Name	**Zelefack Marie Noelle**
Task number	**1**

Web designing documentation

What is web designing

Web designing is the process of planning, conceptualizing, and implementing the plan for designing a website in a way that is functional and offers a good user experience.

1. WEB DESIGN PRINCIPLES

1.1.Simplicity

- **Clarity and Ease of Use:** Aim for a clean and straightforward design. Avoid clutter by focusing on essential elements and minimizing distractions.

- **Minimalism:** Use space effectively and include only necessary elements to make the site easier to navigate and more visually appealing.

1.2.Consistency

- **Uniform Layout and Design:**Ensure consistency in layout, color schemes , typography, and design elements across all pages. This helps create a cohesive user experience.

- **Design Patterns:**Utilize familiar design and conventions to make the website intuitive

1.3.Visual Hierachy

- **Prioritization of Elements:**Use size , color, contrast and placement to highlight the most important elements , guiding users' attention to key areas.

1.4. Accessibility

- **Inclusive Design:** Ensure the website is accessible to people with disabilities by following the Web Content Accessibility Guidelines WCAG
- **Assistive Technologies:** Incorporate features such as alt for images ,proper heading structures, and keyboard navigation support.

1.5.Responsiveness

- **Mobile-Friendly Design:** Ensure your website functions well on various devices including desktops, tablets and smartphones.
- **Flexible Layouts:**Use responsive techniques ,like fluid grids and flexible images to adjust the layout based on screen size

1.6.Navigation

- **Intuitive Navigation:** Design a clear and straightforwared navigation menu that helps users find information easily.
- **Breadcrumbs:** Provide breadcrumbs to help users understand their location within the site and navigate back easily.

1.7.Load Time

- **Optimized Performance:** Optimize images, use efficient coding practices and leverage browser caching to improve load times.
- **Fast loading:** Ensure that the website loads quickly to enhance user experience and SEO rankings.

1.8.Typography

- **Readability:**Choose fonts that are easy to read on all devices and sizes.
- **Visual Appeal:**Use fonts that are easy to read on all devices and sizes.

These are just some few web designing principles and by applying them on your designs you'll be able to produce what will satisfy your user's needs

2. Web Semantics

Web semantics, also known as the Semantic Web, refers to a vision for the future of the web in which data is structured and linked in a way that it can be easily processed and understood by machines, enabling better data integration, sharing, and reuse across various applications and platforms.

In front-end development it involves incorporating semantic HTML elements, using ARIA roles for accessibility, and embedding structured data to enhance the meaning and context of your web content. Here's a detailed guide on how to apply these concepts effectively:

2.1. Use Semantic HTML Elements:

2.1. Use Semantic HTML Elements: Elements clearly describe their meaning in a human and machine readable way. These elements help with both SEO and accessibily.

- **Header:** Represents introductory content or a set of navigational links.
- **Nav:** Represents a section of a page intended for navigation.
- **Section:** Represents a standalone section of a document.
- **Article:** Represents a self-contained composition that can be distributed independently.
- **Footer:** Represents a footer for its nearest sectioning content or sectioning root element.
- **Aside:** Represents content that is tangentially related to the content around it.

2.1.1. Use ARIA(Accessible Rich Internet Applications) Roles

- **Role="navigation"**: Defines a region of the page intended for navigation.
- **Role="main"**: Indicates the primary content of the document.
- **Role="complementary"**: Denotes a supporting section of the main content.
- **Role="banner"**: Defines site-wide information at the top of the page.
- **Role="contentinfo"**: Marks the area containing information about the parent document or website.

2.1.2 . Embed Structured Data: Structured data helps search engines understand the content of your website better, which can enhance your search results with rich snippets.

2.1.3. Improve Content Accessibility and Usability: Accessible and usable content ensures a wider audience can effectively interact with your website.

- **Alt text for images:** Provide alternative text for images to describe their content. Provide alternative text for images to describe their content.
- **Descriptive links:** Use clear and descriptive text for links.
- **Heading Structure:** Use a logical heading structure to organize content.
- **Form labels:** Ensure all form inputs have associated labels.

2.1.4. Validate and Test: Regularly validate your HTML and structured data to ensure compliance and functionality.

- **HTML Validation:** Use the W3C Markup Validation Service to check the validity of your HTML.
- **Structured Data Testing:** Use tools like Google's Structured Data Testing Tool to validate your structured data.
- **Accessibility Testing:** Use tools like WAVE and Lighthouse to assess and improve the accessibility of your website.

CSS (Cascading Style Sheets) can also benefit from semantic principles. While CSS itself doesn't have semantic meaning, you can write and structure your CSS in ways that promote better readability, maintainability, and accessibility, which align with the goals of semantic web design. Here are some best practices and techniques

- ❖ **Use Meaningful and ID names**

- ❖ **Structured CSS:** makes it easier to read , understand and maintain

- ❖ **Use CSS Variables:** makes it easier to manage and reuse variables

- ❖ **Improve Accesibility With CSS:** Accessible css enhances the usability for all users including those with disabilities that is the use of focus states and Accessibility

- ❖ **Responsive Design:** ensures your site looks good on all devices

- ❖ **Documentation and Comments:** documenting your css with comments help others understand the code and its purpose.

3. CORE CSS DOCUMENTATION

4.

- **BOX MODEL**

  It is basically used for design and layout , each element is surrounded by rectangular box consisting of margin, padding, border and margin.Understanding the box model is crucial for controlling an elements layout and dimension.

- **Layout and Positioning**

  - ❖ Display properties such as **inline , block , inline-block** can be used for layout purposes , with the primary difference between the three being : inline-block the top and bottom margins are respected but with display inline they are not , compared to display block the major difference is that display inline-block does not add add a line-break after the elements ,so the element can sit next each other.

  - ❖ Display **Float** property can also be used for positioning, taking values like left ,right , none(default), inherit, initial

  - ❖ Display **Flex** property which is a one dimensional layout can also be used for positioning it has the following properties flex-direction, flex-wrap, flex-flow, justify-content , align-items , align-content, we also have flex-items properties like order, flex-grow, flex-shrink, flex-basis, flex, align-self

- Display **Grid** property is two dimensional and can be used for positioning, it has the following properties column-gap, gap, grid, grid-area, grid-auto-columns, grid-auto-flow, grid-column,grid-column-start, grid-gap,grid-template,grid-row-item,grid-template,grid-template-areas, grid-template-rows, grid-template-columns,row-gap, justify-content,align-content

- We also have **Position** property which can take values such as static, absolute, fixed, relative, sticky, initial, inherit.The z-index property specifies the stack order of an element , elements with greater stack order is placed infront of an element with lower stack order and also it works with position and flex properties only

- **CSS Selectors**

  - Simple selectors: selects elements based on name, id , class

  - Combinator selectors: selects based on a specific relationship between them

  - Pseudo-class selectors: select and style a part of an element

  - Attribute selectors: select element based on an attribute or attribute value

- **Transform property**

  It applies to 2D or 3D transformation to an element. It has the following values rotate , skew, scale, translate, initial, inherit

- **Transition property**

  It transitions allows you to change property values smoothly , over a given duration. It has the following values transition , transition-delay , transition-duration(default 0), transition-property, transition-timing-function

- **CSS Animations**

  An animation lets an element gradually change from one style to another. It has the following values keyframes, animation name, animation-duration, animation-delay, animation-iteration-count, animation -direction, animation-timing-function, animation-fill mode animation

- **CSS Responsiveness**

It Ensures the webpage looks good on all devices , it uses only html and CSS .For responsive layouts media queries can be used, adjusting the display flex and grid and ensuring the meta tag with viewport is included