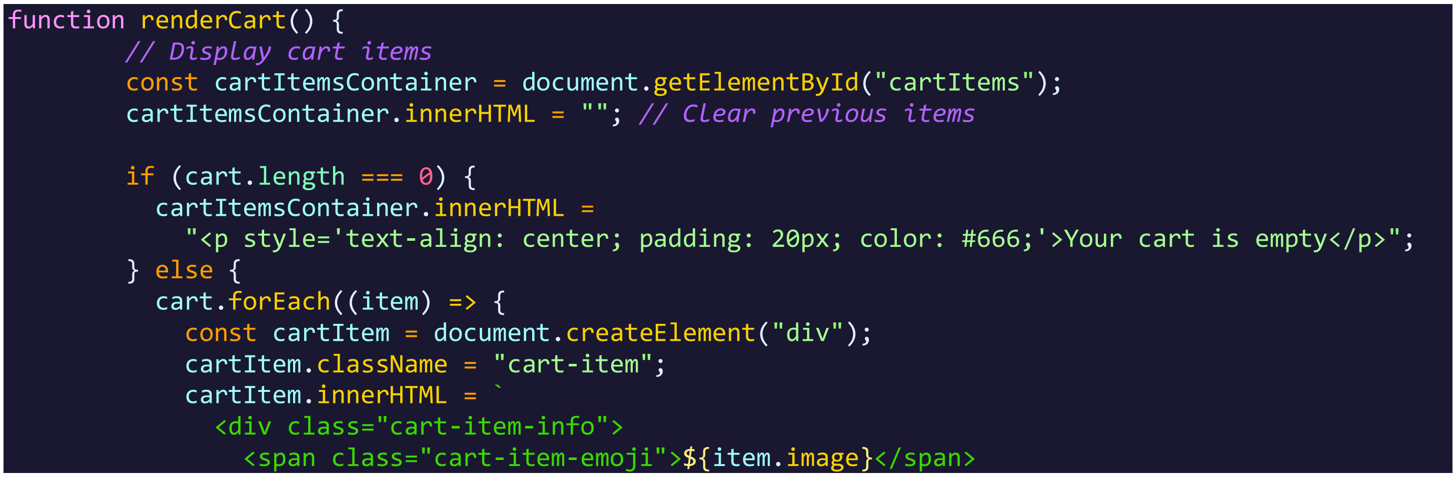
Headphones

$199.99

Smartwatch

$299.99

Render card function

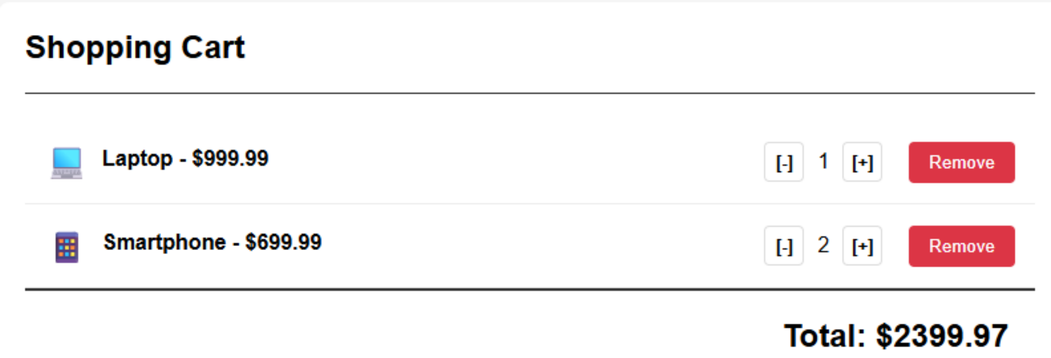


```
function renderCart() {
        // Display cart items
        const cartItemsContainer = document.getElementById("cartItems");
        cartItemsContainer.innerHTML = ""; // Clear previous items

        if (cart.length === 0) {
          cartItemsContainer.innerHTML =
            "<p style='text-align: center; padding: 20px; color: #666;'>Your cart is empty</p>";
        } else {
          cart.forEach((item) => {
            const cartItem = document.createElement("div");
            cartItem.className = "cart-item";
            cartItem.innerHTML = `
              <div class="cart-item-info">
                <span class="cart-item-emoji">${item.image}</span>
                <div class="cart-item-details">
                  <div class="cart-item-name">${
                    item.name
                  } - $${item.price.toFixed(2)}</div>
                </div>
              </div>
              <div class="quantity-controls">
                <button onclick="updateQuantity(${item.id}, -1)">[-]</button>
                <span class="cart-item-quantity">${item.quantity}</span>
                <button onclick="updateQuantity(${item.id}, 1)">[+]</button>
              </div>
              <button class="remove-btn" onclick="removeFromCart(${
                item.id
              })">Remove</button>
            `;
            cartItemsContainer.appendChild(cartItem);
          });
        }

        // Update total
        const cartTotal = document.getElementById("cartTotal");
        cartTotal.textContent = calculateTotal().toFixed(2);
      }
```

- First we also find the cartItems element
- Clear old content: Sets innerHTML = " " , to remove previous render
- Checking if cart is empty
  - Display: Your cart is empty
- Else, for each item
  - Create div with cart item layout
  - Shows emoji, name, price
  - Adds [-] [quantity] [+] controls
  - Adds Remove button



Shopping Cart

💻 Laptop - $999.99 [-] 1 [+] Remove

📱 Smartphone - $699.99 [-] 2 [+] Remove

Total: $2399.97

-

Toggle Cart function

```
function toggleCart() {
        // Show/hide cart section
        const cartSection = document.getElementById("cartSection");
        cartSection.classList.toggle("show");
      }
```

This function use to

- **Get cart section**: Finds the hidden cart div
- **Toggle visibility**: Adds/removes CSS class show
- **CSS does the work**: .cart-section.show { display: block; }
- **Called by**: Clicking the cart icon in header