Interview Based Assessment Solutions

HTML Questions

Answer 1: <!DOCTYPE html> is not a tag in html. It is a declaration called the Document Type Declaration. It is used to specify the version of HTML being used in a document. It informs the web browser about rules and regulations that should be applied when rendering HTML content.

Answer 2: Semantic tags in HTML are a set of elements that provide specific meaning and structure of web content. These tags are designed to show the purpose of content they enclose.
There are many tags like <header>, <footer>, <nav>, <section>, <main> etc. It would be easily readable and clarify the code and help to remain content well structured.

Answer 3: HTML tags : HTML tags are used to mark up and enclose content and they are represented by angled brackets (< >). They define the structure and behavior of the elements within a HTML document. There are two types of HTML tags, first one is written in opening tag and closing tag like <p> tag used for paragraph and </p> used to close tag of paragraph. Second one is a self closing tag like <img> tag, they do not need to close the tag.
HTML Element : HTML elements are created by using HTML tags. It contains opening tags, content like <p> Hello World </p>. Here <p> is the opening tag and Hello World is content and the closing tag is </p>. This whole line is an HTML element.

Answer 4 : Github Repo Link

Answer 5 : Github Repo Link

Answer 6 : There are some advantages of HTML5 over its previous version. HTML5 supports multimedia like <video> and <audio> tags. HTML5 introduced <canvas> element, which allows dynamic rendering of 2D graphics and animation without plugins. HTML5 improved form validation also in url, date or email. HTML5 introduced a new set of elements like <header>, <footer>, <nav> and <section> for better code readability and well structured. It introduced localStorage and sessionStorage to store data locally and offline. HTML5 improves performance and capability for better error handling, code optimization and script handling.

Answer 7 : Github Repo Link

Answer 8 : The <img> tag is used to display an image on a webpage and <figure> tag is used to group content like an image with its caption.

Answer 9 : HTML Tag: An HTML tag represents an element or a structure within an HTML document. It is enclosed with an angle bracket(< >) and it contains strat tag, content and end tag.
Example: <p>This is paragraph</p>.
HTML Attributes: HTML attributes used to modify behavior, appearance and functionality of the tags. Attributes are applied within the opening tag of an HTML element and consist of a name and a value. Example: <img src="image.jpg" alt="image">.
Global Attributes: class, id, style, title, tabindex.

Answer 10 : Github Repo Link

CSS Questions

Answer 1 : Box model in CSS is a concept that describes how elements on a webpage are rendered and what their dimensions and space are. It consists of the content area, padding, border, and margin.

Answer 2 : Selectors in CSS are used to target specific HTML elements and apply style on them. There are lots of selector in CSS and all with its own advantages and use cases.

1. Type Selector: This selector targets their tag name and applies style on the tag. Ex. ‘p’, ‘h1’, ‘div’.
2. Class Selector: Class selector target specific class attribute. It's also reusable to use the same style on different code. Ex. .my-class.
3. Id Selector: Id selector target specific id attribute. They are used to style unique elements. Ex. #my-id.
4. Attribute Selector: Attribute selectors allow you to target elements based on their attribute values. Ex. [type=”text”]
5. Descendant Selector: Descendant selectors target elements that are descendants of other elements. They allow you to apply styles to specific elements that are nested inside other elements. Ex. div p.
6. Child Selector: Child selectors are similar to descendant selectors but only target elements that are direct children of other elements. Ex. div > p.
7. Pseudo-classes and Pseudo-elements: Pseudo-classes and pseudo-elements allow you to apply styles to elements based on various states or positions within the document. Ex. :hover, :first-child, ::before.

Answer 3 : VW (viewport width) and VH (viewport height) are units of measurement in CSS that are based on the size of the screen. Instead of using fixed sizes like pixels, VW and VH use percentages of the width and height of the screen. This means that when the screen size changes, elements sized with VW and VH will adjust accordingly. They are often used to make web designs look good on different devices and screen sizes.

Answer 4 : Inline, inline-block, and block are three different display properties.
Inline: Inline flow along with text content. They only take as much space as needed for their content. Ex. <span>, <a>, <em>.
Inline-block: Inline-block element same as inline element but inline-block element has width, margin and padding similar to block element. Ex. <img>, <button>, <input>.
Block: Block line elements start on a new line and occupy the entire width available. Block line elements have height, width, margin, padding. Ex. <div>, <p>, <h1>.

Answer 5 : Content Box: In the content-box model, the size of an element is calculated by considering only the content area.
BorderBox: The border-box model includes the content area, padding, and border within the specified width and height values.

Answer 6 : The z-index property in CSS determines the order of positioned elements along the z-axis, allowing the object to keep the object upper and down. It has numeric values with positive, negative and zero.The higher the z-index value, the closer it is to the viewer. Elements with the higher z-index value overlap the lower z-index element. If there are the same z-index of many elements then it will go for position or transform.

Answer 6 : Layout Approach: Grid is a two-dimensional layout system, while Flexbox is a one-dimensional layout system.
Dimensions: Grid allows control over both rows and columns, while Flexbox operates along a single axis..
Alignment: Flexbox provides powerful alignment options along the main and cross axes, while Grid offers more control over the placement and alignment of items within grid cells.
Overlapping Elements: Grid allows elements to overlap within the grid cells, whereas Flexbox does not have built-in support for element overlapping.

Answer 7 : Absolute positioning removes the element from its usual position in the document, allowing you to place it anywhere you want. Relative positioning keeps the element in its regular position within the document flow.
With absolute positioning, you can position the element based on its closest positioned ancestor or the document itself. Relative positioning allows you to shift the element from its original position within the normal flow by specifying how much it should move.
Example link : github repo link

Sticky positioning makes an element behave like it's glued in a certain position as you scroll, sticking to that spot when you reach a specific point on the page. Fixed positioning keeps an element in a fixed position on the screen, regardless of scrolling.
With sticky positioning, the element stays in its normal flow until you scroll past a particular point, at which it becomes fixed in place temporarily. Fixed positioning, however, keeps the element fixed at all times, even when scrolling.
Example link: github repo link

Answer 8 : Github Repo Link

Answer 9 : HTML: Github Repo Link   CSS: Github Repo Link

Answer 10 : Github Repo Link

Answer 11 : Github Repo Link

Answer 12 : Pseudo-classes and pseudo-elements are both selectors in CSS that allow you to target specific elements based on certain conditions or states. They serve different purposes:
Pseudo-classes: Pseudo-classes are used to select elements that are in a particular state or meet specific criteria. They are indicated by a colon (:) followed by the name of the pseudo-class. Examples of common pseudo-classes include `:hover`, `:active`, `:focus`, `:first-child`, and `:nth-child(n)`. Pseudo-classes are used to style elements based on user interaction or the document structure.

Pseudo-elements: Pseudo-elements are used to style a specific part of an element. They are indicated by two colons (::) followed by the name of the pseudo-element. Examples of common pseudo-elements include `::before`, `::after`, `::first-line`, and `::selection`. Pseudo-elements allow you to add content before or after an element, or style specific parts of an element's content.

JavaScript Questions

Answer 1 : Hoisting is a feature in JavaScript that allows variable and function declarations to be recognized and accessible throughout their containing scope, regardless of where they are physically placed in the code. It happens during the compilation phase before the code is executed, and it essentially moves these declarations to the top of their respective scopes. So, even if you declare a variable or function later in the code, you can still use it earlier in the same scope because it gets hoisted to the top.

Answer 2 : Higher order functions in JavaScript are functions that can take another function as an argument or return function as a result. There are some higher order functions in JavaScript including map(), forEach(), filter(), reduce(), sort(). But .map() ans.forEach function is widely used to iterate over arrays. '.map()' function creates a new array and returns the same length of the previous array. It does not change the old array. While the '.forEach()' function modifies values in the same array and does not create a new array.

Answer 3 : .call() and .apply() are used to invoke a function with a specific 'this' value, but they differ in how arguments are passed. .call() accepts arguments individually, while .apply() accepts arguments as an array.
.bind() is used to create a new function with a specific 'this' value and, optionally, predefined arguments. Unlike .call() and .apply(), .bind() does not immediately invoke the function but returns a new function with the bound this value and arguments.
Example : Github Repo Link
In this example, .bind() is used to create a new function greetWithAlice that has anotherPerson as the 'this' value. When greetWithAlice is called with the argument 'Hello', it logs 'Hello, Alice!' to the console.

Answer 4 : Event bubbling and event capturing are two different phases of the event process in javascript.
Event bubbling : In the event bubbling all events occurring in an element should also occur in their parent element.
Example : Github Repo Link
Event capturing: In the event capturing, when an event occurs it will target first their parent element before reaching to target. Flow of the targeting element is outermost to innermost.
Example: Github Repo Link

Answer 5 : Function currying is a technique in JavaScript where a function with multiple arguments is transformed into a lot of functions and each one is taking one argument.
Example : Github Repo Link

Answer 6 : Github Repo Link

Answer 7 : Promises are objects in JavaScript that represent the event completion or failure for a given asynchronous function. They have three different states: pending, fulfilled and rejected. If status is pending that means asynchronous operation is still in progress, if its fulfilled that means operation is completed successfully and if it's rejected that means given operation is failure or error.
Example : Github Repo Link

Answer 8 : In JavaScript, the `this` keyword represents the current situation or the object that the current function or method is working with. It lets you access properties and methods within the current area of code.
Example : Github Repo Link

Answer 9 : The event loop in JavaScript manages the execution flow of asynchronous operations. It ensures that tasks are processed in a specific order.

The call stack is a data structure that keeps track of function calls. It allows the program to know which functions are currently being executed.

The callback queue, or task queue, holds callback functions for asynchronous operations. These callbacks are executed when their corresponding tasks are completed.

The micro task queue holds higher-priority tasks that need to be executed promptly, such as promises and certain browser APIs. Micro tasks are processed before regular callback functions.

Answer 11 : Closures are a programming concept that allows a function to remember and access its lexical scope even when the function is executed outside of that scope.

Use cases of closures :
1. Data Privacy: Closures can be used to create private variables and functions. By defining variables within a closure, they are inaccessible from outside the closure, providing data privacy and encapsulation.

2. Callbacks: Closures are commonly used in asynchronous programming and event handling. By capturing variables in a closure, you can create callbacks that retain access to those variables when the callback is invoked later.

Answer 12 : Github Repo Link

React Questions

Answer 1 : React is a JavaScript library developed by Facebook for building user interfaces. It utilizes a component-based approach, where UI elements are broken down into reusable components, allowing for modular and maintainable code.

Answer 2 : The Virtual DOM in React is a virtual representation of the actual DOM. It is a lightweight copy of the DOM tree that React uses to perform efficient updates. There are some advantages of DOM including performance, efficiency, platform compatibility and main advantage is reusability. You can reuse any component whenever you need. So it's also easily readable and helps developers to find mistakes speedily.

Answer 3 : The lifecycle of React components consists of three main phases: mounting, updating, and unmounting. In the mounting phase the component is being initialized and inserted into the DOM. In the Updating phase components props or state is changed and in the Unmounting component is being removed from the DOM.

Answer 4 : Functional components and class components are two ways of defining components in React:
Syntax: Functional components are defined as JavaScript functions, while class components are defined as JavaScript classes that extend the `React.Component` base class.

Code complexity: Functional components have a simpler method to code than class components.

Answer 5 : Hooks in React are functions that enable functional components to have state and utilize other React features. They cannot be directly used in class components. There are some commonly used hooks including useState, useEffect, useContext, useReducer, useCallback and more.

Answer 6 : Lifecycle methods in React are special methods that are called at specific stages during the lifespan of a component. They allow developers to hook into different points in a component's existence and perform specific actions or updates.
Advantages of lifecycle methods include:

Initialization and setup: Lifecycle methods like `constructor()` and `componentDidMount()` allow you to initialize component state, bind event handlers, and perform any necessary setup tasks.

Handling updates: Lifecycle methods like `componentDidUpdate()` allow you to respond to changes in component props or state.

Cleanup and resource management: Lifecycle methods like `componentWillUnmount()` helps prevent memory leaks and ensures that any necessary cleanup tasks are performed when the component is removed from the DOM.

Answer 7 : The useState hook is a built-in hook in React that allows functional components to manage state. It provides a way to declare and update state variables within a functional component. Here some advantages of the useState hook :
The useState hook simplifies the process of managing state within functional components. It is easy to use and It follows a simple syntax where you declare a state variable and a function to update that variable. Functional components can use the useState hook multiple times to manage multiple state variables independently.

Answer 8 : The useEffect hook in React is used to perform side effects in functional components. Side effects can include data fetching, subscribing to events, manipulating the DOM, or any action that needs to be executed outside the regular component rendering process. Here some advantages of the useEffect hook:
The useEffect hook consolidates all side effects into a single hook. It eliminates the need to spread side effect logic across different lifecycle methods in class components, making it easier to read, understand, and maintain.
The useEffect hook combines the functionality of multiple lifecycle methods, such as componentDidMount, componentDidUpdate, and componentWillUnmount, into a single hook. This reduces the complexity of managing component side effects and provides a consistent and unified approach.
The useEffect hook can be combined with other hooks, such as useState or useContext, to create powerful and flexible functionality within functional components. This allows for easy composition and reusability of component logic.

Answer 10 : `useReducer` is a React hook that allows you to manage state and state transitions in a predictable and efficient way. It is especially useful for managing complex state logic and provides advantages such as centralized state management, easier debugging, and better code organization.

Answer 11 : Github Repo Link

Answer 12 : Github Repo Link

Answer 13 : Github Repo Link

Answer 14 : Github Repo Link

Answer 15 : Prop drilling is the practice of passing data through multiple components in a hierarchy to reach a specific component that needs access to that data. To avoid it, you can use the Context API or state management libraries like Redux to share data across components without the need for prop drilling.

Answer 16 : Github Repo Link

Express Questions

Answer 1 : Github Repo Link

Answer 2 : Middleware is a software component that acts as a bridge between different applications, systems, or components. It facilitates communication and data exchange by intercepting requests and responses, allowing for additional processing or modifications to be performed before they reach their destination. In web development, middleware is often used to add functionalities to the server-side application, such as authentication, logging, or request/response manipulation.
Example : Github Repo Link

Answer 3 : Github Repo Link

Answer 4 : Authentication and authorization are two fundamental concepts in computer security and access control. Here's the difference between the two:

Authentication: Authentication refers to the process of verifying the identity of an individual, system, or entity. It ensures that the claimed identity is valid and reliable. The goal of authentication is to establish trust and confirm that the user or system is who they claim to be.

Authorization: On the other hand, deals with granting or denying access to resources or actions based on the authenticated identity. Once a user or system is successfully authenticated, authorization determines what actions or resources they are allowed to access.

Answer 5 : CommonJS (CJS) is a way for JavaScript code in server-side environments to be organized into separate modules, allowing them to be used and shared easily. EJS (Embedded JavaScript) is a tool that lets developers create web pages with dynamic content by combining HTML templates and JavaScript code.

Answer 6 : JWT stands for JSON Web Token, and it is a compact, URL-safe means of representing claims to be transferred between two parties. It is commonly used for authentication and authorization in web applications.
Example : Github Repo Link

Answer 7 : In order to secure a user's password before storing it into a database we need to perform at least these two actions, Hash Password and Salt the Password.
Hash the Password : Use a strong cryptographic hashing algorithm, such as bcrypt. Apply the hashing algorithm to the user's password, which will convert it into a fixed-length string of characters.
Salt the Password: Add a unique and randomly generated salt value to the password before hashing it. The salt value adds additional complexity and uniqueness to each hashed password.

Answer 8 : The event loop in Node.js is like a manager that keeps track of tasks and makes sure everything runs smoothly. It handles tasks that take time, like reading files or making requests, without stopping the program from doing other things. It listens for events and runs the appropriate code when an event occurs. It keeps repeating this process until there are no more tasks left to do.

Answer 9 : Github Repo Link