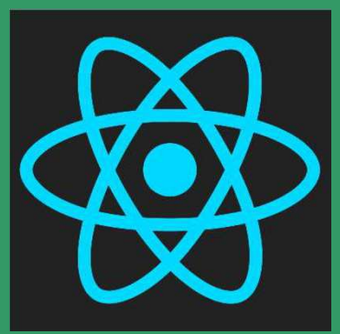
React JS Hello World

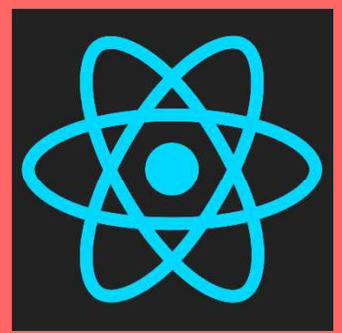


The React HelloWorld

Setup for Development Environment

React We Need

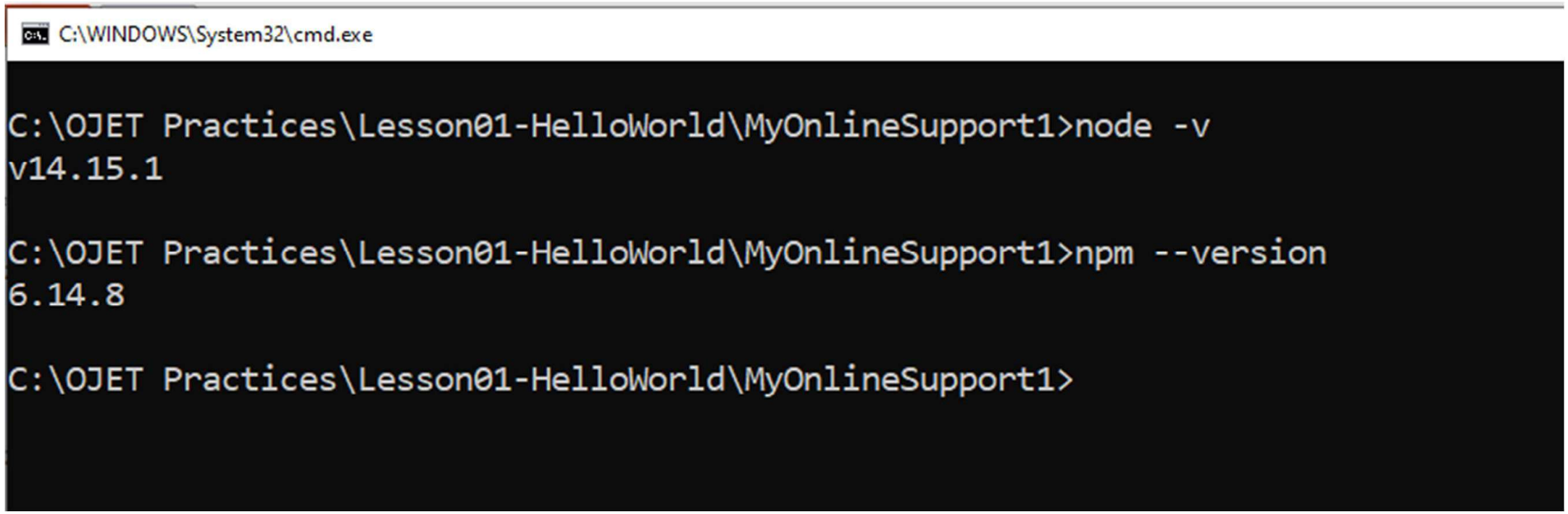
- Node : www.nodejs.org
- Text Editor : VS Code



Installation

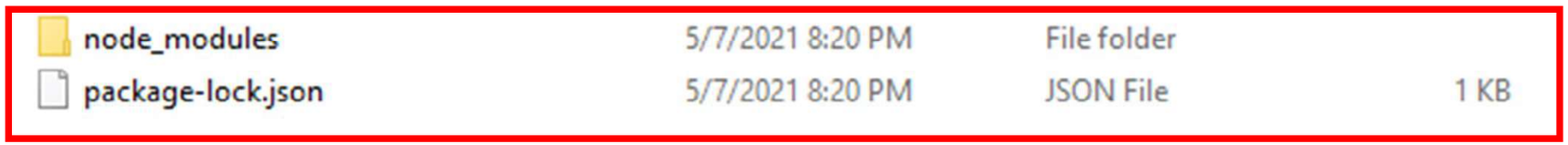
https://nodejs.org/en/

Type these Commands in Command Prompt

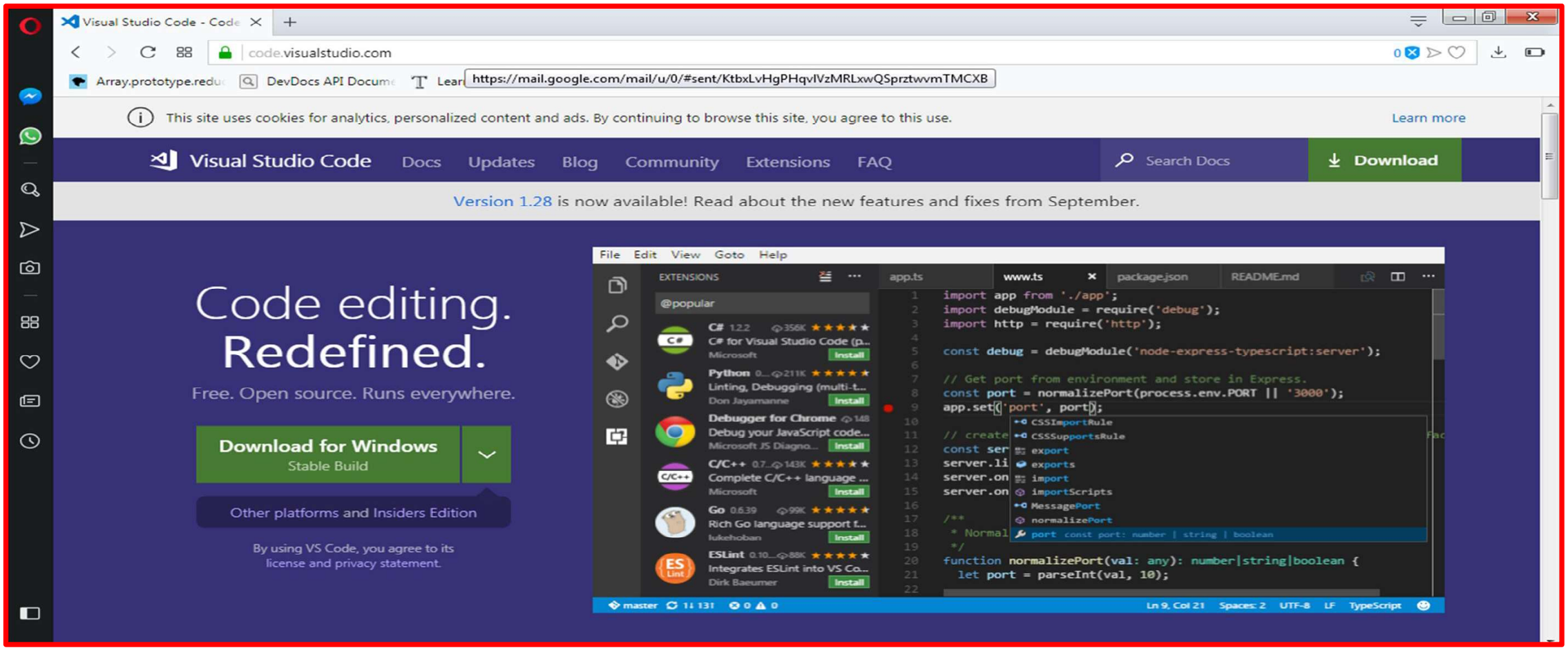


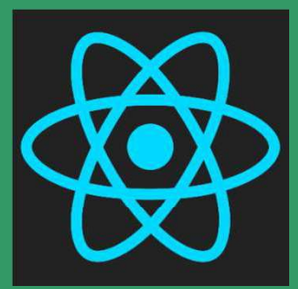
```
npm install [insert library name]
```

- The command npm install will look up the specified library on the NPM repository and install the library into a folder called node_modules.
- This folder will store all the JavaScript libraries installed via NPM, such as RequireJS and KnockoutJS.
- Running the command npm install jquery in an empty directory will download jQuery into an automatically created node_modules folder.

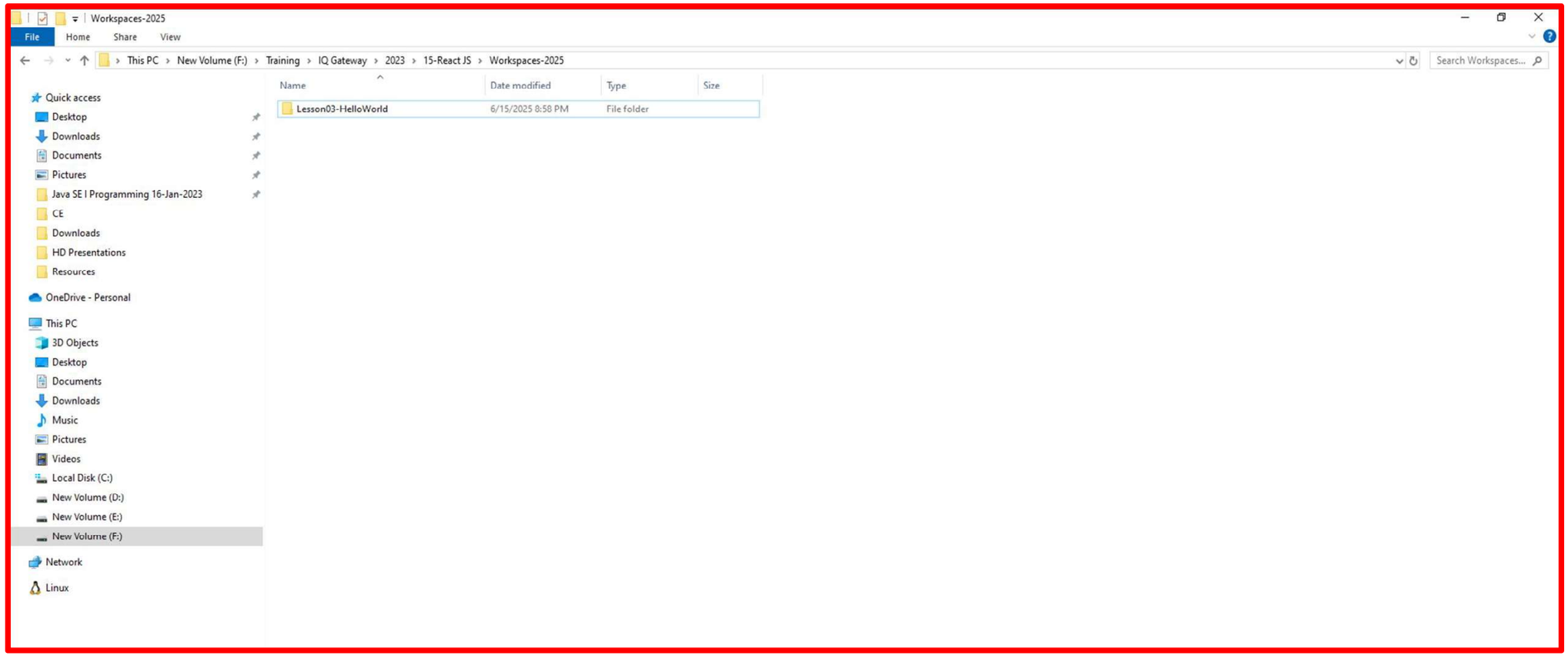


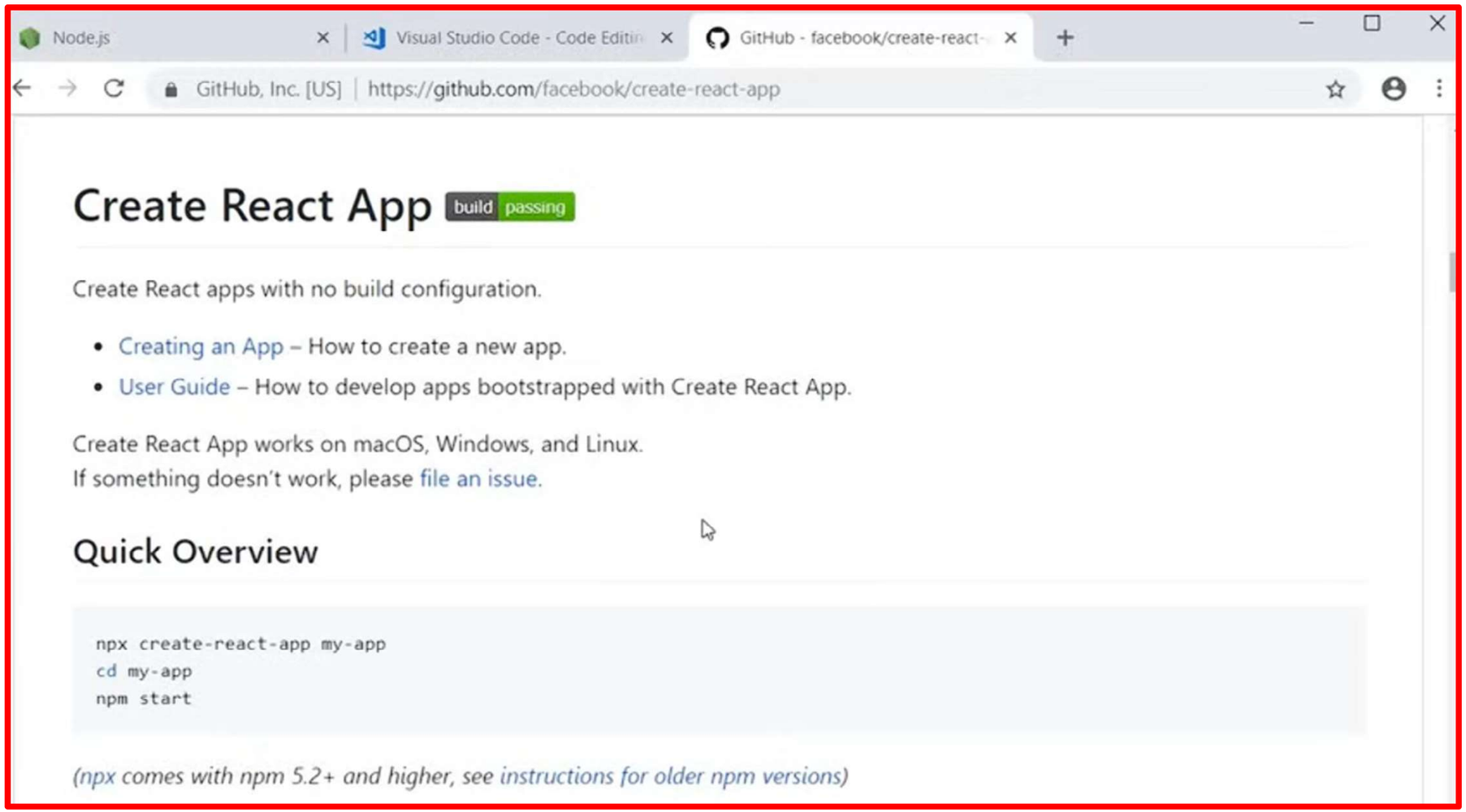
Visual Studio Code

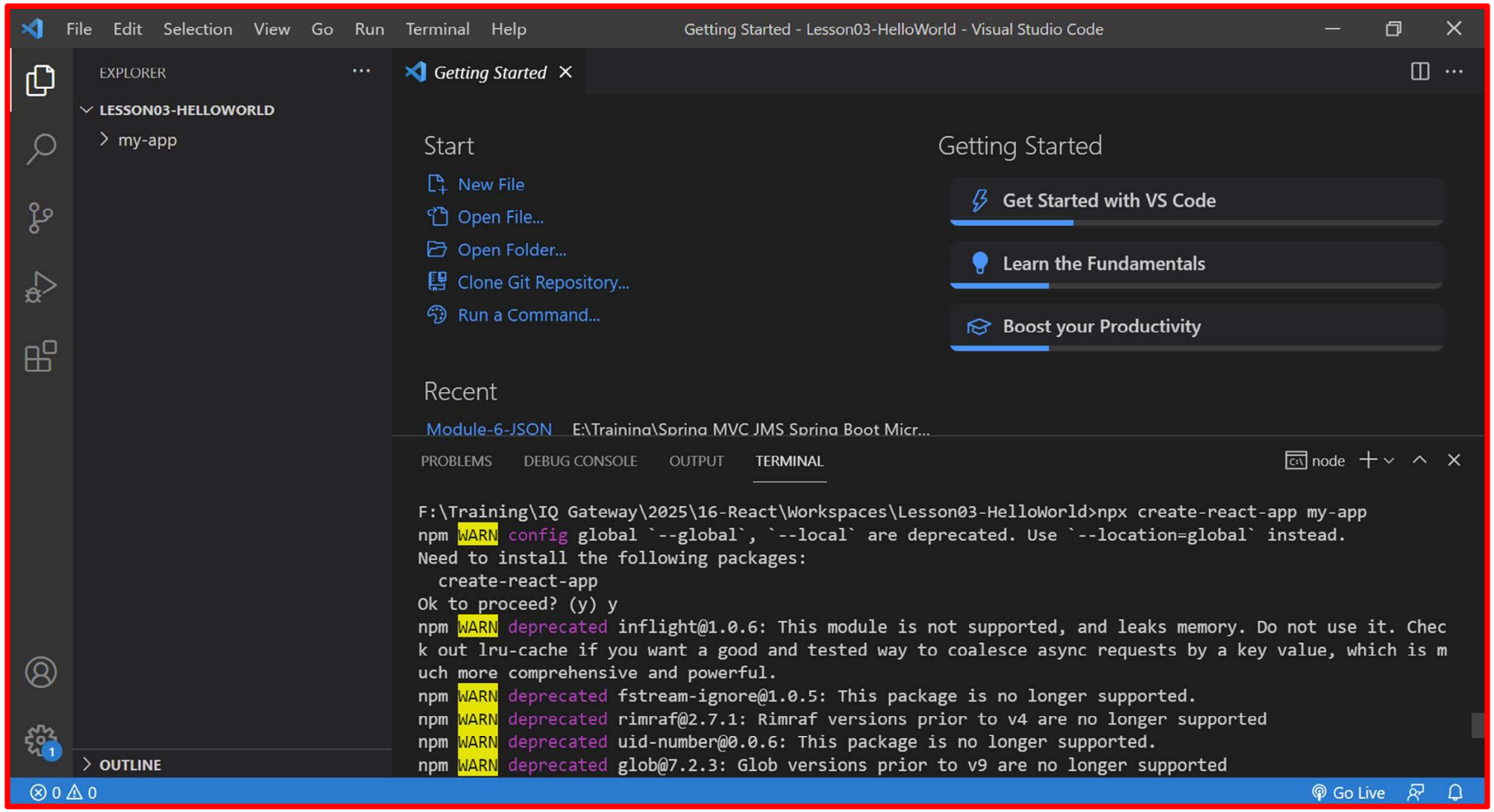


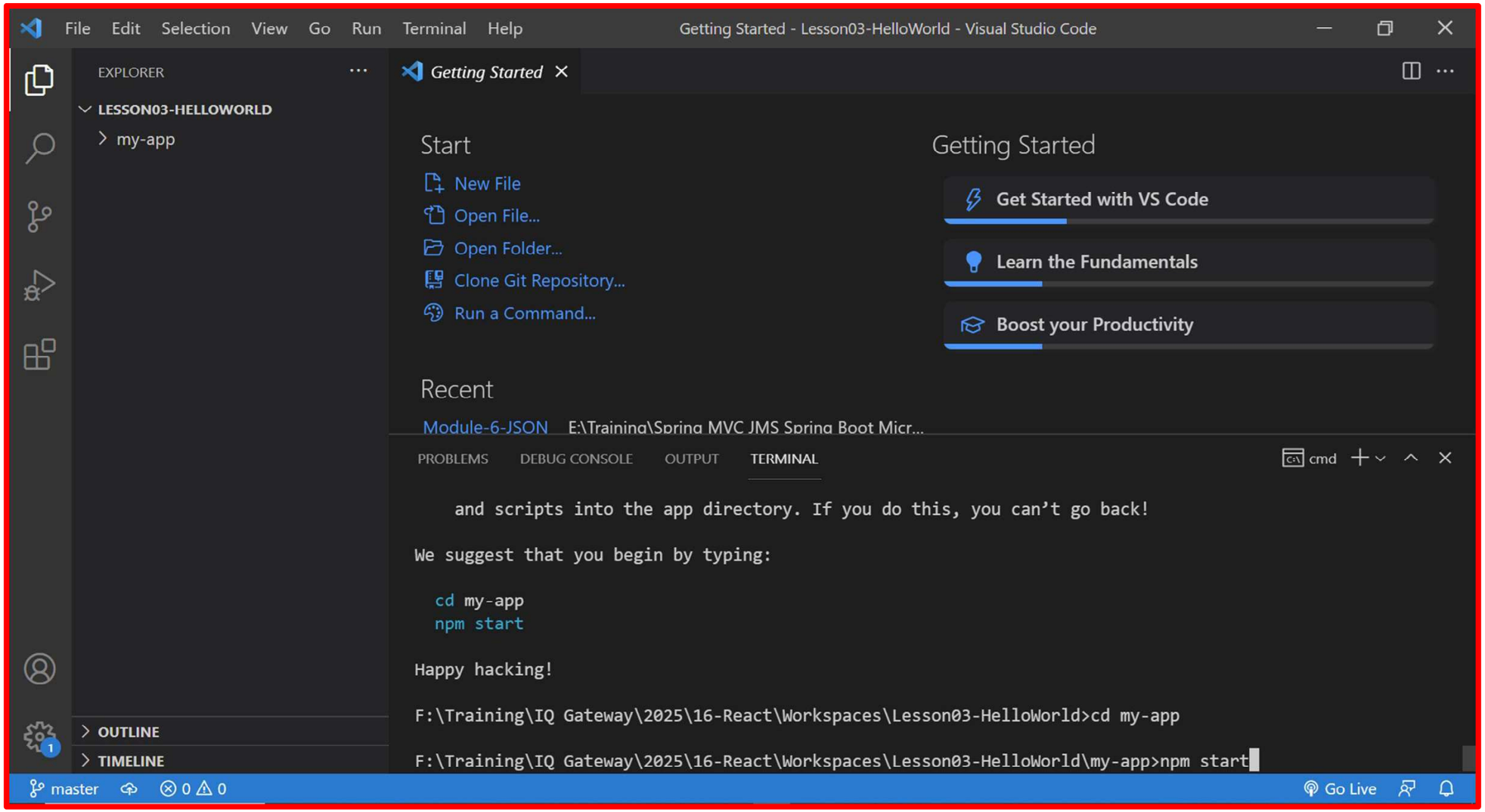


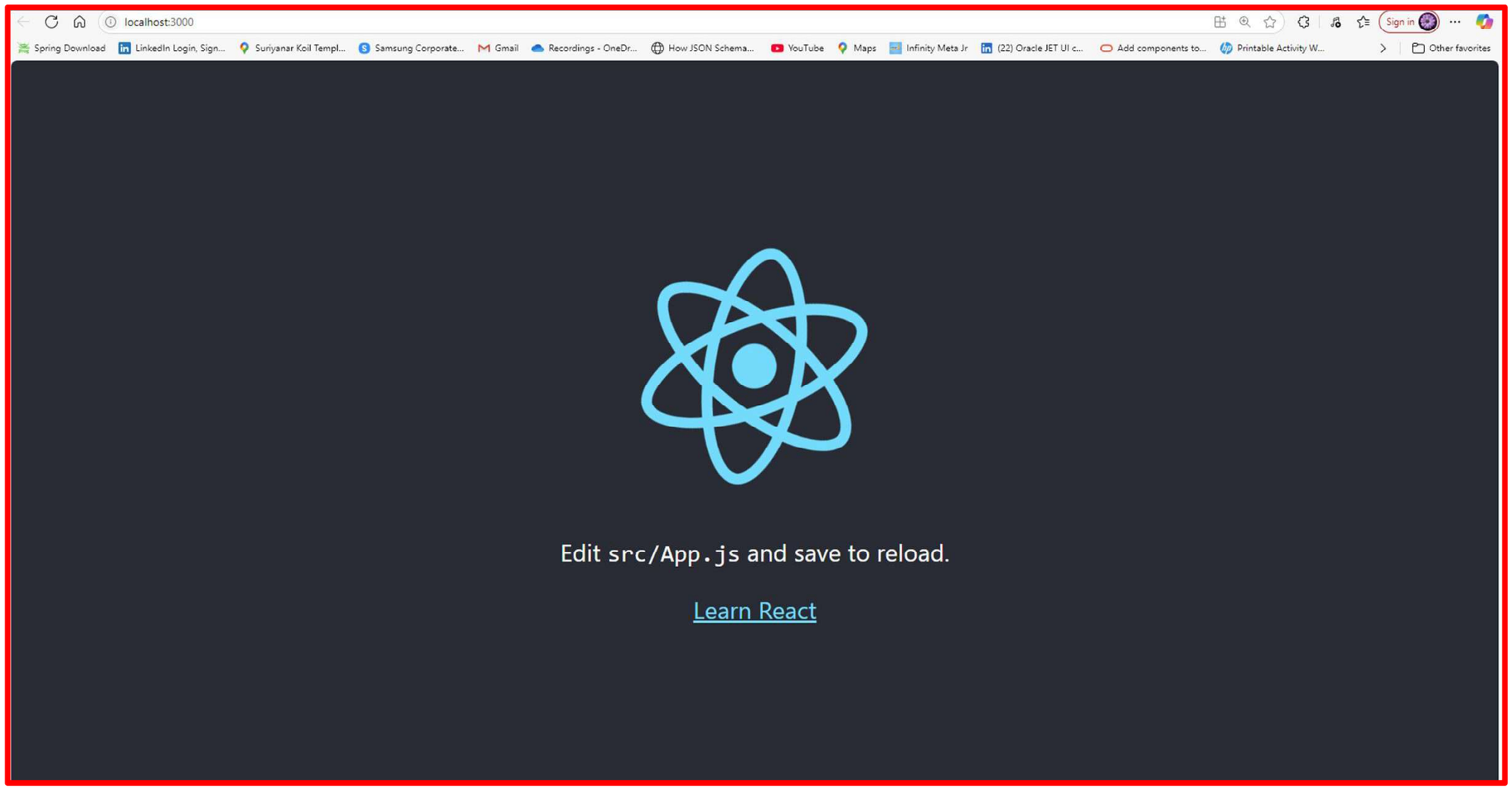
Scaffolding an Application


Workspaces-2025
File
Home
Share
View
This PC › New Volume (F:) › Training › IQ Gateway › 2023 › 15-React JS › Workspaces-2025
Search Workspaces...
Quick access
Desktop
Downloads
Documents
Pictures
Java SE I Programming 16-Jan-2023
CE
Downloads
HD Presentations
Resources
OneDrive - Personal
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
New Volume (D:)
New Volume (E:)
New Volume (F:)
Network
Linux
Name
Date modified
Type
Size
Lesson03-HelloWorld
6/15/2025 8:58 PM
File folder

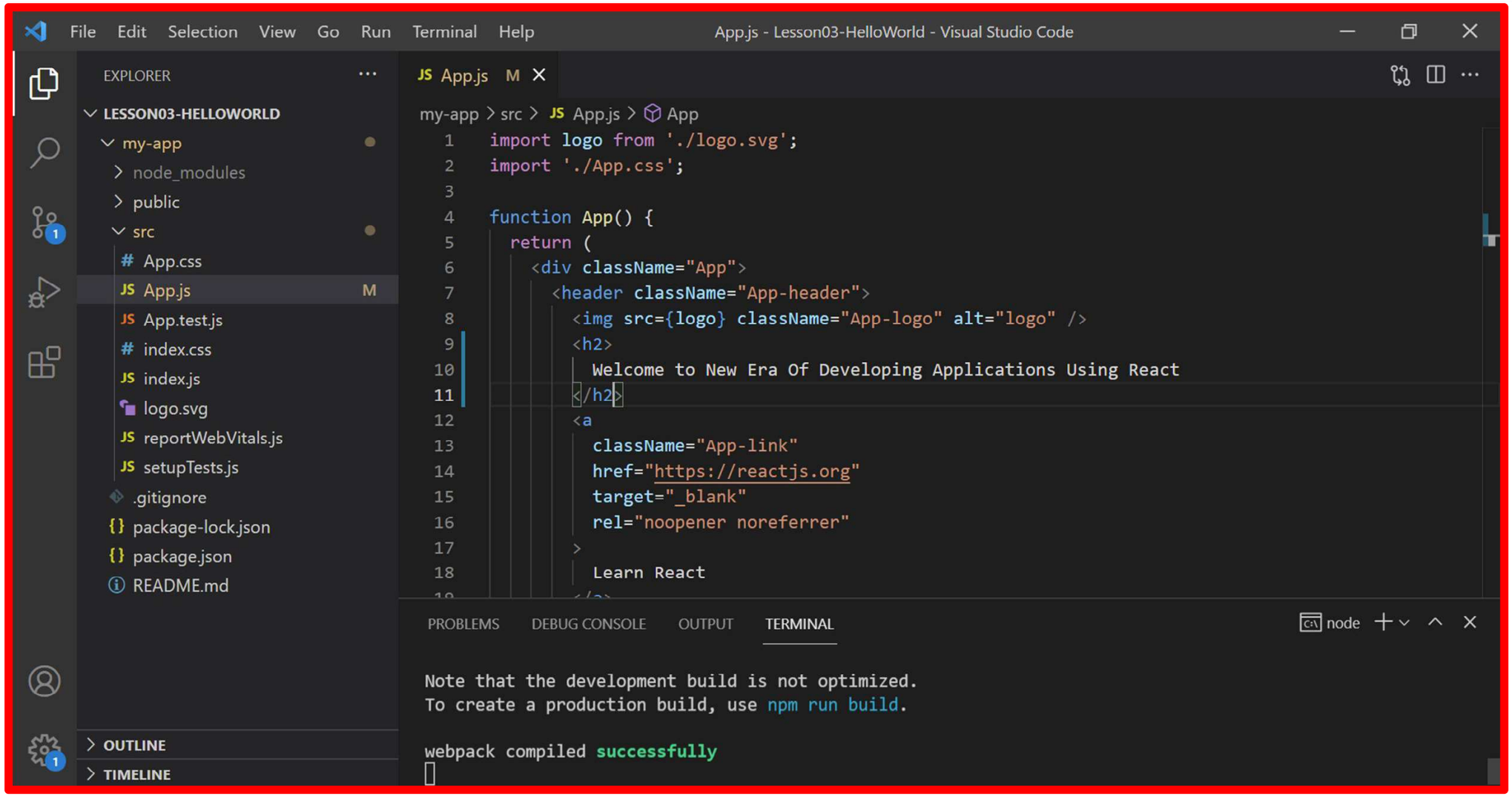

Node.js
Visual Studio Code - Code Editin
GitHub - facebook/create-react-
GitHub, Inc. [US] | https://github.com/facebook/create-react-app
Create React App
build passing
Create React apps with no build configuration.
Creating an App – How to create a new app.
User Guide – How to develop apps bootstrapped with Create React App.
Create React App works on macOS, Windows, and Linux.
If something doesn't work, please file an issue.
Quick Overview
npx create-react-app my-app
cd my-app
npm start
(npx comes with npm 5.2+ and higher, see instructions for older npm versions)

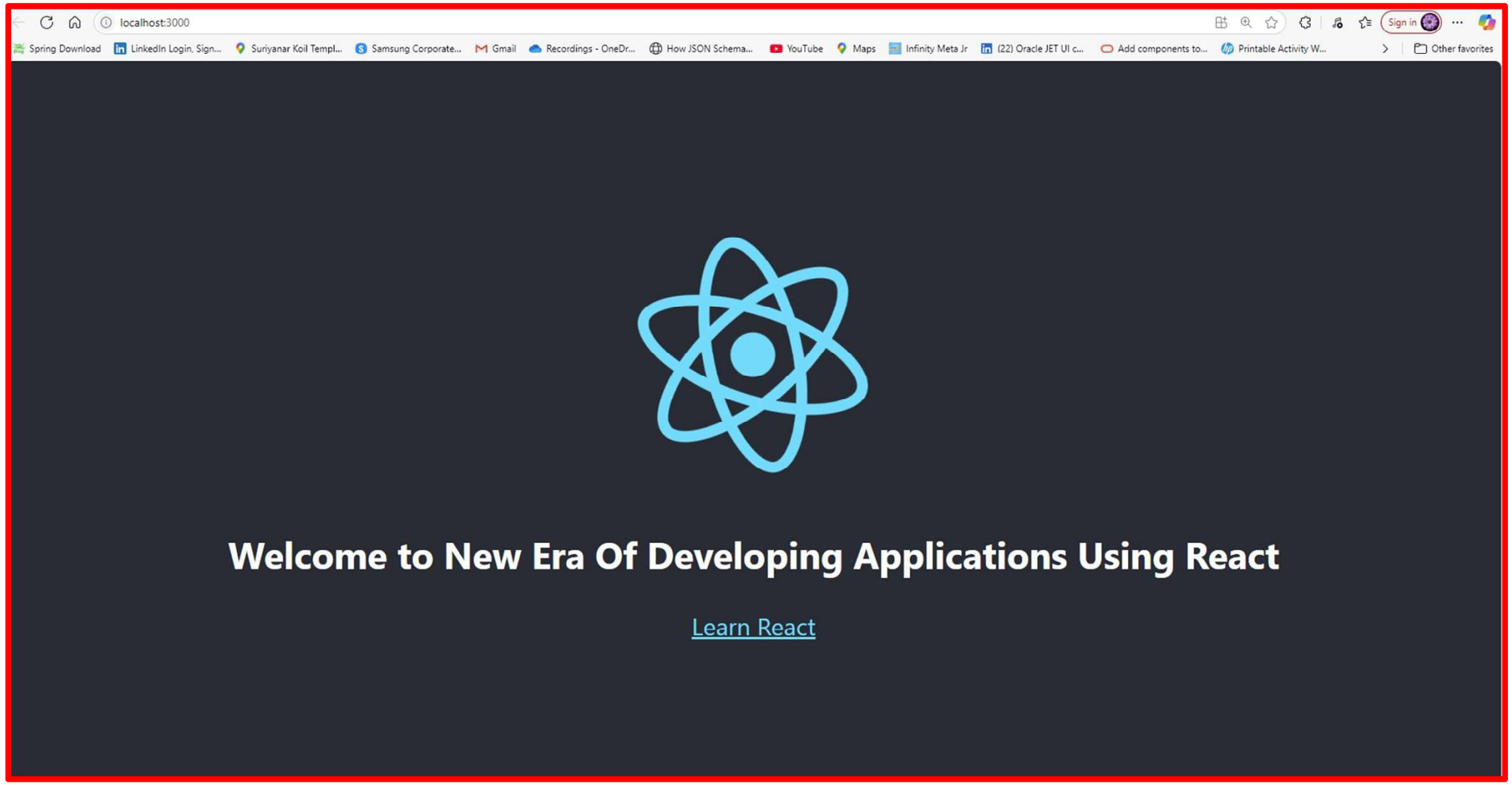

File Edit Selection View Go Run Terminal Help
Getting Started - Lesson03-HelloWorld - Visual Studio Code
EXPLORER
LESSON03-HELLOWORLD
my-app
Getting Started
Start
New File
Open File...
Open Folder...
Clone Git Repository...
Run a Command...
Getting Started
Get Started with VS Code
Learn the Fundamentals
Boost your Productivity
Recent
Module-6-JSON E:\Training\Spring MVC JMS Spring Boot Micr...
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
node
F:\Training\IQ Gateway\2025\16-React\Workspaces\Lesson03-HelloWorld>npx create-react-app my-app
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
Need to install the following packages:
create-react-app
Ok to proceed? (y) y
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Chec
k out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is m
uch more comprehensive and powerful.
npm WARN deprecated fstream-ignore@1.0.5: This package is no longer supported.
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated uid-number@0.0.6: This package is no longer supported.
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
OUTLINE
0 0
Go Live


File Edit Selection View Go Run Terminal Help
Getting Started - Lesson03-HelloWorld - Visual Studio Code
EXPLORER
LESSON03-HELLOWORLD
my-app
Getting Started
Start
New File
Open File...
Open Folder...
Clone Git Repository...
Run a Command...
Getting Started
Get Started with VS Code
Learn the Fundamentals
Boost your Productivity
Recent
Module-6-JSON E:\Training\Spring MVC JMS Spring Boot Micr...
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
cmd
and scripts into the app directory. If you do this, you can't go back!
We suggest that you begin by typing:
cd my-app
npm start
Happy hacking!
F:\Training\IQ Gateway\2025\16-React\Workspaces\Lesson03-HelloWorld>cd my-app
F:\Training\IQ Gateway\2025\16-React\Workspaces\Lesson03-HelloWorld\my-app>npm start
OUTLINE
TIMELINE
master
0 0
Go Live


localhost:3000
Sign in
Spring Download
LinkedIn Login, Sign...
Suriyanar Koil Templ...
Samsung Corporate...
Gmail
Recordings - OneDr...
How JSON Schema...
YouTube
Maps
Infinity Meta Jr
(22) Oracle JET UI c...
Add components to...
Printable Activity W...
Other favorites
Edit src/App.js and save to reload.
Learn React

Edit App.js




localhost:3000
Sign in
Spring Download
LinkedIn Login, Sign...
Suriyanar Koil Templ...
Samsung Corporate...
Gmail
Recordings - OneDr...
How JSON Schema...
YouTube
Maps
Infinity Meta Jr
(22) Oracle JET UI c...
Add components to...
Printable Activity W...
Other favorites
Welcome to New Era Of Developing Applications Using React
Learn React

Alternative Approach for Using create-react-app

Create-react-app

npx

npx create-react-app <project_name>

npm package runner

npm

npm install create-react-app -g

create-react-app<project_name>

Using Vite

Step 1: Install a build tool

The first step is to install a build tool like `vite`, `parcel`, or `rsbuild`. These build tools provide features to package and run source code, provide a development server for local development and a build command to deploy your app to a production server.

Vite

Vite is a build tool that aims to provide a faster and leaner development experience for modern web projects.

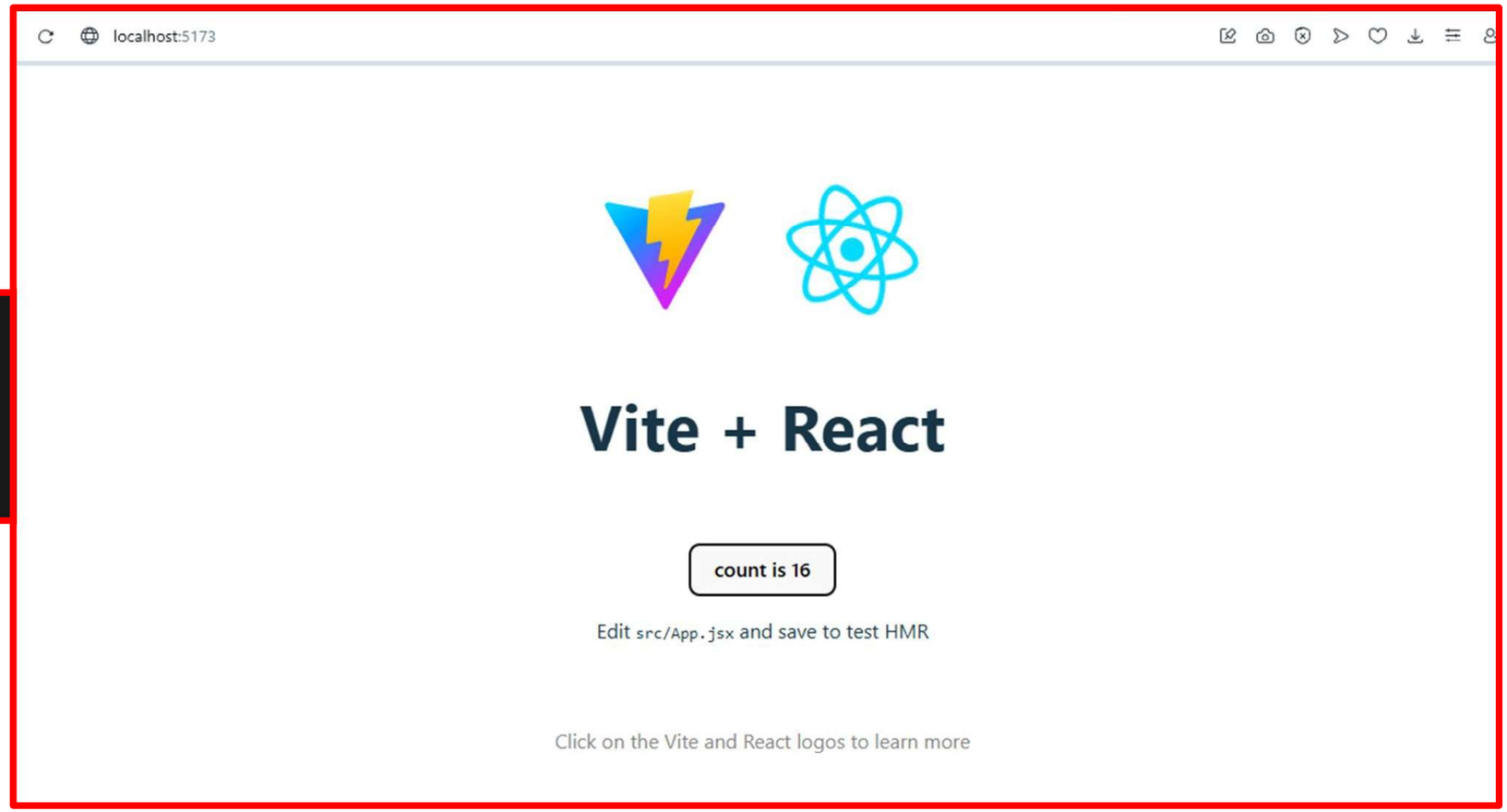
Terminal Copy

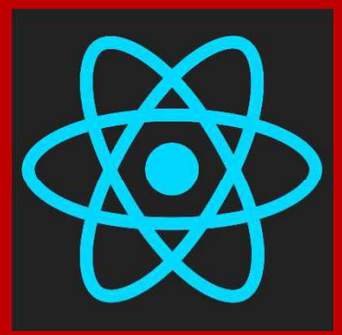
```
npm create vite@latest my-app -- --template react
```

Vite is opinionated and comes with sensible defaults out of the box. Vite has a rich ecosystem of plugins to support fast refresh, JSX, Babel/SWC, and other common features. See Vite's React plugin or React SWC plugin and React SSR example project to get started.

Running the React Application using Vite

```
cd my-app
npm install
npm run dev
```



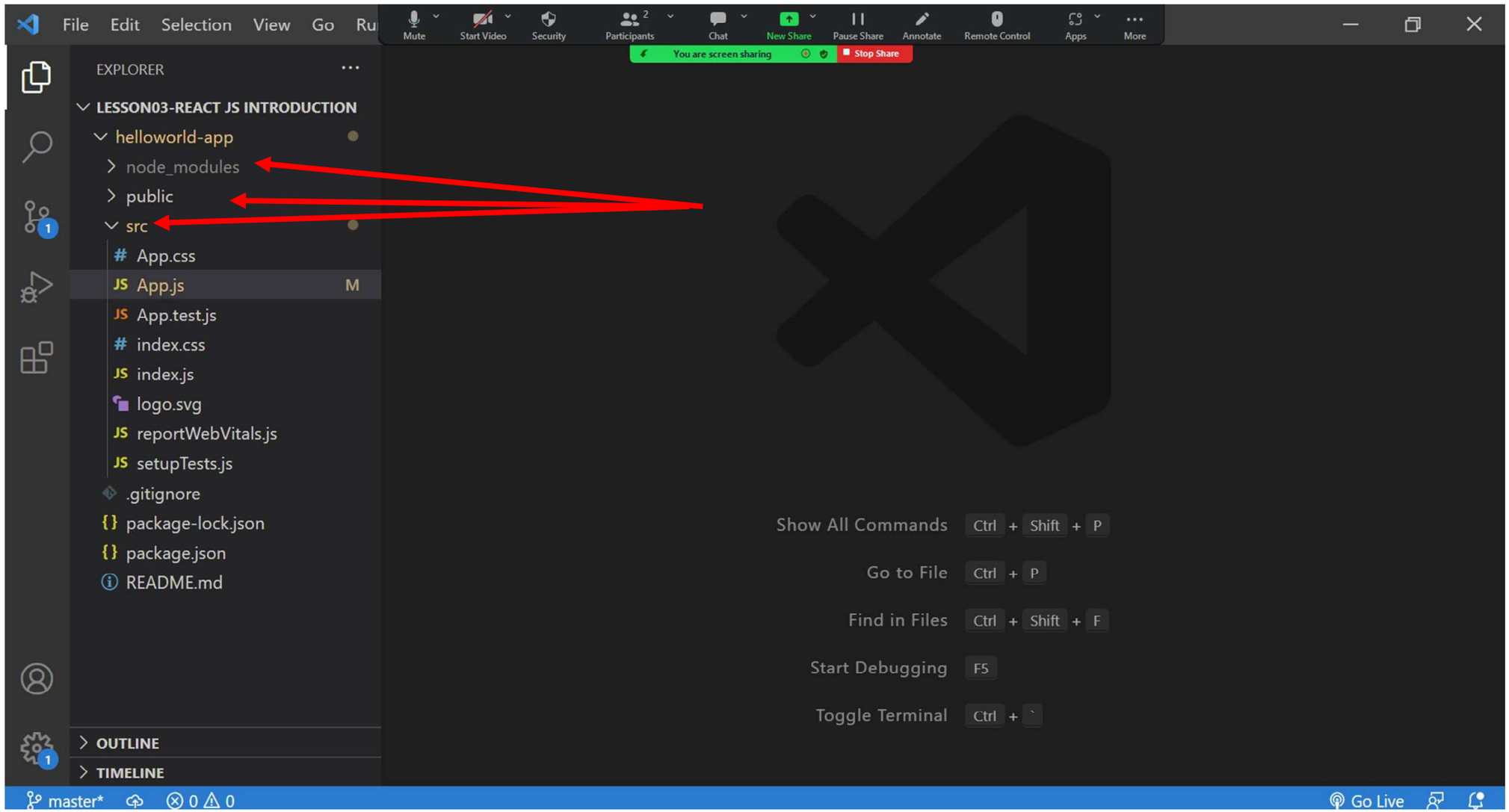


Code Analysis

Code Analysis

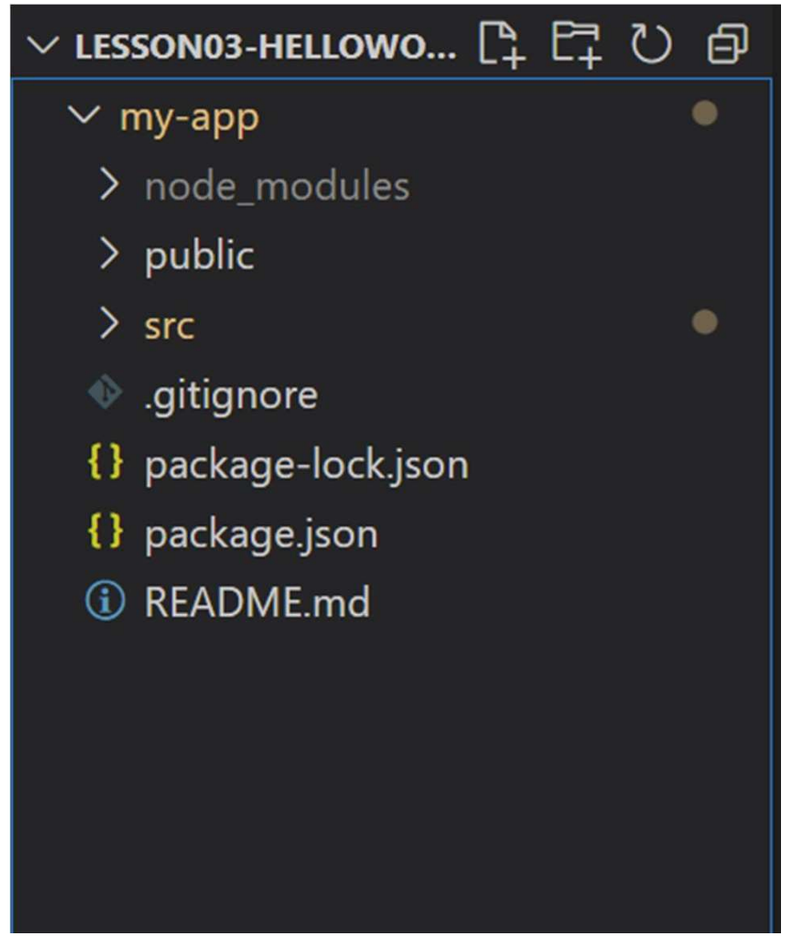
- So now that we have created and run a basic react application using create react app.
- Now at the same time it is also really important to understand the files and folders involved and how the control flows when you run the application

Project Structure [When React Project is Constructed]

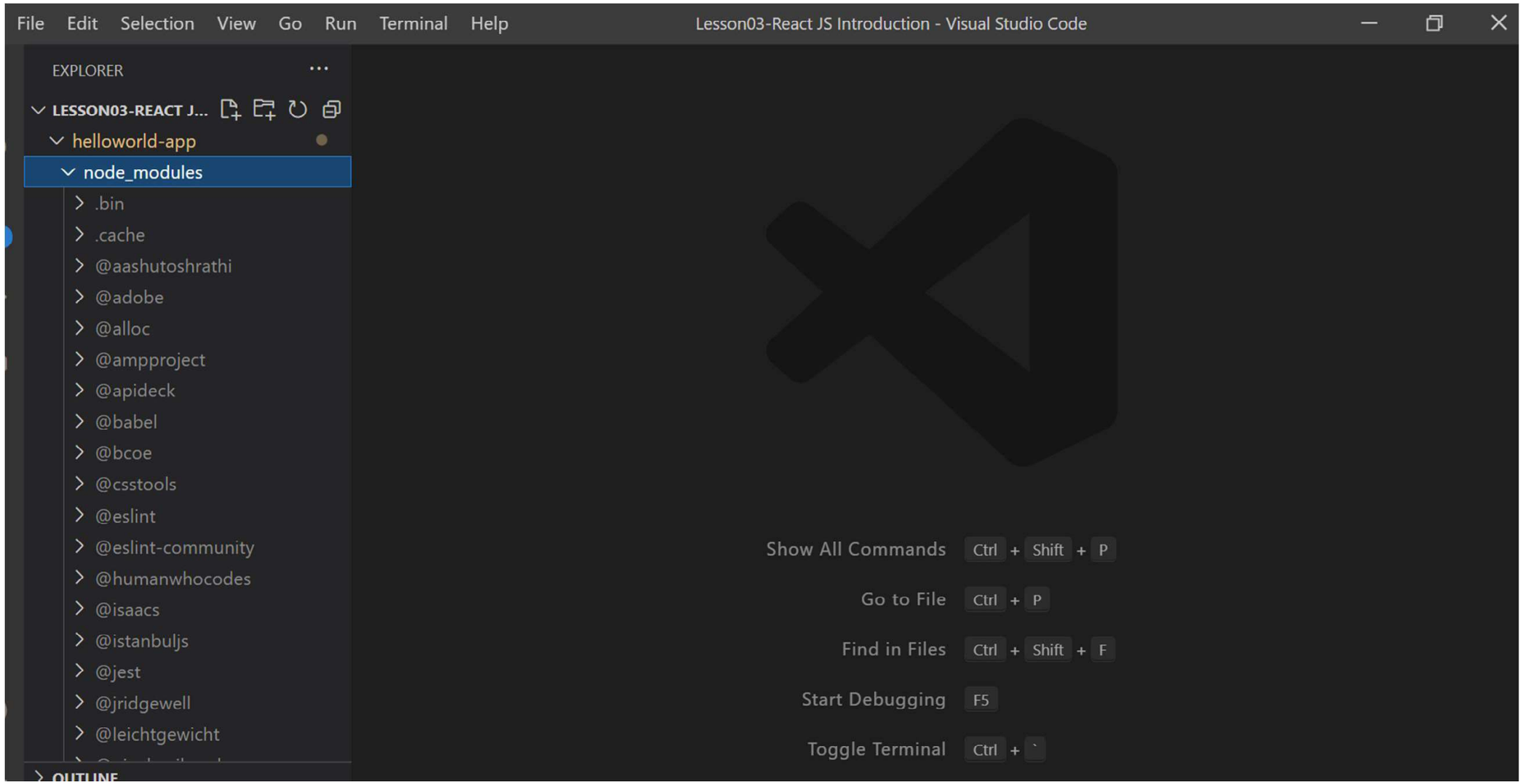


OJET App Folder Structure

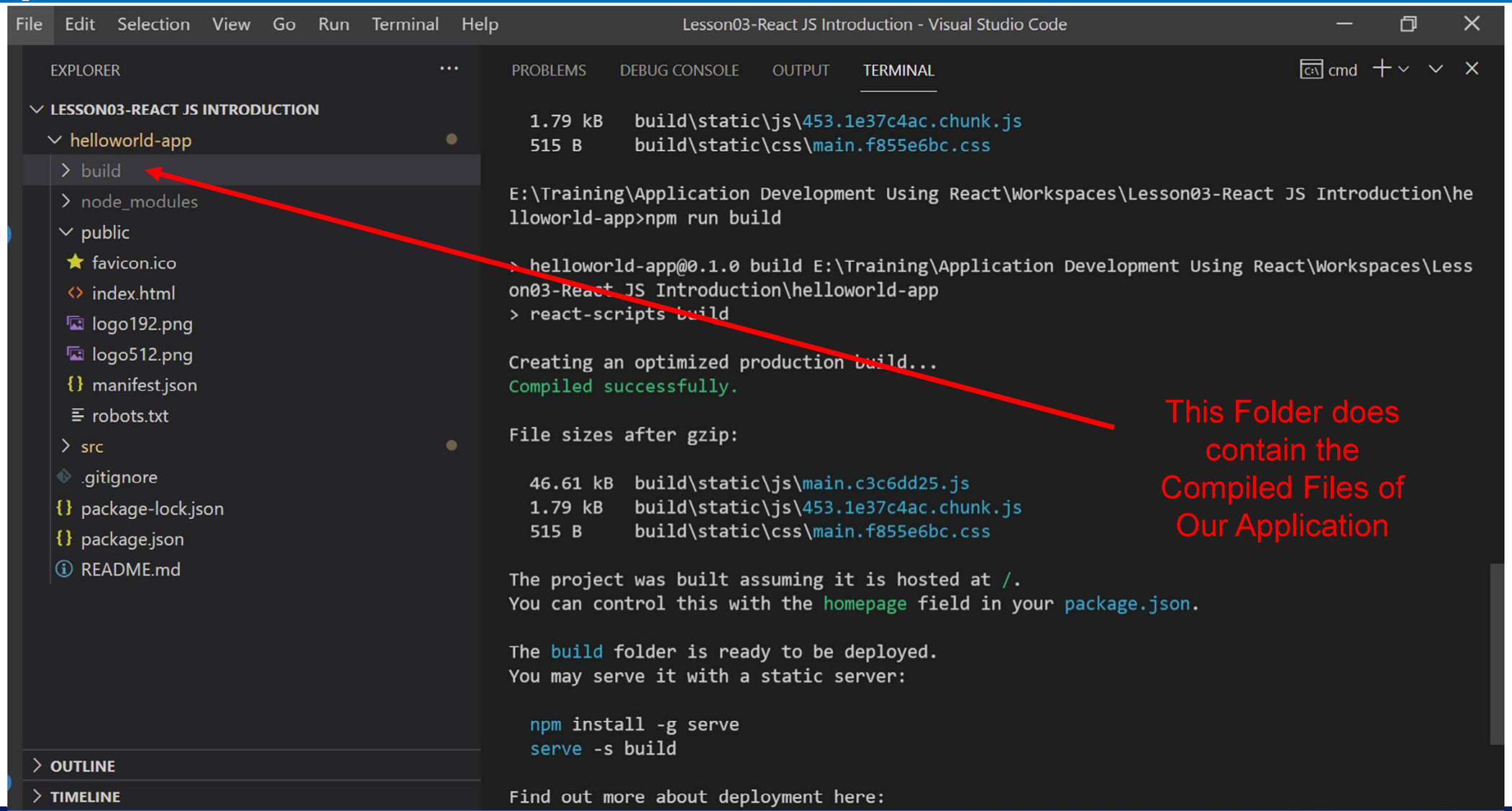
- **node_modules** − The npm package installed is node_modules. You can open the folder and see the packages available.
- **public** – This folder contains Fav Icons and index.html
- **src** − This folder is where we will work on the project using React



Node_modules [Libs]



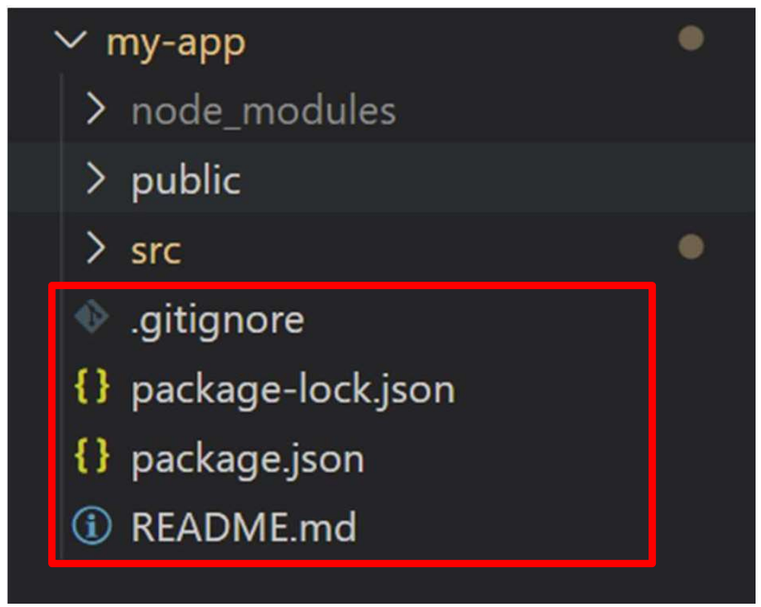
npm run build



Package.json

- This file contains the dependencies and the scripts required for the project you can see that we are using React version.
- Few scripts to run the application build the application or even run tests now based on whether you have just NPM or yarn as a package manager you are going to see yarn lock or package lock files they simply ensure consistent installation of your dependencies

- **package.json and package-lock.json --** Both the package.json and package-lock.json files contain a list of dependencies that must be installed for the application to run. Whenever you run an npm install against a project, it will use these two files to work out what dependencies to install, and it will fetch the libraries and place them all within the local node_modules folder.



package vs. package-lock

- You may have noticed that the package and package-lock files are very similar in nature.
- The truth is that they are indeed similar, but there is a history behind why that is. When working within a project, if you come to install a new library, for example, **trumbowyg** you can run the following command, and notice that trumbowyg will be added to the package.json file

```
npm install trumbowyg
```

Actual Reason of using package-lock.json

- Note that the version of trumbowyg within the package.json file has a caret at the beginning (^2.11.1). This essentially instructs NPM to install the latest version of the major release line (so, the latest 2.x.x release). This means that if someone else were to check out this project in the future, and rerun an npm install following a newer release (such as version 2.12.1), they would get a newer (and, therefore mismatched) version to the one originally installed.
- The package-lock.json file was introduced to overcome this inconsistency. The file contains a long list of the dependencies and the *specific* version that should be installed. It also includes extra information, such as the module location and a list of packages that it requires. If we take a look at my package-lock.json file after the install we see that it also includes the trumbowyg installation and the exact version that was installed initially.

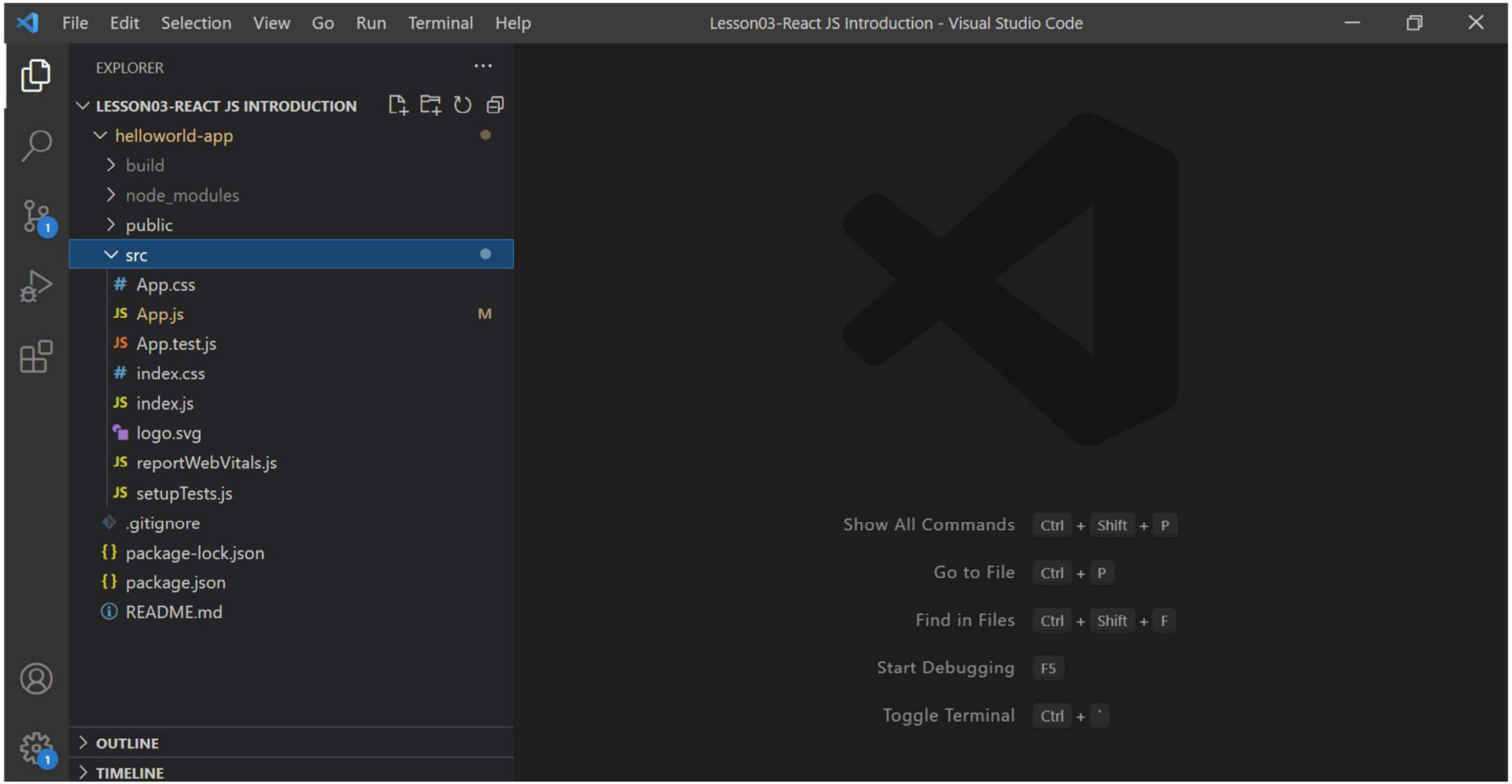
Index.html [Is the File Which gets Served Out]

- The index file will hold the initial HTML markup required to initialize the application.
- Some of this markup will persist through the lifetime of a session; however, most of it will be dynamically changed as a user interacts with the application.
- This is why most client-side applications are referred to as single-page applications (SPAs). The user remains on a single page (index.html), and the content of the page will be dynamically switched, compared to a traditional web site, wherein they would navigate between different pages (about-us.html, etc.).

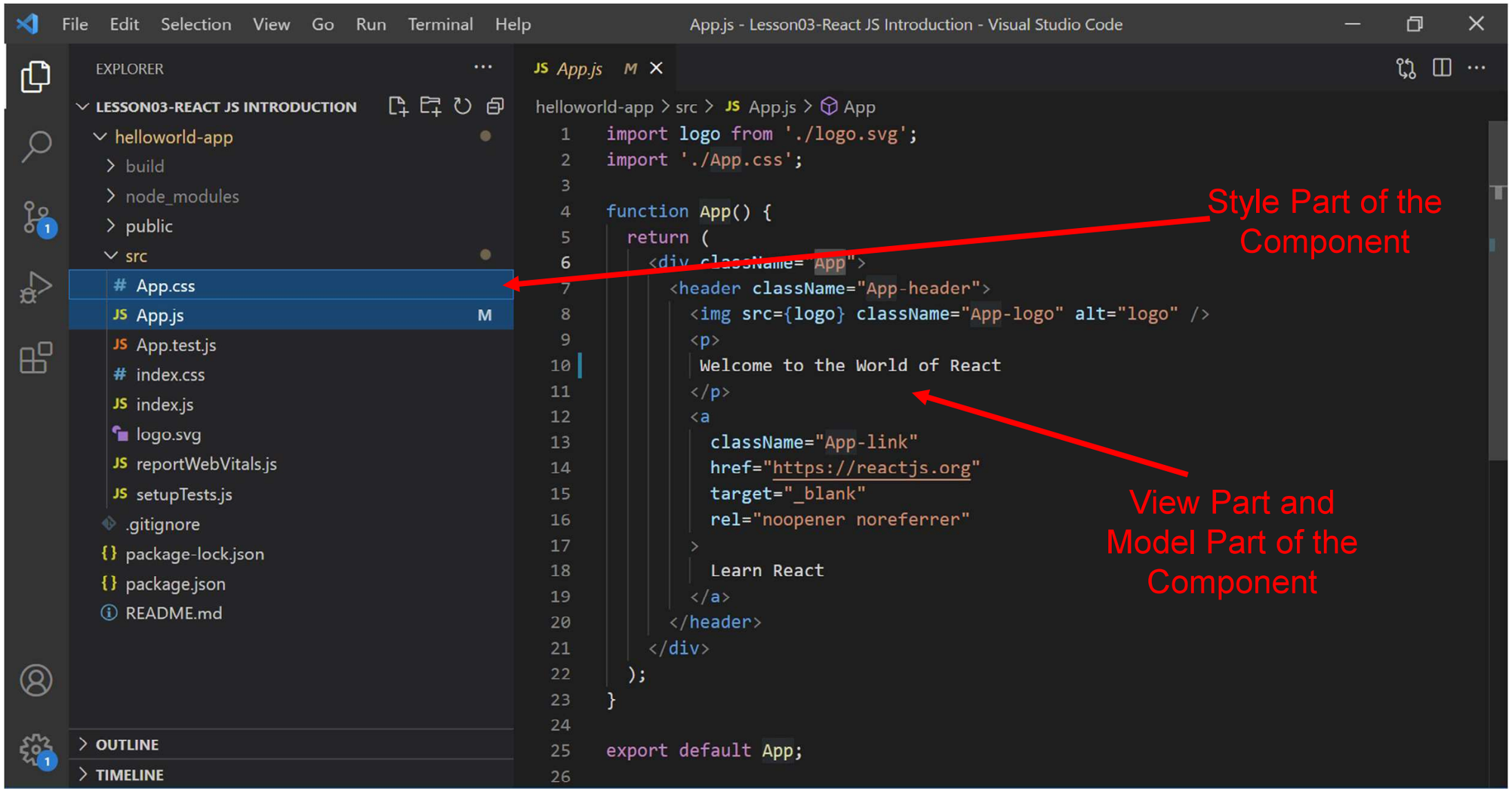
- It is this HTML file that gets served up typically you are not going to add any code in this file maybe some changes in the head tag but definitely not in the body tag you want react to control the UI and for that purpose we have one div tag with ID is equal to root at runtime the react application takes over this div tag and is ultimately responsible for the UI
- **Please make a note of this div tag**

```
    work correctly both with client-side routing and a non-root public URL.
    Learn how to configure a non-root public URL by running `npm run build`
  -->
  <title>React App</title>
</head>
<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.

    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn build`.
  -->
</body>
</html>
```

[Src – A Place Where Developer Writes the Code Functionality]

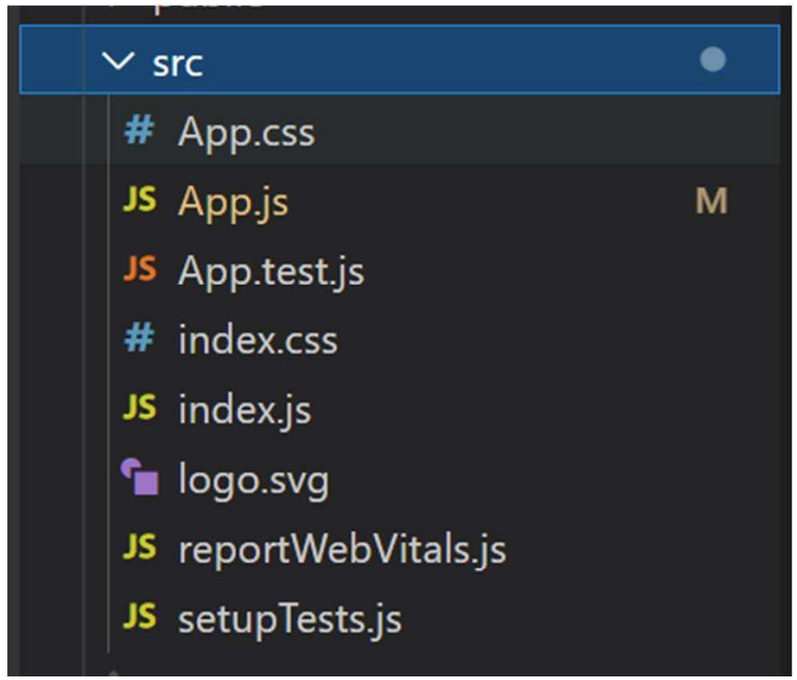


Root Component [App Component]

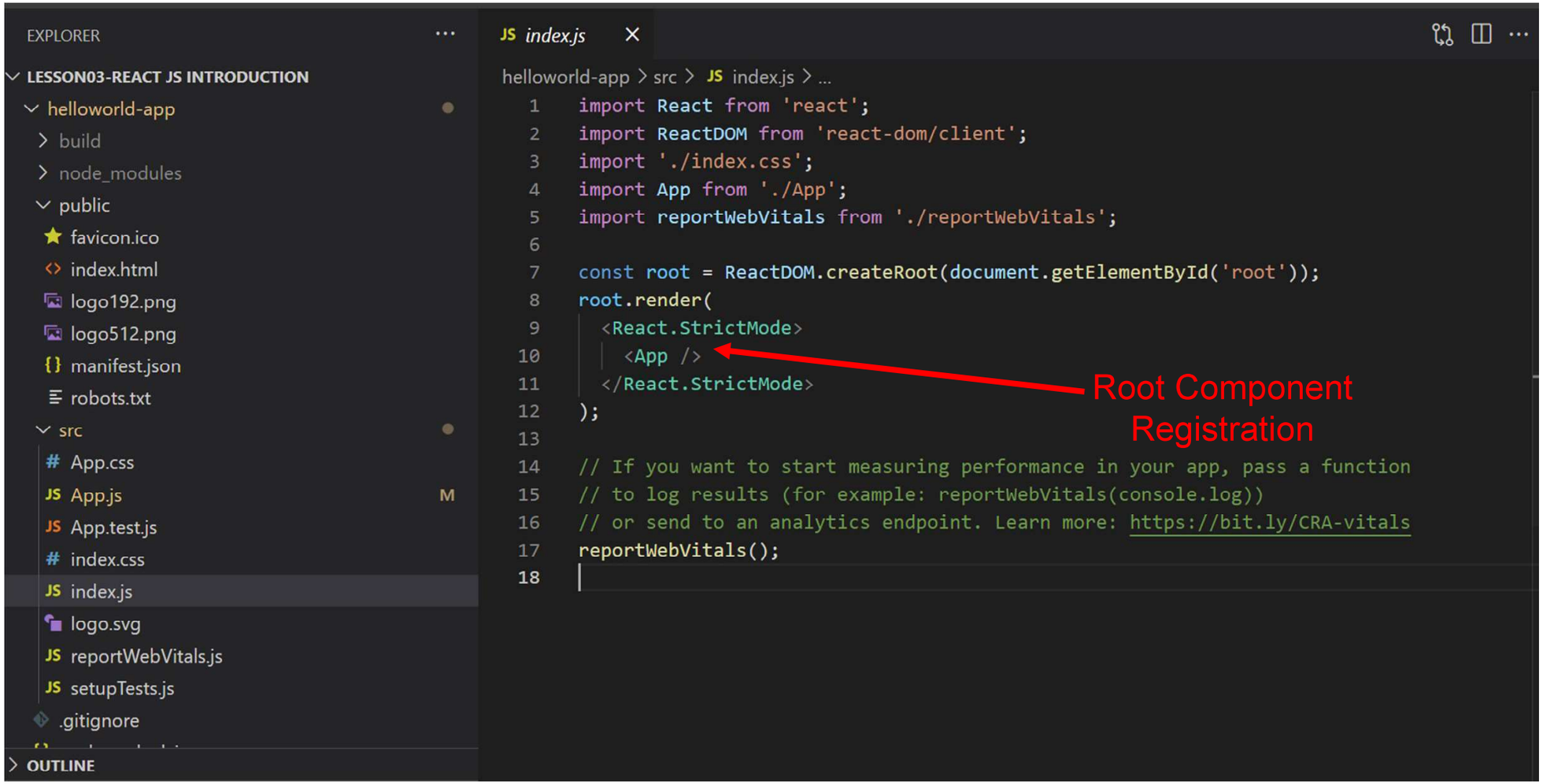


src Folder

- src folder which is the folder you will be working with the most during development

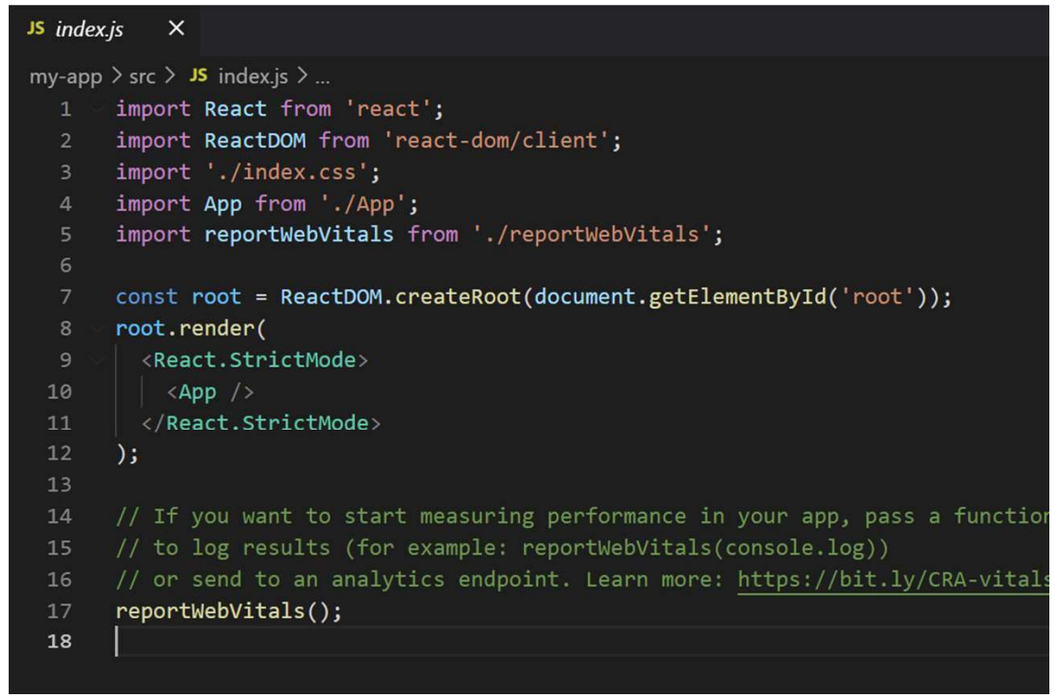


Registering the Root Component



Index.js

- The starting point for a react application **is index.js.**
- In index.js we specify the root component which is **<App> component** and the Dom element which will be controlled by react the Dom element in our example is an element with an ID of root and if you can recollect this is the element in our **index.html** file we call this div element as the root Dom node because everything inside it will be controlled by react for the hello world application



App.js

- The app component is rendered inside the root Dom node that brings us to the **<App>** component which is present in app.js
- Is responsible for the HTML displayed in the browser in other words the app component represents the view which we see in the browser

App.js M

my-app > src > App.js > App

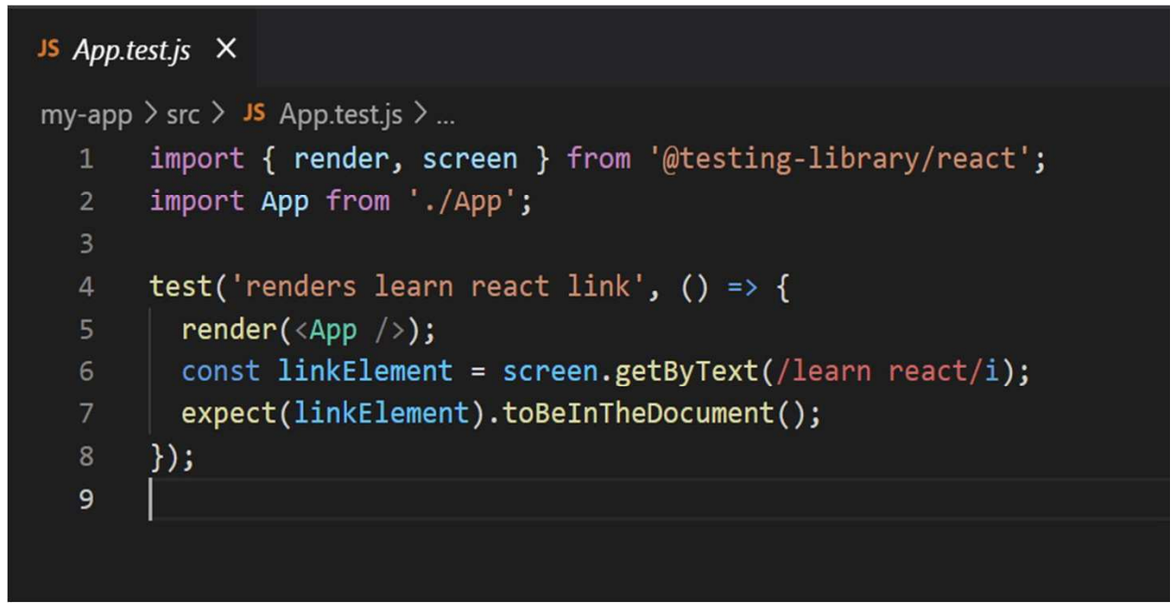
```
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <img src={logo} className="App-logo" alt="logo" />
9          <h2>
10           Welcome to New Era Of Developing Applications Using React
11         </h2>
12         <a
13           className="App-link"
14           href="https://reactjs.org"
15           target="_blank"
16           rel="noopener noreferrer"
17         >
18           Learn React
19         </a>
20       </header>
21     </div>
22   );
23 }
24
25 export default App;
```

App.css

- create react app also generates the CSS file for **styling CSS file** contains classes which are applied in the app component

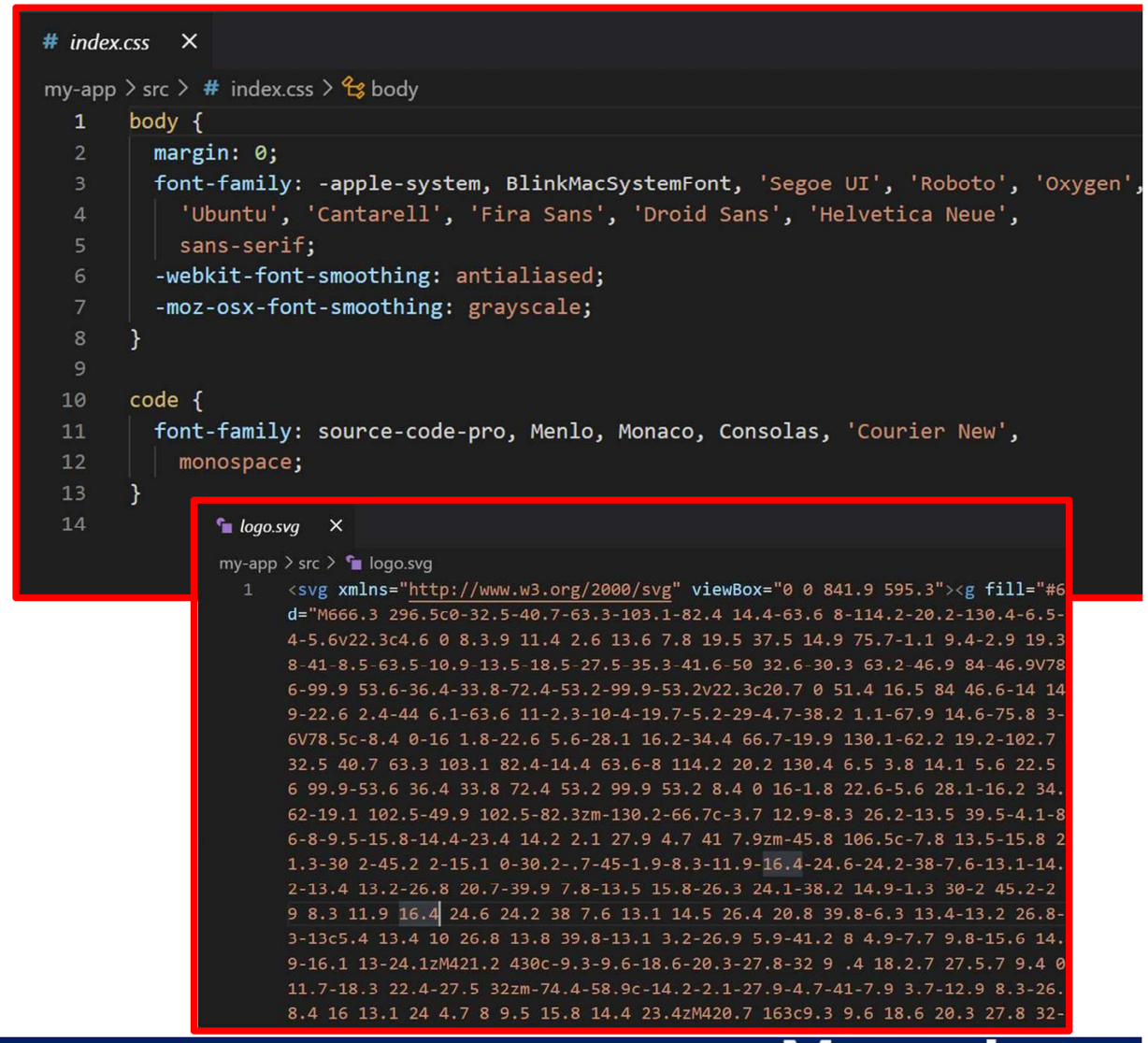
App.test.js

- The test file contains a simple test cases



Index.css

- An index dot CSS file which apply styles to the body tag and the logo SVG which is referenced in the app component



index.css

my-app > src > # index.css > body

```
1   body {
2     margin: 0;
3     font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
4       'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
5       sans-serif;
6     -webkit-font-smoothing: antialiased;
7     -moz-osx-font-smoothing: grayscale;
8   }
9
10  code {
11    font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
12      monospace;
13  }
14
```

logo.svg

my-app > src > logo.svg

```
1   <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 841.9 595.3"><g fill="#6
    d="M666.3 296.5c0-32.5-40.7-63.3-103.1-82.4 14.4-63.6 8-114.2-20.2-130.4-6.5-
    4-5.6v22.3c4.6 0 8.3.9 11.4 2.6 13.6 7.8 19.5 37.5 14.9 75.7-1.1 9.4-2.9 19.3
    8-41-8.5-63.5-10.9-13.5-18.5-27.5-35.3-41.6-50 32.6-30.3 63.2-46.9 84-46.9V78
    6-99.9 53.6-36.4-33.8-72.4-53.2-99.9-53.2v22.3c20.7 0 51.4 16.5 84 46.6-14 14
    9-22.6 2.4-44 6.1-63.6 11-2.3-10-4-19.7-5.2-29-4.7-38.2 1.1-67.9 14.6-75.8 3-
    6V78.5c-8.4 0-16 1.8-22.6 5.6-28.1 16.2-34.4 66.7-19.9 130.1-62.2 19.2-102.7
    32.5 40.7 63.3 103.1 82.4-14.4 63.6-8 114.2 20.2 130.4 6.5 3.8 14.1 5.6 22.5
    6 99.9-53.6 36.4 33.8 72.4 53.2 99.9 53.2 8.4 0 16-1.8 22.6-5.6 28.1-16.2 34.
    62-19.1 102.5-49.9 102.5-82.3zm-130.2-66.7c-3.7 12.9-8.3 26.2-13.5 39.5-4.1-8
    6-8-9.5-15.8-14.4-23.4 14.2 2.1 27.9 4.7 41 7.9zm-45.8 106.5c-7.8 13.5-15.8 2
    1.3-30 2-45.2 2-15.1 0-30.2-.7-45-1.9-8.3-11.9-16.4-24.6-24.2-38-7.6-13.1-14.
    2-13.4 13.2-26.8 20.7-39.9 7.8-13.5 15.8-26.3 24.1-38.2 14.9-1.3 30-2 45.2-2
    9 8.3 11.9 16.4 24.6 24.2 38 7.6 13.1 14.5 26.4 20.8 39.8-6.3 13.4-13.2 26.8-
    3-13c5.4 13.4 10 26.8 13.8 39.8-13.1 3.2-26.9 5.9-41.2 8 4.9-7.7 9.8-15.6 14.
    9-16.1 13-24.1zM421.2 430c-9.3-9.6-18.6-20.3-27.8-32 9 .4 18.2.7 27.5.7 9.4 0
    11.7-18.3 22.4-27.5 32zm-74.4-58.9c-14.2-2.1-27.9-4.7-41-7.9 3.7-12.9 8.3-26.
    8.4 16 13.1 24 4.7 8 9.5 15.8 14.4 23.4zM420.7 163c9.3 9.6 18.6 20.3 27.8 32-
```

- When we **run npm start**
  - Index.html file is served in the browser index.html contains the root **DOM** node
  - Next the control enters index.js [ReactDOM renders the <App> Component on to the root **DOM Node**
  - **<App> Component contains the HTML which is ultimately displayed in the Browser**

```
JS App.js M
my-app > src > JS App.js > App
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <img src={logo} className="App-logo" alt="logo" />
9          <h2>
10           Welcome to New Era Of Developing Applications Using React
11         </h2>
12         <a
13           className="App-link"
14           href="https://reactjs.org"
15           target="_blank"
16           rel="noopener noreferrer"
17         >
18           Learn React
19         </a>
20       </header>
21     </div>
22   );
23 }
24
25 export default App;
```