**<https://github.com/JhanaR/my_learnReact.git>**

**If need refer below links too**[**https://github.com/JhanaR/CloneMalini.git**](https://github.com/JhanaR/CloneMalini.git)[**https://github.com/JhanaR/UiClass\_React.js-Prereq.git**](https://github.com/JhanaR/UiClass_React.js-Prereq.git)[**https://github.com/JhanaR/React\_MaliniClass.git**](https://github.com/JhanaR/React_MaliniClass.git)

**REACT JS**

React is a popular JavaScript library used to build user interfaces, especially for single-page applications (SPAs).

It follows a component-based architecture, where the UI is broken into small, reusable pieces called components.

React uses a concept called the Virtual DOM to optimize rendering and updates to the UI. This makes the app faster and more efficient.

By updating only the necessary parts of the DOM, React helps create high-performance single-page applications, where the page doesn't fully reload on each interaction.

A **CDN (Content Delivery Network)** is a globally distributed network of servers that deliver static content (like JS, CSS, images) from a location **closest to the user**.

In React apps, CDNs are used to serve:

* **React libraries** (like React and ReactDOM)
* **Static assets** from production builds (e.g., main.js, styles.css)

This improves performance, reduces loading time, and helps handle more users efficiently.

Browsers **only understand HTML, CSS, and JavaScript(through JS engine)**.

**BABLE:** React code is usually written in **JSX,** which looks like HTML inside JavaScript. It must be **transpired** (by tools like Babel) into regular JavaScript before browsers can execute it.

**(WEB PACK)BUNDLE our code:** When we run **“npm run build”**, Webpack compiles and merges all our components, assets, and dependencies into a single optimized folder that can be deployed to a live server.

The public/ folder is meant only for static files served as-is. React components and logic should be placed in the src/ folder so Webpack can process and optimize them during the build process.

Index.html under Public folder:

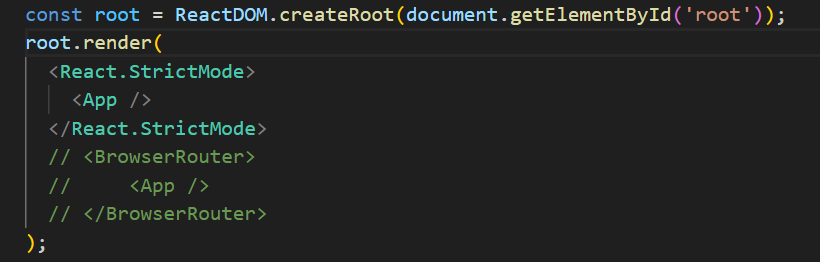
In the index.html file (usually located inside the public/ folder), there's a 

This div acts as a **container** where the entire React app will be injected.

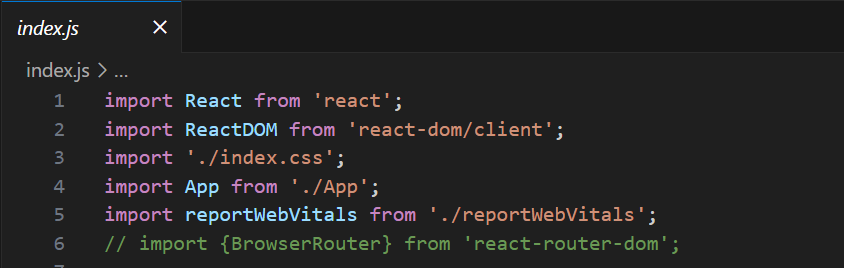
Index.js:

In the index.js (or main.js) file located in the src/ folder, we:

* Import the React component (typically App)
* Create a **reference to the root DOM node**
* Render the App component into the root using ReactDOM.
  + Actual logic will go here.
  + React.StrictMode will warn us if we don’t follow the rules properly and its only present in development environment. (To write efficient code)



Attaching a displaying element called App(component).



**'react' , ‘react-dom/client’** is a **package** used to build user interfaces in a React application.

It is installed via npm (Node Package Manager) and resides in the node\_modules directory.

If it is not already available in your project, you can install it using: “npm install”.

The concept of the **Virtual DOM** is implemented in **React** to optimize UI rendering and enhance performance.  
Instead of updating the entire DOM, **only the components with changes** are updated in the real DOM — making the process efficient and fast.

The package 'react-dom/client' (introduced in React 18) is specifically designed for **web applications**.  
It provides the createRoot() API, which helps **connect the Virtual DOM to the actual DOM**, enabling advanced features like **concurrent rendering**.

**‘./index.css’ , './App' , './reportWebVitals'** are **files(module)** in application.

**'./reportWebVitals'**

Web Applications: React, React-DOM

Mobile-Applications: React, React-native

**NOTE: A function with returning jsx syntax is a component.**

In **React**, you must **return only one parent element** from a component. That’s why you need to wrap multiple JSX elements (like your <h1>, <h2>, and <h3>) inside a **single wrapper** — either a **<div> or a <Fragment>**.

-> Creating a Component

To call a function as a component it should return JSX

React Fragments => save the memory and space of dom

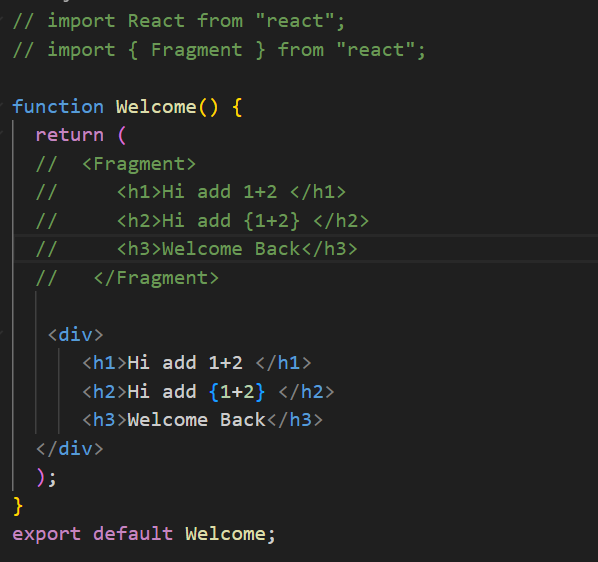
1) import React from 'react'

<React.Fragment> </React.Fragment>

2) import {Fragment} from 'react '

<Fragment> </Fragment>

3) <> </>



Dependencies => The project completely depends on this part,

react and react dom is a dependency.

Dev Dependencies => developer wants some flexibility (only for dev process)

indentation , some plugins are required.

REACT Props:

Props stand for properties they are equal to function parameters.