

## Task 2: Create a Portfolio Landing Page

### Objective

Design and develop a single-page portfolio website that showcases your skills, projects, and contact information. This task focuses on using modern CSS layout techniques, particularly Flexbox and Grid, to create a responsive and visually appealing design.

### Requirements

1. HTML Structure:
  - Create a single HTML file (`index.html`) that includes all sections of your portfolio.
  - Use semantic HTML5 tags to structure your content (e.g., `<header>`, `<nav>`, `<main>`, `<section>`, `<footer>`).
2. Sections to Include:
  - Header with navigation
  - Hero/Introduction section
  - Skills or Services section
  - Project showcase
  - About Me
  - Contact form or contact information
  - Footer
3. CSS Styling:
  - Create a separate CSS file for your styles.
  - Implement a cohesive color scheme and typography that reflects your personal brand.
  - Use CSS Grid for the overall page layout.
  - Use Flexbox for smaller component layouts within grid areas.
4. Responsive Design:
  - Make the website fully responsive using media queries.
  - Ensure the layout adapts well to mobile, tablet, and desktop screen sizes.
5. Navigation:
  - Create a navigation menu that smoothly scrolls to different sections of the page.
6. Project Showcase:
  - Display at least 4 projects with images, titles, and brief descriptions.
  - Use Grid or Flexbox to create an attractive layout for the projects.
7. Interactivity:
  - Add hover effects to buttons and project items using CSS transitions.

### Additional Guidelines

- Be creative with your design while maintaining professionalism.
- Consider using CSS custom properties (variables) for colors and repeated values.
- Implement subtle animations or transitions to enhance user experience.
- Test your website across different devices and browsers.

### Advice for Using Grid and Flexbox

- Use Grid for the main page layout, defining different areas for each section.
- Utilize Flexbox for aligning items within each grid area, such as centering content or creating flexible rows/columns.
- Combine Grid and Flexbox: For example, use Grid for the project showcase layout and Flexbox for aligning content within each project card.
- Take advantage of Grid's ability to create complex layouts with overlapping elements if desired.
- Use Flexbox for the navigation menu to easily align items and make it responsive.

## **Submission**

Submit a zip file containing your HTML file, CSS file, and any images or assets used in the project. Include a README.md file with:

- A brief description of your portfolio
- Any special instructions for viewing the website
- A list of the key CSS techniques you used (especially regarding Grid and Flexbox)
- Any challenges you faced and how you overcame them

Bonus: Deploy your portfolio to a free hosting service (like GitHub Pages or Netlify) and include the live URL in your submission.

⇒ **Submit [here](#)**