ENSF-381: Assignment 3

Winter 2024

Department of Electrical and Software Engineering

Schulich School of Engineering

Due: March 11, 2024 at 11:59 PM

Congratulations on successfully building the foundation of your e-commerce website using HTML and enhancing its visual appeal with CSS. Now, it is time to elevate your project to new heights by introducing interactivity and dynamic features using JavaScript. In this assignment, your goal is to implement JavaScript to bring life to your e-commerce website. By the end of this assignment, your website should not only look good but also provide a seamless and responsive user interface.

Administrative Details

- The submission should be as a single compressed file (.zip) which includes a complete source code and resources (e.g., images) ready for execution without modifications. The name of the submitted file would have the following format: "Assignment#_ ENSF381_Section#_Group#". For example: "Assignment3_ENSF381_L01_Group01".
- When working in a group, submit ONLY ONE copy; i.e. do not submit 2 separate copies.
- Please mention the group member's name and UCID at the top of your code as source code comments in the main page (index.html).

Specifications:

Assignment 3 aims to enhance the e-commerce website's functionality using JavaScript. You need to work with your final code from Assignment 2. If you missed Assignment 1 or Assignment 2, you can get their solutions from D2L. The solution of Assignment 2 will be available on February 28.

Product Page (products.html) - 40 Marks:

Add to Cart Functionality:

- You must use JavaScript DOM manipulation to create a dynamic cart display section at the bottom of the page, just before the footer. This section will display the 1) product name, 2) unit price, 3) quantity and 4) button named "Remove".
- The "Remove" button should use the same style as the "Add to Cart" button.
- When a user clicks on the "Add to Cart" button for a product, the product details including name, unit price, and quantity should be added to the cart. An alert should be displayed confirming that the product has been added to the cart.
- If the item is already in the cart, you need to increase the quantity by 1 instead of creating a new cart item.

- Once you hover the mouse over the “Remove” button, change the background color to **#ff0000**.
- When the user clicks on the “Remove" button, it removes the quantity of the product by 1. If the quantity becomes 0 after removing, the item will be removed from the cart display.
- **See the “products.gif” for a visual representation of the dynamic cart.**

Login Page (login.html) - 30 Marks:

- When the user clicks on the “Login” button, make an API call to fetch user data from the endpoint: https://jsonplaceholder.typicode.com/users
- If there is an error in fetching data from the API endpoint, display an alert indicating that the API call was unsuccessful.
- Otherwise, the successful call returns a JSON array containing a list of user objects. Then, extract the “name” and “email” properties from each element in the returned JSON array.
- Validate the user-entered combination of username and password, input in the "Username" and "Password" fields, against the "name" and "email" properties of a user object obtained from the extracted JSON array, respectively. In other words, ensure that the entered username matches the extracted "name" property, and the provided password corresponds to the "email" address **for any user object obtained from the JSON array**.
- If there is a match between the user input and the retrieved data, display a success message, otherwise an error message.
- You must use JavaScript DOM manipulation to create the dynamic message box to display the success or error message. The specifications for the message box (HTML and CSS) are as follows:

  HTML:

  - Create a <div> element to serve as the message box container.
  - Inside the container, include a <p> element to display the message.

  CSS:

  - Style the message box container to have a border of **2px**, background color of **#f9f9f9**, padding of **10px**, and a margin **20px**.

- If you enter a different username and password, clicking the “Login” button will repeat the process of validating the user-input against the extracted data and replace the previous message with a new message according to the validation result. **Do not create a new message box every time you click the “Login” button**. **The website should not refresh the page.**
- **See “login.gif” for a visual representation of the interactive login page of your website.**
- Clicking the signup link will direct you to the signup page (signup.html) that you are going to build in the next step of this assignment.

Signup Page (signup.html) - 30 Marks:

Requirements:

- Create a signup page with HTML.
- Include a form with fields for username, password, confirm password, and email**.** Add a “Signup” button.

Place the following elements in your HTML at these specific locations and do not change the order:

- **<header>**: This is where your website's logo and name should be placed. The logo should be placed at the left and name should be placed right after the logo.
- **<nav>**: The navigation links to other pages (products.html, contact.html, and login.html) are placed directly below the header.
- **<main>**: Below the navigation bar, where the signup form, username, password, confirm password, and email, and a "Signup" button, should be placed.
- **<footer>**: Add a footer with a copyright notice at the bottom of your page.

You need to add the following CSS styling for this page:
Signup Form Section:

- Use flexbox to align content both vertically and horizontally.
- Align the content of the flexbox at the center of the section.
- Style the text content with a background color of **#ddd** (medium gray), padding of **20px**, margin of **20px**.
- Use a border-radius of **10px** to round the corners of the form.

Form Fields:

- Style the label of the form fields in a block container and margin of **10px.**
- Style the form fields with a background color of #fff (white), padding of **10px**, margin of **10px**.
- Add a border-radius of **5px** to round the corners of the fields.
- Add placeholder text in the username, password, confirm password, and email fields.
- Make sure all the input fields have “required” attribute.

“Signup” Button:

- Style the "Signup" button with a contrasting color of **#4CAF50** (green), padding of **10px**, border radius of **5px**, margin of **10px,** and opacity of **0.5.**
- Add a hover effect by changing the background color to **#45A049** (a slightly darker shade of green) and opacity of **1.0.**

Here are the formatting requirements for the input in the username, password, confirm password, and email fields:

Username:

- Must be between 3 and 20 characters long.
- Allowed characters: alphanumeric characters (letters A-Z, numbers 0-9), hyphens (-), and underscores (_).
- Must start with a letter.
- Cannot contain spaces or special characters other than hyphens and underscores.

Password:

- Must be at least 8 characters long.
- Must contain at least one uppercase letter, one lowercase letter, one number, and one special character.
- Allowed special characters: !@#$%^&*()-_=+[]{}|;:'",.<>?/`~.
- Cannot contain spaces.

Confirm Password:

- Must match the password entered in the password field.

Email:

- Must be a valid email address format (e.g., username@example.com).
- Cannot contain spaces.
- Must contain an "@" symbol followed by a domain name (.com, .net, .io).

- Now, you can make your signup page interactive using JavaScript. If the user's input does not follow the requirements mentioned above, when you click the "signup" button a message box will indicate which field/fields does/do have the wrong input. You do not need to provide which rules have been violated in the user inputs. If the user inputs meet the requirements, a successful message will be displayed instead.
- You must use JavaScript DOM manipulation to create the dynamic message box to display the error or success message. Specifications for the message box (HTML and CSS):

  HTML:

  - Create a <div> element to serve as the message box container.
  - Inside the container, include a <p> element to display the error message.

  CSS:

  - Style the message box container to have a border of **2px**, background color of **#f9f9f9**, padding of **10px**, and a margin **20px**.

See "signup.png" for an interactive signup page of your website.