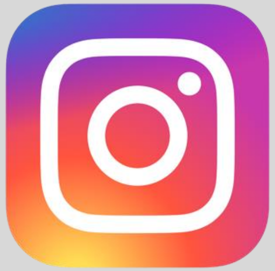
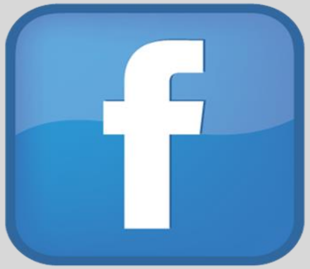
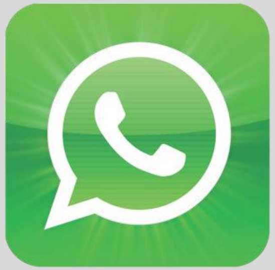
REACT PORTFOLIO

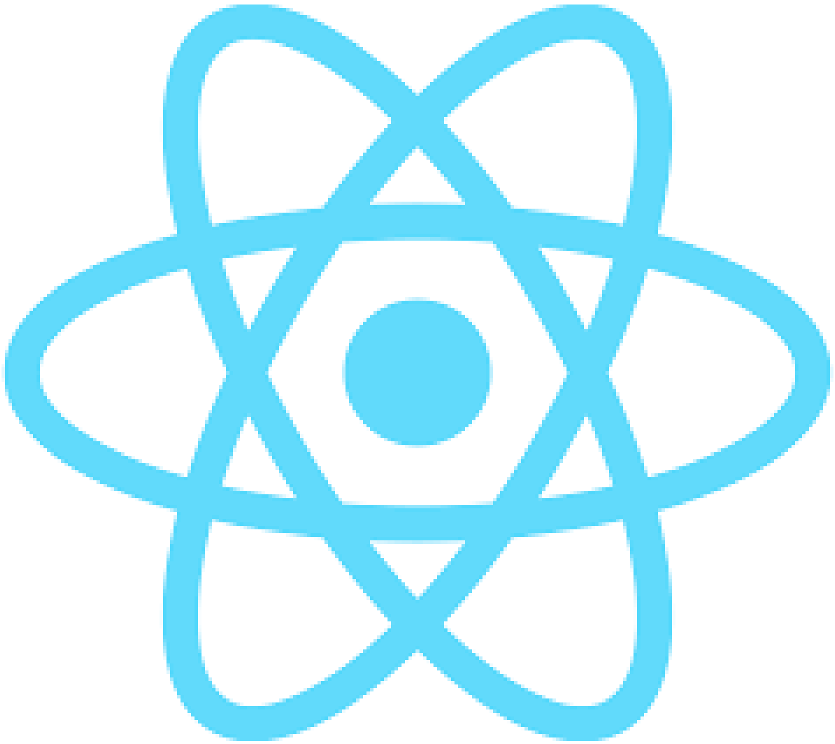
Presented By
Chandrika Mukherjee

Do you have any examples of where React is used?





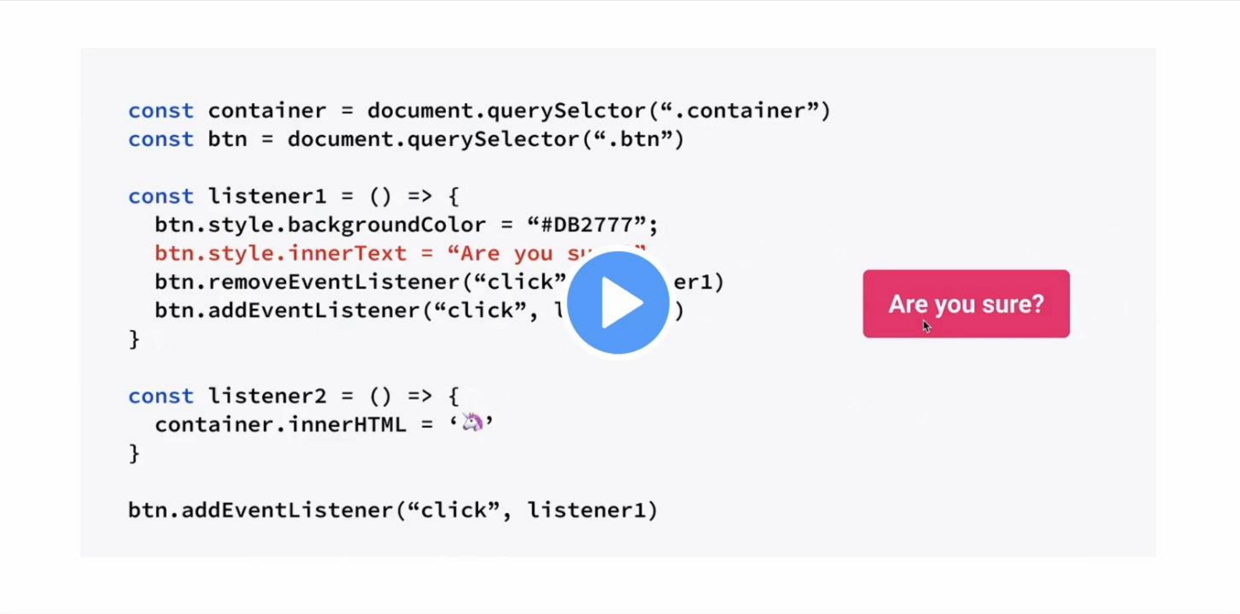




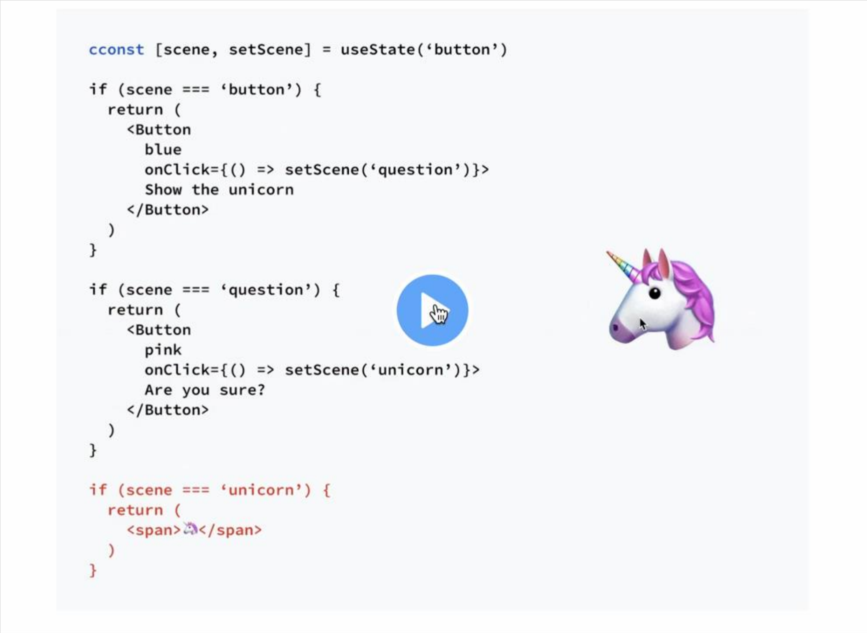
React

- JavaScript library to build user interface
- Useful for building single page application
- Reusable UI components
- Declarative (React handles the rendering itself based on current states)
- JSX – XML like syntax
- Fast, efficient and easy to learn.

Declarative or Imperative?



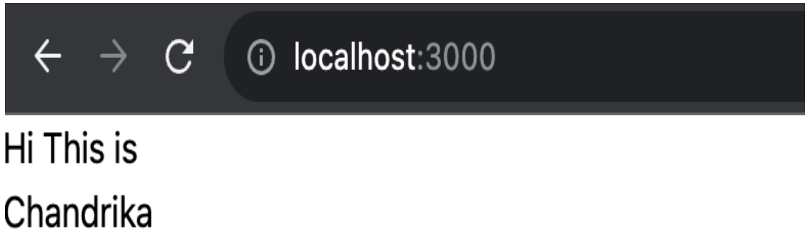
```
const container = document.querySelctor(“.container”)
const btn = document.querySelector(“.btn”)

const listener1 = () => {
  btn.style.backgroundColor = “#DB2777”;
  btn.style.innerText = “Are you su
  btn.removeEventListener(“click”      er1)
  btn.addEventListener(“click”, l      )
}

const listener2 = () => {
  container.innerHTML = ‘🦄’
}

btn.addEventListener(“click”, listener1)
```



```
cconst [scene, setScene] = useState(‘button’)

if (scene === ‘button’) {
  return (
    <Button
      blue
      onClick={() => setScene(‘question’)}>
      Show the unicorn
    </Button>
  )
}

if (scene === ‘question’) {
  return (
    <Button
      pink
      onClick={() => setScene(‘unicorn’)}>
      Are you sure?
    </Button>
  )
}

if (scene === ‘unicorn’) {
  return (
    <span>🦄</span>
  )
}
```

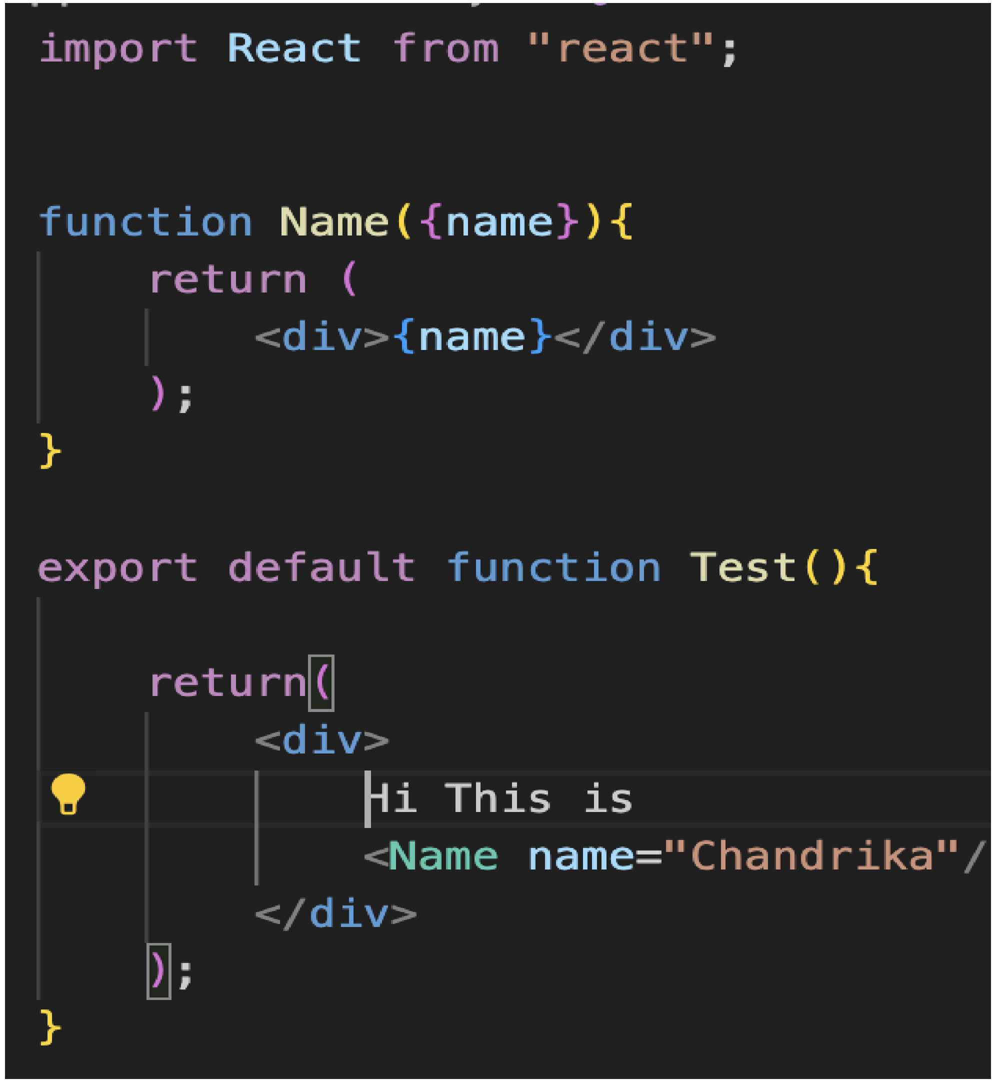
Let's understand some basic structure

Output



Hi This is

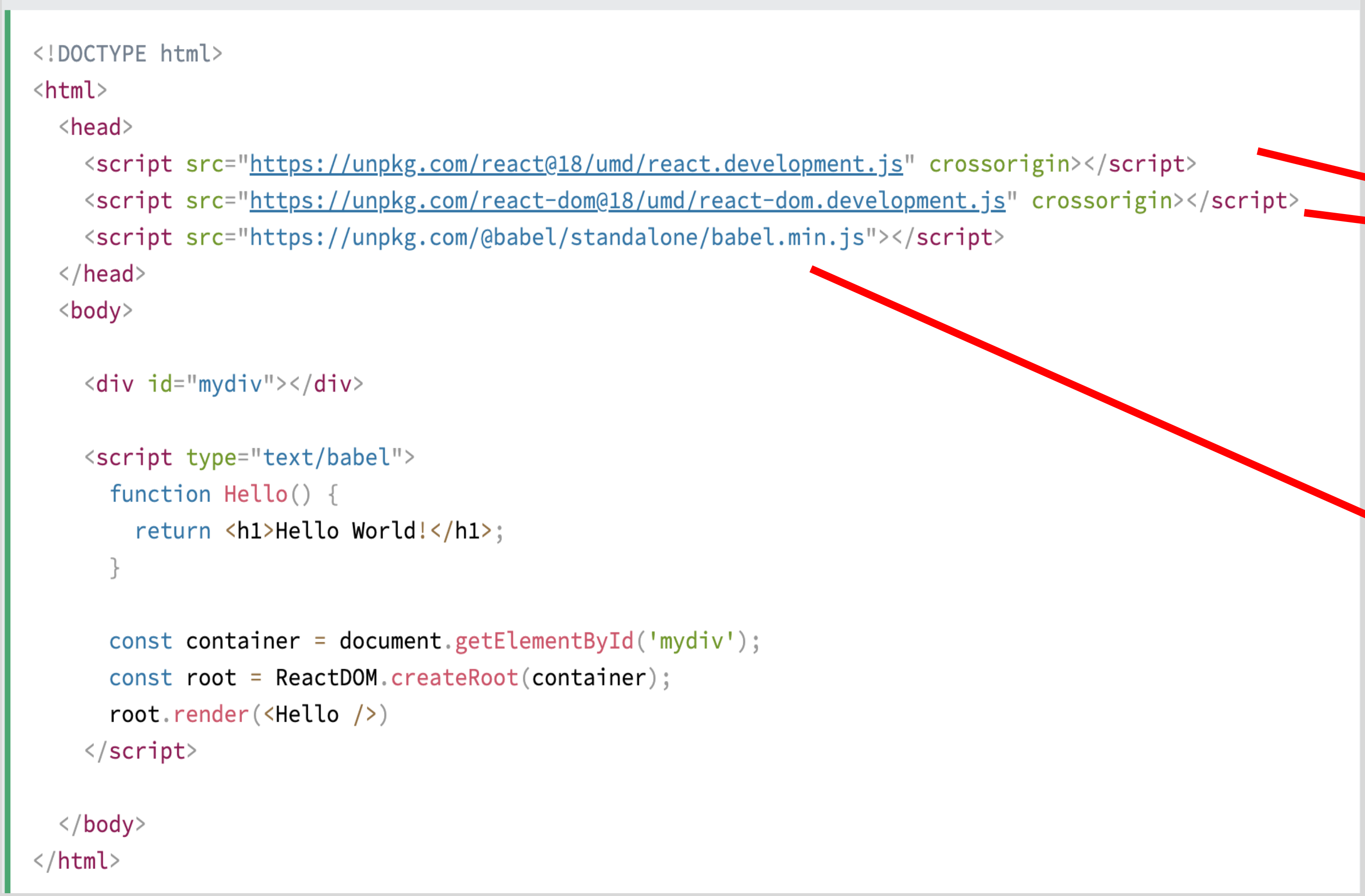
Chandrika



```
import React from "react";


function Name({name}){
    return (
        <div>{name}</div>
    );
}

export default function Test(){

    return(
        <div>
            Hi This is
            <Name name="Chandrika"/
        </div>
    );
}
```

HOW TO CREATE THE ENVIRONMENT FOR DEVELOPMENT

React Directly in HTML



```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  </head>
  <body>

    <div id="mydiv"></div>

    <script type="text/babel">
      function Hello() {
        return <h1>Hello World!</h1>;
      }

      const container = document.getElementById('mydiv');
      const root = ReactDOM.createRoot(container);
      root.render(<Hello />)
    </script>

  </body>
</html>
```

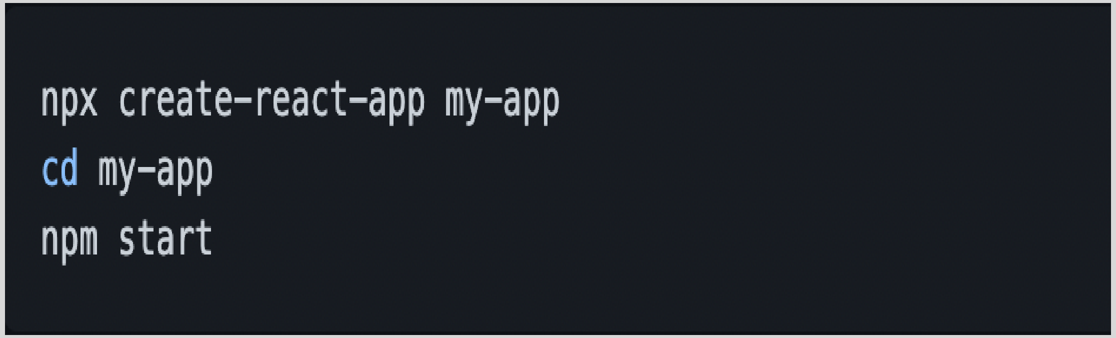
Write React Code in Javascript

Write JSX Syntax and ES6

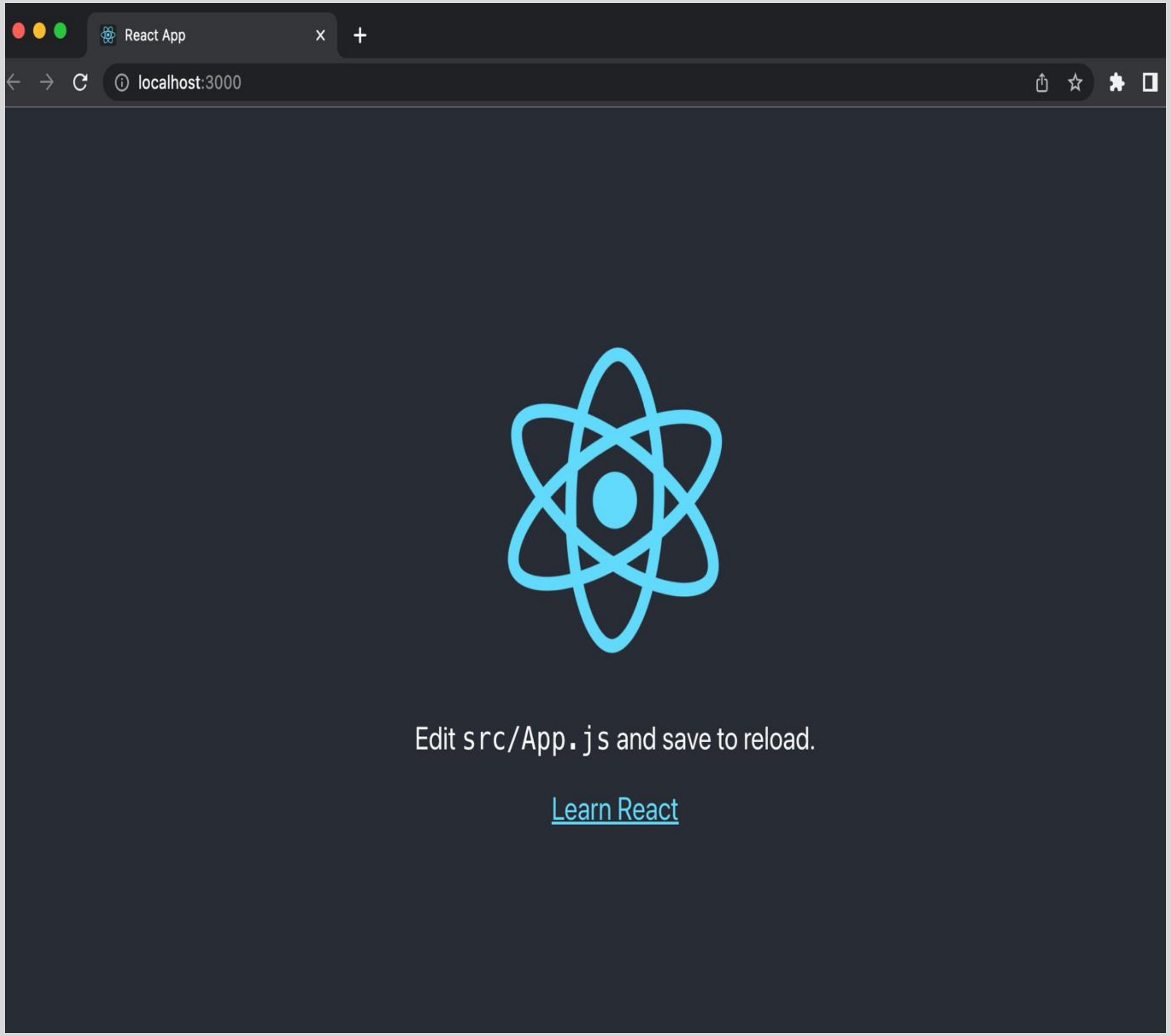
INSTALL NODE.JS
AND
NPM [NODE PACKAGE MANAGER]

Installing Node.js, npm

Creating React App



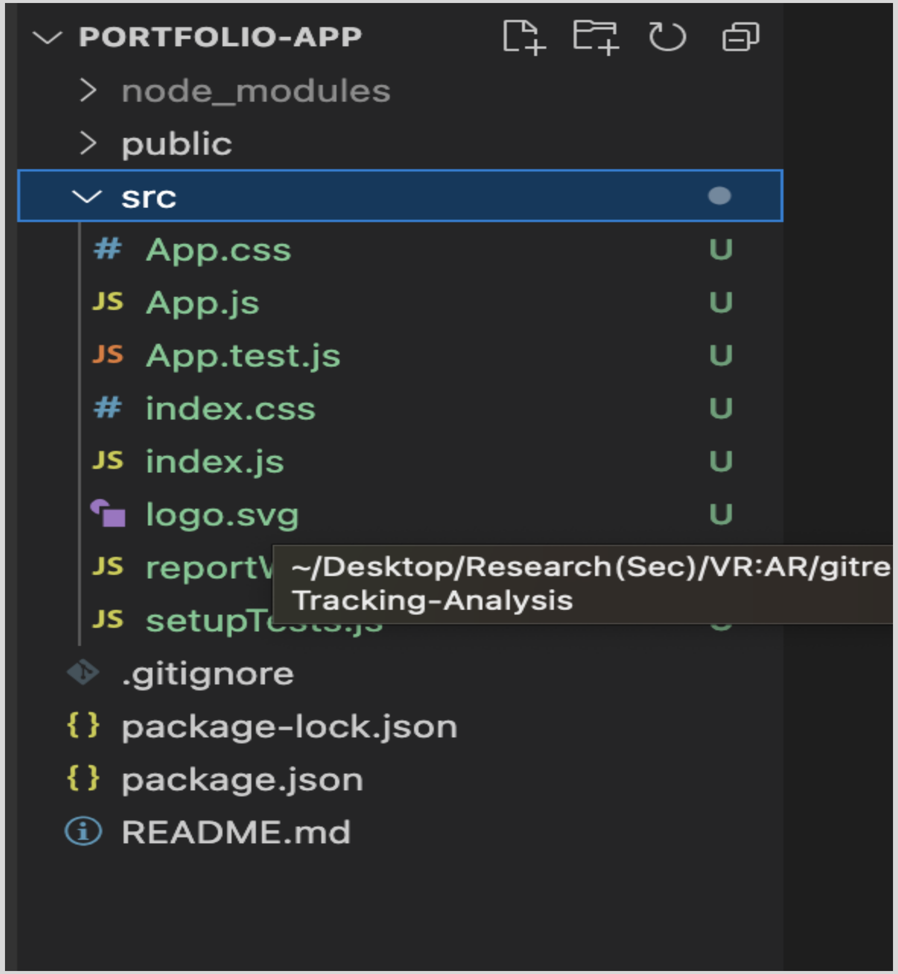
```
npx create-react-app my-app
cd my-app
npm start
```



Edit `src/App.js` and save to reload.

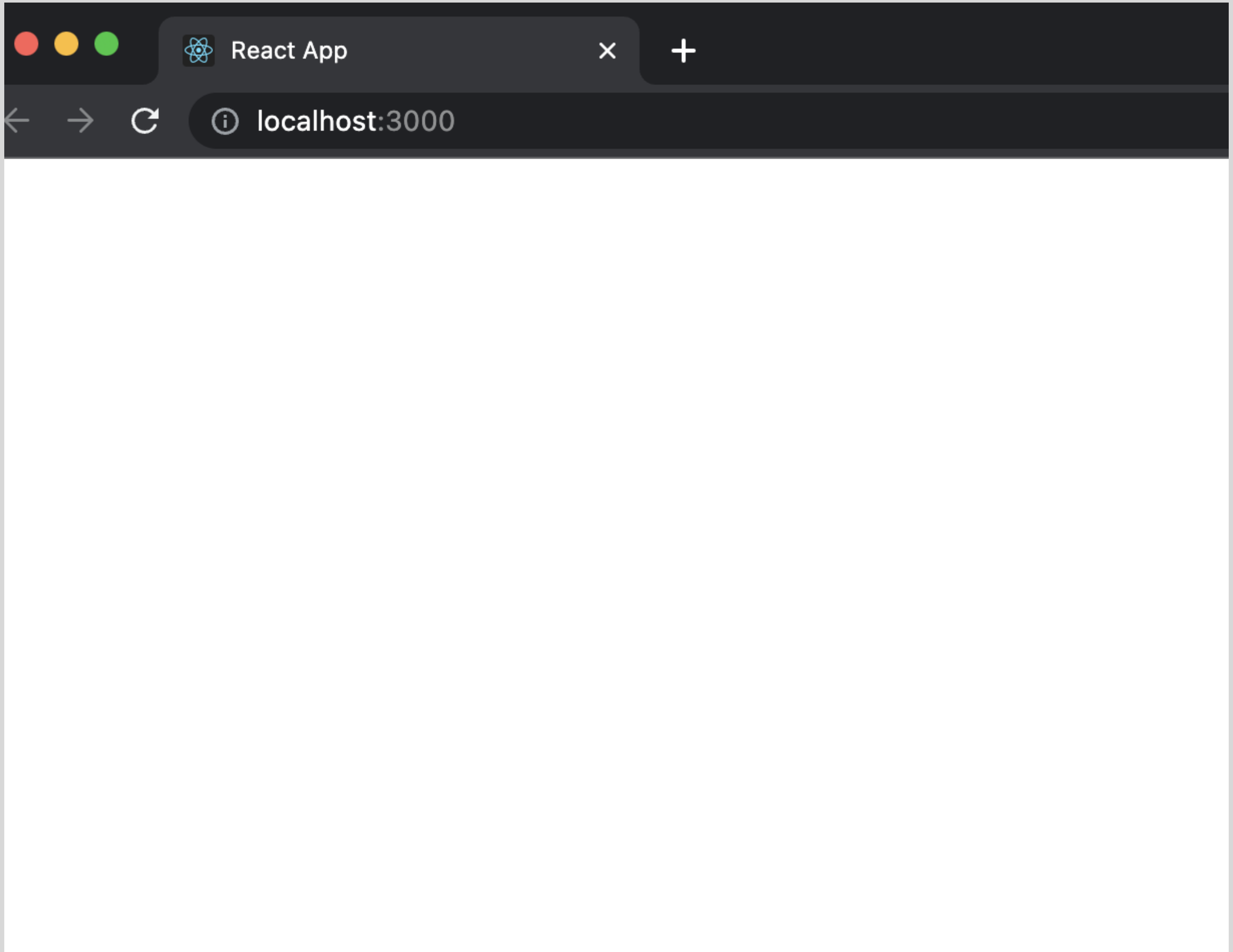
Learn React

Directory Structure



```
PORTFOLIO-APP
  > node_modules
  > public
  v src                      ●
      # App.css              U
      JS App.js              U
      JS App.test.js         U
      # index.css            U
      JS index.js            U
      logo.svg               U
      JS reportV
      JS setupTests.js
    .gitignore
    { } package-lock.json
    { } package.json
    ⓘ README.md
```

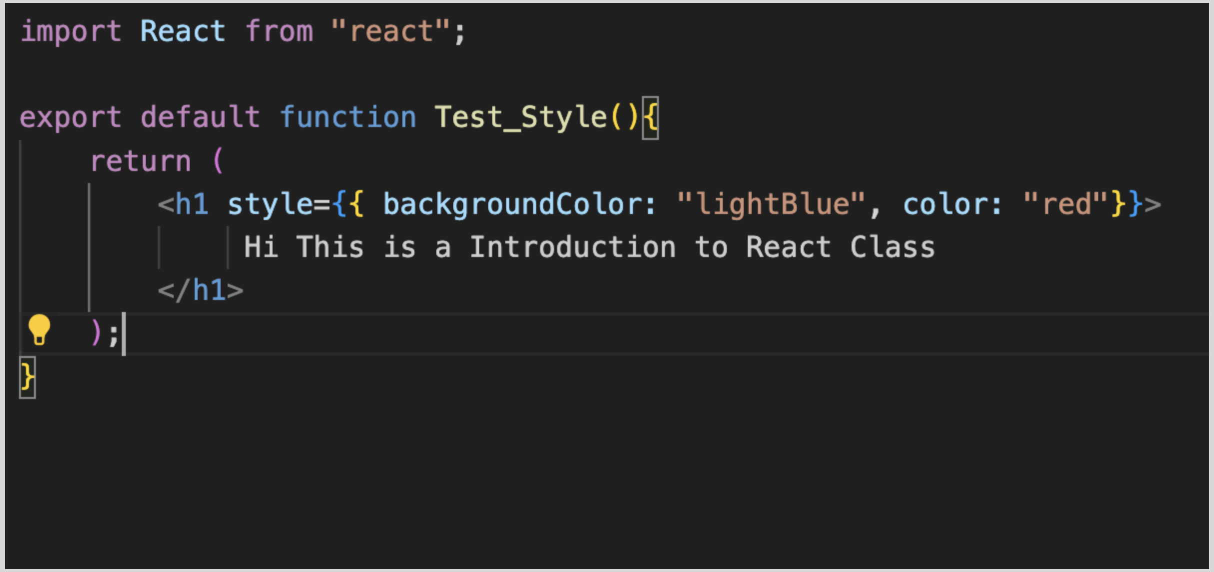
~/Desktop/Research(Sec)/VR:AR/gitre
Tracking-Analysis



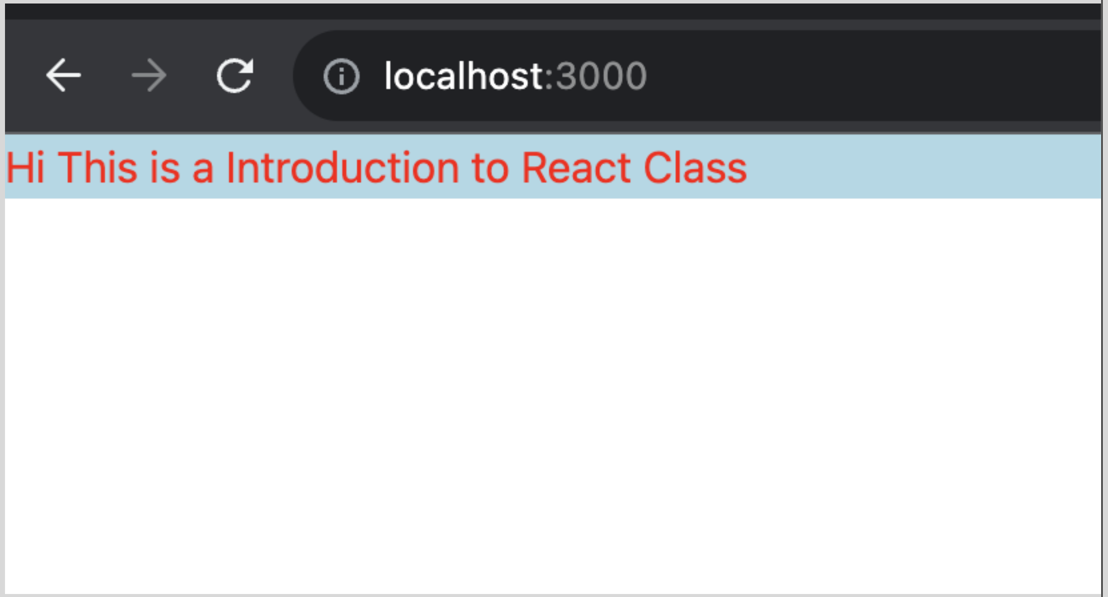
React App
localhost:3000

Add some Styles to Our React Pages

Inline Style

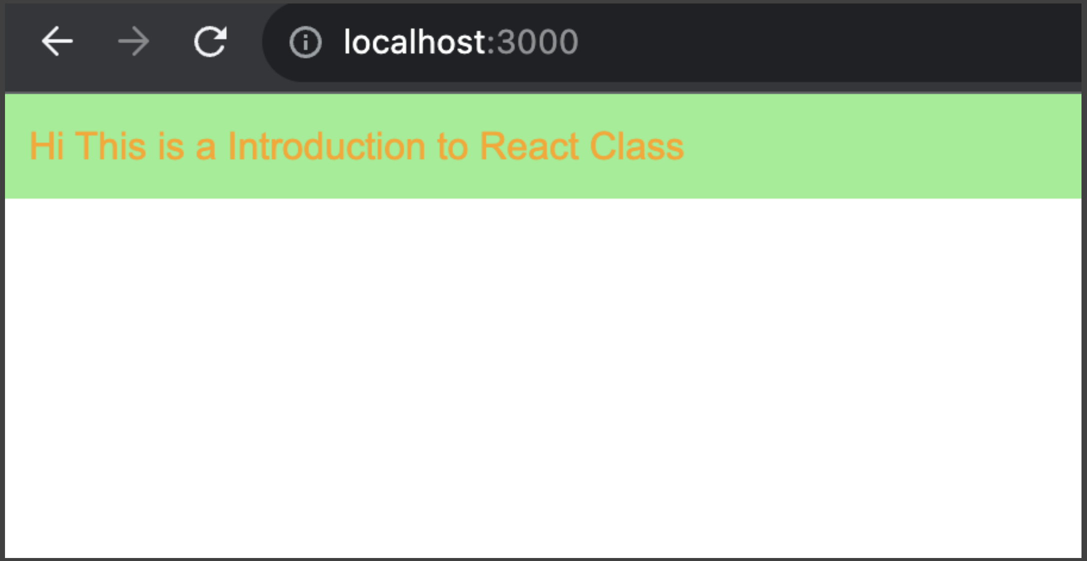


```
import React from "react";

export default function Test_Style(){
    return (
        <h1 style={{ backgroundColor: "lightBlue", color: "red"}}>
            Hi This is a Introduction to React Class
        </h1>
    );
}
```

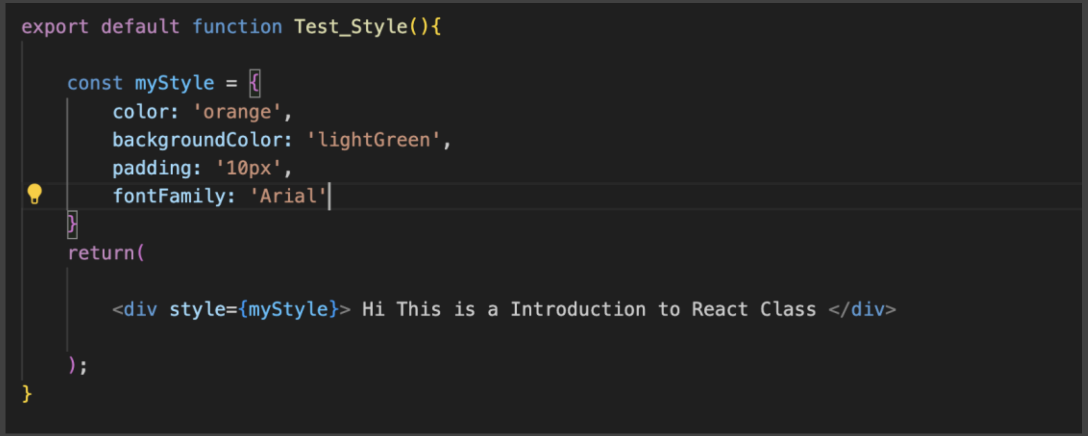


Hi This is a Introduction to React Class

Property Names are camelCased



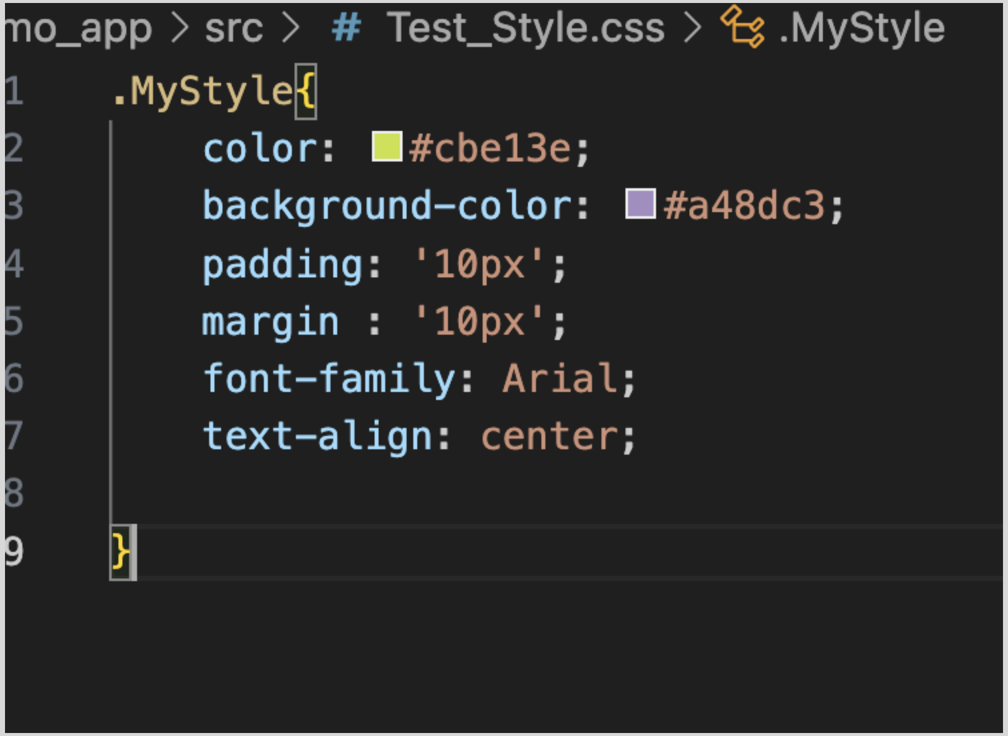
Hi This is a Introduction to React Class



```
export default function Test_Style(){

    const myStyle = {
        color: 'orange',
        backgroundColor: 'lightGreen',
        padding: '10px',
        fontFamily: 'Arial'
    }
    return(

        <div style={myStyle}> Hi This is a Introduction to React Class </div>

    );
}
```

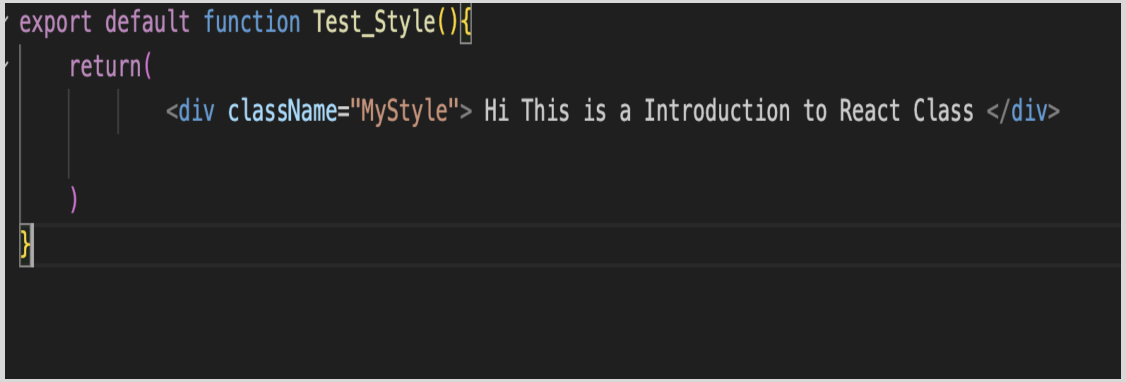
USING JAVASCRIPT OBJECT

CSS Stylesheets

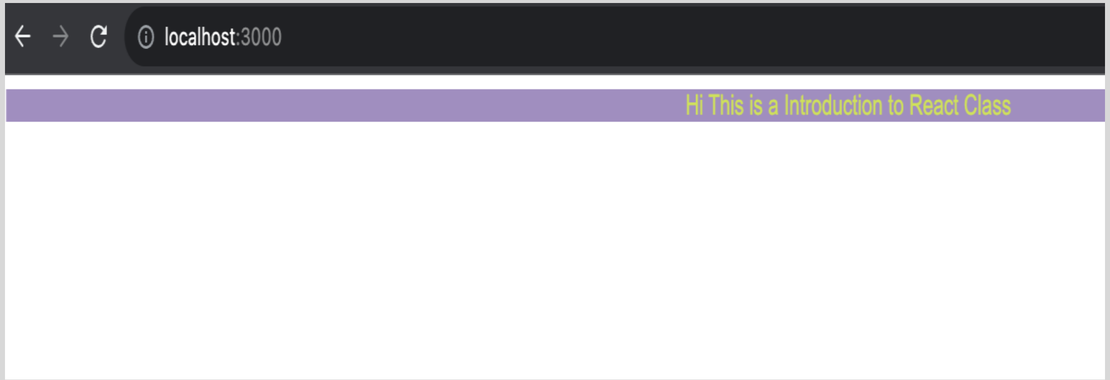


mo_app > src > # Test_Style.css > .MyStyle

```css
.MyStyle{
    color: #cbe13e;
    background-color: #a48dc3;
    padding: '10px';
    margin : '10px';
    font-family: Arial;
    text-align: center;

}
```

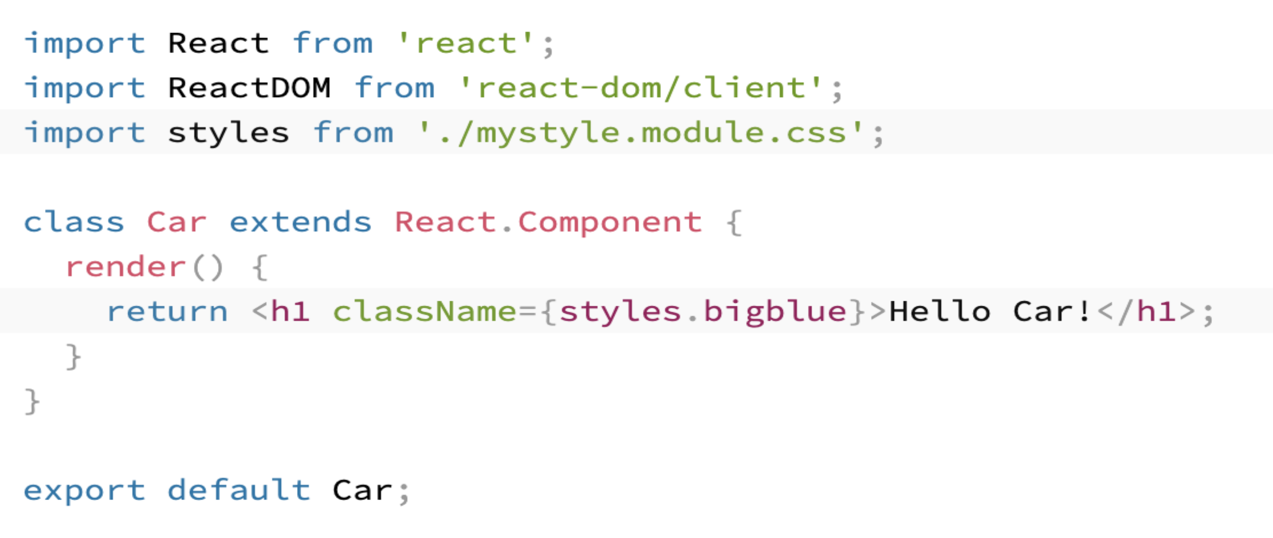


```jsx
export default function Test_Style(){
    return(
        <div className="MyStyle"> Hi This is a Introduction to React Class </div>

    )
}
```

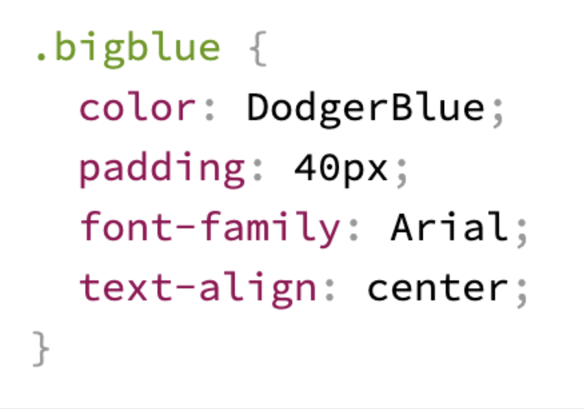


Hi This is a Introduction to React Class

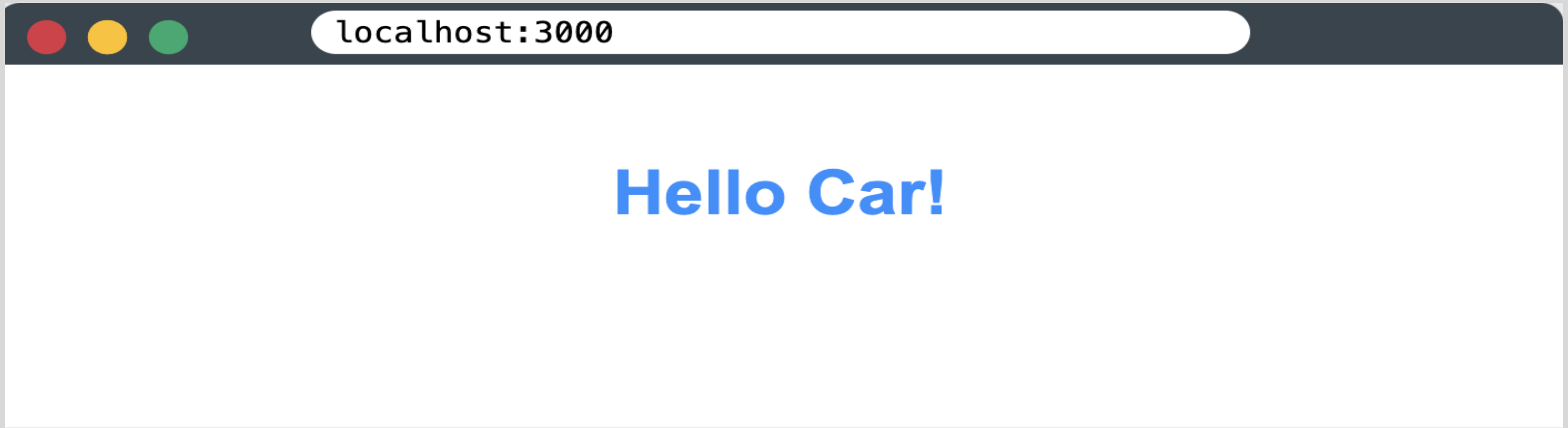
You can also use .module.css



```
import React from 'react';
import ReactDOM from 'react-dom/client';
import styles from './mystyle.module.css';

class Car extends React.Component {
  render() {
    return <h1 className={styles.bigblue}>Hello Car!</h1>;
  }
}

export default Car;
```

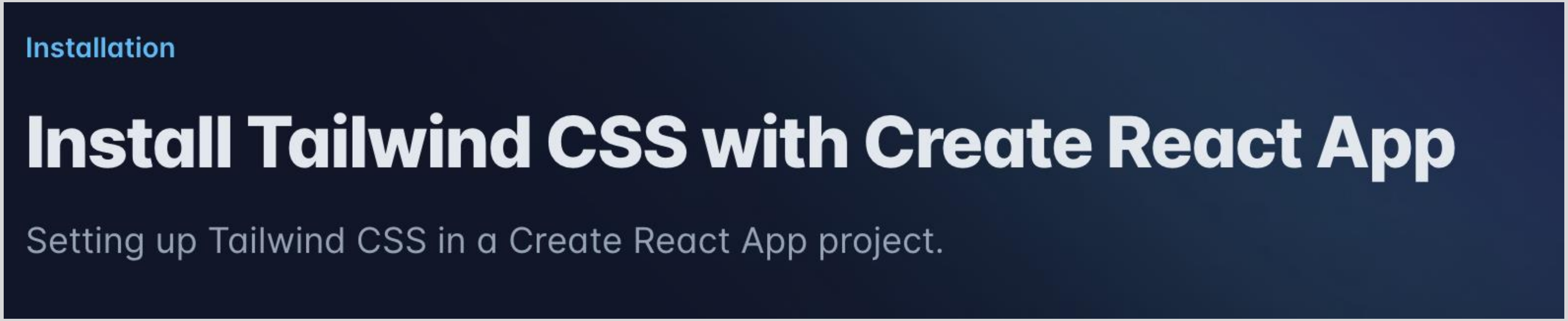


```
.bigblue {
  color: DodgerBlue;
  padding: 40px;
  font-family: Arial;
  text-align: center;
}
```



Hello Car!

What If You don’t want to make CSS classes?



Installation

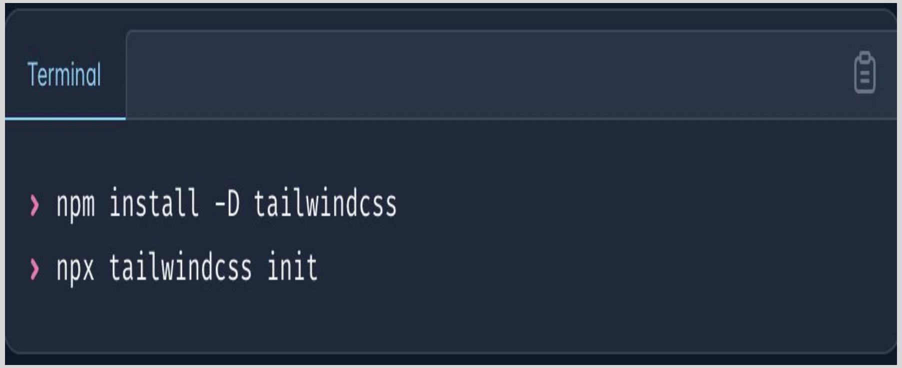
Install Tailwind CSS with Create React App

Setting up Tailwind CSS in a Create React App project.

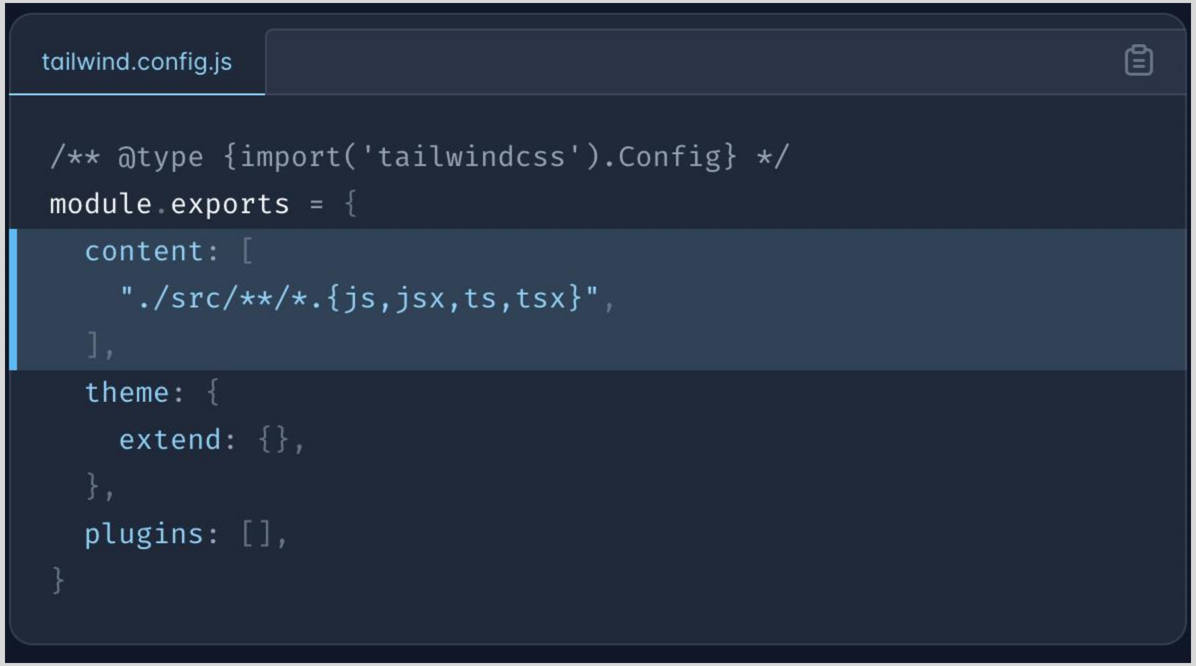
- Easy to use
- Already built classes
- Rapid Development

Tailwind CSS Adding

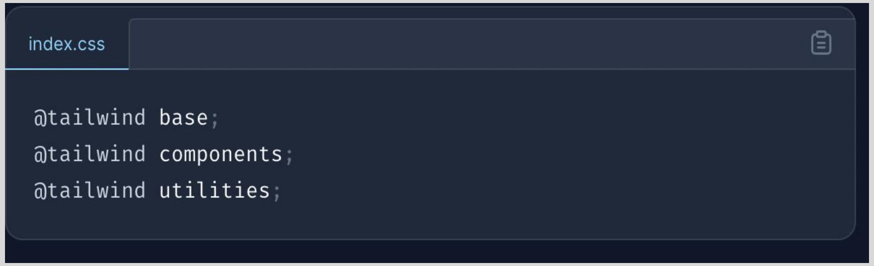
- Tailwind CSS



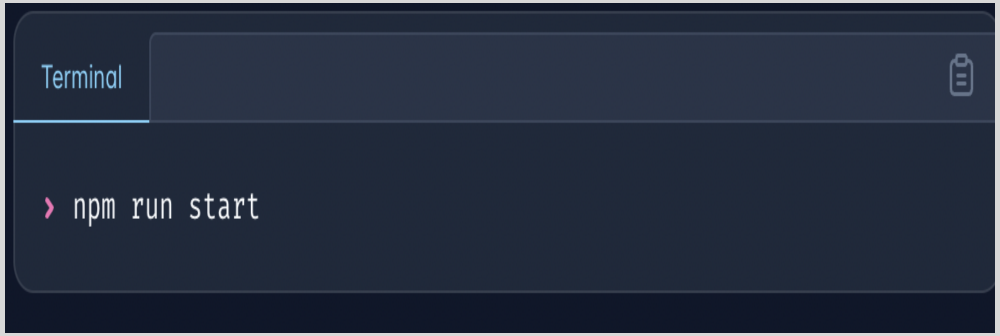
```
❯ npm install -D tailwindcss
❯ npx tailwindcss init
```



```js
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: [
    "./src/**/*.{js,jsx,ts,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

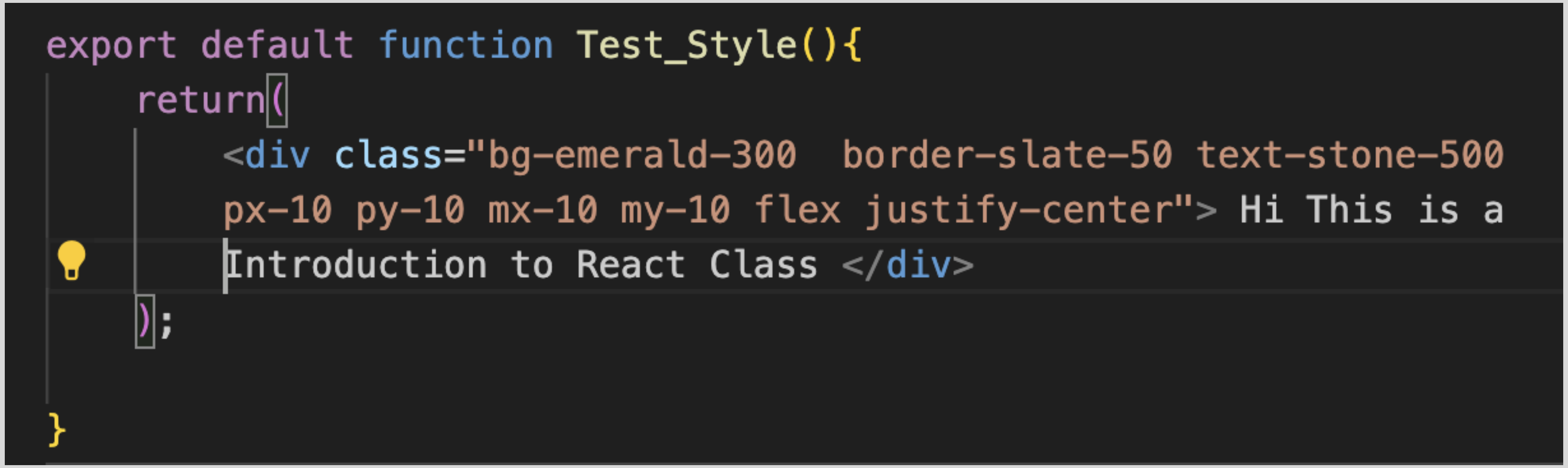


```css
@tailwind base;
@tailwind components;
@tailwind utilities;
```

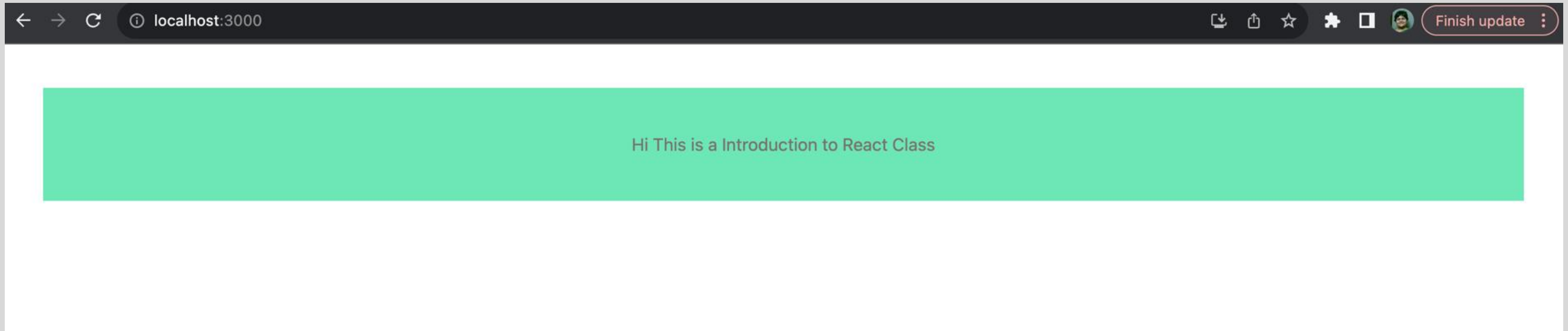


```
❯ npm run start
```

With Tailwind



```
export default function Test_Style(){
    return(
        <div class="bg-emerald-300  border-slate-50 text-stone-500
        px-10 py-10 mx-10 my-10 flex justify-center"> Hi This is a
        Introduction to React Class </div>
    );

}
```

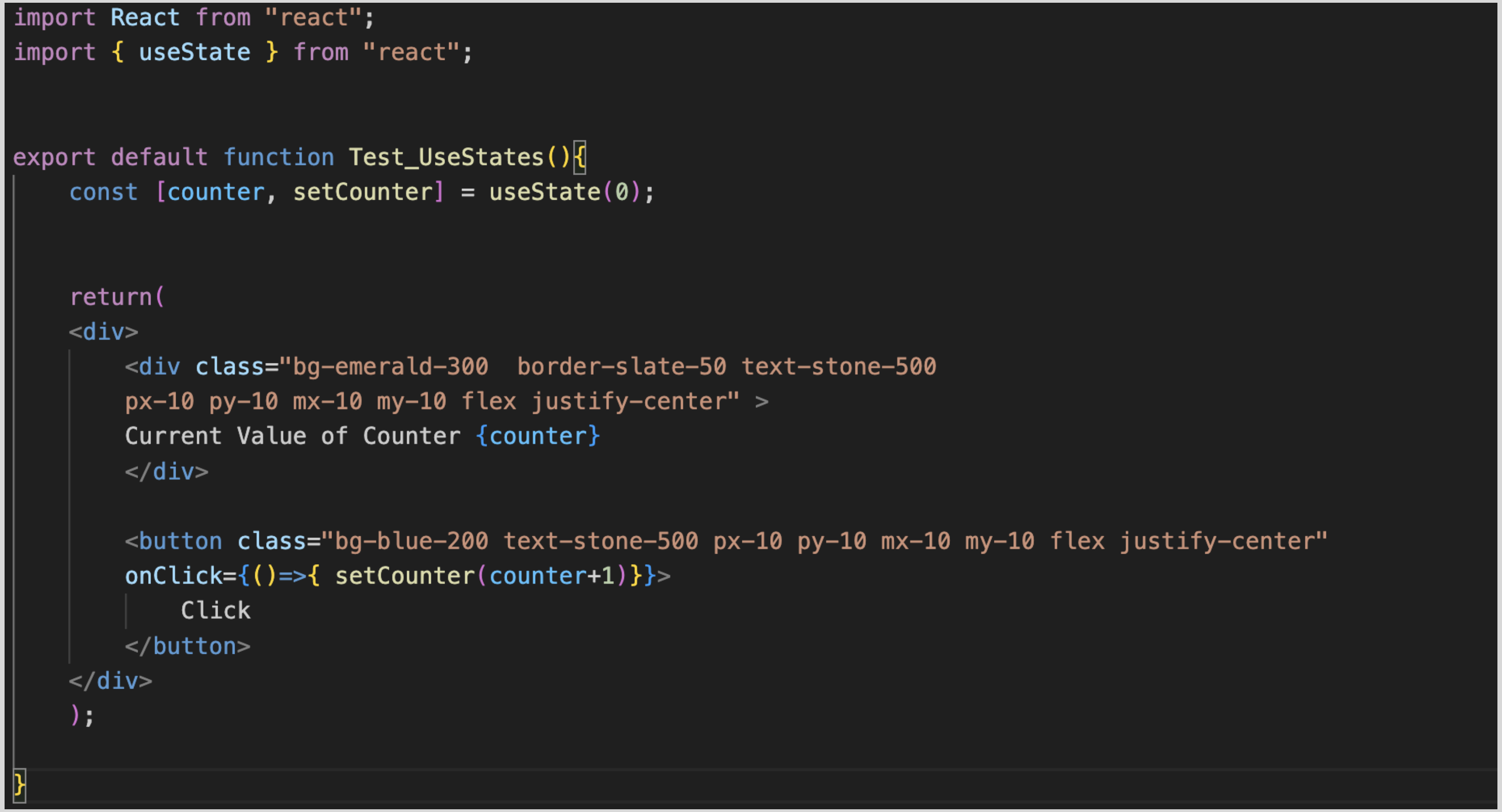


Hi This is a Introduction to React Class

React Hooks

Handle States and Lifecycle Methods

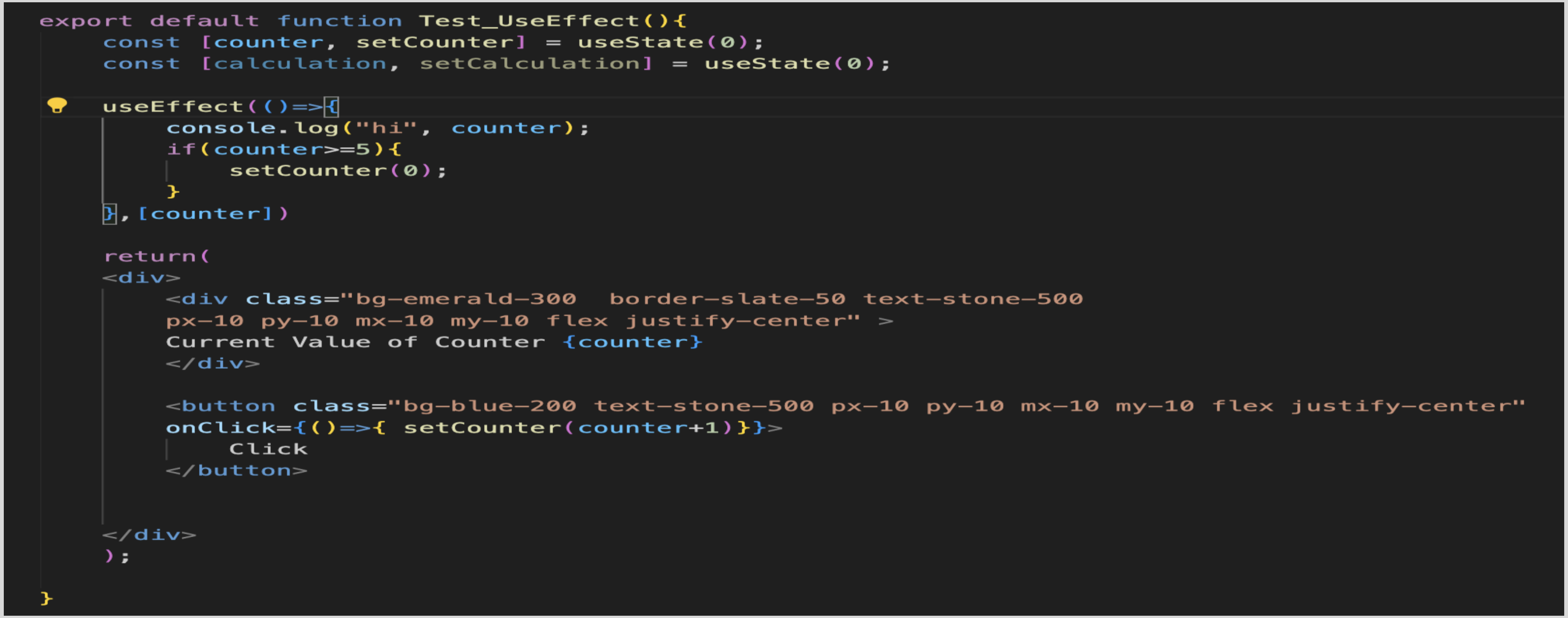
useState – initialize and update object states



```
import React from "react";
import { useState } from "react";


export default function Test_UseStates(){
    const [counter, setCounter] = useState(0);


    return(
    <div>
        <div class="bg-emerald-300  border-slate-50 text-stone-500
        px-10 py-10 mx-10 my-10 flex justify-center" >
        Current Value of Counter {counter}
        </div>

        <button class="bg-blue-200 text-stone-500 px-10 py-10 mx-10 my-10 flex justify-center"
        onClick={()=>{ setCounter(counter+1)}}>
            Click
        </button>
    </div>
    );

}
```

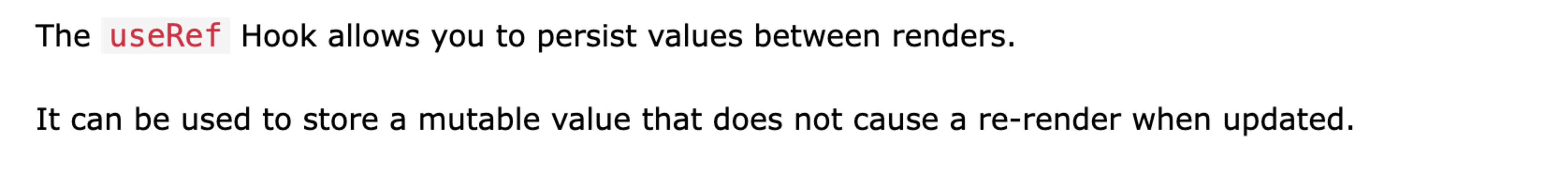
Let’s check the output

useEffect - Perform Side Effects In Component



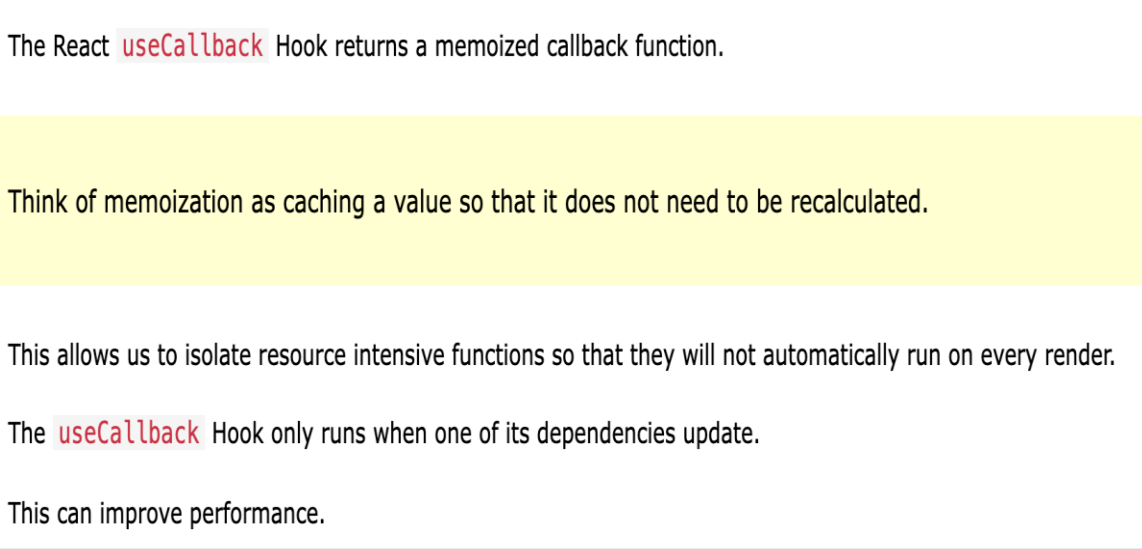
```
export default function Test_UseEffect(){
    const [counter, setCounter] = useState(0);
    const [calculation, setCalculation] = useState(0);

    useEffect(()=>{
        console.log("hi", counter);
        if(counter>=5){
            setCounter(0);
        }
    },[counter])

    return(
    <div>
        <div class="bg-emerald-300  border-slate-50 text-stone-500
        px-10 py-10 mx-10 my-10 flex justify-center" >
        Current Value of Counter {counter}
        </div>

        <button class="bg-blue-200 text-stone-500 px-10 py-10 mx-10 my-10 flex justify-center"
        onClick={()=>{ setCounter(counter+1)}}>
            Click
        </button>


    </div>
    );

}
```

Other Hooks



The `useRef` Hook allows you to persist values between renders.

It can be used to store a mutable value that does not cause a re-render when updated.



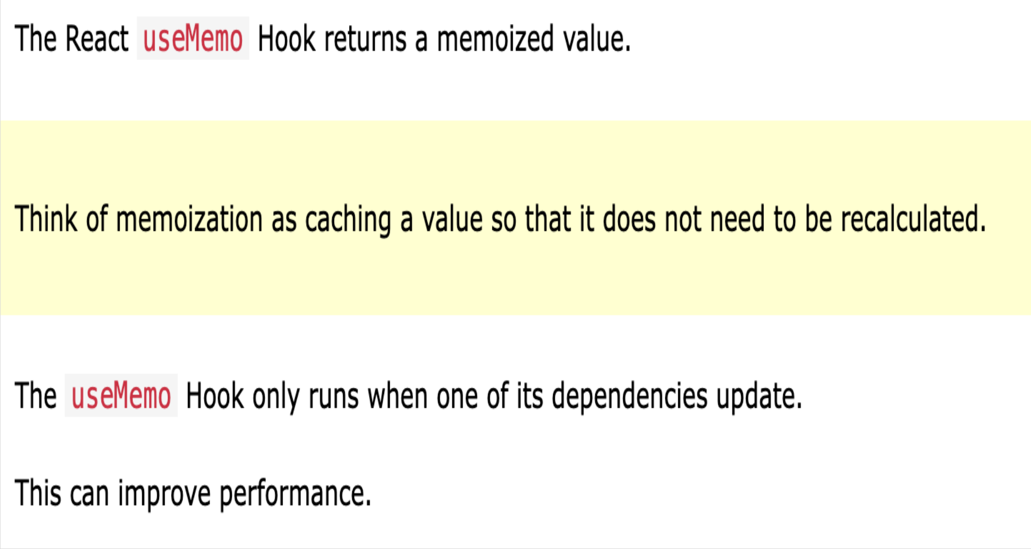
The React `useCallback` Hook returns a memoized callback function.

Think of memoization as caching a value so that it does not need to be recalculated.

This allows us to isolate resource intensive functions so that they will not automatically run on every render.

The `useCallback` Hook only runs when one of its dependencies update.

This can improve performance.



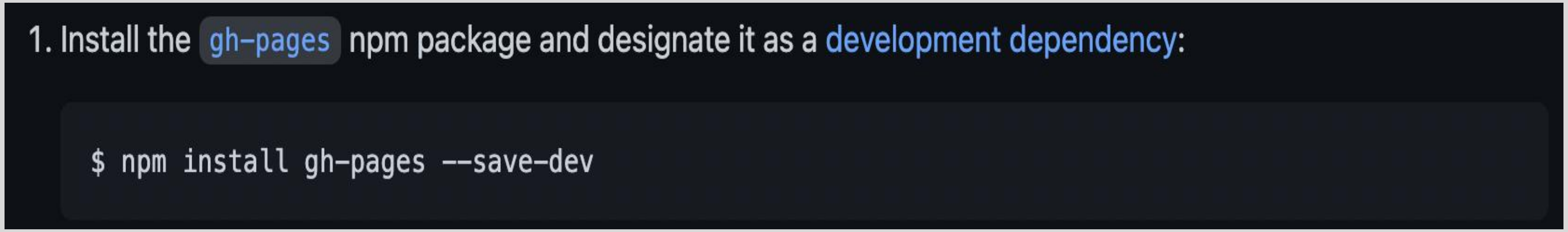
The React `useMemo` Hook returns a memoized value.

Think of memoization as caching a value so that it does not need to be recalculated.

The `useMemo` Hook only runs when one of its dependencies update.

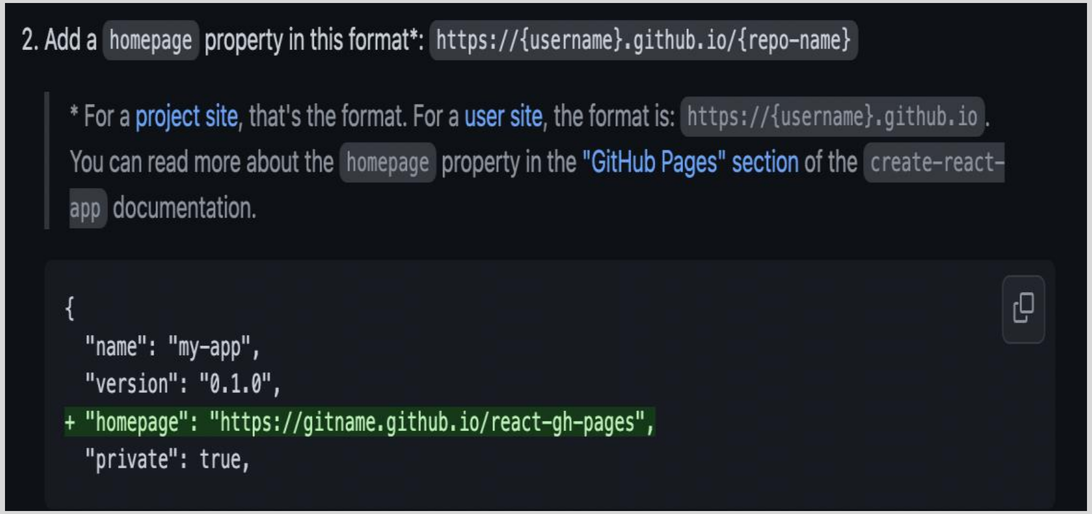
This can improve performance.

Hosting on GitHub



1. Install the `gh-pages` npm package and designate it as a development dependency:

```
$ npm install gh-pages --save-dev
```

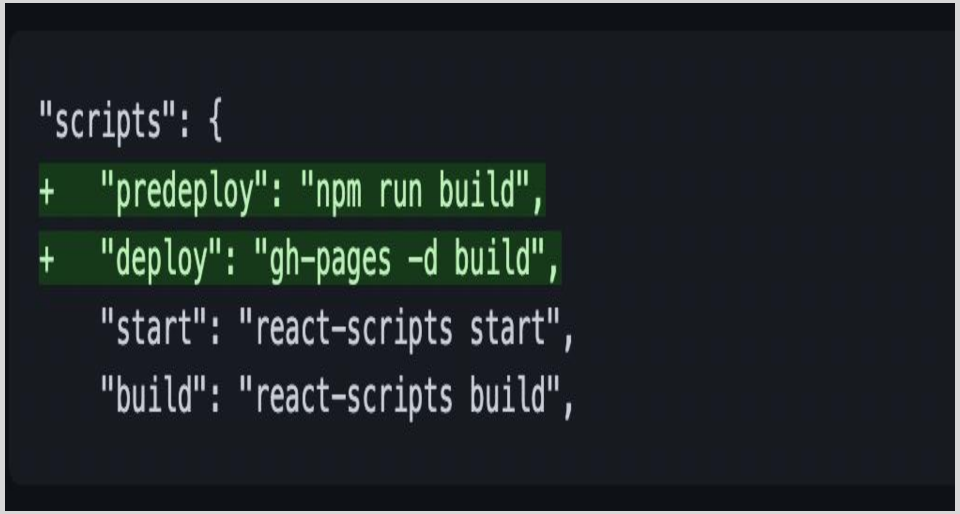


2. Add a `homepage` property in this format*: `https://{username}.github.io/{repo-name}`

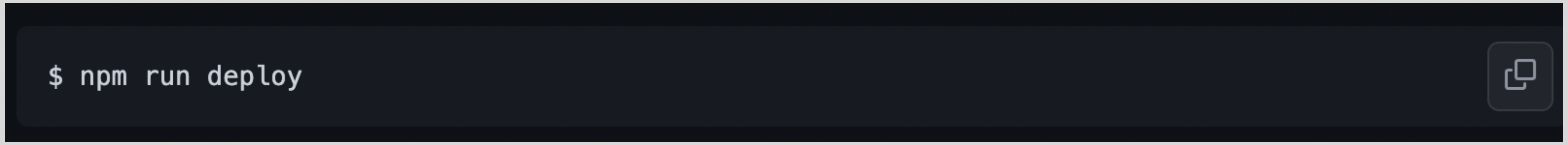
> * For a project site, that's the format. For a user site, the format is: `https://{username}.github.io`. You can read more about the `homepage` property in the "GitHub Pages" section of the `create-react-app` documentation.

```
{
  "name": "my-app",
  "version": "0.1.0",
+ "homepage": "https://gitname.github.io/react-gh-pages",
  "private": true,
```

But First Create A Repo!!!!



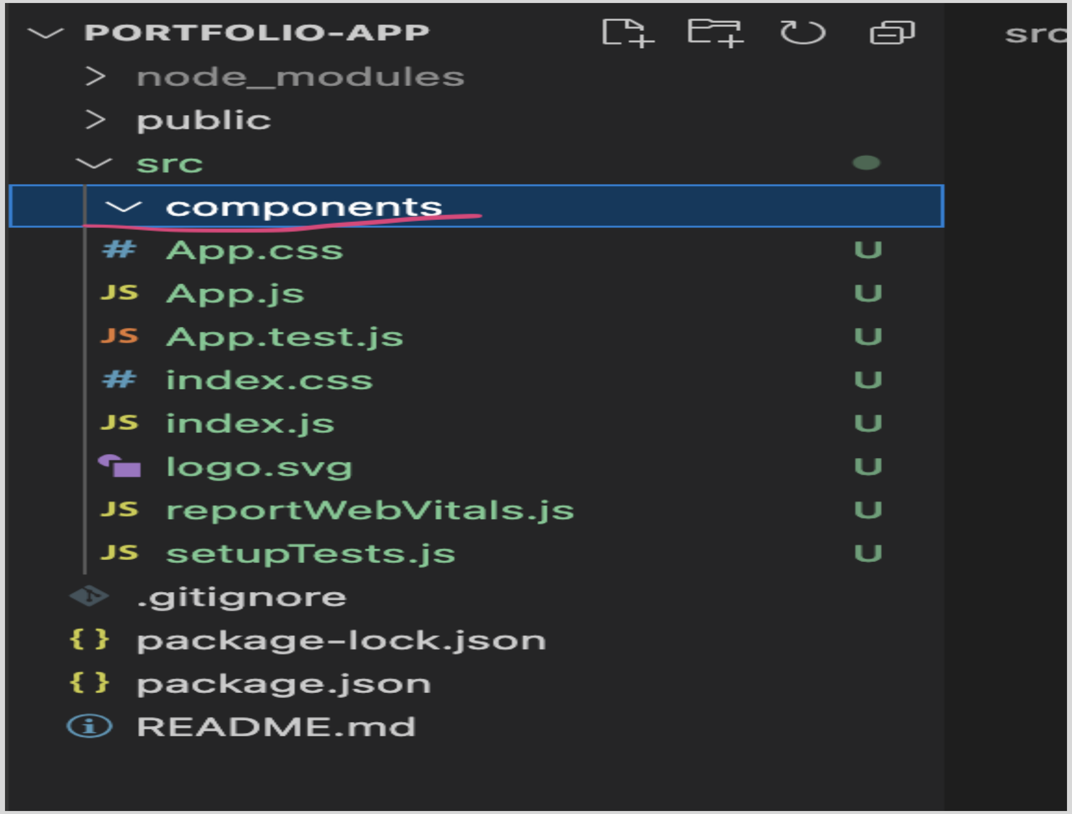
```
"scripts": {
+   "predeploy": "npm run build",
+   "deploy": "gh-pages -d build",
    "start": "react-scripts start",
    "build": "react-scripts build",
```



```
$ npm run deploy
```

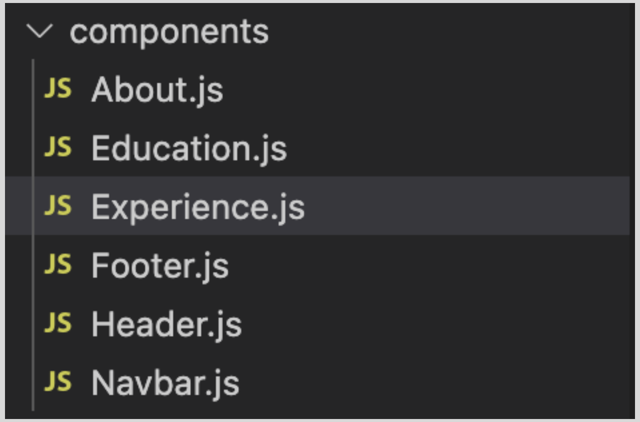
Let’s Design a Portfolio

Create a Folder Under Src

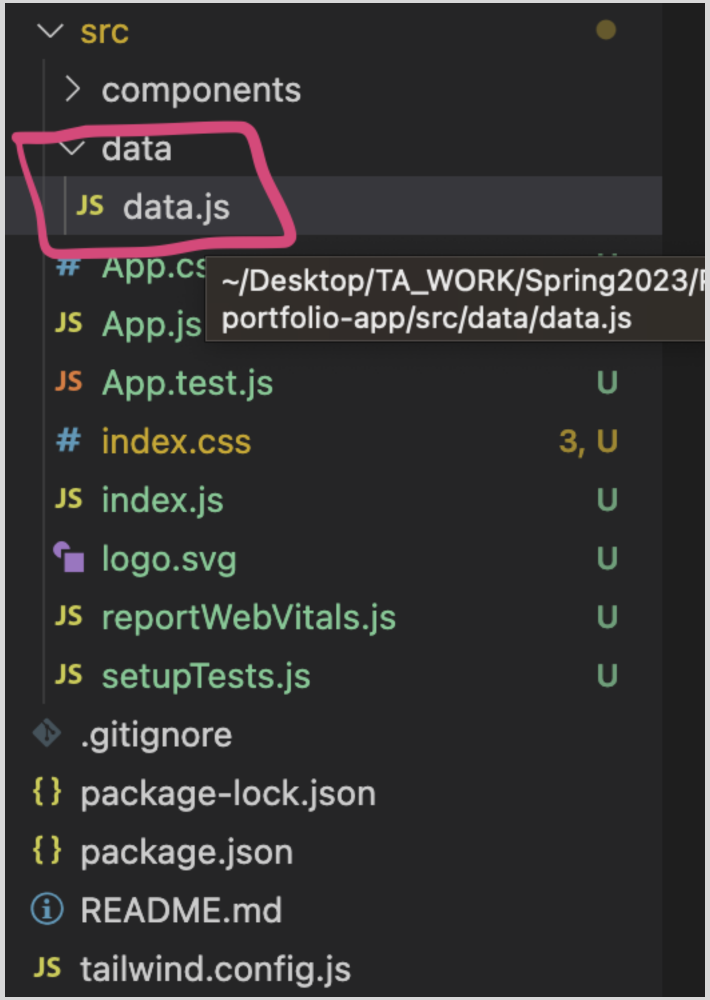


```
∨ PORTFOLIO-APP                                     src
  > node_modules
  > public
  ∨ src                                    ●
    ∨ components
    # App.css                              U
    JS App.js                              U
    JS App.test.js                         U
    # index.css                            U
    JS index.js                            U
    logo.svg                               U
    JS reportWebVitals.js                  U
    JS setupTests.js                       U
  .gitignore
  { } package-lock.json
  { } package.json
  ⓘ README.md
```

We are going to store different components here.



```
∨ components
  JS About.js
  JS Education.js
  JS Experience.js
  JS Footer.js
  JS Header.js
  JS Navbar.js
```



```
∨ src
  > components
  ∨ data
    JS data.js
  # App.cs  ~/Desktop/TA_WORK/Spring2023/
  JS App.js  portfolio-app/src/data/data.js
  JS App.test.js          U
  # index.css          3, U
  JS index.js             U
  logo.svg                U
  JS reportWebVitals.js   U
  JS setupTests.js        U
.gitignore
{} package-lock.json
{} package.json
ⓘ README.md
JS tailwind.config.js
```

Create Data folder and data.js

Goal – To loosely attach data with code

CREATE A REACT APP

Exercise

- Write a small static HTML in Header.js with a welcome message to your portfolio
- Call the Header.js from App.js
- Check the changes in the app

Exercise – Publish Your App

- Create a GitHub Repo
- Link this Repo to GitHub
- Install gh-pages if you have not
- Publish

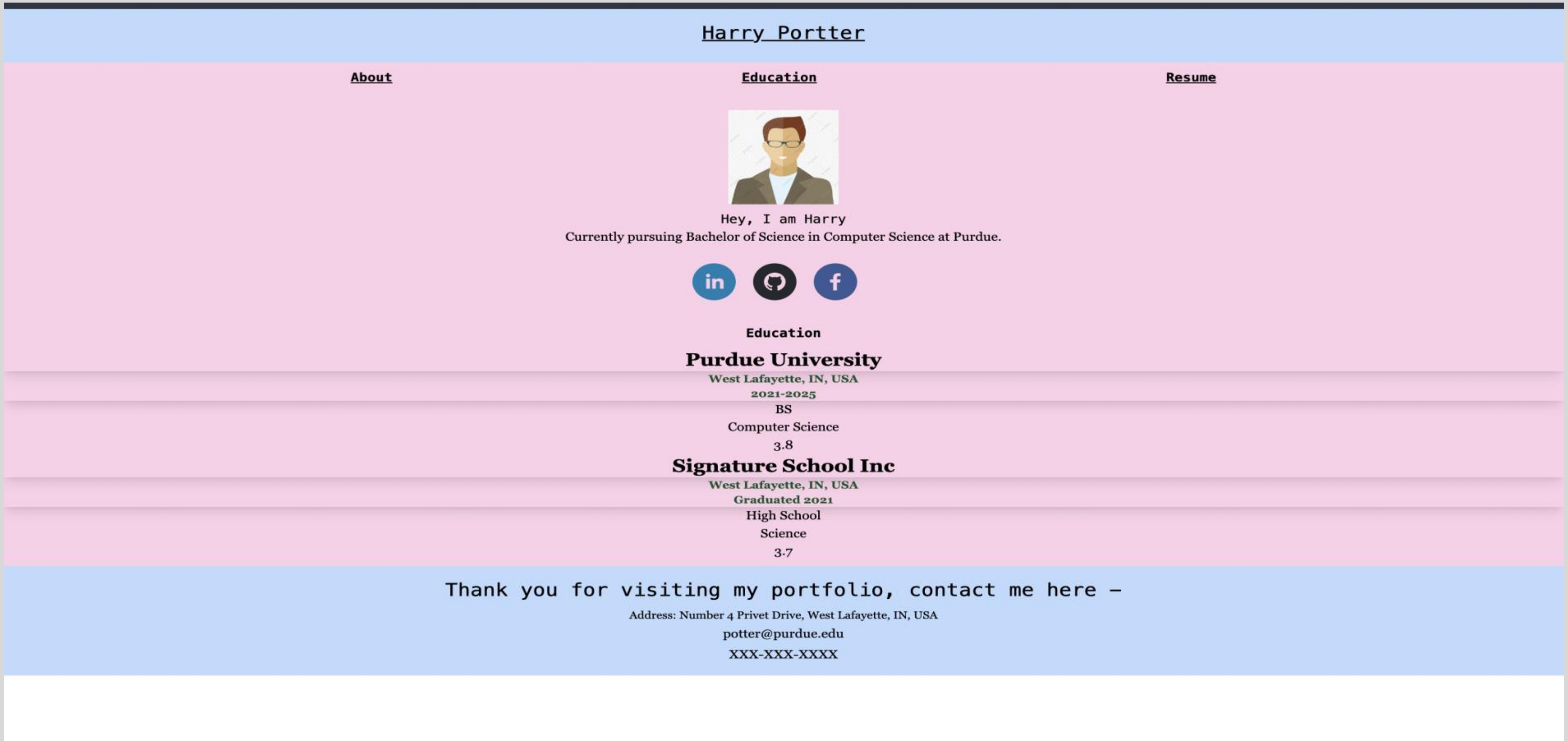
Exercise

- Write a small static HTML in Footer.js with your contact information

- Call the Footer.js from App.js

- Check the changes in the app

Adding Navbar, About and Education



Harry Portter

About Education Resume

Hey, I am Harry

Currently pursuing Bachelor of Science in Computer Science at Purdue.

Education

Purdue University

West Lafayette, IN, USA

2021-2025

BS

Computer Science

3.8

Signature School Inc

West Lafayette, IN, USA

Graduated 2021

High School

Science

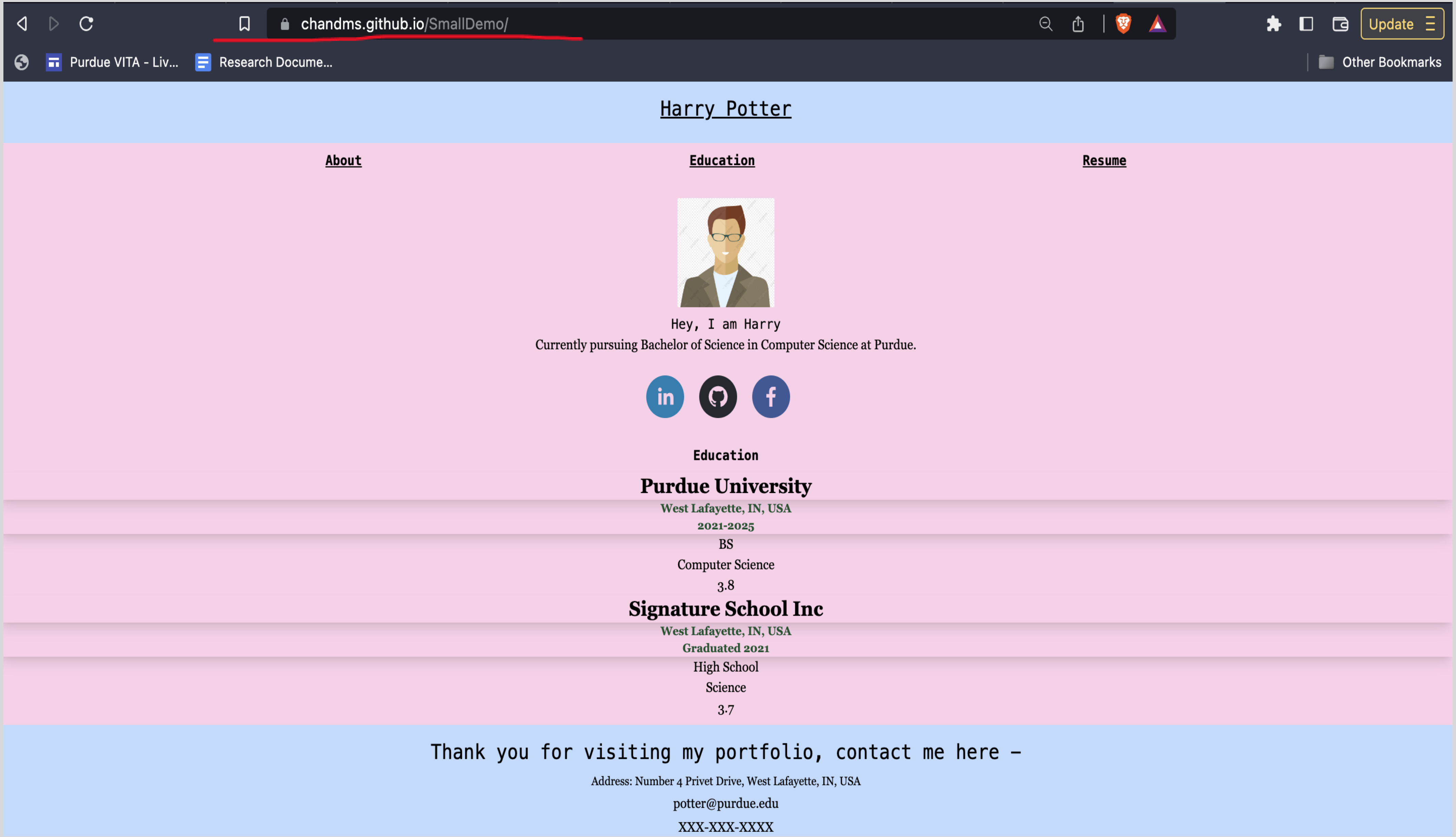
3.7

Thank you for visiting my portfolio, contact me here –

Address: Number 4 Privet Drive, West Lafayette, IN, USA

potter@purdue.edu

XXX-XXX-XXXX



Harry Potter

About

Education

Resume

Hey, I am Harry

Currently pursuing Bachelor of Science in Computer Science at Purdue.

Education

Purdue University

West Lafayette, IN, USA

2021-2025

BS

Computer Science

3.8

Signature School Inc

West Lafayette, IN, USA

Graduated 2021

High School

Science

3.7

Thank you for visiting my portfolio, contact me here -

Address: Number 4 Privet Drive, West Lafayette, IN, USA

potter@purdue.edu

XXX-XXX-XXXX

THANK YOU!!