# Build a Multi-page Responsive Website

## Challenge

Using the wireframe you created as your first exercise, develop it into a fully functioning, multi-page website. The website should showcase the following techniques you've learned this week:

1. \*\*Flexbox\*\* for layout structure.

2. \*\*Flexbox alignment\*\* to position elements.

3. At least \*\*one pseudo-element or pseudo-class\*\*.

4. Use of \*\*CSS variables\*\* for consistent styling (e.g., colors, borders).

5. A \*\*contact page\*\* with a styled contact form.

Your site should include multiple pages (e.g., Home, About, Services, Contact) and must be fully responsive, adapting to various screen sizes and devices.

## Key Learnings

By completing this exercise, you will:

- Gain hands-on experience building and laying out a real, responsive website.

- Understand how to use \*\*Flexbox\*\* for modern page layouts.

- Learn how to apply \*\*CSS variables\*\* for consistent theming across your website.

- Practice integrating \*\*pseudo-elements or pseudo-classes\*\* for dynamic or decorative styling.

- Learn how to build and style a functional \*\*contact form\*\*.

## User Story

As a web developer, I want to create a multi-page, responsive website that uses modern layout techniques like Flexbox, CSS variables, and pseudo-elements so that I can build a flexible, maintainable website that adapts to various screen sizes and devices.

## Acceptance Criteria

1. \*\*Structure and Layout:\*\*

- Use \*\*Flexbox\*\* to structure the layout of your pages.

- Properly utilize \*\*flexbox alignment\*\* to ensure items are well-positioned within their containers.

- Ensure the layout is \*\*responsive\*\*, adapting to different screen sizes (desktop, tablet, mobile).

2. \*\*Styling:\*\*

- Include at least one \*\*pseudo-element or pseudo-class\*\* in your styling (e.g., `:hover`, `:before`, `:after`).

- Use \*\*CSS variables\*\* for managing consistent color schemes, and optionally for other properties like borders and spacing.

3. \*\*Multi-page Navigation:\*\*

- Your website should contain more than one page (e.g., Home, About, Services, Contact).

- Each page should be well-structured and follow a cohesive design.

- Include a \*\*navigation bar\*\* that links to the different pages.

4. \*\*Contact Page:\*\*

- Create a \*\*contact form\*\* on the Contact page with styled inputs for name, email, message, and a submit button.

- Ensure the contact form is styled properly and provides a good user experience on both desktop and mobile devices.

5. \*\*Code Quality:\*\*

- Use clean, well-commented, and formatted HTML and CSS.

- Ensure your website is \*\*accessible\*\*, using semantic HTML tags where appropriate (`<header>`, `<main>`, `<section>`, `<footer>`, etc.).

## Example Flow

1. \*\*Step 1\*\*: Start by reviewing the wireframe you created in the first exercise.

2. \*\*Step 2\*\*: Set up your project folder and create multiple HTML pages (e.g., index.html, about.html, contact.html).

3. \*\*Step 3\*\*: Implement your layout using \*\*Flexbox\*\*. Apply flexbox alignment techniques to position elements as needed.

4. \*\*Step 4\*\*: Use \*\*CSS variables\*\* to handle consistent color schemes and optionally other properties like borders and spacing.

5. \*\*Step 5\*\*: Style your contact page with a \*\*contact form\*\* and ensure it adapts well to different screen sizes.

6. \*\*Step 6\*\*: Incorporate \*\*pseudo-elements or pseudo-classes\*\* into your styling for interactive or decorative effects.

7. \*\*Step 7\*\*: Test your website on different devices and screen sizes to ensure it is fully responsive.

## Resources

- [CSS Tricks: A Complete Guide to Flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)

- [MDN Web Docs: Pseudo-elements](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements)

- [MDN Web Docs: CSS Variables](https://developer.mozilla.org/en-US/docs/Web/CSS/Using\_CSS\_custom\_properties)

- [MDN Web Docs: Responsive Web Design Basics](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\_layout/Responsive\_Design)

# challenge2-multipagewebsite