

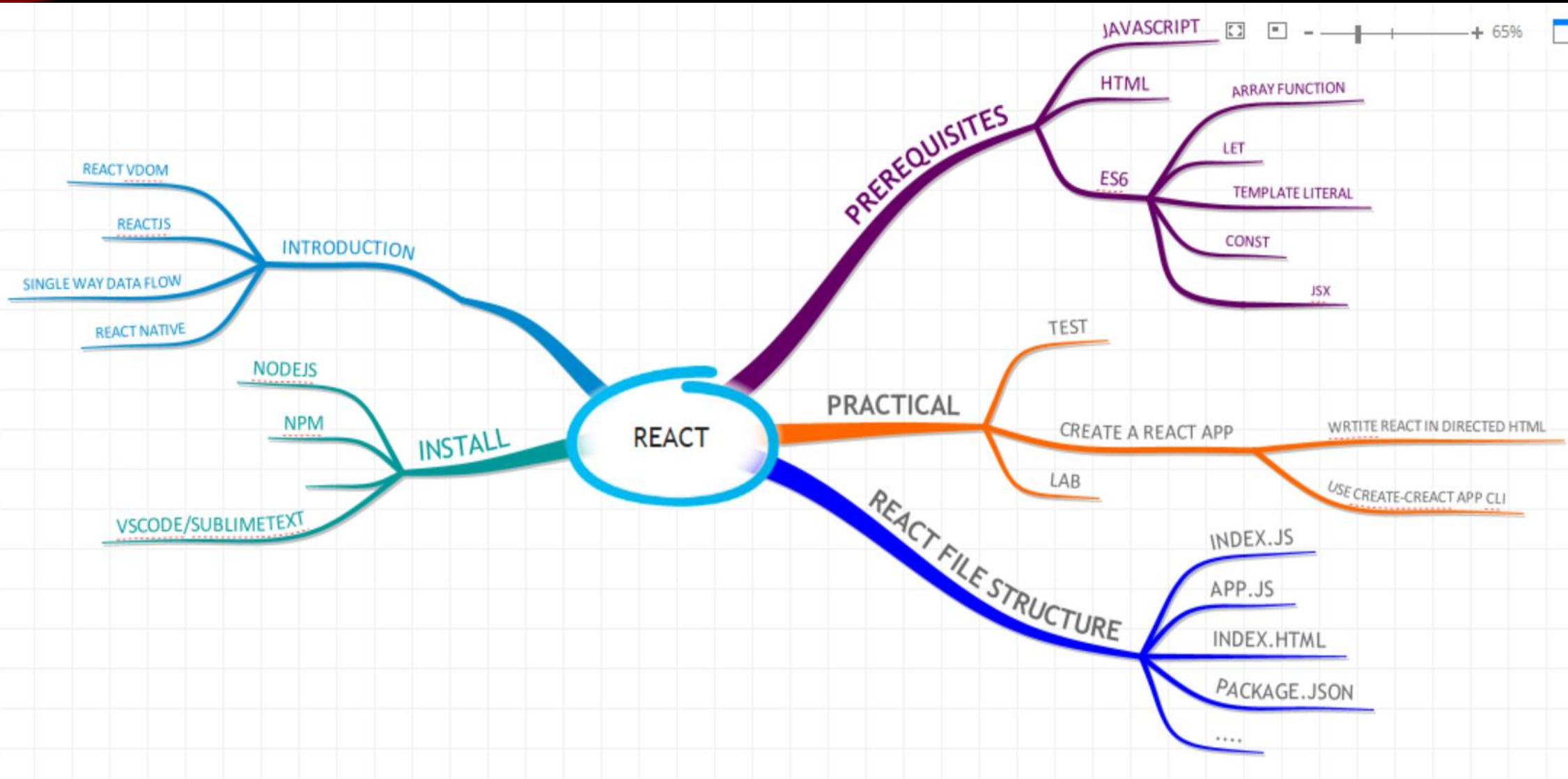
WEB DEVELOPMENT WITH



Session 1

React Introduction

OBJECTIVES



INTRODUCTION TO REACT

- React is a **JavaScript library** for creating user interfaces by Facebook and Instagram.
- React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “**components**”.
- React is a front-end library that is responsible for the user interface. It is neither a **whole framework nor a language**. It is an open-source JavaScript library for building the UI (user interface) of an application.

INTRODUCTION TO REACT

- React.js is the most popular front-end “framework” for Web applications.
- In this workshop, we will learn what React.js (or **simply React** or **Reactjs**) is and why we should use React.js instead of other JavaScript frameworks like Angular.

WHAT'S REACT ?



Open source library for building user interface.

Not a framework.

Focus on UI.

Rich ecosystem.

WHAT'S REACT ?

- React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications.
- It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.
- React allows developers to create large web applications that **can change data, without reloading the page**. The main purpose of React is to be fast, scalable, and simple.
- It works only on user interfaces in the application. This corresponds to the view in the MVC template.

PREREQUISITES OF REACT

- You should have knowledge of HTML, CSS, and JavaScript fundamentals.
- You should also be aware of ES6 : let & const, arrow functions, template literals, default parameters, object literals, rest and spread operators and destructuring assignment.
- Java Script - 'this' keyword, filter, map and reduce.

WHY LEARN REACT ?

- Created and maintained by Facebook.
- More than 100k stars on GitHub.
- Huge community.
- In demand skillset.

CREATE REACT APP

- In order to create a React application, we will be using the *create-react-app* *node_module* provided by React. This helps us by creating a consistent folder structure automatically so, that we don't need to care about proper and valid folder structure.

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE 1: powershell
PS D:\React2020WorkShop> npm i -g create-react-app
```

Then we make lab2-component react app

```
PS D:\React2020WorkShop> create-react-app lab2-component
```

Or

```
PS C:\Users\nguye\OneDrive\Desktop\WorkShopReact> npx create-react-app lab1
npx: installed 98 in 7.11s
```

Creating a new React app in C:\Users\nguye\OneDrive\Desktop\WorkShopReact\lab1.

CREATE REACT APP

```
npm test
```

Starts the test runner.

```
npm run eject
```

Removes this tool and copies build dependencies, configuration files and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

```
cd lab1
```

```
npm start
```

Happy hacking!

```
PS C:\Users\nguye\OneDrive\Desktop\WorkShopReact>
```

CREATE REACT APP

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Compiled successfully!

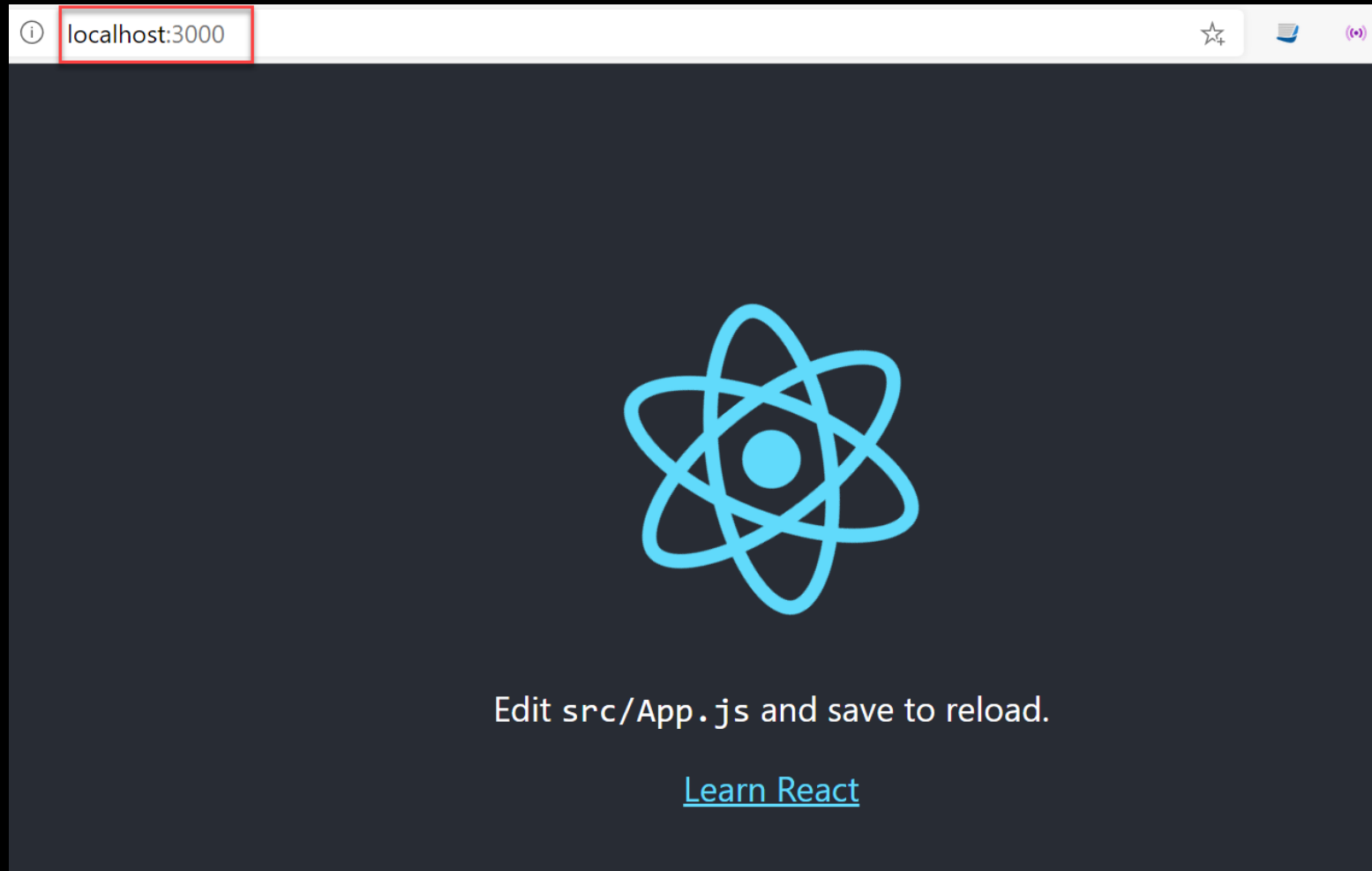
You can now view **lab1** in the browser.

Local: <http://localhost:3000>

On Your Network: <http://192.168.1.5:3000>

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

RUN THE REACT APPLICATION

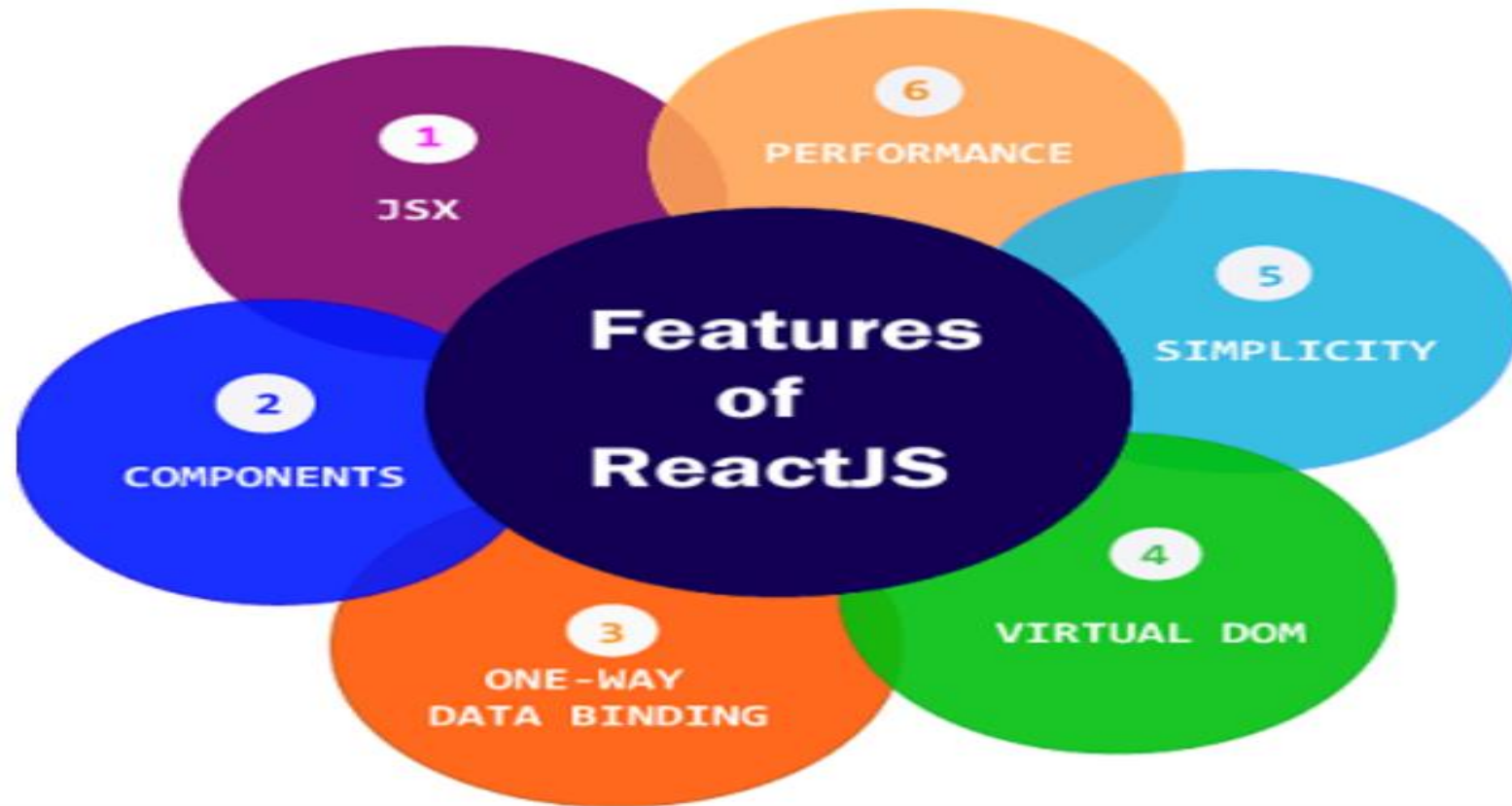


WHAT ARE THE REACT.JS FEATURES?

Let us take a closer look at some important features of React.

- JSX
- React Native
- Single-Way data flow
- Virtual Document Object Model
- Simplicity
- Performance

WHAT ARE THE REACT.JS FEATURES?



INTRODUCTION TO JSX

- In React, instead of using regular JavaScript for templating, it uses JSX.
- JSX is a simple JavaScript that allows HTML quoting and uses these HTML tag syntax to render subcomponents.
- HTML syntax is processed into JavaScript calls of React Framework. We can also write in pure old JavaScript.

INTRODUCTION TO JSX

- Consider this variable declaration:

```
const element = <h1>Hello, world!</h1>;
```

- This tag syntax is neither a string nor HTML.
- It is called JSX, and it is a syntax extension to JavaScript. We recommend using it with React to describe what the UI should look like.
- JSX may remind you of a template language, but it comes with the full power of JavaScript.
- JSX produces React “elements”.

INTRODUCTION TO JSX

JSX

- "JavaScript Syntax Extension"
- Allows us to write HTML-looking code inside of JavaScript.
- Generates React "elements"
- NOT STANDARD JS, WE NEED BABEL

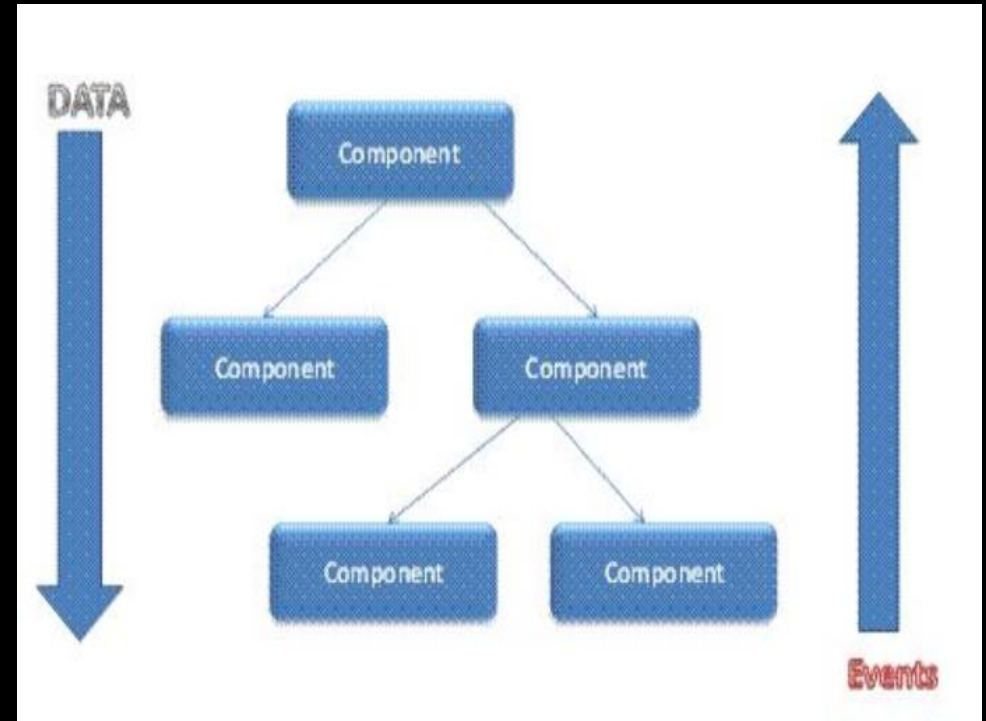
```
function Counter() {  
  const [ count, setCount ] = useState(0);  
  
  return (  
    <div>  
      <p>count: {count}</p>  
      <p>Is this JS or HTML??</p>  
    </div>  
  );  
}
```


REACT NATIVE

- React has native libraries that were announced by Facebook in 2015, which provides the react architecture to native applications like IOS, Android and UPD.
- React-native is a mobile apps building framework using only Java script. It uses the same design as React, letting you utilize/include a rich mobile UI library/declarative components.
- It uses the same fundamental UI building blocks as regular iOS and Android apps. The best part of using react-native is to allow/adopt components written in Objective-C, Java, or Swift.

SINGLE-WAY DATA FLOW

- In React, a set of immutable values are passed to the components renderer as properties in its HTML tags. The component cannot directly modify any properties but can pass a call back function with the help of which we can do modifications.
- This complete process is known as “properties flow down; actions flow up”.



VIRTUAL DOCUMENT OBJECT MODEL

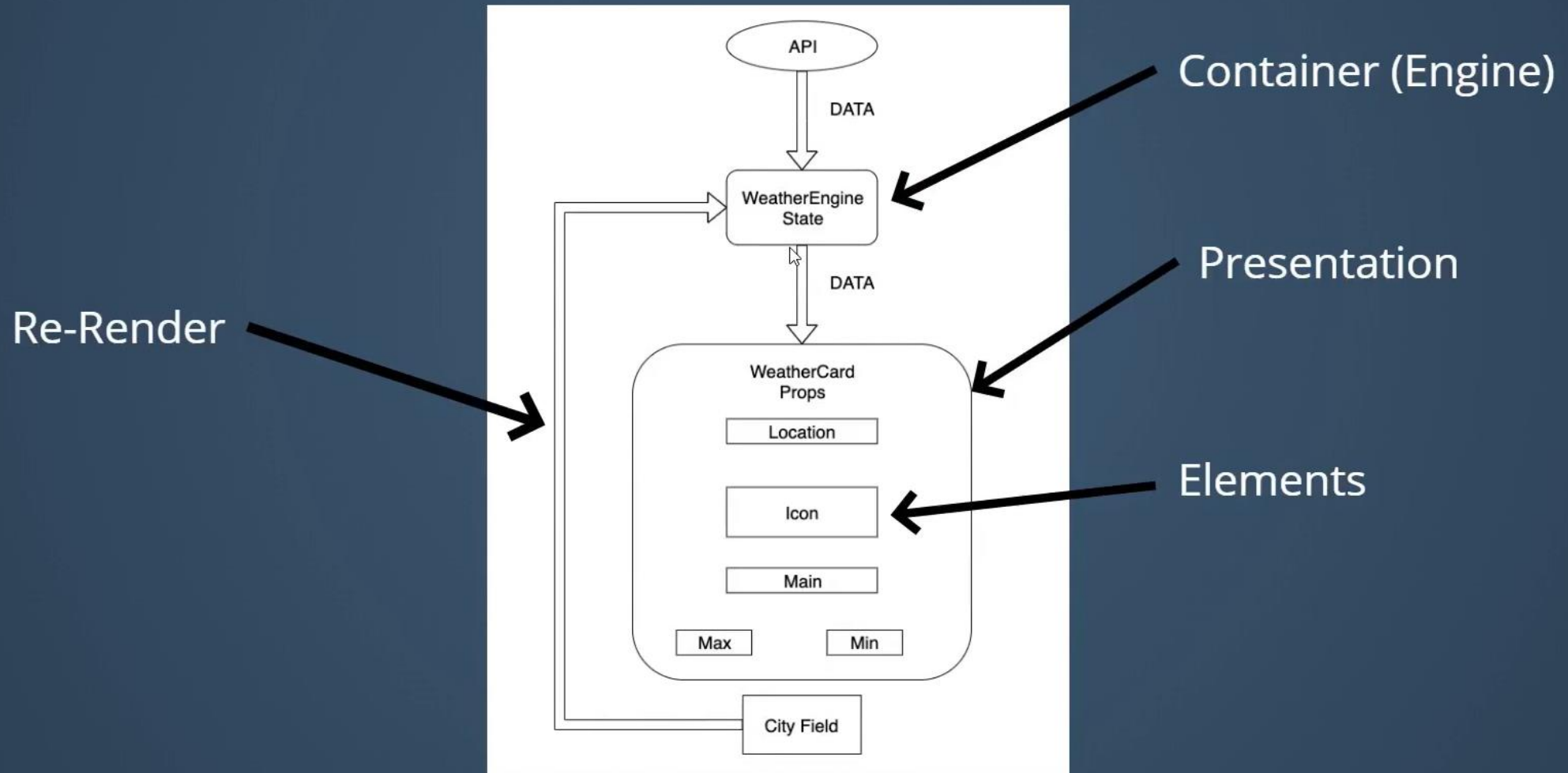
- React creates an in-memory data structure cache which computes the changes made and then updates the browser.
- This allows a special feature that enables the programmer to code as if the whole page is rendered on each change whereas react library only renders components that actually change.
- In React, for every DOM object, there is a corresponding “virtual DOM object.” A virtual DOM object is a *representation* of a DOM object, like a lightweight copy.

VIRTUAL DOCUMENT OBJECT MODEL

- A virtual DOM object has the same properties as a real DOM object, but it lacks the real thing's power to directly change what's on the screen.
- Manipulating the DOM is slow. Manipulating the virtual DOM is much faster, because nothing gets drawn onscreen. Think of manipulating the virtual DOM as editing a blueprint, as opposed to moving rooms in an actual house.

DATA FLOW IN REACT

One direction data flow



ACTIVITY

Create a React app named “Demo”. When the app run, it displays below:



Welcom to React WorkShop FPT Aptect 2020.

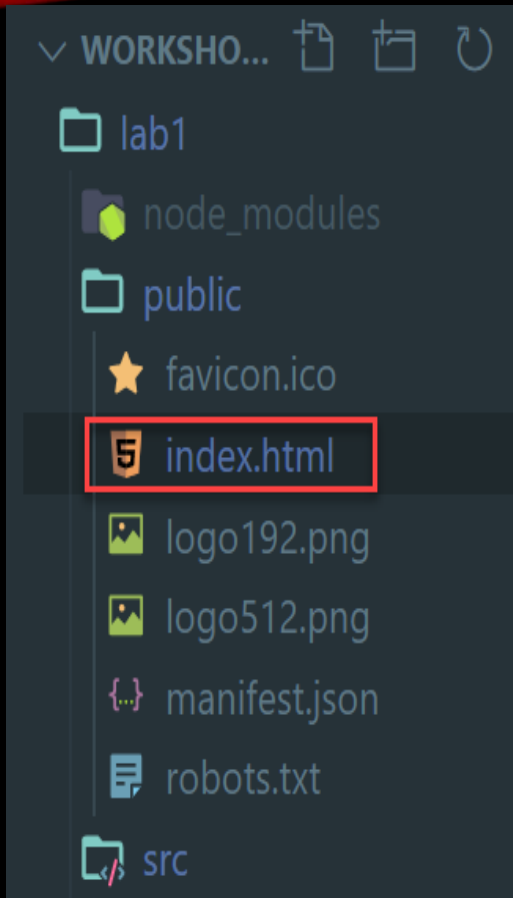
FILE REACT ORGANIZATION

The file-structure example app demonstrates one possible way of organizing your React app.

While there is not necessarily one file structure that is better than another, it is important to keep your files organized.

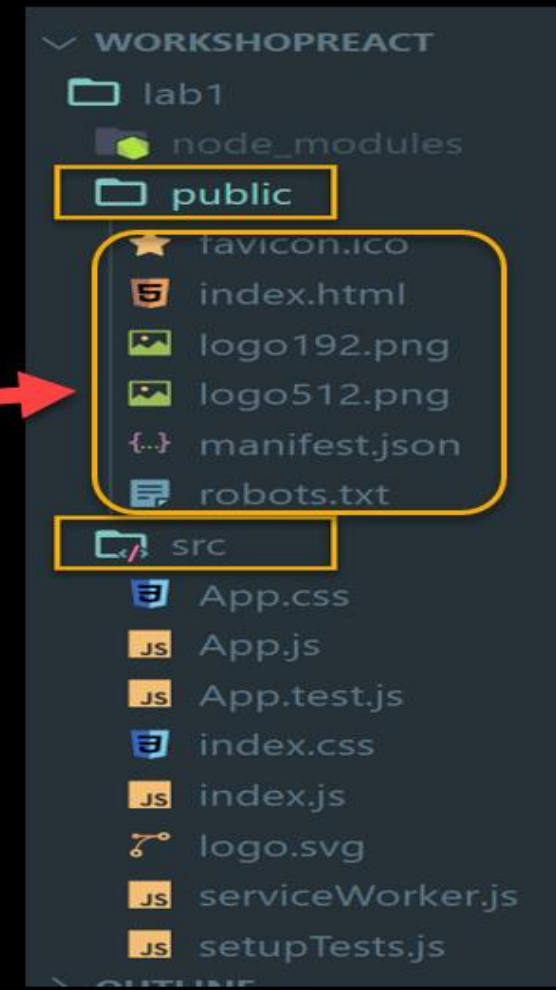
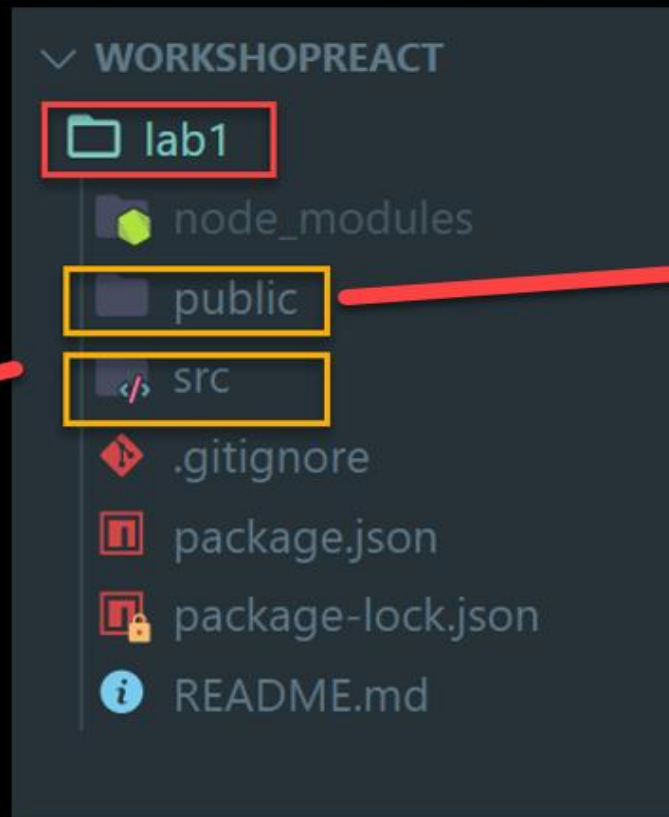
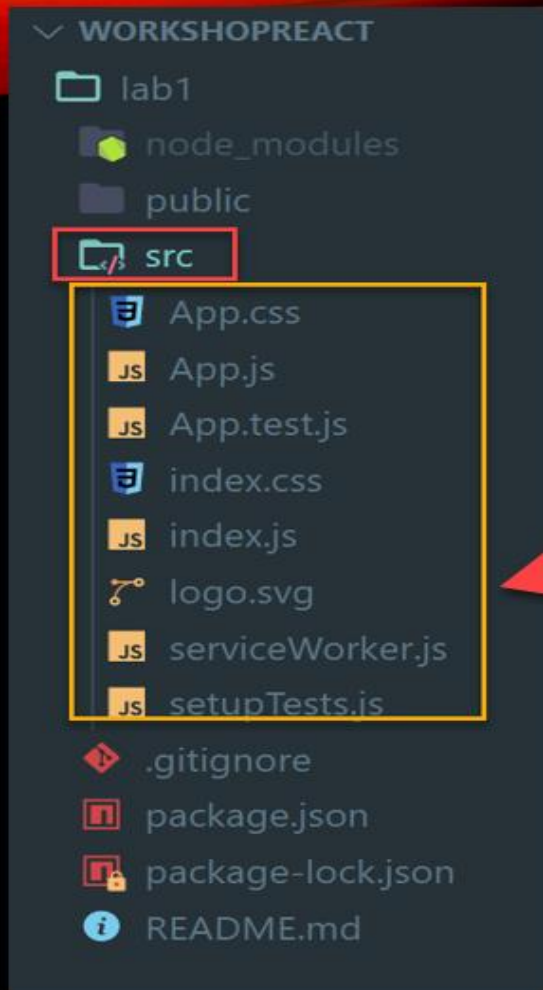
In React your file structure will grow rapidly considering every component has at least one file associated with it.

FILE REACT ORGANIZATION



```
index.html × App.js index.js package.json
lab1 > public > index.html > html > head
You, a few seconds ago | 1 author (You)
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8" />
6   <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
7   <meta name="viewport" content="width=device-width, initial-scale=1" />
8   <meta name="theme-color" content="#000000" />
9   <meta name="description" content="Web site created using create-react-app" />
10  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
11
12  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
13
14  <title>React App</title>
15 </head>
16
17 <body>
18   <noscript>You need to enable JavaScript to run this app.</noscript>
19   <div id="root"></div>
20 </body>
21
22 </html>
```

FILE REACT ORGANIZATION



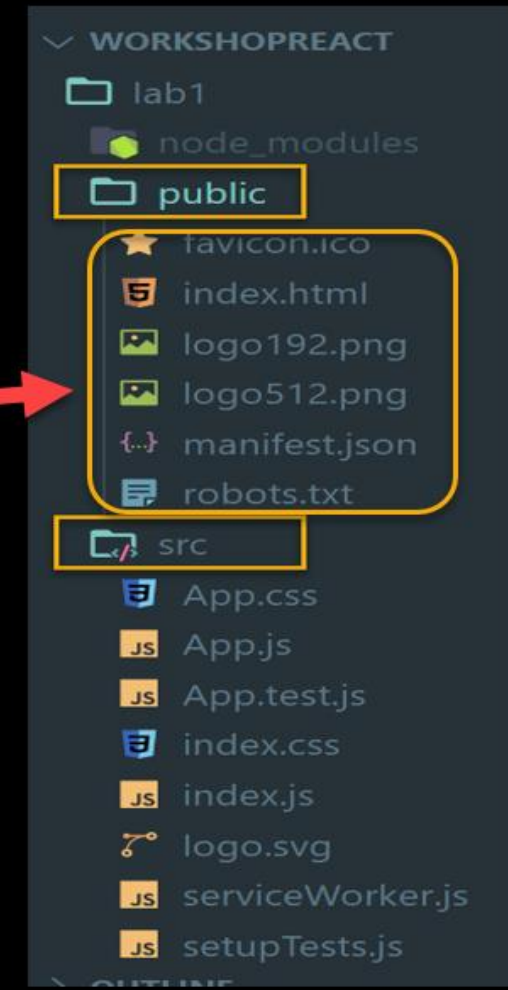
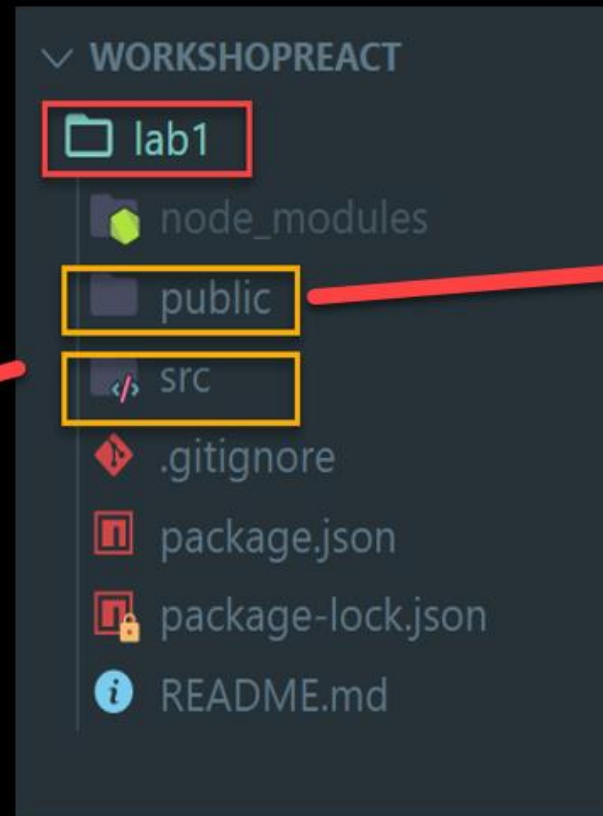
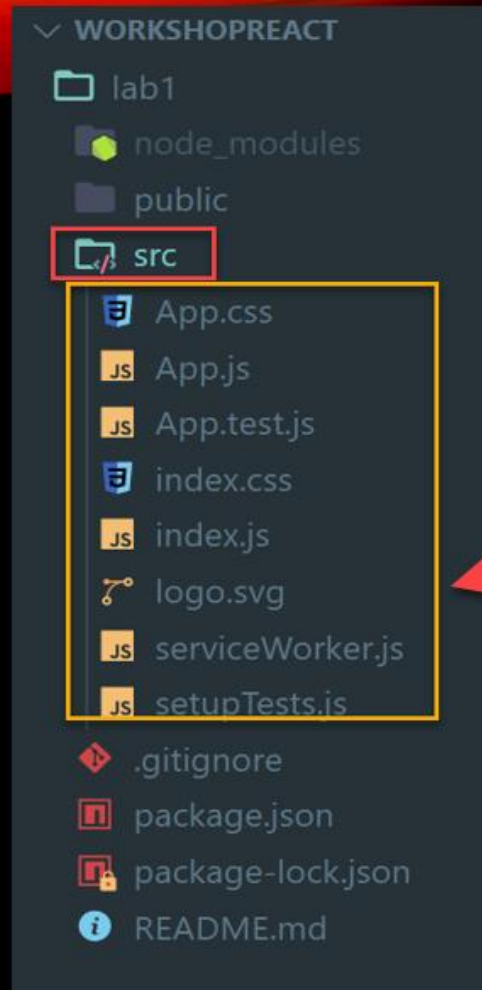
FILE REACT ORGANIZATION

The image shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'WORKSHOPREACT' with a 'src' directory containing 'App.js', 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'serviceWorker.js', and 'setupTests.js'. The 'App.js' file is selected and highlighted with a yellow box. A red arrow points from this box to the 'function App()' line in the code editor. The code editor shows the following code:

```
lab1 > src > JS App.js
4
5 function App() {
6   return ( <
7     div className = "App" >
8       <header className = "App-header" >
9         <img src = { logo }
10          className = "App-logo"
11          alt = "logo" / >
12       <p>
13         Welcom to React WorkShop FPT Aptect 2020.
14       </p>
15       <a className = "App-link"
16         href = "https://reactjs.org"
17         target = "_blank"
18         rel = "noopener noreferrer" >
19         Learn React </a>
20       </header> </div>
21   );
22 }
23
24 export default App;
```

The 'function App()' line is highlighted with a red box, and the 'export default App;' line is also highlighted with a red box.

FILE REACT ORGANIZATION



FILE REACT ORGANIZATION

WORKSHOPREACT lab1 > src > JS index.js

lab1

- node_modules
- public
 - favicon.ico
 - index.html
 - logo192.png
 - logo512.png
 - manifest.json
 - robots.txt
- src
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js**
 - logo.svg
 - serviceWorker.js
 - setupTests.js
 - .gitignore

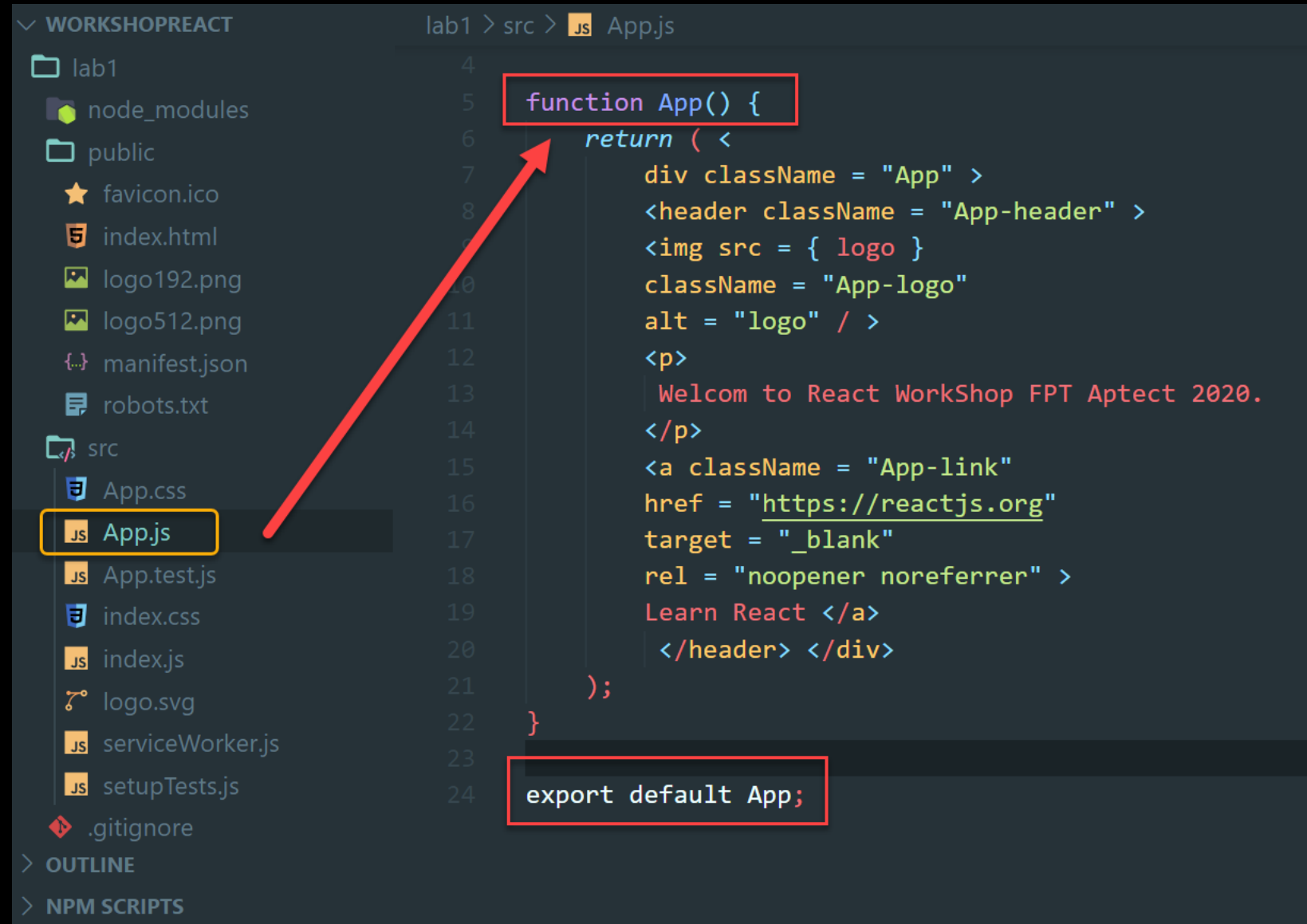
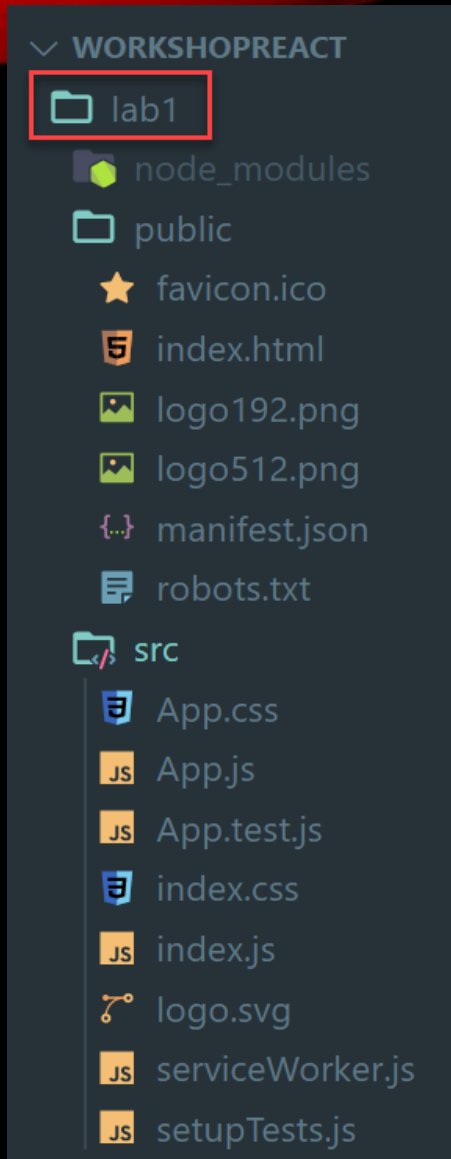
```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
6
7 ReactDOM.render(
8   <React.StrictMode>
9     <App />
10   </React.StrictMode>,
11   document.getElementById('root')
12 );
13
14 serviceWorker.unregister();
15
```

FILE REACT ORGANIZATION

The image shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'WORKSHOPREACT' with a folder 'lab1' containing files like 'node_modules', 'public', 'src', 'App.css', 'App.js', 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'serviceWorker.js', 'setupTests.js', '.gitignore', 'package.json' (highlighted with a yellow box), 'package-lock.json', and 'README.md'. The code editor shows the content of 'package.json' with line numbers 1 to 24. A red box highlights the 'dependencies' section of the file.

```
lab1 > package.json
1  {
2    "name": "lab1",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^4.2.4",
7      "@testing-library/react": "^9.5.0",
8      "@testing-library/user-event": "^7.2.1",
9      "react": "^16.13.1",
10     "react-dom": "^16.13.1",
11     "react-scripts": "3.4.3"
12   },
13   "scripts": {
14     "start": "react-scripts start",
15     "build": "react-scripts build",
16     "test": "react-scripts test",
17     "eject": "react-scripts eject"
18   },
19   "eslintConfig": {
20     "extends": "react-app"
21   },
22   "browserslist": {
23     "production": [
24       ">0.2%",
```

FILE REACT ORGANIZATION



FILE REACT ORGANIZATION

WORKSHOPREACT lab1 > src > JS index.js

lab1

- node_modules
- public
 - favicon.ico
 - index.html
 - logo192.png
 - logo512.png
 - manifest.json
 - robots.txt
- src
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js**
 - logo.svg
 - serviceWorker.js
 - setupTests.js
 - .gitignore

```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  ReactDOM.render(
8    <React.StrictMode>
9      <App />
10    </React.StrictMode>,
11    document.getElementById('root')
12  );
13
14  serviceWorker.unregister();
15
```


FILE REACT ORGANIZATION

WORKSHOPREACT

- lab1
 - node_modules
 - public
 - src
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js**
 - logo.svg
 - serviceWorker.js
 - setupTests.js
 - .gitignore
 - package.json
 - package-lock.json
 - README.md

lab1 > src > JS index.js

You, 9 minutes ago | 1 author (You)

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
```

```
6
7 ReactDOM.render(
8   <React.StrictMode>
9     <App />
10  </React.StrictMode>,
11  document.getElementById('root')
12 );
```

```
13
14 // If you want your app to work offline and load faster, you can change
15 // unregister() to register() below. Note this comes with some pitfalls.
16 // Learn more about service workers: https://bit.ly/CRA-PWA
17 serviceWorker.unregister();
18
```


The image features a solid black background. At the top, there is a decorative, wavy border with a color gradient. From left to right, the colors transition from a bright yellow, through orange and red, into a dark green, and finally into a light cyan/blue at the far right edge. The waves of the border are smooth and fluid, creating a sense of motion.

THE END