



Basic Web Development Assessment questions

PLACEMENTDOST

Note:

Please be advised that these assessment inquiries are designed to accommodate interns with diverse skill levels, ranging from novices to seasoned analysts. Should you encounter any challenging questions, you are encouraged to seek solutions independently or reach out to us for assistance at intern@Placementdost.com

Best wishes for success in completing the assessment!

- 1. Basic HTML Structure Creation:** Create a simple webpage with HTML, including essential elements like headings, paragraphs, lists, and images. Ensure proper semantic structure and adherence to HTML5 standards.
- 2. CSS Styling and Layout Design:** Style the webpage using CSS to enhance its visual appearance. Apply basic styling techniques such as color schemes, font styles, margins, and padding. Design a basic layout using CSS positioning and box model properties.
- 3. Responsive Design Implementation:** Make the webpage responsive to different screen sizes by using CSS media queries. Adjust the layout and styling to provide an optimal viewing experience on desktops, tablets, and smartphones.
- 4. Introduction to JavaScript Interactivity:** Add basic interactivity to the webpage using JavaScript. Implement simple functionalities such as button click events, form validation, or image sliders. Use JavaScript to manipulate the DOM (Document Object Model) dynamically.
- 5. Integration of External Resources:** Incorporate external resources into the webpage, such as CSS frameworks (e.g., Bootstrap) or JavaScript libraries (e.g., jQuery). Learn how to link external files and utilize pre-built components to enhance the webpage's design and functionality.
- 6. Debugging:** Practice debugging skills by identifying and fixing common issues in the webpage's HTML, CSS, and JavaScript code. Use browser developer tools to inspect elements, analyze network requests, and troubleshoot errors.

Projects:

Build a Personal Portfolio Website: Create a simple portfolio website that showcases your skills, projects, and contact information. Use HTML for structure, CSS for styling, and optionally add interactivity with JavaScript.

Design a Landing Page: Design and implement a landing page for a fictional product or service. Focus on creating an attractive layout with clear call-to-action buttons and compelling content. Use CSS frameworks like Bootstrap or Tailwind CSS for responsive design.

Build a Simple Calculator: Create a basic calculator application with HTML, CSS, and JavaScript. Implement arithmetic operations (addition, subtraction, multiplication, division) and provide a user-friendly interface for inputting numbers and performing calculations.