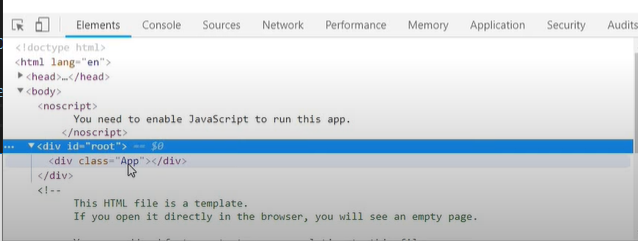
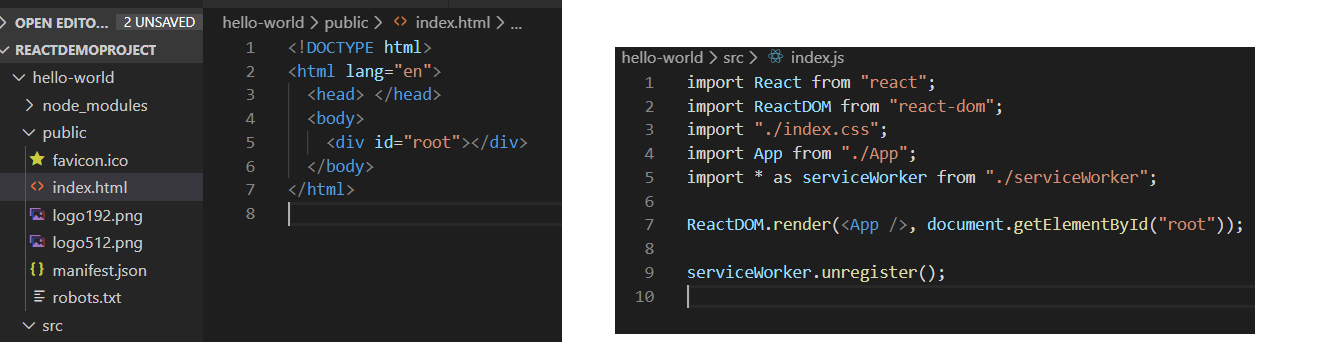
**Chapter-14**

**Portals:**

* React portals provides way to render children into a DOM node that exist outside the DOM hierarchy of the parent component
* When we inspect and check the we see every component in react falls under **root** element i.e. a div tag with id = ‘root’



This happen because to this



* But react portals provides ability to break out of this DOM tree, so we can render a component to a DOM node which is not under the **root** element

If we are creating an app to render something on DOM it’ll automatically be render inside root element

Hello-world🡪src🡪PortalDemo.js

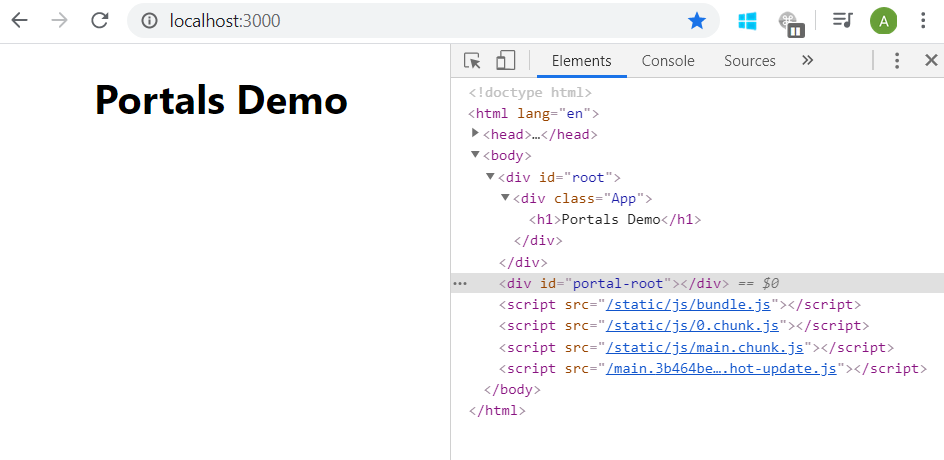
import React from "react";

function PortalDemo() {

  return <h1>Portals Demo</h1>;

}

export default PortalDemo;



But if we want it to be render inside some other element (say ‘portal-root’ element) then following steps need to followed.

1. Create a div tag with desired id (say ‘portal-root’) in index.html

Hello-world🡪public🡪index.html

<!DOCTYPE html>

<html lang="en">

  <head> </head>

  <body>

    <div id="root"></div>

    <div id="portal-root"></div>

  </body>

</html>

1. Create a class or functional component where you want to use it.

* Import ReactDOM from ‘react-dom’
* return **ReactDOM.createPortal()** method which accepts 2 parameters , 1st is what you want to return to DOM and 2nd is id of element inside which u want to render

Hello-world🡪src🡪PortalDemo.js

import React from "react";

import ReactDOM from "react-dom";

function PortalDemo() {

  return ReactDOM.createPortal(

    <h1>Portals Demo</h1>,

    document.getElementById("portal-root")

  );

}

export default PortalDemo;

