VIRTUAL CODE

The ULTIMATE REACT COURSE

Level – 1 : Introduction to React

Level – 2 : Create Our First React App
Practice Project 1

Level – 3 : Props In React
Practice Project 2

Level – 4 : Hooks In React
Practice Project 3

Level – 5 : React Router
Practice Project 4

Level – 6 : Context API

VIRTUAL CODE

LEVEL-1

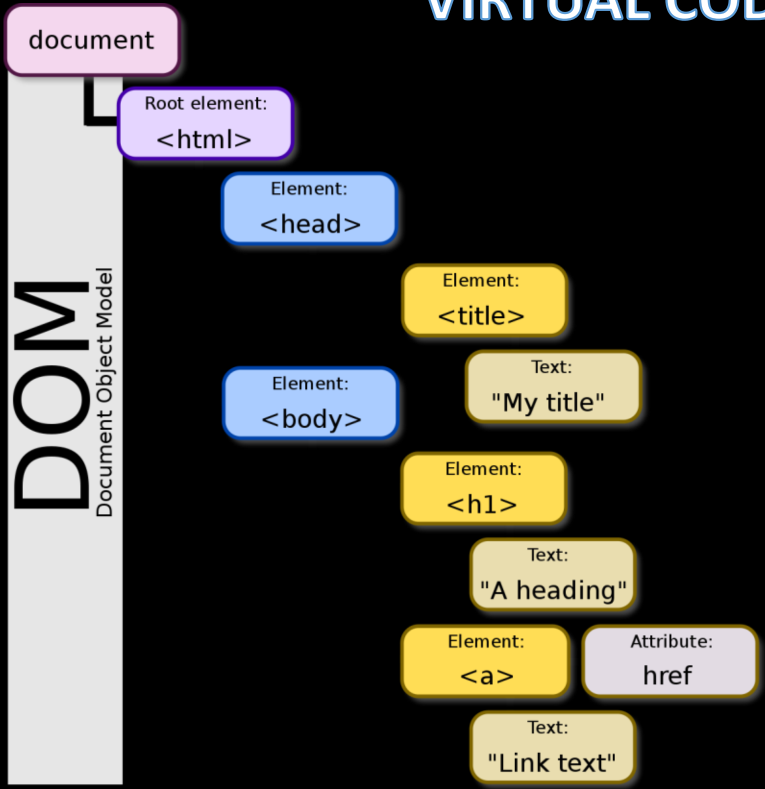
Introduction To React

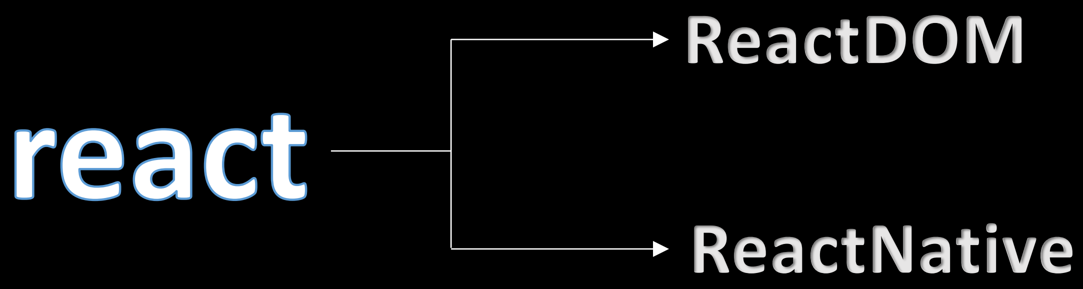
- REACT (also known as React.js or ReactJs) is a free and open source FRONT-END JAVASCRIPT LIBRARY for building user interfaces based on components by Facebook inc.

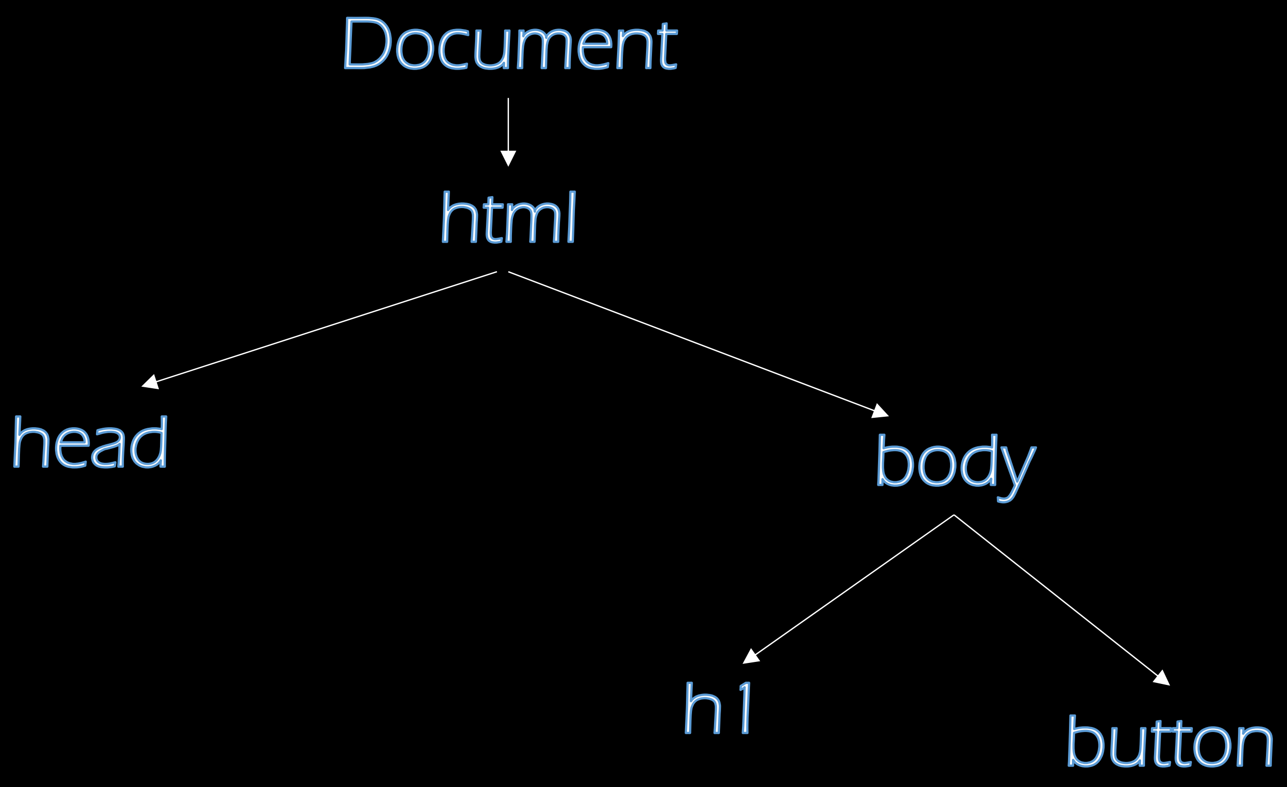
- REACT can be used to develop single-page application

- A key Advantage of REACT is that it only rerenders those parts of the page that have changed, avoiding unnecessary rerendering of unchanged DOM elements.

- You can use CDN for add React in your JavaScript file






Document
html
head
body
h1
button

LEVEL-2

Create our First React App

Create our first React app

There are some popular utility for creating React App

1. Create-react-app
2. Vite (recommended)

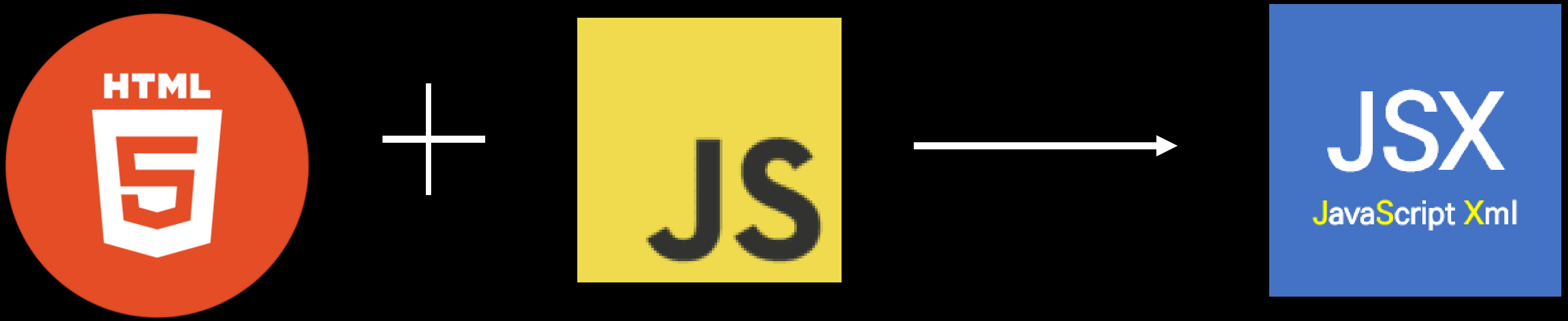
Some frequently used words-

1. npm - npm is the package manager for the javascript runtime environment Node.js

1. npx- node package Executor

JSX –

JSX (Javascript XML , formally Javascript Syntax Extension)



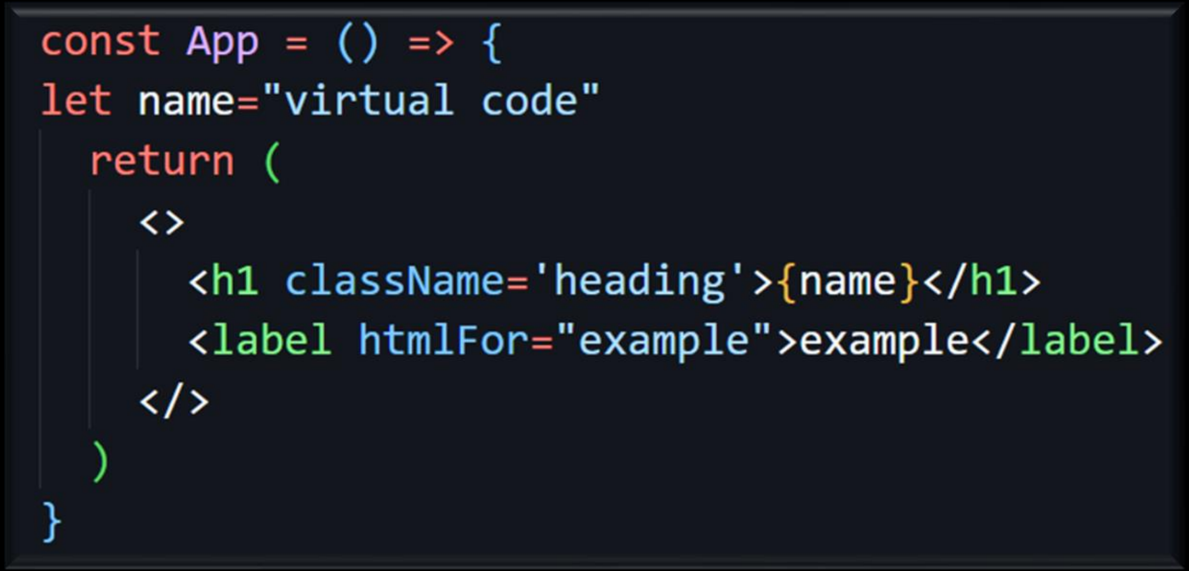
Create our first React app

VIRTUAL CODE

Some Important KeyPoint For JSX-

HTML	JSX
class	className
for	htmlFor

You can write javascript in JSX using curly brackets

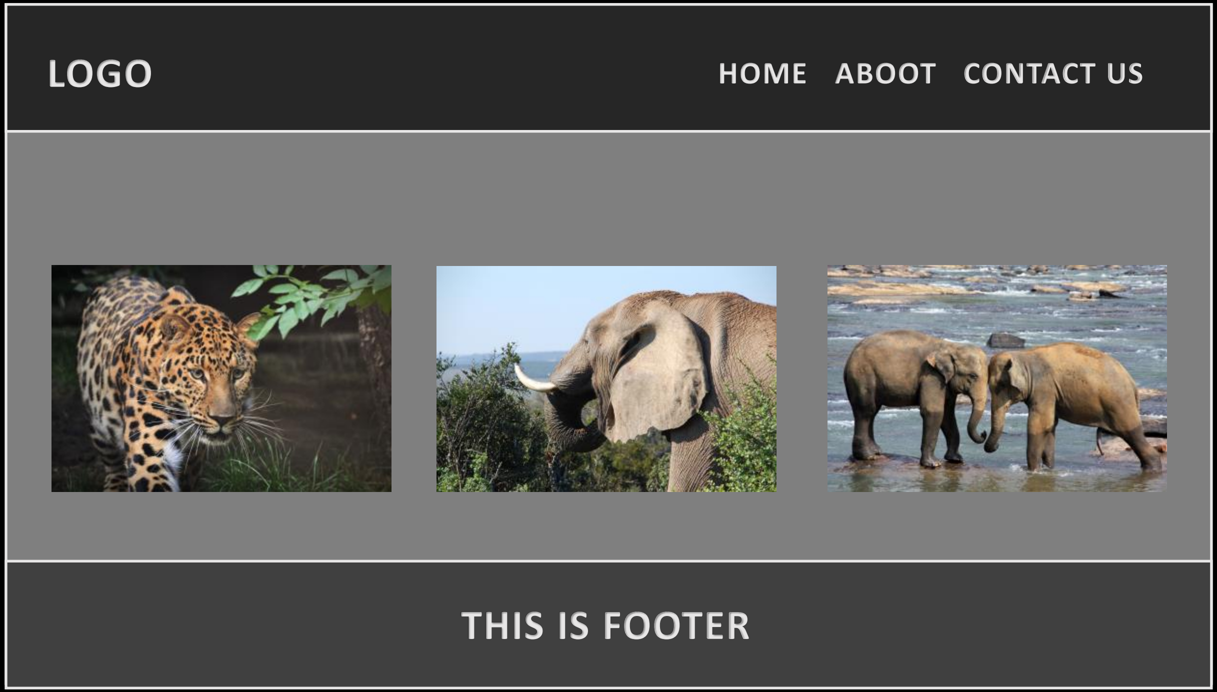


```
const App = () => {
let name="virtual code"
  return (
    <>
      <h1 className='heading'>{name}</h1>
      <label htmlFor="example">example</label>
    </>
  )
}
```

Practice Project 1

VIRTUAL CODE

Create this layout using React JS-



LEVEL-3

Props in React

Props In react

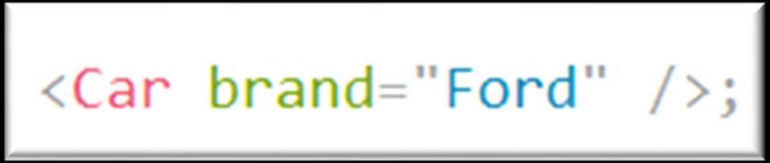
- Props stands for Properties
- Props are arguments passed into React Components
- React Props are like function arguments in Javascript and attributes in HTML
- To send props into a component use the same syntax as HTML attributes

Props In react

VIRTUAL CODE

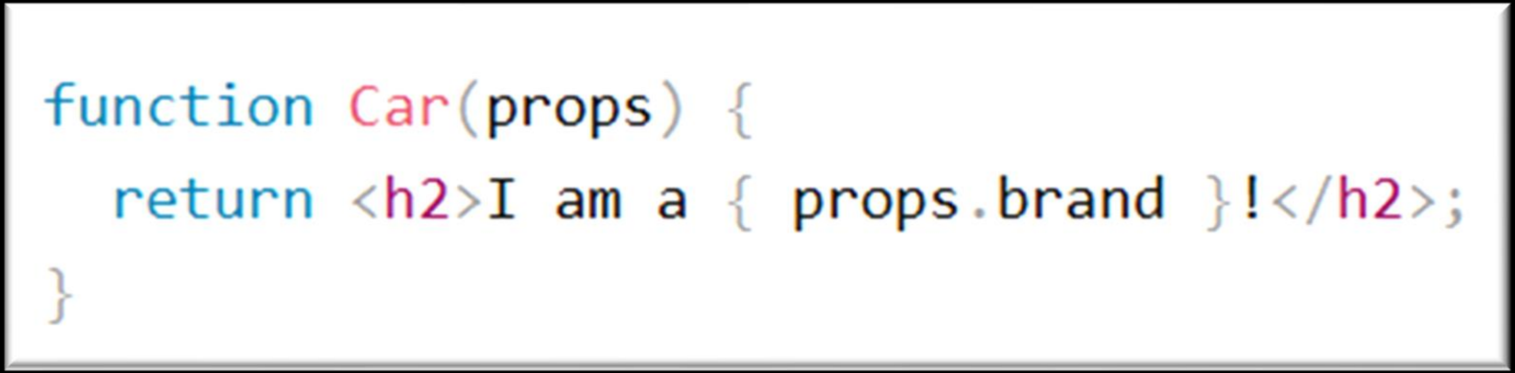
Example-

add a “brand” attribute in car element



```
<Car brand="Ford" />;
```

The component receives the argument as a `props` object

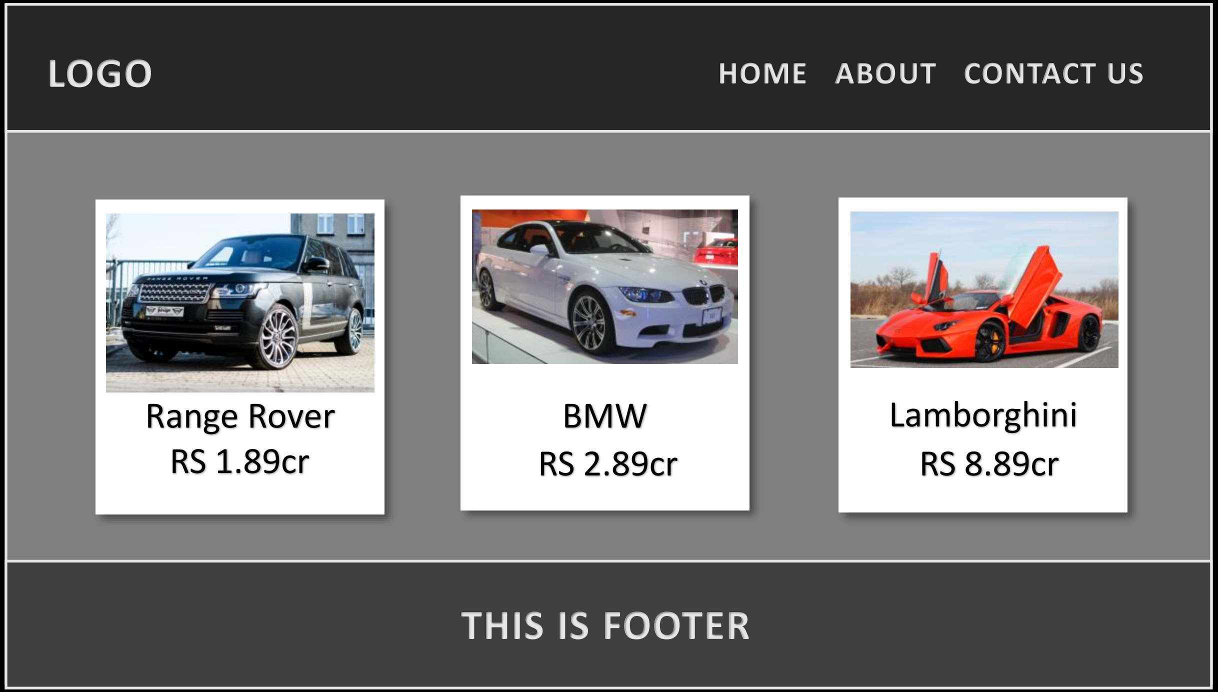


```
function Car(props) {
  return <h2>I am a { props.brand }!</h2>;
}
```

Practice Project 2

VIRTUAL CODE

Create this layout using React JS-



LEVEL-4

Hooks in React

Hooks In react

- Hooks allows function components to have access to state and other React features . Because of this , class components are generally no longer needed.

- Now what is state in React?

- In React , State is a javascript object that holds data and information about a component.

- There are so many Types of Hooks in React-

1. useState( ) Hook
2. useEffect( ) Hook
3. useRef( ) Hook
4. useMemo( ) Hook
5. useCallback( ) Hook
6. useContext( ) Hook
7. useReducer( ) Hook
8. useLayoutEffect( ) Hook
9. customHook( ) Hook

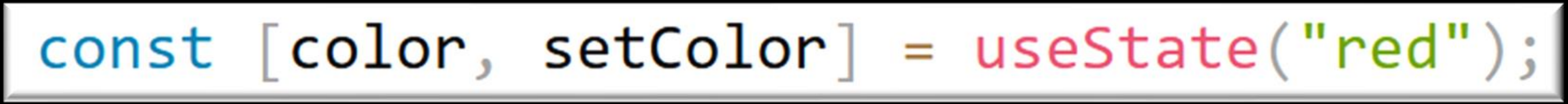
1. useState( ) Hook-

The react ,useState Hook allows us to track state in a function component.

State generally refers to data or properties that need to be tracking in an application.

We initialize our state by calling useState in our function component

useState accepts an initial state and returns two values .
1. Current state
2. A function that updates the state



```
const [color, setColor] = useState("red");
```

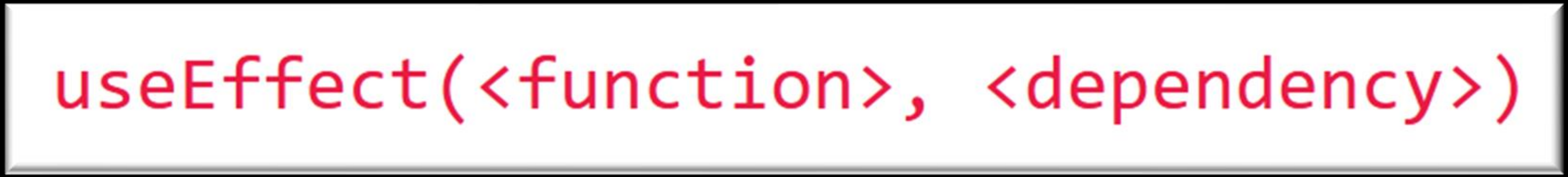
2. useEffect( ) Hook-

The useEffect Hook allows you to perform side effects in your components.

Some examples of side effects are –

1. Fetching data
2. Directly updating DOM Element
3. Timers

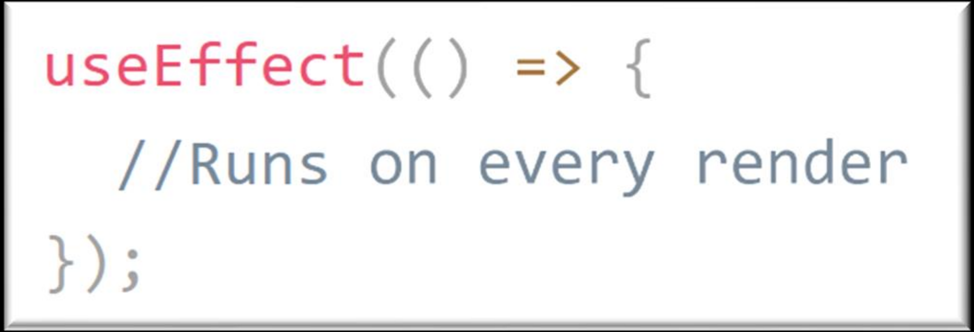
useEffect accepts two arguments . The second argument is optional



```
useEffect(<function>, <dependency>)
```

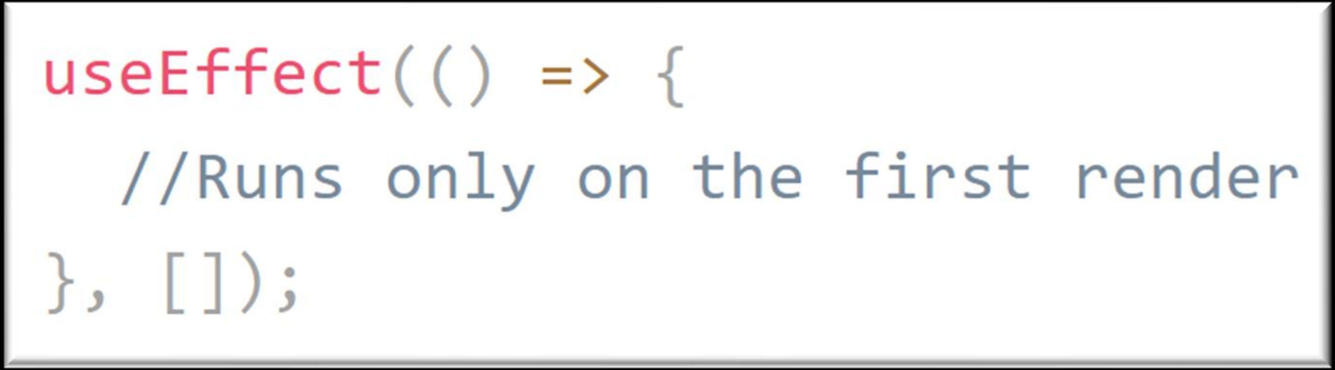
2. useEffect( ) Hook-

When No dependency passed-



```
useEffect(() => {
  //Runs on every render
});
```

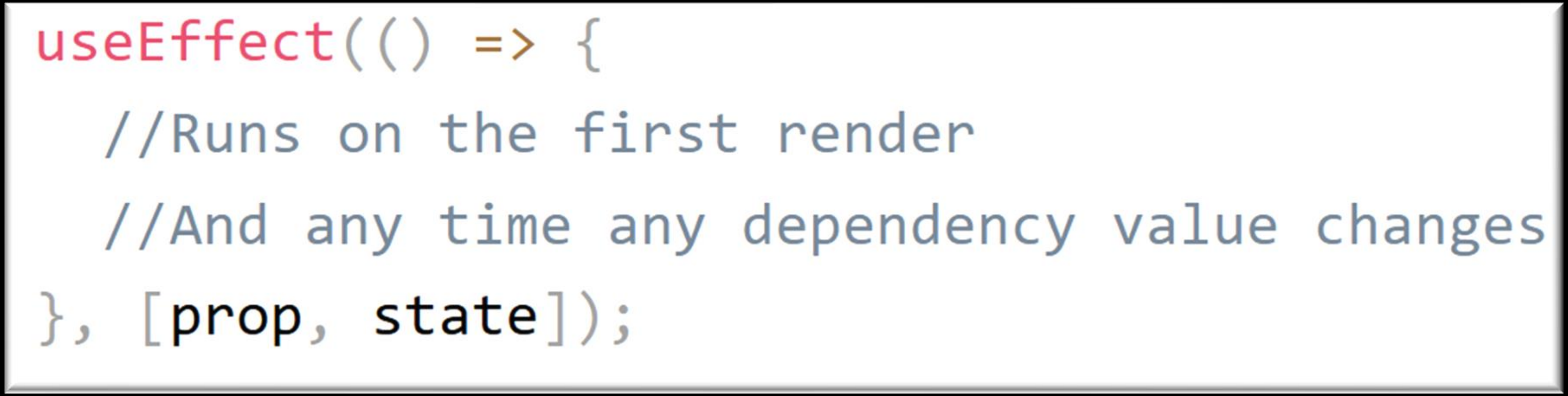
When An Empty array passed-



```
useEffect(() => {
  //Runs only on the first render
}, []);
```

2. useEffect( ) Hook-

When state values or props passed-



```
useEffect(() => {
  //Runs on the first render
  //And any time any dependency value changes
}, [prop, state]);
```

3. useRef( ) Hook-

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.

It can be used to access a DOM element directly

useRef( ) only returns one item . It returns an object called current.

When we initialize useRef we set the initial value useRef(0)

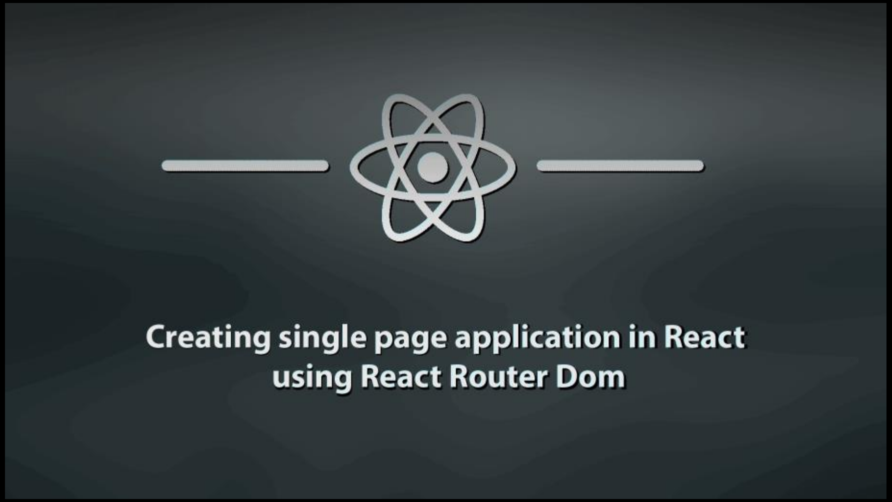
LEVEL-5

React Router

React Router

- React Router is a library for handling routing and navigation in React JS Application .

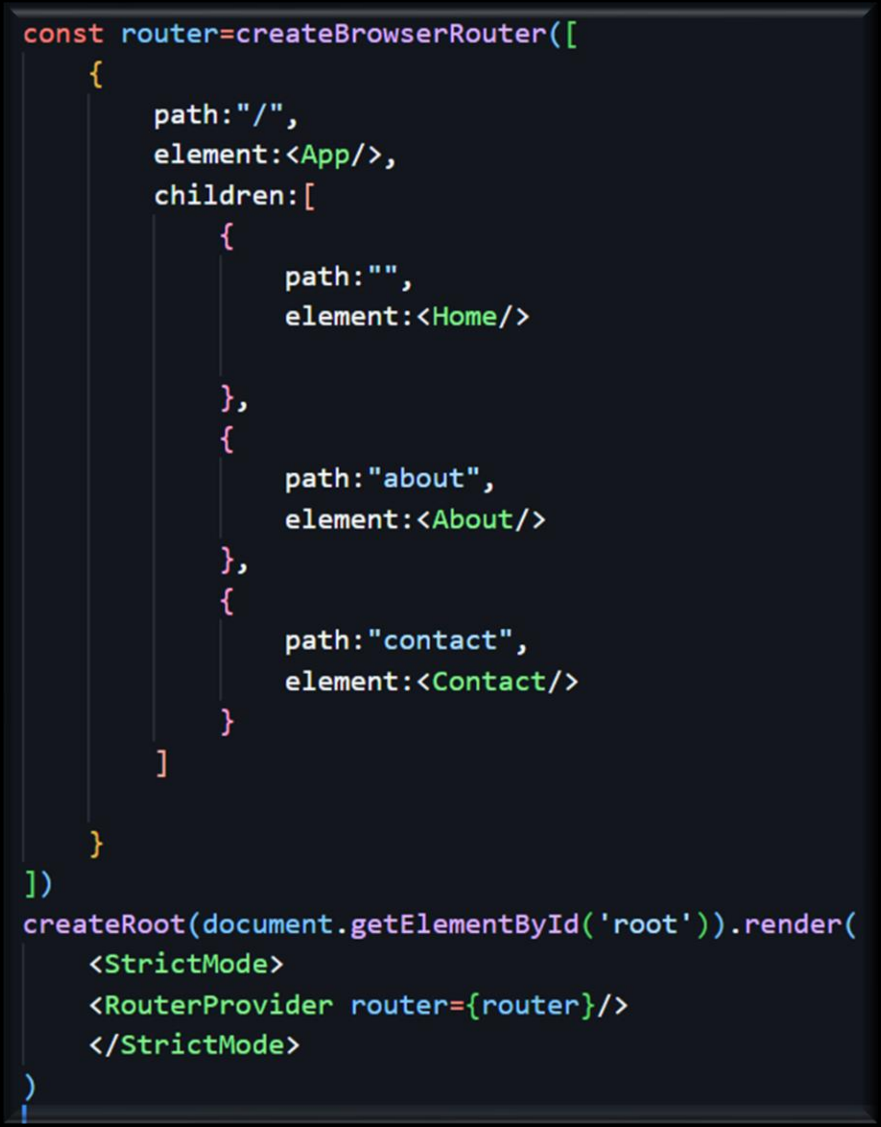
- It enables navigation in single page Application (SPA) without refreshing the entire page


Creating single page application in React using React Router Dom

React Router

VIRTUAL CODE

Method 1-

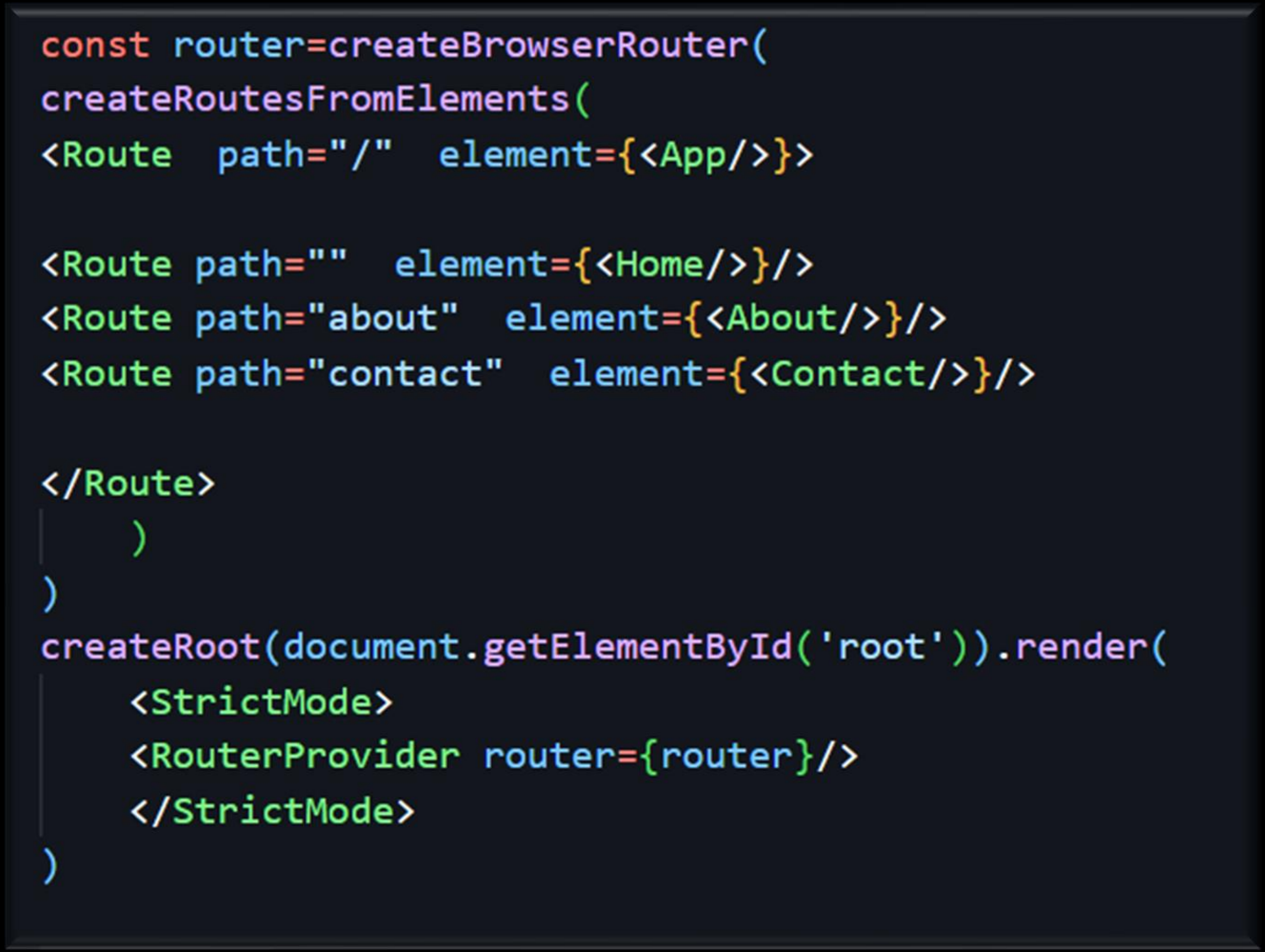


```
const router=createBrowserRouter([
    {
        path:"/",
        element:<App/>,
        children:[
            {
                path:"",
                element:<Home/>

            },
            {
                path:"about",
                element:<About/>
            },
            {
                path:"contact",
                element:<Contact/>
            }
        ]

    }
])
createRoot(document.getElementById('root')).render(
    <StrictMode>
    <RouterProvider router={router}/>
    </StrictMode>
)
```

React Router

VIRTUAL CODE

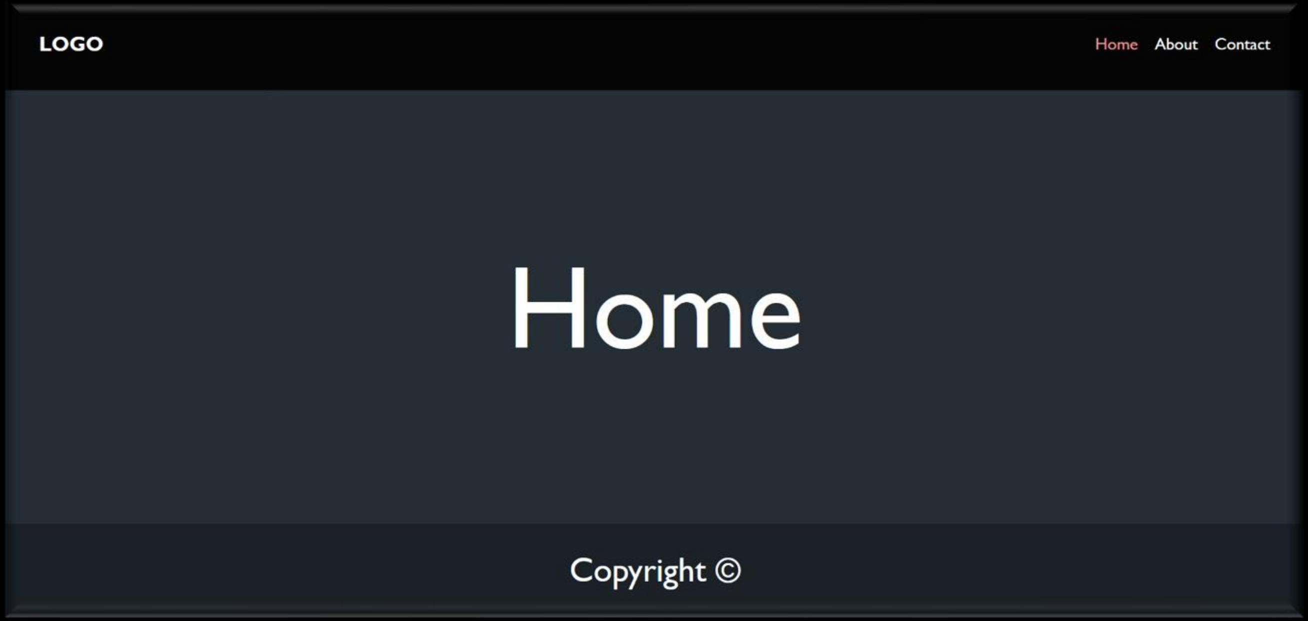
Method 2-



```
const router=createBrowserRouter(
createRoutesFromElements(
<Route  path="/"  element={<App/>}>

<Route path=""  element={<Home/>}/>
<Route path="about"  element={<About/>}/>
<Route path="contact"  element={<Contact/>}/>

</Route>
    )
)
createRoot(document.getElementById('root')).render(
    <StrictMode>
    <RouterProvider router={router}/>
    </StrictMode>
)
```

Practice Project 3

Create This single page application using react



LEVEL-6

Context API

Context API

- Context API in React is used to share data across the components without passing the props manually through every level.

- Context API is used to pass global variables anywhere in the code without the prop drilling

- It helps when there is a need for Sharing state between a lot of nested components.

- It is light in weight and easier to use, to create a context just need to call React.createContext() .

- No need to install other dependencies or third-party libraries like Redux for state management.

Context API

VIRTUAL CODE