Assignment 1: Inception

1.What is Emmet?

Emmet is a free add-on for your text editor that allows you to type shortcuts that are then expanded into full pieces of code.
Emmet comes as a built-in feature in VS Code. Emmet greatly improves and speeds up workflow, saving us the stress of having to manually type out the code in full.
Emmet uses different abbreviations and short expressions depending on what's passed, and then dynamically converts the abbreviations into the full code. Emmet is mostly used for HTML, XML, and CSS, but it can also be used with programming languages.

2. Difference between Library and Framework.

Framework	Library
Provides ready to use tools, standards, templates and policies for fast application development.	Provides reusable function for our code.
The framework controls calling of libraries for our code.	Our code controls when and where to call a library.
To leverage the benefit of a framework, a fresh application can be developed following the framework's guidelines.	Library can be added to augment the features of an existing application.
Easy to create and deploy an application.	Facilitates program binding.
Helps us to develop a software application quickie and itself is an application.	Helps us to reuse a software function.
Intent of the framework is to reduce the complexity of the software development process.	Intent of a library is to provide reusable software functionality.

3. What is CDN and why do we use it?

A content delivery network (CDN) is a geographically distributed group of servers that caches content close to end users. A CDN allows for the quick transfer of assets needed for loading Internet content, including HTML pages, JavaScript files, stylesheets, images, and videos. The popularity of CDN services continues to grow, and today the majority of web traffic is served through CDNs, including traffic from major sites like Facebook, Netflix, and Amazon.

4. Why is React known as React?

React is called "React" because of its core feature, which is its ability to "react" or respond dynamically to changes in data.

5. What is cross-origin in the script tag?

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.

CORS is used to manage cross-origin requests.

CORS: It stands for cross-origin resource sharing. It is a mechanism by which one webpage requests to another domain for fetching out the resource like audio, video, script, etc. from the third party server without leaking their credentials information.

Values: This attribute contains two values which are given below –

anonymous: It has a default value. It defines a CORS request which will be sent without passing the credential information.
use-credentials: A cross-origin request will be sent with credentials, cookies, and certificate.

6. What is the difference between React and ReactDOM?

React is a JavaScript library for building User Interfaces and ReactDOM is the JavaScript library that allows React to interact with the DOM(Document Object Model).

7. What is the difference between react.development.js and react.production.js files via CDN?

react.development.js is a development version of the React library. It contains extra warnings and debugging information that can be helpful during development and testing, but these features also make the file size larger. This file is not optimized for performance and is therefore slower compared to the production version.

On the other hand, **react.production.js** is an optimized and minified version of the React library, designed for use in production environments. It has been stripped of unnecessary code and debugging information, resulting in a smaller file size and better performance. This file does not include any debugging information or warnings, making it faster but less helpful during development.

8. What are async and defer?

In web development, "async" and "defer" are attributes that can be added to the <script> tag to control how the browser loads and executes JavaScript files.
"**async**" stands for asynchronous, and when added to a script tag, it tells the browser to continue loading the rest of the page while the script is being fetched from the server. This means that the script is downloaded in the background, and as soon as it's available, it will be executed. This can improve the page loading speed because the browser doesn't have to wait for the script to be downloaded before continuing to load other resources.

"**defer**" also tells the browser to continue loading the page while the script is being fetched, but it ensures that the script is only executed after the page has finished parsing. This can be useful if the script depends on other resources on the page to be loaded, such as images or stylesheets, because it ensures that the script won't be executed until those resources are available.

In summary, "async" loads and executes the script asynchronously, while "defer" loads the script asynchronously but defers its execution until after the page has finished parsing.

NOTE: If there are multiple javascript files, then async doesn’t guarantee the order of execution in which the files have been mentioned, but defer maintains the order of execution of the files in the order in which they are provided.