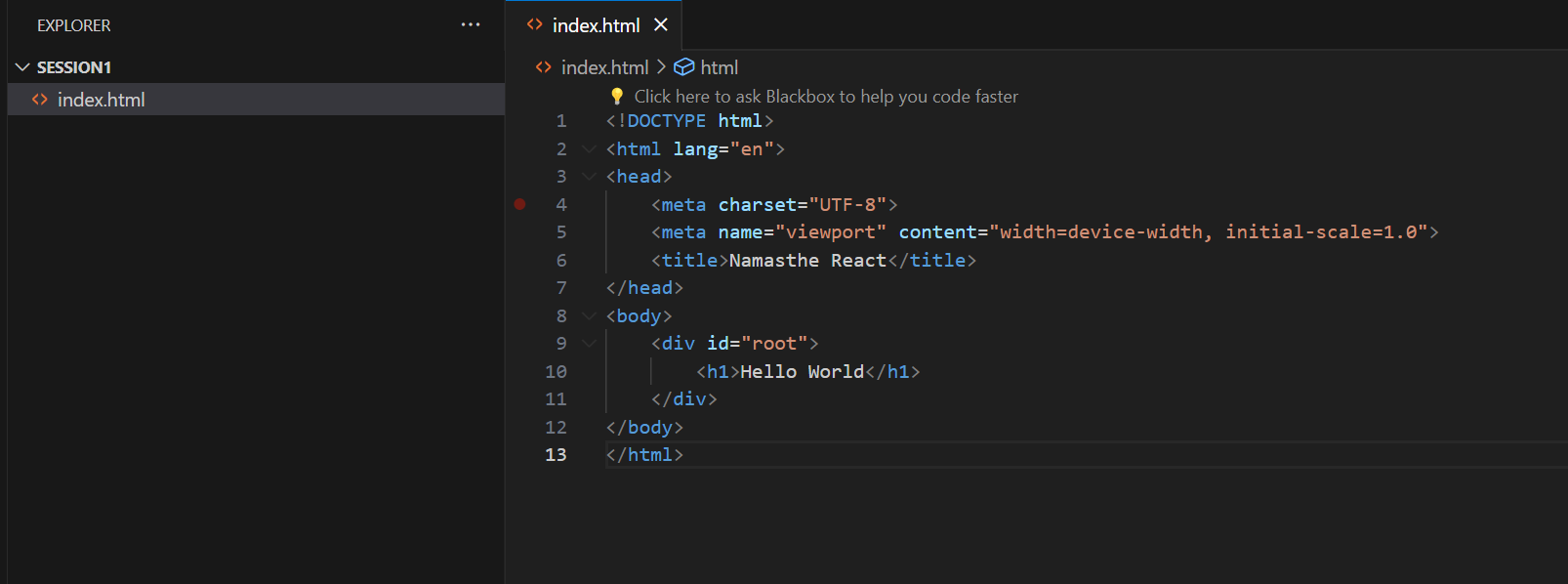
session -1 (Inception)

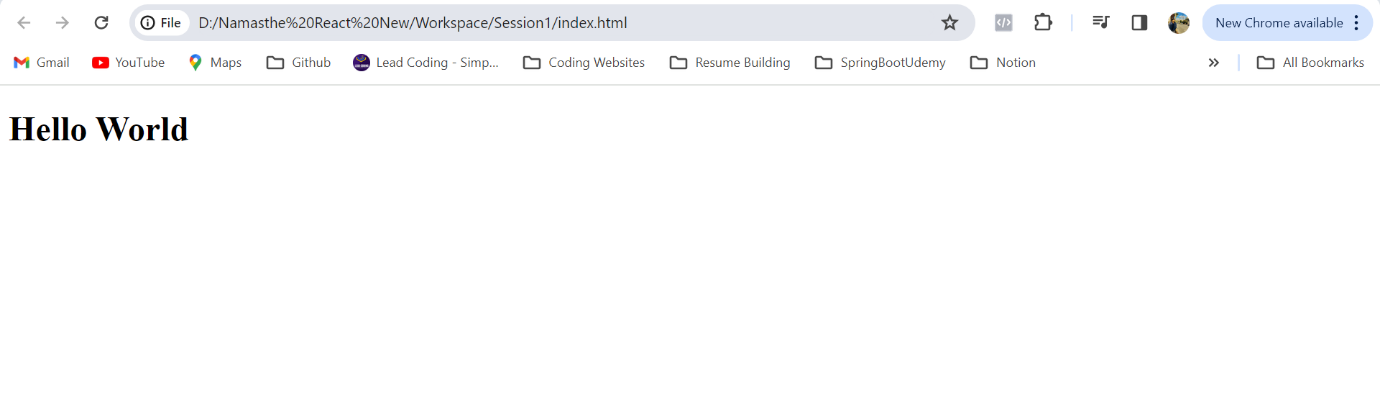
**Create one simple index.html**

Note : We need to install one plugin called emmet in VS code

We are just creating a sample Hello World page in html As below



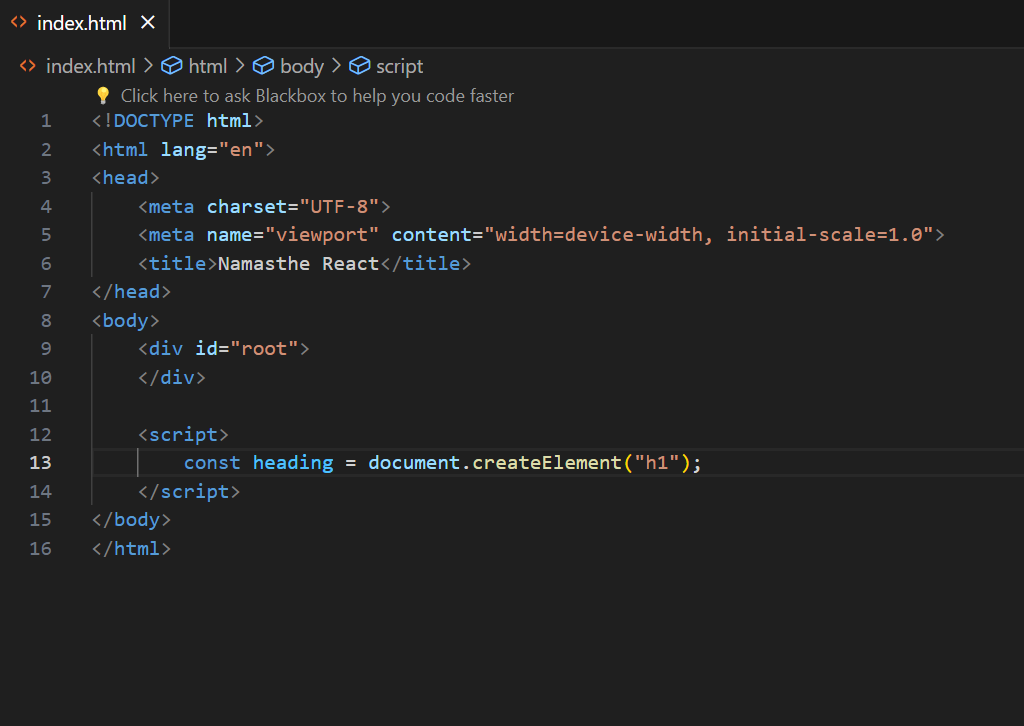
Output :-



Now we will write Hello World Program using Javascript now, previously as above we have done that using only plain HTML

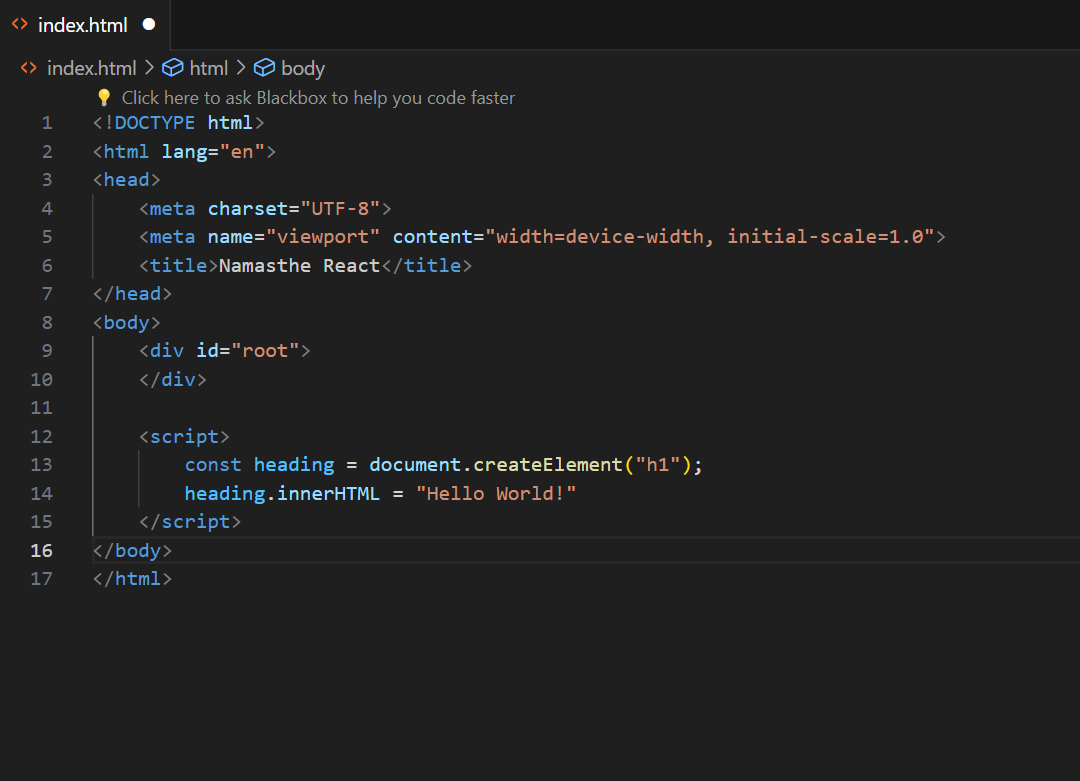
We will just inject that Heading in to the root using Javascript →

We will use script tag for writing Javascript as below and we will use document.createElement API



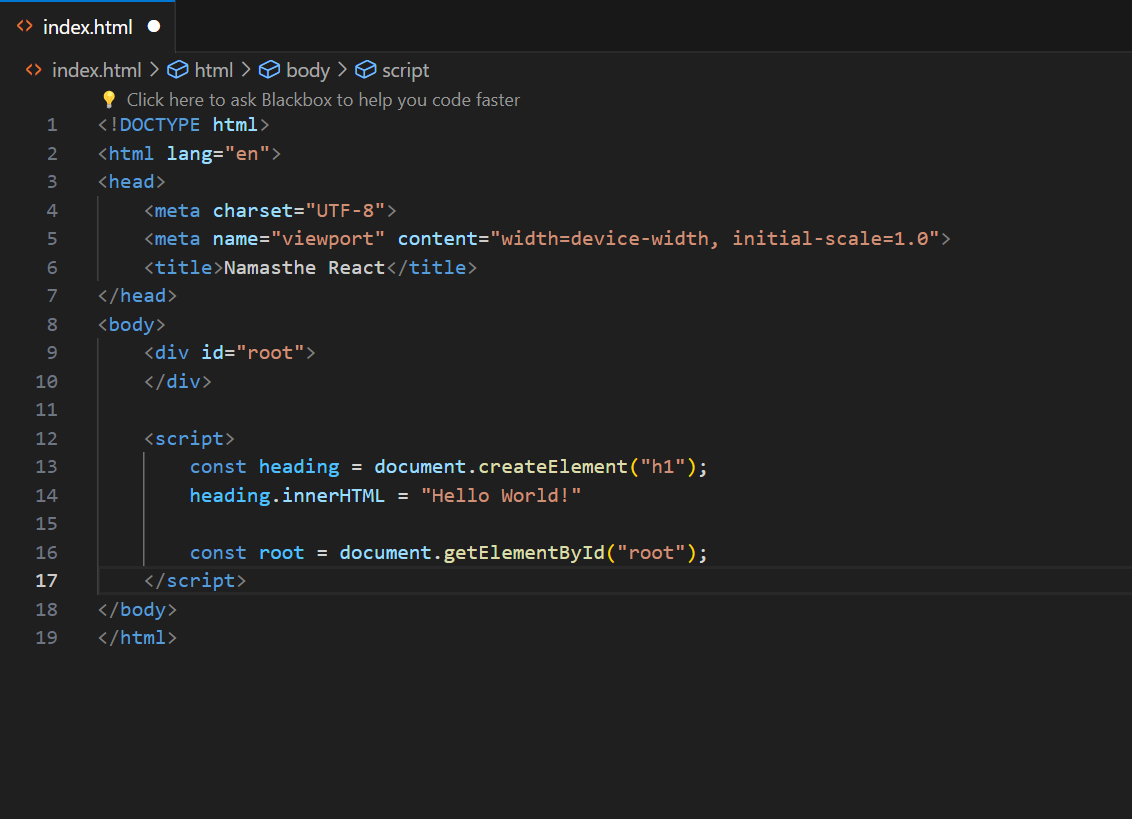
And we are just capturing h1 in to a const variable heading as above

and we are using innerHTML to add some HTML content inside heading as below

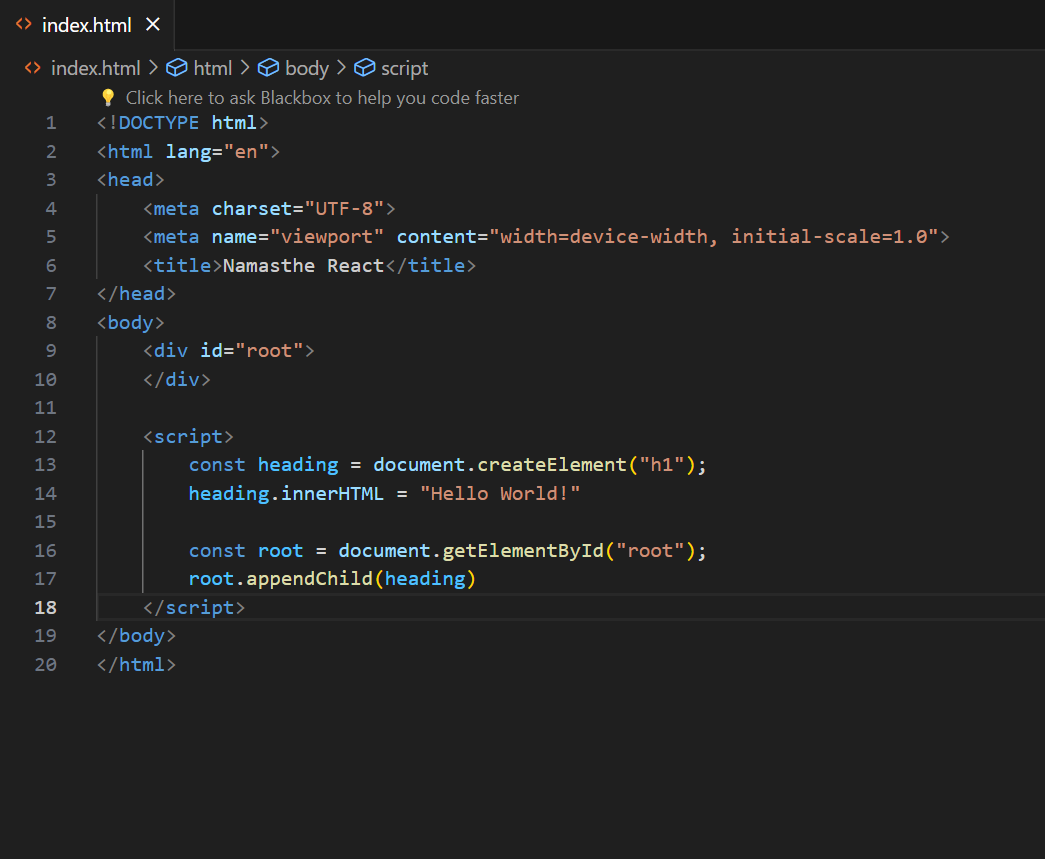


Now we have created Heading but how do we put that inside our root ?

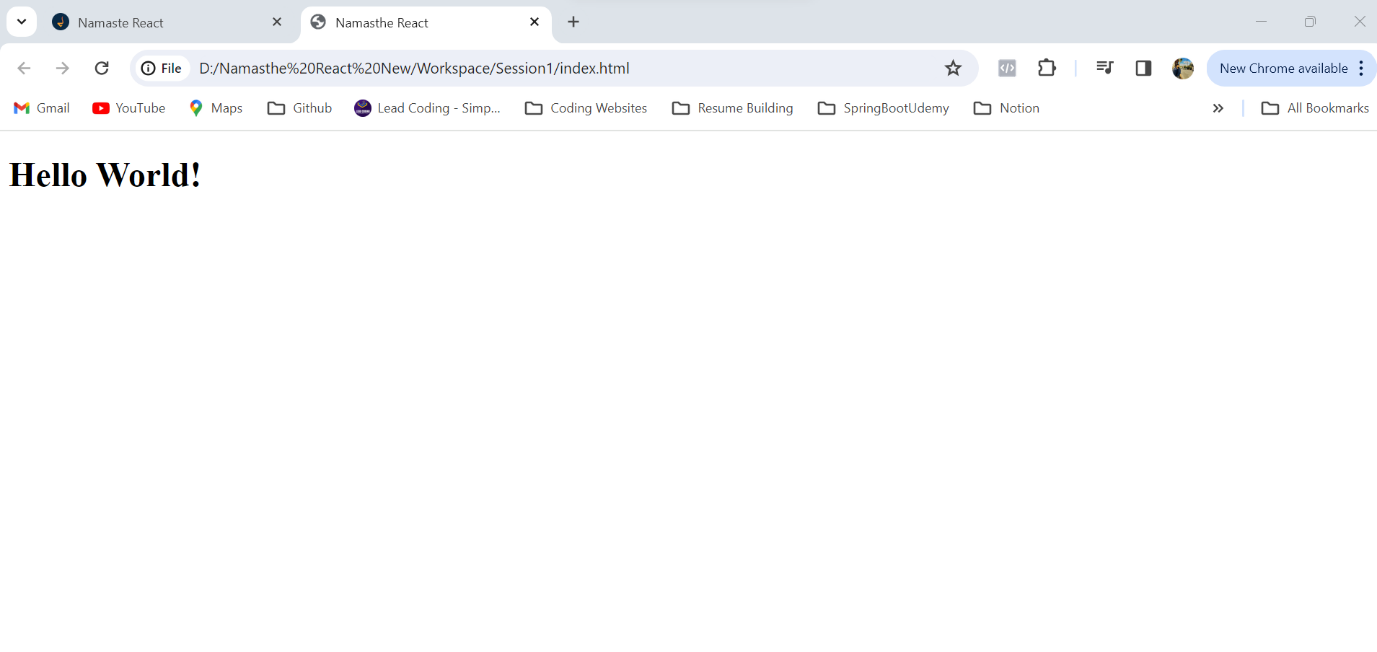
We will just capture to a variable like below by using document.getElementById API

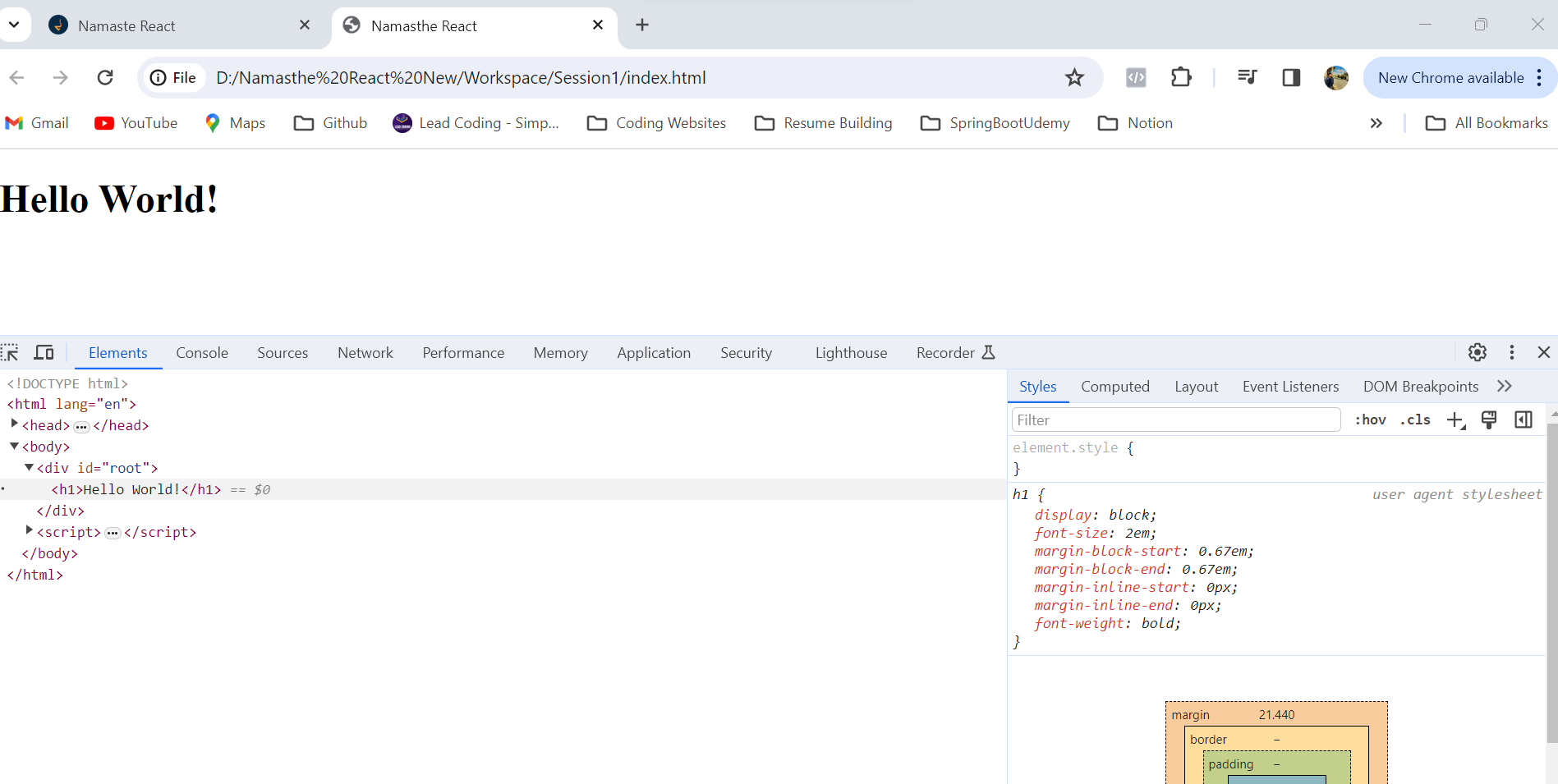


Now we will append the child in to the root by using appendChild API like below



Output –

Now if we just inspect the page we can see a h1 tag created by JS and inserted some text into that as below



**Till now we have done this Hello World page using HTML and JS**

**Now we will see how we can create the same using REACT !!**

Now as we created few things using document API like document.createElement, document.innerHTML etc …

How that project understand what is document ?

How does our browser understand all these ?

→ These are the super powers which browsers already have in it

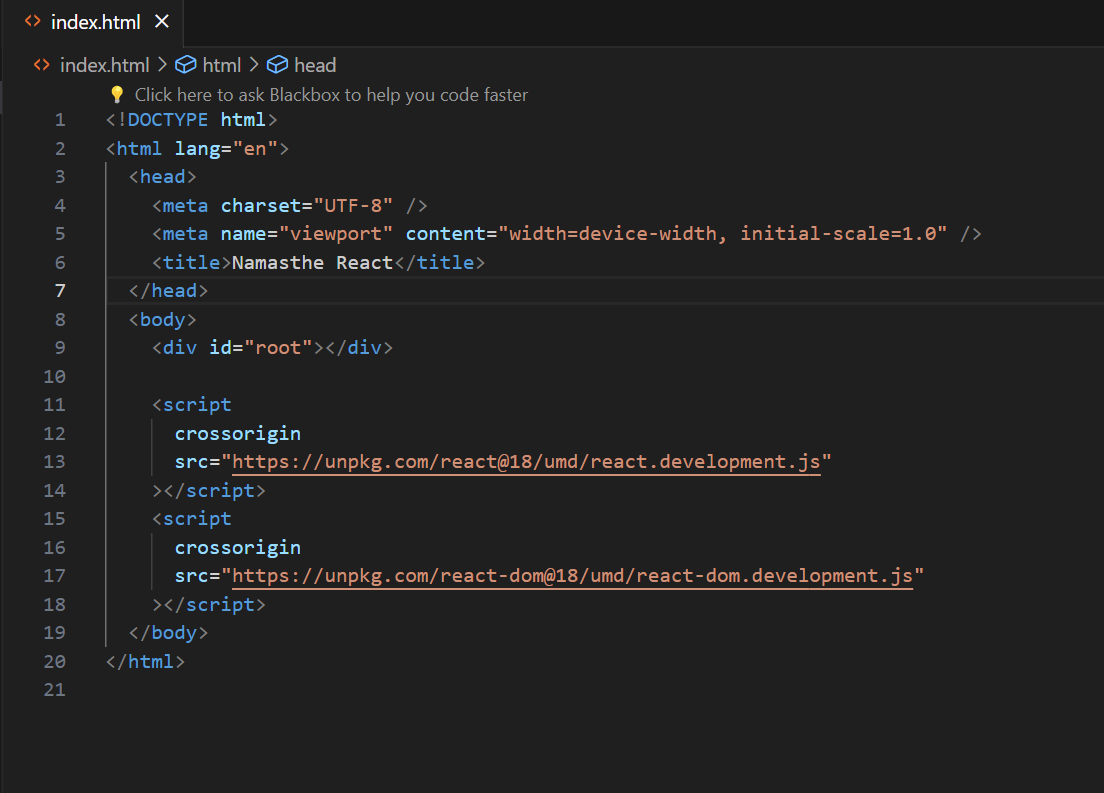
→ browser **has JS engine** in it that executes JS

But does the browser knows what is REACT ? → **NO**

So, we need to get REACT in to our project to use that in our browser

1st Way of **Adding REACT** in to our Project is via **CDN →**

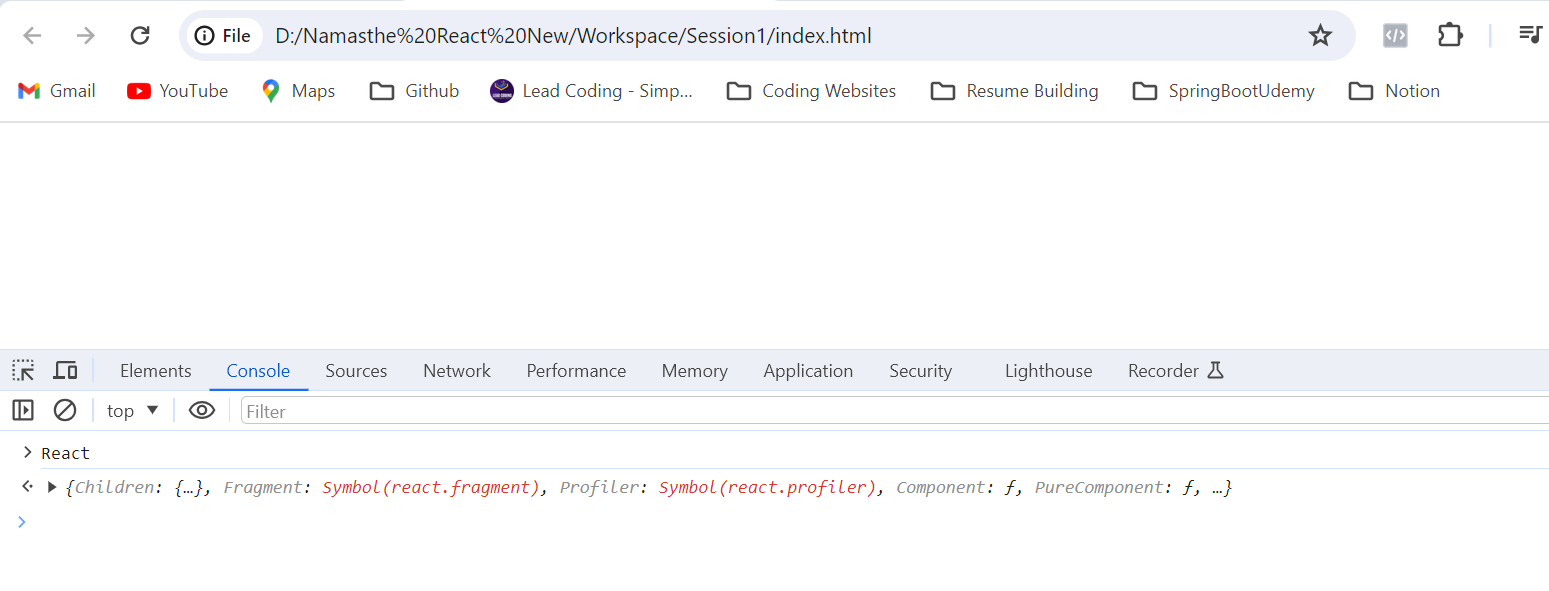
***CDN - content delivery networks → These are the websites where the REACT has been hosted, we are just pulling REACT into our project***



Now with this our project is having **REACT** in it

Now previously written code is removed which we have used HTML and JS

Now after putting REACT cdns in our project there is something happening in behind the scenes Now we will just go to the console in our project and type React in the console like below →



Now this **React is coming from the CDN which we have injected**

Now this React will give us lot of **functions and methods** that we are going to use

Till now this came from the first CDN link

This first file is the core framework algorithm which is written in REACT



Now we will see what is 2nd CDN link used for

This is the REACT library which is useful for DOM operations, or this is the React DOM which we need to modify the DOM



**Why REACT made two different files ? cant we club both and make it one ?**

**No, because REACT doesn’t only work on browsers also works on mobile phones (React Native)**

so there are 2 files one is core one and another one is like a bridge for REACT and Browsers like REACT DOM which is a bridge to connect to the DOM

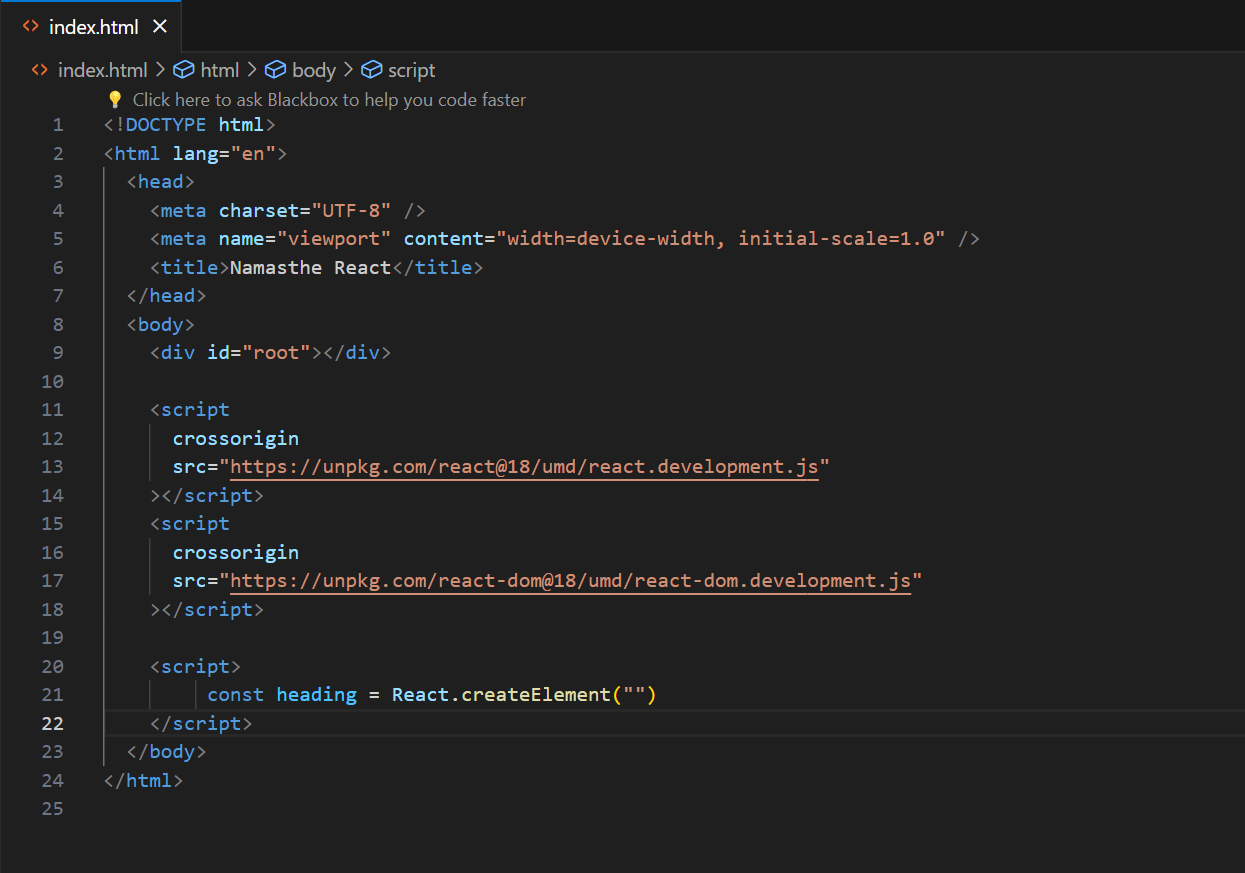
**Now we need to use this powers which are given by REACT and build Real Time Applications**

**We will create a Hello World Project Using REACT -**

Lets see how we can create h1 tag inside REACT →

for that we need script tag and inside that previously we have used in JS like document.createElement

but here in react we will use React.createElement like below



This createElement will take 3 arguments

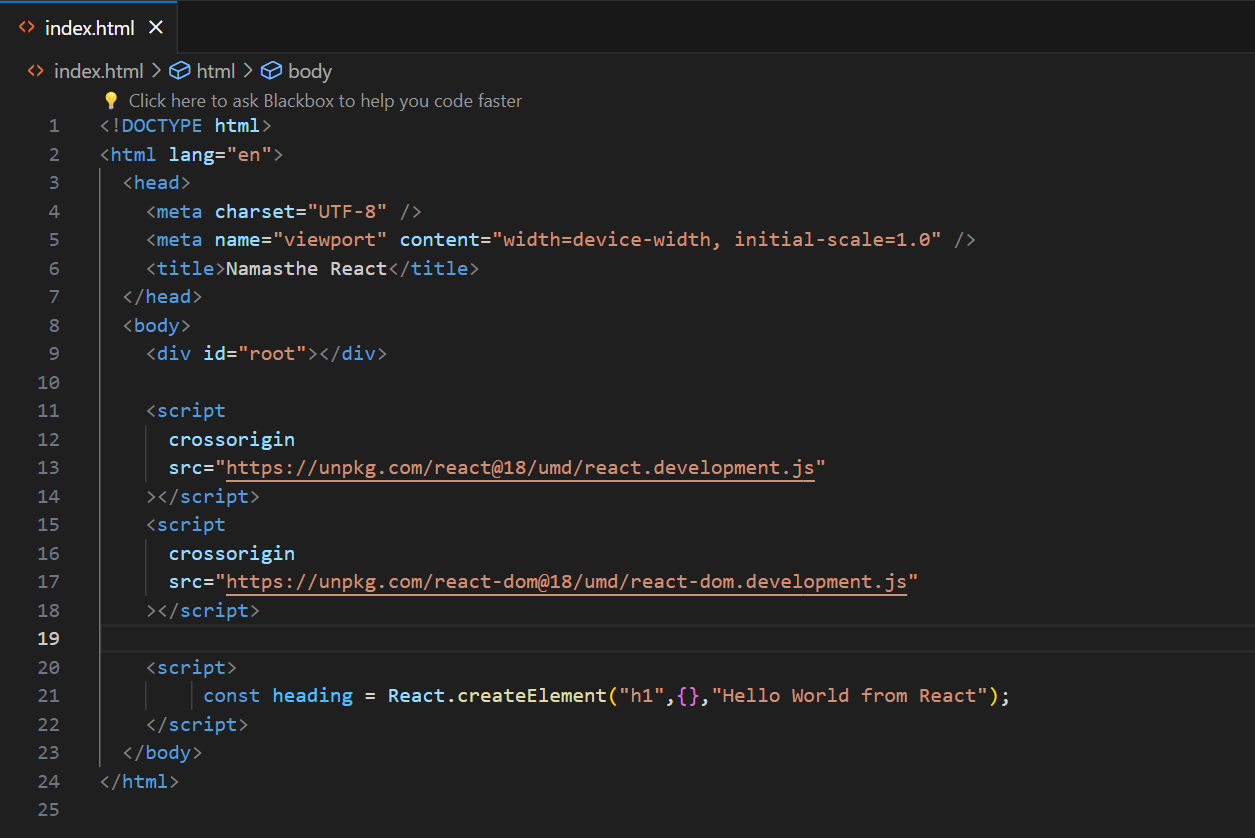


1st is the tag

2nd is the object

3rd is the what we have to put inside that tag

Now we have created a heading and now we need to put that heading inside the div id root



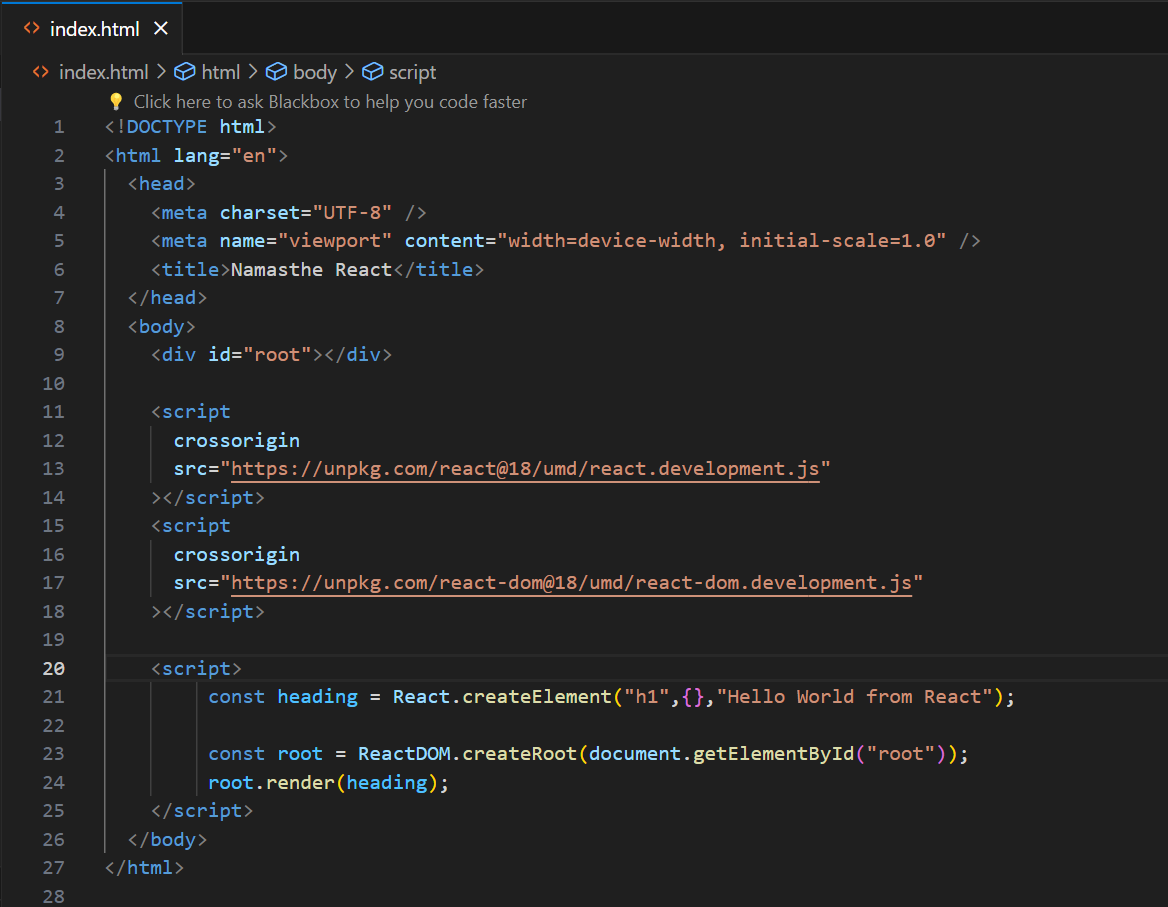
How we will do that ?

First we need to tell to REACT what is **root,** where u need to render

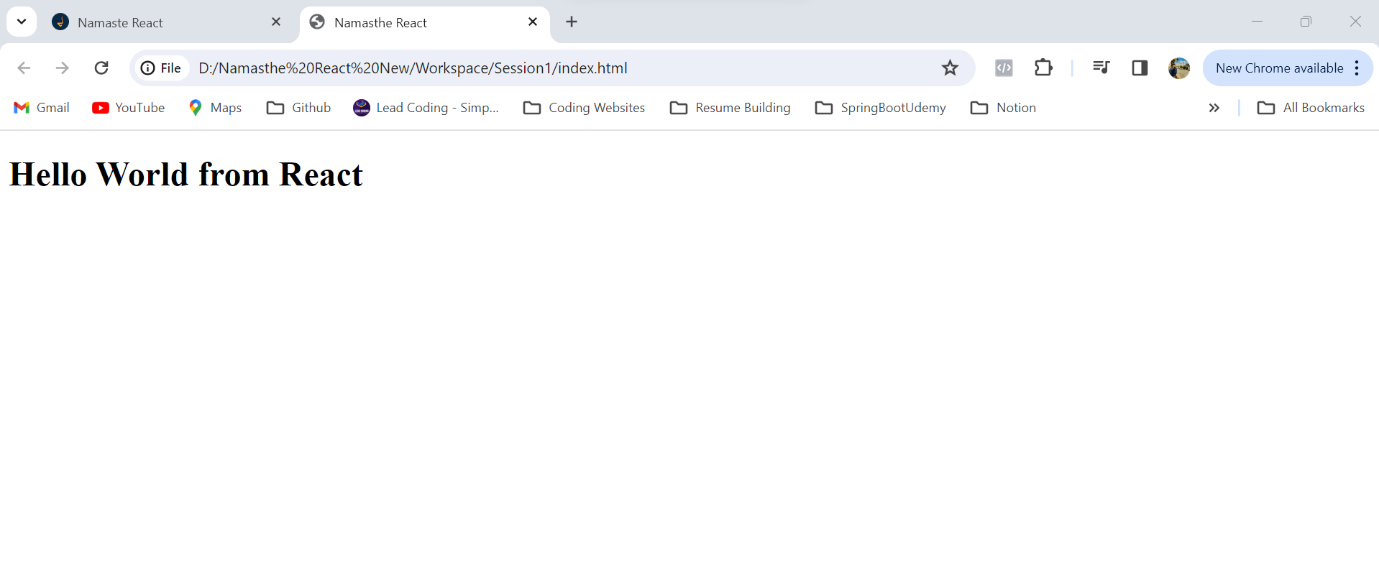
React needs a root where it does all the DOM manipulations

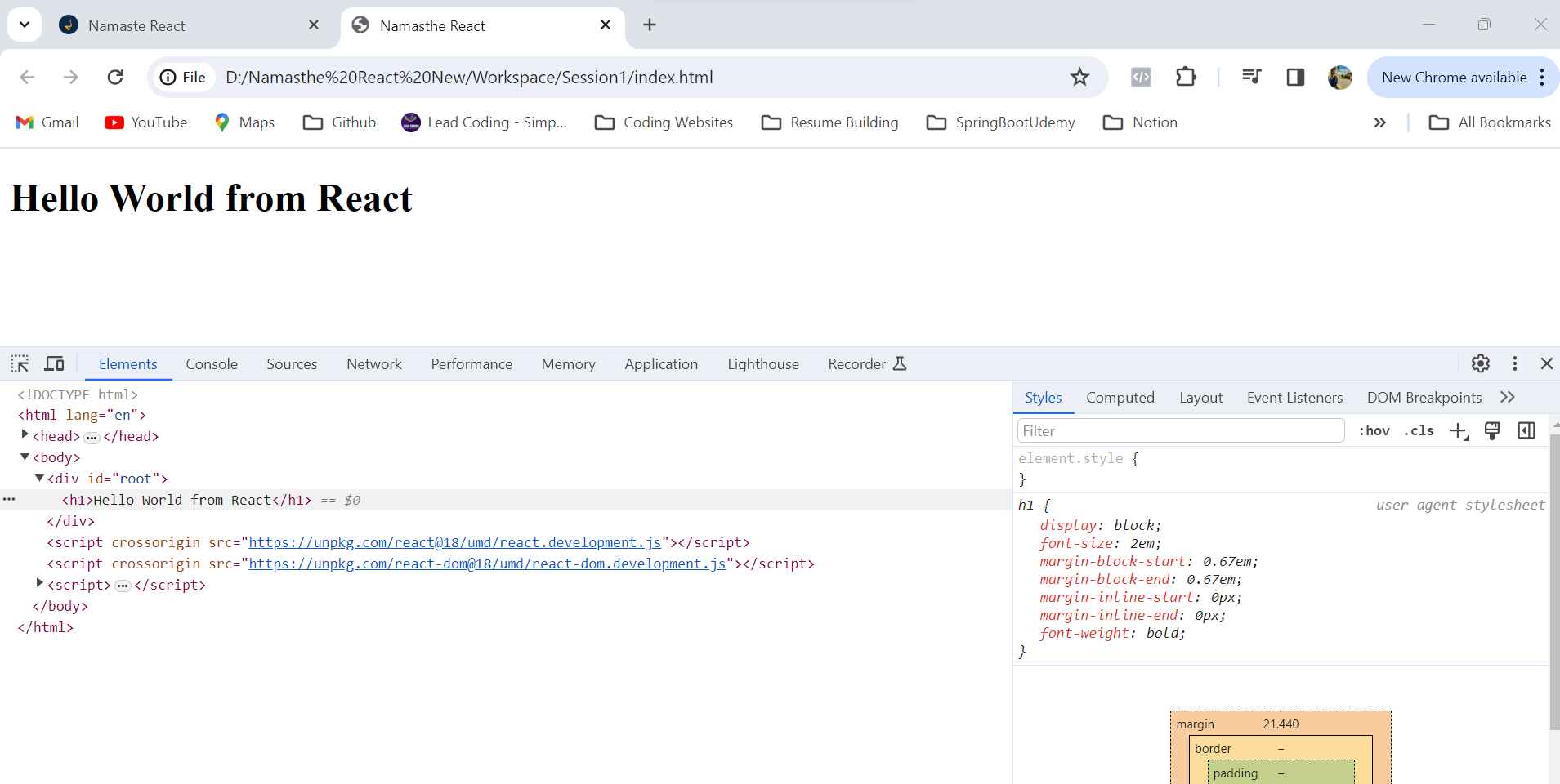
we will use ReactDOM.createRoot(), this will create a root first

And now we need to render that heading inside that root like below



Output –

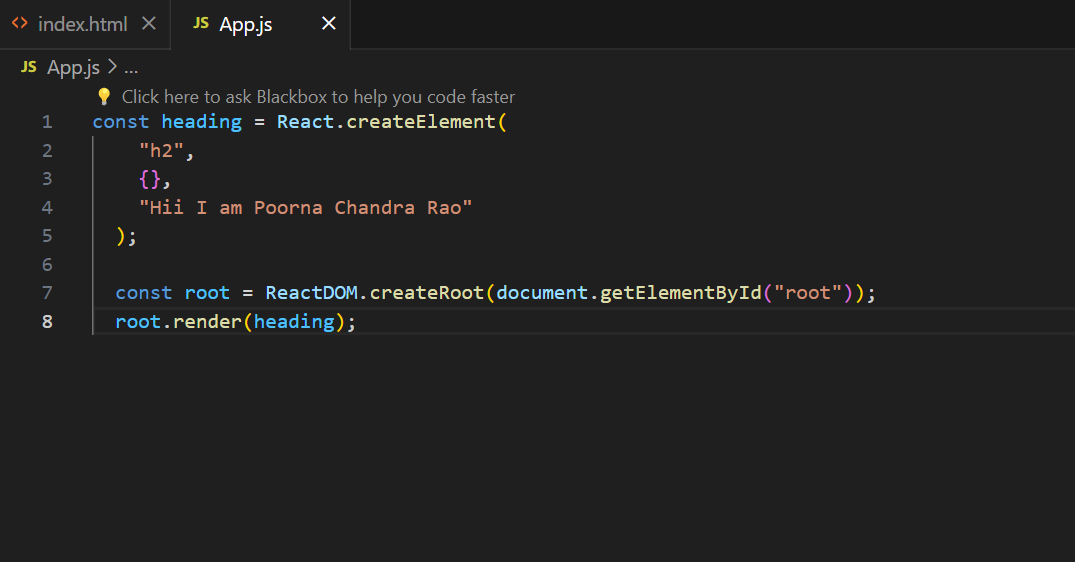




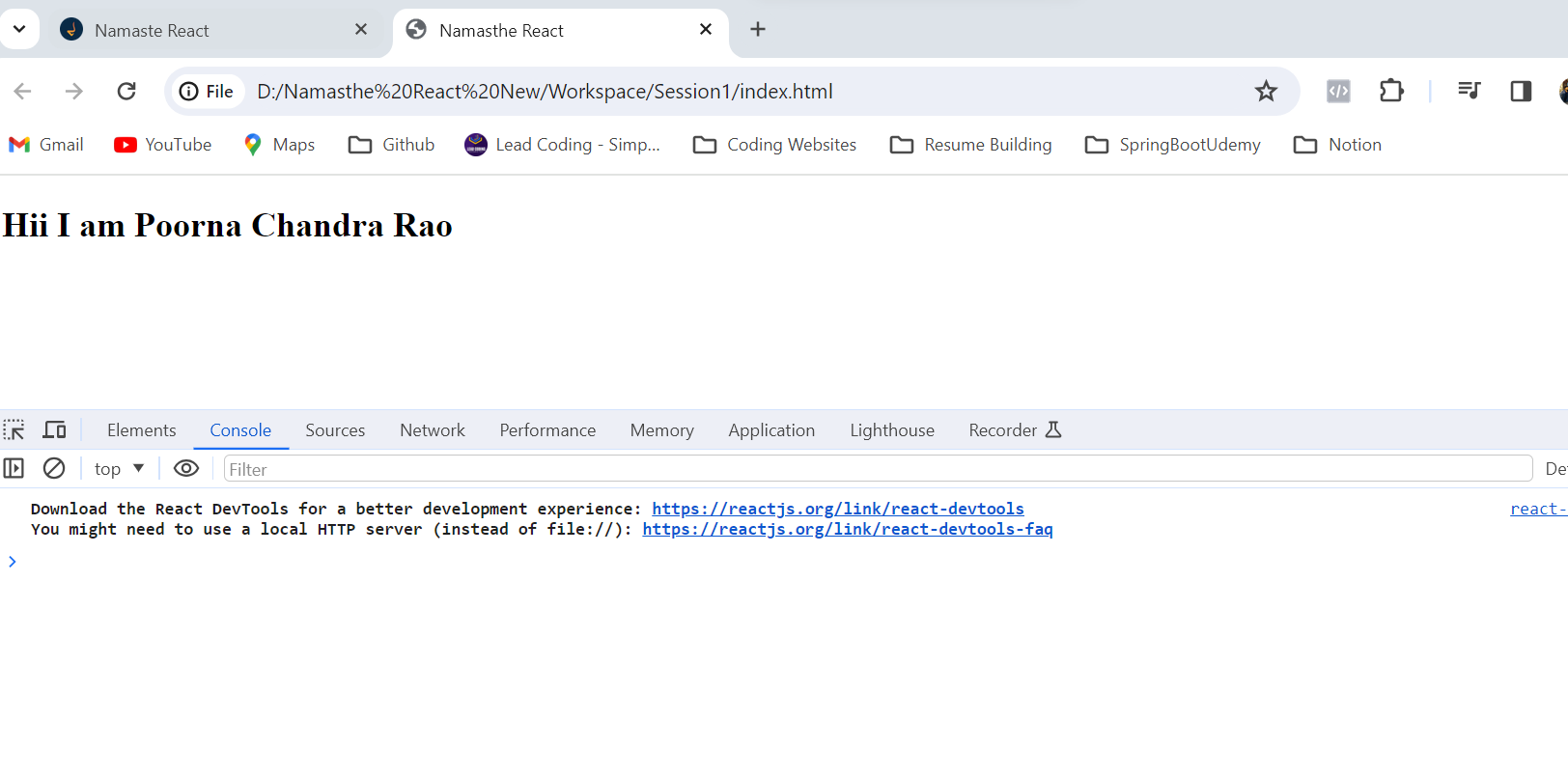
Till Now we wrote all our code in HTML file by using script tag …

Now we will write the same code in separate JS file like below





Same output like before –

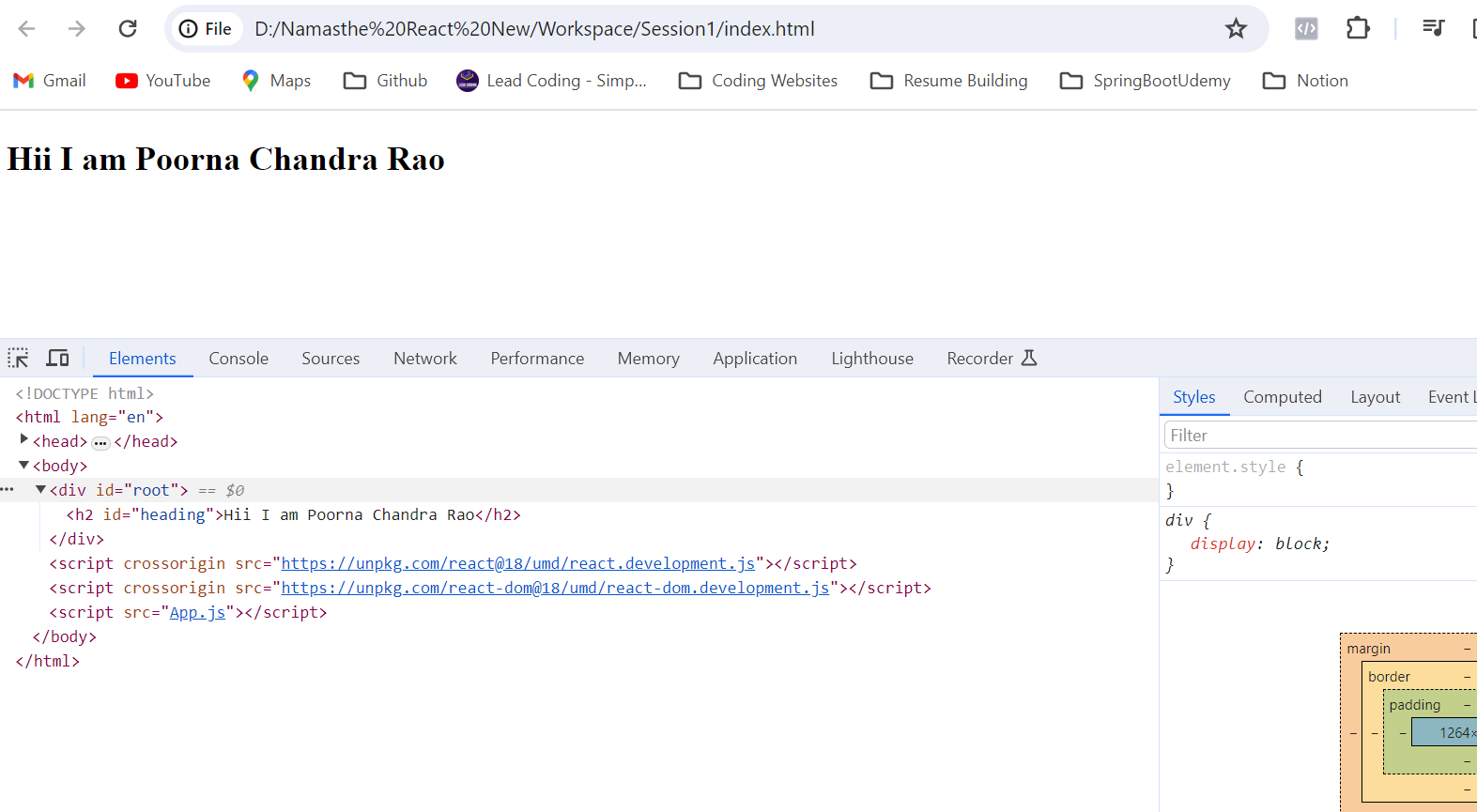
In the below code we have left 2nd parameter as empty



That is the object where we can give attributes to the Tag

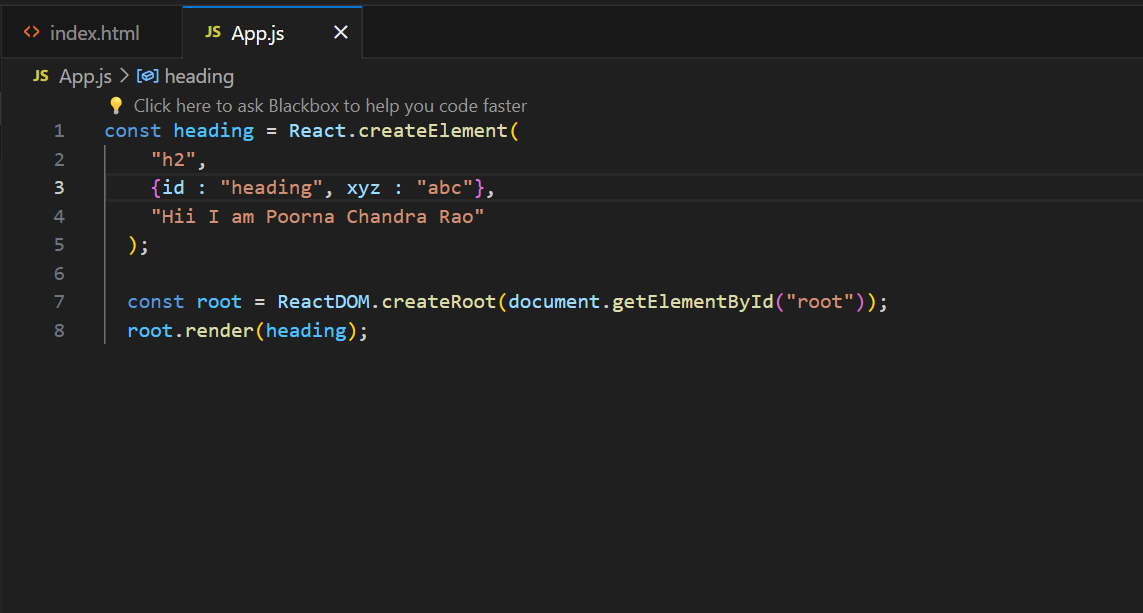
For suppose if our h1 needs some id like below

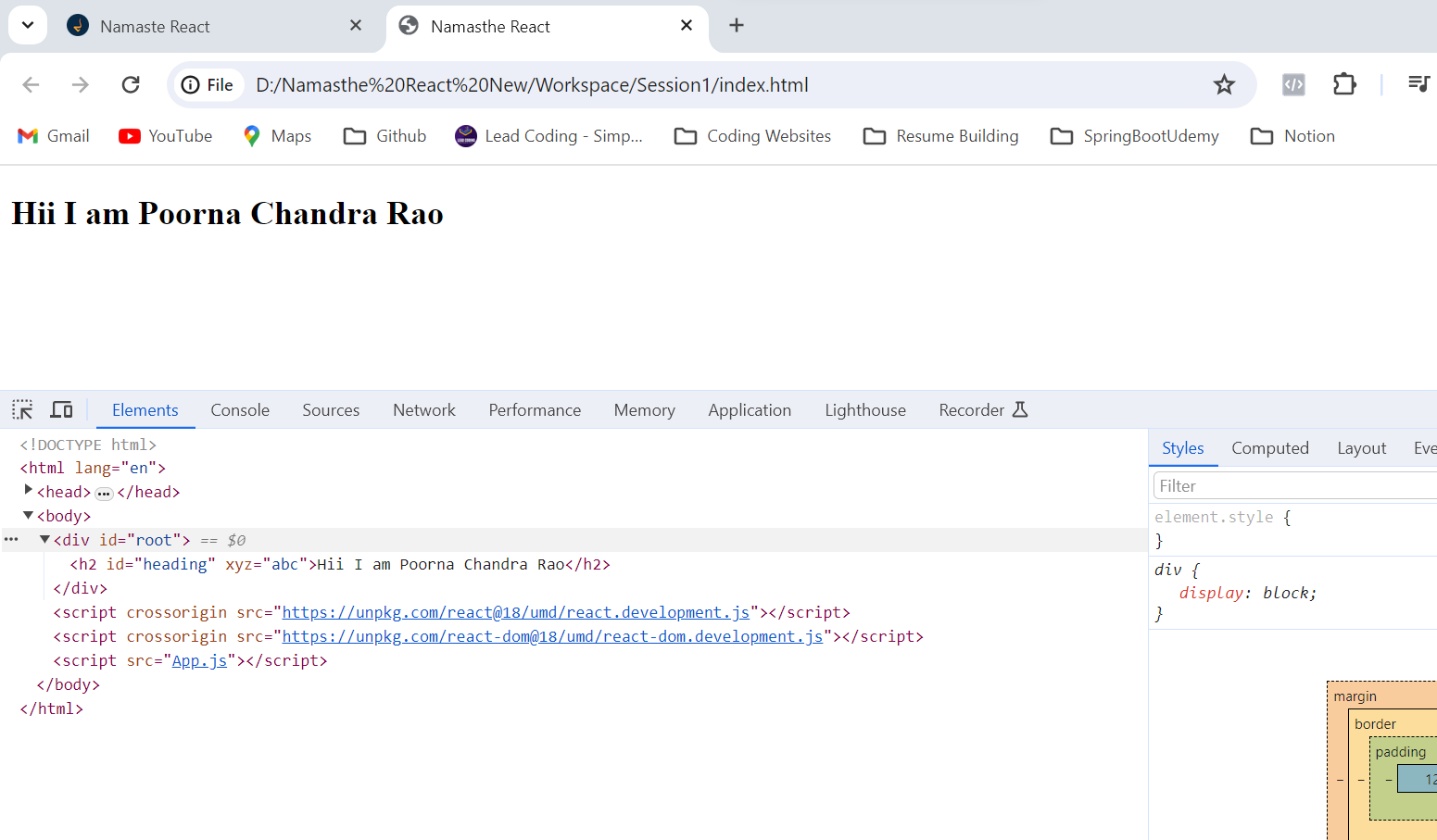




Here in the above we can see that id=”heading” added to our tag when we give in that Object

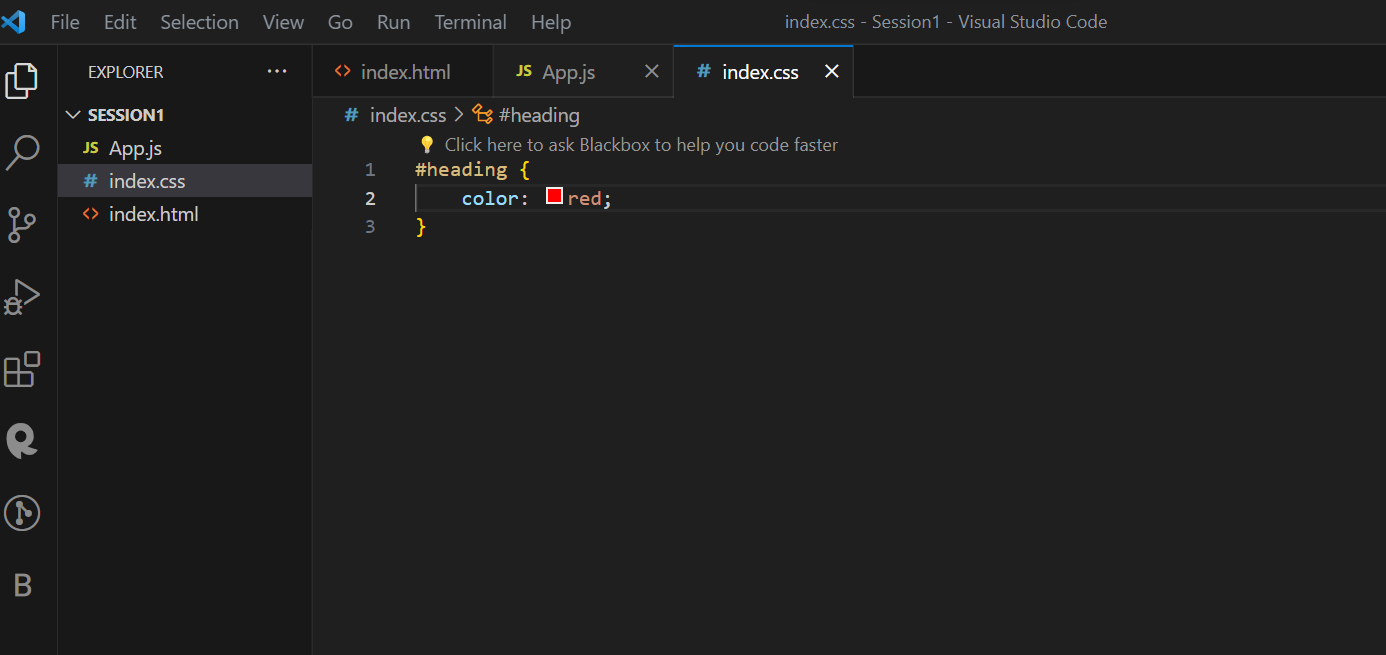
Like wise we can give multiple attributes also

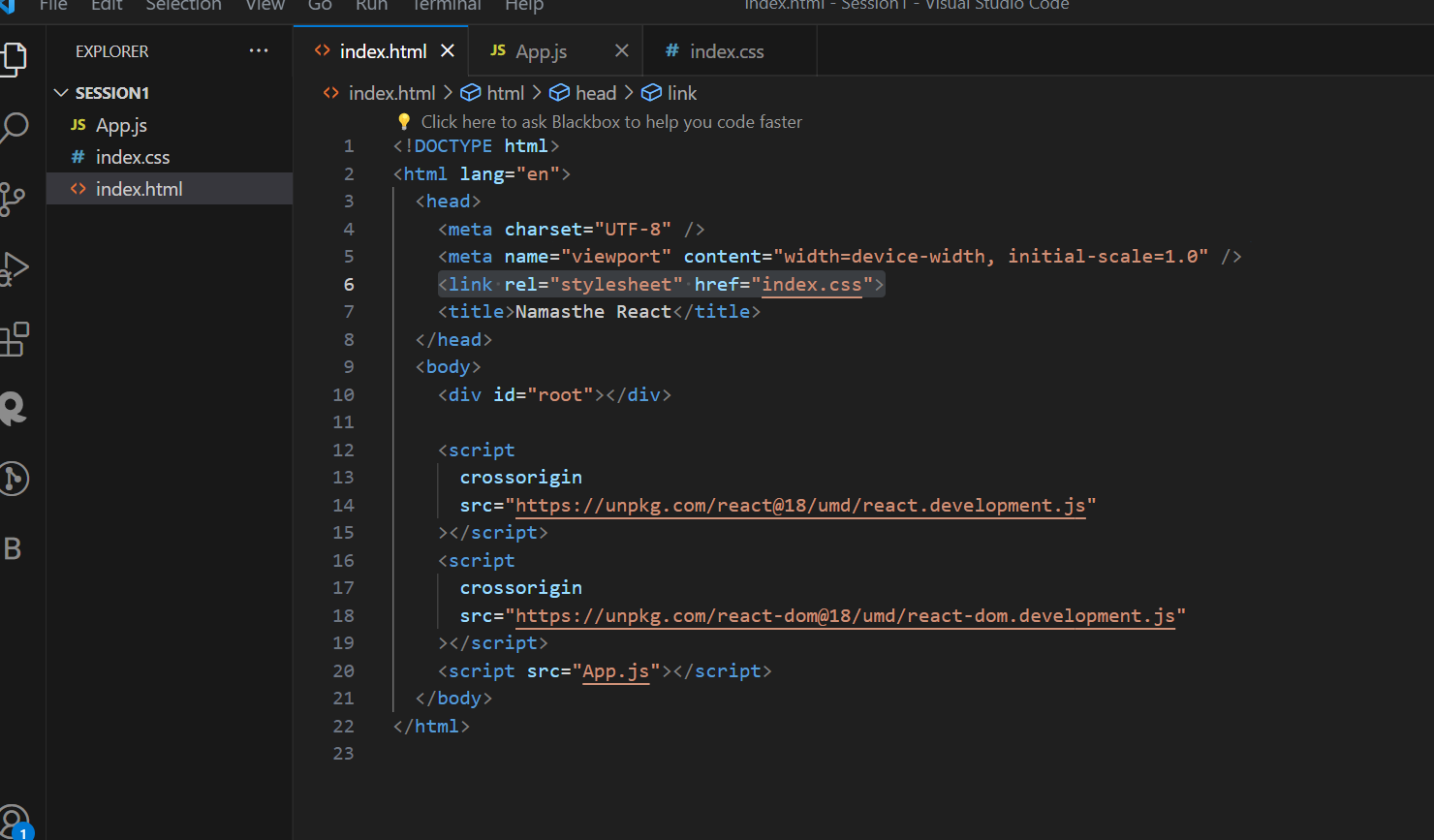


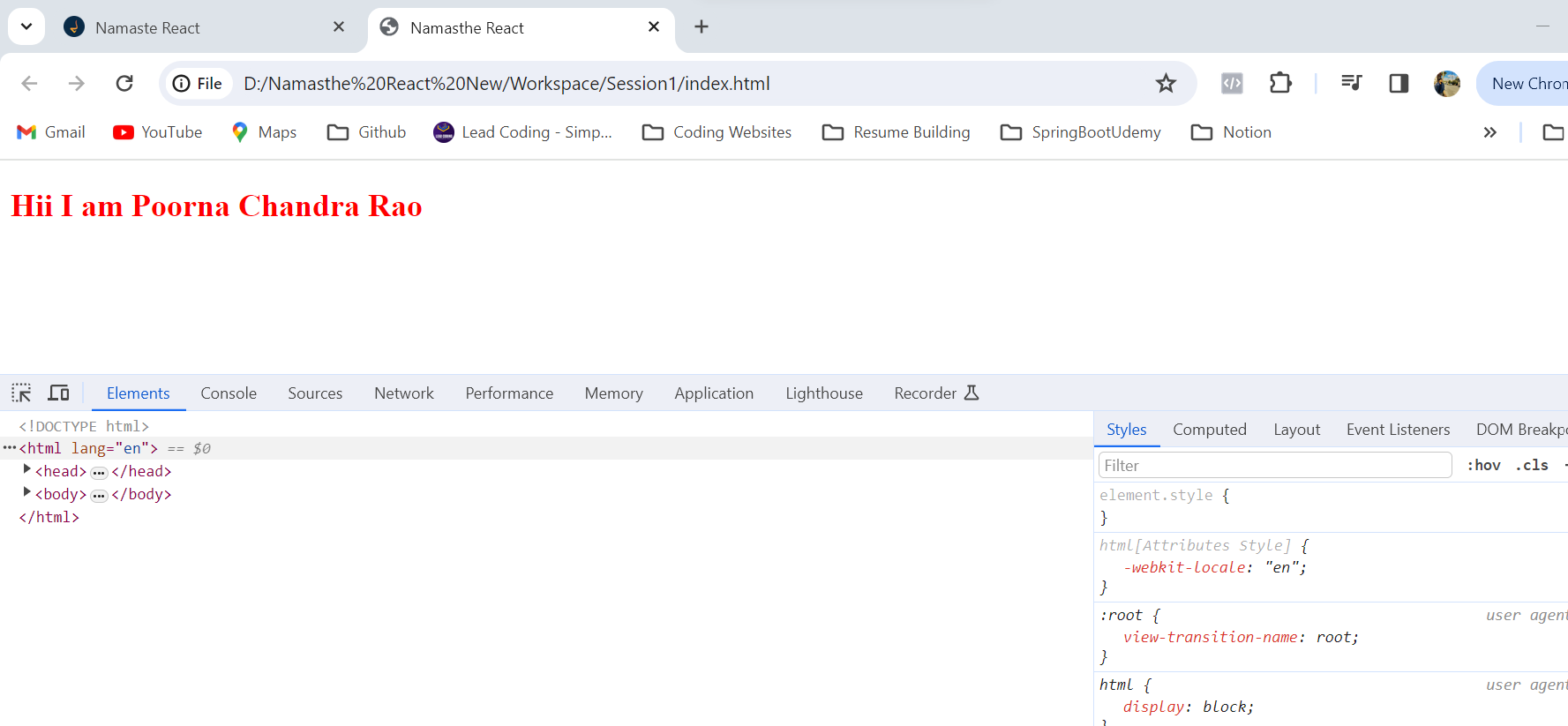


If we need to add any CSS

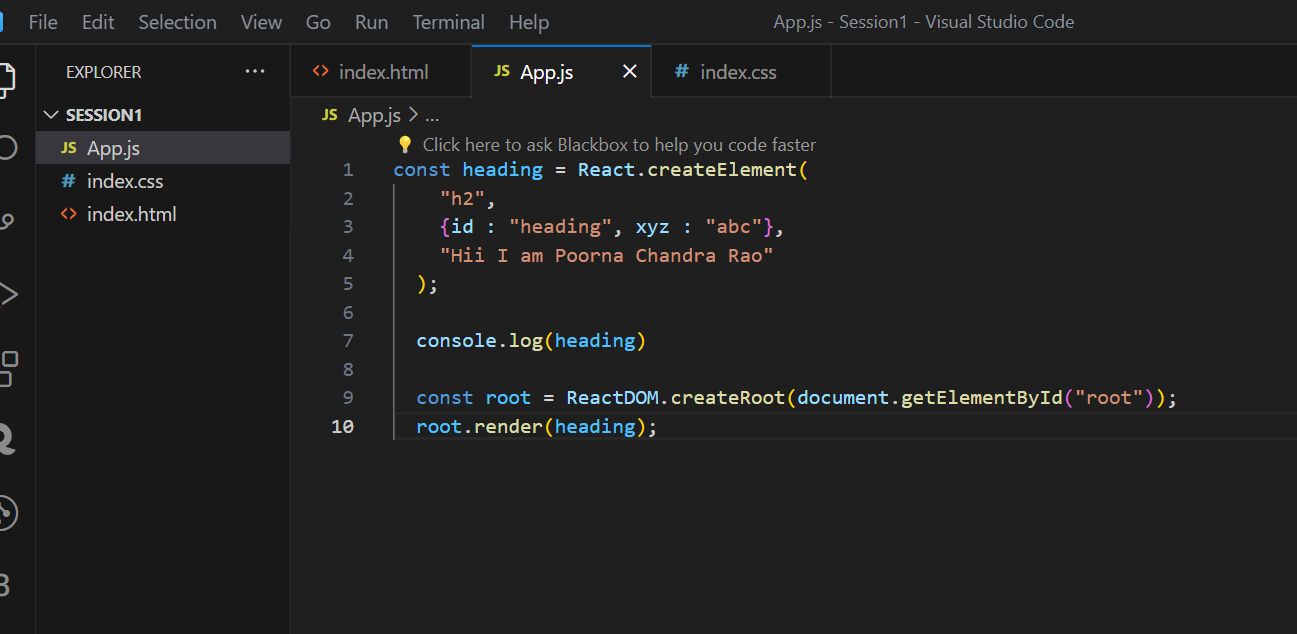
we will create one CSS file and we will link that in our HTML like below



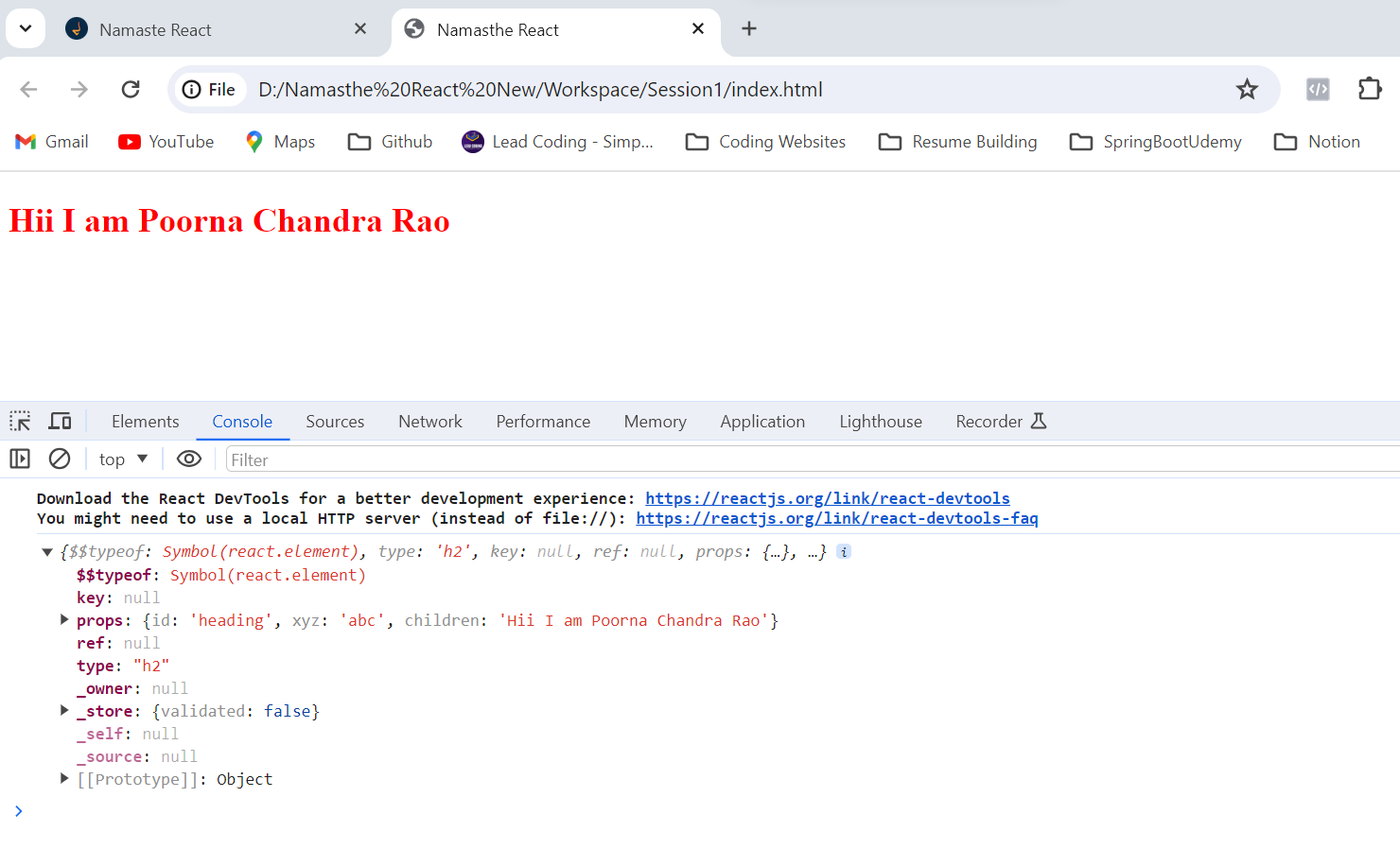




Now that we have been using React.createElement … but what is that exactly ? we will just console that and check



Here it is just returning one **React Object** like below



And it has something known as props → **props are children and attributes** we pass in

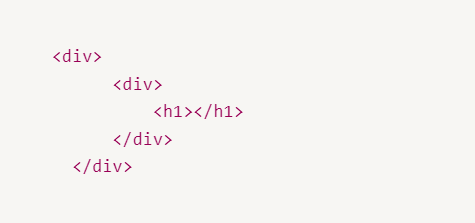
in the root.render we are just passing and React element



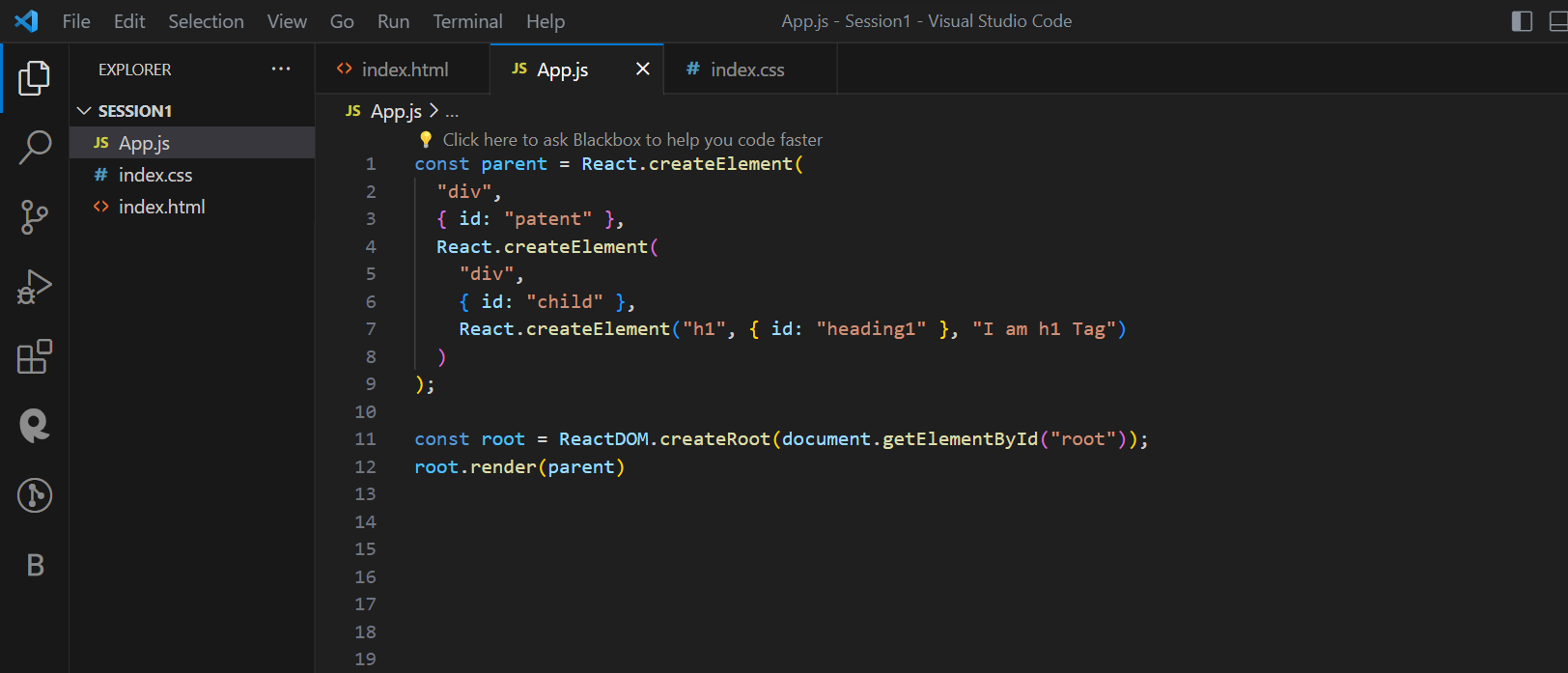
Now this **root.render** job is basically to take this object and **create the h1** tag which the browser understands and put that under **root**

Now as we know HTML wont be in simple structure …

So when we want to create some complex HTML with nested tags like below



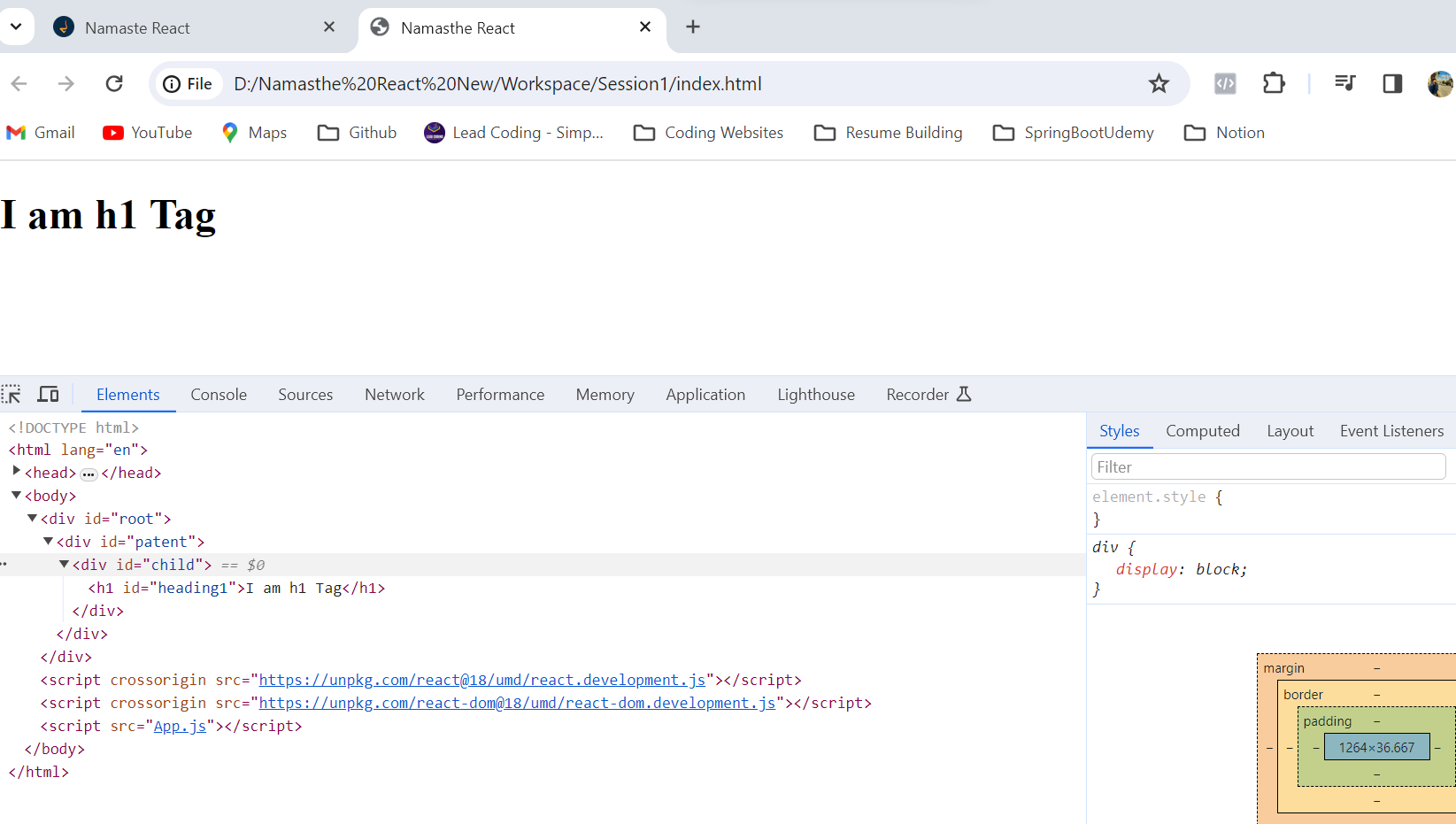
How we can create nested elements in REACT ?



**Here what we did means in every React.createElement the last argument is nothing but the content or child which we need to attach for that particular element we need to give there. So, we have given child as content to parent and inside child we have given h1 as content as above**

In this way we can achieve **nested elements in React !!**

Output –

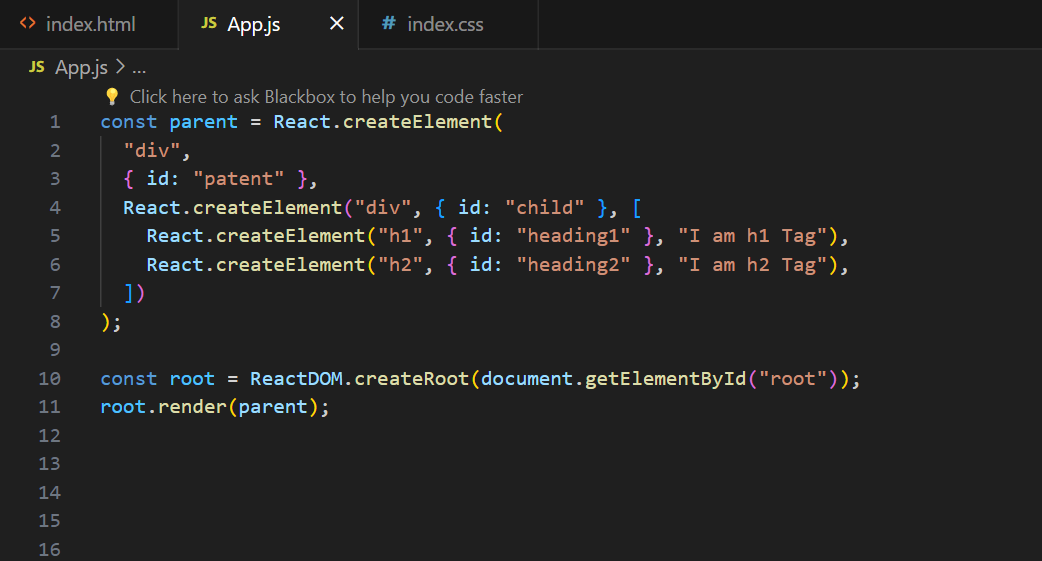


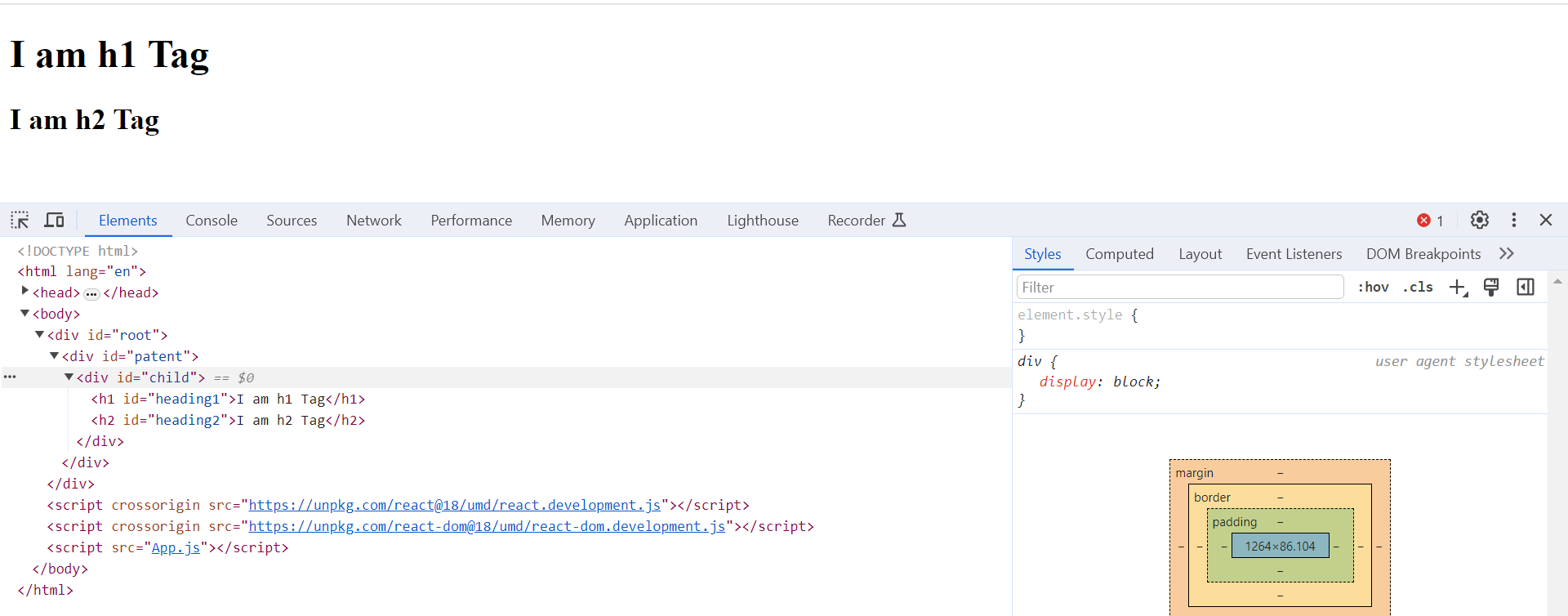
Now how to add siblings →

If we need to have siblings,

Now the 3rd argument in that React.CreateElement is for passing children, and in that we can either one child or we can give more that one Children with Array

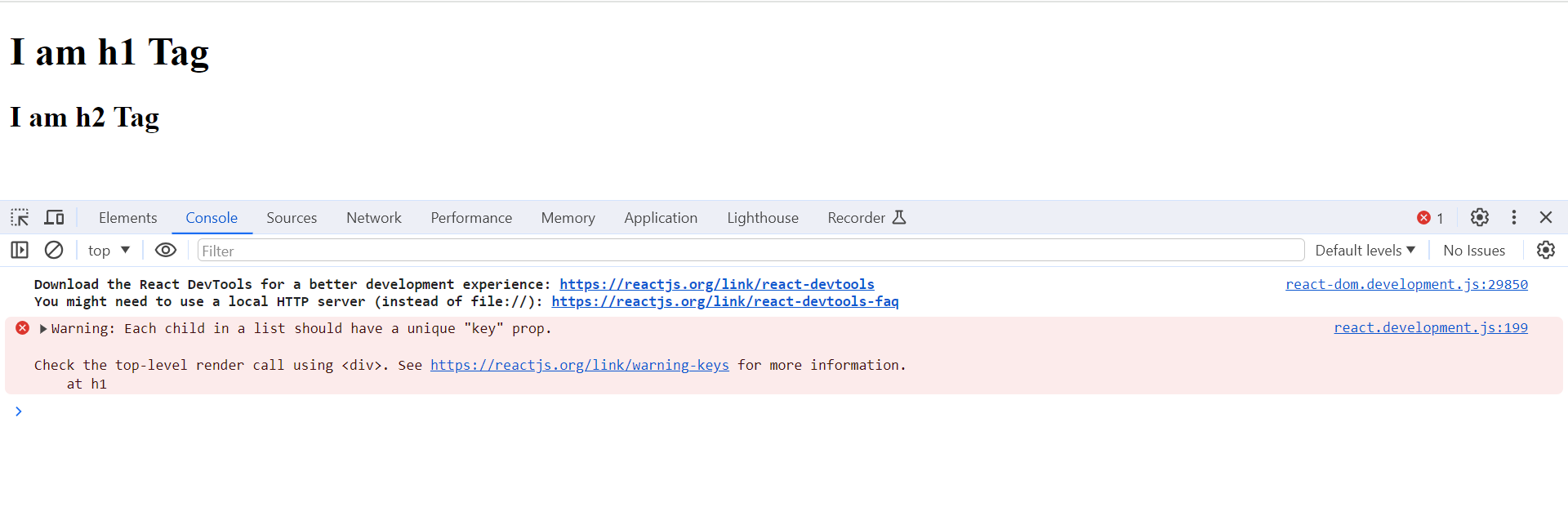
It can be a single child or an Array of children like below





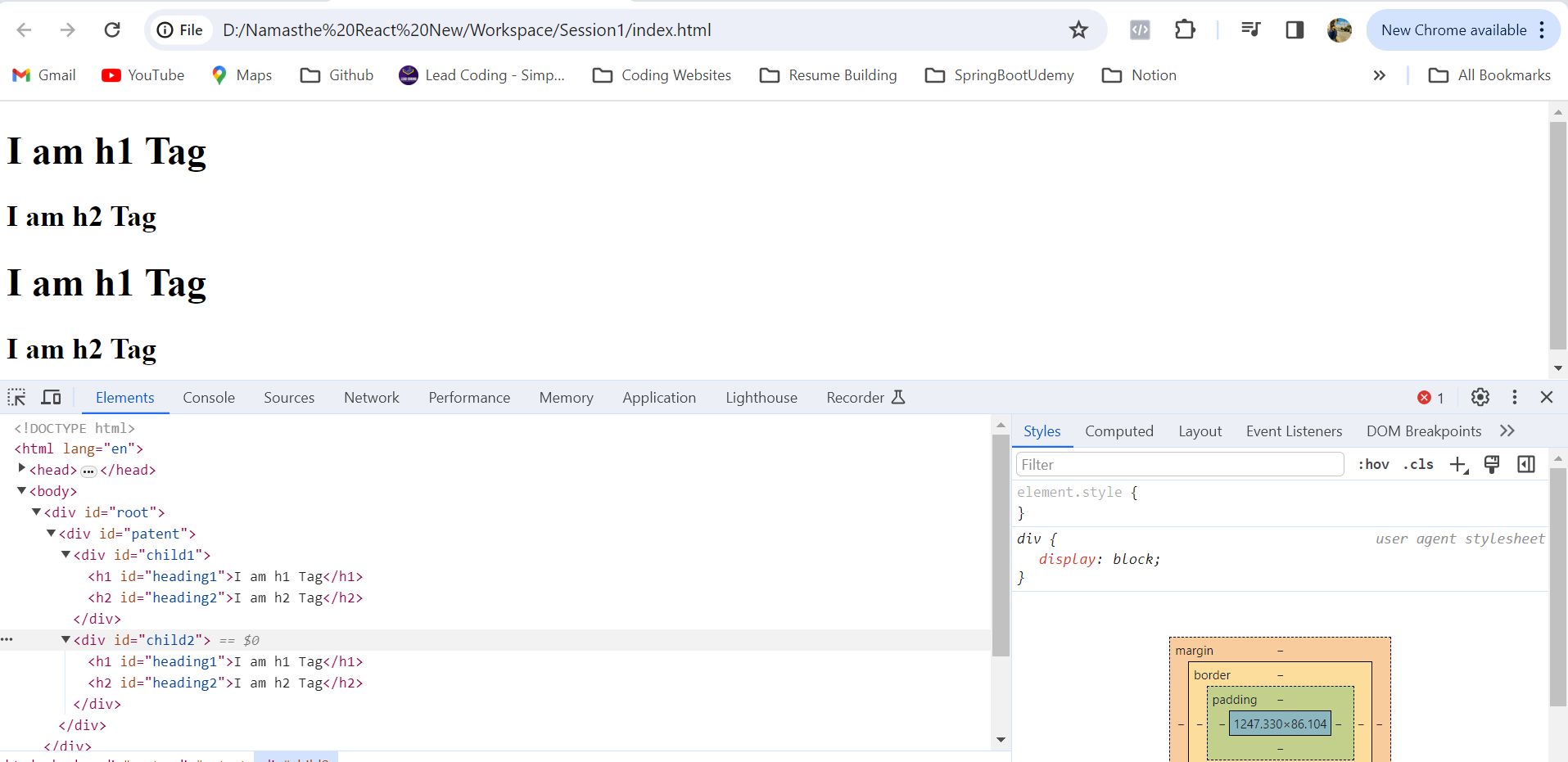
Now there I have given array of children in the 3rd Argument

When you did like this React will throw an error like below

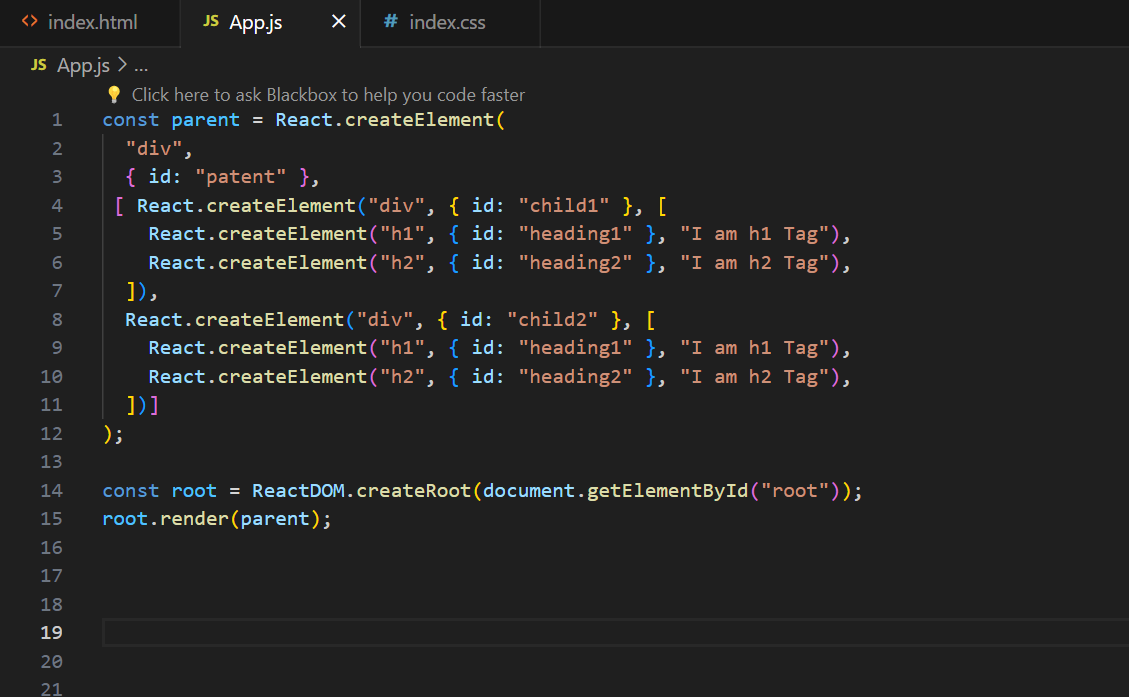


For this we can discuss later

Now we will something like child1 and child2 which are having h1 and h2 as children



Code –



Here we gave like for Parent 3rd Argument we have given and Array of two children namely child1 and child2

And in that child1 3rd argument I have given array of h1 and h2

same like child1 I have given child2 3rd Argument of array of h1 and h2

As of now this code looks complex and untidy

For this reason there is something called **JSX**

Note :- Here we are rendering parent to root using root.Render() method But what if that root is already having some elements ?

Ans :- It will replace the present content and render parent over that