**Namaste React**

**Lesson 1: Inception**

1. Browsers do not understand what React JS is.
2. For browsers to understand react we will have to add react to our project.
3. One way of adding React JS to the code is by **Using the CDN (Content Delivery Network) link.**
   1. When we added the CDN links in our code we imported React JS into our project.
   2. So, whatever code we write in React JS our browser will be able to understand React.
   3. <https://legacy.reactjs.org/docs/cdn-links.html>
4. When we look at the script tags and open the src link we see that all the code that is written is plain JavaScript code.
5. At the end of the day, React JS is just JavaScript code.
6. This code is written by some Facebook developers.
7. When we now go to the developer console and type React, we will have access to all the superpowers of React.
8. We have imported two CDN links, the first one is React, which is the core of React, and the second one is React DOM, this is the library that we need to modify the DOM (Document Object Model).
9. Why do we have two different files instead of one? This is because react works with different devices such as Laptops, Mobile phones (React Native), iPad, etc. There are different functions that are used in mobiles and laptops so there are different files.
10. React DOM is like a bridge between the code and the browser.
11. In JavaScript, we use something called **document.createElementById** but in React we use something called React.createElement
12. This createElement takes three arguments.
    1. The tag that we want to create. For example, h1, h2, h3, div, p etc.
    2. Object – These are attributes to a tag. For example: id, class, etc.
    3. Content to put inside the tag.
13. Things are a bit different in React JS. To inject the heading in our application we will have to have a root element. All the DOM manipulation is done here.
14. Creating an element is the responsibility of react.development.js. Creating a root and rendering some data in the element is the job of the react DOM library. We will use reactDOM.createRoot(). It will take the root element id as the parameter.
15. Const heading = React.createElement("h1", {}, "Hello from react JS");
16. const root = ReactDOM.createRoot(document.getElementById("root"));
17. root.render(heading);
18. React element is nothing but an object.
19. We will have complex or nested structure in our application. To create this structure, we will use the createElement and do it for all the parent child required.
20. Writing such complex structures is difficult and we have something called JSX.
21. Order of the script files matters in ReactJS. So, if we place app.js above the React JS CDN files then our application will break saying that react is not defined.
22. When we do the root.render and if there is something that already exists in the root tag then the data in the root tag will be replaced by the latest code that we want to render.
23. React is a library and it can be used in a particular section of the whole application.
24. We can use it only for header, footer, or main body.

**Extra Points**

1. The most important operation that happens in a browser is the DOM manipulation.
2. What is CDN?
   1. CDN stands for Content Delivery Network. It is a distributed network of servers located in different geographic locations that work together to deliver internet content, such as web pages, images, videos, and other resources, to users in a faster and more efficient manner.
3. Why do we use CDN?
   1. Improved website performance
   2. Global Reach
   3. Traffic Offloading
   4. Load Balancing
   5. Caching
   6. Content Optimization
   7. Security and DDoS protection
   8. Scalability
   9. Content Availability
4. How does CDN work?
   1. Content replication - When a website or online service uses a CDN, it duplicates its content (like HTML files, images, videos, etc.) and stores copies on multiple servers in various locations around the world.
   2. Geographic Distribution - These servers, often referred to as "edge servers" or "nodes," are strategically placed in data centers across different cities or countries. This geographical distribution helps reduce the physical distance between a user and the server, which can significantly improve load times.
   3. Request Routing - When a user tries to access a piece of content, the request is automatically routed to the nearest CDN server rather than the original server where the website is hosted.
   4. Content Delivery - The CDN server then fulfills the request by serving the content directly to the user. This leads to faster loading times and better performance because the data doesn't have to travel as far.
   5. Load balancing & Scalability - CDNs are designed to handle many simultaneous requests. They use techniques like load balancing to distribute traffic evenly among the servers.
   6. Caching - CDNs employ caching techniques. When a CDN server receives a request for a specific piece of content, it checks if it already has a copy. If it does, it serves the cached copy, which is faster than fetching it from the original server.
5. What is the crossorigin keyword in script tag?
   1. The crossorigin attribute sets the mode of the request to an HTTP CORS request.
   2. Web pages often make requests to load resources on other servers. Here is where CORS comes into picture.
   3. A cross-origin request is a request for a resource from another domain.