前端網站開發人員證書課程
(二) 進階網絡程式設計--專業React.js應用

2. Create a React App

Presented by Krystal Institute

Lesson Outline

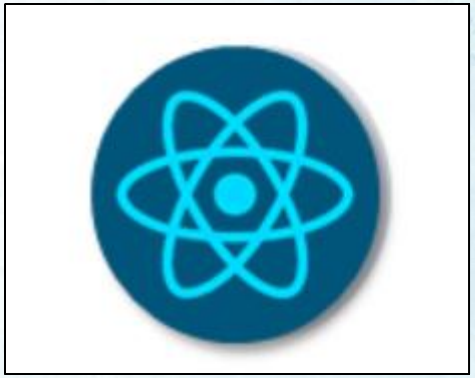
- The environmental requirements of creating React app
- The steps to install the required packages
- How to create and run the react app in your local environment

2.1 Environmental Requirements

2.1.1 Review

What is React?

- React is a JavaScript-based UI development library.
- Although React is a library rather than a language, it is widely used in web development.
- The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web development.
- React helps in:
  - Easy creation of dynamic applications.
  - Reusable components.
  - Unidirectional data flow.
  - It can be used for the development of both web and mobile apps(React Native).



2.1.2 Environmental Requirements

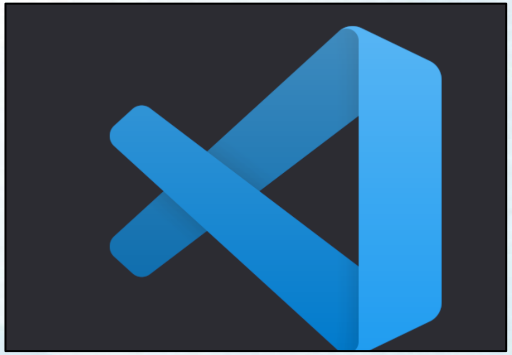
NodeJS and NPM

- NodeJS is the runtime environment needed for ReactJS development.
- After installing NodeJS, we can start installing React upon it using npm.
- React JS can be installed in two ways:
- Using webpack and babel.
- Using the create-react-app Command Line Tool.

2.1.2 Environmental Requirements

Visual studio code

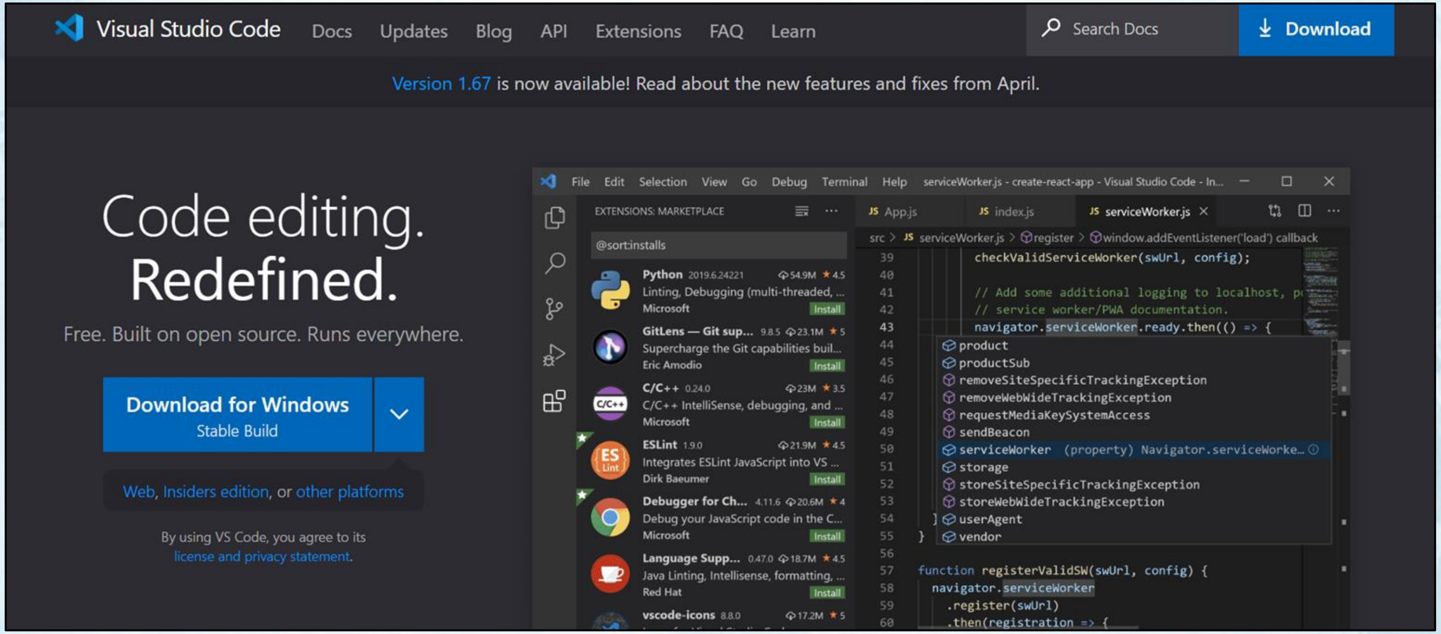
- Visual studio code is a source code editor available for Windows, macOS, and Linux.
- Visual studio code has built-in support for JavaScript, TypeScript, and Node.js.
- And Visual studio code can be extended for use of other languages(C, C++, C#, Java, Python, PHP) and runtimes(.NET, Unity).



2.1.2 Environmental Requirements

Install Visual Studio Code

- Visual studio code can be downloaded using the Official Link.



Version 1.67 is now available! Read about the new features and fixes from April.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

Web, Insiders edition, or other platforms

By using VS Code, you agree to its license and privacy statement.

serviceWorker.js - create-react-app - Visual Studio Code - In...

Python 2019.6.24221 54.9M ★4.5
Linting, Debugging (multi-threaded, ...
Microsoft

GitLens — Git sup... 9.8.5 23.1M ★5
Supercharge the Git capabilities buil...
Eric Amodio

C/C++ 0.24.0 23M ★3.5
C/C++ IntelliSense, debugging, and ...
Microsoft

ESLint 1.9.0 21.9M ★4.5
Integrates ESLint JavaScript into VS ...
Dirk Baeumer

Debugger for Ch... 4.11.6 20.6M ★4
Debug your JavaScript code in the C...
Microsoft

Language Supp... 0.47.0 18.7M ★4.5
Java Linting, Intellisense, formatting, ...
Red Hat

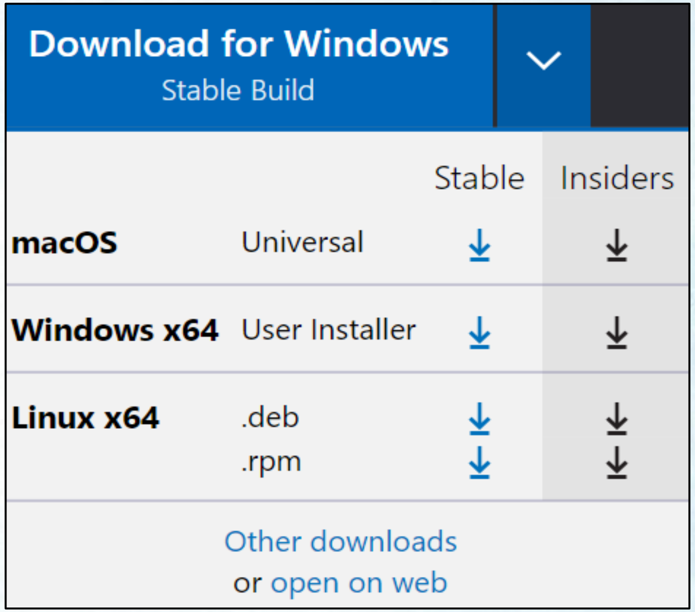
vscode-icons 8.8.0 17.2M ★5

```
      checkValidServiceWorker(swUrl, config);

      // Add some additional logging to localhost, p
      // service worker/PWA documentation.
      navigator.serviceWorker.ready.then(() => {
        product
        productSub
        removeSiteSpecificTrackingException
        removeWebWideTrackingException
        requestMediaKeySystemAccess
        sendBeacon
        serviceWorker  (property) Navigator.serviceWorke...
        storage
        storeSiteSpecificTrackingException
        storeWebWideTrackingException
      } userAgent
    }   vendor

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
```

2.1.2 Environmental Requirements

Install Visual Studio Code

- Depending on the OS VS code can be downloaded.
- Open the Visual studio code and install the required extensions.



Download for Windows
Stable Build

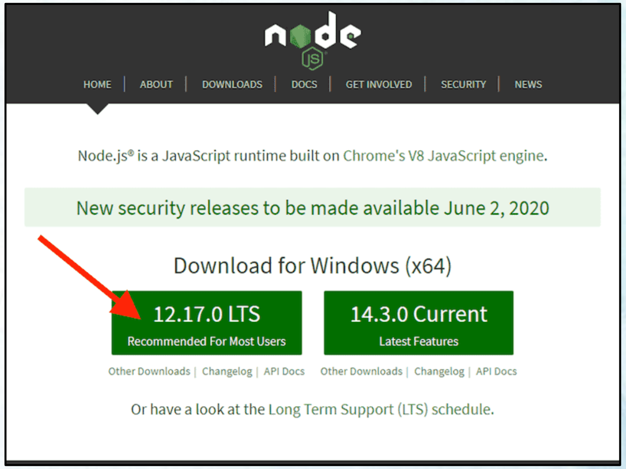
		Stable	Insiders
macOS	Universal	↓	↓
Windows x64	User Installer	↓	↓
Linux x64	.deb	↓	↓
	.rpm	↓	↓

Other downloads
or open on web

2.1.2 Environmental Requirements

NPM and NodeJS install

- Node.js and npm are the run-time and command-line tools required to build and run React applications.
- **Node.js** is a javascript runtime environment that enables you to run js code outside of a browser.
- **npm** is a package manager used to download javascript packages built to run on Node.
- **NPM** comes bundled together when you install Node.js.
- Download Node.js and npm from https://nodejs.org/en/


HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | NEWS
12.17.0 LTS
Recommended For Most Users
14.3.0 Current
Latest Features
Other Downloads | Changelog | API Docs
Other Downloads | Changelog | API Docs

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

New security releases to be made available June 2, 2020

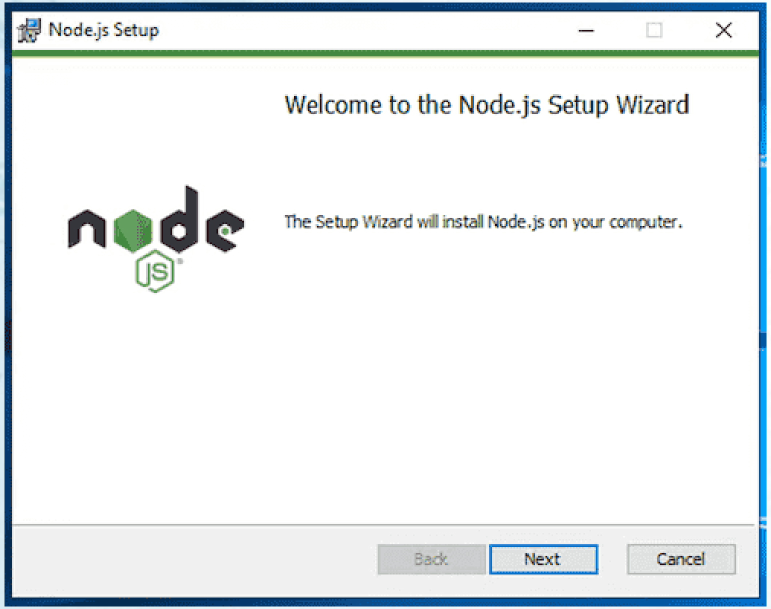
Download for Windows (x64)

Or have a look at the Long Term Support (LTS) schedule.

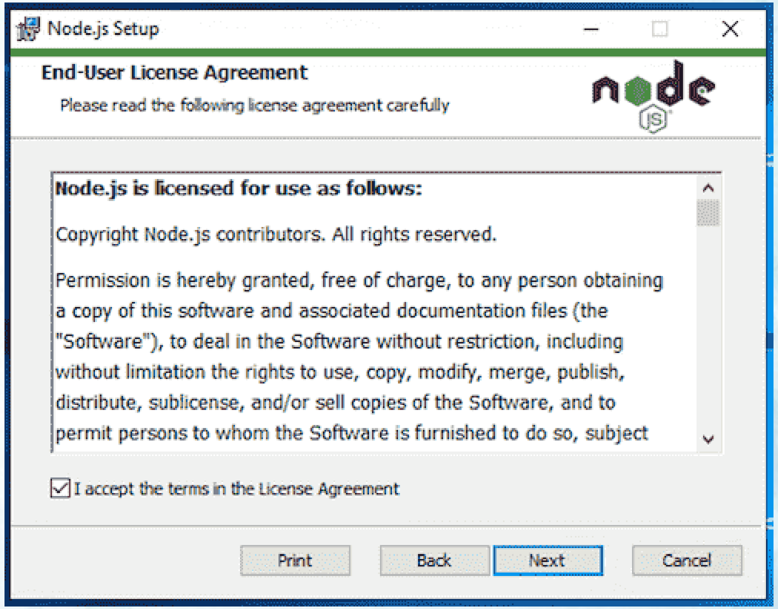
2.1.2 Environmental Requirements

NPM and NodeJS install

- Install Node.js and npm by opening the downloaded installer and following the prompts.


Node.js Setup
Welcome to the Node.js Setup Wizard
The Setup Wizard will install Node.js on your computer.
Back
Next
Cancel

1


Node.js Setup
End-User License Agreement
Please read the following license agreement carefully
I accept the terms in the License Agreement
Print
Back
Next
Cancel

Node.js is licensed for use as follows:

Copyright Node.js contributors. All rights reserved.

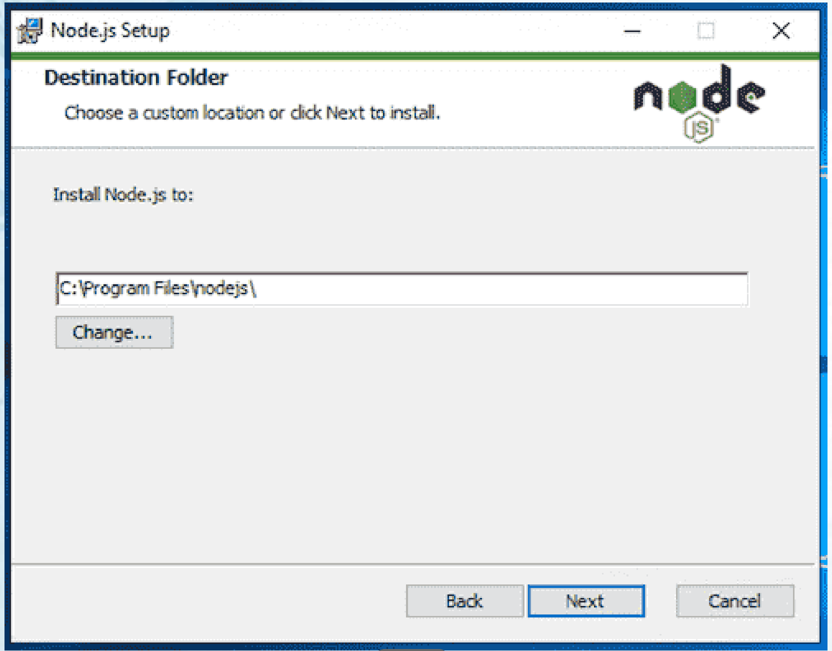
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject

2

2.1.2 Environmental Requirements

NPM and NodeJS install

- Install Node.js and npm by opening the downloaded installer and following the prompts.



Node.js Setup

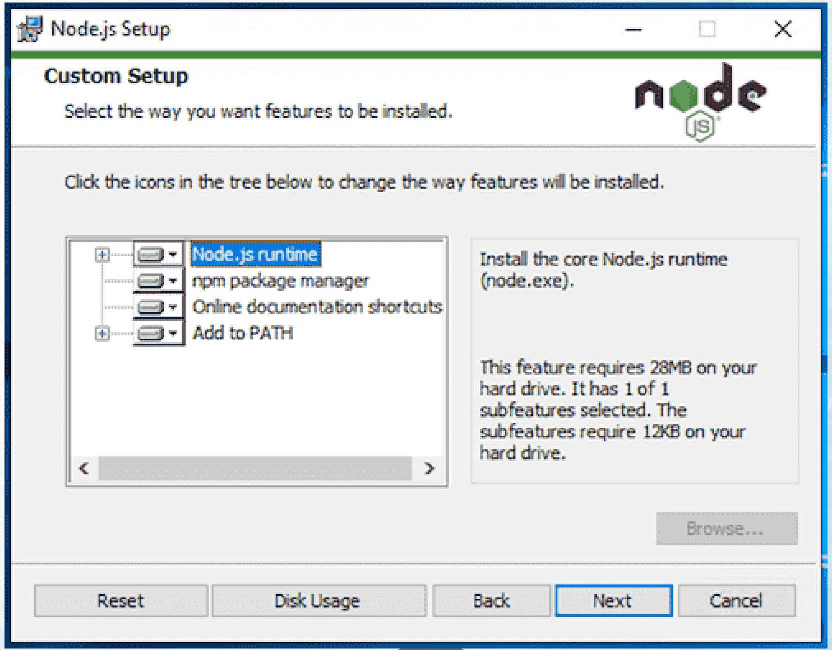
Destination Folder

Choose a custom location or click Next to install.

Install Node.js to:

C:\Program Files\nodejs\

3



Node.js Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- Node.js runtime
- npm package manager
- Online documentation shortcuts
- Add to PATH

Install the core Node.js runtime (node.exe).

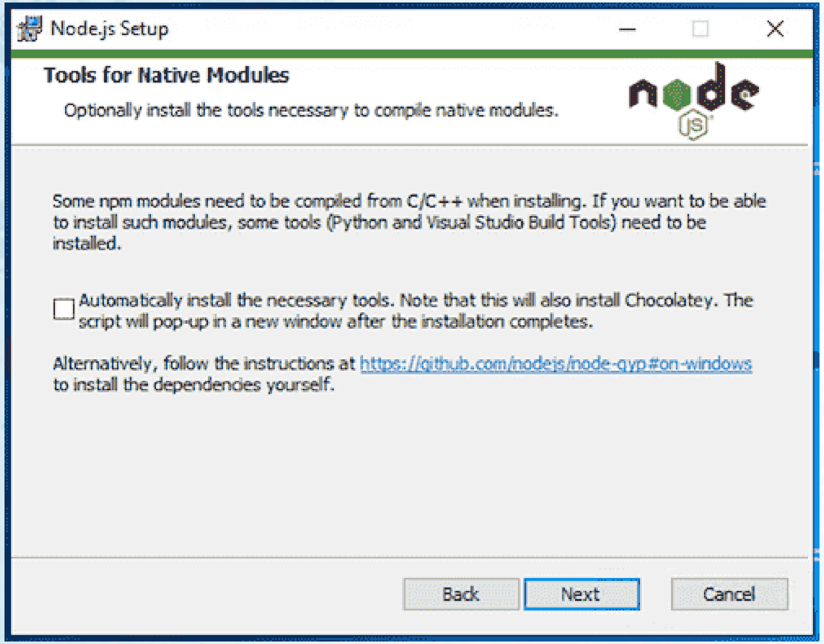
This feature requires 28MB on your hard drive. It has 1 of 1 subfeatures selected. The subfeatures require 12KB on your hard drive.

4

2.1.2 Environmental Requirements

NPM and NodeJS install

- Install Node.js and npm by opening the downloaded installer and following the prompts.



Node.js Setup

Tools for Native Modules

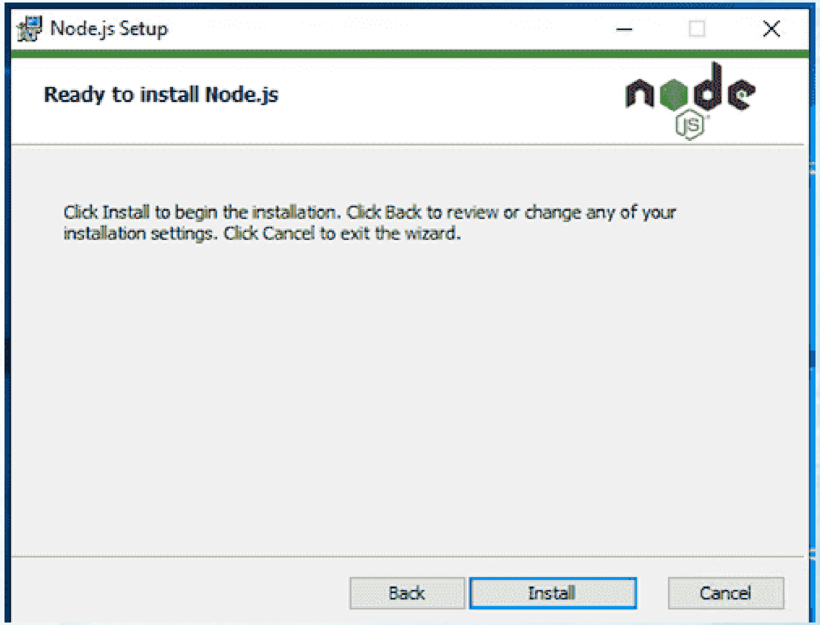
Optionally install the tools necessary to compile native modules.

Some npm modules need to be compiled from C/C++ when installing. If you want to be able to install such modules, some tools (Python and Visual Studio Build Tools) need to be installed.

☐ Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.

Alternatively, follow the instructions at https://github.com/nodejs/node-gyp#on-windows to install the dependencies yourself.

5



Node.js Setup

Ready to install Node.js

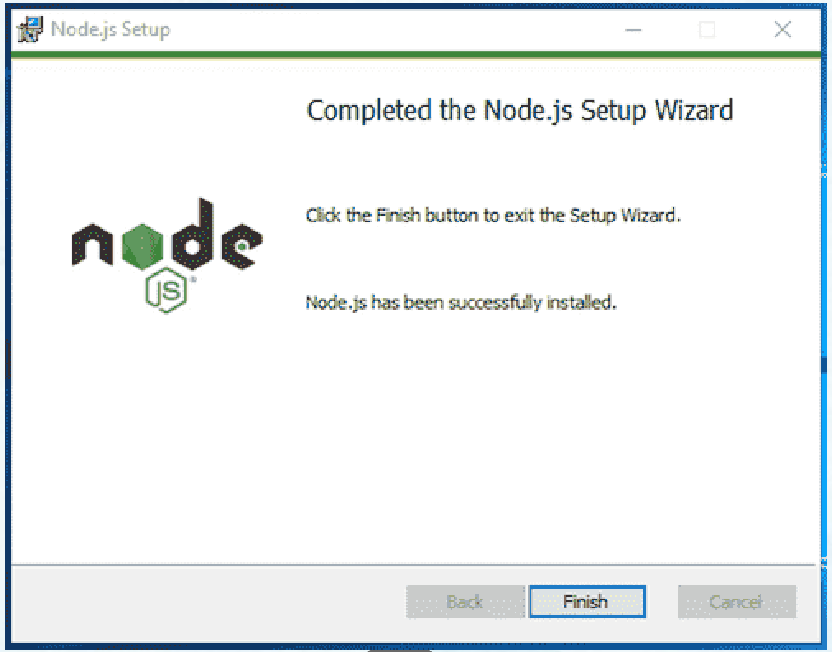
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

6

2.1.2 Environmental Requirements

NPM and NodeJS install

- Install Node.js and npm by opening the downloaded installer and following the prompts.



Completed the Node.js Setup Wizard

Click the Finish button to exit the Setup Wizard.

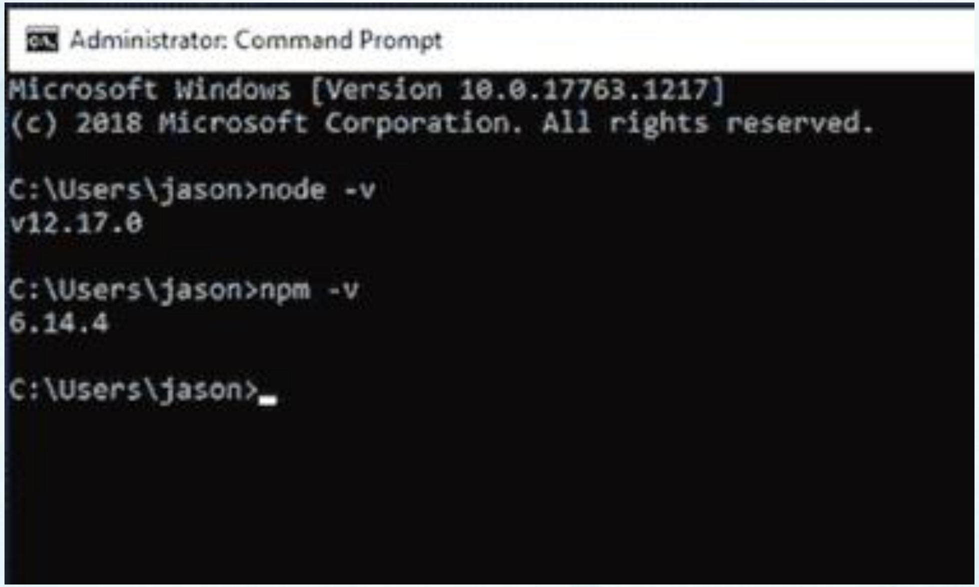
Node.js has been successfully installed.

7

2.1.2 Environmental Requirements

Test

- Test that Node.js and npm were installed successfully by running the commands node -v and npm -v.



```
Microsoft Windows [Version 10.0.17763.1217]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\jason>node -v
v12.17.0

C:\Users\jason>npm -v
6.14.4

C:\Users\jason>_
```

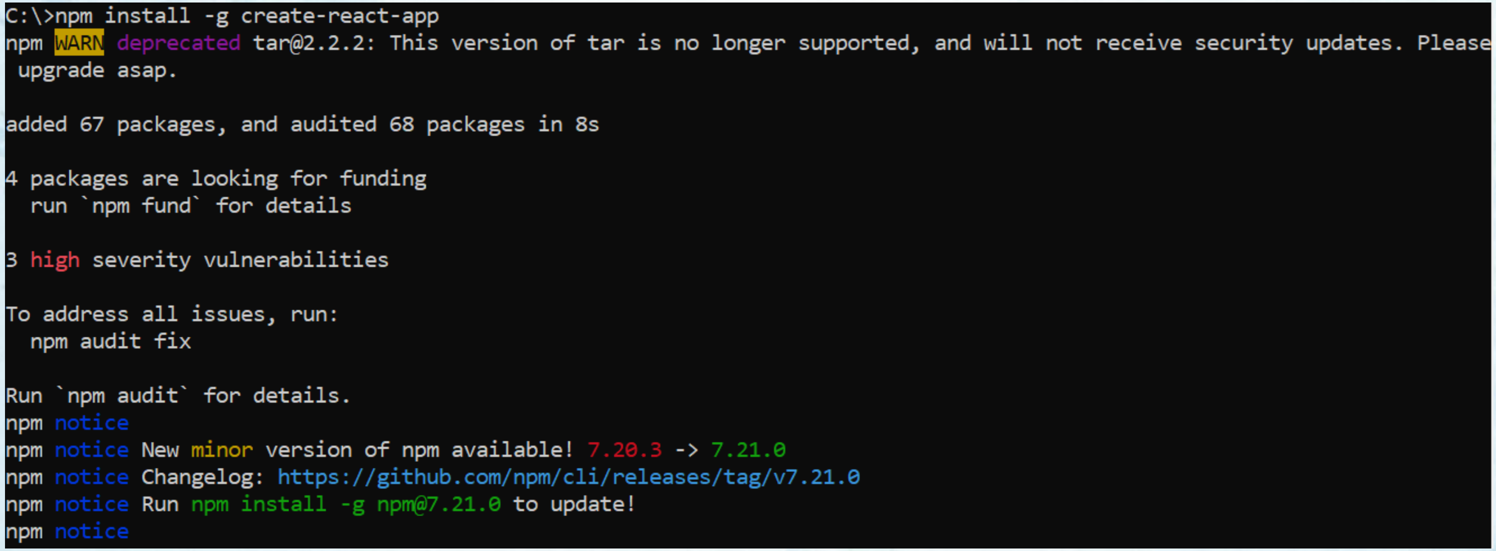
2.2 Create React App

2.2 Create React App

- create-react-app is a much easier way which does all the configuration and necessary package installations for us automatically and starts a new React app locally, ready for development.
- Once the React installation is successful, we can create a new React project using a **create-react-app** command.
- Run the **npm install -g create-react-app** in your terminal or command prompt to install the create-react-app Command Line Interface(CLI).

2.2 Create React App

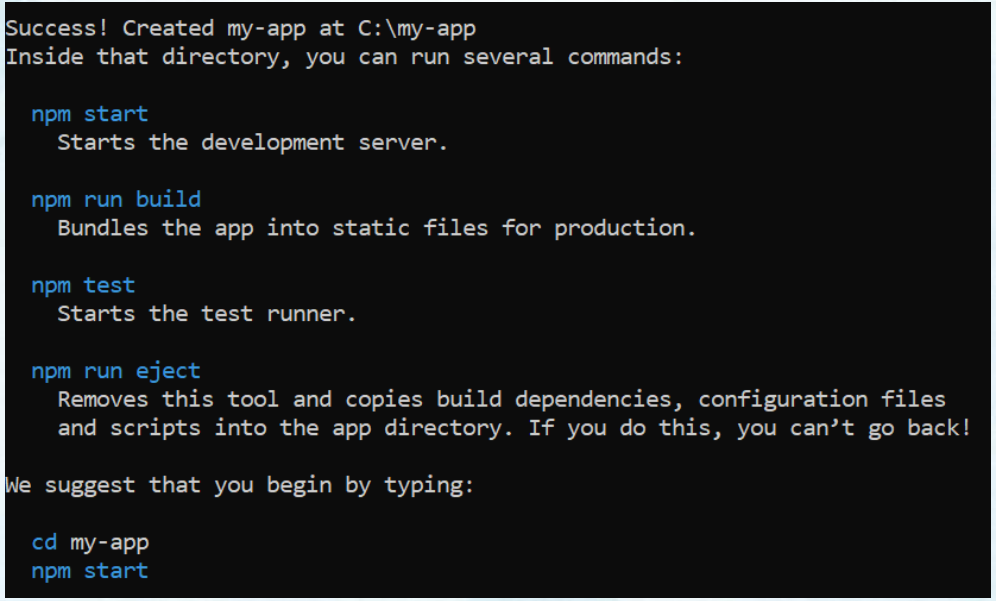
- After running the above command and successfully installing the create-react-app CLI, it will show output as shown in the below image.



```
C:\>npm install -g create-react-app
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please
 upgrade asap.

added 67 packages, and audited 68 packages in 8s

4 packages are looking for funding
  run `npm fund` for details

3 high severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 7.20.3 -> 7.21.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.21.0
npm notice Run npm install -g npm@7.21.0 to update!
npm notice
```

2.2 Create React App

- Run the npx create-react-app my-app to create a new project in the current directory.
- The above command will create the app name my-app as shown in the image below:



```
Success! Created my-app at C:\my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  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 my-app
  npm start
```

2.3 React App Structure

2.3.1 React App Structure

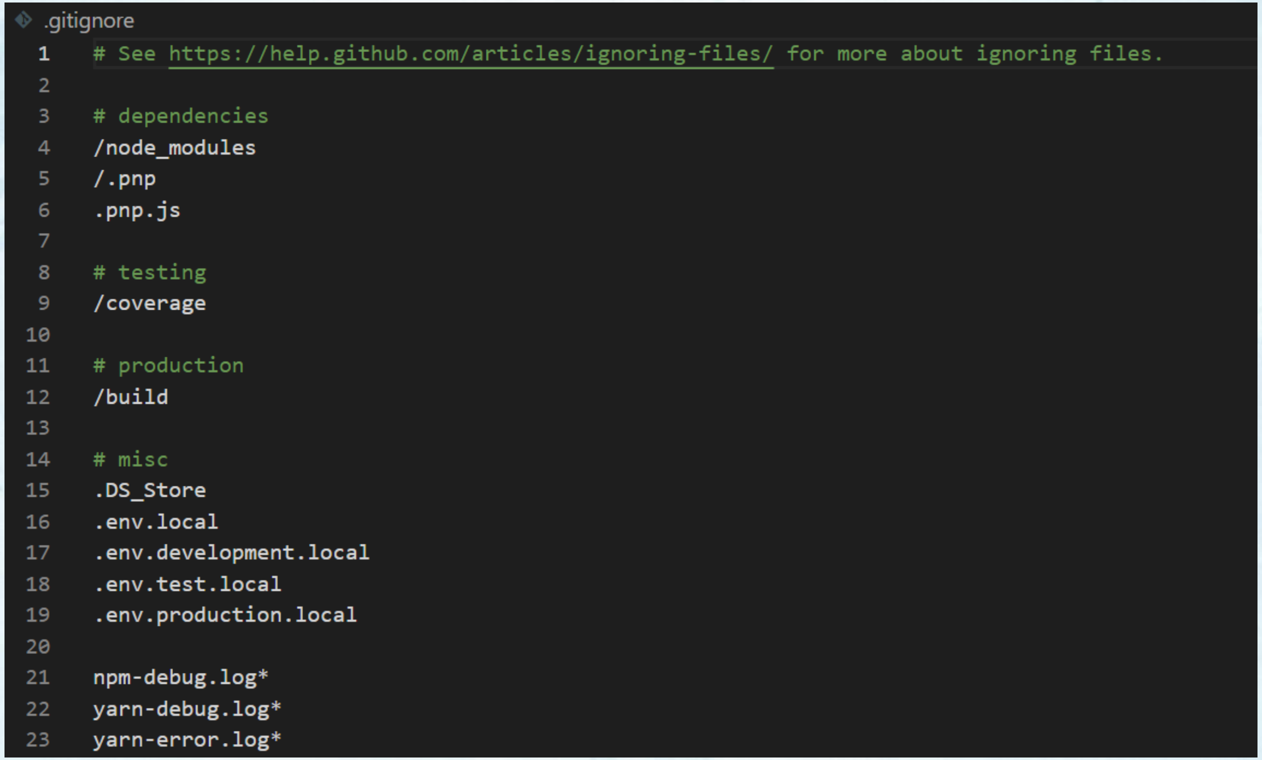
- Open the app created in the text editor, you will see the following file structure:
- Most of what you see will not be visible to the visitor of your web app.
- React uses a tool called webpack which transforms the directories and files here into static assets. Visitors to your site are served those static assets.

```
myfirstreactapp
├── 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
├── .gititgnore
├── package.json
├── package-lock.json
└── README.md
```

2.3.2 .gitignore

- This is the standard file used by the source control tool git to determine which files and directories to ignore when committing code.
- While this file exists, create-react-app did not create a git repo within this folder.
- If you take a look at the file, it has taken care of ignoring a number of items (even **.DS_Store** for Mac users)

2.3.2 .gitignore

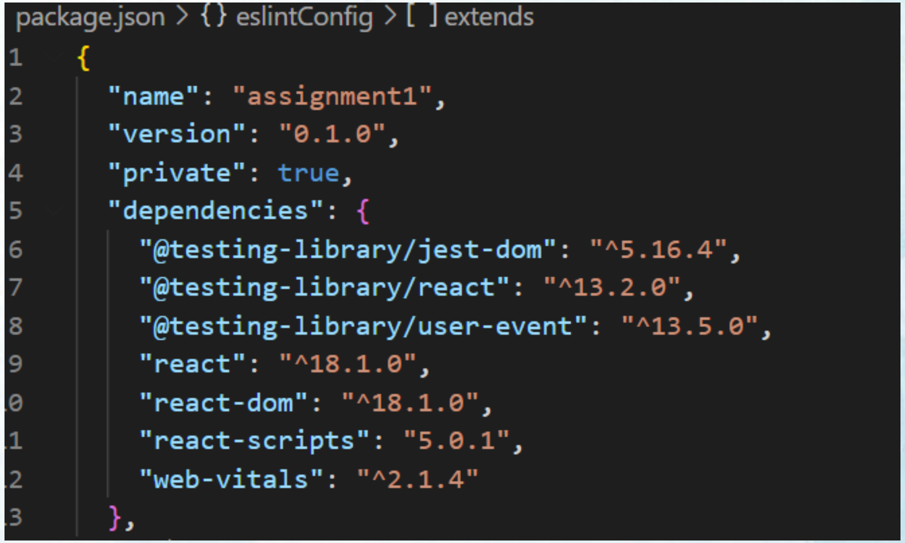


.gitignore

```
# See https://help.github.com/articles/ignoring-files/ for more about ignoring files.

# dependencies
/node_modules
/.pnp
.pnp.js

# testing
/coverage

# production
/build

# misc
.DS_Store
.env.local
.env.development.local
.env.test.local
.env.production.local

npm-debug.log*
yarn-debug.log*
yarn-error.log*
```

2.3.3 Package JSON

- **name:** The name of your app
- **version:** The current version
- **"private":** true is a failsafe setting to avoid accidentally publishing your app as a public package within the npm ecosystem.



package.json > {} eslintConfig > [ ] extends

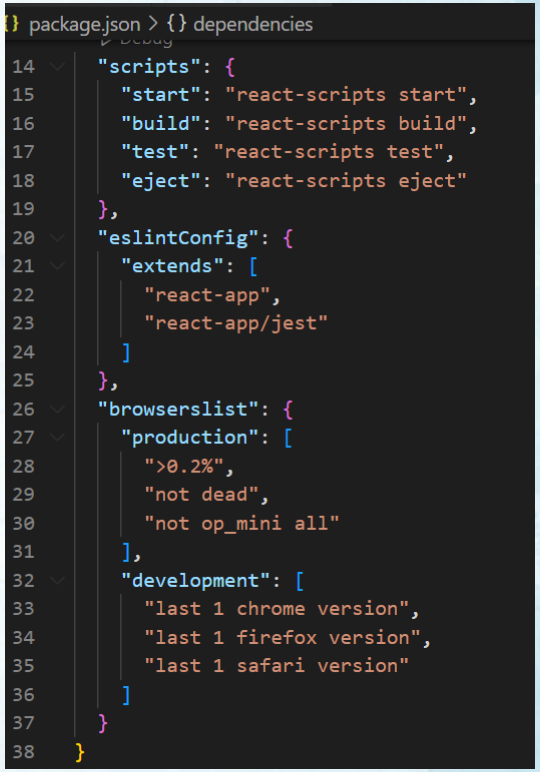
```
{
  "name": "assignment1",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.2.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.1.0",
    "react-dom": "^18.1.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
```

2.3.3 Package JSON

- dependencies:
    - contain all the required Node modules and versions required for the application.
    - In the given image, you'll see six dependencies.
        - The first three, as you may have guessed, are for the purpose of testing.
        - The next two dependencies allow us to use react and react-dom in our JavaScript.
        - Finally, react-scripts provides a useful set of development scripts for working with React.
    - In the screenshot above, the react version specified is ^18.1.0.
    - This means that npm will install the most recent major version matching 18.x.x.
    - In contrast, you may also see something like ~1.2.3 in package.json, which will only install the most recent minor version matching 1.2.x.

2.3.3 Package JSON

- scripts:
    - Specifies aliases that you can use to access some of the react-scripts commands in a more efficient manner.
    - For example, running the **npm test** in your command line will run **react-scripts test --env=jsdom** behind the scenes.
- You will also see two more attributes: eslintConfig and browserslist.
- Both of these are Node modules having their own set of values.
- **browserslist:** Provides information about browser compatibility of the app.
- **eslintConfig:** Takes care of the code linting.

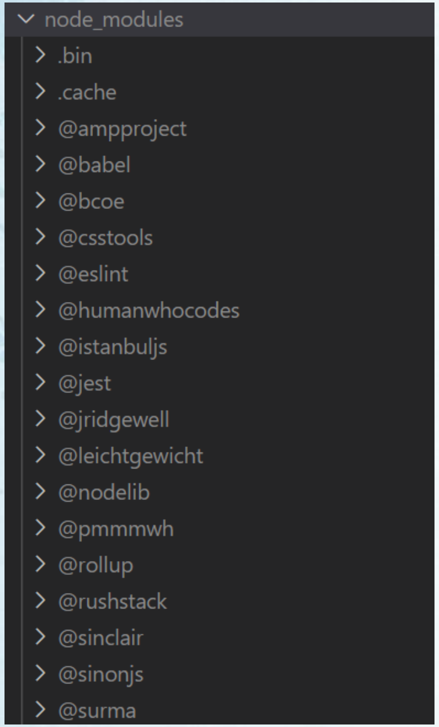

{} package.json > {} dependencies

```
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

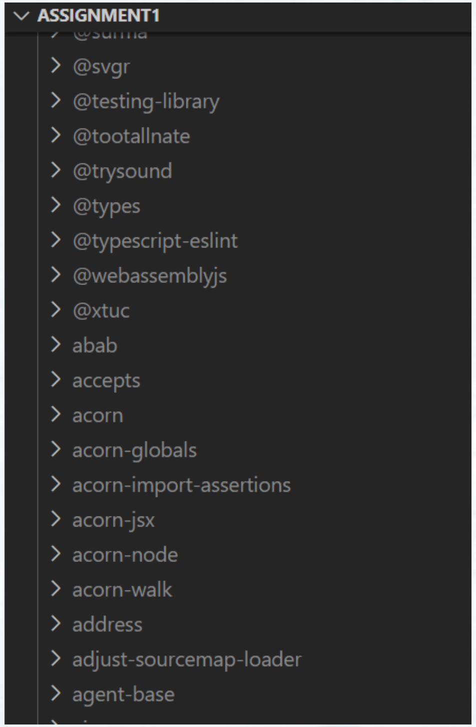
2.3.4 node_modules

- It contains the React library and any other third-party libraries needed.
- This directory contains dependencies and sub-dependencies of packages used by the current React app, as specified by package.json.
- If you take a look, you may be surprised by how many there are.
- Running ls -1 | wc -l within the node_modules/ directory will yield more than 800 subfolders.
- This folder is automatically added to the .gitignore.

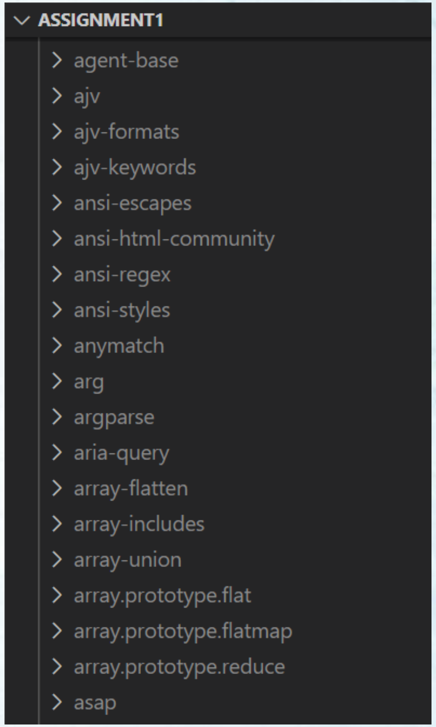
2.3.4 node_modules



```
∨ node_modules
  > .bin
  > .cache
  > @ampproject
  > @babel
  > @bcoe
  > @csstools
  > @eslint
  > @humanwhocodes
  > @istanbuljs
  > @jest
  > @jridgewell
  > @leichtgewicht
  > @nodelib
  > @pmmmwh
  > @rollup
  > @rushstack
  > @sinclair
  > @sinonjs
  > @surma
```



```
∨ ASSIGNMENT1
  > @svgr
  > @testing-library
  > @tootallnate
  > @trysound
  > @types
  > @typescript-eslint
  > @webassemblyjs
  > @xtuc
  > abab
  > accepts
  > acorn
  > acorn-globals
  > acorn-import-assertions
  > acorn-jsx
  > acorn-node
  > acorn-walk
  > address
  > adjust-sourcemap-loader
  > agent-base
```

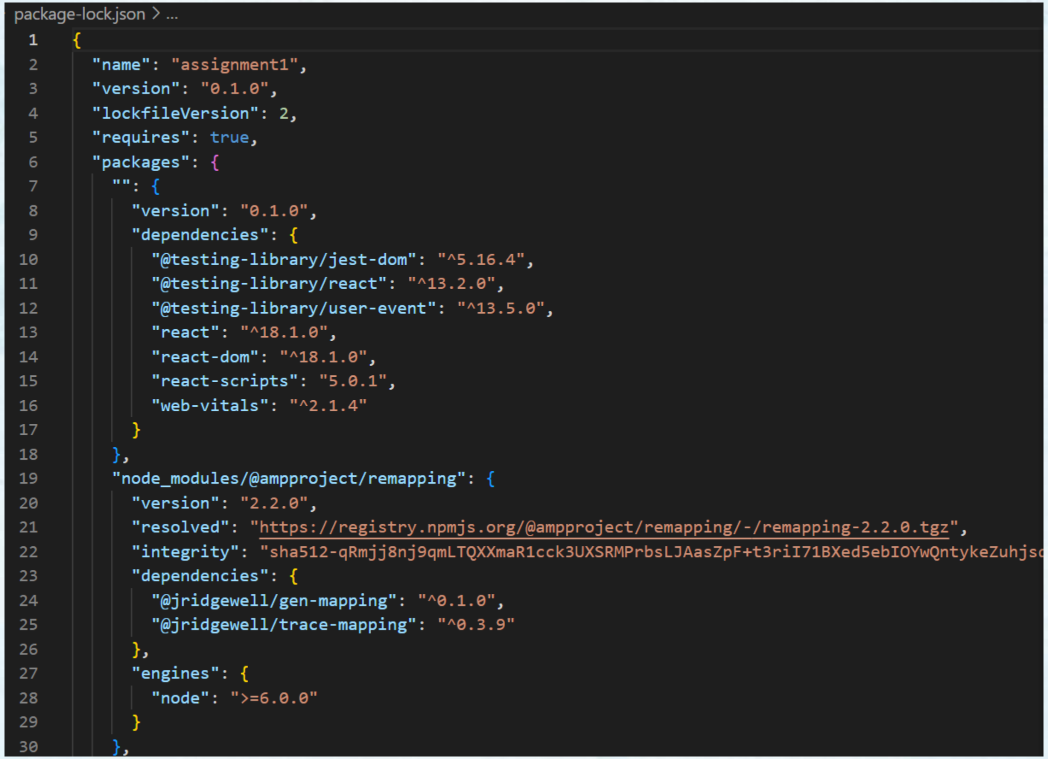


```
∨ ASSIGNMENT1
  > agent-base
  > ajv
  > ajv-formats
  > ajv-keywords
  > ansi-escapes
  > ansi-html-community
  > ansi-regex
  > ansi-styles
  > anymatch
  > arg
  > argparse
  > aria-query
  > array-flatten
  > array-includes
  > array-union
  > array.prototype.flat
  > array.prototype.flatmap
  > array.prototype.reduce
  > asap
```

2.3.5 package-lock.json

- This file contains the exact dependency tree installed in node_modules/.
- This provides a way for teams working on private apps to ensure that they have the same version of dependencies and sub-dependencies.
- It also contains a history of changes to package.json, so you can quickly look back at dependency changes.

2.3.5 package-lock.json

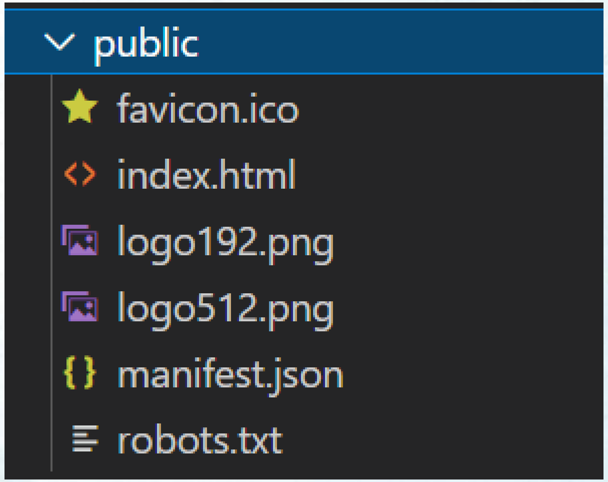


package-lock.json > ...

```
{
  "name": "assignment1",
  "version": "0.1.0",
  "lockfileVersion": 2,
  "requires": true,
  "packages": {
    "": {
      "version": "0.1.0",
      "dependencies": {
        "@testing-library/jest-dom": "^5.16.4",
        "@testing-library/react": "^13.2.0",
        "@testing-library/user-event": "^13.5.0",
        "react": "^18.1.0",
        "react-dom": "^18.1.0",
        "react-scripts": "5.0.1",
        "web-vitals": "^2.1.4"
      }
    },
    "node_modules/@ampproject/remapping": {
      "version": "2.2.0",
      "resolved": "https://registry.npmjs.org/@ampproject/remapping/-/remapping-2.2.0.tgz",
      "integrity": "sha512-qRmjj8nj9qmLTQXXmaR1cck3UXSRMPrbsLJAasZpF+t3riI71BXed5ebIOYwQntykeZuhjs
      "dependencies": {
        "@jridgewell/gen-mapping": "^0.1.0",
        "@jridgewell/trace-mapping": "^0.3.9"
      },
      "engines": {
        "node": ">=6.0.0"
      }
    },
```

2.3.6 public

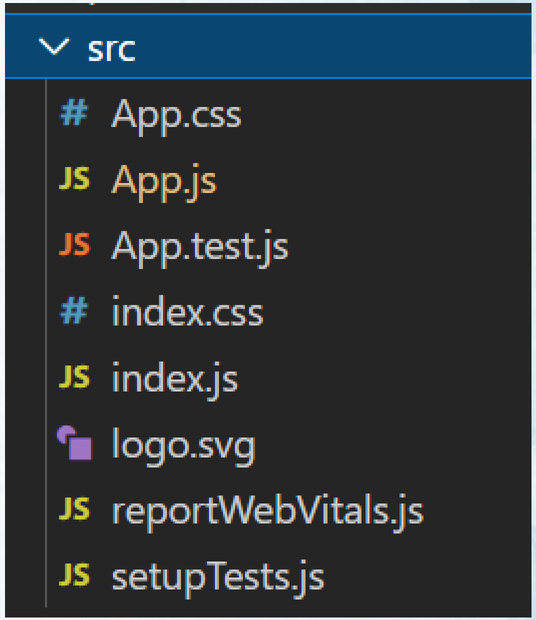
- This directory contains assets that will be served directly without additional processing by webpack.
- index.html provides the entry point for the web app. You will also see a favicon (header icon) and a manifest.json.



```
public
  favicon.ico
  index.html
  logo192.png
  logo512.png
  manifest.json
  robots.txt
```

2.3.7 src

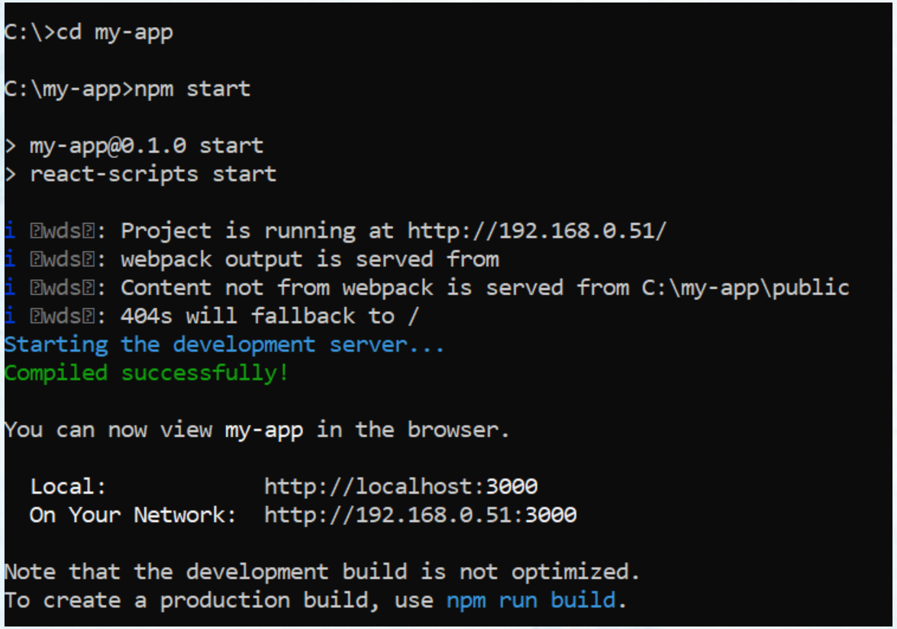
- This contains the JavaScript that will be processed by webpack and is the heart of the React app.
- Browsing this folder, you see the main App JavaScript component (App.js), its associated styles (App.css), and the test suite (App.test.js). index.js and its styles (index.css) provides an entry point to the App and also kick off the registerServiceWorker.js.
- This service worker takes care of caching and updating files for the end-user. It allows for offline capability and faster page loads after the initial visit.



2.4 Start the React App

2.4 Start the React App

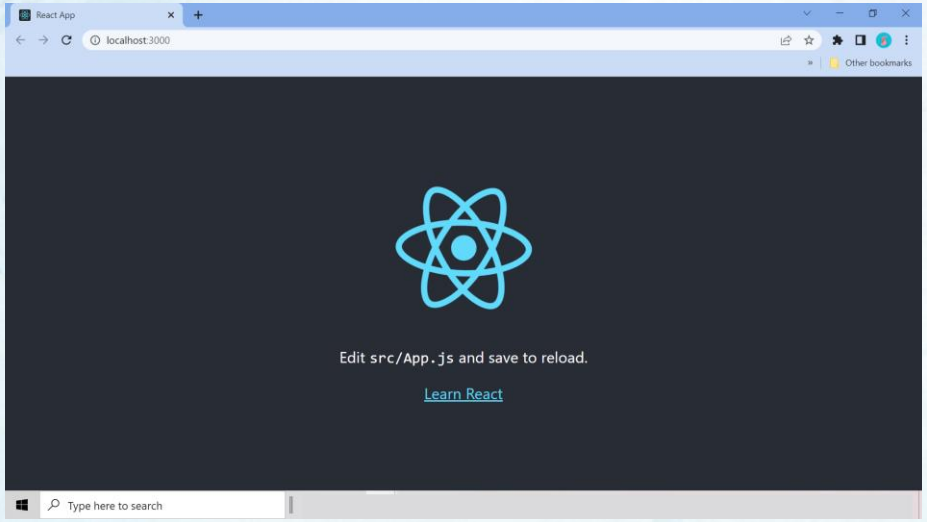
- You just need to run npm start in your app directory to begin serving the development server.



```
C:\>cd my-app

C:\my-app>npm start

> my-app@0.1.0 start
> react-scripts start

i ?wds?: Project is running at http://192.168.0.51/
i ?wds?: webpack output is served from
i ?wds?: Content not from webpack is served from C:\my-app\public
i ?wds?: 404s will fallback to /
Starting the development server...
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.0.51:3000

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

2.4 Start the React App

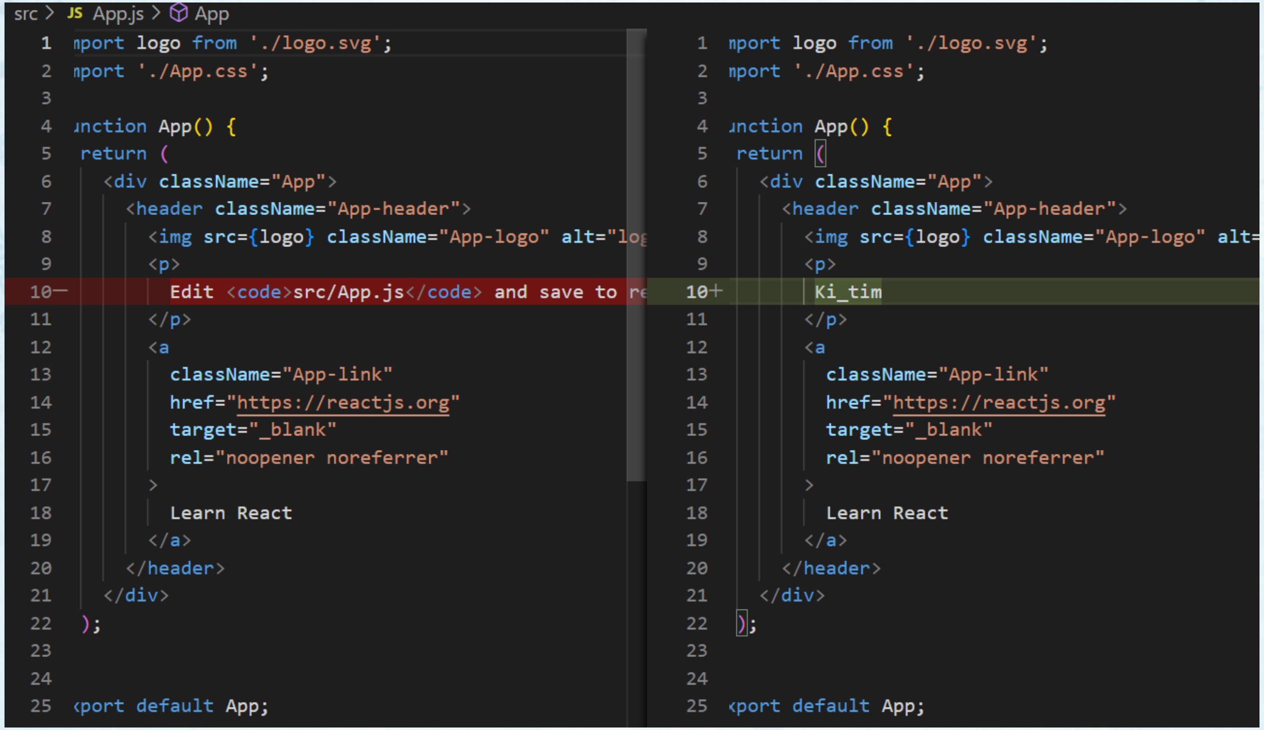
- It should auto-open a tab in your browser that points to http://localhost:3000/
- The app will be automatically launched in the default browser.



2.4 Start the React App

- As stated, any changes to the source code will live updated here.
- Open src/App.js in your text editor. You will see JSX, which is how React adds XML syntax to JavaScript.
- It provides an intuitive way to build React components and is compiled to JavaScript at runtime.
- Change the main paragraph text to read “Your Name” in App.js and save the file.
- Switch over to your browser and see the update.

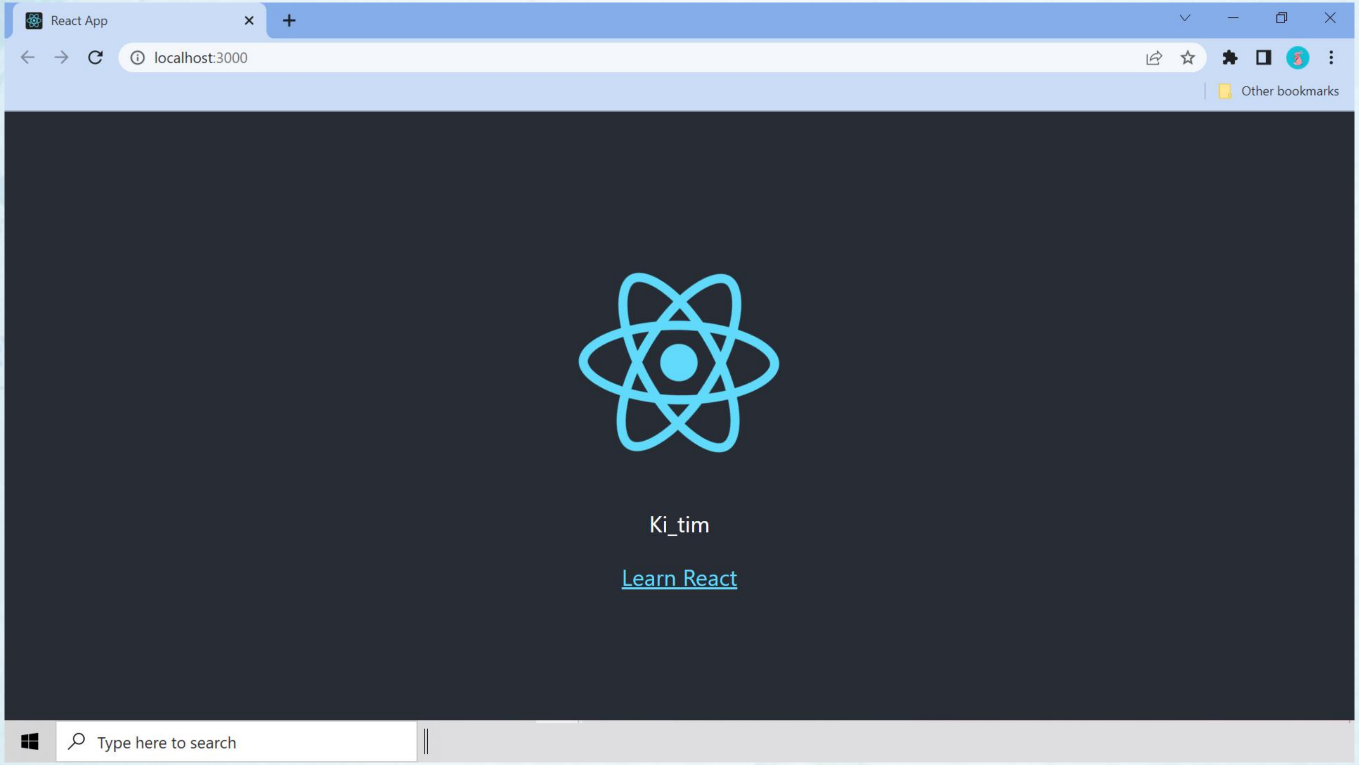
2.4 Start the React App


src > JS App.js > App

```
mport logo from './logo.svg';
mport './App.css';

ınction App() {
 return (
   <div className="App">
     <header className="App-header">
       <img src={logo} className="App-logo" alt="log
       <p>
-        Edit <code>src/App.js</code> and save to re
       </p>
       <a
         className="App-link"
         href="https://reactjs.org"
         target="_blank"
         rel="noopener noreferrer"
       >
         Learn React
       </a>
     </header>
   </div>
 );


‹port default App;
```

```
mport logo from './logo.svg';
mport './App.css';

ınction App() {
 return (
   <div className="App">
     <header className="App-header">
       <img src={logo} className="App-logo" alt=
       <p>
+        Ki_tim
       </p>
       <a
         className="App-link"
         href="https://reactjs.org"
         target="_blank"
         rel="noopener noreferrer"
       >
         Learn React
       </a>
     </header>
   </div>
 );


‹port default App;
```

2.4 Start the React App

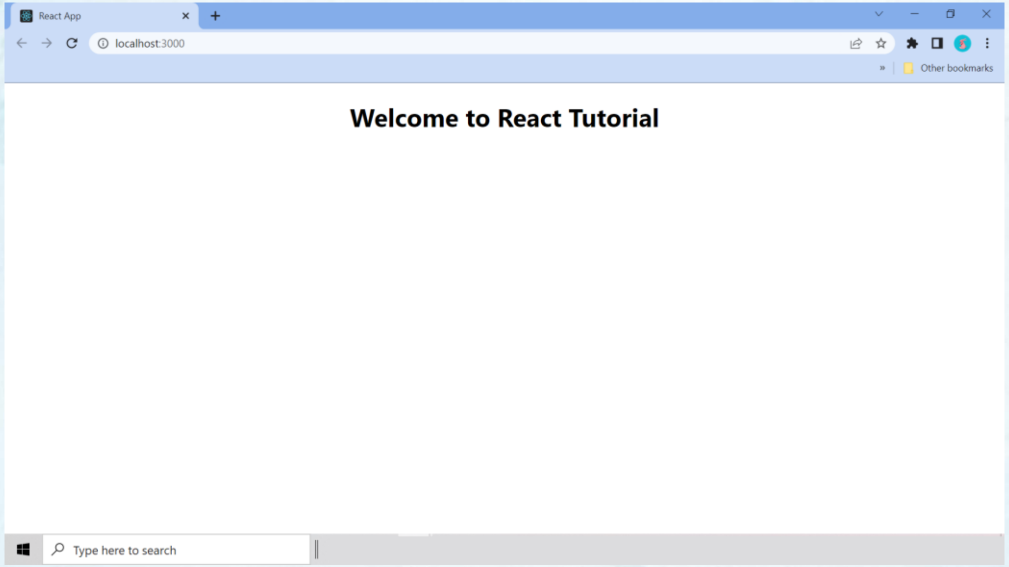


2.5 Assignment

2.5 Assignment

Outline:

To create a React App, remove the existing content of the App and add a welcome note as a heading.



Welcome to React Tutorial