Chapter 01: Inception- Namaste React

Q1: What is Emmet?

Ans: Emmet is set of tools and plugins for text editors that can greatly speed up the writing of HTML and CSS code. It provides a shorthand syntax for writing HTML and CSS, allowing developers to write a lot of code very quickly. This syntax is transformed into the corresponding HTML and CSS code by the Emmet tools.

For example, if you type `**ul>li\*5**` in an HTML file and press the Emmet expansion key (often the TAB key), Emmet will expand that shorthand syntax into a full HTML list with five items.

Code:

<ul>

<li></li>

<li></li>

<li></li>

<li></li>

<li></li>

</ul>

Emmet is supported by wide range of text editors, including popular once like Sublime Text, Visual Studio Code, Atom and more. It is widely used by front-end developers and designer to save time and improve their productivity when writing HTML and CSS.

Q2: Difference between a Library and Framework?

Ans: A library and a framework are both collections of code that can be used to perform a specific problems, but there is a key difference between the two.

A library is a collection of functions or classes that you can call directly from your own code to perform specific tasks. A library is essentially a set of tools that you can use to solve specific problems. When you use a library, you are in control of the application flow and you call the library functions as needed to perform specific task.

A framework, on the other hand, is more opinionated set of tools and conventions for building applications. When you use a framework, you work within the framework structure and follow it’s conventions to build your application. The framework provides you with a set of pre-build components and the over all application structure, and you fill in the details of your specific implementation. The framework controls the application flow, and you respond to events and callbacks triggered by the framework.

In other words, a library gives you the tool you want, but it’s up to you to put the pieces together. A framework provides a pre-made structure for building applications, and you fill in the details.

So, the main difference between a library and a framework is that a library is a collection of tools that you can use to build what you want, whereas a framework is a pre-made structure for building applications that you need to work within.

Q3: What is CDN? Why do we use it?

Ans: A CDN or Content Delivery Network is network of servers located in different geographic locations that work together to deliver web content to users. The goal of CDN is to reduce the latency and improve the performance of serving content to users who are far away from the origin server.

When a user request content from a website that is served by a CDN, the CDN will direct the request to the server that is closest to the user’s location. This helps to reduce the latency of the request and ensure that the content is delivered as quickly as possible.

There are several benefits to using a CDN including:

* Improved performance: By serving content from server that are close to the user, a CDN can significantly improve the speed and the performance of delivering content to users.
* Reduced bandwidth costs: By serving content from multiple users, a CDN can help to distribute the load and reduce the bandwidth cost for the websites owner.
* Increase availability: a CDN can help to ensure that the content is still available even if the origin server is down or experiencing a high traffic.
* Improved security: a CDN can help to protect against some types of attacks, such as distributed-denial-of-service (DDoS) attacks, by absorbing the attack traffic at the edge of the network.

Overall, a CDN is valuable tool for websites that need to deliver high-quality, fast-loading content to use across the world.

Q4: Why is React is known as React?

Ans: React is known as React because it was created by Jordan Walke, a software engineer at Facebook, it was initially used to build the user interface of Facebook’s newsfeed in 2011. The name “React” is short for “responsive” and was chosen because one of the main goals of the library was to provide a way to build user interfaces that are fast and responsive to user interactions.

React is based on the concept of “reactive updates”, which means that it can update the user interface efficiently in response to changes in data. This is a key advantage of React over other JavaScript libraries for building user interfaces, as it allows for building complex and dynamic user interfaces with ease and performance.

Over the years, React has become one of the most popular JavaScript libraries for building user interfaces, and it has been adopted by many companies and organisations for building their web applications. The name “React” has become synonymous with efficient and fast user interfaces, and it is widely recognized in the web development community.

Q5: What is crossorigin in script tag?

Ans: The crossorigin attribute sets the mode of the request to an HTTP CORS Request.

Web pages often make requests to load resources on other servers. Here is where CORS comes in.

A cross-origin request is a request for a resource (e.g. style sheets, iframes, images, fonts, or scripts) from another domain.

Q6: What is the difference between React and ReactDOM?

Ans: React and ReactDOM are two different libraries that are both maintained by Facebook and used for building web applications.

React is a core library for building user interfaces, while ReactDOM is a library that provides specific implementations of React for the web. To use React in web application, you typically need to include both the React library and ReactDOM library in your project.

Q7: What is difference between react.development.js and react.production.js files via CDN?

Ans: The ‘react.development.js’ and ‘react.production.js’ files are different builds of the react library that are optimized for different environments.

**‘react.development.js’:** is the development version of the React library. It is designed for use during the development process and includes features that help with debugging and development, such as error messages and extra warnings. This version of the library is typically larger in size and slower in performance, but provides more information and feedback to help with development.

**‘react.production.js’:** is the production version of the React library. It is optimized for performance and is smaller in size, making it faster to load and more efficient. This version of the library is designed for use in production environment, where the goal is to provide the best user experience with the fastest load time and the most efficient use of resources. This version of library includes only the code necessary to run the application and dose not include any debugging or development-specific features.

Q8: What is async and defer?

Ans: async and defer are JavaScript attributes, used along with the script tag to load the external script into our HTML page efficiently.

Normal scenario: while parsing an HTML page , when it encounters to script tag it stops HTML parsing and fetch the script and execute the script. After executing the script, it completes the remaining HTML parsing. Creating time lag.

Async: async attribute fetches the script asynchronously along with HTML parsing, and executes immediately as soon as script is fetched crating comparatively less time lag.

Async attribute does not guarantee the order of the execution.

Defer: defer attribute fetches the script simultaneously and executes the script only after the HTML is fully parsed, crating zero time lag.

Defer attribute does guarantee the order of the execution.