SparkINN Novice Solution Pvt. Ltd

Week 2 CSS Weekend Task: Building a Responsive Portfolio Website

---

Objective:
In this weekend task, you will apply the concepts learned in the CSS course to design and style a **responsive portfolio website**. This task will help you consolidate your understanding of CSS layout techniques, media queries, flexbox, grid, and responsive design principles. You will create a personal portfolio website that showcases your projects, skills, and contact information, focusing on making the site visually appealing and responsive across different devices.

Task Overview:

You will be required to:

1. Set Up the Project Environment:

- Create a new folder named `portfolio_website` on your computer.
- Inside the folder, create the following structure:
  - **css/**: For storing stylesheets.
  - **images/**: For storing image assets.
  - **index.html**: The main HTML file for your website.

---

2. Create the Basic HTML Structure:

- **Problem**: Use HTML5 to create the basic structure of your portfolio website.
  - Include sections such as **Home**, **About**, **Projects**, and **Contact**.
  - Use semantic HTML tags like `<header>`, `<section>`, `<footer>`, etc.
  - Ensure the HTML is well-structured and accessible.
- **Hint**: Add basic content in each section to be styled later with CSS.

---

3. Implement Responsive Layouts Using Flexbox and Grid:

- **Problem**: Use **CSS Flexbox** and **CSS Grid** to create responsive layouts for different sections of the website.
  - Use **Flexbox** for the navigation menu to horizontally align the links.
  - Use **CSS Grid** to design the projects section, displaying the projects in a grid format.
  - Ensure the layout adjusts based on screen size (mobile, tablet, and desktop).

- **Hint**: Use `display: flex;` and `display: grid;` along with responsive units (e.g., `fr`, `auto`, `minmax()`).

---

4. Implement Media Queries for Responsive Design:

- **Problem**: Use **CSS Media Queries** to ensure that your website looks great on mobile devices.
  - Create breakpoints for mobile (max-width: 600px), tablet (max-width: 900px), and desktop (above 900px).
  - Adjust font sizes, layouts, and other design elements based on the device's screen size.
- **Hint**: Use media queries to modify the grid layout for the projects section to display in a single column on mobile devices and multiple columns on desktop.

---

5. Styling with CSS:

- **Problem**: Use **CSS properties** to style different sections of the portfolio website.
  - Add fonts, colors, margins, padding, and background images to enhance the design.
  - Style buttons and links for interactivity.
  - Use **hover** and **active** states for buttons and links to make them interactive.
- **Hint**: Use Google Fonts for typography, and make use of CSS variables for consistent color theming.

---

6. Styling the Projects Section with Grid:

- **Problem**: Use **CSS Grid** to create a responsive projects section.
  - Display projects in a grid with 2 columns on tablets and 3 columns on desktops.
  - Each project card should have a title, description, and image.
- **Hint**: Use `grid-template-columns` to define the layout structure.

---

7. Implement CSS Animations and Transitions:

- **Problem**: Use CSS **transitions** and **animations** to add subtle interactivity.
  - Animate the navigation links when the user hovers over them.
  - Create a smooth scrolling effect when navigating to different sections of the page.
  - Use `:hover` to add transition effects to project cards.

---

8. Bonus Task (Optional): Adding Dark Mode

- **Problem**: Implement **Dark Mode** using CSS variables.

    - Use CSS variables for colors and background, and toggle between light and dark mode using a button.
    - Write JavaScript to toggle the class on the body and switch between the two modes.
- **Hint**: Define two sets of color variables for light and dark mode and switch between them based on a button click.

Submission:

- Zip your `portfolio_website` folder and submit it.
- Ensure that the website is fully responsive and all sections are properly styled.
- Include a **README** file with instructions on how to run the website.