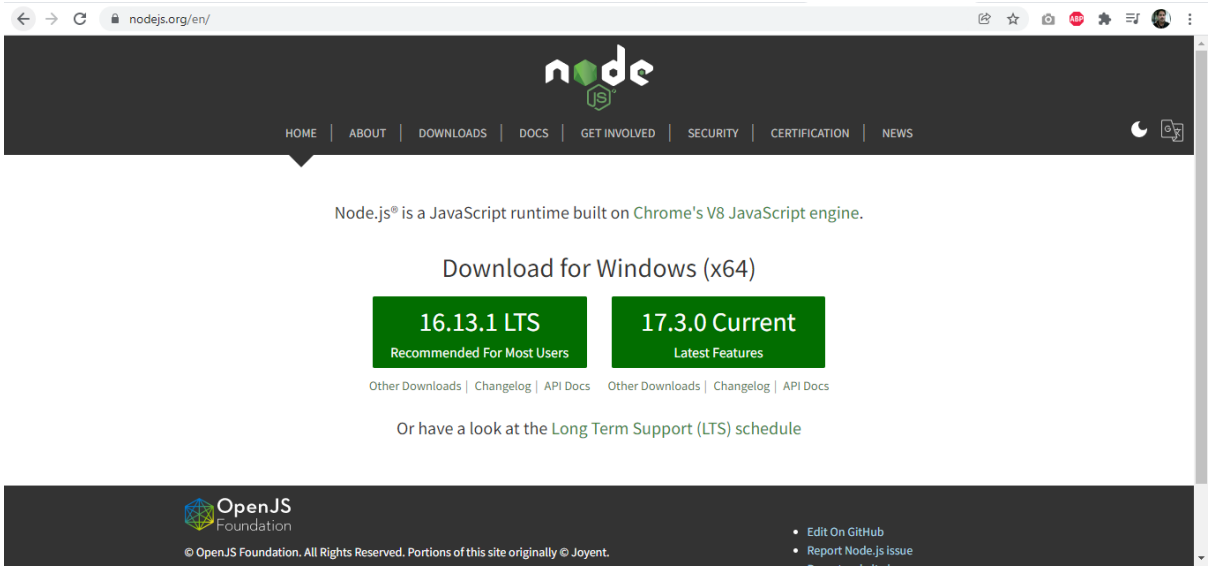
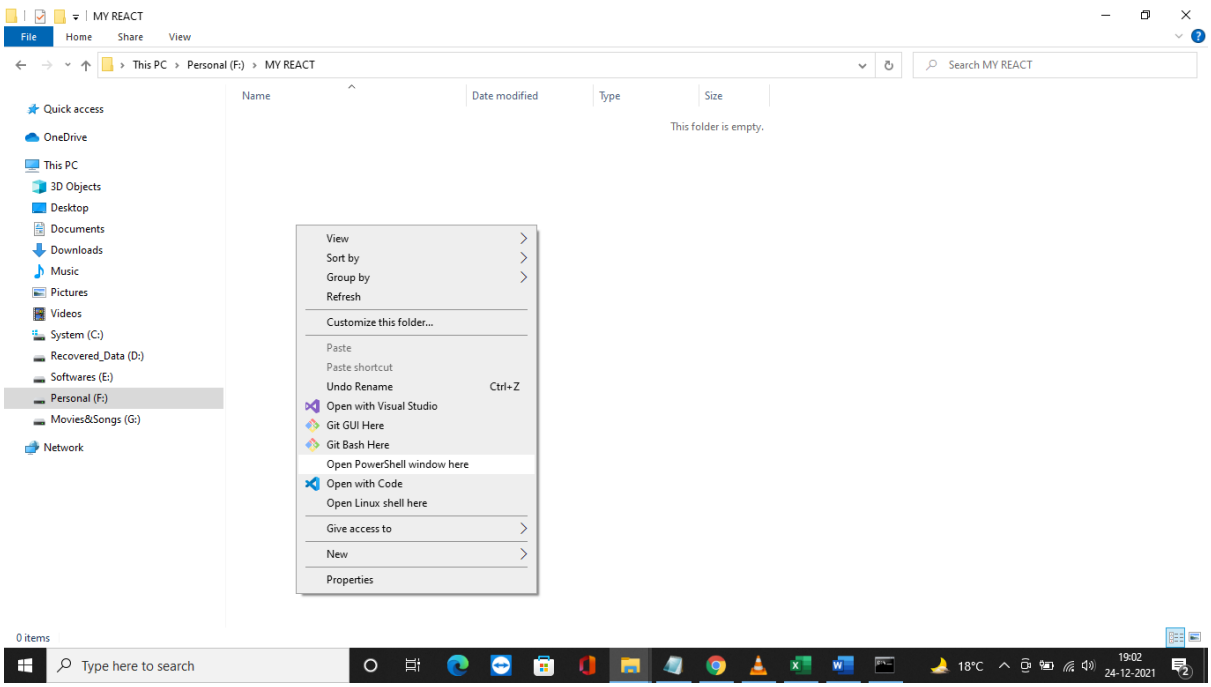
# Learn step by step REACT.JS with CURD operation




C:\Users\Hewlett Packard>node --version

v14.17.6

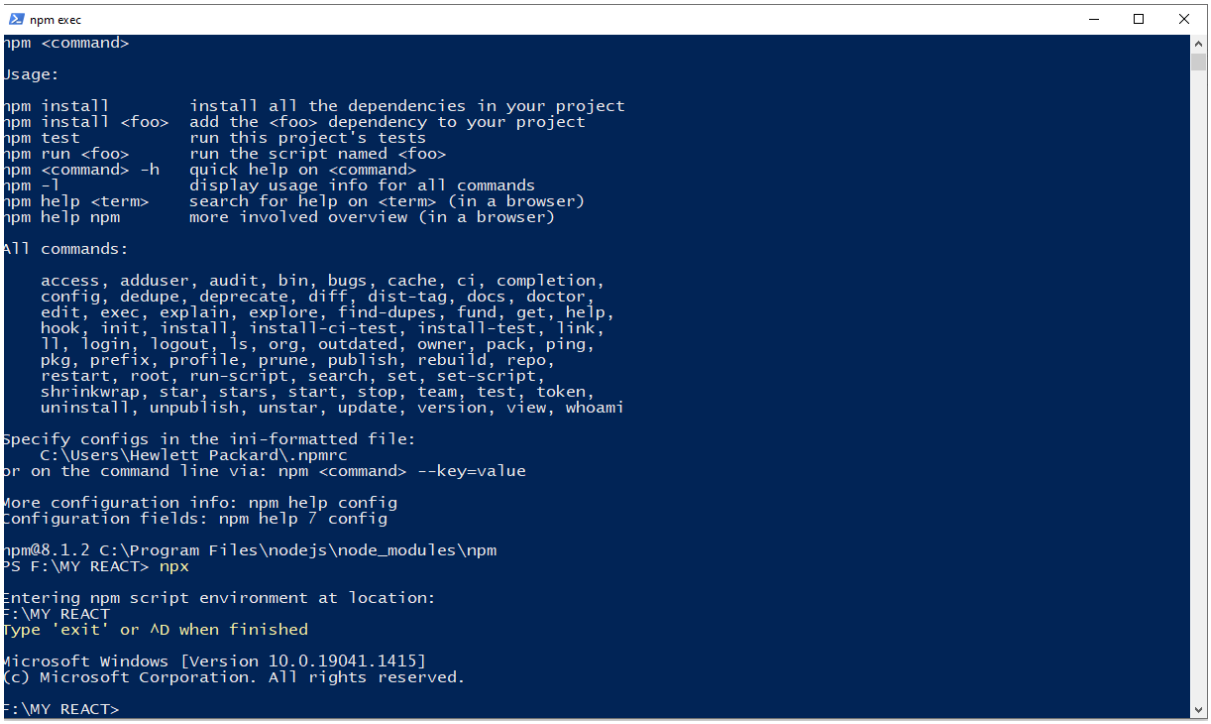



**Shift + Right click, open power shell**

C:\Users\Hewlett Packard>npm

```
Windows PowerShell
PS F:\MY REACT> node
Welcome to Node.js v16.13.1.
Type ".help" for more information.
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>
PS F:\MY REACT> node --version
v16.13.1
PS F:\MY REACT> npm
npm <command>

Usage:

npm install        install all the dependencies in your project
npm install <foo>  add the <foo> dependency to your project
npm test           run this project's tests
npm run <foo>      run the script named <foo>
npm <command> -h   quick help on <command>
npm -l             display usage info for all commands
npm help <term>    search for help on <term> (in a browser)
npm help npm       more involved overview (in a browser)

All commands:

    access, adduser, audit, bin, bugs, cache, ci, completion,
    config, dedupe, deprecate, diff, dist-tag, docs, doctor,
    edit, exec, explain, explore, find-dupes, fund, get, help,
    hook, init, install, install-ci-test, install-test, link,
    ll, login, logout, ls, org, outdated, owner, pack, ping,
    pkg, prefix, profile, prune, publish, rebuild, repo,
    restart, root, run-script, search, set, set-script,
    shrinkwrap, star, stars, start, stop, team, test, token,
    uninstall, unpublish, unstar, update, version, view, whoami

Specify configs in the ini-formatted file:
    C:\Users\Hewlett Packard\.npmrc
or on the command line via: npm <command> --key=value

More configuration info: npm help config
Configuration fields: npm help 7 config

npm@8.1.2 C:\Program Files\nodejs\node_modules\npm
PS F:\MY REACT>
```

C:\Users\Hewlett Packard>npx





F:\MY REACT>node --version
v16.13.1

F:\MY REACT>npm --version
8.1.2

F:\MY REACT>npx --version
8.1.2

F:\MY REACT>

If I have to install the skeleton project, I have to type `npx create-react-app my-app` in command-line.

I was wondering why does the Facebook in Github have `npx create-react-app my-app` rather than `npm create-react-app my-app`?

NPM - *Manages* packages *but* doesn't make life easy *executing* any.
NPX - A tool for *executing* Node packages.
NPX comes bundled with NPM version 5.2+

NPM by itself does not simply run any package. it doesn't run any package in a matter of fact. If you want to run a package using NPM, you must specify that package in your package.json file.

When executables are installed via NPM packages, NPM links to them:

1. *local* installs have "links" created at ./node_modules/.bin/ directory.
2. *global* installs have "links" created from the global bin/ directory (e.g. /usr/local/bin) on Linux or at %AppData%/npm on Windows.

**Documentation you should read**

---

# NPM:

One might install a package locally on a certain project:

```
npm install some-package
```

Now let's say you want NodeJS to execute that package from the command line:

```
$ some-package
```

The above will **fail**. Only **globally installed** packages can be executed by typing their name *only*.

To fix this, and have it run, you must type the local path:

```
$ ./node_modules/.bin/some-package
```

You can technically run a locally installed package by editing your packages.json file and adding that package in the scripts section:

```
{
  "name": "whatever",
  "version": "1.0.0",
  "scripts": {
    "some-package": "some-package"
  }
}
```

Then run the script using npm run-script (or npm run):

```
npm run some-package
```

---

# NPX:

npx will check whether <command> exists in $PATH, or in the local project binaries, and execute it. So, for the above example, if you wish to execute the locally-installed package some-package all you need to do is type:

```
npx some-package
```

Another **major** advantage of npx is the ability to execute a package which wasn't previously installed:

```
$ npx create-react-app my-app
```

The above example will generate a react app boilerplate *within* the path the command had run in, and ensures that you always use the latest version of a generator or build tool without having to upgrade each time you're about to use it.

---

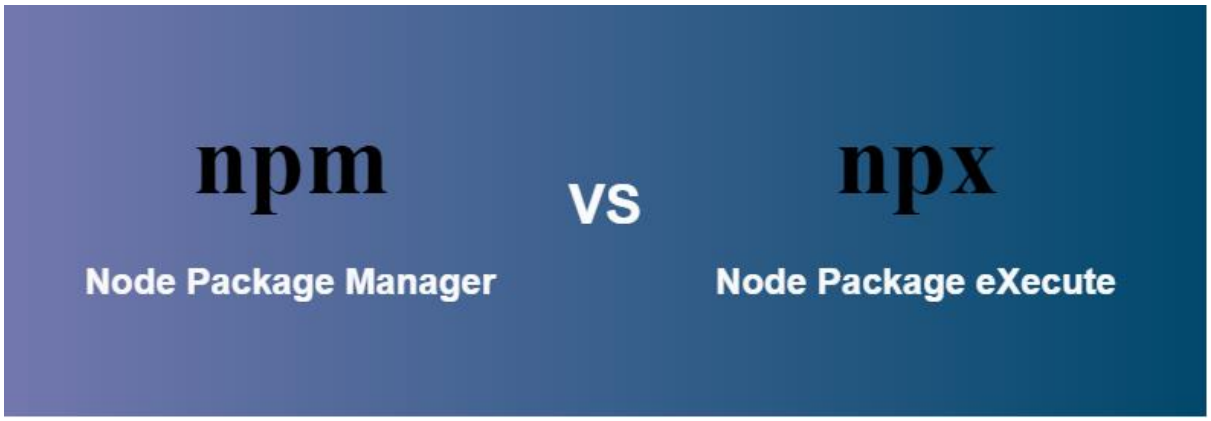
Use-Case Example:

npx command may be helpful in the script section of a package.json file, when it is unwanted to define a dependency which might not be commonly used or any other reason:

```
"scripts": {
   "start": "npx gulp@3.9.1",
   "serve": "npx http-server"
}
```

Call with: npm run serve

# What are the differences between npm and npx?




**NPM:** The npm stands for **Node Package Manager** and it is the default package manager for **Node.js.** It is written entirely in **JavaScript,** developed by *Isaac Z. Schlueter*, it was initially released on January 12, 2010. The **npm** manages all the packages and modules for node.js and consists of command–line client **npm**. It gets installed into the system with the **installation of node.js.** The required

packages and modules in the Node project are installed using **npm**. A package contains all the files needed for a module and modules are the JavaScript libraries that can be included in the Node project according to the requirement of the project.

**Execute package with npm:**

- **By typing the local path:** You have to write down the local path of your package like below:
  `./node_modules/.bin/your-package-name`
- **Locally installed:** You have to open the **package.json** file and write down the below scripts:

```
{
    "name": "Your app",
    "versiuon":  "1.0.0",
    "scripts":  {
            "your-package":  "your-package-name"
     }
}
```

**To run package:** After that, you can run your package by running the below command:

```
npm run your-package-name
```

**NPX:** The npx stands for **Node Package Execute** and it comes with the npm, when you installed npm above 5.2.0 version then automatically npx will installed. It is an npm package runner that can execute any package that you want from the npm registry without even installing that package. The npx is useful during a single time use package. If you have installed npm below 5.2.0 then npx is not installed in your system. You can check npx is installed or not by running the following command:

```
npx -v
```

If npx is not installed you can install that separately by running the below command.

```
npm install -g npx
```

**Execute package with npx:**

- **Directly runnable:** You can execute your package without installation, to do so run the following command.
  `npx your-package-name`

**Differences between npm and npx:**

| npm | npx |
|---|---|
| If you wish to run package through npm then you have to specify that package in your package.json and installed it locally. | A package can be executable without installing the package, it is an npm package runner so if any packages that aren't already installed it will installed automatically. |
| To use create-react-app in npm the commands are npm install create-react-app then create-react-app myApp(Installation required). | But in npx you can use that without installing like npx create-react-app myApp, this command is required in every app's life cycle only once. |
| Npm is a tool that use to install packages. | Npx is a tool that use to execute packages. |
| Packages used by npm are installed globally you have to care about pollution for the long term. | Packages used by npx are not installed globally so you have to carefree for the pollution for the long term. |

# Difference between npm and yarn


**NPM** and **Yarn** are package managers that help to manage a project's dependencies. A dependency is, as it sounds, something that a project depends on, a piece of code that is required to make the project work properly. We need them because managing the project's dependencies is a difficult task and it

quickly becomes tedious, and out of hand when the project grows. By managing the dependencies, we mean to include, un-include, and update them.
npm: It is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command-line client, also called npm, and an online database of public and paid-for private packages called the npm registry.

**yarn:** It stands for **Yet Another Resource Negotiator** and it is a package manager just like npm. It was developed by Facebook and is now open-source. The intention behind developing yarn (at that time) was to fix performance and security concerns with npm.
**The differences between npm and yarn are explained below:**

**Installation procedure**

- **npm:** npm is installed with Node automatically.
- **yarn:** To install yarn npm have to be installed.
  `npm install yarn --global`

**The lock file**

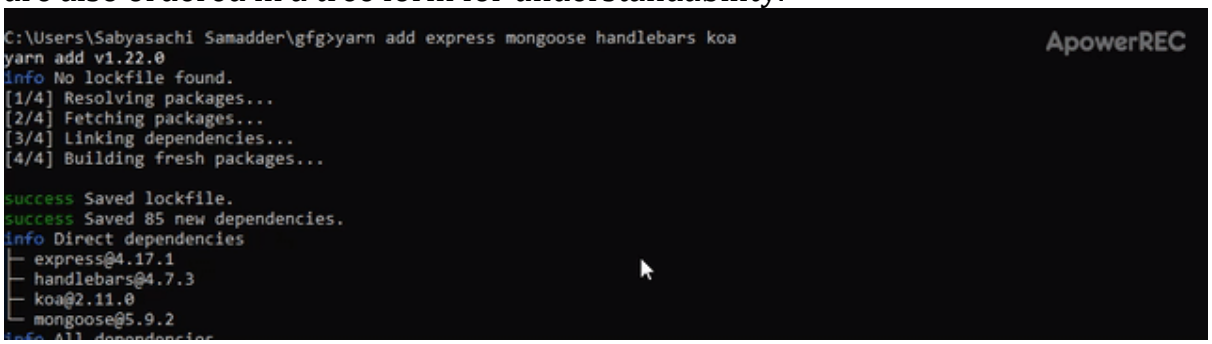
- **npm:** NPM generates a 'package-lock.json' file. The package-lock.json file is a little more complex due to a trade-off between determinism and simplicity. Due to this complexity, the package-lock will generate the same node_modules folder for different npm versions. Every dependency will have an exact version number associated with it in the package-lock file.
- **yarn:** Yarn generates a 'yarn.lock' file. Yarn lock files help in easy merge. The merges are predictable as well, because of the design of the lock file.

**Output log**

- **install:** The npm creates massive output logs of npm commands. It is essentially a dump of stack trace of what npm is doing.

```
C:\Users\Sabyasachi Samadder\gfg>npm install express mongoose handlebars koa
npm WARN saveError ENOENT: no such file or directory, open 'C:\Users\Sabyasachi Samadder\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\Sabyasachi Samadder\package.json'
npm WARN jquery-awesome-cursor@0.3.1 requires a peer of jquery@2.x but none is installed. You must install peer dependen
cies yourself.
npm WARN Sabyasachi Samadder No description
npm WARN Sabyasachi Samadder No repository field.
npm WARN Sabyasachi Samadder No README data
npm WARN Sabyasachi Samadder No license field.

+ koa@2.11.0
+ express@4.17.1
+ mongoose@5.9.2
+ handlebars@4.7.3
added 99 packages from 92 contributors and audited 775 packages in 19.968s
found 0 vulnerabilities
```

- **add:** The yarn output logs are clean, visually distinguishable and brief. They are also ordered in a tree form for understandability.




**Installing global dependencies**

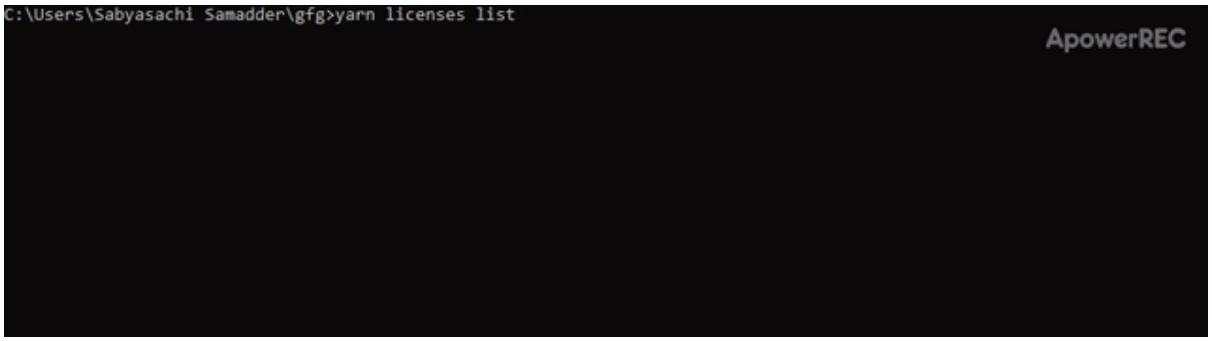
- **npm:** To install a global package, the command template for npm is:
  `npm install -g package_name@version_number`
- **yarn:** To install a global package, the command template for yarn is:
  `yarn global add package_name@version_number`

**The 'why' command:**

- **npm:** npm yet doesn't has a 'why' functionality built in.
- **yarn:** Yarn comes with a 'why' command that tells why a dependency is present in the project. For example, it is a dependency, a native module, or a project dependency.

**License Checker**

- **npm:** npm doesn't has a license checker that can give a handy description of all the licenses that a project is bound with, due to installed dependencies.
- **yarn:** Yarn has a neat license checker. To see them, run
  `yarn licenses list`




**Fetching packages**

- **npm:** npm fetches dependencies from the npm registry during every 'npm install' command.
- **Yarn:** yarn stores dependencies locally, and fetches from the disk during a 'yarn add' command (assuming the dependency(with the specific version) is present locally).

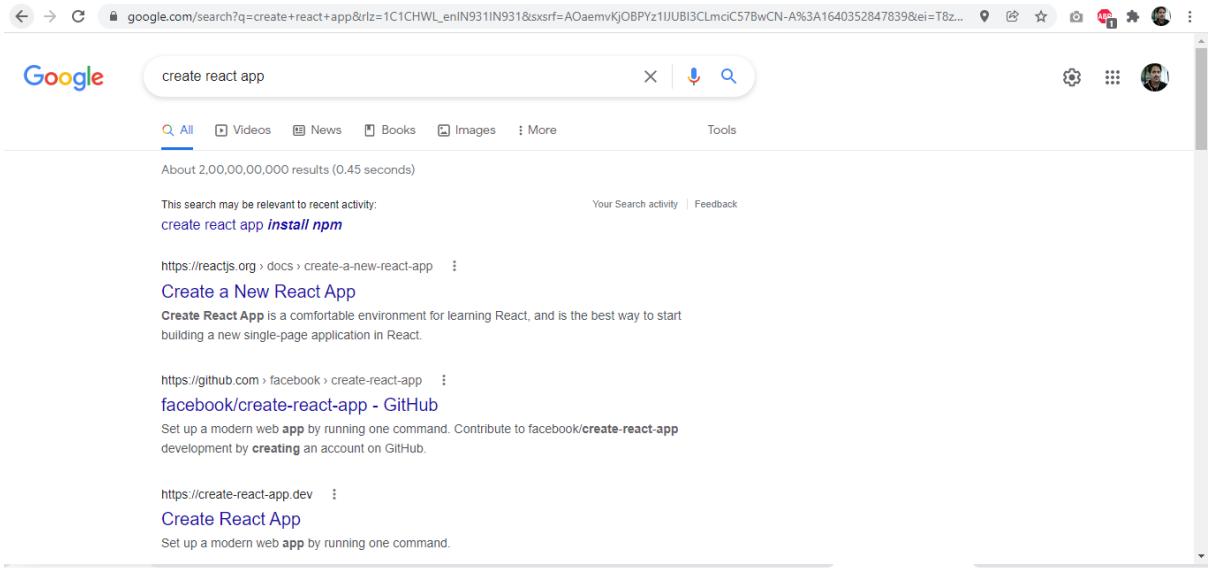
**Commands changed in yarn after npm**

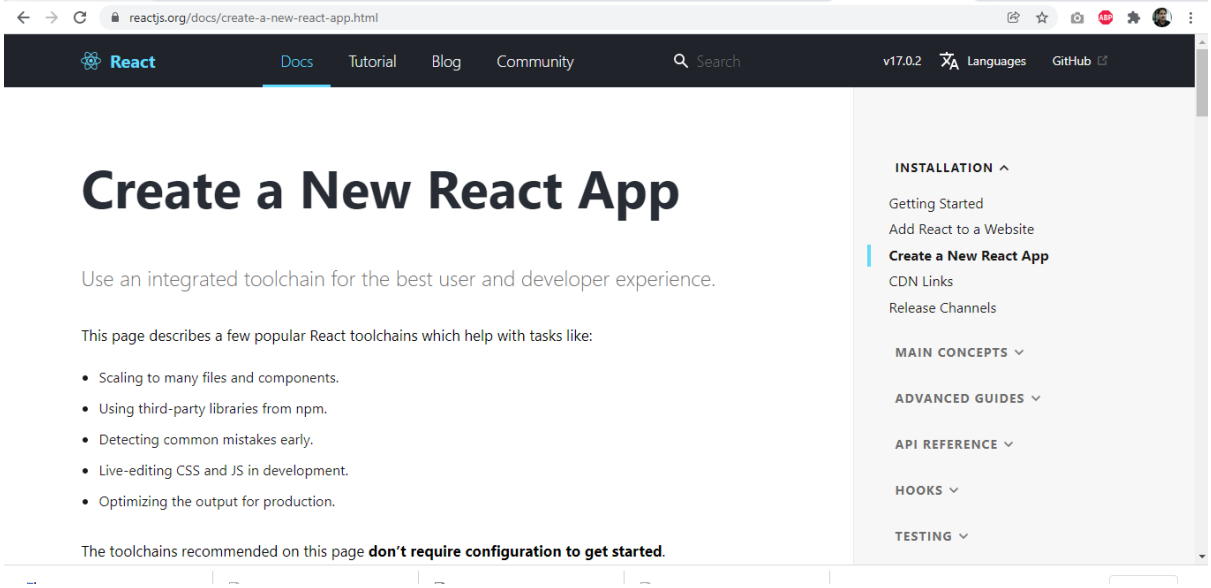
| command | npm | yarn |
|---|---|---|

| | | |
|---|---|---|
| Install dependencies | npm install | yarn |
| Install package | npm install package_name<br>npm install package_name@version_number | yarn add package_name<br>yarn add package_name@version_number |
| Uninstall package | npm uninstall package_name | yarn remove package_name |
| Install dev package | npm install package_name –save-dev | yarn add package_name –dev |
| Update dev package | npm update package_name<br>npm update package_name@version_number | yarn upgrade package_name<br>yarn upgrade package_name@version_number |
| View package | npm view package_name | yarn info package_name |
| Global install package | npm install -g package_name | yarn global add package_name |

**Commands same for npm and yarn:**

| npm | yarn |
|---|---|
| npm init | yarn init |
| npm run [script] | yarn run [script] |
| npm list | yarn list |
| npm test | yarn test |
| npm link | yarn link |
| npm login or logout | yarn login or logout |
| npm publish | yarn publish |



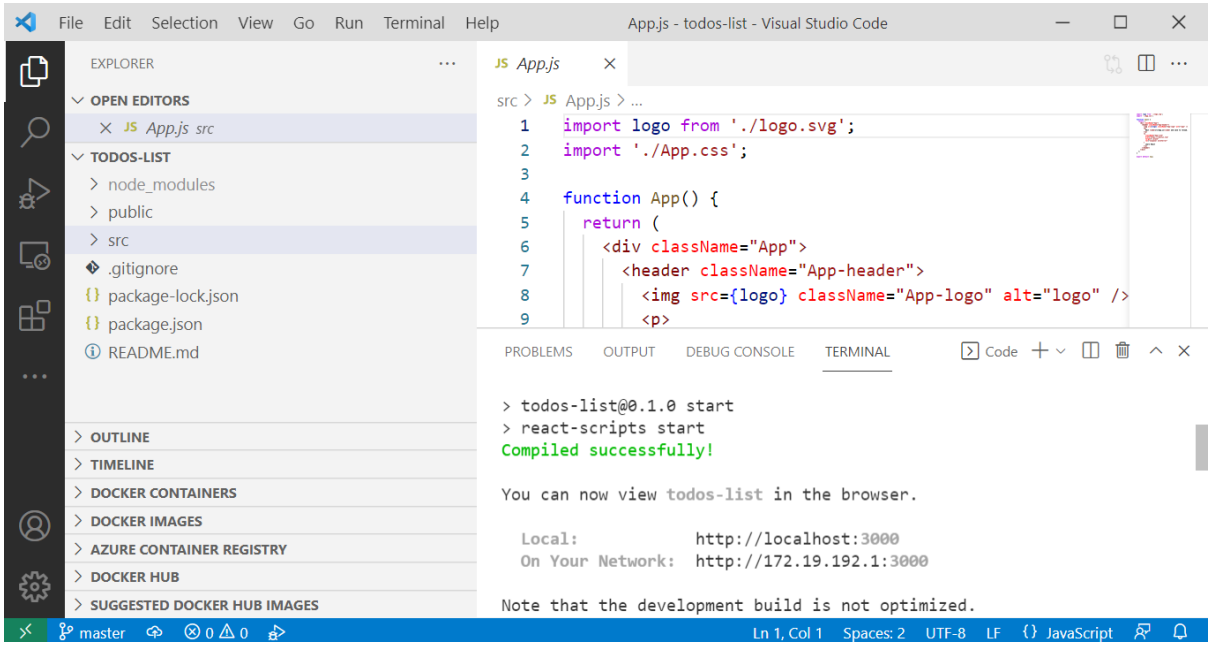

https://reactjs.org/docs/create-a-new-react-app.html

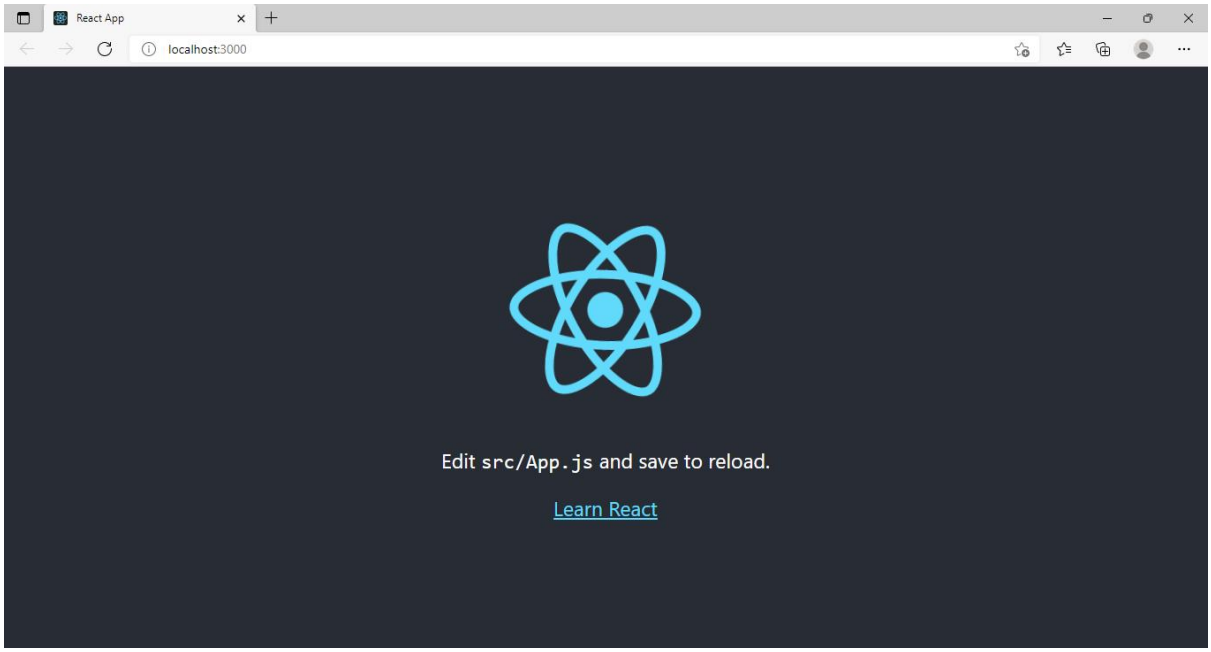
have Node >= 14.0.0 and npm >= 5.6 on your machine. To create a project, run:

```
npx create-react-app my-app
cd my-app
npm start
```

```
npm exec
F:\MY REACT>npx create-react-app todos-list
Need to install the following packages:
  create-react-app
Ok to proceed? (y) y
npm WARN              tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.

Creating a new React app in F:\MY REACT\todos-list.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...


added 1374 packages in 6m

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

Initialized a git repository.

Installing template dependencies using npm...

added 33 packages in 13s

163 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...


removed 1 package, and audited 1407 packages in 7s

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

6 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Created git commit.

Success! Created todos-list at F:\MY REACT\todos-list
Inside that directory, you can run several commands:
```

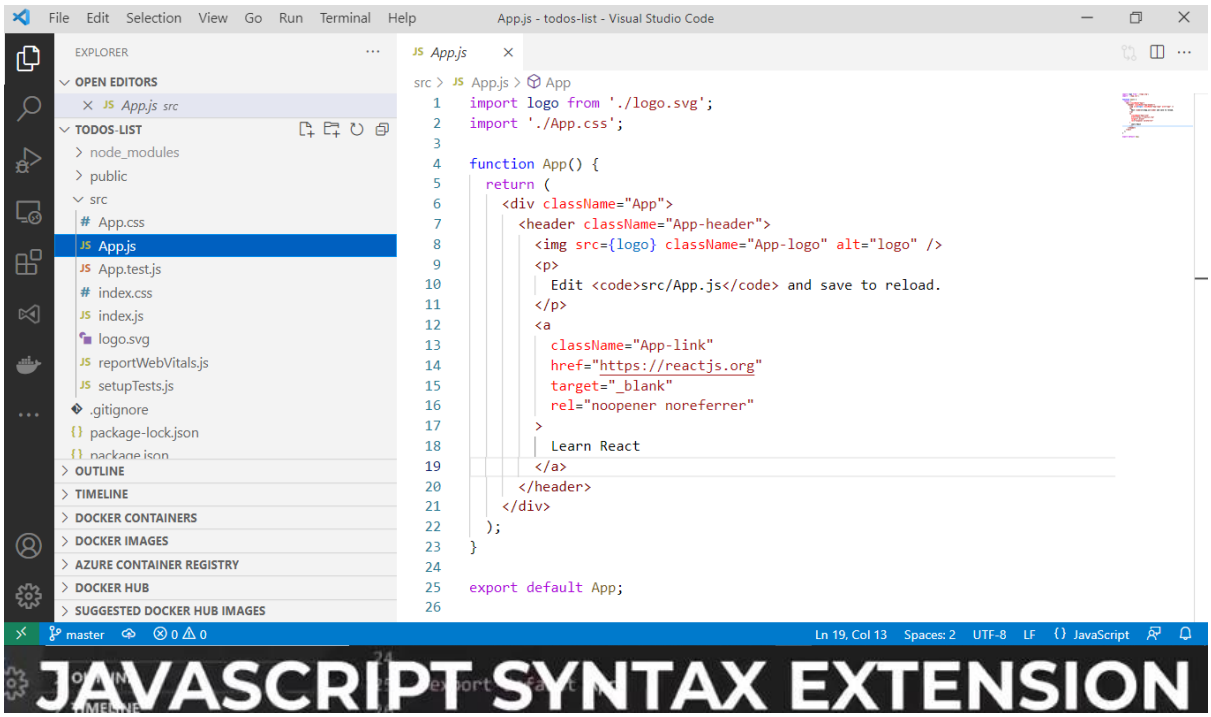
## Npm start






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

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

export default App;
```

Note : Why to use **className** **despite of class , the reason class is keyword in the javascript and to give unique name className here we are using. Same like for and html we have labelfor**




JSX makes it easier to write or add HTML in React. JSX can **easily convert HTML tags to react elements**. It is faster than regular JavaScript. JSX allows us to put HTML elements in DOM without using appendChild() or createElement() method.

JSX stands for JavaScript XML. It is simply a syntax extension of JavaScript. It allows us to directly write HTML in React (within JavaScript code). It is easy to create a template using JSX in React, but it is not a simple template language instead it comes with the full power of JavaScript. It is faster than normal JavaScript as it performs optimizations while translating to regular JavaScript. Instead of separating the markup and logic in separated files, React uses *components* for this purpose. We will learn about components in detail in further articles.

**Syntax:**

```
const element = <h1>Welcome to GeeksforGeeks.</h1>;
```

**Characteristics of JSX:**

- JSX is not mandatory to use there are other ways to achieve the same thing but using JSX makes it easier to develop react application.
- JSX allows writing expression in { }. The expression can be any JS expression or React variable.
- To insert a large block of HTML we have to write it in a parenthesis i.e, ().
- JSX produces react elements.
- JSX follows XML rule.

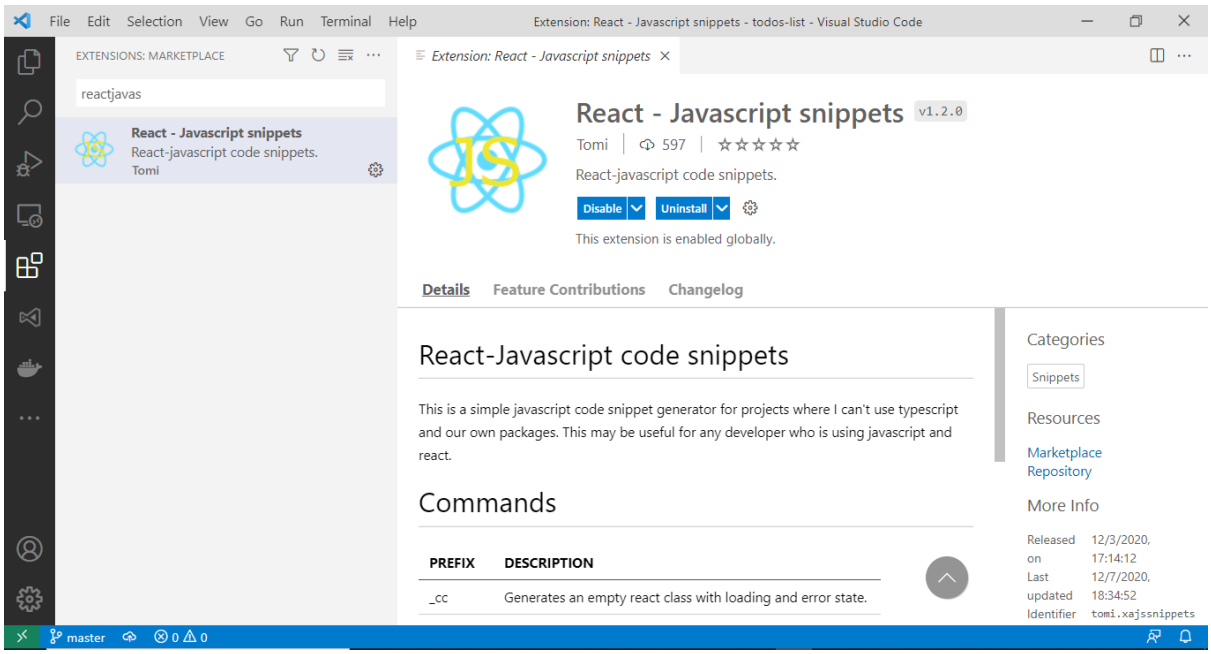
- After compilation, JSX expressions become regular JavaScript function calls.
- JSX uses camelcase notation for naming HTML attributes. For example, tabindex in HTML is used as tabIndex in JSX.

**Advantages of JSX:**

- JSX makes it easier to write or add HTML in React.
- JSX can easily convert HTML tags to react elements.
- It is faster than regular JavaScript.
- JSX allows us to put HTML elements in DOM without using appendChild() or createElement() method.
- As JSX is an expression, we can use it inside of if statements and for loops, assign it to variables, accept it as arguments, or return it from functions.
- JSX prevents XSS (cross-site-scripting) attacks popularly known as injection attacks.
- It is type-safe, and most of the errors can be found at compilation time.

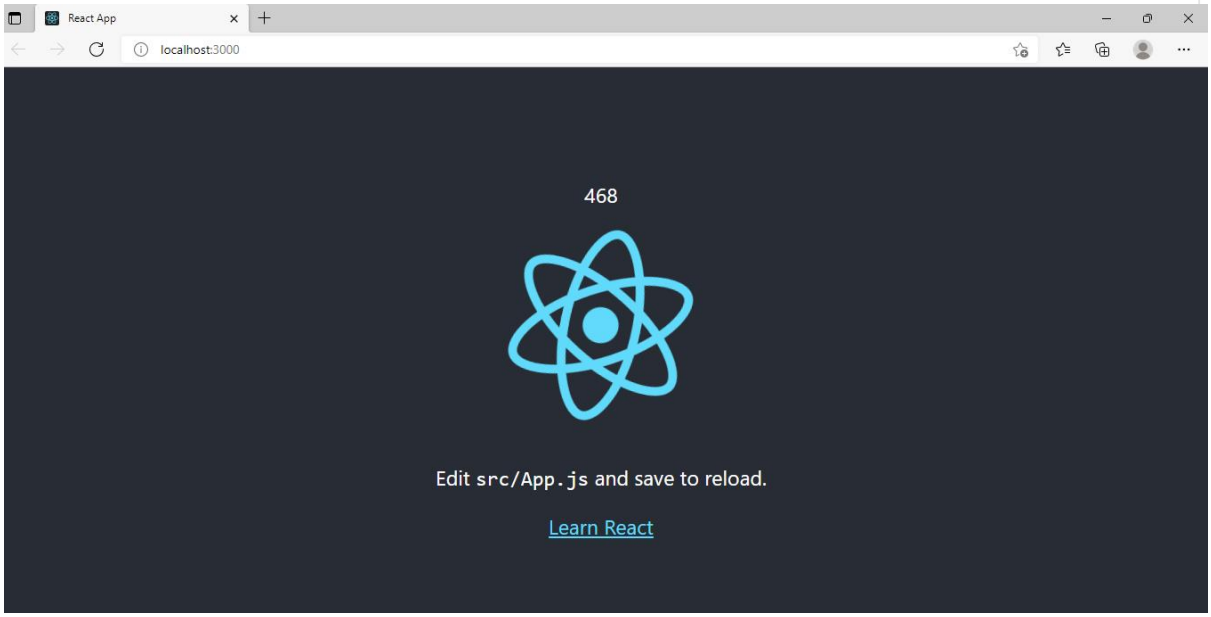
**Disadvantages of JSX:**

- JSX throws an error if the HTML is not correct.
- **In JSX HTML code must be wrapped in one top-level element otherwise it will give an error.**
- If HTML elements are not properly closed JSX will give an error.



rc > JS App.js > App

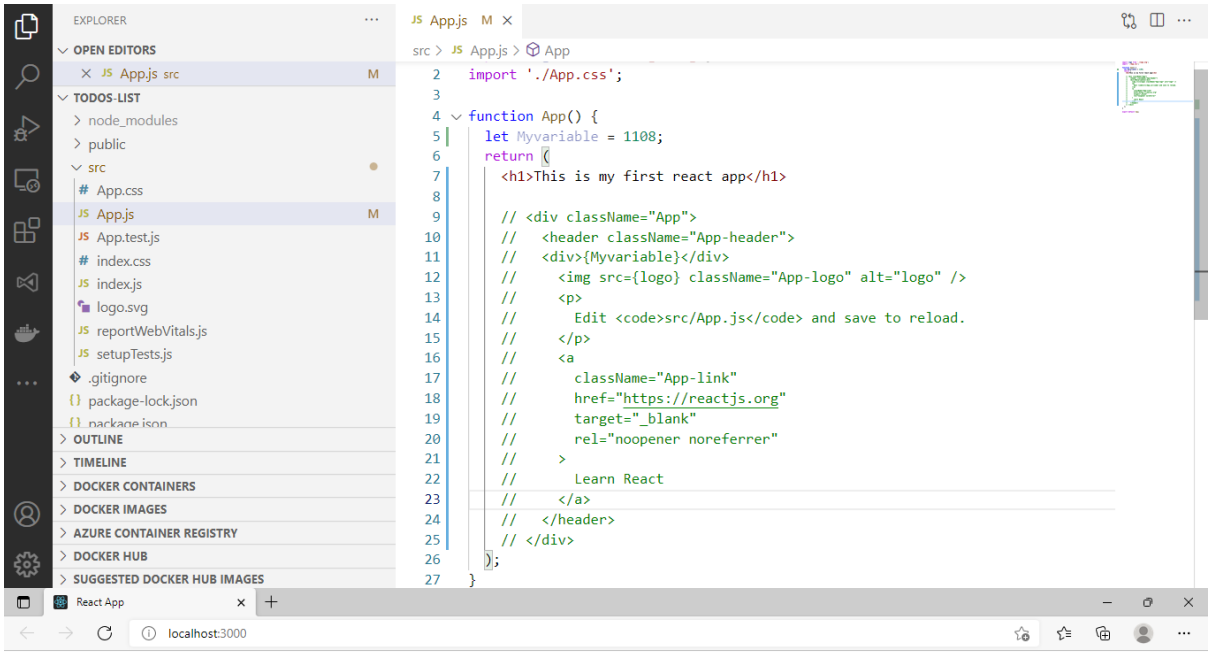
```
1   import logo from './logo.svg';
2   import './App.css';
3
4   function App() {
5     return (
6
7       <div className="App">
8         <header className="App-header">
9         <div>{123+345}</div>
10          <img src={logo} className="App-logo" alt="logo" />
11          <p>
12            Edit <code>src/App.js</code> and save to reload.
13          </p>
14          <a
15            className="App-link"
16            href="https://reactjs.org"
17            target="_blank"
18            rel="noopener noreferrer"
19          >
20            Learn React
21          </a>
22        </header>
23      </div>
24    );
25  }
26
```




```
function App() {
  return (

    <div className="App">
      <header className="App-header">
      <div>{123+345}</div>
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
```

```
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}
```


Note: In react we can create component in two ways 1. Function based component and 2. Class based component.




**This is my first react app**

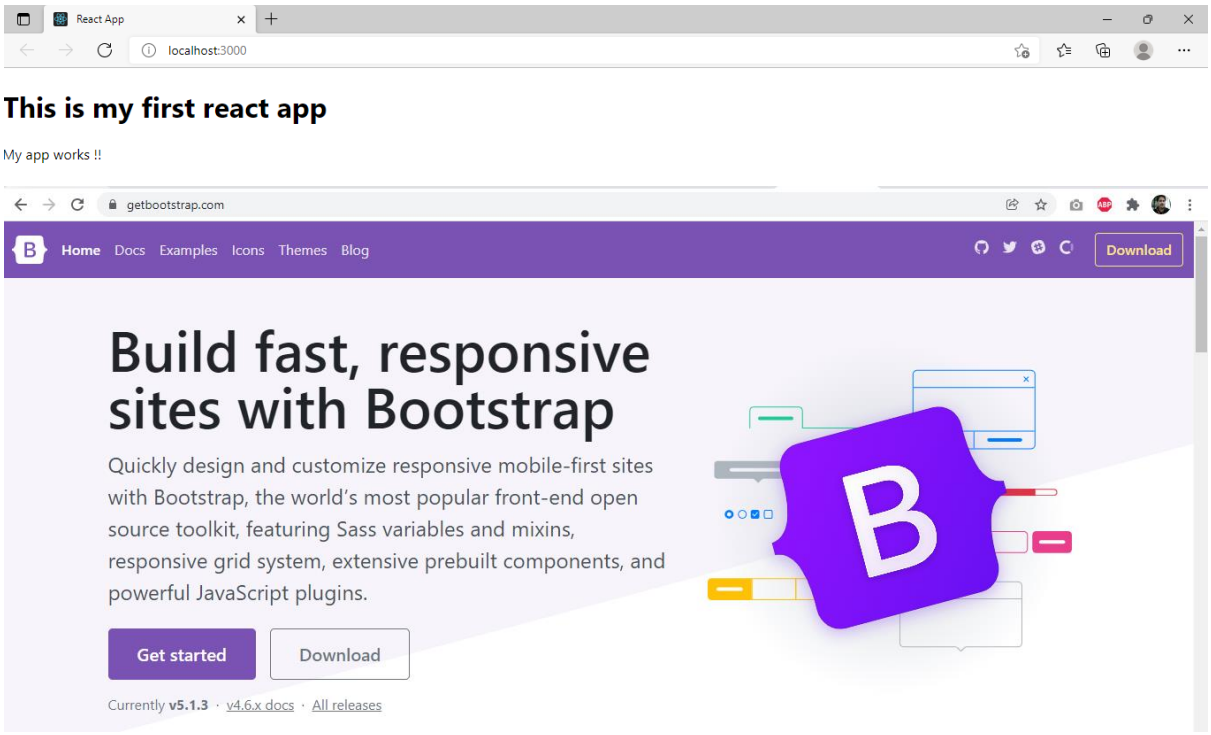
```
function App() {
  let Myvariable = 1108;
  return (
    <>
    <h1>This is my first react app</h1>
    <p>My app works !!</p>
    </>
  );
}

export default App;
```

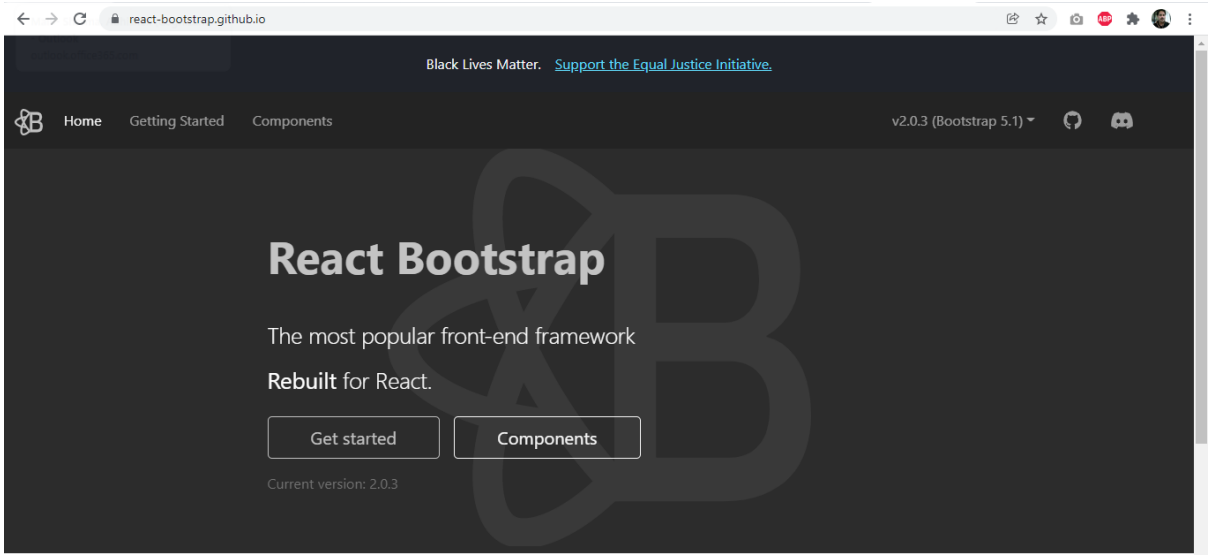
Note: it should have warp up with one main root tag.

## App.js

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

function App() {
  let Myvariable = 1108;
  return (
    <>
    <h1>This is my first react app</h1>
    <p>My app works !!</p>
```

```
    </>

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

export default App;
```






# App.js

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

function App() {
  let Myvariable = 1108;
  return (
    <nav className="navbar navbar-expand-lg navbar-light bg-light">
  <div className="container-fluid">
    <a className="navbar-brand" href="#">Navbar</a>
    <button className="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
      <span className="navbar-toggler-icon"></span>
    </button>
    <div className="collapse navbar-collapse" id="navbarSupportedContent">
      <ul className="navbar-nav me-auto mb-2 mb-lg-0">
        <li className="nav-item">
          <a className="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li className="nav-item">
          <a className="nav-link" href="#">Link</a>
        </li>
        <li classNameName="nav-item dropdown">
          <a className="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown
          </a>
          <ul className="dropdown-menu" aria-labelledby="navbarDropdown">
```

```
            <li><a className="dropdown-item" href="#">Action</a></li>
            <li><a className="dropdown-item" href="#">Another action</a></li>
            <li><hr className="dropdown-divider"/></li>
            <li><a className="dropdown-item" href="#">Something else
here</a></li>
          </ul>
        </li>
        <li className="nav-item">
          <a className="nav-link disabled">Disabled</a>
        </li>
      </ul>
      <form className="d-flex">
        <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search"/>
        <button className="btn btn-outline-success"
type="submit">Search</button>
      </form>
    </div>
  </div>
</nav>
  );
}

export default App;
```

**index.html**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />


    <title>React App</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
```

```
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
  </body>
</html>
```

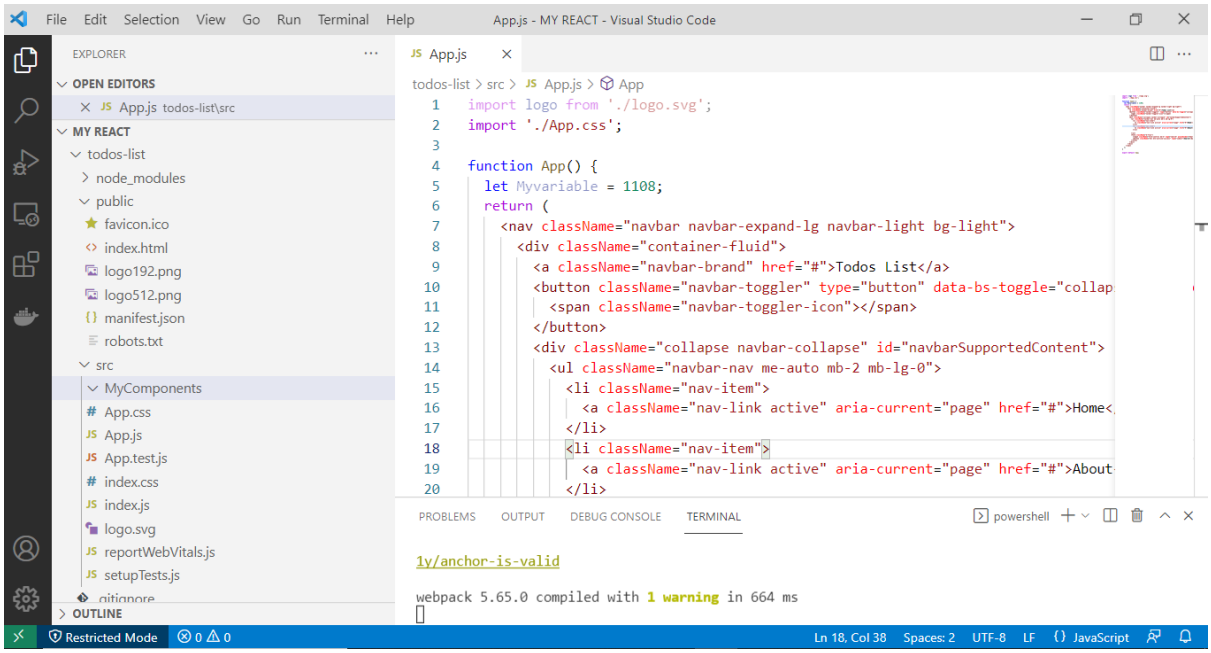

Note: Let's do some changes in the app.js reduce menu option.

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

function App() {
  let Myvariable = 1108;
  return (
    <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="container-fluid">
        <a className="navbar-brand" href="#">Todos List</a>
        <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span className="navbar-toggler-icon"></span>
        </button>
        <div className="collapse navbar-collapse" id="navbarSupportedContent">
          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">Home</a>
            </li>
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">About</a>
            </li>

          </ul>
```

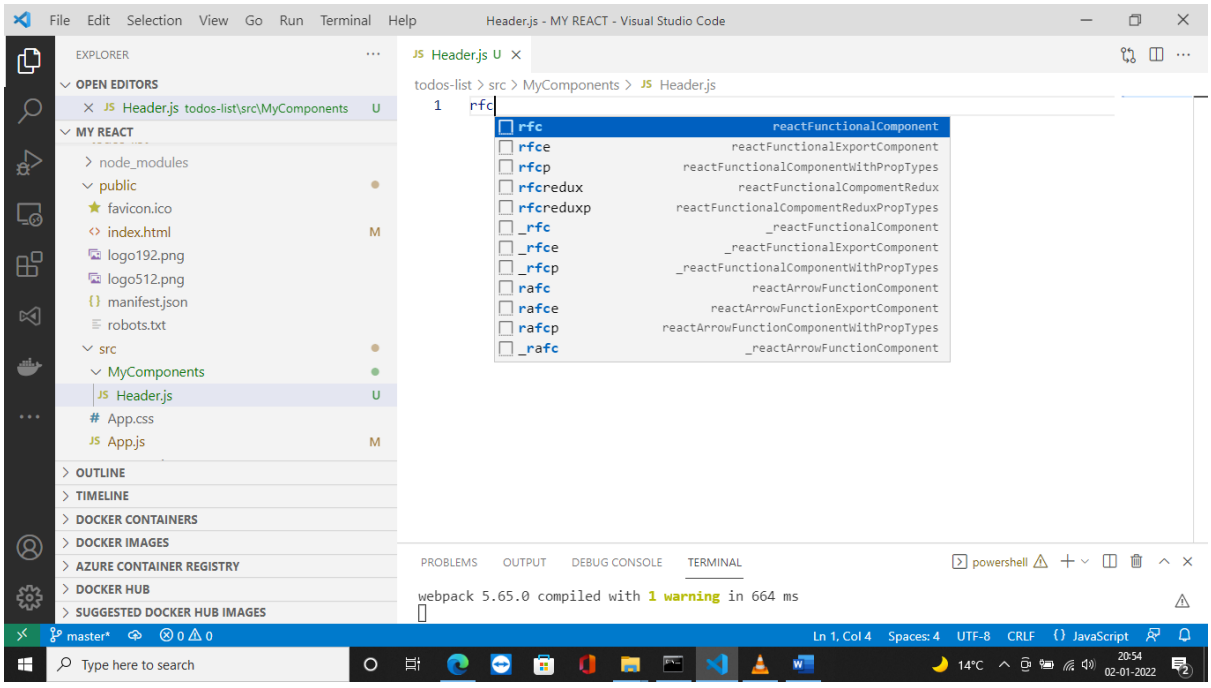
```
          <form className="d-flex">
            <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
            <button className="btn btn-outline-success"
type="submit">Search</button>
          </form>
        </div>
      </div>
    </nav>
  );
}

export default App;
```


## Install ES7 React/Redux/React-Native/JS snippets

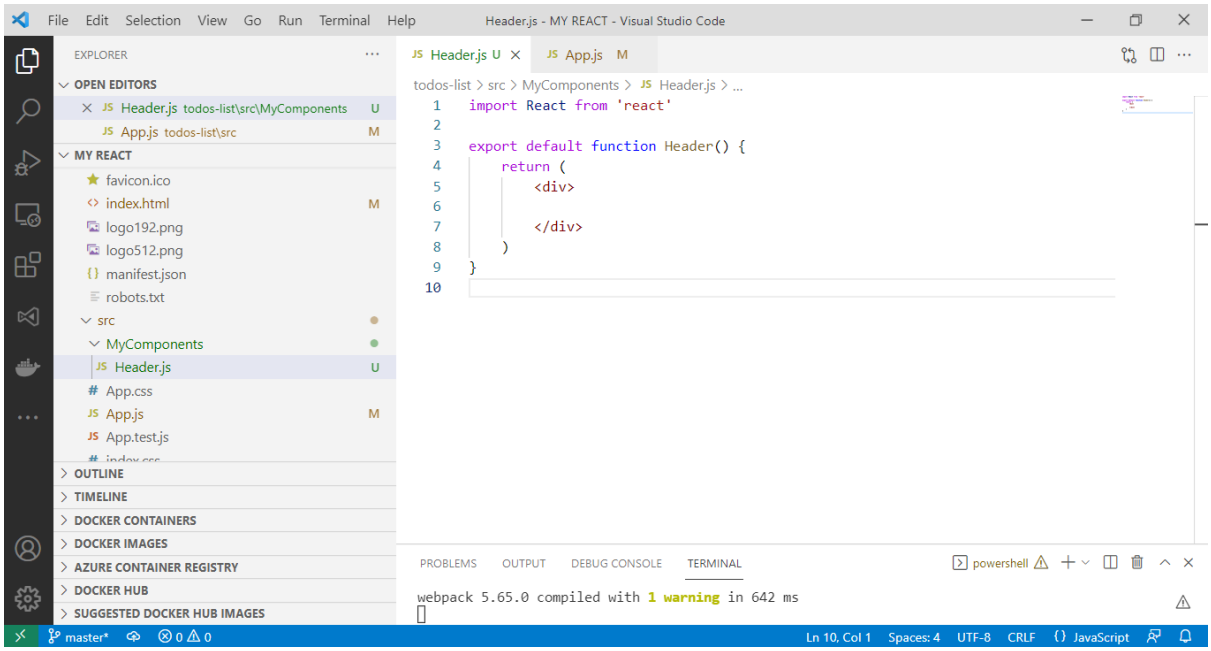

Now let's add new folder in SRC folder.




Now add header.js




**Now let's cut the app.js nav bar code and put into Header.js file.**

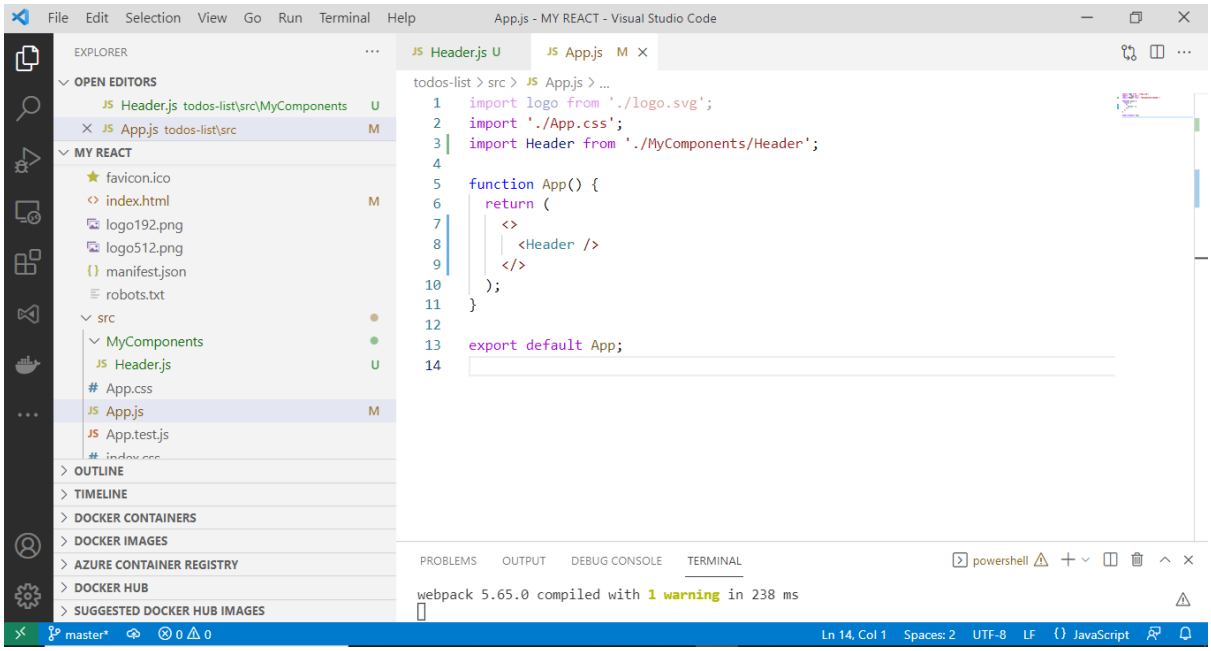


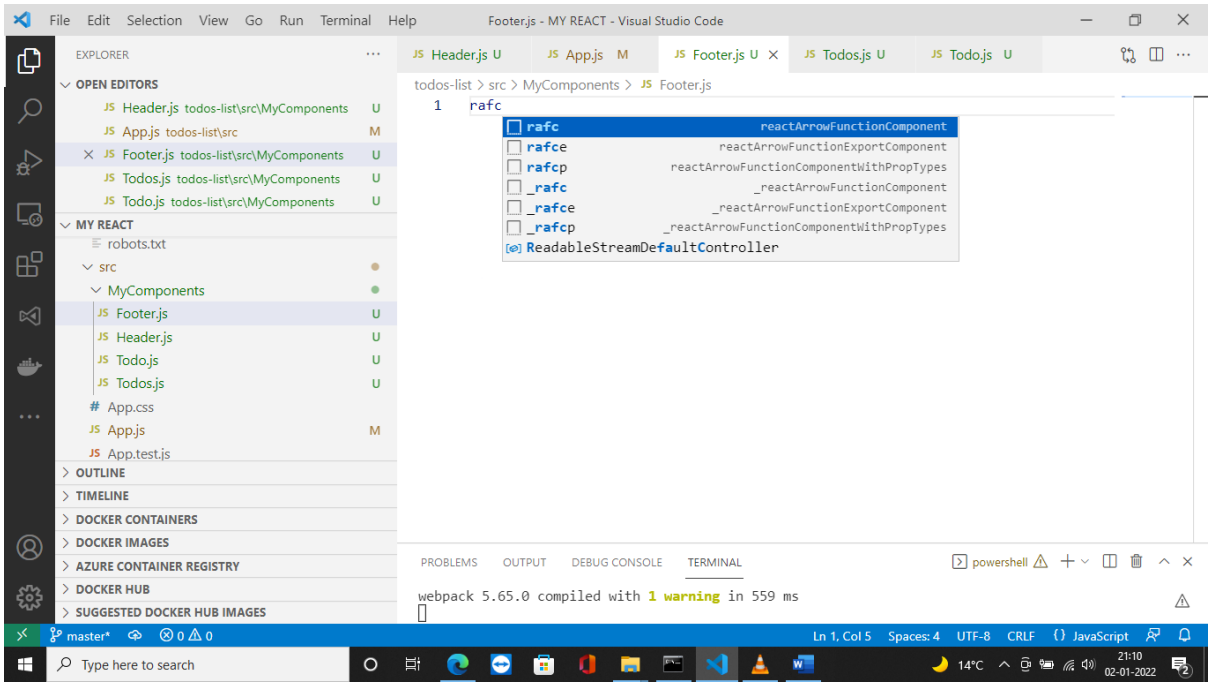

```
import React from 'react'

export default function Header() {
    return (
        <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="container-fluid">
        <a className="navbar-brand" href="#">Todos List</a>
        <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span className="navbar-toggler-icon"></span>
        </button>
        <div className="collapse navbar-collapse" id="navbarSupportedContent">
          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">Home</a>
            </li>
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">About</a>
            </li>

          </ul>
          <form className="d-flex">
            <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
            <button className="btn btn-outline-success"
type="submit">Search</button>
          </form>
        </div>
```

```
      </div>
    </nav>
    )
}
```

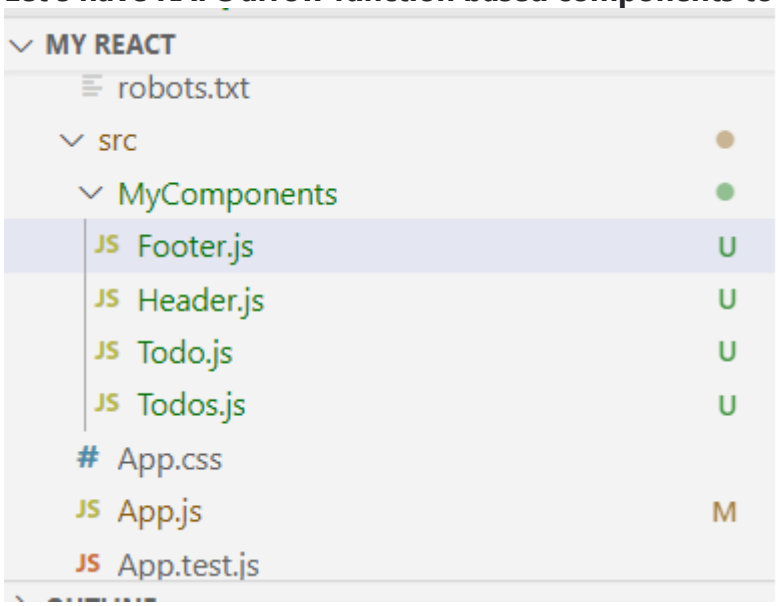
**And import Header.js file into app.js file as following**

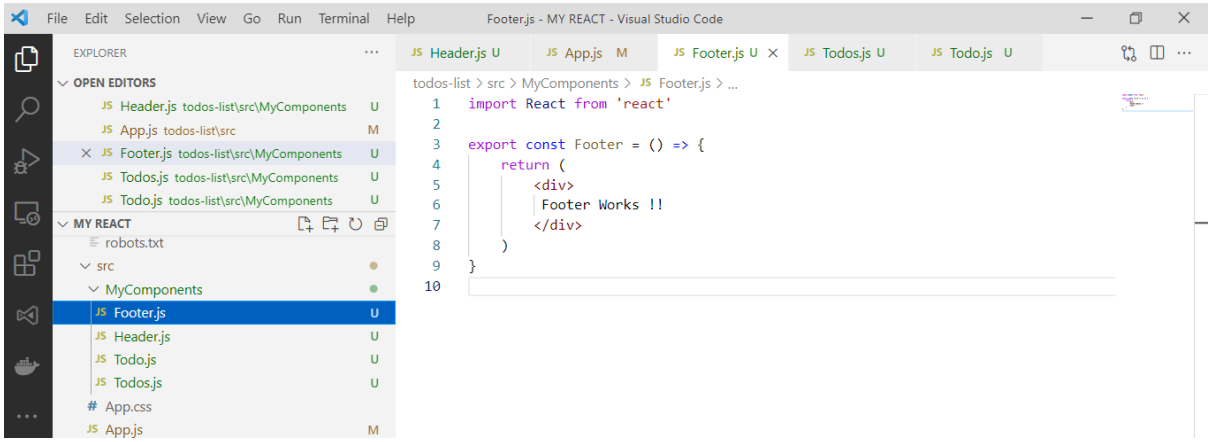
```
import logo from './logo.svg';
import './App.css';
import Header from './MyComponents/Header';

function App() {
  return (
    <>
      <Header />
    </>
  );
}

export default App;
```

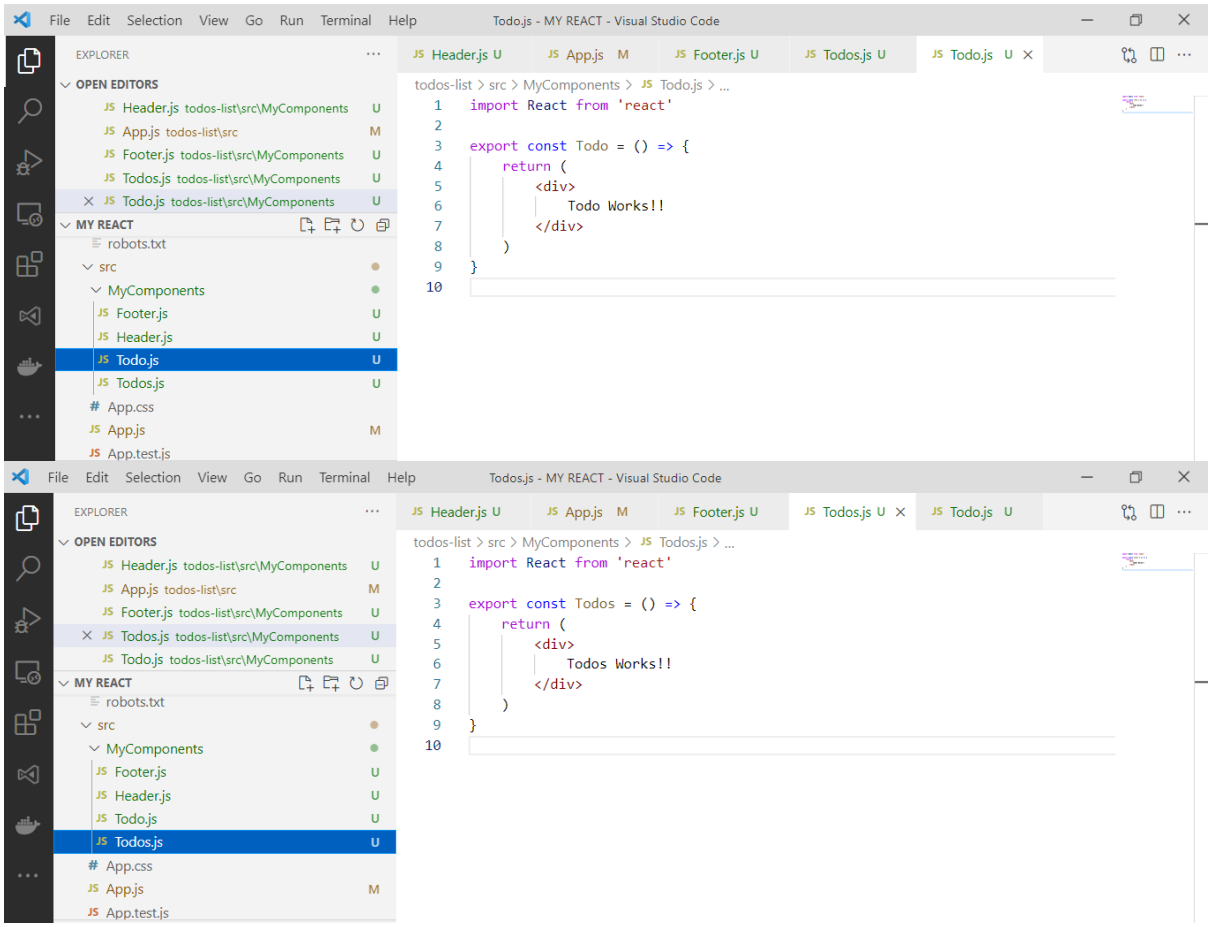



## Let's have RAFC arrow function based components to use





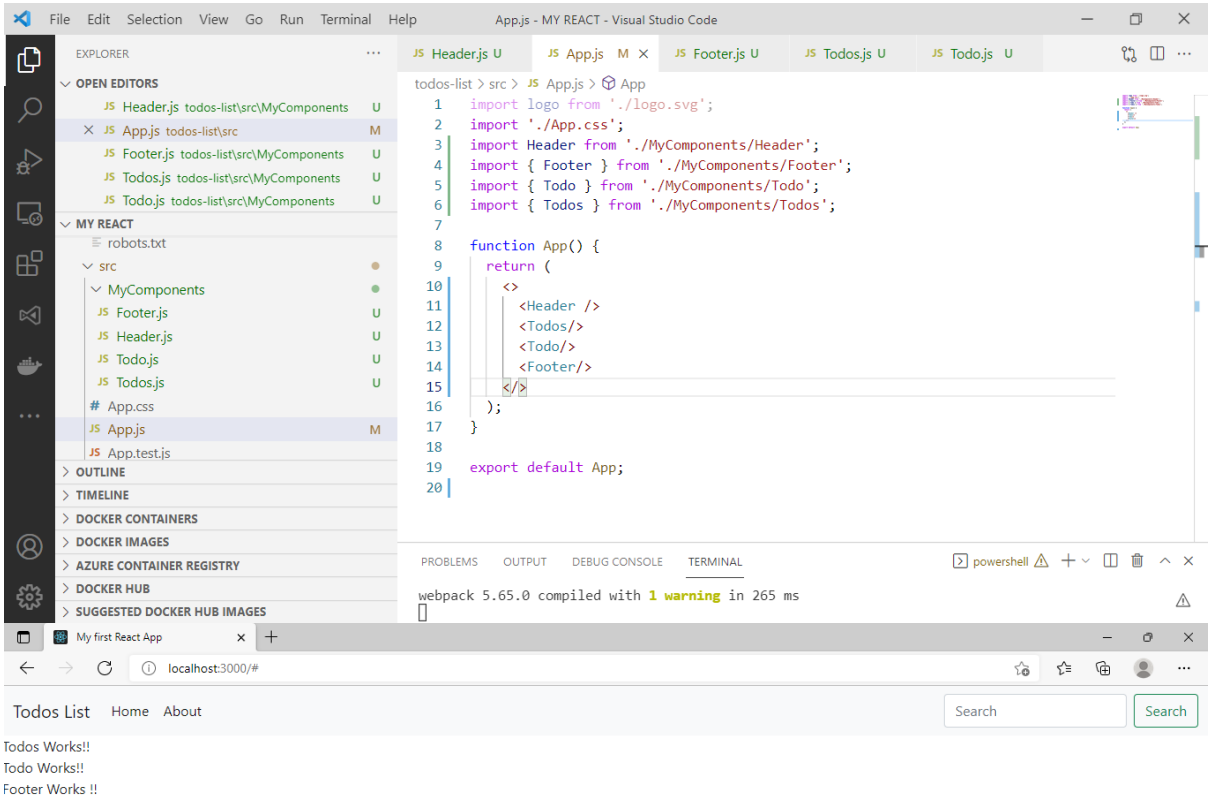

When to import and how

```
import Header from './MyComponents/Header'; //  export default

import { Footer } from './MyComponents/Footer'; // export const

import { Todo } from './MyComponents/Todo';
import { Todos } from './MyComponents/Todos';
```

**Now write the following into app.js**

```
import logo from './logo.svg';
import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { Todo } from './MyComponents/Todo';
import { Todos } from './MyComponents/Todos';

function App() {
  return (
    <>
      <Header />
      <Todos/>
```

```
      <Todo/>
      <Footer/>
    </>
  );
}

export default App;
```




**Now let's pass title in the Header component with the help PROPS object**

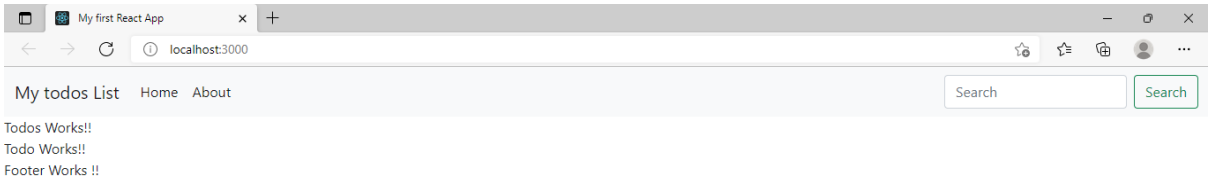
**Header.js**

```
import React from 'react'

export default function Header(props) {
    return (
        <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="container-fluid">
        <a className="navbar-brand" href="#">{props.title}</a>
        <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span className="navbar-toggler-icon"></span>
        </button>
        <div className="collapse navbar-collapse" id="navbarSupportedContent">
```

```
          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">Home</a>
            </li>
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">About</a>
            </li>

          </ul>
          <form className="d-flex">
            <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
            <button className="btn btn-outline-success"
type="submit">Search</button>
          </form>
        </div>
      </div>
    </nav>
    )
}
```

**App.js**

```
import logo from './logo.svg';
import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { TodoItem } from './MyComponents/TodoItem';
import { Todos } from './MyComponents/Todos';

function App() {
  return (
    <>
      <Header title="My todos List"/>
      <Todos/>
      <TodoItem/>
      <Footer/>
    </>
  );
}

export default App;
```




**You can also do without using props as following**

```
export default function Header({title}) {
    return (
        <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="container-fluid">
        <a className="navbar-brand" href="#">{title}</a>
        <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span className="navbar-toggler-icon"></span>
```

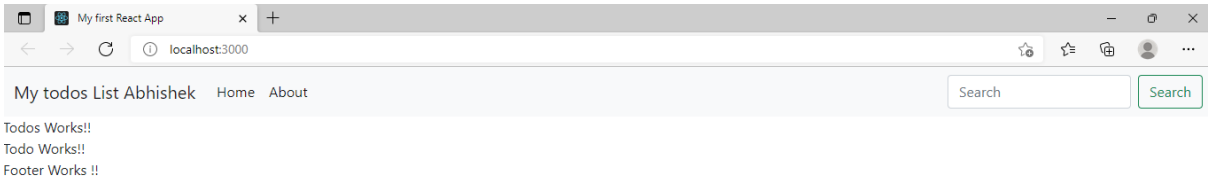
**Same output you will get , you could use multiple variable also as follows**

**App.js**

```
function App() {
  return (
    <>
      <Header title="My todos List" myvar="Abhishek"/>
      <Todos/>
      <TodoItem/>
      <Footer/>
    </>
  );
}
```

**Header.js**

```
export default function Header({ title, myvar }) {
  return (
    <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="container-fluid">
        <a className="navbar-brand" href="#">{title} {myvar}</a>
        <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span className="navbar-toggler-icon"></span>
```




**Note: Better to use props only, its easy to distinguish different variables.**
**You can apply Boolean values also, as follows**

**App.js**

```
function App() {
  return (
    <>
      <Header title="My todos List" myvar="Abhishek" searchBar={false}/>
      <Todos/>
      <TodoItem/>
      <Footer/>
    </>
  );
}
```

**Header.js**

```
{props.searchBar ?   <form className="d-flex">
            <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
            <button className="btn btn-outline-success"
type="submit">Search</button>
          </form> : "No Search Bar"}
```


**Now usage prop-types**

npm install --save prop-types

**Header.js**

```
import React from 'react'
import PropTypes from 'prop-types';

export default function Header(props) {
  return (
    <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="container-fluid">
        <a className="navbar-brand" href="#">{props.title} {props.myvar}</a>
```

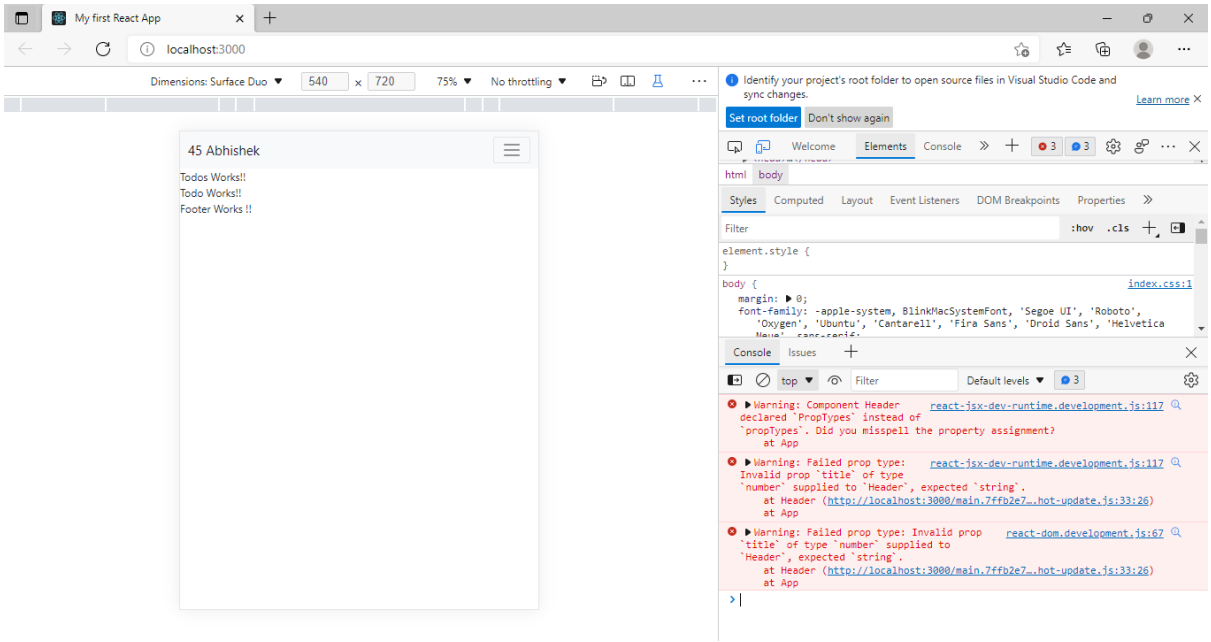
```
        <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span className="navbar-toggler-icon"></span>
        </button>
        <div className="collapse navbar-collapse" id="navbarSupportedContent">
          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">Home</a>
            </li>
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">About</a>
            </li>

          </ul>
         {props.searchBar ?   <form className="d-flex">
            <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
            <button className="btn btn-outline-success"
type="submit">Search</button>
          </form> : "No Search Bar"}
        </div>
      </div>
    </nav>
  )
}

Header.propTypes={
  title : PropTypes.string
  }
```

**Now if you pass any variable apart from the datatype you have specified it will give error.**

**App.js**

```
import logo from './logo.svg';
import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { TodoItem } from './MyComponents/TodoItem';
import { Todos } from './MyComponents/Todos';

function App() {
  return (
```

```
    <>
      <Header title={45} myvar="Abhishek" searchBar={false}/>
      <Todos/>
      <TodoItem/>
      <Footer/>
    </>
  );
}

export default App;
```




# How to set default values

## Header.js

```
Header.defaultProps={
  title: "Yours Title Here",
  searchBar: true
}
Header.propTypes={
  title : PropTypes.string,
  searchBar : PropTypes.bool
  }
```

**App.js**

```
import logo from './logo.svg';
import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { TodoItem } from './MyComponents/TodoItem';
import { Todos } from './MyComponents/Todos';

function App() {
  return (
    <>
      <Header  myvar="Abhishek" searchBar={false}/>
      <Todos/>
      <TodoItem/>
      <Footer/>
    </>
  );
}

export default App;
```

Note: Even you have not passing the title, it's showing form default.


# How to set property is required

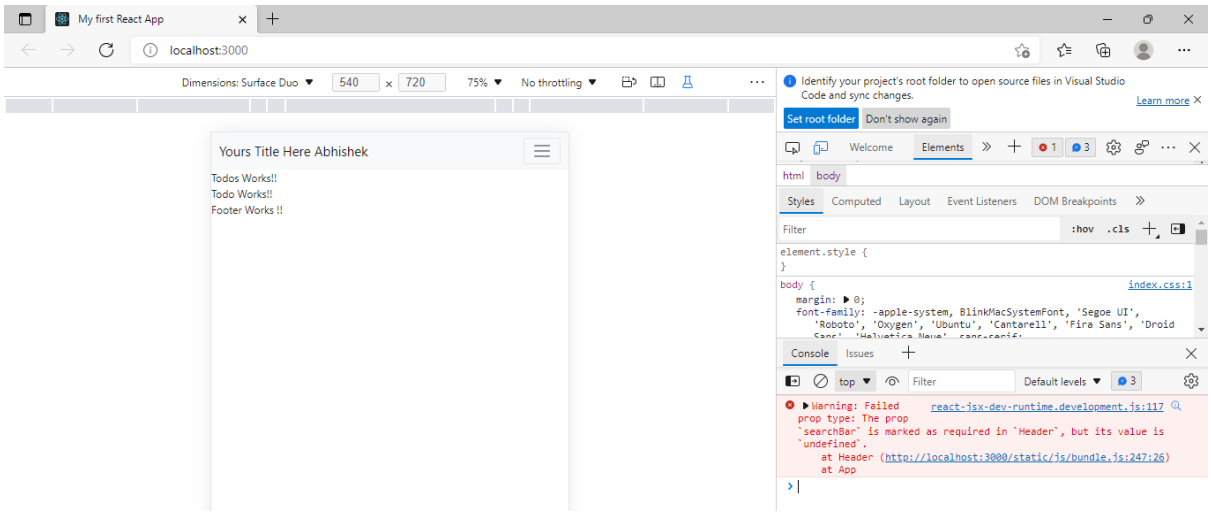
**Header.js**

```
Header.defaultProps={
  title: "Yours Title Here",
  //searchBar: true
}
Header.propTypes={
  title : PropTypes.string,
  searchBar : PropTypes.bool.isRequired
  }
```

**App.js**

```
 import logo from './logo.svg';

import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { TodoItem } from './MyComponents/TodoItem';
import { Todos } from './MyComponents/Todos';

function App() {
  return (
    <>
      <Header  myvar="Abhishek" />
      <Todos/>
      <TodoItem/>
      <Footer/>
    </>
  );
}

export default App;
```

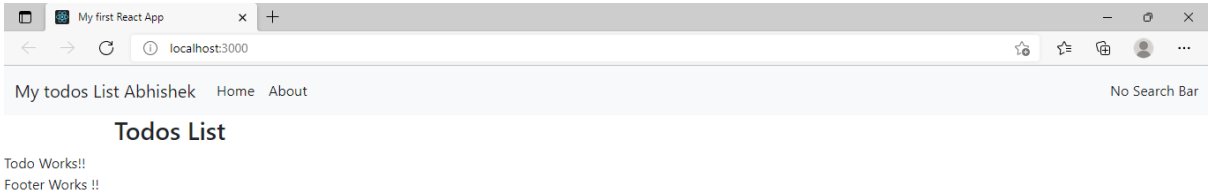
Note: if you will not pass that attribute or property , it will show you error. It is mandate to add



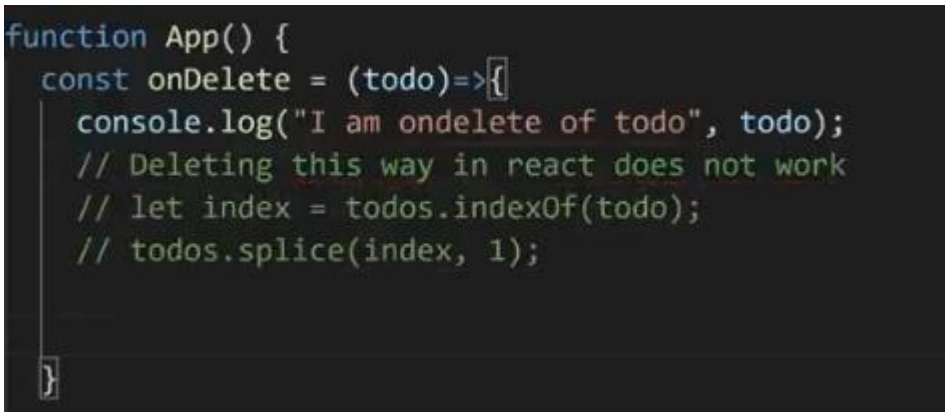

**Now let's have some changes in the Todos.js, this time I have to display list of todos, as follows.**

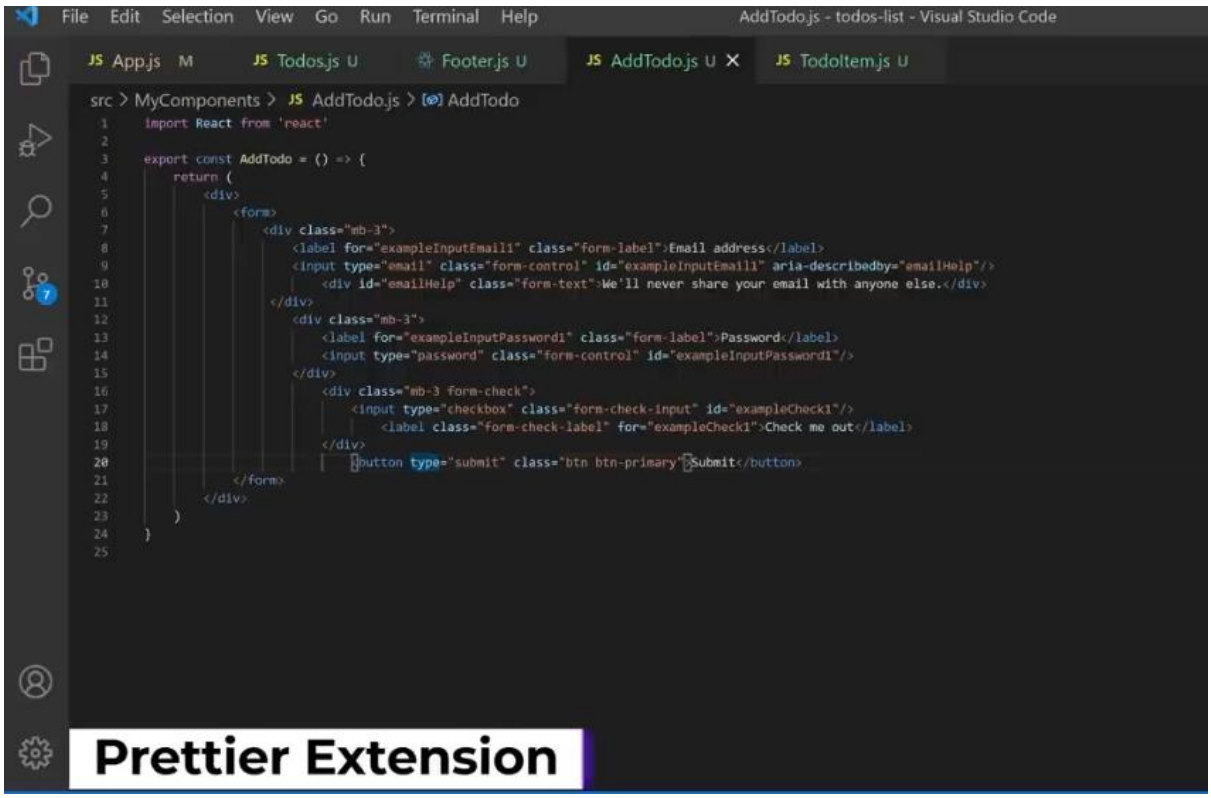
**Todos.js**

```
import React from 'react'

export const Todos = () => {
    return (
        // <div>
        //     Todos Works!!
        // </div>
        <div className="container">
            <h3>Todos List</h3>
        </div>
    )
}
```




## Now let's create some todo list in the app.js file which we are going to display in the todos.js




File Edit Selection View Go Run Terminal Help
AddTodo.js - todos-list - Visual Studio Code
App.js M
Todos.js U
Footer.js U
AddTodo.js U
TodoItem.js U
src > MyComponents > AddTodo.js > AddTodo
import React from 'react'
export const AddTodo = () => {
return (

<form>

<label for="exampleInputEmail1" class="form-label">Email address</label>
<input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp"/>
We'll never share your email with anyone else.

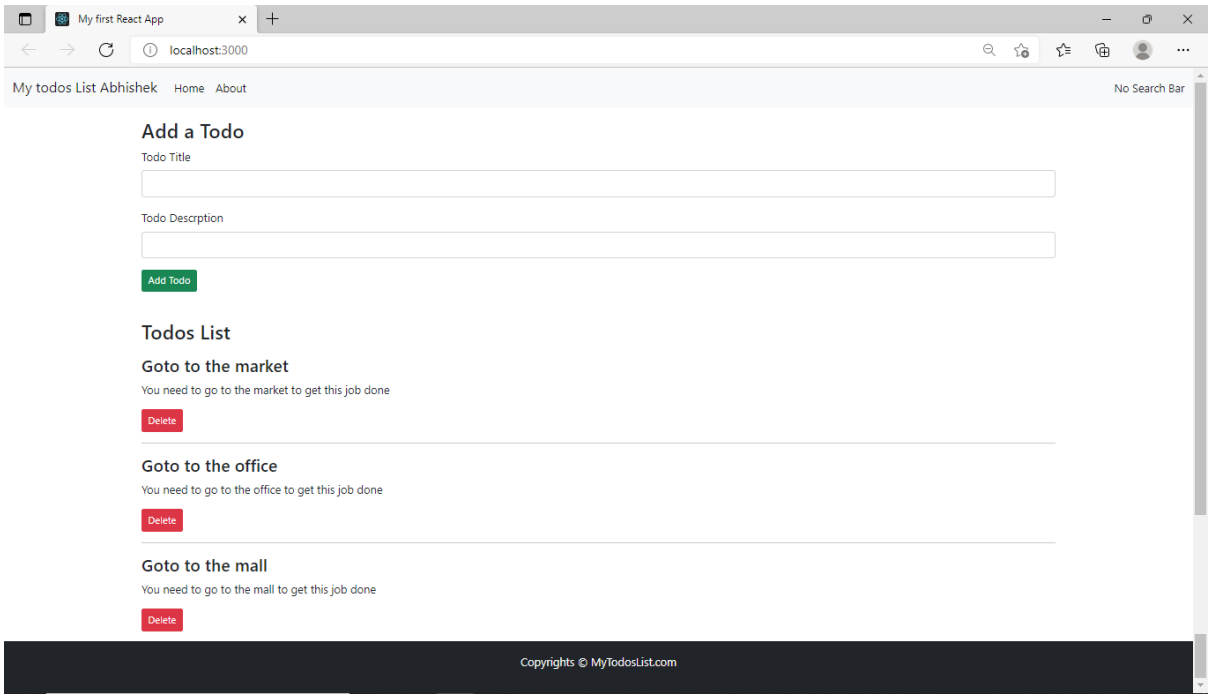
<label for="exampleInputPassword1" class="form-label">Password</label>
<input type="password" class="form-control" id="exampleInputPassword1"/>

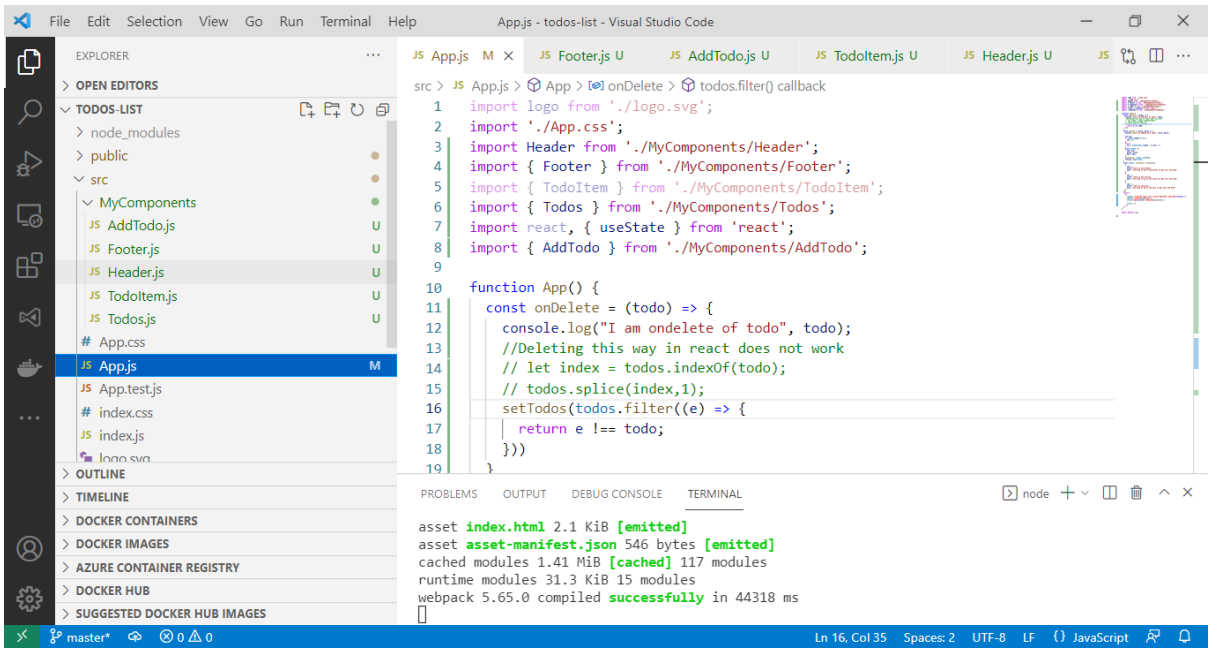
<input type="checkbox" class="form-check-input" id="exampleCheck1"/>
<label class="form-check-label" for="exampleCheck1">Check me out</label>

<button type="submit" class="btn btn-primary">Submit</button>
</form>

)
}
Prettier Extension

## Now this is the updated code.



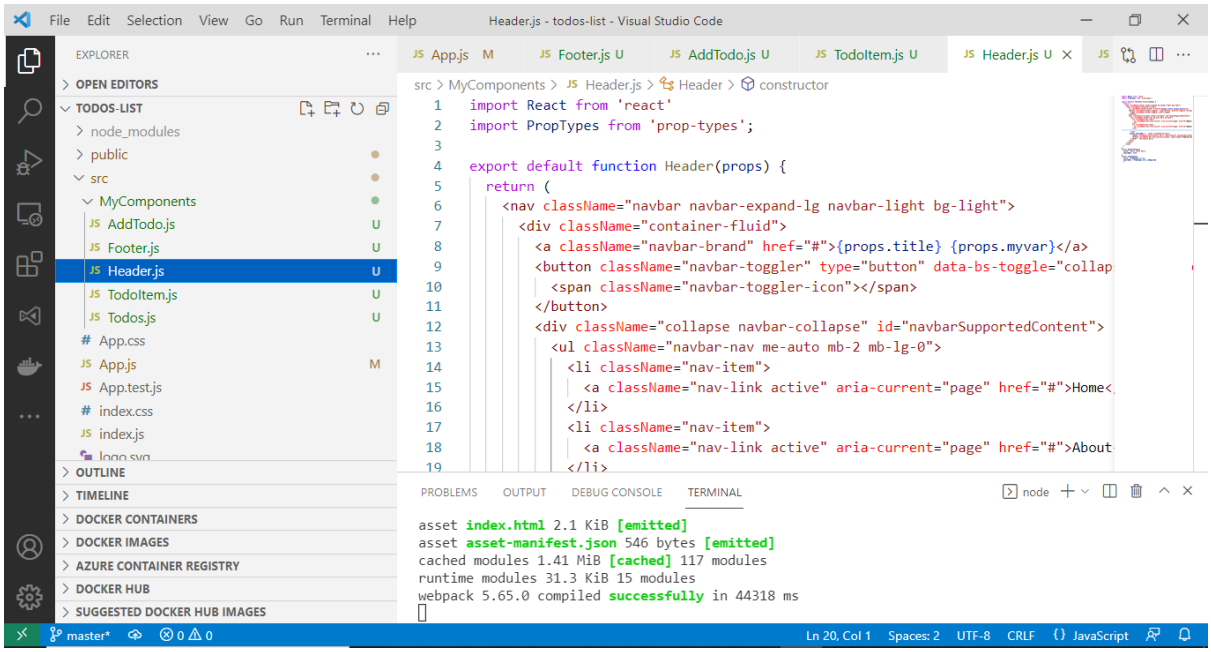




## App.js

```
import logo from './logo.svg';
import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { TodoItem } from './MyComponents/TodoItem';
import { Todos } from './MyComponents/Todos';
```

```
import react, { useState } from 'react';
import { AddTodo } from './MyComponents/AddTodo';

function App() {
  const onDelete = (todo) => {
    console.log("I am ondelete of todo", todo);
    //Deleting this way in react does not work
    // let index = todos.indexOf(todo);
    // todos.splice(index,1);
    setTodos(todos.filter((e) => {
      return e !== todo;
    }))
  }
  const addTodo = (title, desc) => {
    console.log("I am adding this todo", title, desc);

    let sno;
    if (todos.length == 0) {
      sno = 1;
    }
    else {
      sno = todos[todos.length - 1].sno + 1;
    }
    const myTodo = {
      sno: sno,
      title: title,
      desc: desc,
    }
    setTodos([...todos, myTodo]);
    console.log(myTodo);
  }
  const [todos, setTodos] = useState([
    {
      sno: 1,
      title: "Goto to the market",
      desc: "You need to go to the market to get this job done"
    },
    {
      sno: 2,
      title: "Goto to the office",
      desc: "You need to go to the office to get this job done"
    },
    {
      sno: 3,
      title: "Goto to the mall",
      desc: "You need to go to the mall to get this job done"
    },
  ]);
```

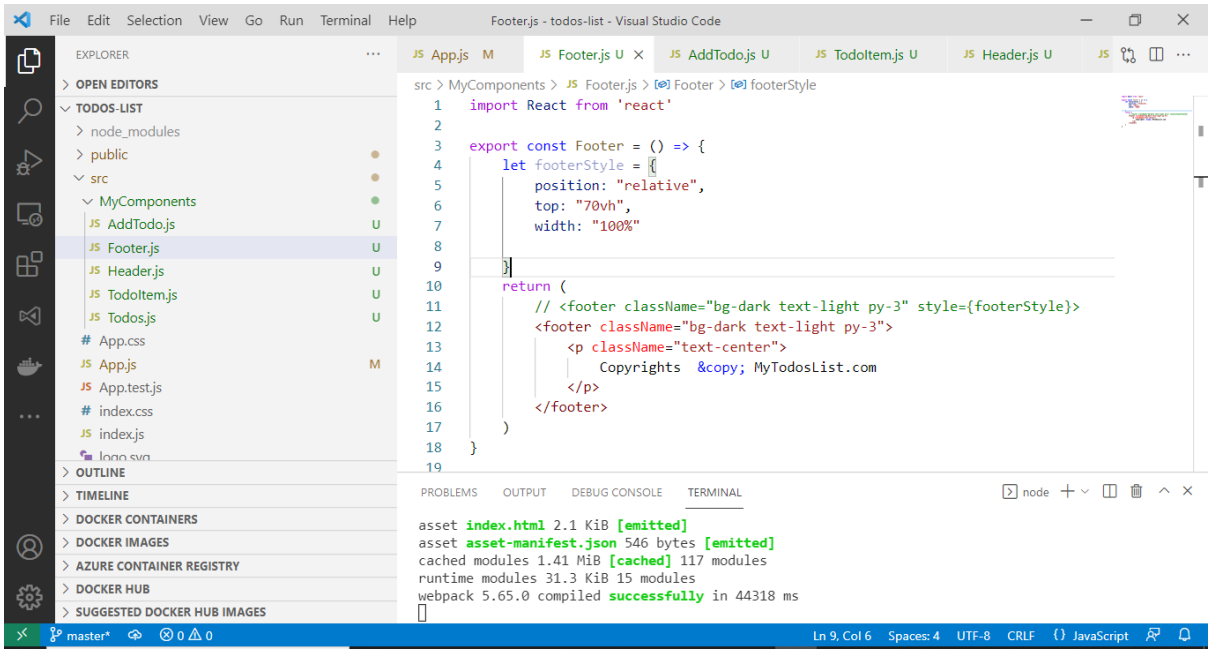
```
  return (
    <>
      <Header title="My todos List" myvar="Abhishek" searchBar={false} />
      <AddTodo addTodo={addTodo} />
      <Todos todos={todos} onDelete={onDelete} />

      <Footer />
    </>
  );
}

export default App;
```




## Header.js

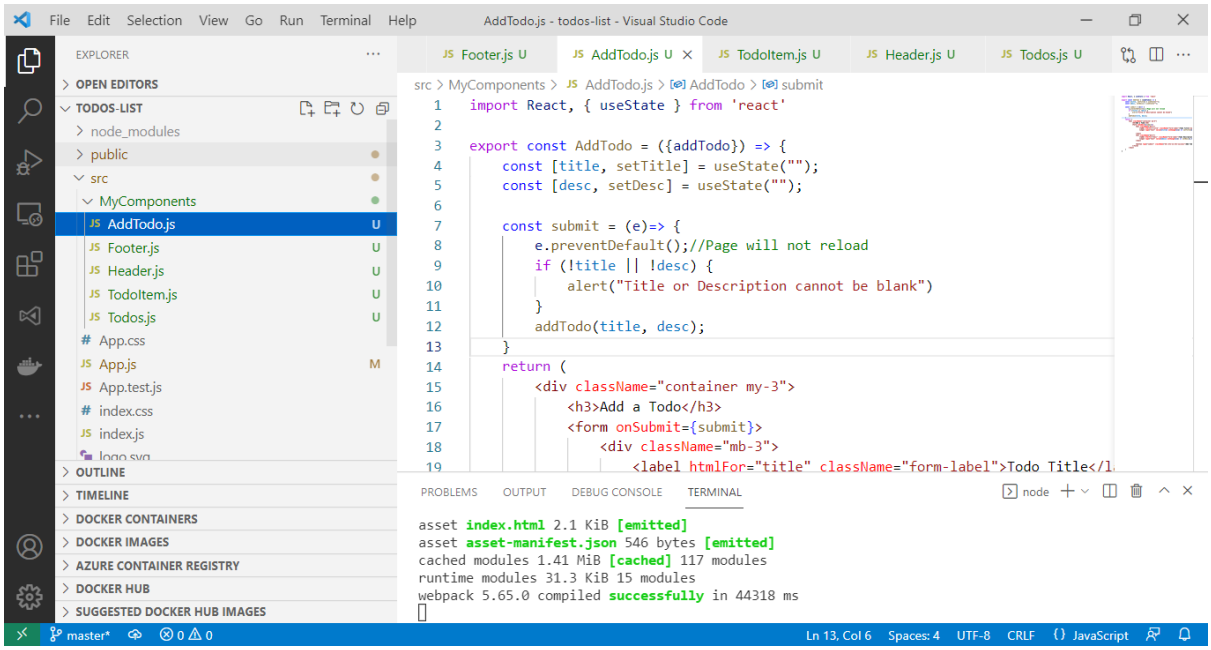
```
import React from 'react'
import PropTypes from 'prop-types';

export default function Header(props) {
  return (
    <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="container-fluid">
        <a className="navbar-brand" href="#">{props.title} {props.myvar}</a>
        <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span className="navbar-toggler-icon"></span>
        </button>
```

```
        <div className="collapse navbar-collapse" id="navbarSupportedContent">
          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">Home</a>
            </li>
            <li className="nav-item">
              <a className="nav-link active" aria-current="page"
href="#">About</a>
            </li>

          </ul>
         {props.searchBar ?   <form className="d-flex">
            <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
            <button className="btn btn-outline-success"
type="submit">Search</button>
          </form> : "No Search Bar"}
        </div>
      </div>
    </nav>
  )
}
Header.defaultProps={
  title: "Yours Title Here",
  searchBar: true
}
Header.propTypes={
  title : PropTypes.string,
  searchBar : PropTypes.bool.isRequired
  }
```




## Footer.js

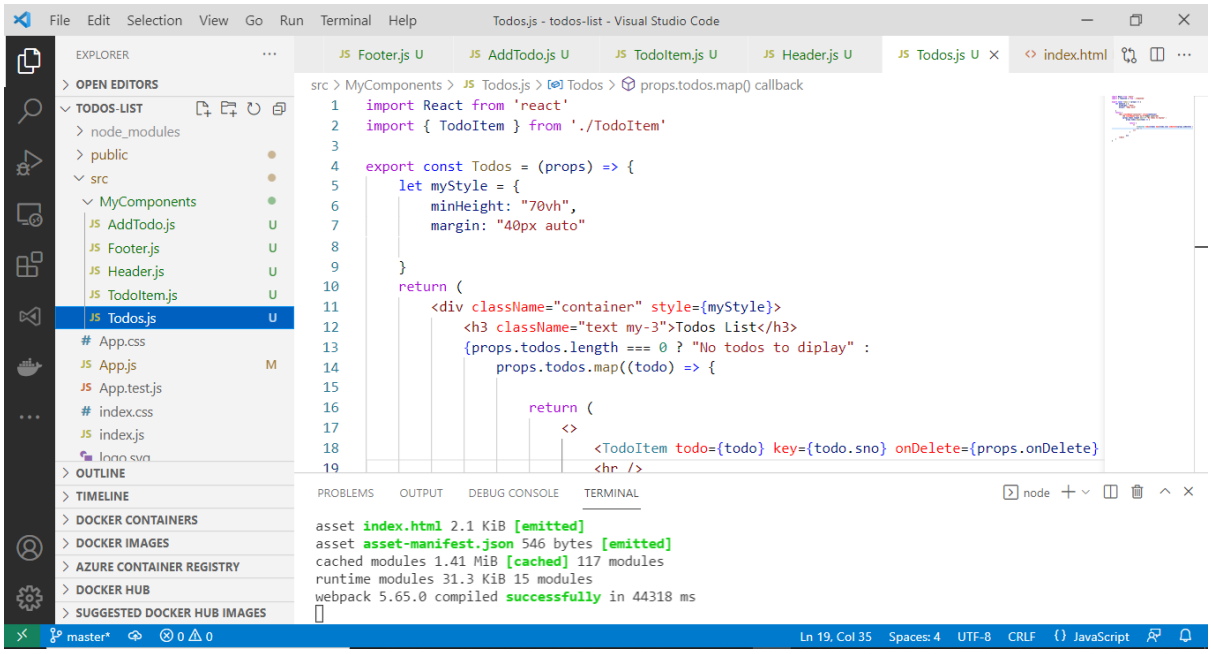
```
import React from 'react'

export const Footer = () => {
    let footerStyle = {
        position: "relative",
        top: "70vh",
        width: "100%"

    }
    return (
        // <footer className="bg-dark text-light py-3" style={footerStyle}>
        <footer className="bg-dark text-light py-3">
            <p className="text-center">
                Copyrights  &copy; MyTodosList.com
            </p>
        </footer>
    )
}
```




## AddTodo.js

```
import React, { useState } from 'react'

export const AddTodo = ({addTodo}) => {
    const [title, setTitle] = useState("");
    const [desc, setDesc] = useState("");

    const submit = (e)=> {
        e.preventDefault();//Page will not reload
        if (!title || !desc) {
            alert("Title or Description cannot be blank")
        }
        addTodo(title, desc);
    }
    return (
        <div className="container my-3">
            <h3>Add a Todo</h3>
            <form onSubmit={submit}>
                <div className="mb-3">
                    <label htmlFor="title" className="form-label">Todo
Title</label>
                    <input type="text" value={title} onChange={(e) =>
setTitle(e.target.value)} className="form-control" id="title" aria-
describedby="TodoItem" />

                </div>
                <div className="mb-3">
```

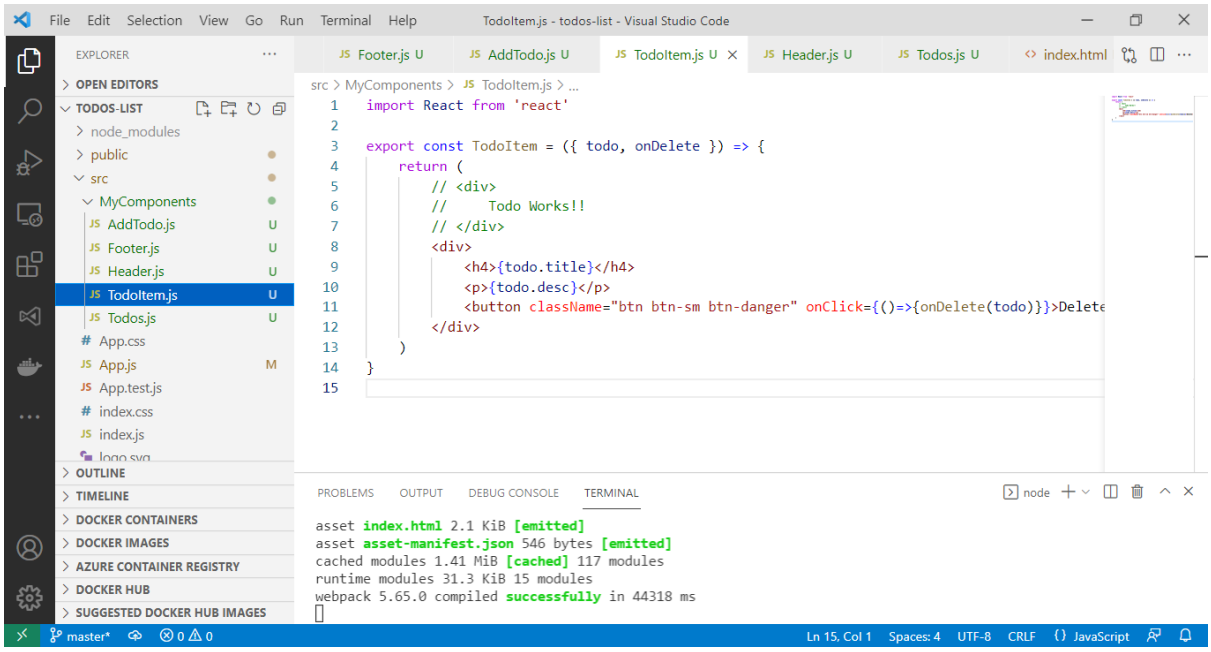
```
                    <label htmlFor="desc" className="form-label">Todo
Descrption</label>
                    <input type="text" value={desc} onChange={(e) =>
setDesc(e.target.value)} className="form-control" id="desc" />
                </div>

                <button type="submit" className="btn btn-sm btn-success">Add
Todo</button>
            </form>
        </div>
    )
}
```




## Todos.js

```
import React from 'react'
import { TodoItem } from './TodoItem'

export const Todos = (props) => {
    let myStyle = {
        minHeight: "70vh",
        margin: "40px auto"

    }
    return (
        <div className="container" style={myStyle}>
            <h3 className="text my-3">Todos List</h3>
            {props.todos.length === 0 ? "No todos to diplay" :
                props.todos.map((todo) => {
```

```
                return (
                    <>
                        <TodoItem todo={todo} key={todo.sno}
onDelete={props.onDelete} />
                        <hr />
                    </>
                )

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



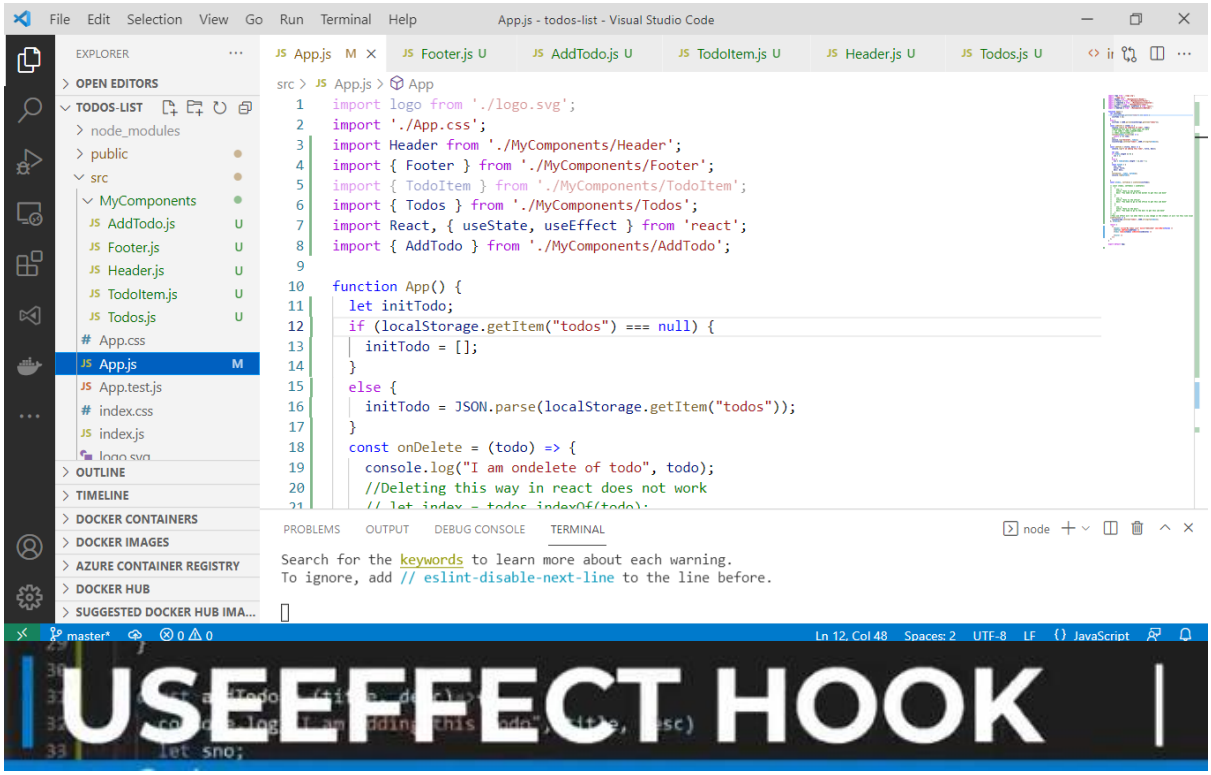

## TodoItem.js

```
import React from 'react'

export const TodoItem = ({ todo, onDelete }) => {
    return (
        // <div>
        //     Todo Works!!
        // </div>
        <div>
            <h4>{todo.title}</h4>
            <p>{todo.desc}</p>
            <button className="btn btn-sm btn-danger"
onClick={()=>{onDelete(todo)}}>Delete</button>
        </div>
```

```
    )
}
```

# NOW LET'S MAKE CHANGES, HOLDING THE STATE

- **Add todo's list in the Local storage, so after reload, the list shouldn't be reinitializing.**



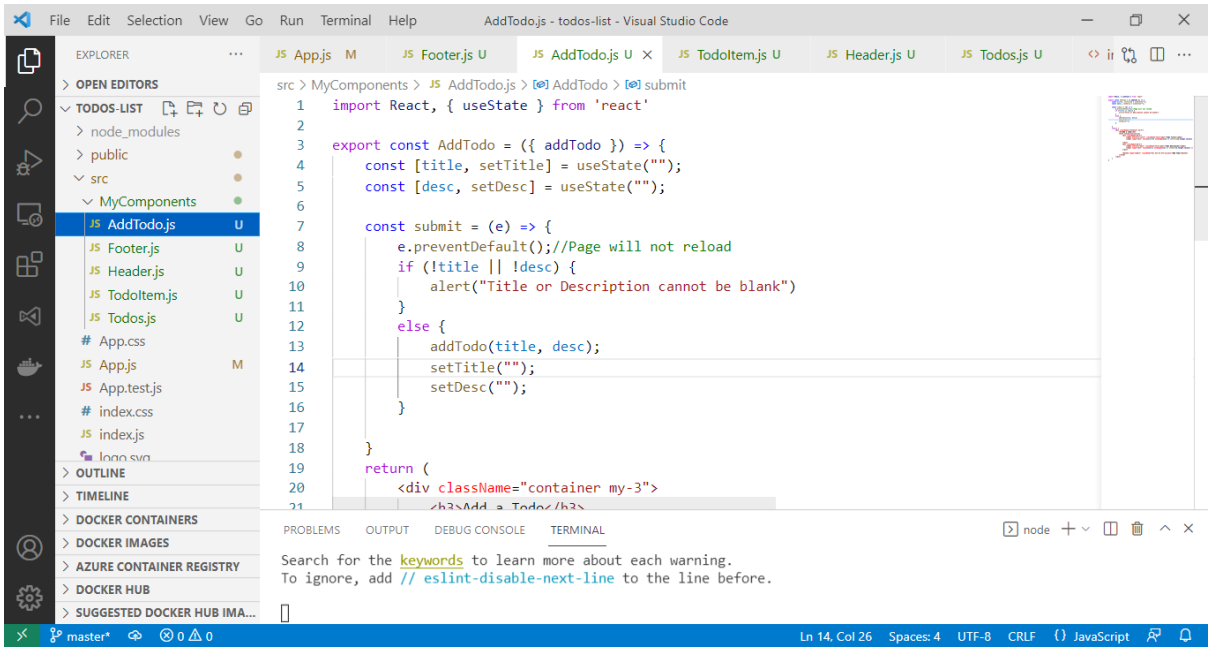

```
import React, { useState, useEffect } from 'react';
```

**App.js**

```
import logo from './logo.svg';
import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { TodoItem } from './MyComponents/TodoItem';
import { Todos } from './MyComponents/Todos';
import React, { useState, useEffect } from 'react';
import { AddTodo } from './MyComponents/AddTodo';

function App() {
  let initTodo;
  if (localStorage.getItem("todos") === null) {
    initTodo = [];
```

```
  }
  else {
    initTodo = JSON.parse(localStorage.getItem("todos"));
  }
  const onDelete = (todo) => {
    console.log("I am ondelete of todo", todo);
    //Deleting this way in react does not work
    // let index = todos.indexOf(todo);
    // todos.splice(index,1);
    setTodos(todos.filter((e) => {
      return e !== todo;
    }));
    console.log("deleted", todos);
    localStorage.setItem("todos", JSON.stringify(todos));

  }
  const addTodo = (title, desc) => {
    console.log("I am adding this todo", title, desc);

    let sno;
    if (todos.length == 0) {
      sno = 1;
    }
    else {
      sno = todos[todos.length - 1].sno + 1;
    }
    const myTodo = {
      sno: sno,
      title: title,
      desc: desc,
    }
    setTodos([...todos, myTodo]);
    console.log(myTodo);


  }
  const [todos, setTodos] = useState(initTodo);

  // const [todos, setTodos] = useState([
  //   {
  //     sno: 1,
  //     title: "Goto to the market",
  //     desc: "You need to go to the market to get this job done"
  //   },
  //   {
  //     sno: 2,
  //     title: "Goto to the office",
  //     desc: "You need to go to the office to get this job done"
  //   },
```

```
  //   {
  //     sno: 3,
  //     title: "Goto to the mall",
  //     desc: "You need to go to the mall to get this job done"
  //   },
  // ]);
  //This use effect will run when there is any change in the [todos] it will run this line localStorage.setItem("todos", JSON.stringify(todos));
  useEffect(() => {
    localStorage.setItem("todos", JSON.stringify(todos));
  }, [todos]);

  return (
    <>
      <Header title="My todos List" myvar="Abhishek" searchBar={false} />
      <AddTodo addTodo={addTodo} />
      <Todos todos={todos} onDelete={onDelete} />

      <Footer />
    </>
  );
}

export default App;
```




## AddTodo.js

```
import React, { useState } from 'react'

export const AddTodo = ({ addTodo }) => {
```

```
    const [title, setTitle] = useState("");
    const [desc, setDesc] = useState("");

    const submit = (e) => {
        e.preventDefault();//Page will not reload
        if (!title || !desc) {
            alert("Title or Description cannot be blank")
        }
        else {
            addTodo(title, desc);
            setTitle("");
            setDesc("");
        }

    }
    return (
        <div className="container my-3">
            <h3>Add a Todo</h3>
            <form onSubmit={submit}>
                <div className="mb-3">
                    <label htmlFor="title" className="form-label">Todo
Title</label>
                    <input type="text" value={title} onChange={(e) =>
setTitle(e.target.value)} className="form-control" id="title" aria-
describedby="TodoItem" />

                </div>
                <div className="mb-3">
                    <label htmlFor="desc" className="form-label">Todo
Descrption</label>
                    <input type="text" value={desc} onChange={(e) =>
setDesc(e.target.value)} className="form-control" id="desc" />
                </div>

                <button type="submit" className="btn btn-sm btn-success">Add
Todo</button>
            </form>
        </div>
    )
}
```


**Very important if you are following old routing approach with switch routing which in version 5.2.0, please follow the below steps**

**https://stackoverflow.com/questions/63124161/attempted-import-error-switch-is-not-exported-from-react-router-dom**

**So I uninstall the version 6 of react-router-dom**

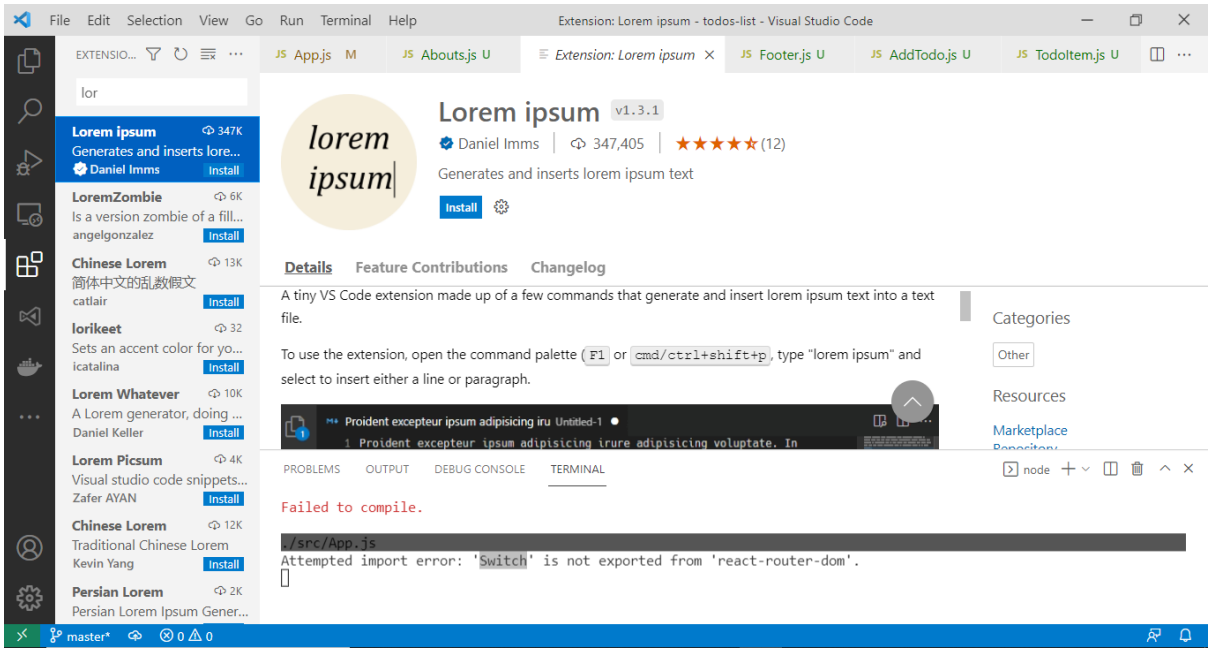
npm uninstall react-router-dom

**And installed version 5.2.0 of react-router-dom**

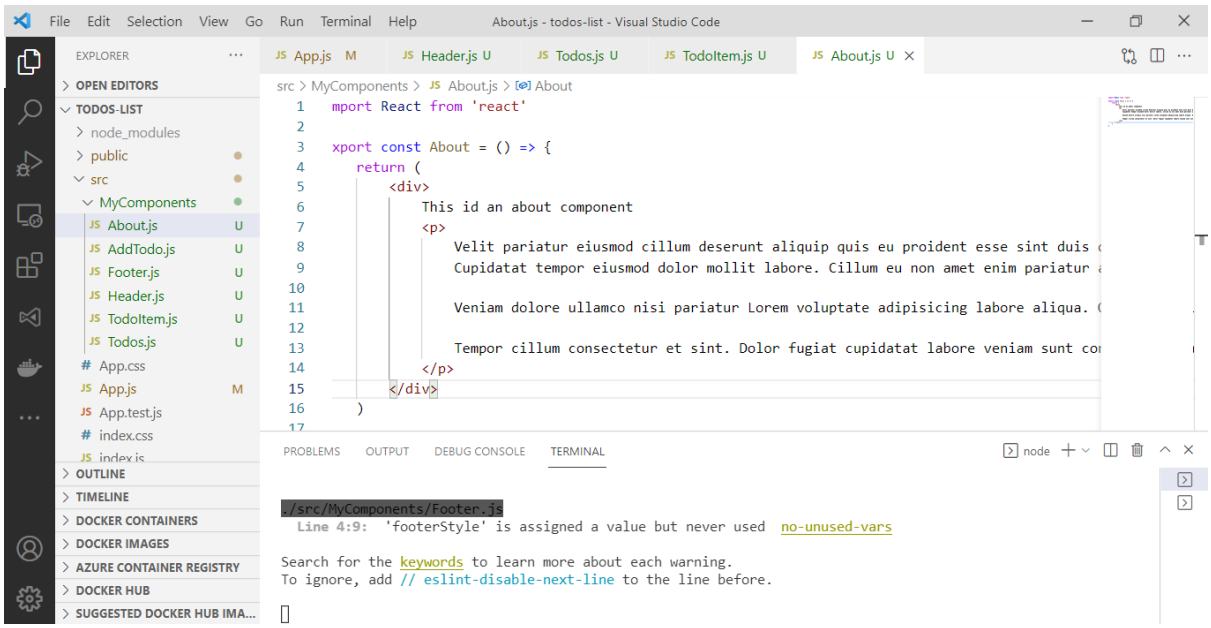
npm install react-router-dom@5.2.0

**If newer version**

npm install react-router-dom



## Now create About component and add some dummy content with the help of Lorem

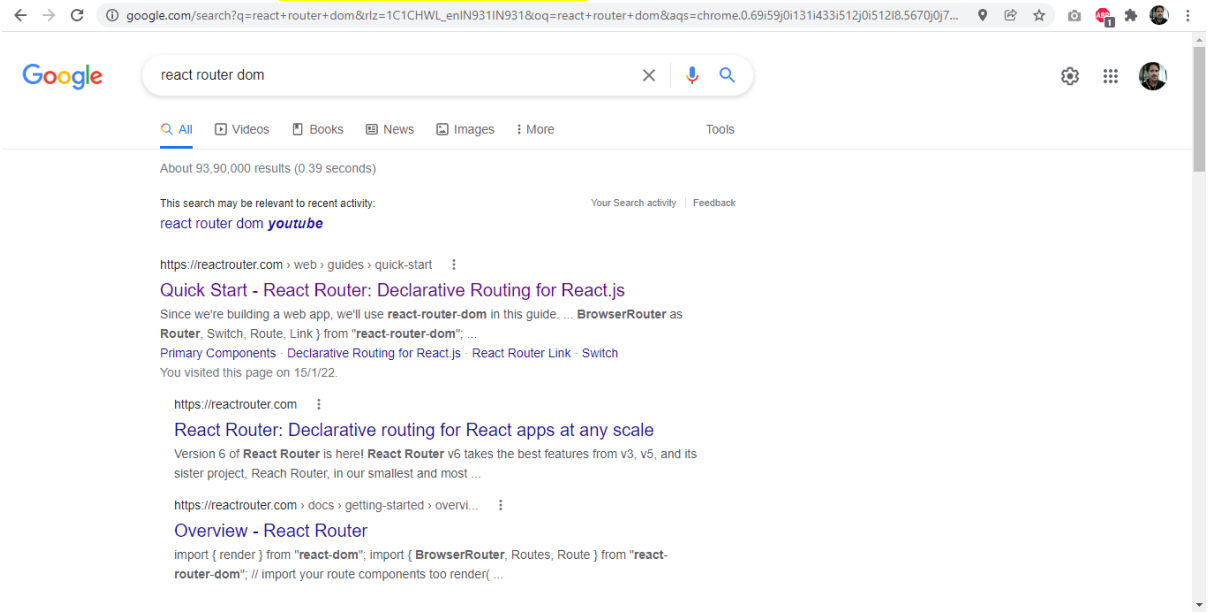



## About

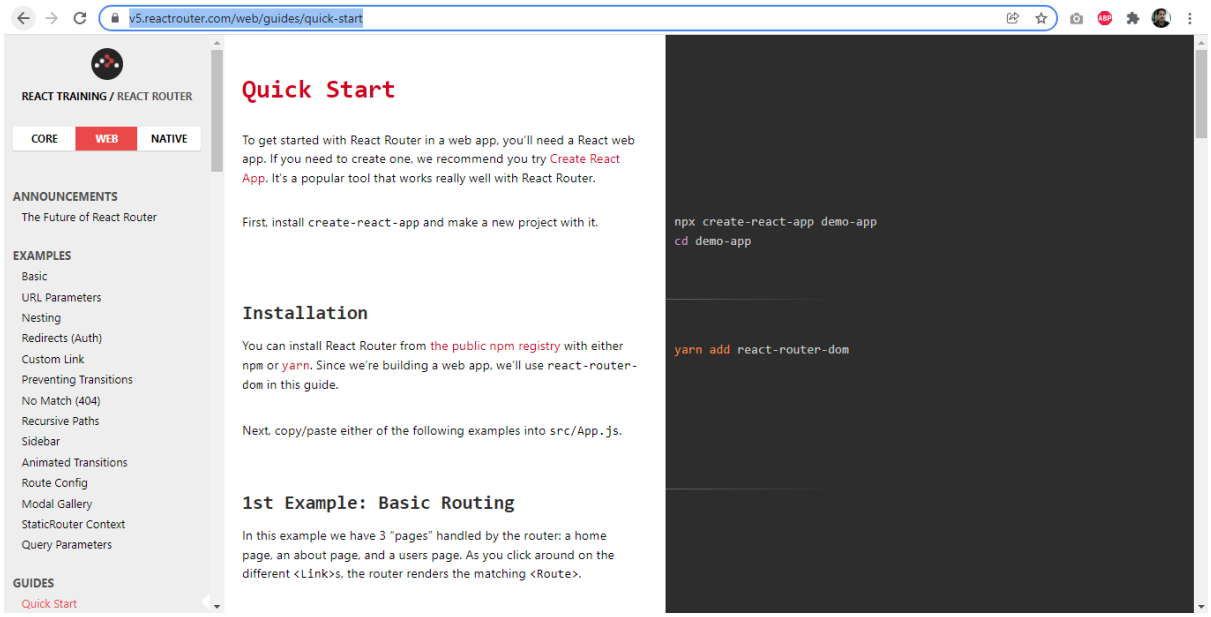
```
import React from 'react'

export const About = () => {
    return (
        <div>
            This id an about component
            <p>
                Velit pariatur eiusmod cillum deserunt aliquip quis eu proident esse sint duis do deserunt aliqua. Mollit veniam sint consectetur eu. Consectetur qui amet veniam excepteur sint ad ut sit enim aliquip culpa consequat do Lorem.
                Cupidatat tempor eiusmod dolor mollit labore. Cillum eu non amet enim pariatur ad aute. Eu mollit et esse Lorem ea culpa proident eu consectetur cillum est laborum. Enim ullamco minim nisi dolor mollit ullamco sit exercitation qui aute ipsum. Cupidatat mollit ut laboris irure laborum ad laborum mollit ut dolor eiusmod. Duis dolor voluptate et nulla veniam exercitation nisi deserunt.

                Veniam dolore ullamco nisi pariatur Lorem voluptate adipisicing labore aliqua. Officia eu laboris eiusmod sint amet dolor elit do ea est laborum. Id aute Lorem laborum nulla deserunt nisi et adipisicing velit minim ipsum ipsum laborum. Irure minim sunt dolore veniam commodo amet sunt.
```

```
                Tempor cillum consectetur et sint. Dolor fugiat cupidatat
labore veniam sunt consequat in anim. Magna voluptate excepteur qui excepteur
adipisicing laboris consectetur. Ea irure duis incididunt ullamco aute
consequat adipisicing elit sit labore veniam ea ut. Ad Lorem elit velit ex
dolor consectetur. Anim dolor proident qui excepteur officia. Officia pariatur
excepteur culpa do ullamco exercitation do mollit laborum.
            </p>
        </div>
    )
}
```

## Now search for react-router-dom




https://v5.reactrouter.com/web/guides/quick-start



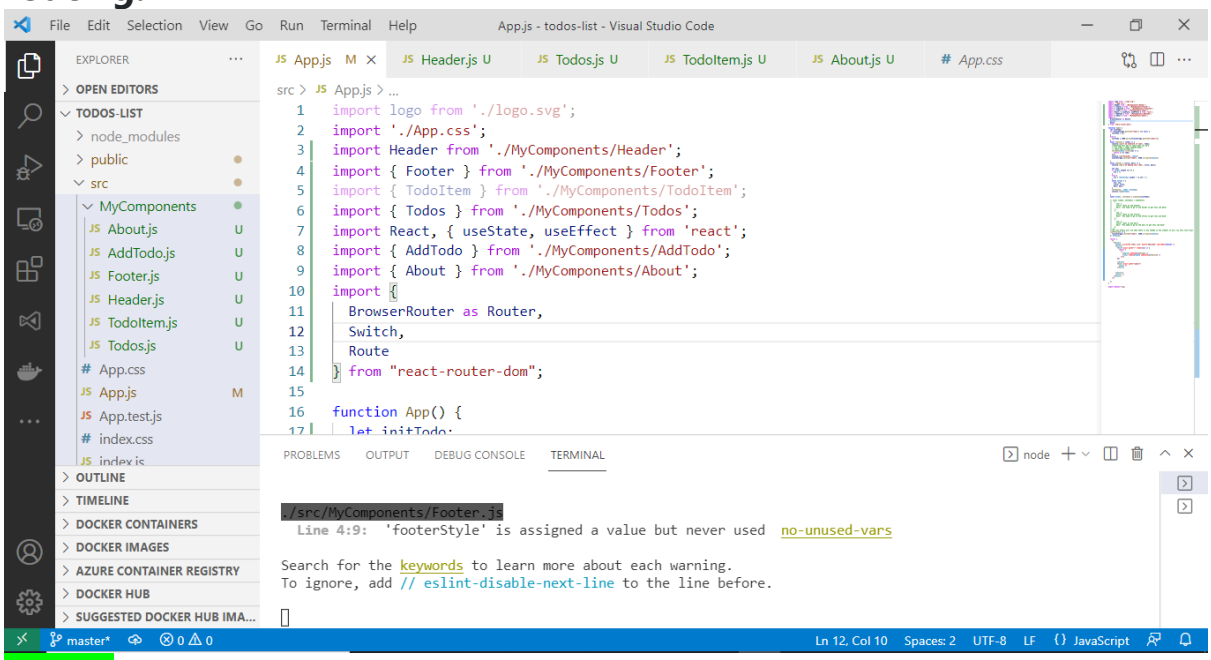

Following the guide lines, copy and paste the following in the **app.js**

```
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Link
} from "react-router-dom";
```

## Now wrap up the component with route tag and put switch tag for conditional rendering.

```
{/* A <Switch> looks through its children <Route>s and
            renders the first one that matches the current URL. */}
        <Switch>
          <Route path="/about">
            <About />
          </Route>
          <Route path="/users">
            <Users />
          </Route>
          <Route path="/">
            <Home />
          </Route>
        </Switch>
      </div>
    </Router>
```

## Now let's import About component in app.js and implement routing.



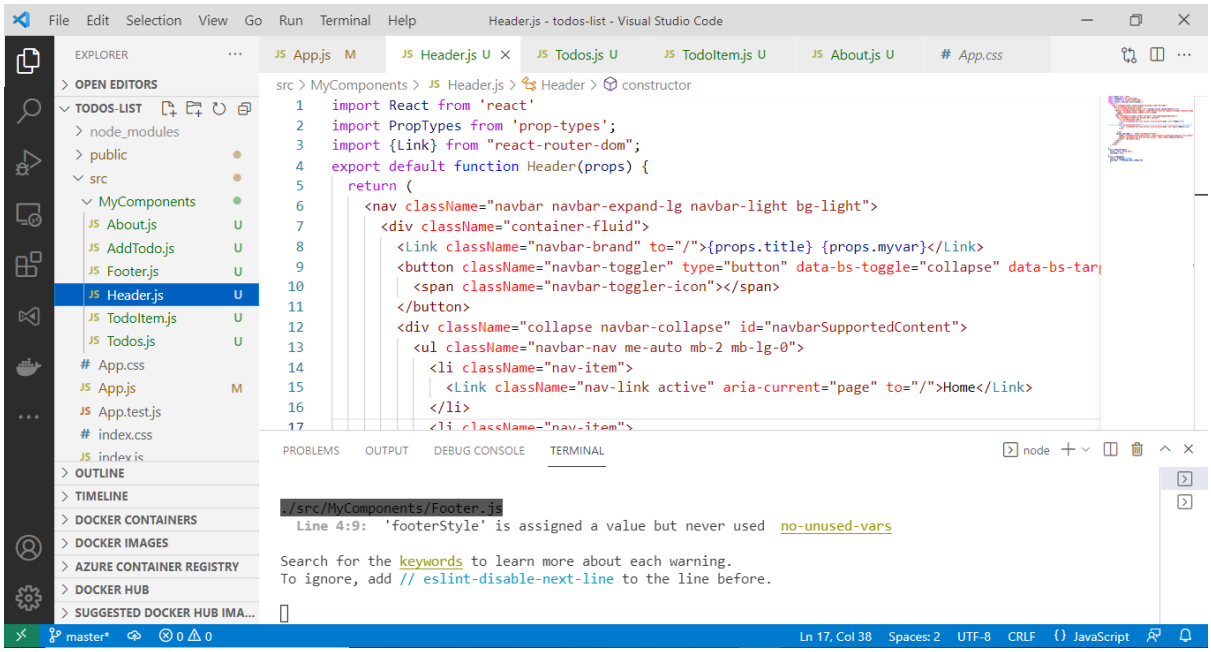

## App.js

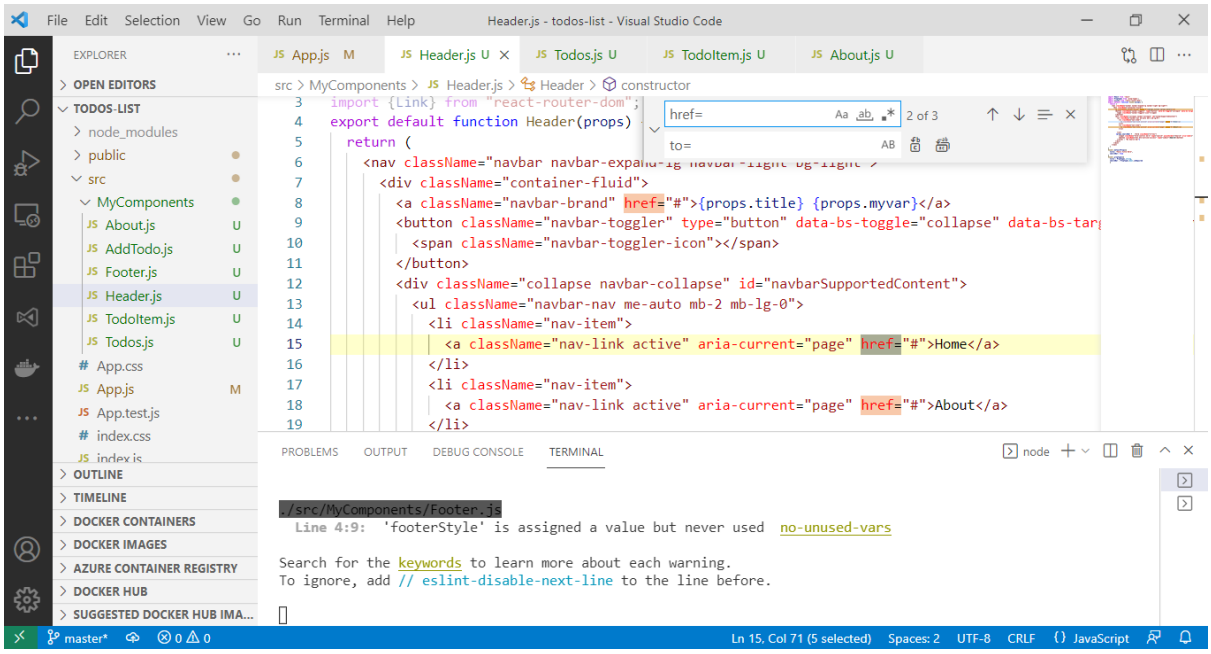
```
import logo from './logo.svg';
import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { TodoItem } from './MyComponents/TodoItem';
import { Todos } from './MyComponents/Todos';
import React, { useState, useEffect } from 'react';
import { AddTodo } from './MyComponents/AddTodo';
import { About } from './MyComponents/About';
import {
  BrowserRouter as Router,
  Switch,
  Route
} from "react-router-dom";

function App() {
  let initTodo;
  if (localStorage.getItem("todos") === null) {
    initTodo = [];
  }
  else {
    initTodo = JSON.parse(localStorage.getItem("todos"));
  }
  const onDelete = (todo) => {
```

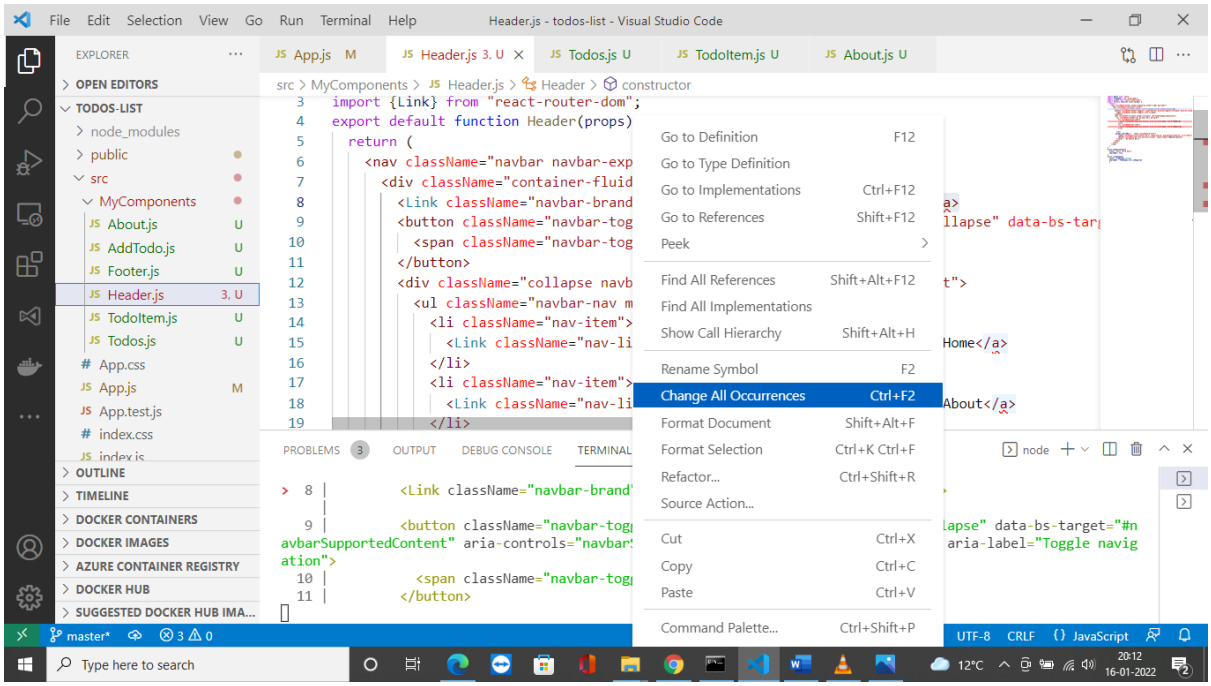
```
    console.log("I am ondelete of todo", todo);
    //Deleting this way in react does not work
    // let index = todos.indexOf(todo);
    // todos.splice(index,1);
    setTodos(todos.filter((e) => {
      return e !== todo;
    }));
    console.log("deleted", todos);
    localStorage.setItem("todos", JSON.stringify(todos));

  }
  const addTodo = (title, desc) => {
    console.log("I am adding this todo", title, desc);

    let sno;
    if (todos.length === 0) {
      sno = 1;
    }
    else {
      sno = todos[todos.length - 1].sno + 1;
    }
    const myTodo = {
      sno: sno,
      title: title,
      desc: desc,
    }
    setTodos([...todos, myTodo]);
    console.log(myTodo);

  }
  const [todos, setTodos] = useState(initTodo);

  // const [todos, setTodos] = useState([
  //   {
  //     sno: 1,
  //     title: "Goto to the market",
  //     desc: "You need to go to the market to get this job done"
  //   },
  //   {
  //     sno: 2,
  //     title: "Goto to the office",
  //     desc: "You need to go to the office to get this job done"
  //   },
  //   {
  //     sno: 3,
  //     title: "Goto to the mall",
  //     desc: "You need to go to the mall to get this job done"
  //   },
```

```
  // ]);
  //This use effect will run when there is any change in the [todos] it will
run this line localStorage.setItem("todos", JSON.stringify(todos));
  useEffect(() => {
    localStorage.setItem("todos", JSON.stringify(todos));
  }, [todos]);

  return (
    <>
      <Router>
        <Header title="My todos List" myvar="Abhishek" searchBar={false} />
        <Switch>
          <Route exact path="/" render={() => {
            return (
              <>
                <AddTodo addTodo={addTodo} />
                <Todos todos={todos} onDelete={onDelete} />
              </>)
          }}>

          </Route>
          <Route exact path="/about">
            <About />
          </Route>


        </Switch>
        <Footer />
      </Router>
    </>

  );
}

export default App;
```

# Now makes the changes in Header.js component



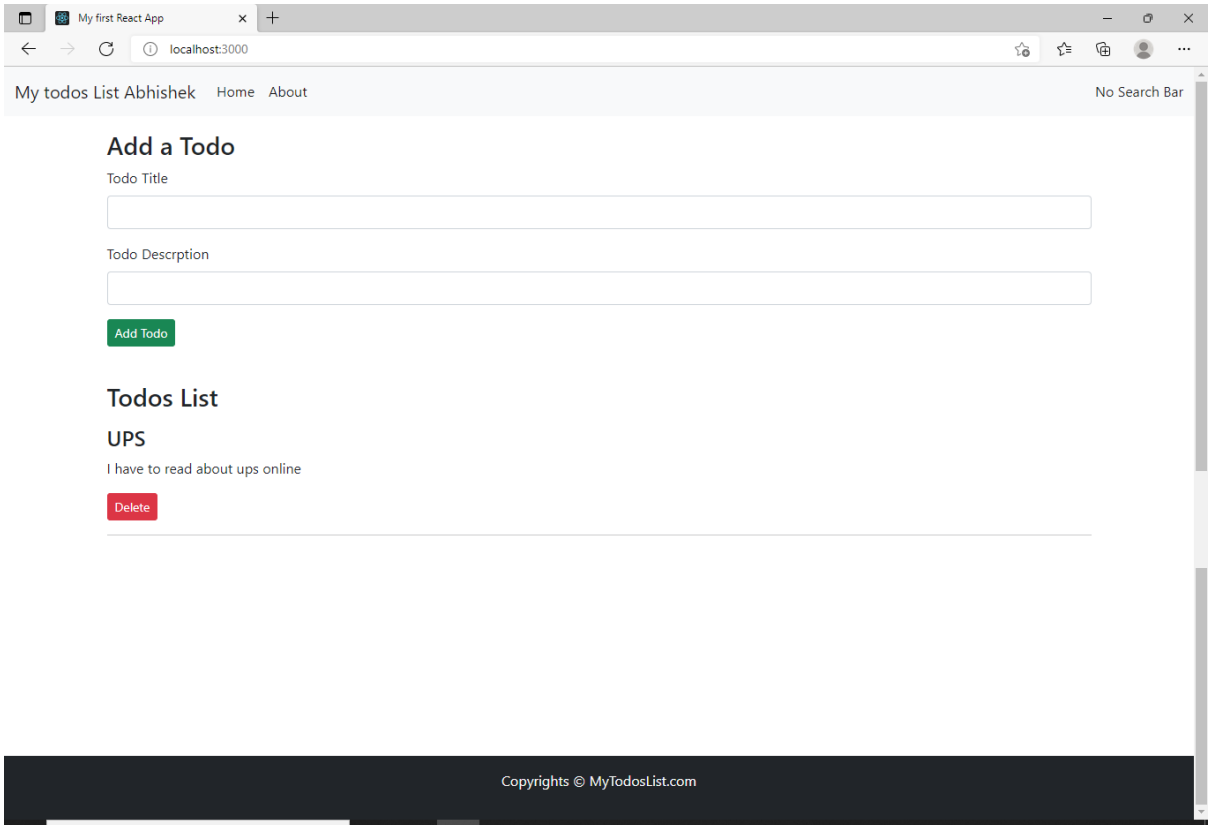



## Header.js

```
import React from 'react'
import PropTypes from 'prop-types';
import {Link} from "react-router-dom";
export default function Header(props) {
  return (
    <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="container-fluid">
        <Link className="navbar-brand" to="/">{props.title}
{props.myvar}</Link>
        <button className="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span className="navbar-toggler-icon"></span>
        </button>
        <div className="collapse navbar-collapse" id="navbarSupportedContent">
          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
            <li className="nav-item">
              <Link className="nav-link active" aria-current="page"
to="/">Home</Link>
            </li>
            <li className="nav-item">
              <Link className="nav-link active" aria-current="page"
to="/about">About</Link>
            </li>

          </ul>
```

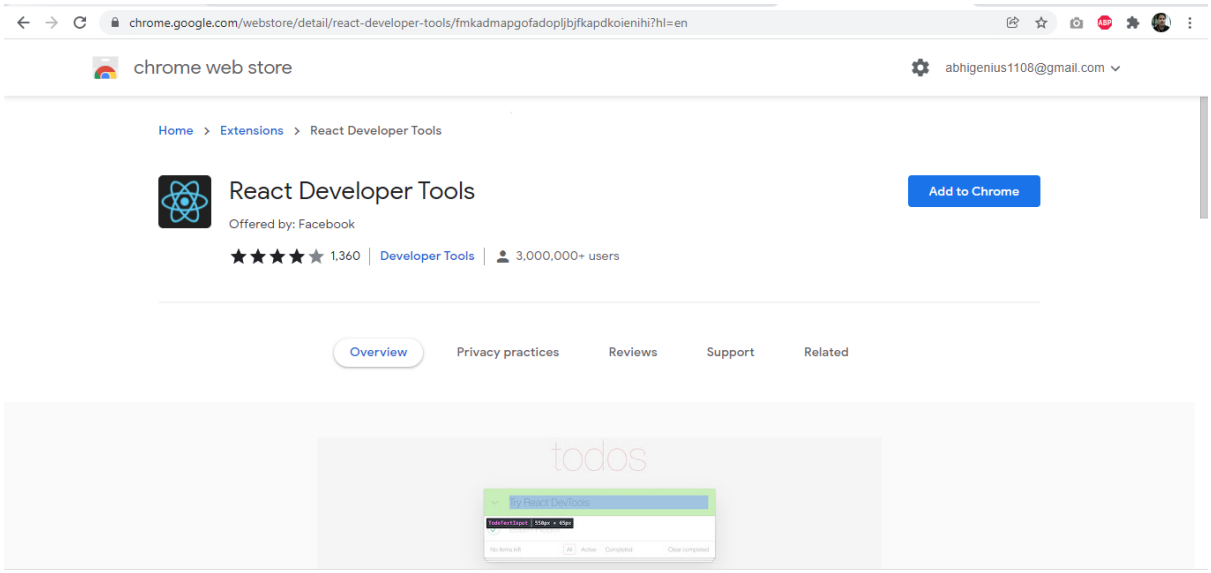
```
        {props.searchBar ?   <form className="d-flex">
           <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
           <button className="btn btn-outline-success"
type="submit">Search</button>
         </form> : "No Search Bar"}
        </div>
      </div>
    </nav>
  )
}
Header.defaultProps={
  title: "Yours Title Here",
  searchBar: true
}
Header.propTypes={
  title : PropTypes.string,
  searchBar : PropTypes.bool.isRequired
  }
```

## Now the final output.
## Home page



## About page

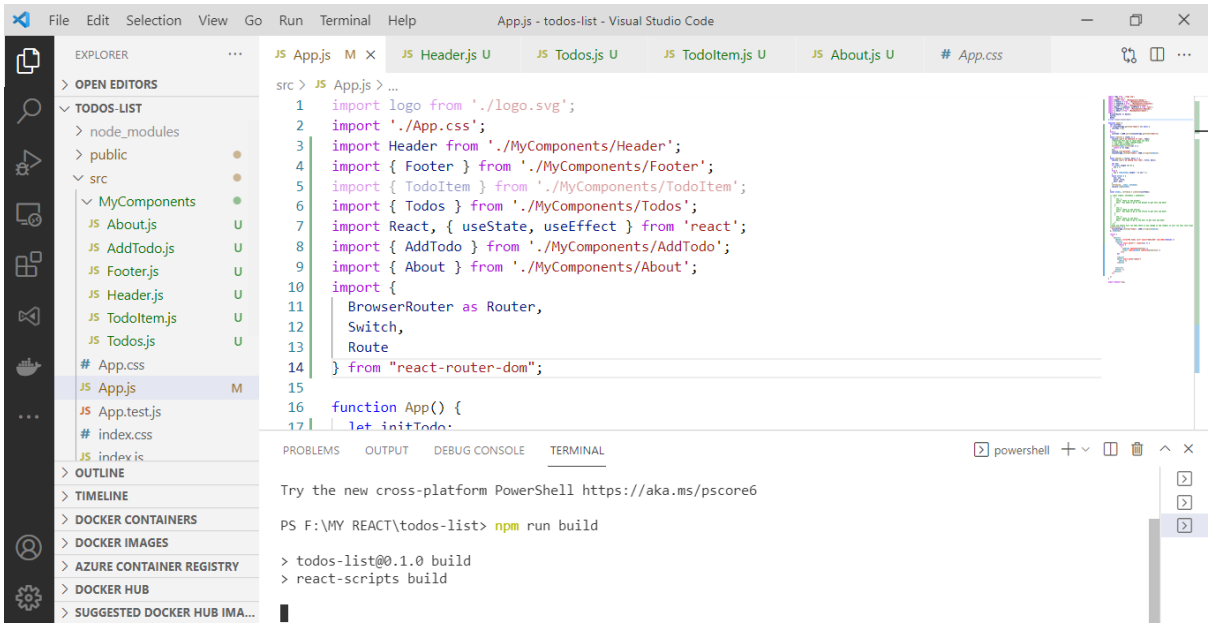

## Now you can also use one extension

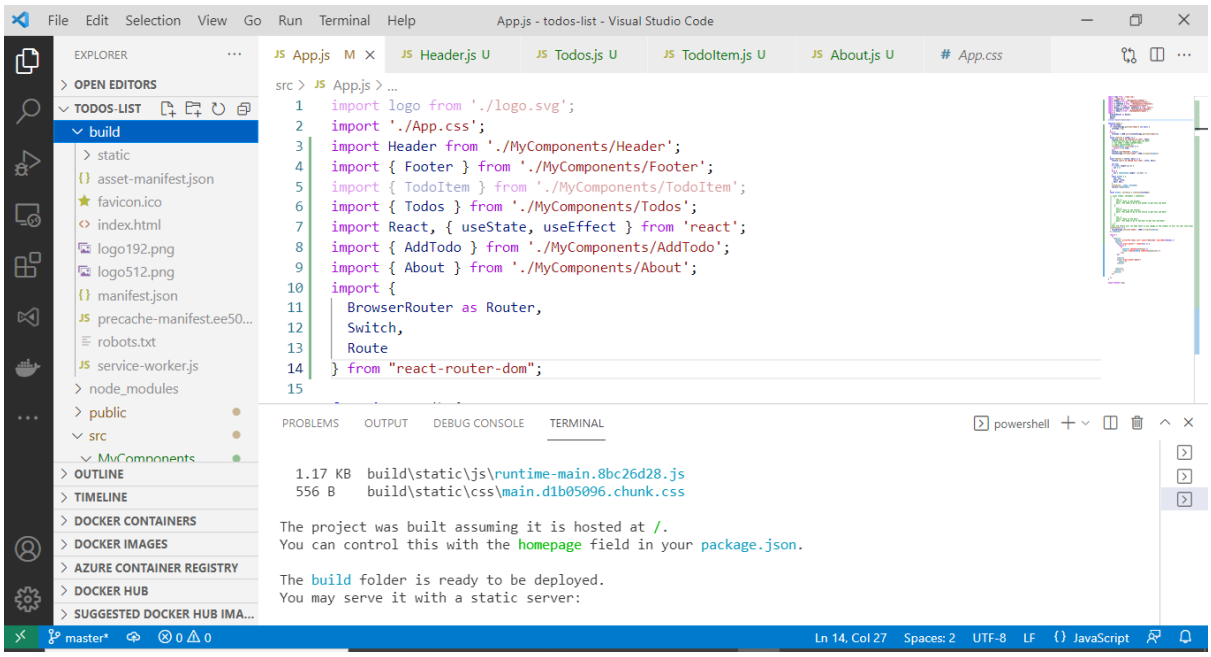



**Now we have to BUILD PROJECT for production**

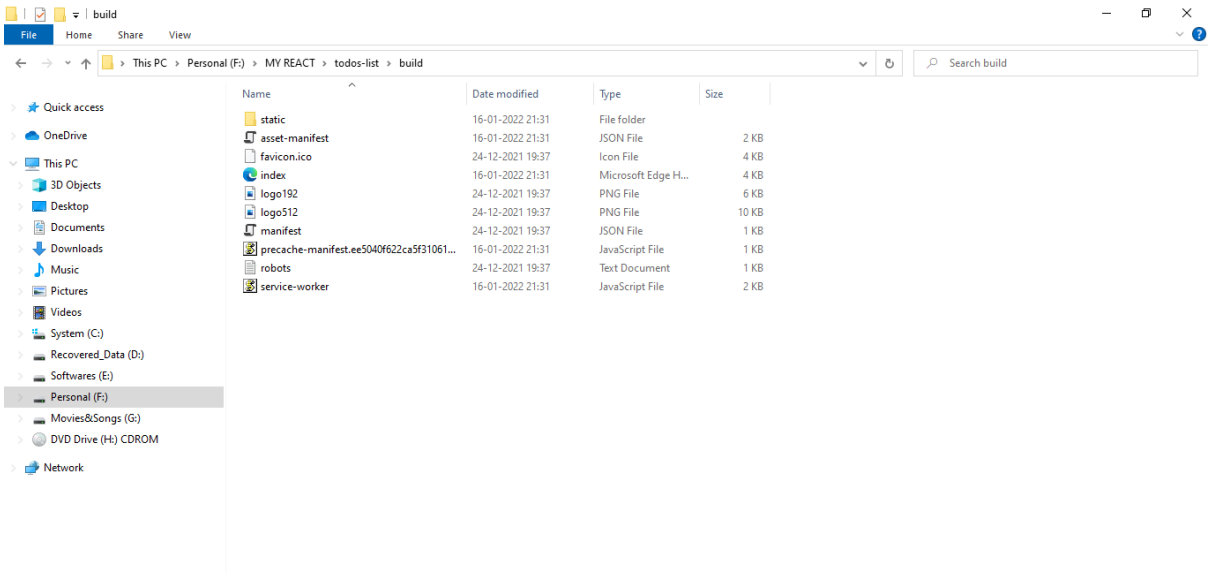
npm run build

It will Creating an optimized production build..

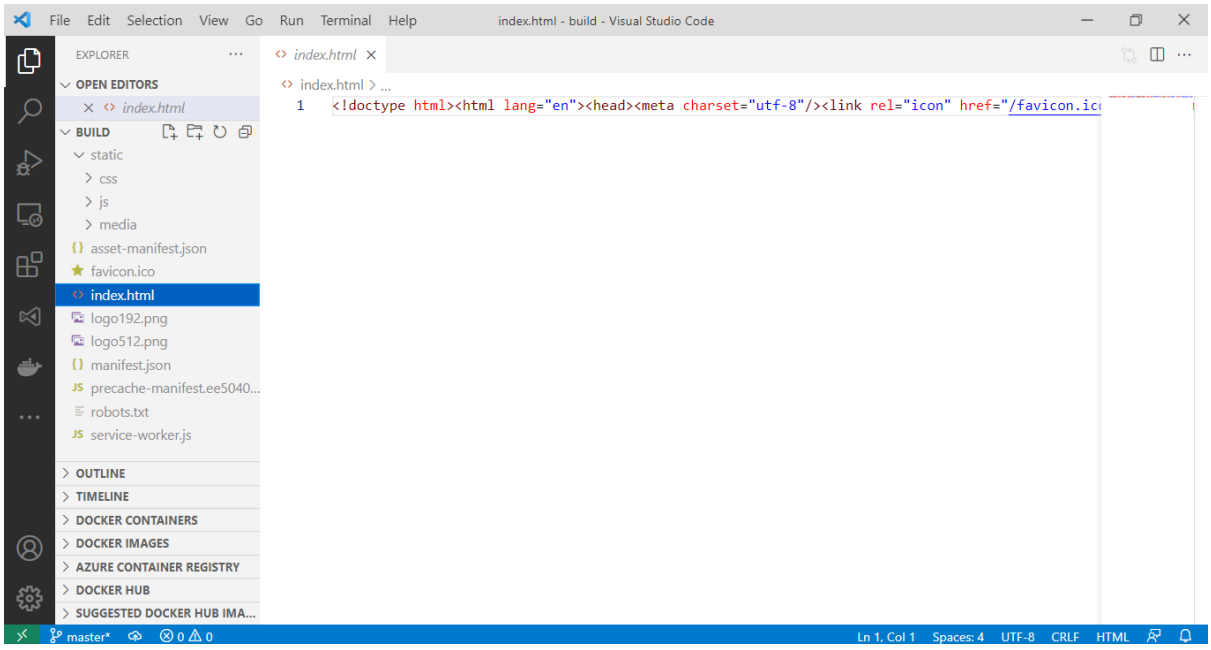


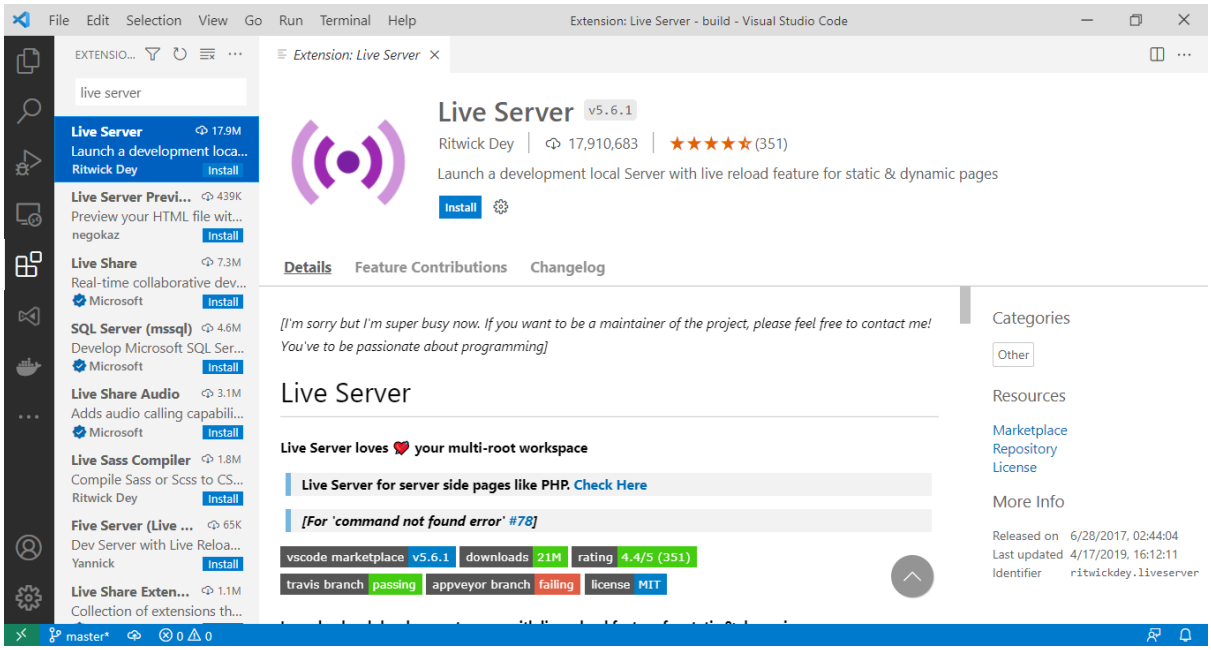

## Now open the build folder in windows explorer



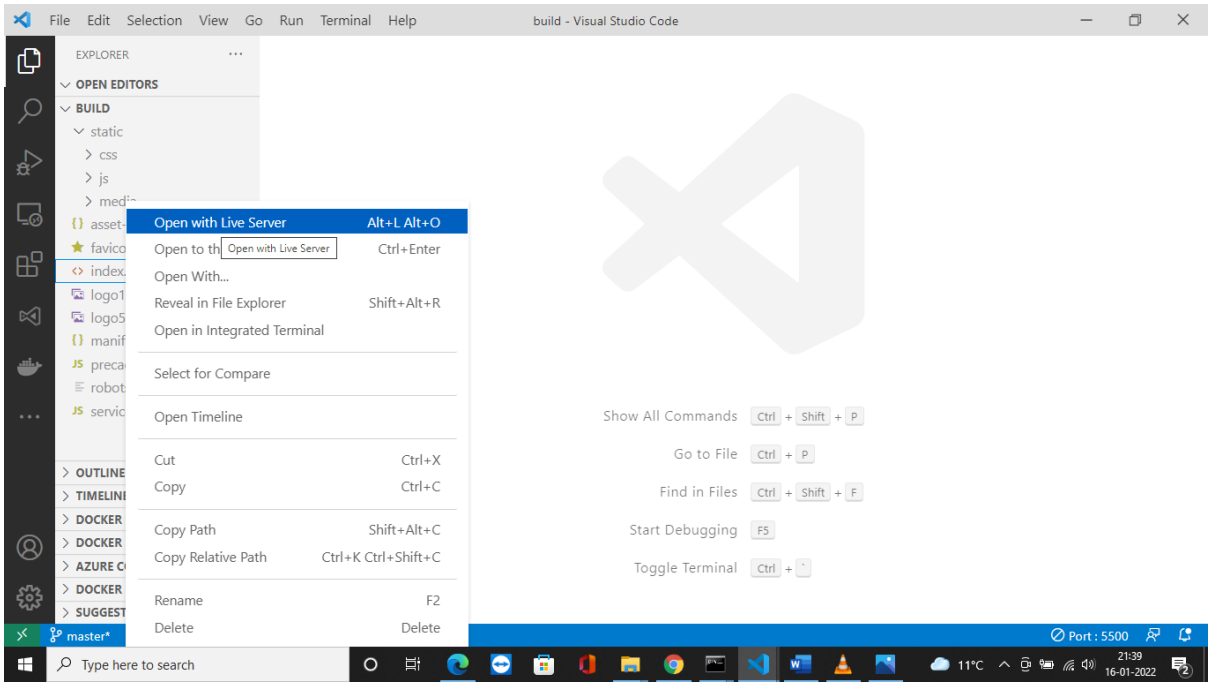

## Now open this folder with VS Code

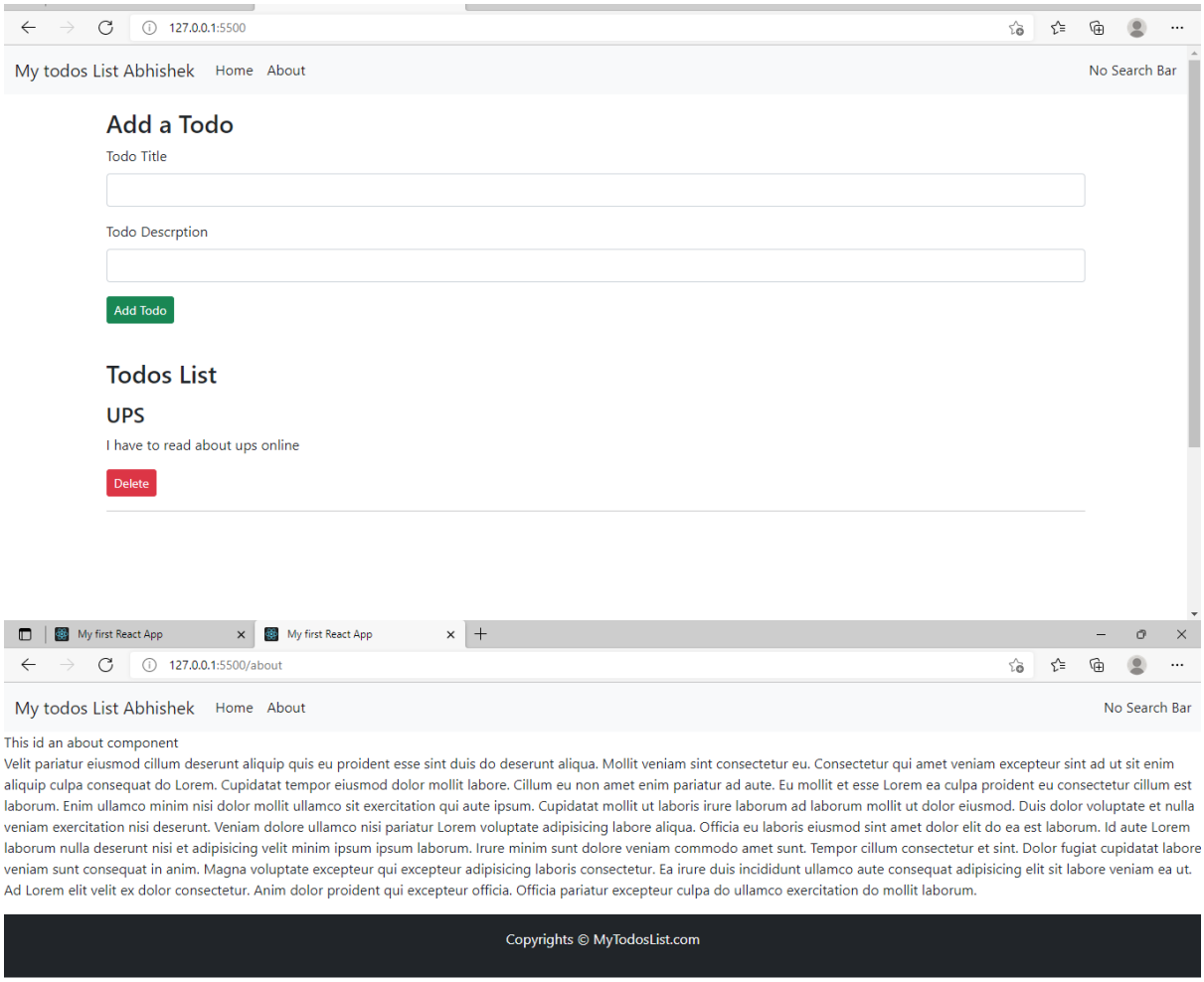



## Now install live server if you don't have.




**Now after installation, right click on index.html page and open with live server and enjoy learning**




127.0.0.1:5500
My todos List Abhishek
Home
About
No Search Bar
Add a Todo
Todo Title
Todo Descrption
Add Todo
Todos List
UPS
I have to read about ups online
Delete


My first React App
My first React App
127.0.0.1:5500/about
My todos List Abhishek
Home
About
No Search Bar
This id an about component
Velit pariatur eiusmod cillum deserunt aliquip quis eu proident esse sint duis do deserunt aliqua. Mollit veniam sint consectetur eu. Consectetur qui amet veniam excepteur sint ad ut sit enim aliquip culpa consequat do Lorem. Cupidatat tempor eiusmod dolor mollit labore. Cillum eu non amet enim pariatur ad aute. Eu mollit et esse Lorem ea culpa proident eu consectetur cillum est laborum. Enim ullamco minim nisi dolor mollit ullamco sit exercitation qui aute ipsum. Cupidatat mollit ut laboris irure laborum ad laborum mollit ut dolor eiusmod. Duis dolor voluptate et nulla veniam exercitation nisi deserunt. Veniam dolore ullamco nisi pariatur Lorem voluptate adipisicing labore aliqua. Officia eu laboris eiusmod sint amet dolor elit do ea est laborum. Id aute Lorem laborum nulla deserunt nisi et adipisicing velit minim ipsum ipsum laborum. Irure minim sunt dolore veniam commodo amet sunt. Tempor cillum consectetur et sint. Dolor fugiat cupidatat labore veniam sunt consequat in anim. Magna voluptate excepteur qui excepteur adipisicing laboris consectetur. Ea irure duis incididunt ullamco aute consequat adipisicing elit sit labore veniam ea ut. Ad Lorem elit velit ex dolor consectetur. Anim dolor proident qui excepteur officia. Officia pariatur excepteur culpa do ullamco exercitation do mollit laborum.
Copyrights © MyTodosList.com