InternCall Full-Stack Web Development Assignments Phase-II

HTML Assignments (20)

1. **Basic HTML Structure**: Create an HTML document with appropriate <html>, <head>, and <body> tags.

2. **HTML Tags and Elements:** Build a webpage containing various HTML elements such as headings, paragraphs, images, and links.

3. **Comments in HTML:** Insert comments within your HTML code to describe the purpose of each section or element

4. **HTML Case Sensitivity:** Experiment with different casing for HTML tags and observe how they render in the browser.

5. **Creating Lists:** Construct an unordered list with at least three list items, each item describing a different topic.

6. **Building Forms:** Design a simple form with fields for name, email, and password using appropriate input types.

7. **Embedding Multimedia:** Embed an audio or video file into your webpage using appropriate HTML tags.

8. **Implementing Navigation:** Develop a navigation bar using an unordered list and style it using CSS to display horizontally.

9. **Constructing Tables:** Create a table with borders, containing information such as student names and their corresponding scores.

10. **Using iFrame:** Embed a Google Maps location using the <iframe> tag.

11. **Creating Clickable Images:** Display multiple images on the webpage and make them clickable, redirecting to different webpages when clicked.

12. **Differentiating <div> and <span>:** Use <div> and <span> elements to style and organize different sections of content within your webpage.

13. **Adding Line Breaks:** Use the <br> tag to add line breaks within a paragraph to format the text appropriately.

14. **Working with Forms:** Create a form with checkboxes and a dropdown menu for users to select their preferred options.

15. **Adding Images with Alt Text**: Insert images into your webpage and include appropriate alt text for accessibility purposes.

16. **Semantic HTML:** Replace generic <div> elements with more semantically meaningful elements such as <header>, <nav>, <main>, <article>, <section>, and <footer>.

17. **Hyperlinks and Anchors:** Create hyperlinks to external websites and anchors within the same webpage.

18. **Lists and Nesting:** Experiment with nested lists to create a hierarchical structure of information.

19. **Tables with Headers and Captions:** Enhance your table by adding headers and captions to improve readability.

20. **Accessibility Practices:** Ensure that your webpage complies with accessibility standards by adding appropriate attributes such as alt text for images and aria-label for interactive elements.

CSS Assignments (20):

1. **Internal CSS Styling:** Style your HTML document internally using CSS to change the font, color, and background of elements.

2. **External CSS Linking:** Link an external CSS file to your HTML document and apply styles to headings, paragraphs, and lists.

3. **Inline CSS Application:** Apply inline CSS styles to individual elements within your HTML document.

4. **Class Selectors:** Define CSS classes and apply them to multiple elements to style them uniformly.

5. **ID Selectors**: Apply unique styles to specific elements using ID selectors.

6. **Element Selectors:** Experiment with styling different HTML elements using element selectors.

7. **Font Properties:** Customize the font family, size, weight, and style of text within your webpage.

8. **Color and Backgrounds:** Apply colors and background images to elements to enhance visual appeal.

9. **Box Model Manipulation:** Adjust the padding, margin, and border of elements to control their spacing and layout.

10. **Float and Clear Techniques:** Float images or elements within a container and clear floats to prevent layout issues.

11. **Creating a Basic Layout:** Design a simple webpage layout with header, footer, sidebar, and main content sections.

12. **Responsive Design Basics:** Use media queries to create a responsive layout that adjusts based on screen size.

13. **Positioning Elements:** Experiment with different positioning techniques such as relative, absolute, fixed, and sticky.

14. **Text Formatting:** Experiment with text formatting properties such as text-align, text-decoration, and text-transform.

15. **Lists and List Styling:** Customize the appearance of lists using CSS to create bullet styles or remove default list styles.

16. **Creating Buttons:** Style <button> elements to create visually appealing buttons with hover and active states.

17. **Image Styling:** Apply styles to images such as borders, shadows, and filters to enhance their appearance.

18. **Flexbox Basics:** Explore basic flexbox properties to create flexible layouts.

19. **Grid Basics:** Experiment with basic grid properties to create grid-based layouts.