NAMASTE REACT COURSE ASSIGNMENT 1

● What is Emmet?

This is a Virtual Studio (VS) Code extension, which helps developer's experience by creating suggestions, whenever letters and shortcuts are typed inside the VSCode. Hence saving time, with minimal errors. In a nutshell, it can be called a productivity booster, because it boosts the productivity of how we write code.

● Difference between a Library and Framework?

Library: This is a set of codes, that were previously written, which can be called upon when writing your own code. They are reusable. Take for instance, we have a project, where a carpenter needs to fabricate a table, then he has a toolbox, which houses all the tools required for this task. The toolbox in this case serves as a library, because he can easily callup any of the tools whenever the need arises; Remember, the carpenter never made these tools himself, but they are always available to him. Furthermore, these tools (library) can be used for a variety of tasks, example, you used the tools in building a table, Mr. B can also use the same tools in crafting out a boat. When using a library, the developers have a full control of the code progression.

- Your code calls the library.

Framework: this is a codebase/foundation, which directs the developer on the steps to be taken during development (codes are built upon this foundation), hence giving them minimal control of the progression of the code. For instance, we have a group of carpenters, who make similar patterns of tables. Other than go through this rigorous process daily, a framework in the place of a manufacturing warehouse can be erected, where the carpentry activity moves in in a line process from Start to End Hence, giving the carpenters little control of how the project should be achieved. Yes, at various designated points of this task, the carpenter has a slight control of how a few things could be, such as: the length, angle etc. of the wood. But the big question here is that, can this same factory build a ship? Obviously, NO... but can only build a variety of tables with various sizes. Basically, the framework provides patterns and make it easy to operate within the confines of those patterns. In addition, the framework is less flexible

You configure the framework and supply it with the required data, needed for it to call your code at the right times.

• What is CDN?

A content delivery network (CDN) is a group of servers spread across different geographical locations worldwide to enable the quick delivery of a website's content. It is also known as a content distribution network.

- **Why do we use it?**

It is used because when a user accesses a website with a CDN, the browser connects to and requests site content from one of the edge servers (The chosen server is usually the one closest to the end-user to reduce the time delay. The usage of multiple caching servers distributes traffic and prevents server overload). The edge server will then forward the request to the origin server. After getting the data from the origin server, the edge server delivers it to the end-user and caches the files locally.

● **Why is React known as React?**

There is no clear explanation as to why the Facebook team named its open-source library, however, there are indications that since the React is used in creating dynamic and interactive interfaces, whenever users click (interact) with the application, the application should be able to React to this event.

● **What is crossorigin in script tag?**

This attribute helps you control whether you would like to send identifying information from your browser to some other source.

We might have to go to another server (src) to fetch data for the functionality of or page; this is known as Cross Origin Request Sharing (CORS). In the course of this request, the crossorigin attribute, is what indicates to the server the kind of user credentials available for this sharing process.

- **What is difference between React and ReactDOM?**
    - **DOM:**

    This is known as Document Object Model; which depicts how a web browser represents a web page internally (html document is represented in a tree structure)
    - **React:**

    React is an open-source JavaScript library used for the development of UI (User Interface) on web application. Its fully component base.
    - **ReactDOM:**

    This is the glue connecting React to the DOM, it is a library used in rendering React applications (components) in the browser.

- **What is difference between react.development.js and react.production.js files in CDN?**

    It is worthy of note, that for the running of React.js projects, there are two modes available which include: development React.js and build production React.js.

    During the development phase, we will be running our code locally using the development mode where React provides us with many helpful warnings and tools for easily detecting and fixing problems in our application code and eliminating potential bugs. However, these extra codes increase bundle size and hence, a slower running

app. While working on the app locally, this slowdown may be acceptable. However, during deployment, this is not something we can afford. According to a Google study, 53% of users will leave a site if it takes more than 3 seconds to load. Consequently, we need to speed up our application at all costs, and this is where the production mode comes into the picture, in order to minify your code, optimize assets, and produce lighter weight source maps. Also, the warning messages and other features present in development mode for debugging will be suppressed.

- **What is async and defer?**
  Async and defer are Boolean attribute which are used along with script tags, to load external scripts efficiently into our web page.
  N/B: when we load a web page, there are two things that happen:
  a. **HTML parsing:** basically, the browser parses the HTML into a DOM tree
  b. **Loading of the script:** This contains two parts,
     i. fetching the script from the network and
     ii. Loading/executes the scripts line-by-line.

  Normal Script tag: <script src=" "/>

  In this scenario, when the webpage is loaded, the HTML is parsed, when the normal script tag is noticed, the parsing is paused. The fetching of script begins, after which the script execution commences. At the end of the execution, the HTML parsing continues until it gets to the final line.

  Script tag with Async attribute: <script async src=" "/>

  In this scenario, when the webpage is loaded, the HTML is parsed, when the script tag with async attribute is noticed, the fetching of script begins and runs simultaneously with the HTML parsing. Immediately the script data has been fetched from the network, the HTML parsing is paused. At this point, the execution of the script commences. At the end of the execution, the HTML parsing continues until it gets to the final line.

  Script tag with defer attribute: <script defer src=" "/>

  In this case, the HTML parsing runs, immediately the defer script is detected, the script fetching happens simultaneously with the parsing. The script is only executed at the end of the parsing.

  - **When do we use them?**

  The Async attribute doesn't maintain the order of execution of the script, however, the defer attribute does. When you have scripts that depend on the previous scripts or follow a pattern of execution, then the best bet is the defer attribute. Whereas, if we have scripts independent of the normal code, then the async attribute can be used. Otherwise, always use a defer attribute because it maintains the order of execution of script.