

My Front-End Code Snippet Assignment: Zach Delapenha

Developing a Responsive Navigation Bar with Login and Sign Up Form validation.

Introduction

As part of my front-end code snippet assignment, I have developed a responsive web page that features a navigation bar with integrated login and sign-up forms. The purpose of this project is to demonstrate my understanding of HTML, CSS, and JavaScript in creating an interactive and user-friendly code snippet, complete with form validation. Here is a detailed overview of the structure, components, and functionality of the web page I designed for the assignment.


Structure of the index.html (Home) page

Head section

The `<head>` element contains meta-information, such as the character set (UTF-8), browser compatibility, and the viewport settings, which are essential for responsive web design. The title of the web page is specified with the `<title>` element. Additionally, the page uses an external stylesheet, "style.css," to apply styles and formatting to its elements.


Body section

The `<body>` element comprises the visible content of the web page, which includes the following components:

1. Navigation bar: For this assignment, I designed a responsive navigation bar. The navigation bar includes a logo, links to the home and about pages, login and registration buttons, and a hamburger menu icon for mobile devices.  Nav bar Navigation bar features:
 - Solid colour navigation bar on scroll - To enhance the user experience, the navigation bar changes to a solid colour as the user scrolls down the page. This feature is achieved by selecting the nav element through a query selector then using an if / else statement that adds a CSS class, 'nav-active', to the navigation bar when the window is scrolled a certain distance.
 - Burger menu for smaller screens - The burger menu icon is designed to open and close the navigation menu on smaller screens. A toggleMenu function is created to add or remove the 'active' class for each menu item, allowing the menu to expand or collapse when the burger menu icon is clicked.
 - Displaying login and registration forms - Event listeners are added to the login and registration buttons in the navigation bar. When clicked, the respective form is displayed while hiding the other form.

- Fully responsive - Using CSS the navigation bar is made to be fully responsive on all device sizes, this is achieved by through a mixture of techniques including media queries.

2. Login form: I integrated a hidden login form that is displayed when the login button is pressed. The form uses the GET method to submit data to the "dashboard.html" file. It contains fields for the user's email and password, along with demo values provided for convenience.

 alt text

3. Registration form: Similar to the login form, I incorporated a hidden registration form that appears when the sign-up button is clicked. The form submits data using the GET method to the "thanks.html" file. It includes fields for the user's username, email, password, and phone number, accompanied by validation requirements for each field.

 alt text Form features:

- Closing popups - Event listeners are added to the red 'X' buttons of the login and registration forms, allowing users to close the popups when the buttons are clicked.
 - Form switching - Event listeners are also added to the anchor tags at the bottom of the login and registration forms, allowing users to switch between the two forms by clicking the links.
 - Login form validation - For the purpose of this assignment, a demo email and password are used to validate the login form. If the entered credentials match the demo values, the form is submitted, and the user is redirected to their dashboard.
 - Registration form validation - The registration form includes several input fields, each with specific validation requirements. The form uses regular expressions to validate the user's input in real-time, adding 'valid' or 'invalid' classes to the input fields accordingly. Upon successful validation, the form is submitted, and the user is redirected to a thank you page.
4. Hero section: The hero section consists of a large image and accompanying text. As this assignment is focused on the form validation and Navigation bar, each page only consists of a simple hero section with the relevant heading.

 alt text

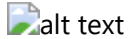
JavaScript inclusion

To ensure that the web page is interactive and functional, and to improve code readability, I included an external JavaScript file, "index.js," which handles various events, such as displaying the hidden login and registration forms when the respective buttons are clicked. I placed the link to this file at the bottom of the `<body>` to ensure the script does not run until the page is fully rendered.

Dashboard.html

Once the user successfully logs in, they are directed to the dashboard page, where the navigation bar is updated to include a logout button. The dashboard page serves as a placeholder for future back-end integration, which will display user-specific information tailored to the individual's needs and preferences. The dashboard page's primary purpose is to establish a foundation for a more personalized user experience, eventually allowing the user to access relevant data and interact with the website in a meaningful way. By

incorporating the logout button within the updated navigation bar, users are provided with a seamless and user-friendly method to log out of their account when desired, enhancing the overall user experience.



Thankyou.html

Upon successful registration, the user is taken to a thank you page that displays a message thanking them for their registration. The page also provides a link that directs users back to the home page, prompting them to log in with their newly-created account. Currently serving as a placeholder, the thank you page has the potential to be expanded with future back-end integration to incorporate additional features such as email verification. This enhancement would improve security and ensure that users have provided valid email addresses during the registration process.



Bugs and fixes

- The original validation function for the registration form was targeting all input elements within the index.html page, which prevented the separation of the login form inputs for different validation rules. This issue was resolved by targeting the registration form inputs specifically with `const inputs = document.forms.regForm.querySelectorAll("input");` instead of using `const inputs = document.querySelectorAll("input");`.
- There was a problem where users could not close a popup once it was opened. To address this issue, a button was added to change the display property of the active form to 'none', allowing users to close the form they are currently viewing.
- Additionally, users were unable to switch between forms once they had clicked on one. This problem was resolved by adding a link at the bottom of each form, which closes the current form and opens the other when clicked.