SIT120 - Introduction to Responsive Web Apps

Distinction Task 3.2D

Advanced Personal Portfolio Website

Overview: This task is based on Task 1.3P - [your previous portfolio website]. You are required to improve your previous version of the portfolio website with the extra new skills you have gained. This task should be done only using HTML, CSS, and **Vue 3**. *You are not allowed to use any third-party libraries or frameworks.*

Minimum Requirements:

1. **Responsive Design:** Implement a fully responsive design using different techniques to ensure the website looks great and is responsive on various devices and screen sizes.
2. **CSS Styling:** Create a visually appealing layout and apply custom styles to all sections of the website, paying attention to typography, colours, and animations.
   a. **Typography:** Choose a legible and complementary font pair for headings and body text. Implement a typographic scale with consistent line-height and letter-spacing to enhance readability.
   b. **Color Scheme and Contrast:** Develop a harmonious color palette that maintains proper contrast ratios for accessibility. Use color variations to distinguish different sections and UI elements.
   c. **CSS Grid and Flexbox:** Utilize CSS Grid for larger layout structures and Flexbox for smaller, flexible components. This combination provides precise control over the placement and alignment of elements. CSS Grid's grid-template-areas can be used to define different layouts for various breakpoints, while Flexbox can handle the inner alignment of items within each grid cell.
   d. **CSS Custom Properties (Variables):** Implement CSS custom properties to centralize color schemes, typography settings, and other design constants. This makes it easier to maintain consistency and make global design changes across the entire website.
3. **HTML Elements:** Use a variety of HTML elements such as tables, forms, headers, favicons, etc. to enhance the website's structure, accessibility, and visual appeal.
   a. **Semantic HTML**: Choose appropriate HTML elements to improve accessibility and SEO. Use semantic elements like <nav>, <article>, <section>, and <footer> to provide meaningful structure and improve screen reader compatibility.
   b. **Forms with Validation:** Create interactive forms using <form> and <input> elements. Apply HTML5 form validation attributes for user-friendly feedback. Use progressive enhancement techniques to ensure forms are functional even if JavaScript is disabled.
4. **Vue 3 Concepts:**
   Implement all Vue 3 concepts from Credit Task 3 (OnTrack Task 2.3) to achieve different features, functionalities and components that you want to add in your website. This should clearly reflect your mastery and clear understanding of those Vue 3 concepts and your capability to implement them as you need. To highlight some:
   a. **Vue Router:** You MUST use Vue Router for routing through multiple pages within the website for seamless navigation. You should have home page, projects page, about page, contact page, and add any other relevant pages.

   b. **Conditional Rendering:** Implement v-for, v-if etc. to handle dynamic rendering of project details, conditional display of elements, and user interactions, respectively.
   c. **Vue Components:** Create reusable Vue components for common elements like the navigation bar, project cards, and contact form, to promote code reusability and maintainability.
5. **Advanced Animations:** Incorporate CSS animations and Vue transitions to add a touch of interactivity and elegance to the website. See https://www.w3schools.com/css/css3_animations.asp for example.
6. **External Data:** Fetch project details for Project page or any other pages from an external JSON file to make it easy to update and manage project information. You can store a local JSON file in your working directory or get JSON file from URLlink.
7. Additional Web Pages/Sections: Apart from pages that you already have, add following pages/sections to your website.
   a. Admin Register/Login Page: Create an admin login page with username and password. You can use localstorage function from javascript (https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_win_localstorage) or use more advanced approach to store the credentials. If admin credentials do not exist then, you need to register first and thus should have option to register also.
   b. Resume/CV Page: Add a resume page for your website similar to or better than this example https://www.w3schools.com/howto/tryw3css_templates_cv.htm
   c. Blog Post Page: In this page, admin can add, edit, update and delete the blogs when logged in using Admin Login Page. These functionalities and layouts (input fields, create button etc.) should be hidden when admin is not logged in. Text only blog is sufficient for this task with only title and body text content enough for the blog. Also, list of blogs should be displayed all the time even if admin is logged in or not. And when users click a single blog item from the list, it should go to single details blog post. You are recommended to utilize your reflections for all the tasks that you have already written and done as blog content but also can be something different.
   d. Footer Section: Your website should have very good footer section with information such as copyright, social media icons with links to your social media profiles like linkedIn etc. Check Deakin website footer section for example https://www.deakin.edu.au/ .
8. **For extra marks:** Publish your website. Provide the URL to your portfolio website. [Only free options recommended]

Resource: Some ideas for portfolio website can be found in this link https://www.w3schools.com/howto/howto_website_create_portfolio.asp .

Note: *You cannot use dummy values such as lorem-ipsum. The whole website should contain meaningful contents, pages, etc. You are encouraged to exceed minimum requirements above and demonstrate your skill as a proficient web developer. Resources provided are only for hints and to provide some idea.*

Reflection (minimum 500 words):

Write a report including:

a. Reflection on the learning experience and how this project helped you grow as a developer. Discuss the challenges you faced and how you overcame them.

b. Describe how you applied HTML, CSS, and Vue 3 concepts to complete the project.
c. Mention the specific features and techniques you used to enhance the website's design, interactivity, and responsiveness in comparison to previous version of the website.
d. Screenshots wherever necessary should be included of your website.

Submission Details:

1. Make sure you follow these instructions strictly.
2. All tasks need to follow these guidelines:
    1. GitHub Repo is a MUST. No zip files will be accepted. Please maintain one task sheet submissions in one GitHub Repo. Don’t group all task sheets in one GitHub repo. [If you are keeping repo private then add tutors as collaborators]
    2. Demo video MUST be submitted. It MUST show website/app running successfully. Share your video through Deakin air, SharePoint, or unlisted YouTube link etc. (You can do the screen recording through Zoom, Zoom is free for all Deakin students)
    3. Provide direct link to both GitHub repo and video separately.
    4. Submit PDF report including reflections, screenshots etc.
3. Plagiarism is unacceptable.
4. Due date: Please follow OnTrack due date.