React JS & Hooks

MA003

ALYANI MAMAD B.

Agenda

01. Introduction

- Basic
- Open Source
- Who Uses React

02. Features Of React

- Responsive Design
- Component Based
- Reusable code

03. Installation & Run

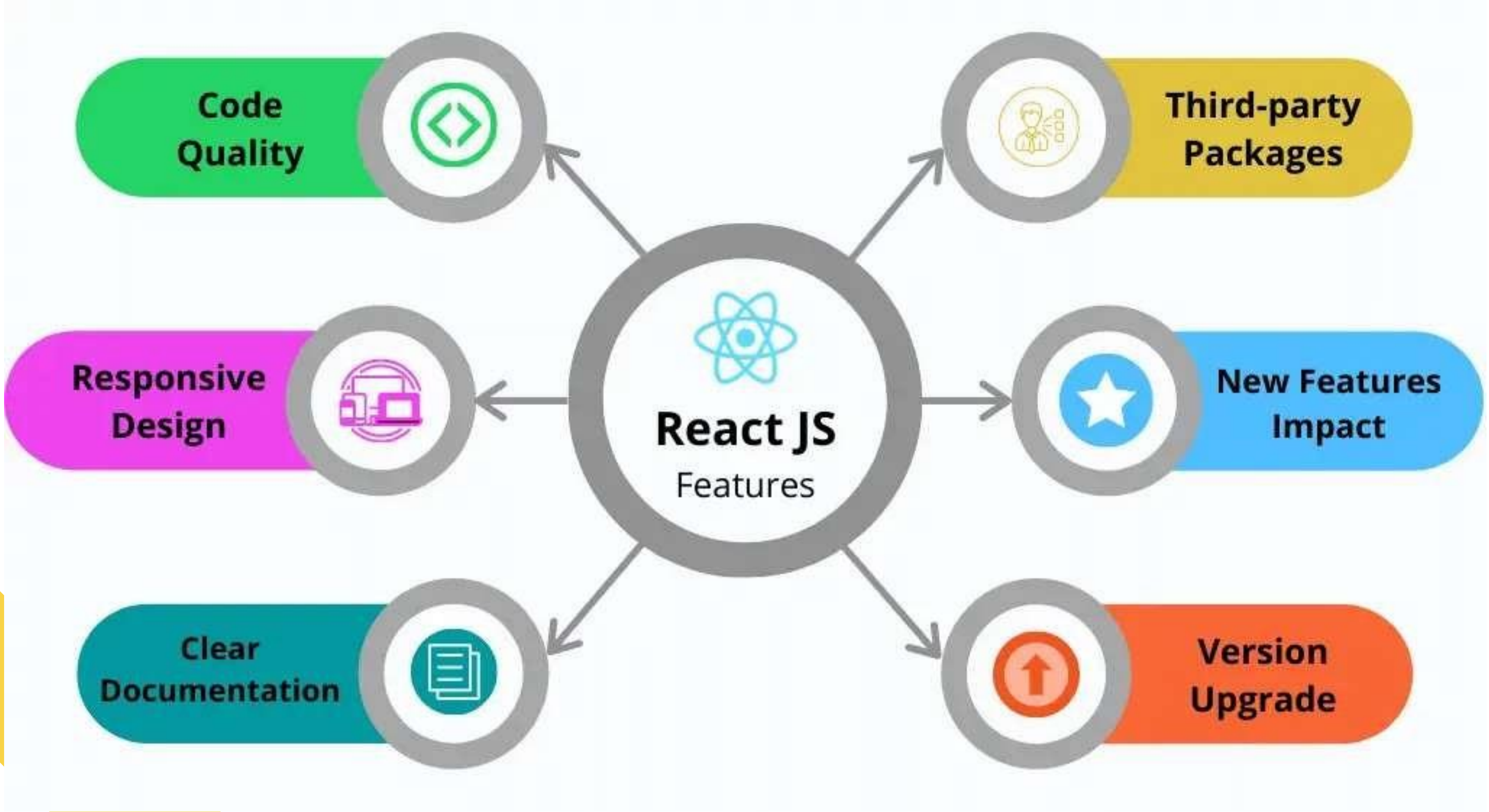
04. React App Structure

05. Why React?

Introduction

- React is an open source, JavaScript library for developing user interface (UI) in web application.
- React is developed and released by **Meta.**
- I assume that the you have the basic knowledge in HTML, CSS and JavaScript ES6 concepts.
- React work on component based approach.
- React is flexible and can be used in variety projects.
- React Office Website : https://reactjs.org/

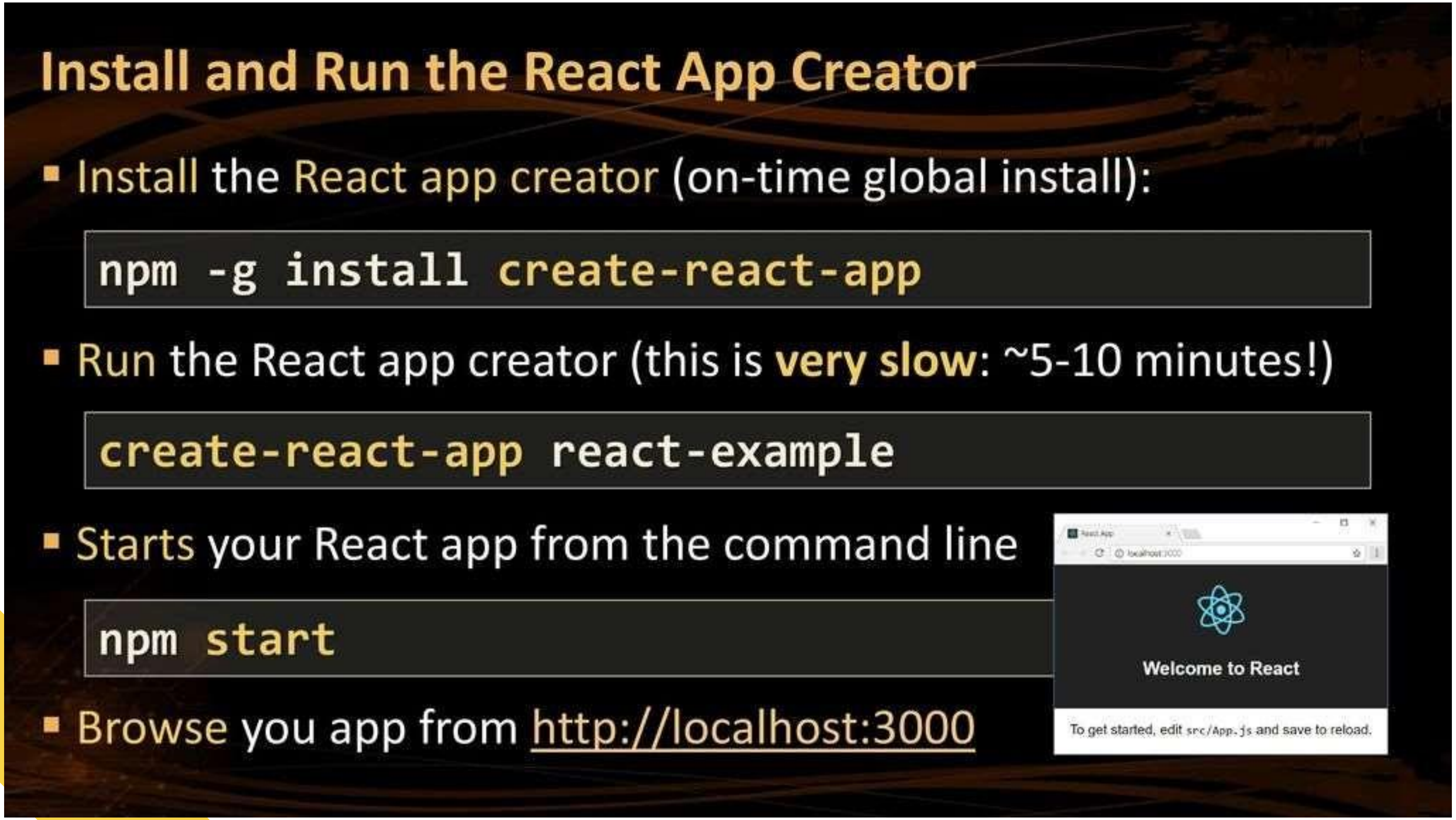
Features React Js



List of App Using Reactjs

1. Facebook
2. Instagram
3. Netflix
4. New York Times
5. Discovery VR
6. WhatsApp
7. Myntra
8. Discord
9. Airbnb
10. Khan Academy

Installation & Run App



Install and Run the React App Creator

- Install the React app creator (on-time global install):

```
npm -g install create-react-app
```

- Run the React app creator (this is **very slow**: ~5-10 minutes!)

```
create-react-app react-example
```

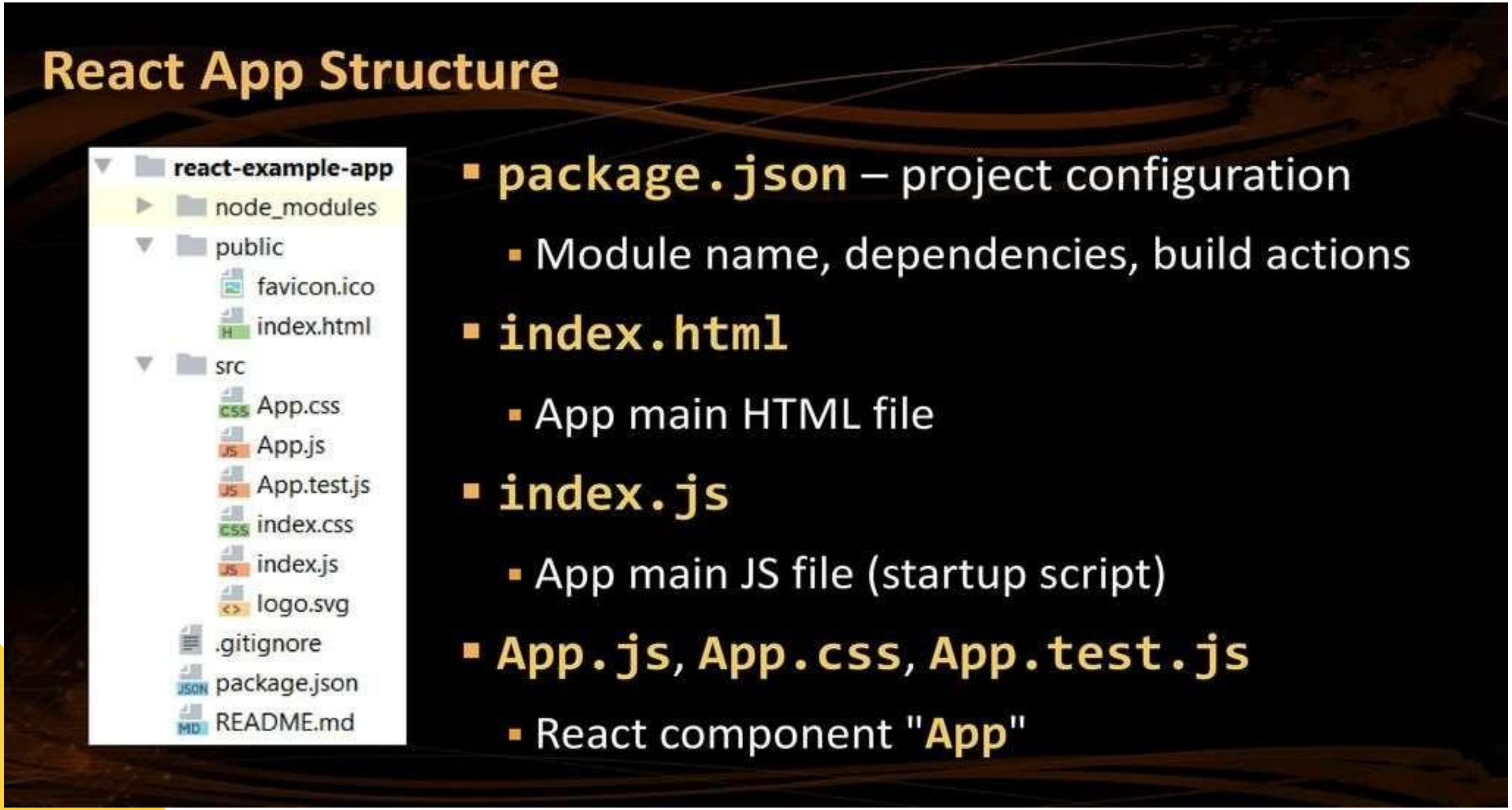
- Starts your React app from the command line

```
npm start
```

- Browse you app from http://localhost:3000

To get started, edit `src/App.js` and save to reload.

React App Structure



React App Structure

```
▼ react-example-app
  ▶ node_modules
  ▼ public
      favicon.ico
      index.html
  ▼ src
      App.css
      App.js
      App.test.js
      index.css
      index.js
      logo.svg
    .gitignore
    package.json
    README.md
```

- **package.json** – project configuration
  - Module name, dependencies, build actions
- **index.html**
  - App main HTML file
- **index.js**
  - App main JS file (startup script)
- **App.js, App.css, App.test.js**
  - React component "**App**"

Why React?

- Reuse of components.
- It is quite easy to create interactive UI’s.
- It has huge community.
- Component Based Architecture.
- Efficient update and render.
- Excellent cross-platform support(Multiple OS).
- Provides amazing developer tools.
- Makes JavaScript coding easier.

Agenda

Installation Of React

Introduction to JSX

Why We Learn EcmaScript Concept

Arrow Function VS Regular Function

“Hello World” Program

Installation

Install JS Runtime Enviroment

- NodeJS is the platform needed for the ReactJS development.
- https://nodejs.org/en/

npm install –g create-react app

- Everything went well run the command.

create-react-app Hello-world

- It will take some time to install the required dependencies.

cd Hello-world

- Move inside the same folder using the above command.

npm start or Yarn start

- To start your app run the above command .

Introduction To JSX

React is often considered developer-friendly and the major reason for it is JSX.

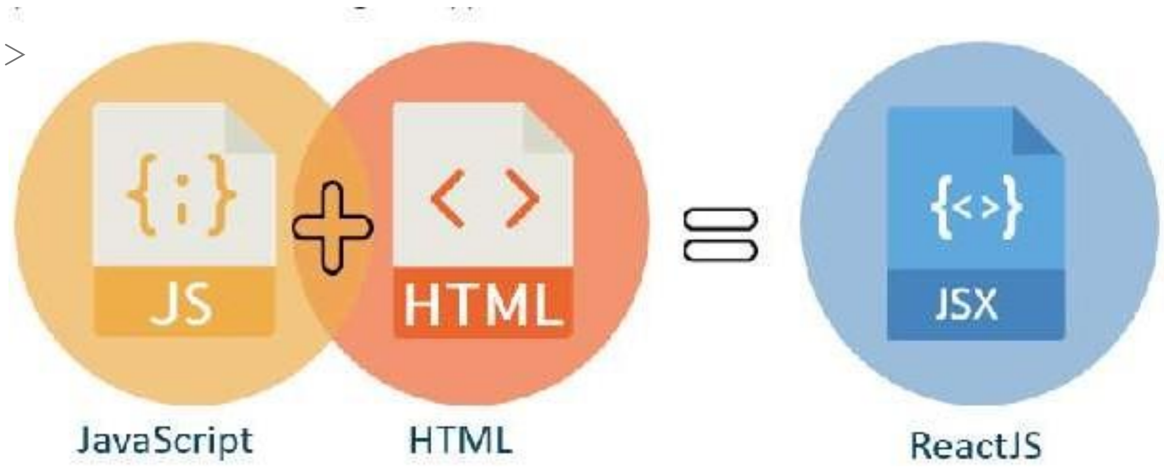
JSX stands for JavaScript XML.

JSX allows us to write HTML in React.

JSX converts HTML tags into react elements.

Example :

```
let name = “Alyani";
let heading = <h1> Hi, my name is {name} </h1>
```



Why We Learn EcmaScript ?

- By using ES6 features, we write less and do more, so the term 'Write less, do more' suits ES6.
- ES6 introduces you many great features such as scope variable, arrow functions, template strings,etc..
- The Main Reason is to Learn React JS.

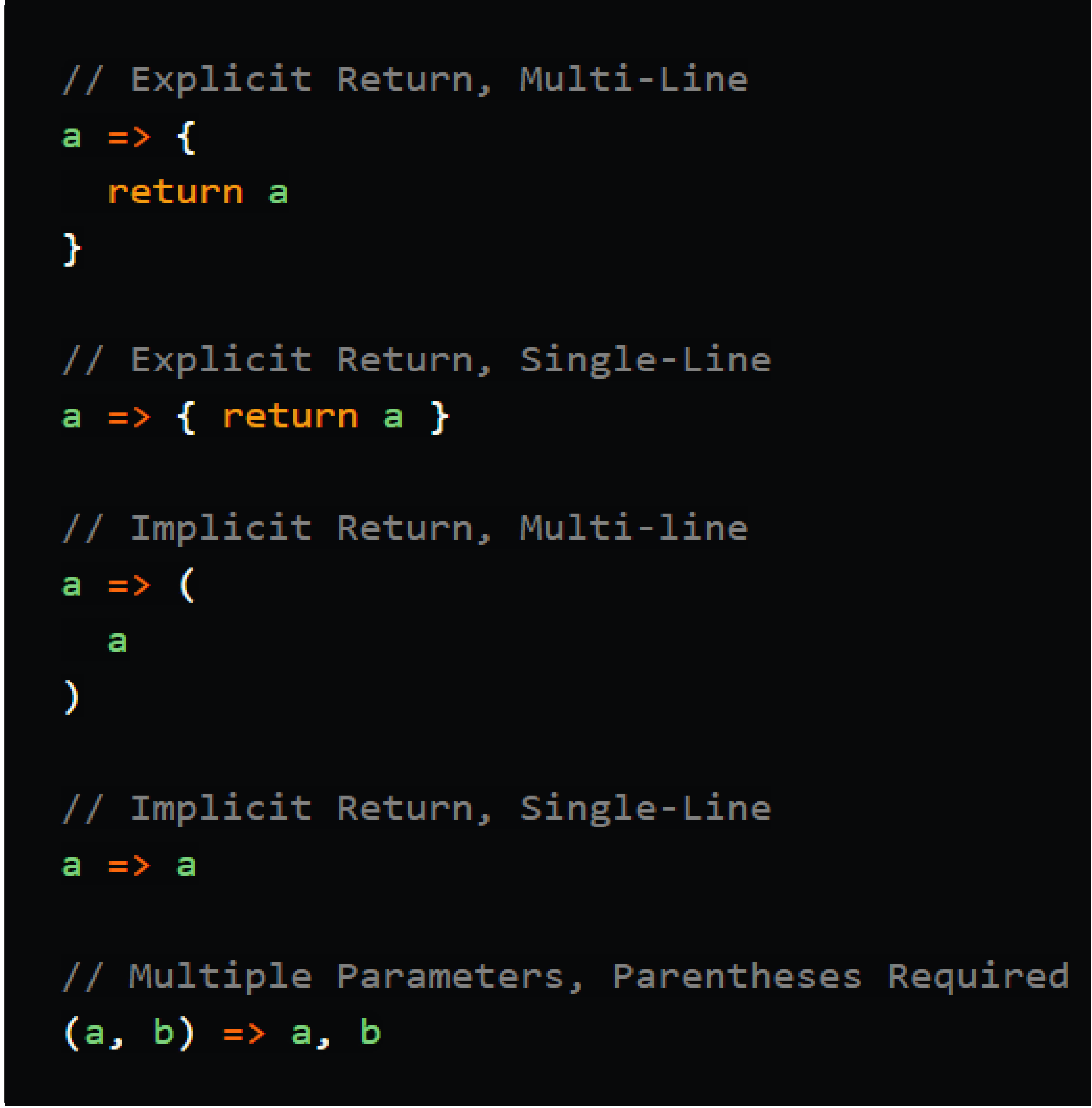
Arrow Function VS Regular Function

1) Syntax

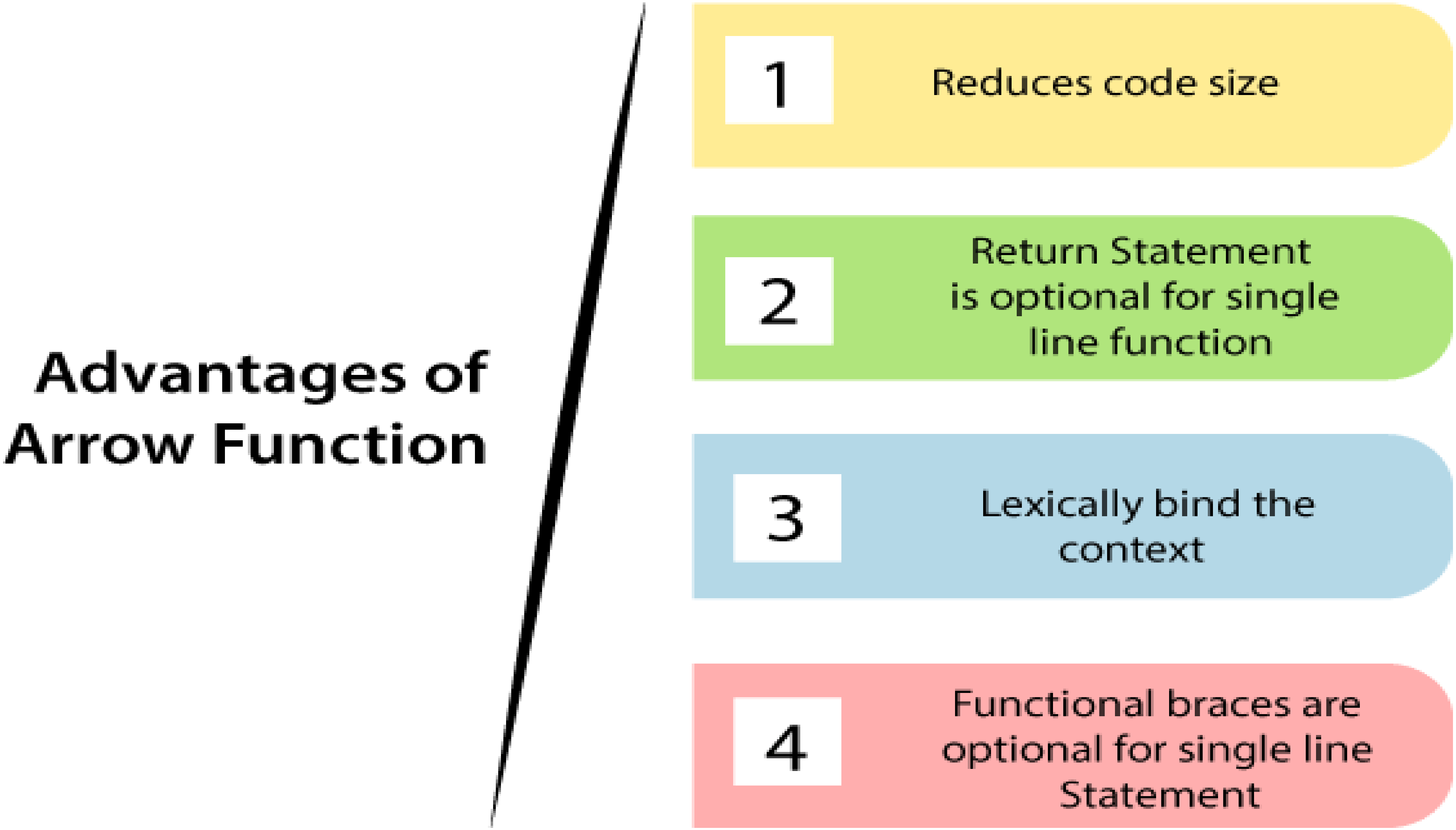
2) Prototype Checking

3) This Keyword

4) Multiple Argument



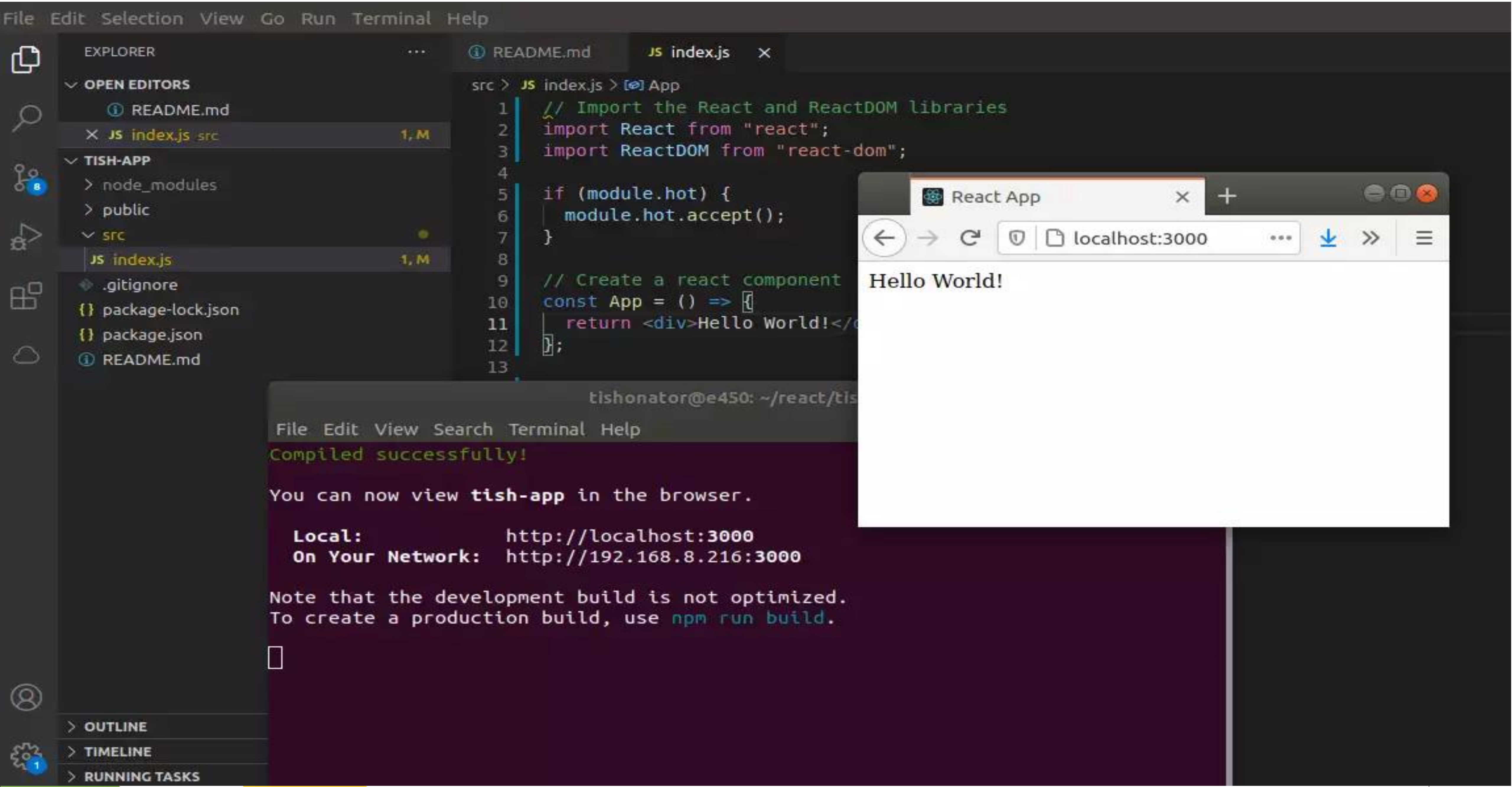
```
// Explicit Return, Multi-Line
a => {
  return a
}

// Explicit Return, Single-Line
a => { return a }

// Implicit Return, Multi-line
a => (
  a
)

// Implicit Return, Single-Line
a => a

// Multiple Parameters, Parentheses Required
(a, b) => a, b
```



Advantages of Arrow Function

1. Reduces code size
2. Return Statement is optional for single line function
3. Lexically bind the context
4. Functional braces are optional for single line Statement

“Hello World” Program



```
// Import the React and ReactDOM libraries
import React from "react";
import ReactDOM from "react-dom";

if (module.hot) {
  module.hot.accept();
}

// Create a react component
const App = () => {
  return <div>Hello World!</
};

```

```
Compiled successfully!

You can now view tish-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.8.216:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Hello World!

Agenda-(Hooks 🦄)

Introduction (Hooks)

Hooks Rules

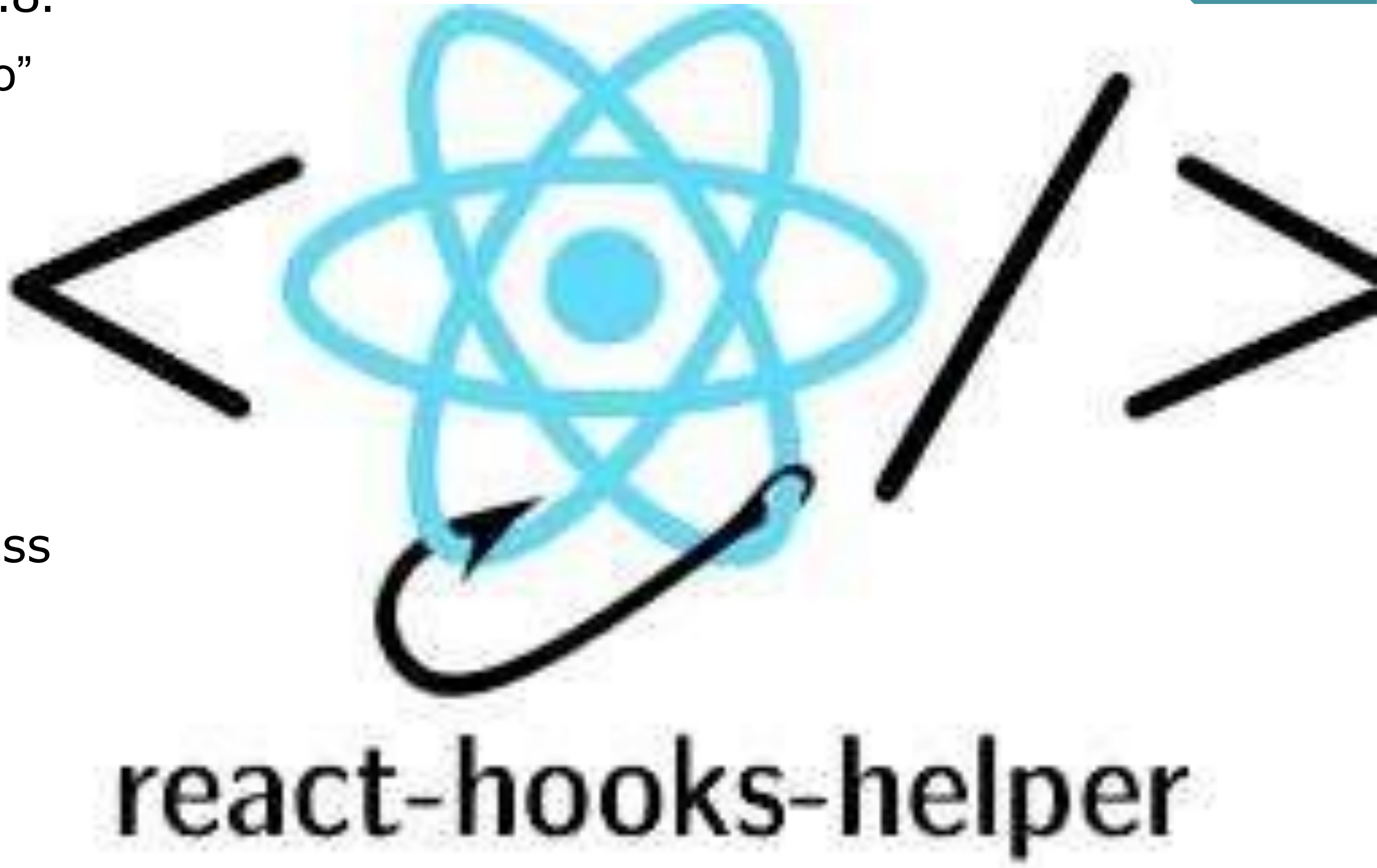
Introduction to useState Hook

Importing & Initialize useState

Example

Introduction

- ➢ Hooks are a new addition in React 16.8.
- ➢ Hooks are functions that let you “hook into” React state and lifecycle features from function components.
- ➢ **Why the need for Hooks?**
  - Use of *‘this’* keyword.
  - Reusable stateful logics.
  - Simplifying complex scenarios.
- ➢ Although Hooks generally replace class components.
- ➢ It does not work inside classes.
- ➢ Node version 6 or above & NPM 5.2 or above



Hook Rules

There are 3 rules for hooks:

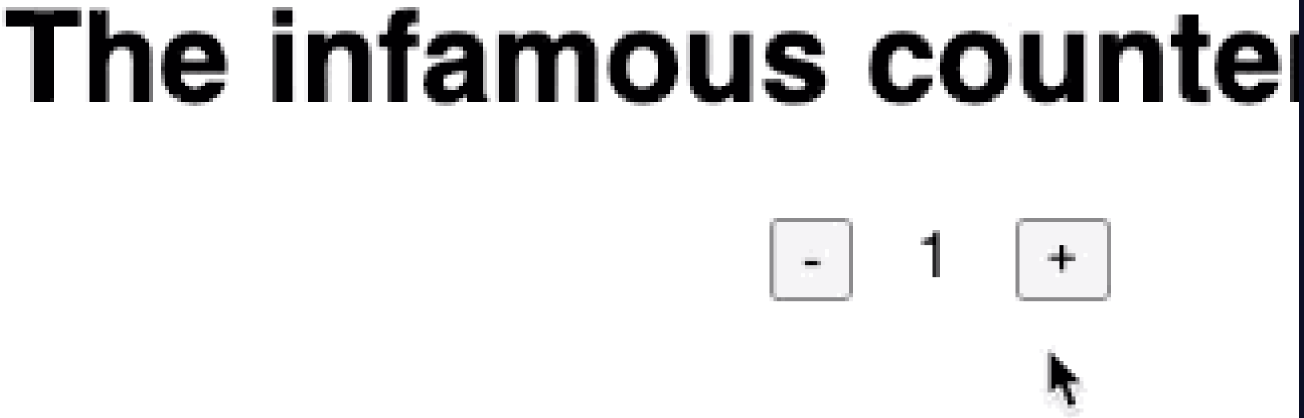
- ❖ Hooks can only be called inside React function components.
- ❖ Hooks can only be called at the top level

  of a component.
- ❖ Hooks cannot be conditional.

Note: Hooks will not work in React class components.

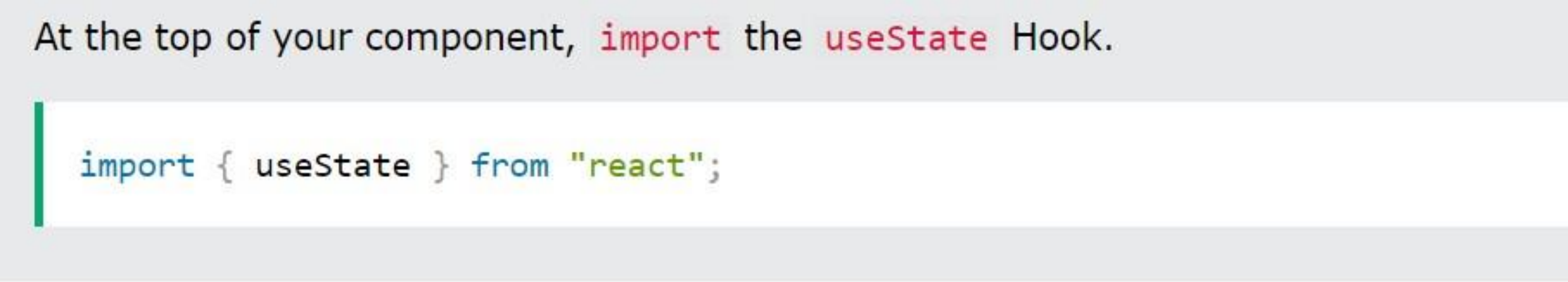
Introduction to useState Hook

- ➢ Do you have data in your component which changes over time? And by using normal variables you are not able to reload your component.
- ➢ This issue can be easily resolved by the useState hook.
- ➢ This hook reloads the component whenever there are any changes in the state thereby updating your user interface with the latest value
- ➢ useState is a react hook that allows you to create state variables



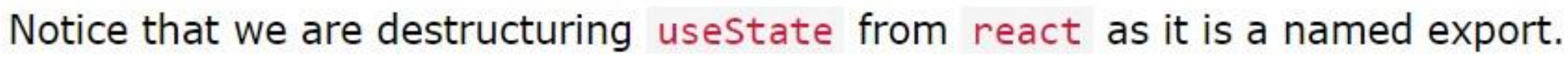
Importing useState

To use the `useState` Hook, we first need to `import` it into our component.



At the top of your component, `import` the `useState` Hook.

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

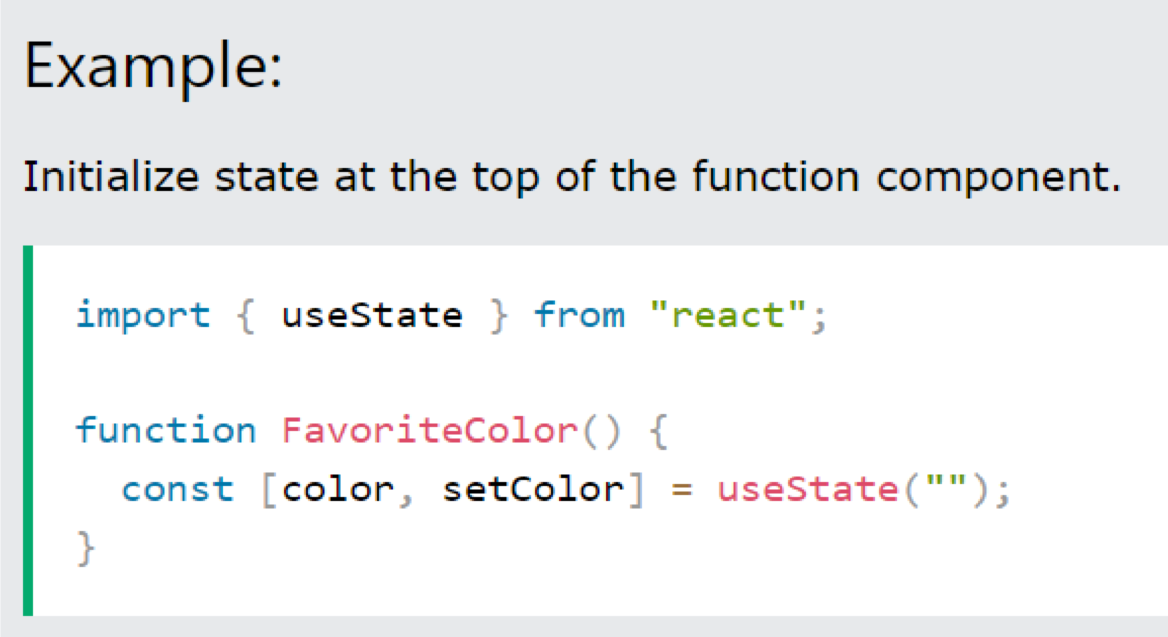


Notice that we are destructuring `useState` from `react` as it is a named export.

Initialize useState

We initialize our state by calling `useState` in our function component. `useState` accepts an initial state and returns two values:

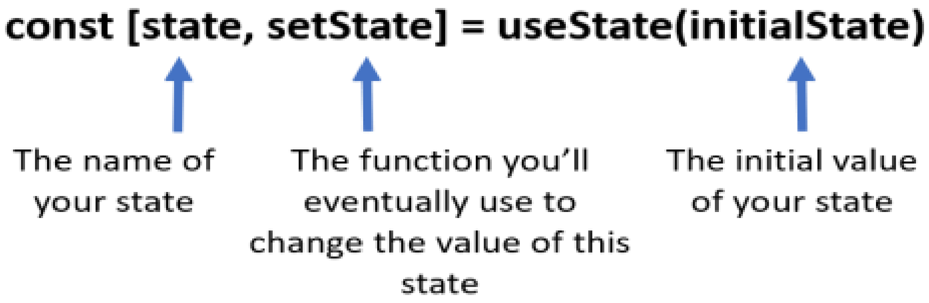
- The current state.
- A function that updates the state.



Example:

Initialize state at the top of the function component.

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

function FavoriteColor() {
  const [color, setColor] = useState("");
}
```



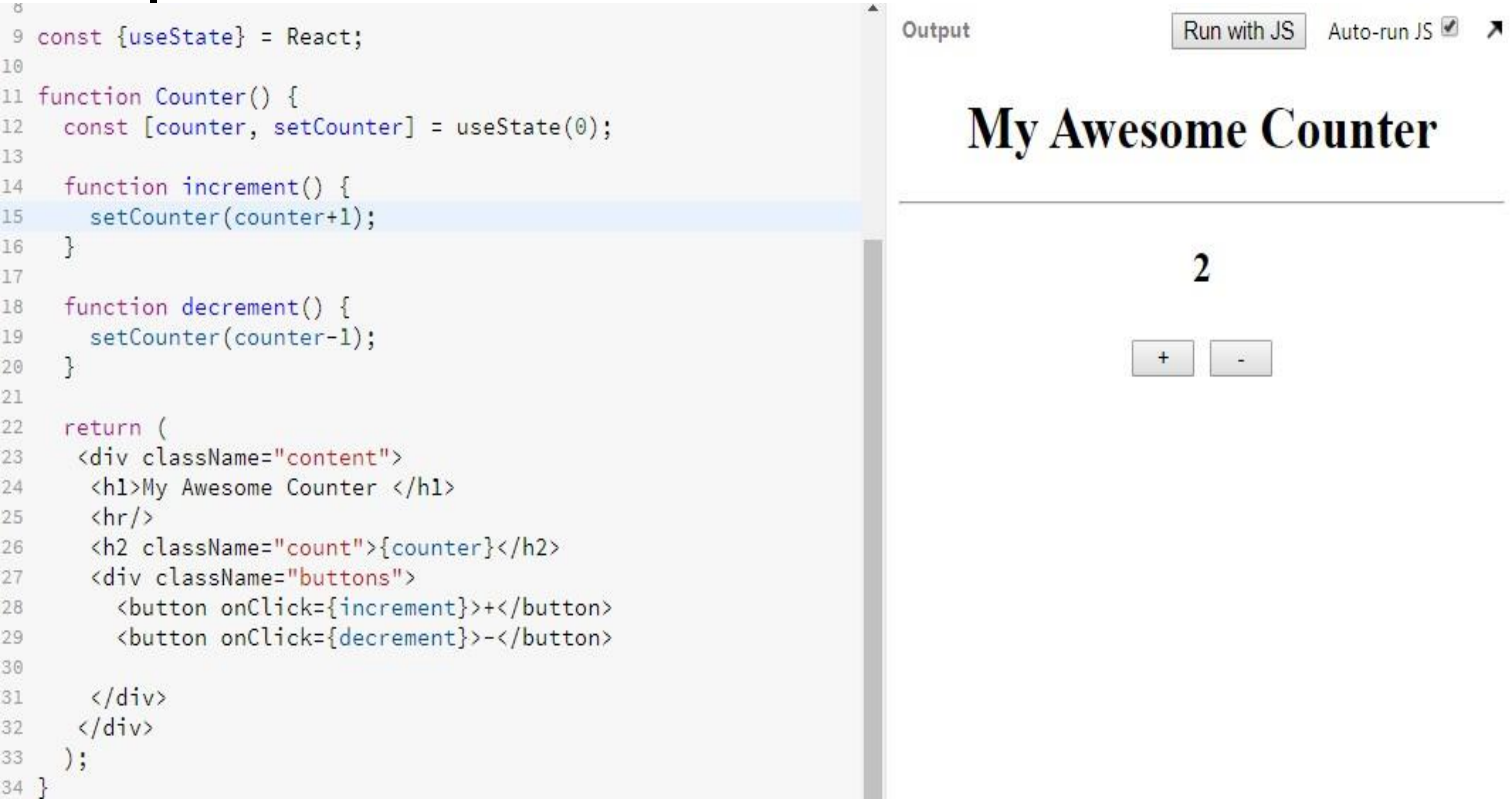
const [state, setState] = useState(initialState)

The name of your state

The function you'll eventually use to change the value of this state

The initial value of your state

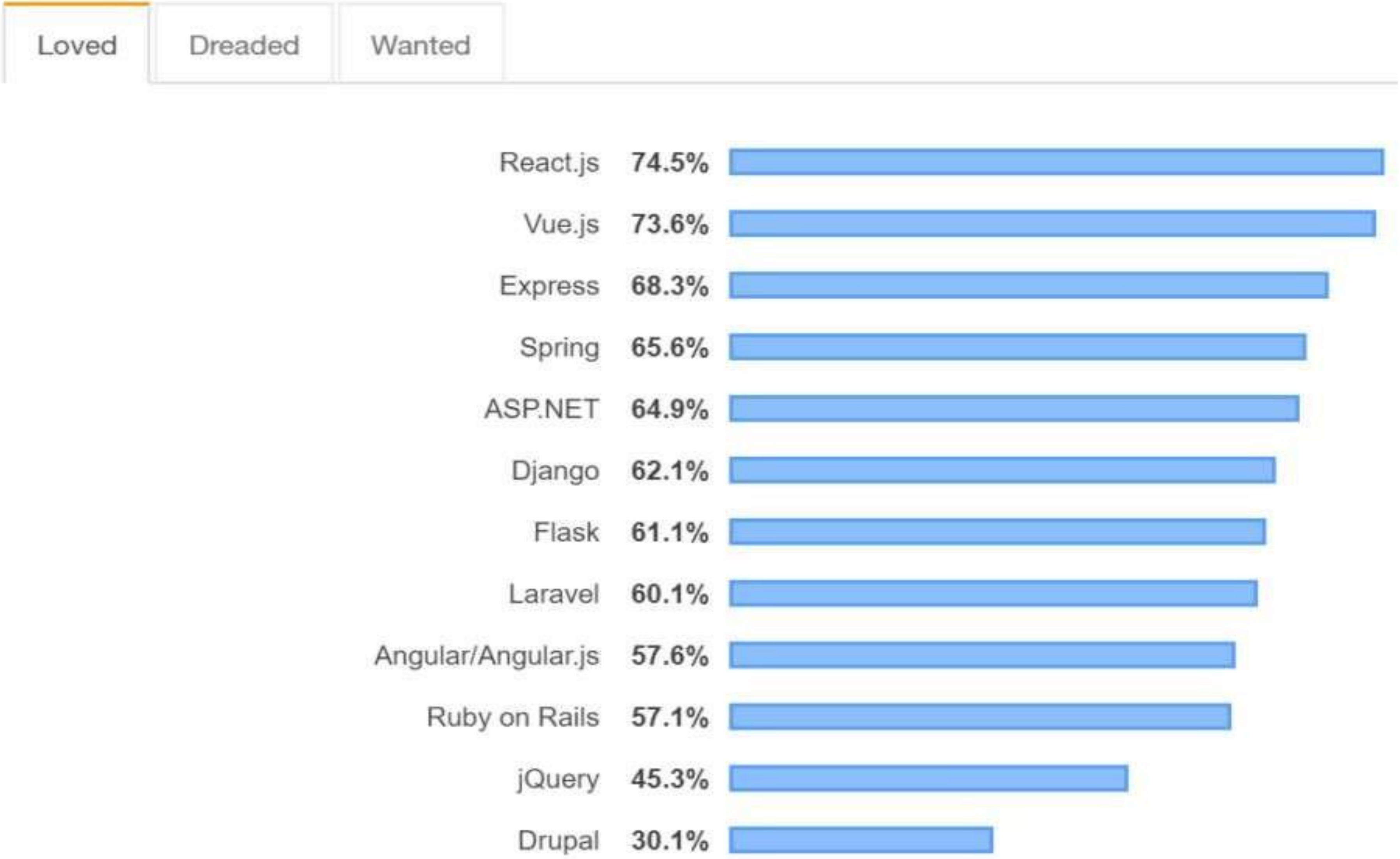
Example :



```
const {useState} = React;

function Counter() {
  const [counter, setCounter] = useState(0);

  function increment() {
    setCounter(counter+1);
  }

  function decrement() {
    setCounter(counter-1);
  }

  return (
   <div className="content">
    <h1>My Awesome Counter </h1>
    <hr/>
    <h2 className="count">{counter}</h2>
    <div className="buttons">
      <button onClick={increment}>+</button>
      <button onClick={decrement}>-</button>

    </div>
   </div>
  );
}
```

```
My Awesome Counter

2

+  -
```

Most Loved, Dreaded, and Wanted Web Frameworks



% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it

Agenda

What We Have Discussed Till Now?

Types Of Hooks

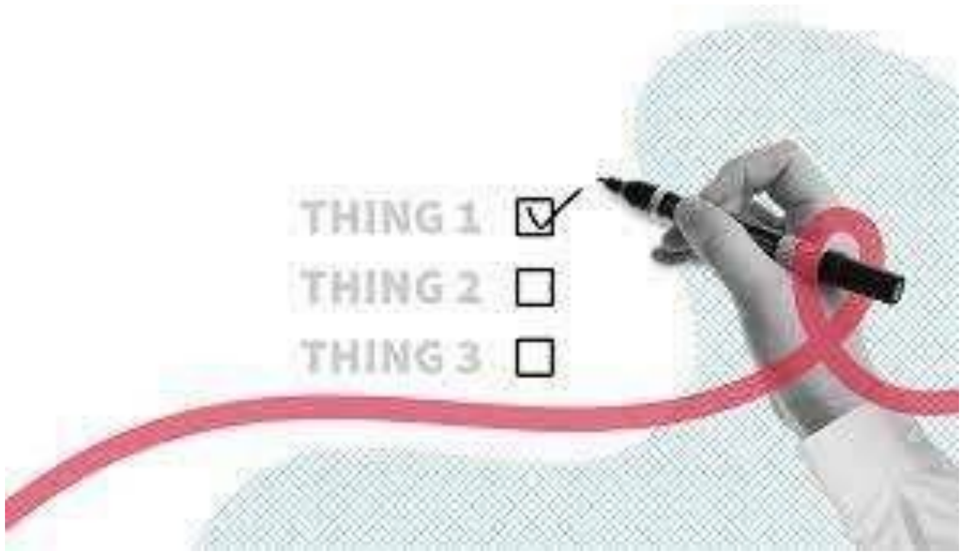
Introduction to useEffect 🦋

Effect Cleanup

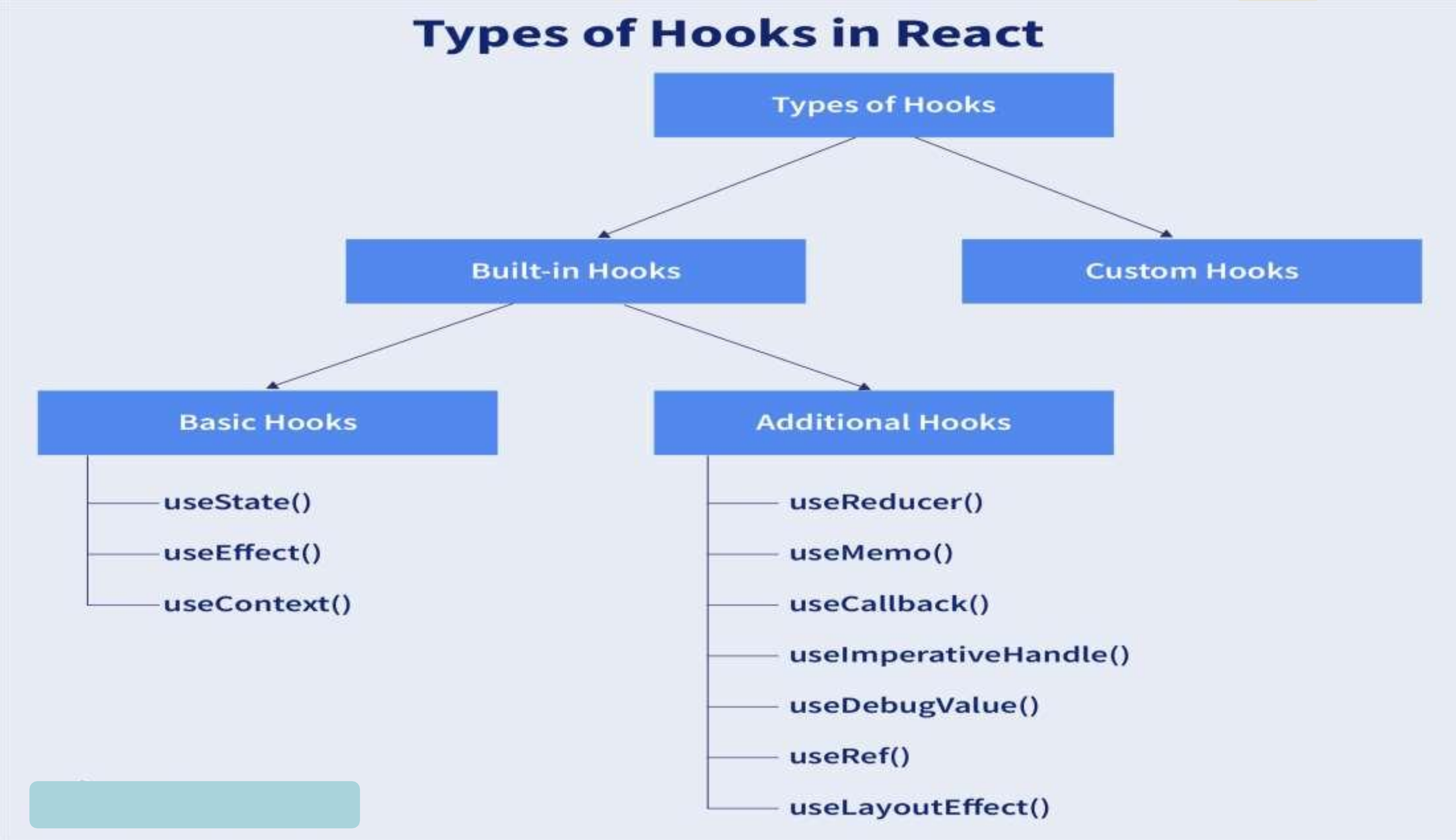
Conclusion

What We Have Discussed Till Now?

- ➢ Basic Introduction Of React Js
- ➢ Installation
- ➢ Some Prerequisite
- ➢ Introduction to Hooks 🦄 and some Rules
- ➢ useState hook done

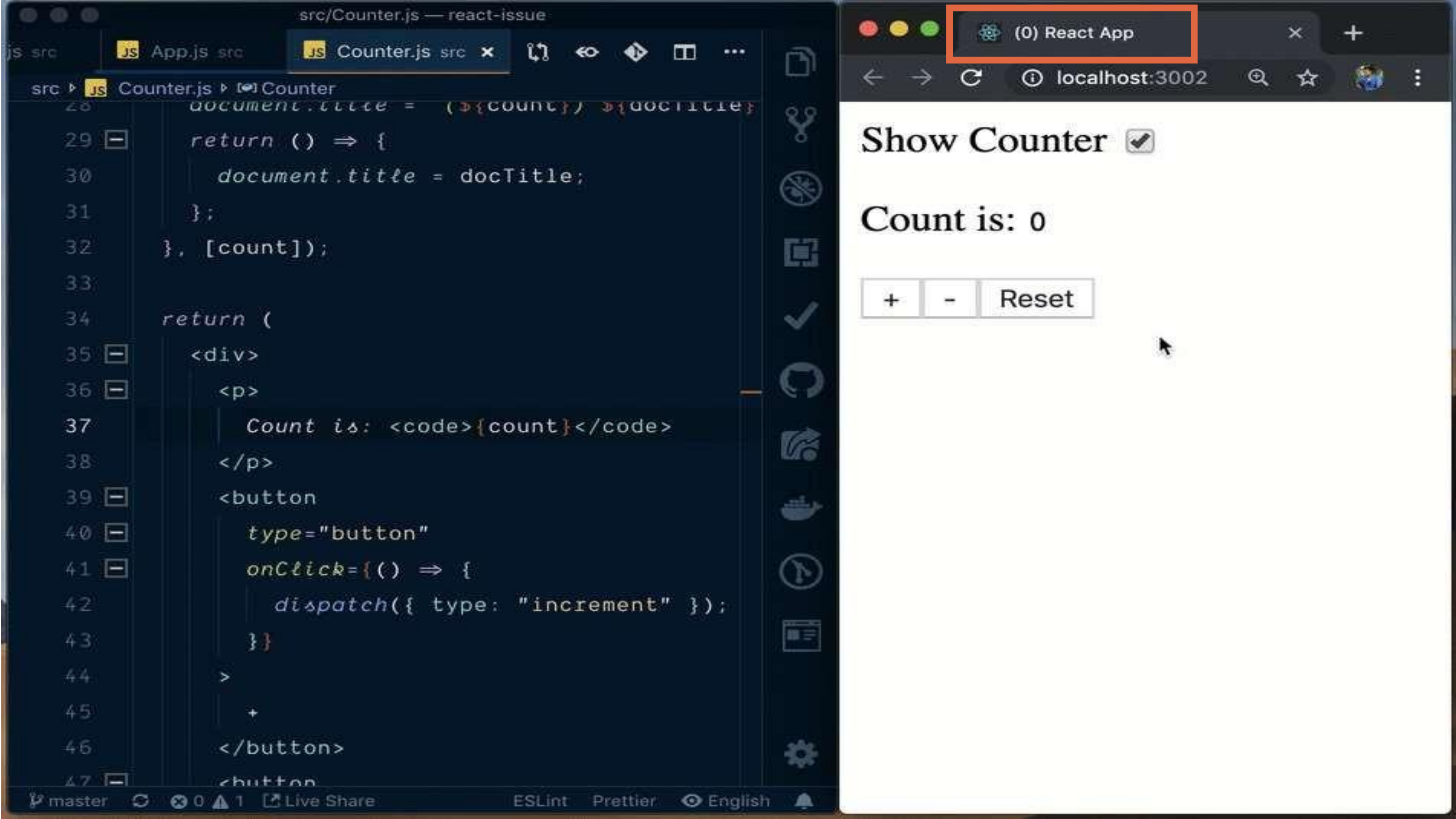


Types Of Hooks :



Introduction to useEffect

- The useEffect helps to perform side-effects(Outside current scope) in functional components.
- Automatically called when page LOADS.
- Eg : WHATSAPP chat count(chats(5)), Error-tracking.
- Used Outside the component to render data(API's).
- useEffect accepts two arguments. The second argument is optional.
- **Syntax :** `useEffect(<function>, <dependency>)`



```
    document.title = `(${count}) ${docTitle}`
    return () => {
      document.title = docTitle;
    };
  }, [count]);

  return (
    <div>
      <p>
        Count is: <code>{count}</code>
      </p>
      <button
        type="button"
        onClick={() => {
          dispatch({ type: "increment" });
        }}
      >
        +
      </button>
      <button
```

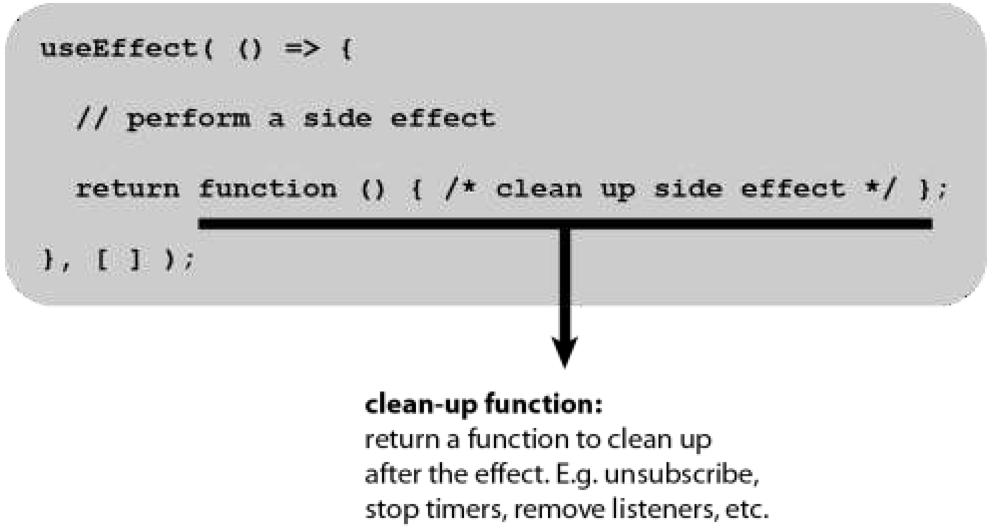
Show Counter ☑

Count is: 0

+ - Reset

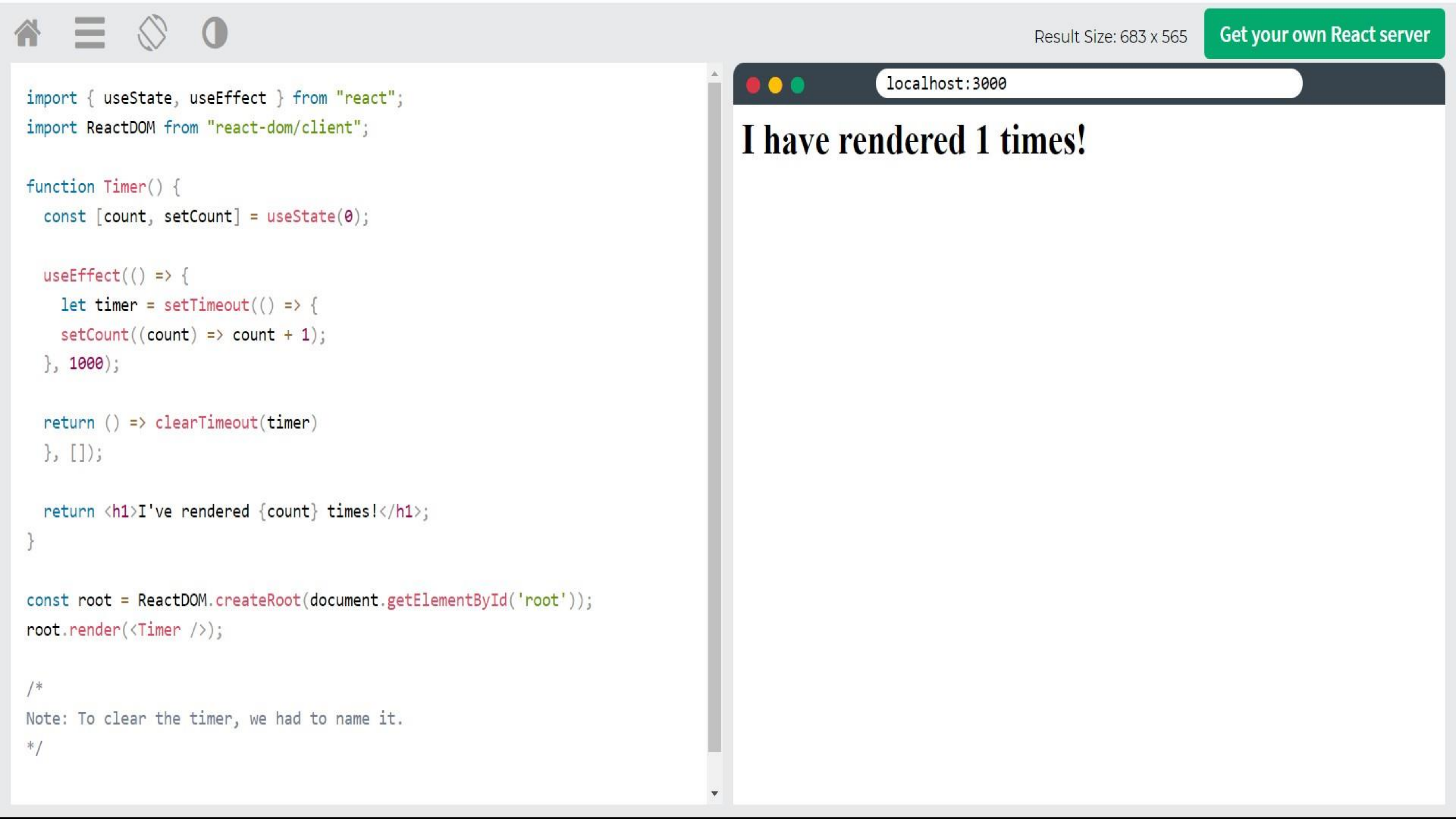
Effect Cleanup

- Some effects require cleanup to reduce memory leaks.
- Timeouts, subscriptions, event listeners, and other effects that are no longer needed should be disposed
- We do this by including a return function at the end of the `useEffect` Hook.



```
useEffect( () => {

  // perform a side effect

  return function () { /* clean up side effect */ };

}, [ ] );
```

clean-up function:
return a function to clean up
after the effect. E.g. unsubscribe,
stop timers, remove listeners, etc.

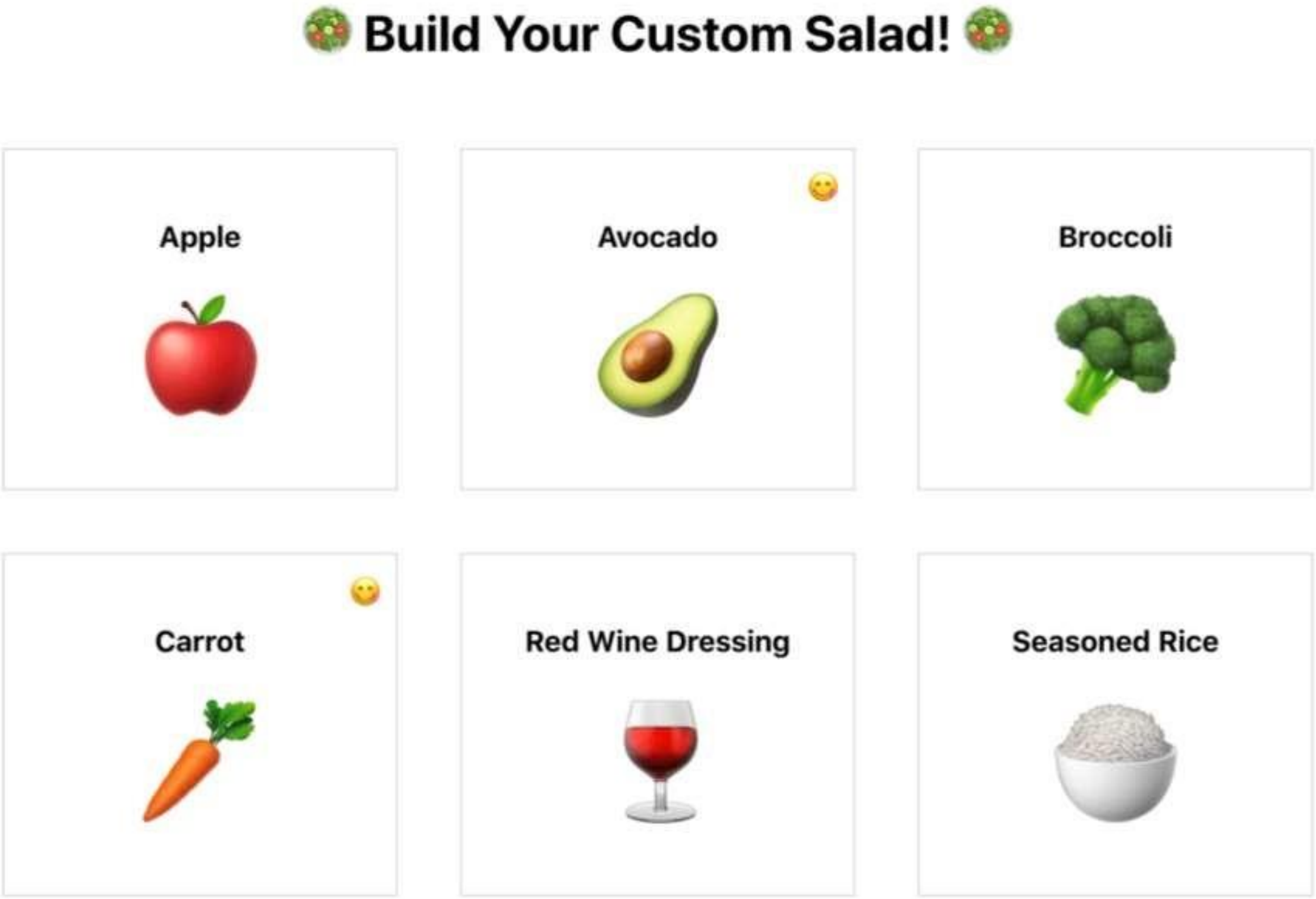


```
import { useState, useEffect } from "react";
import ReactDOM from "react-dom/client";

function Timer() {
  const [count, setCount] = useState(0);

  useEffect(() => {
    let timer = setTimeout(() => {
    setCount((count) => count + 1);
  }, 1000);

  return () => clearTimeout(timer)
  }, []);

  return <h1>I've rendered {count} times!</h1>;
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Timer />);

/*
Note: To clear the timer, we had to name it.
*/
```

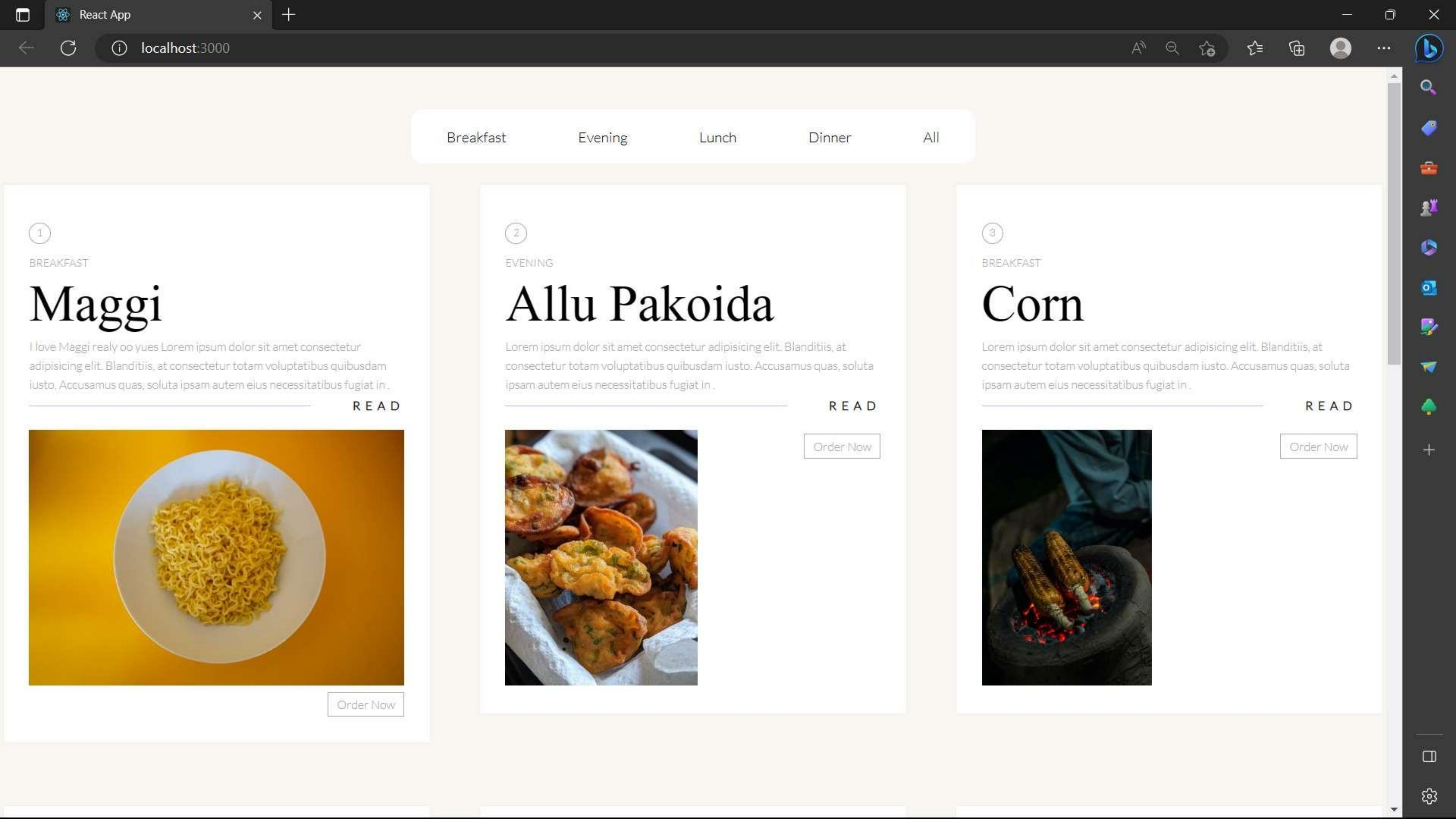
```
I have rendered 1 times!
```

Reusable Components In React

Re-Usable Components



🥗 Build Your Custom Salad! 🥗



1

BREAKFAST

Maggi

I love Maggi realy oo yues Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .

2

EVENING

Allu Pakoida

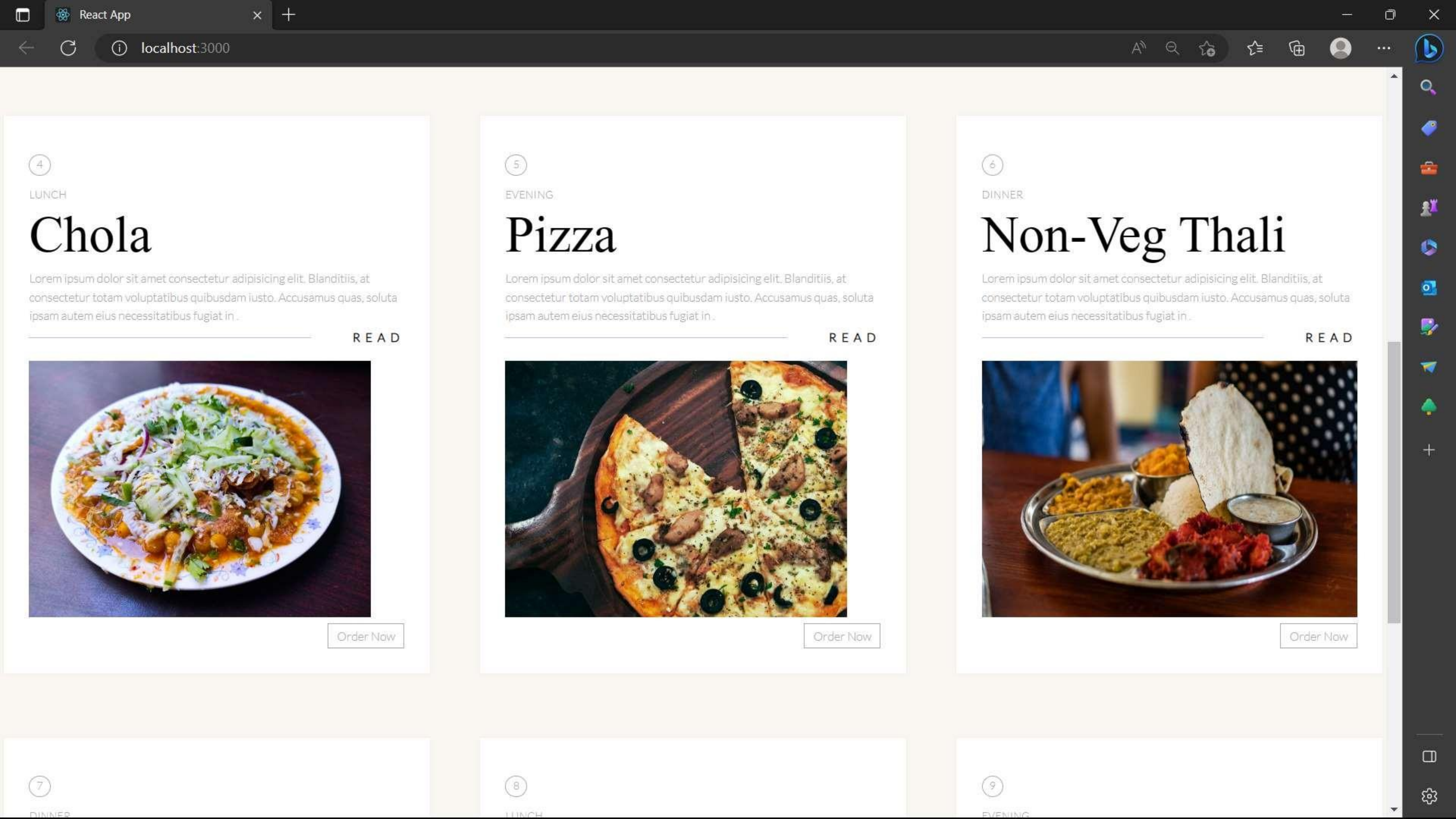
Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .

3

BREAKFAST

Corn

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .



4

LUNCH

Chola

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .

READ

Order Now

5

EVENING

Pizza

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .

READ

Order Now

6

DINNER

Non-Veg Thali

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .

READ

Order Now

7

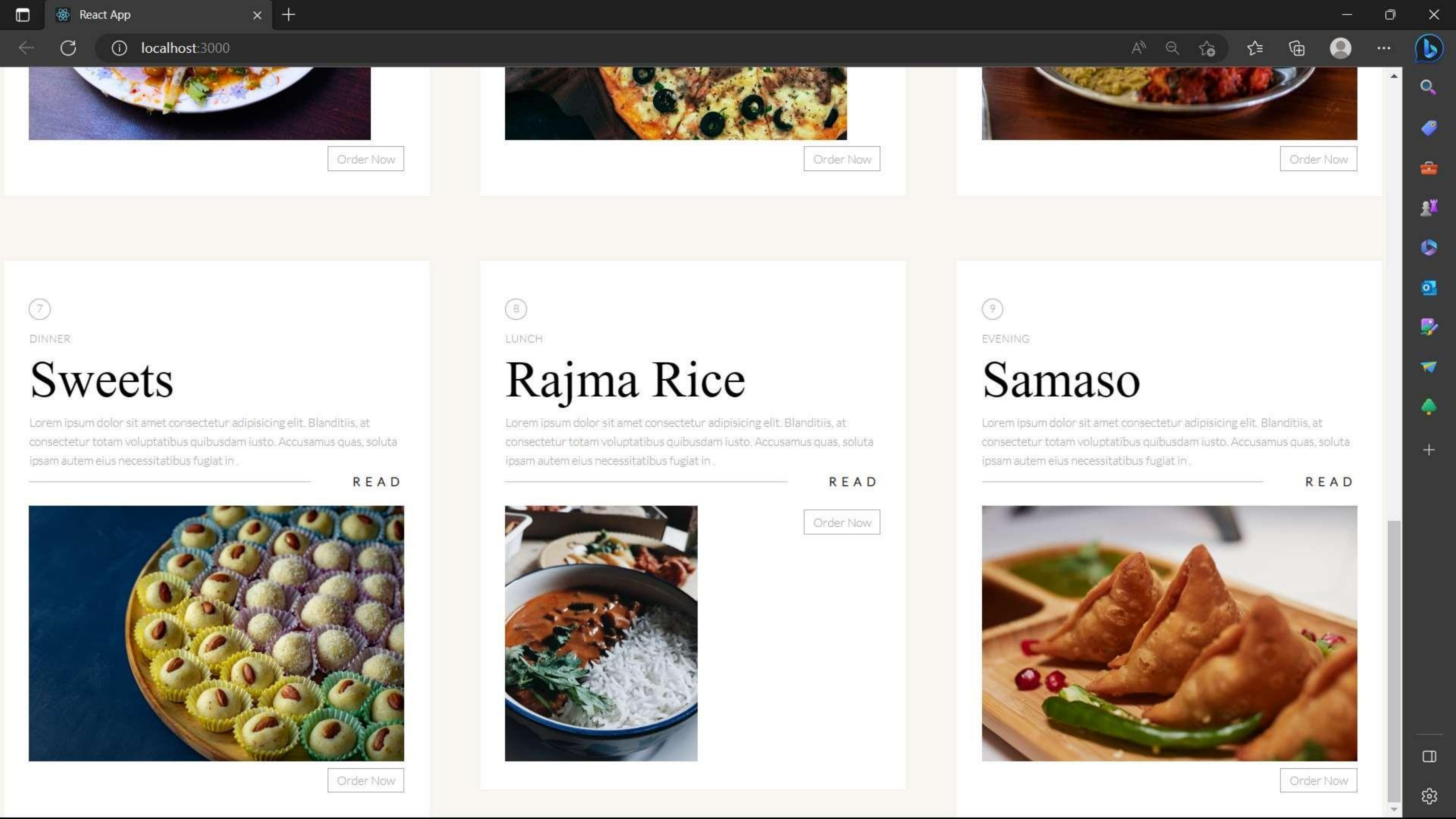
DINNER

8

LUNCH

9

EVENING



7

DINNER

Sweets

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .

READ

8

LUNCH

Rajma Rice

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .

READ

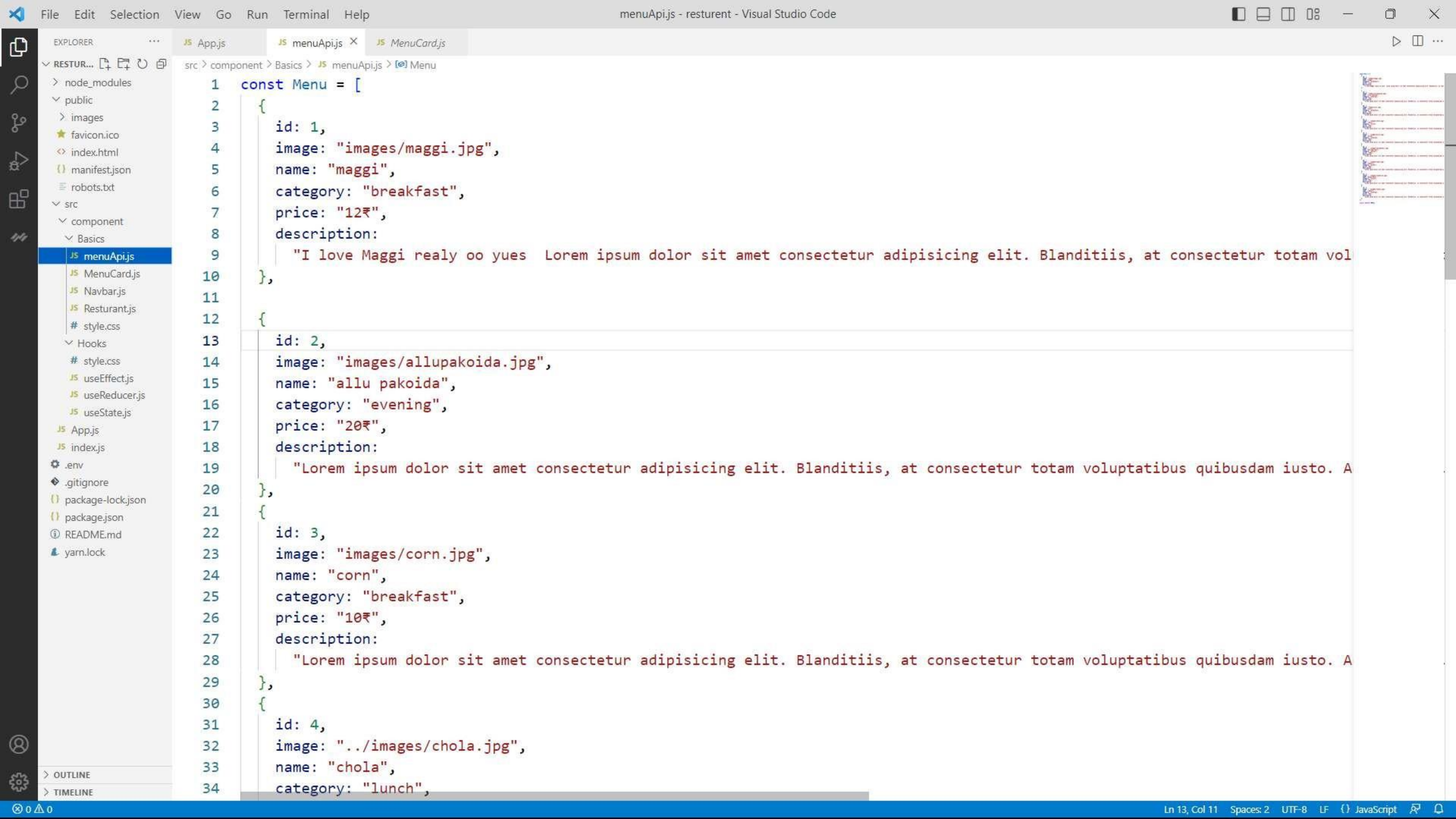
9

EVENING

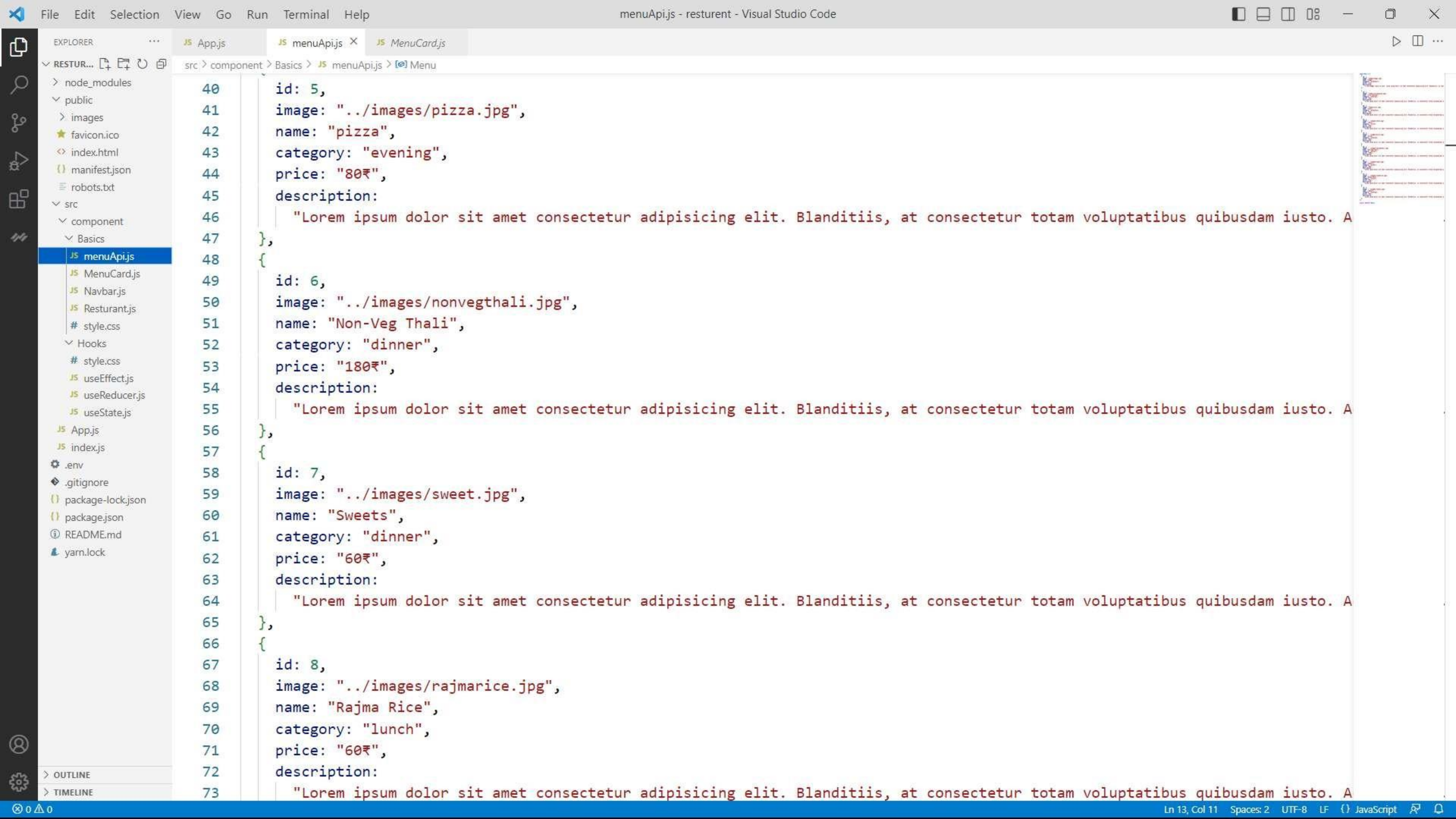
Samaso

Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. Accusamus quas, soluta ipsam autem eius necessitatibus fugiat in .

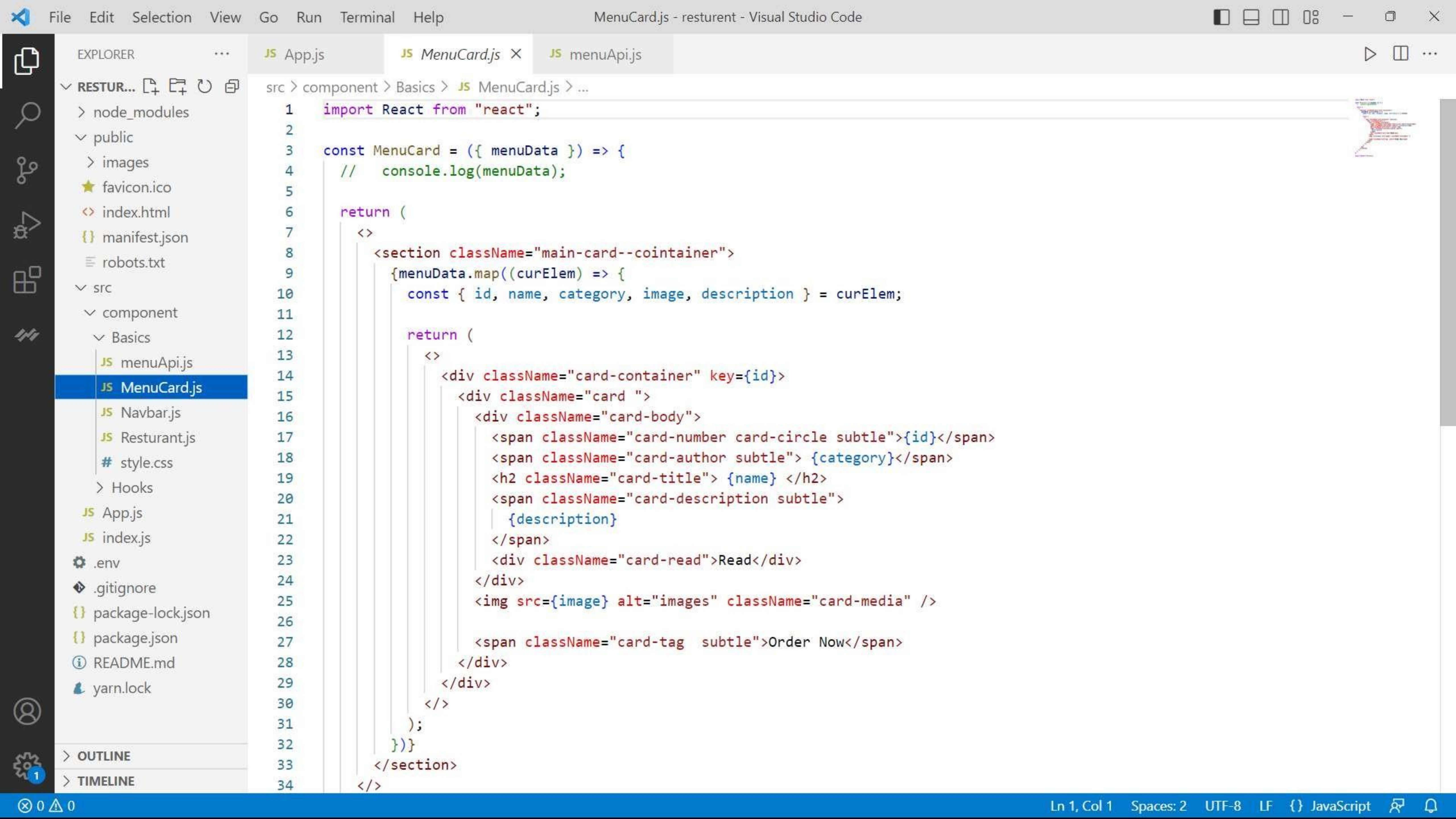
READ



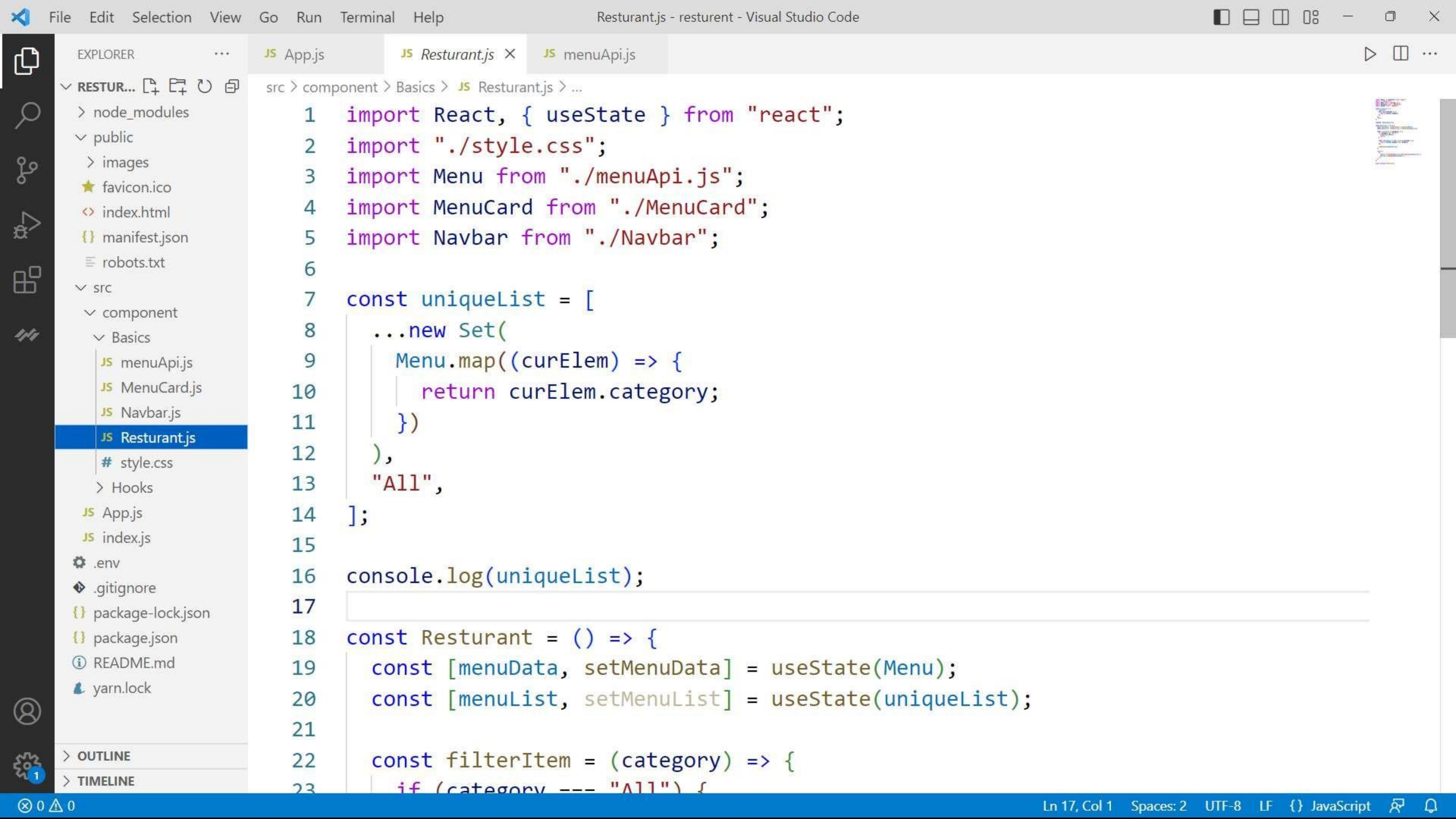
```
const Menu = [
  {
    id: 1,
    image: "images/maggi.jpg",
    name: "maggi",
    category: "breakfast",
    price: "12₹",
    description:
      "I love Maggi realy oo yues  Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam vol
  },

  {
    id: 2,
    image: "images/allupakoida.jpg",
    name: "allu pakoida",
    category: "evening",
    price: "20₹",
    description:
      "Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. A
  },
  {
    id: 3,
    image: "images/corn.jpg",
    name: "corn",
    category: "breakfast",
    price: "10₹",
    description:
      "Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. A
  },
  {
    id: 4,
    image: "../images/chola.jpg",
    name: "chola",
    category: "lunch",
```



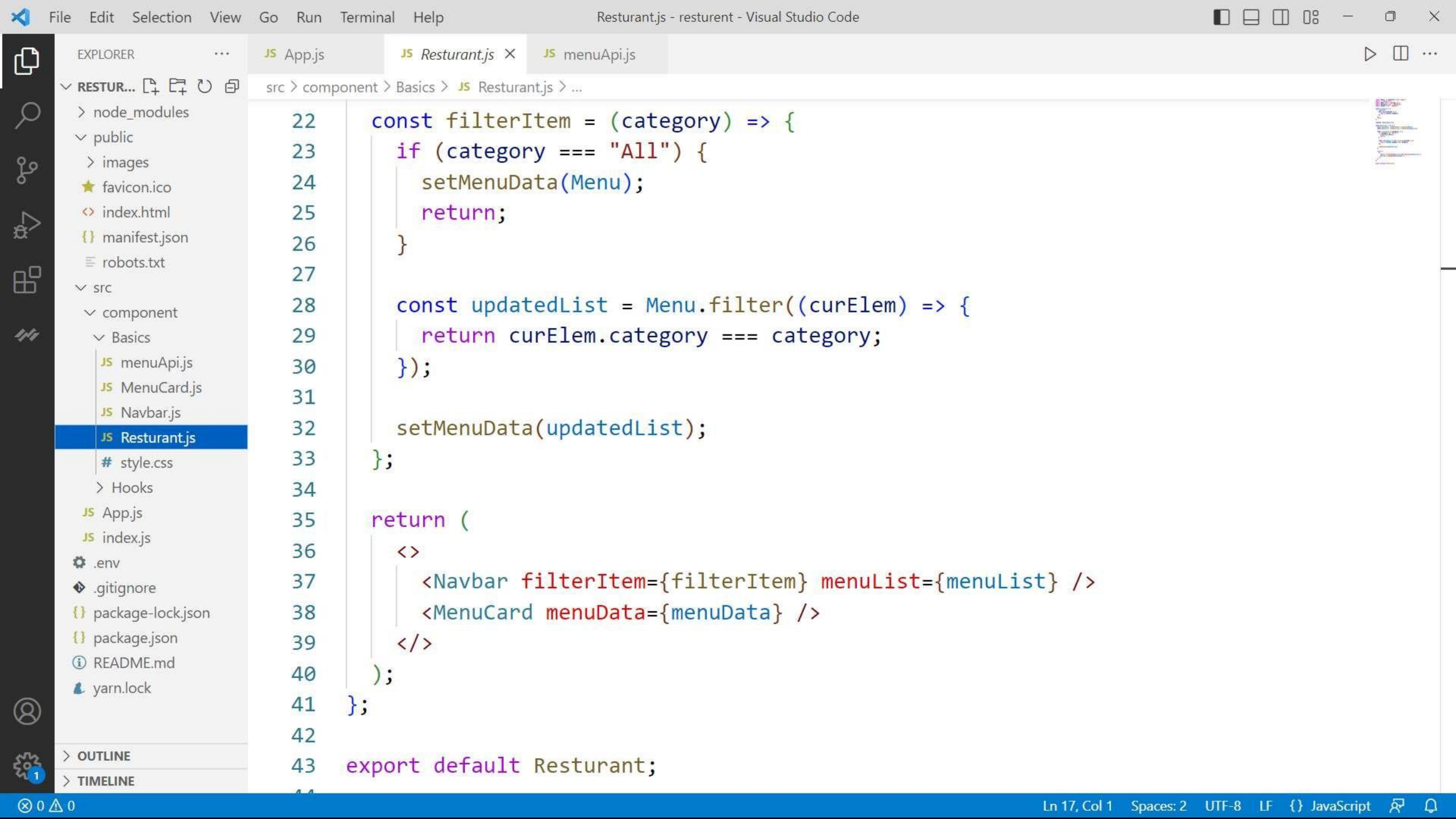
```
    id: 5,
    image: "../images/pizza.jpg",
    name: "pizza",
    category: "evening",
    price: "80₹",
    description:
      "Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. A
  },
  {
    id: 6,
    image: "../images/nonvegthali.jpg",
    name: "Non-Veg Thali",
    category: "dinner",
    price: "180₹",
    description:
      "Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. A
  },
  {
    id: 7,
    image: "../images/sweet.jpg",
    name: "Sweets",
    category: "dinner",
    price: "60₹",
    description:
      "Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. A
  },
  {
    id: 8,
    image: "../images/rajmarice.jpg",
    name: "Rajma Rice",
    category: "lunch",
    price: "60₹",
    description:
      "Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis, at consectetur totam voluptatibus quibusdam iusto. A
```



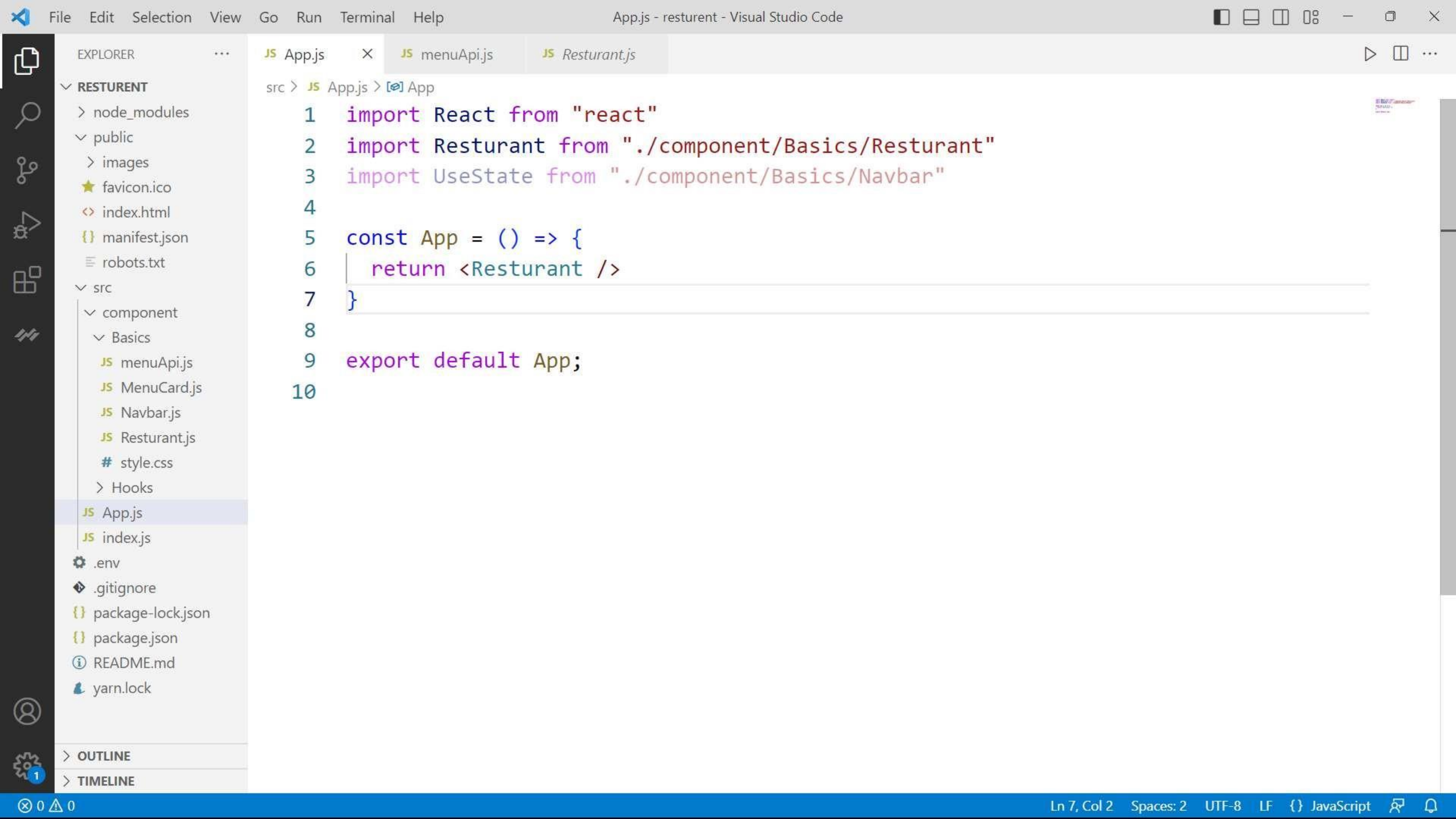
```
import React from "react";

const MenuCard = ({ menuData }) => {
  //   console.log(menuData);

  return (
    <>
      <section className="main-card--cointainer">
        {menuData.map((curElem) => {
          const { id, name, category, image, description } = curElem;

          return (
            <>
              <div className="card-container" key={id}>
                <div className="card ">
                  <div className="card-body">
                    <span className="card-number card-circle subtle">{id}</span>
                    <span className="card-author subtle"> {category}</span>
                    <h2 className="card-title"> {name} </h2>
                    <span className="card-description subtle">
                      {description}
                    </span>
                    <div className="card-read">Read</div>
                  </div>
                  <img src={image} alt="images" className="card-media" />

                  <span className="card-tag  subtle">Order Now</span>
                </div>
              </div>
            </>
          );
        })}
      </section>
    </>
```



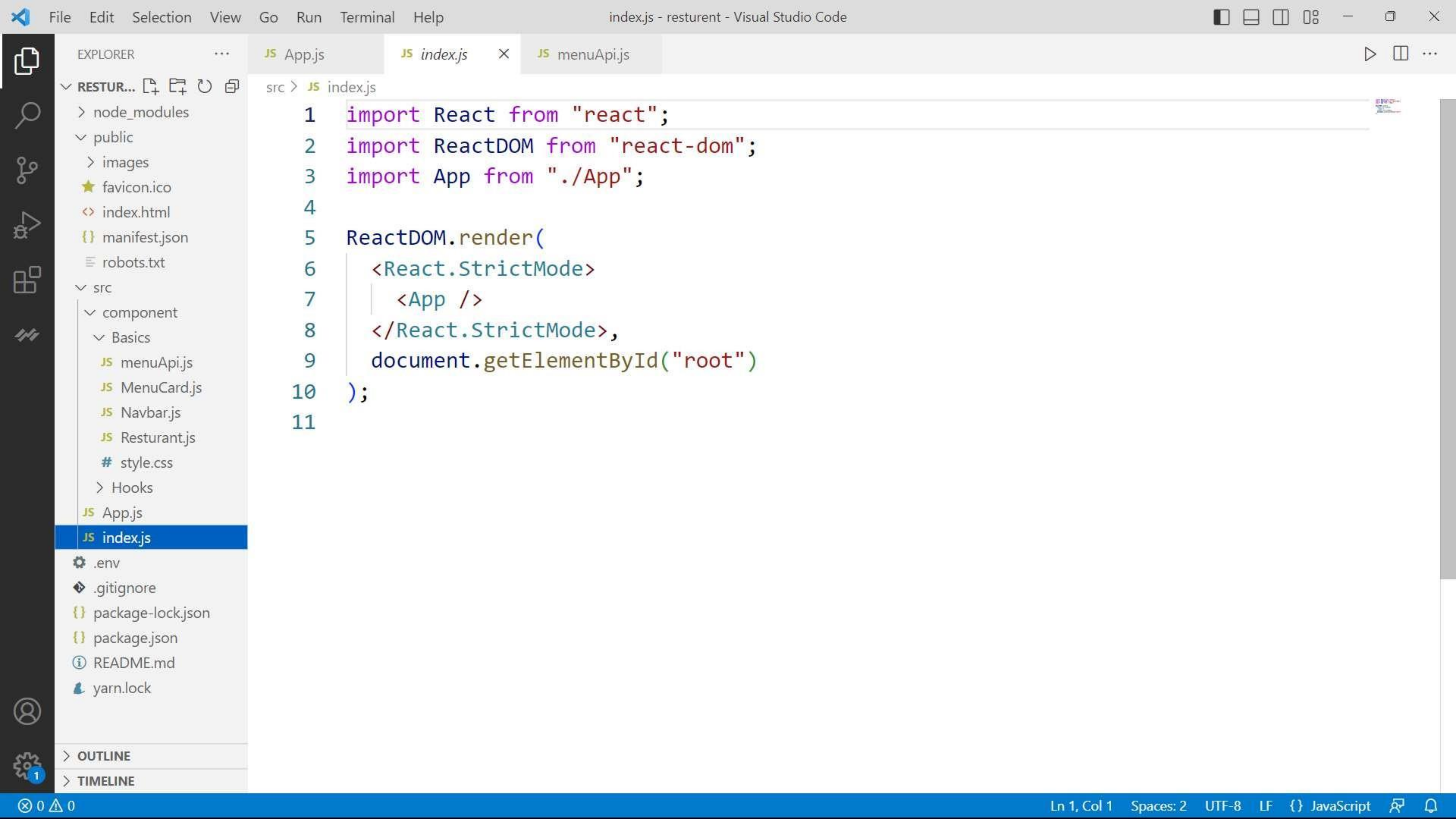
```
import React, { useState } from "react";
import "./style.css";
import Menu from "./menuApi.js";
import MenuCard from "./MenuCard";
import Navbar from "./Navbar";

const uniqueList = [
  ...new Set(
    Menu.map((curElem) => {
      return curElem.category;
    })
  ),
  "All",
];

console.log(uniqueList);

const Resturant = () => {
  const [menuData, setMenuData] = useState(Menu);
  const [menuList, setMenuList] = useState(uniqueList);

  const filterItem = (category) => {
    if (category === "All") {
```



```
  const filterItem = (category) => {
    if (category === "All") {
      setMenuData(Menu);
      return;
    }

    const updatedList = Menu.filter((curElem) => {
      return curElem.category === category;
    });

    setMenuData(updatedList);
  };

  return (
    <>
      <Navbar filterItem={filterItem} menuList={menuList} />
      <MenuCard menuData={menuData} />
    </>
  );
};

export default Resturant;
```



```
import React from "react"
import Resturant from "./component/Basics/Resturant"
import UseState from "./component/Basics/Navbar"

const App = () => {
  return <Resturant />
}

export default App;
```



```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById("root")
);

```

THANK YOU

ANY QUESTION?