Web Technologies

Exercise Sheet 3 – Web-Shop - JavaScript

Team 3: Jiahui Dai, Yana Halamakh, Wei Wei Tang

November 24, 2024

Contents

1 Registeration, Login, Customer Profile

1.1 form-validation.js

```
document.addEventListener("DOMContentLoaded", () => {
    const forms = document.querySelectorAll("form");

    // General form validation logic for all pages
    forms.forEach((form) => {
        form.addEventListener("submit", (e) => {
            let isValid = true;

            const usernameField = form.querySelector("#uName");
            const passwordField = form.querySelector("#password");
            const repeatPasswordField = form.querySelector("#repeatPasswo

            resetField(usernameField);
            resetField(passwordField);
            if (repeatPasswordField) resetField(repeatPasswordField);

            // Validate Username
            if (!validateUsername(usernameField.value)) {
                highlightField(usernameField, false);
                isValid = false;
            } else {
                highlightField(usernameField, true);
            }

            // Validate Password
            if (!validatePassword(passwordField.value)) {
                highlightField(passwordField, false);
                isValid = false;
            } else {
                highlightField(passwordField, true);
            }

            // Validate Password Repetition
            if (repeatPasswordField && passwordField.value !== repeatPass
                highlightField(repeatPasswordField, false);
                isValid = false;
            } else if (repeatPasswordField) {
                highlightField(repeatPasswordField, true);
            }

            if (!isValid) e.preventDefault();
        });
    });
```

```
// New: Add validation logic for Customer Profile page
const profileBox = document.querySelector(".profile-box");

if (profileBox) {
    const usernameField = profileBox.querySelector("#uName");
    const passwordField = profileBox.querySelector("#password");
    const usernameSubmitButton = profileBox.querySelectorAll("input[t
    const passwordSubmitButton = profileBox.querySelectorAll("input[t

    // Validate username change
    usernameSubmitButton.addEventListener("click", (e) => {
        resetField(usernameField);
        if (!validateUsername(usernameField.value)) {
            highlightField(usernameField, false);
            e.preventDefault();
        } else {
            highlightField(usernameField, true);
        }
    });

    // Validate password change
    passwordSubmitButton.addEventListener("click", (e) => {
        resetField(passwordField);
        if (!validatePassword(passwordField.value)) {
            highlightField(passwordField, false);
            e.preventDefault();
        } else {
            highlightField(passwordField, true);
        }
    });
}

// Helper functions
function validateUsername(username) {
    const minLength = 5;
    const hasUpperCase = /[A-Z]/.test(username);
    const hasLowerCase = /[a-z]/.test(username);
    return username.length >= minLength && hasUpperCase && hasLowerCa
}

function validatePassword(password) {
    const minLength = 10;
    return password.length >= minLength;
}

function highlightField(field, isValid) {
    field.style.borderColor = isValid ? "green" : "red";
}
```

```
    function resetField(field) {
        field.style.borderColor = "";
    }
});
```

2 Style Modifications

2.1 style-modification.js

```
document.addEventListener("DOMContentLoaded", () => {
    const toggleButton = document.querySelector(".mode-button");  // Corr
    const toggleImage = toggleButton.querySelector(".mode-img");
    const parentDoc = window.parent.document; // Access the parent docume
    const parentBody = parentDoc.body;
    const childBody = document.body; // Current iframe's body

    // Function to set the mode based on the saved value
    const setMode = (mode) => {
        const loginBox = parentDoc.querySelector('.login-box');
        const registerBox = parentDoc.querySelector('.register-box');

        if (mode === "dark") {
            parentBody.classList.add("dark-mode");
            childBody.classList.add("dark-mode");
            loginBox?.classList.add("dark-mode");  // Optional: Prevent e
            registerBox?.classList.add("dark-mode");
            toggleImage.src = "img/mode/dark.png";
        } else {
            parentBody.classList.remove("dark-mode");
            childBody.classList.remove("dark-mode");
            loginBox?.classList.remove("dark-mode");
            registerBox?.classList.remove("dark-mode");
            toggleImage.src = "img/mode/light.png";
        }
    };

    // Get the saved mode from localStorage
    const savedMode = localStorage.getItem("colorMode");
    setMode(savedMode || "light"); // Default to light mode

    // Add click event listener to toggle the mode
    toggleButton.addEventListener("click", () => {
        const isDarkMode = parentBody.classList.contains("dark-mode");

        if (isDarkMode) {
            setMode("light");  // Switch to light mode
            localStorage.setItem("colorMode", "light");
        } else {
            setMode("dark");  // Switch to dark mode
            localStorage.setItem("colorMode", "dark");
        }
    });
});
```

```
document.addEventListener("DOMContentLoaded", () => {
    // Get the orientation button and the image element
    const oriButton = document.querySelector('.ori-button');
    const oriImg = document.querySelector('.ori-img');
    const parentDoc = window.parent.document; // Access the parent docume

    // Retrieve the saved orientation state from localStorage
    const savedOrientation = localStorage.getItem('orientation') || 'lands

    // Set the initial orientation based on saved state
    if (savedOrientation === 'portrait') {
        parentDoc.body.style.width = '400px';  // Set the width to simula
        oriImg.src = 'img/orientation/toHorizontal.png'; // Set image to
    } else {
        parentDoc.body.style.width = '2000px';  // Set the width to simul
        oriImg.src = 'img/orientation/toPortrait.png'; // Set image to "p
    }

    // Function to toggle between portrait and landscape
    let isPortrait = savedOrientation === 'portrait';  // Initialize base

    oriButton.addEventListener('click', () => {
        if (isPortrait) {
            // If the screen is in portrait mode, change to landscape
            parentDoc.body.style.width = '2000px'; // Set to landscape wi
            oriImg.src = 'img/orientation/toPortrait.png';
            localStorage.setItem('orientation', 'landscape');
        } else {
            // If the screen is in landscape mode, change to portrait
            parentDoc.body.style.width = '400px'; // Set to portrait widt
            oriImg.src = 'img/orientation/toHorizontal.png';
            localStorage.setItem('orientation', 'portrait');
        }
        isPortrait = !isPortrait; // Toggle the state
    });
});
```

3 Collection List

3.1 collection-list.js

```
document.addEventListener("DOMContentLoaded", () => {
    const addToCollectionButton = document.querySelector(".add-to-collecti
    const quantityInput = document.querySelector(".quantity-input");
    const iframe = document.querySelector(".collection-list-iframe");

    // Function to send data to iframe
    function sendCollectionToIframe(collectionData) {
        iframe.contentWindow.postMessage({ type: 'updateCollection', colle
    }

    // Add event listener to the "Add to Collection" button
    addToCollectionButton.addEventListener("click", () => {
        const productName = document.querySelector("h1").textContent;
        const quantity = parseInt(quantityInput.value, 10);
        console.log(quantity)

        if (quantity <= 0) {
            alert("Quantity must be at least 1!");
            return;
        }

        // Get the existing collection data from iframe
        iframe.contentWindow.postMessage({ type: 'getCollection' }, '*');

        // Handle collection data from iframe and update
        window.addEventListener("message", (event) => {
            if (event.origin !== window.origin) return; // Validate the o
            const { type, collectionData } = event.data;

            if (type === 'collectionData') {
                console.log('--')
                console.log(collectionData[productName])
                // If the product already exists, accumulate the quantity
                if (collectionData[productName]) {
                    collectionData[productName].quantity += quantity; //
                } else {
                    // Otherwise, add the new product with the specified
                    collectionData[productName] = { quantity };
                }

                // Send the updated collection data back to the iframe
                sendCollectionToIframe(collectionData);
            }
        });
```

```
    });
});


document.addEventListener("DOMContentLoaded", () => {
    const collectionItemsContainer = document.getElementById("collection-
    const collectionList = {};

    // Function to render the collection list
    function renderCollectionList() {
        collectionItemsContainer.innerHTML = ""; // Clear the list

        for (const [productName, productData] of Object.entries(collectio
            const listItem = document.createElement("li");
            listItem.textContent = `${productName} - Quantity: ${productD

            const removeButton = document.createElement("button");
            removeButton.textContent = "Remove";
            removeButton.style.marginLeft = "10px";
            removeButton.addEventListener("click", () => {
                delete collectionList[productName];
                renderCollectionList();
                // Send the updated collection to the parent
                window.parent.postMessage({ type: 'updateCollection', col
            });

            listItem.appendChild(removeButton);
            collectionItemsContainer.appendChild(listItem);
        }
    }

    // Listen for messages from the parent page
    window.addEventListener("message", (event) => {
        if (event.origin !== window.origin) return; // Validate the origi

        const { type, collectionData } = event.data;

        if (type === 'updateCollection') {
            // Update the collection with the data from the parent
            Object.assign(collectionList, collectionData);
            renderCollectionList();
        } else if (type === 'getCollection') {
            // Send the current collection data to the parent
            window.parent.postMessage({ type: 'collectionData', collection
        }
    });

    // Initial render of the collection list (if any data is already ther
```

```
    renderCollectionList();
});
```

4 Calculating Prices and Further Functions

4.1 calculating-prices.js

```
document.addEventListener("DOMContentLoaded", () => {
    let calculateButton = document.getElementById("calculate-button");
    let priceInput = document.getElementById("priceWOTax");
    let priceResult = document.getElementById("priceWTax");
    let taxResult = document.getElementById("tax-result");

    // //Function to calculate price with taxes of 19%
    calculateButton.addEventListener("click", getTotalPrice);

    function getTotalPrice() {
        let tax = priceInput.value * 19 / 100;
        taxResult.value = tax;
        priceResult.value = Number(priceInput.value) + Number(tax);
    }
})
```

4.2 extra-function.js

```
document.addEventListener("DOMContentLoaded", () => {
    let convertButton = document.getElementById("convert-button");
    let fromCurrency = document.getElementById("from-currency");
    let toCurrency = document.getElementById("to-currency");
    let convertedResult = document.getElementById("converted-result");

    // Function to convert currency from EUR to < >
    convertButton.addEventListener("click", convert);

    function convert() {
        switch (toCurrency.value) {
            case "usd":
                convertedResult.value = fromCurrency.value * 1.04;
                break;
            case "gbp":
                convertedResult.value = fromCurrency.value * 0.83;
                break;
            case "cad":
                convertedResult.value = fromCurrency.value * 1.46;
                break;
            case "aud":
                convertedResult.value = fromCurrency.value * 1.60;
                break;
            case "jpy":
                convertedResult.value = fromCurrency.value * 161.25;
                break;
            case "sgd":
```

```
                convertedResult.value = fromCurrency.value * 1.40;
                break;
            case "hkd":
                convertedResult.value = fromCurrency.value * 8.11;
                break;
            case "cny":
                convertedResult.value = fromCurrency.value * 7.55;
                break;
            case "myr":
                convertedResult.value = fromCurrency.value * 4.64;
                break;
            case "krw":
                convertedResult.value = fromCurrency.value * 1463.80;
                break;
            case "btc":
                convertedResult.value = fromCurrency.value * 0.000011;
                break;
        }
    }
})
```

4.3 extra-function2.js

```
document.addEventListener("DOMContentLoaded", () => {
let iChooseYou = document.getElementById("i-choose-you");

    // Random choice of product
    iChooseYou.addEventListener("click", random);

    function random() {
        index = Math.floor(Math.random() * 6);
        switch (index) {
            case 0:
                window.location.href= "product_id/0001.html";
                break;
            case 1:
                window.location.href= "product_id/0002.html";
                break;
            case 2:
                window.location.href= "product_id/0003.html";
                break;
            case 3:
                window.location.href= "product_id/0007.html";
                break;
            case 4:
                window.location.href= "product_id/0008.html";
                break;
            case 5:
                window.location.href= "product_id/0009.html";
```

```
                    break;
                }

            }
        })
```

END OF DOCUMENT