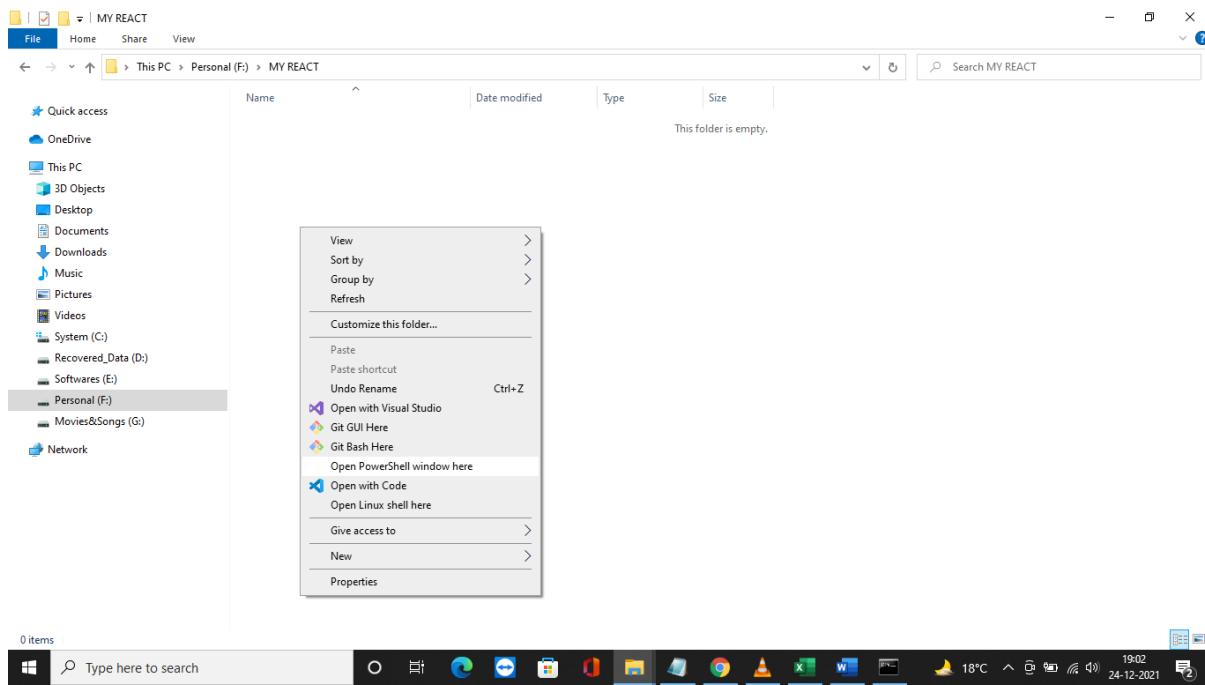


Learn step by step REACT.JS with CURD operation

The screenshot shows the official Node.js website at nodejs.org/en/. The main navigation bar includes links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. Below the navigation, a message states "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." Two prominent download buttons are displayed: "16.13.1 LTS Recommended For Most Users" and "17.3.0 Current Latest Features". Smaller links for "Other Downloads", "Changelog", and "API Docs" are visible below each button. A note at the bottom suggests looking at the "Long Term Support (LTS) schedule". The footer features the OpenJS Foundation logo and links for "Edit On GitHub" and "Report Node.js issue".

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ll, login, logout, ls, org, outdated, owner, pack, ping,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

```

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 / config

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

Entering npm script environment at location:
F:\MY REACT
Type 'exit' or ^D when finished

Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

F:\MY REACT>

```

```

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>

```

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

If you wish to run package through npm then you have to specify that package in your package.json and installed it locally.

To use create-react-app in npm the commands are npm install create-react-app then create-react-app myApp(Installation required).

Npm is a tool that use to install packages.

Packages used by npm are installed globally you have to care about pollution for the long term.

npx

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.

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.

Npx is a tool that use to execute packages.

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



Node Package manager

VS



Yet Another Resource Negotiator

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 dependencies 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.

```
C:\Users\Sabyasachi Samadder\gfg>yarn add express mongoose handlebars koa
yarn add v1.22.0
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
success Saved 85 new dependencies.
info Direct dependencies
├── express@4.17.1
├── handlebars@4.7.3
└── koa@2.11.0
    └── mongoose@5.9.2
    └── all dependencies
```

ApowerREC

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`

```
C:\Users\Sabyasachi Samadder\gfg>yarn licenses list
```

ApowerREC

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 npm install package_name@version_number | yarn add package_name 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 npm update package_name@version_number | yarn upgrade package_name 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 |

Google search results for "create react app". The top result is "Create a New React App" from reactjs.org.

[Create a New React App](https://reactjs.org/docs/create-a-new-react-app.html)

Set up a modern web app by running one command. Contribute to [facebook/react](#) development by [creating](#) an account on GitHub.

[Create React App](https://create-react-app.dev/)

Set up a modern web app by running one command.

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

React Documentation

Docs Tutorial Blog Community Search v17.0.2 Languages GitHub

Create a New React App

Use an integrated toolchain for the best user and developer experience.

This page describes a few popular React toolchains which help with tasks like:

- Scaling to many files and components.
- Using third-party libraries from npm.
- Detecting common mistakes early.
- Live-editing CSS and JS in development.
- Optimizing the output for production.

The toolchains recommended on this page **don't require configuration to get started.**

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

Have Node ≥ 14.0.0 and npm ≥ 5.0 on your machine. To create a project, run:

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

```

E:\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 E:\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 E:\MY REACT\todos-list
Inside that directory, you can run several commands:

```

Npm start

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with a file named "src" selected.
- Editor:** Displays the "App.js" file content:

```

1 import logo from './logo.svg';
2 import './App.css';
3
4 function App() {
5   return (
6     <div className="App">
7       <header className="App-header">
8         <img src={logo} className="App-logo" alt="logo" />
9       <p>

```
- Terminal:** Shows the output of the "npm start" command:

```

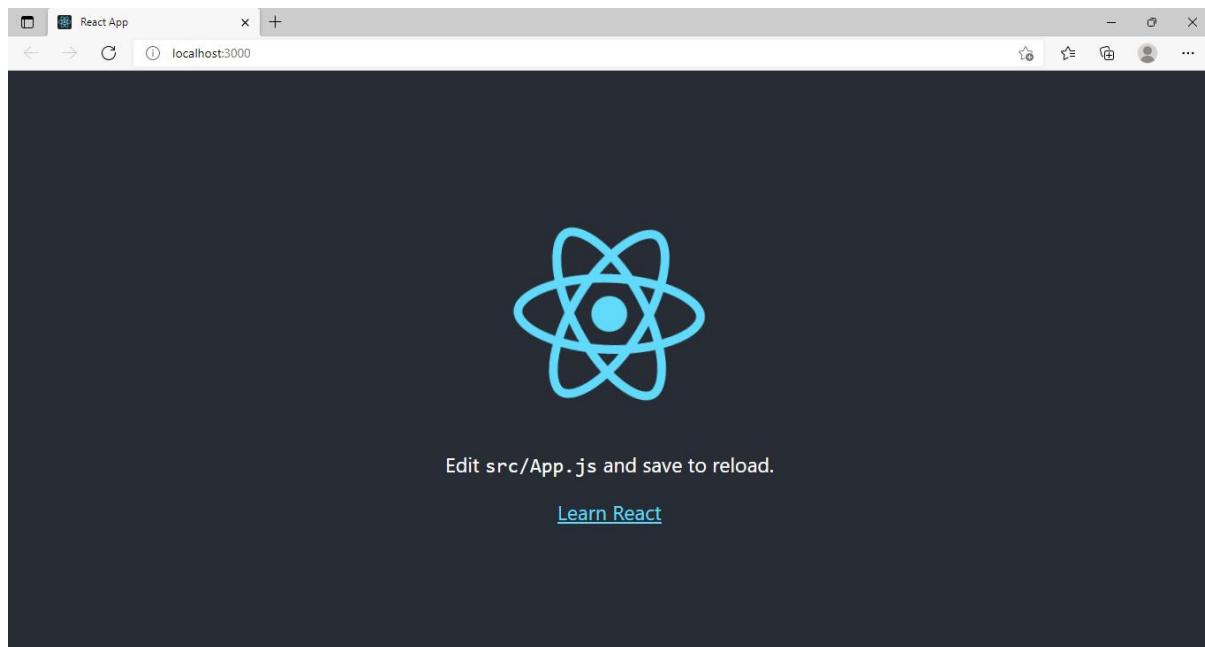
> todos-list@0.1.0 start
> react-scripts start
Compiled successfully!

You can now view todos-list in the browser.

Local:          http://localhost:3000
On Your Network: http://172.19.192.1:3000

Note that the development build is not optimized.

```

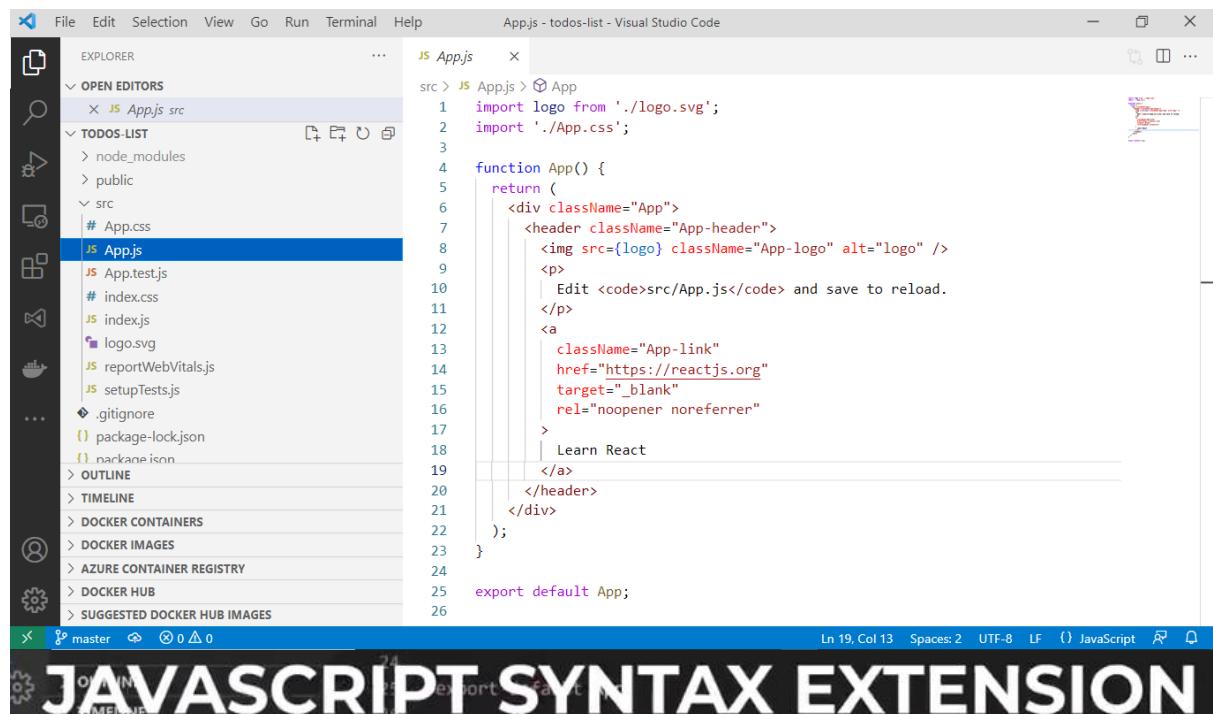


```
import logo from './logo.svg';
import './App.css';

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 label for



```

File Edit Selection View Go Run Terminal Help
App.js - todos-list - Visual Studio Code
EXPLORER ... JS App.js ...
OPEN EDITORS ... src > JS App.js > App
TODOS-LIST > node_modules
> public
src # App.css
JS App.js
JS App.test.js
# index.css
JS index.js
logo.svg
JS reportWebVitals.js
JS setupTests.js
.gitignore
{} package-lock.json
{} package.json
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
master 0 △ 0 Ln 19, Col 13 Spaces: 2 UTF-8 LF () JavaScript ⌂ ⌂
JS App.js
src > JS App.js > App
1 import logo from './logo.svg';
2 import './App.css';
3
4 function App() {
5   return (
6     <div className="App">
7       <header className="App-header">
8         <img src={logo} className="App-logo" alt="logo" />
9         <p>
10            Edit <code>src/App.js</code> and save to reload.
11        </p>
12        <a
13          className="App-link"
14          href="https://reactjs.org"
15          target="_blank"
16          rel="noopener noreferrer"
17        >
18          Learn React
19        </a>
20      </header>
21    </div>
22  );
23}
24
25 export default App;

```

JAVASCRIPT SYNTAX EXTENSION

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.

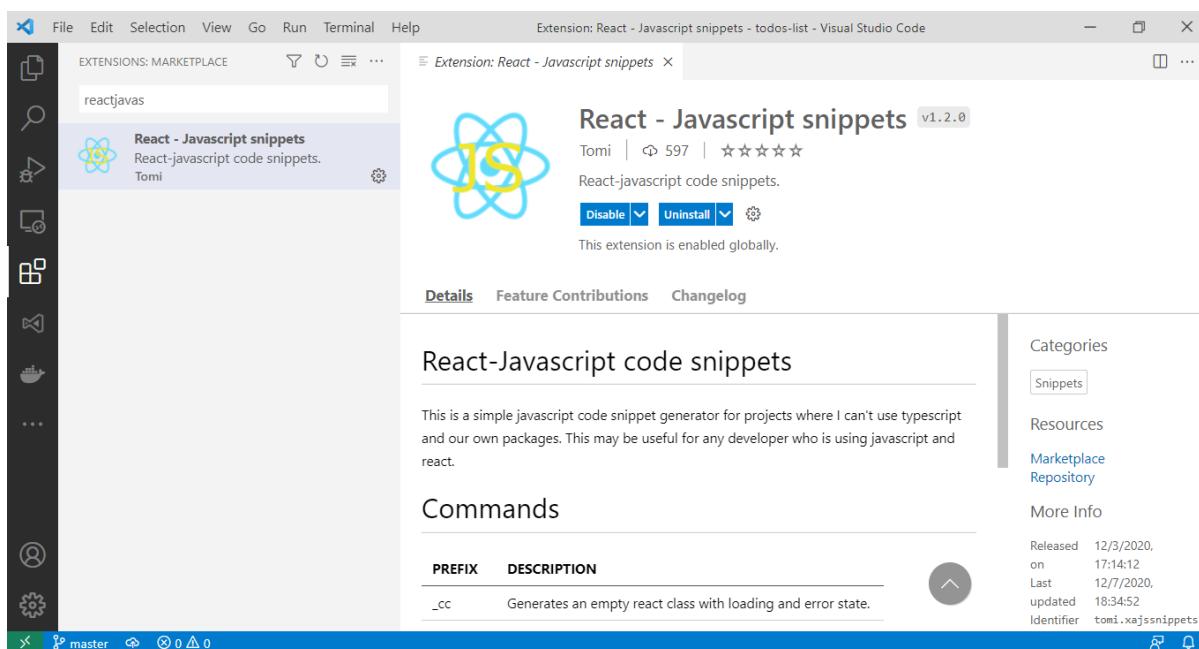
- After compilation, JSX expressions become regular JavaScript function calls.
- JSX uses camelcase notation for naming HTML attributes. For example, tabindex in HTML is used as tabIndex in JSX.

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:

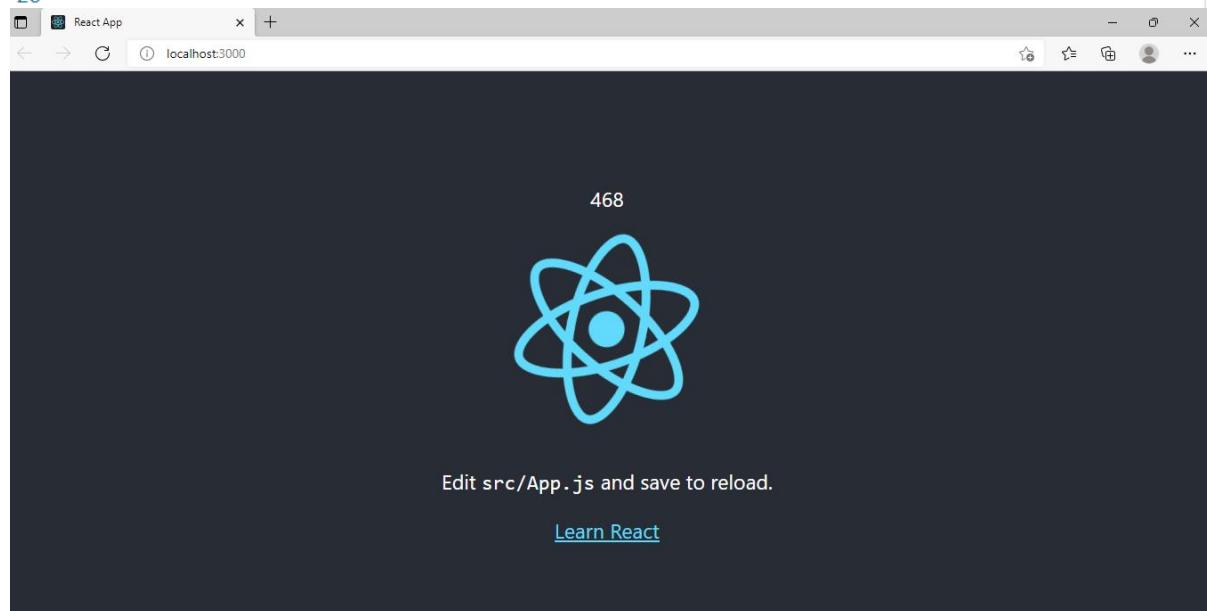
- JSX throws an error if the HTML is not correct.
- In JSX HTML code must be wrapped in one top-level element otherwise it will give an error.
- If HTML elements are not properly closed JSX will give an error.



```

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

```

```

src > JS App.js > ⚡ App
1 import logo from './logo.svg';
2 import './App.css';
3
4 function App() {
5   let Myvariable = 1108;
6   return (
7
8     <div className="App">
9       <header className="App-header">
10        <div>{Myvariable}</div>
11        <img src={logo} className="App-logo" alt="logo" />
12        <p>
13          | Edit <code>src/App.js</code> and save to reload.
14        </p>
15        <a
16          | className="App-link"
17          | href="https://reactjs.org"
18          | target="_blank"
19          | rel="noopener noreferrer"
20        >
21          | Learn React
22        </a>
23      </header>
24    </div>
25  );
26}

```

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

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure with files like App.css, App.js, App.test.js, index.css, index.js, logo.svg, reportWebVitals.js, setupTests.js, .gitignore, package-lock.json, and package.json.
- EDITOR:** The active file is App.js, displaying the following code:


```

1 import './App.css';
2
3 function App() {
4   let Myvariable = 1108;
5   return (
6     <>
7       <h1>This is my first react app</h1>
8       <p>My app works !!</p>
9     </>
10    );
11 }
12
13 <div className="App">
14   <header className="App-header">
15     <div>{Myvariable}</div>
16     <img alt="logo" src={logo} className="App-logo" />
17     <p>
18       Edit <code>src/App.js</code> and save to reload.
19     </p>
20     <a href="https://reactjs.org" rel="noopener noreferrer" target="_blank">
21       Learn React
22     </a>
23   </header>
24 </div>
25
26
27
      
```
- STATUS BAR:** Shows "React App" and "localhost:3000".

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 wrap 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>
    </>
  );
}

export default App;
      
```

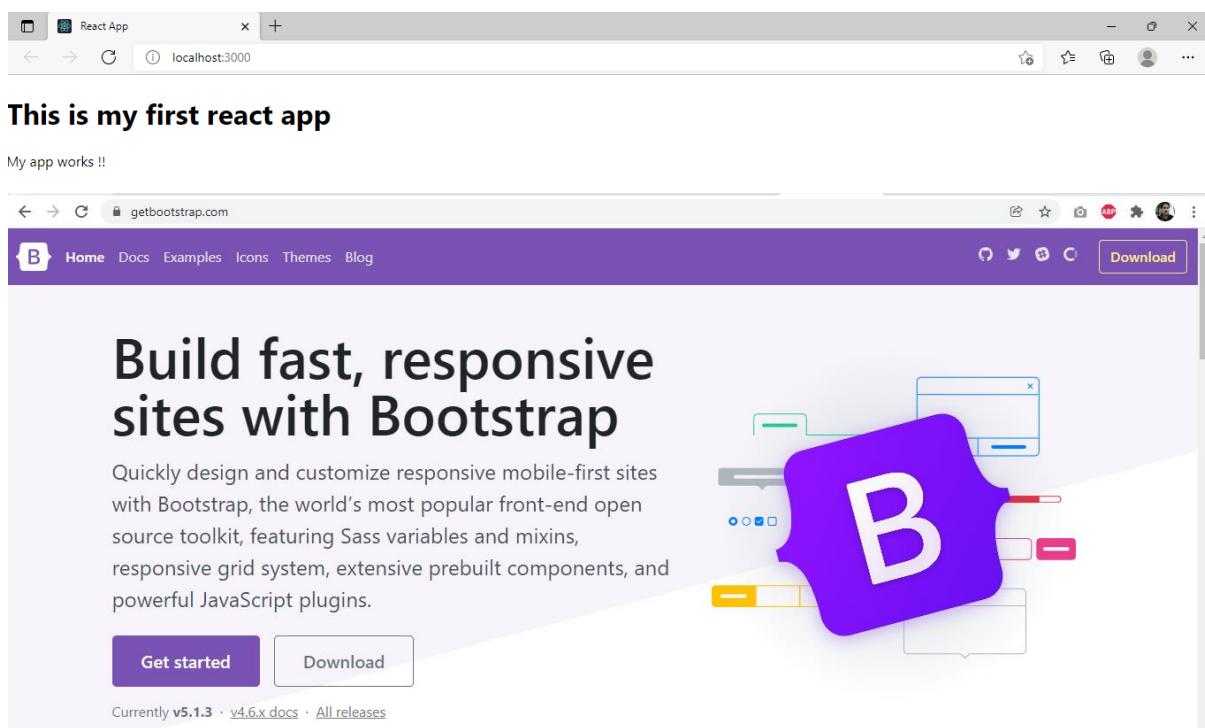
```

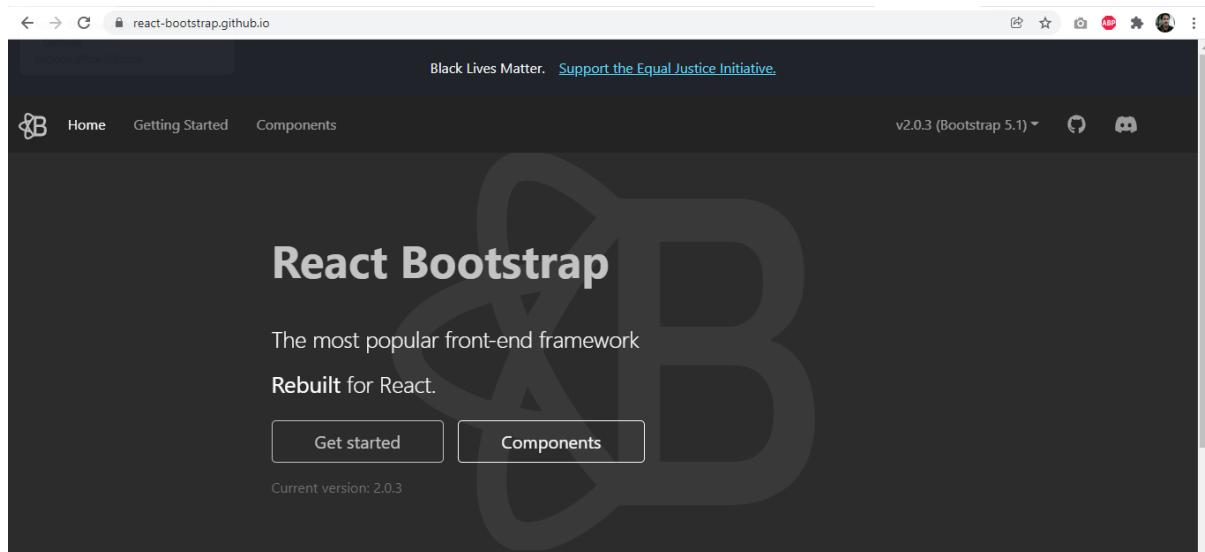
        </>

    // <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 className="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-

```

```

1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl20vZ6jIW3"
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-ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+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>
        </div>
      </div>
    </nav>
  );
}

export default App;

```

```

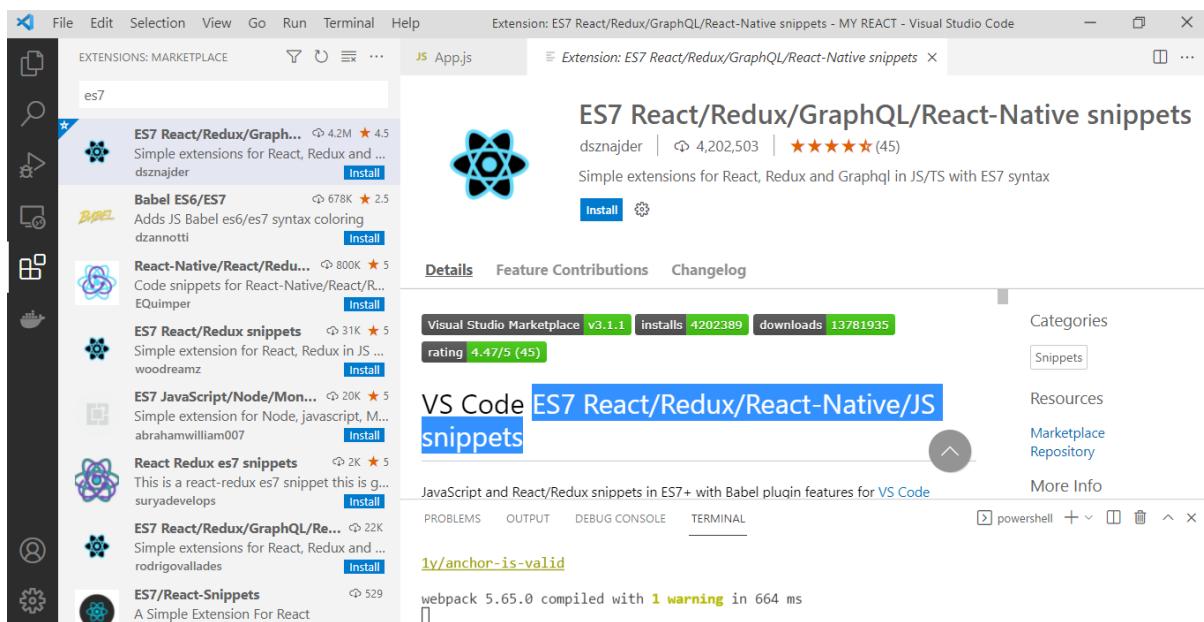
        <form className="d-flex">
          <input className="form-control me-2" type="search"
placeholder="Search" aria-label="Search" />
          <button className="btn btn-outline-success"
type="submit">Search</button>
        </form>
      </div>
    </div>
  </nav>
)
}

export default App;

```



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



Now let's add new folder in SRC folder.

```

File Edit Selection View Go Run Terminal Help
App.js - MY REACT - Visual Studio Code
EXPLORER OPEN EDITORS ...
todos-list > src > JS App.js todos-list\src
MY REACT todos-list
node_modules
public
  favicon.ico
  index.html
  logo192.png
  logo512.png
  manifest.json
  robots.txt
src
  MyComponents
    # App.css
    JS App.js
    JS App.test.js
    # index.css
    JS index.js
    JS logo.svg
    JS reportWebVitals.js
    JS setupTests.js
    # initmate
> OUTLINE
Restricted Mode 0 △ 0
JS App.js
1 import logo from './logo.svg';
2 import './App.css';
3
4 function App() {
5   let Myvariable = 1108;
6   return (
7     <nav className="navbar navbar-expand-lg navbar-light bg-light">
8       <div className="container-fluid">
9         <a className="navbar-brand" href="#">Todos List</a>
10        <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
11          <span className="navbar-toggler-icon"></span>
12        </button>
13        <div className="collapse navbar-collapse" id="navbarSupportedContent">
14          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
15            <li className="nav-item">
16              <a className="nav-link active" aria-current="page" href="#">Home</a>
17            </li>
18            <li className="nav-item">
19              <a className="nav-link active" aria-current="page" href="#">About</a>
20            </li>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell + ×
1/y/anchor-is-valid
webpack 5.65.0 compiled with 1 warning in 664 ms
Ln 18, Col 38 Spaces: 2 UTF-8 LF () JavaScript ⚡

```

Now add header.js

```

File Edit Selection View Go Run Terminal Help
Header.js - MY REACT - Visual Studio Code
EXPLORER OPEN EDITORS ...
todos-list > src > MyComponents > JS Header.js
MY REACT
  node_modules
  public
    favicon.ico
    index.html
    logo192.png
    logo512.png
    manifest.json
    robots.txt
  src
    MyComponents
      JS Header.js
        # App.css
        JS App.js
      JS Header.js
        rfc
          reactFunctionalComponent
          reactFunctionalExportComponent
          reactFunctionalComponentWithPropTypes
          reactFunctionalComponentReduxPropTypes
          reactFunctionalComponentReduxPropTypes
          _reactFunctionalComponent
          _reactFunctionalExportComponent
          _reactFunctionalComponentWithPropTypes
          reactArrowFunctionComponent
          reactArrowFunctionExportComponent
          reactArrowFunctionComponentWithPropTypes
          _reactArrowFunctionComponent
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell + ×
webpack 5.65.0 compiled with 1 warning in 664 ms
Ln 1, Col 4 Spaces: 4 UTF-8 CRLF () JavaScript ⚡

```

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

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like Header.js, App.js, favicon.ico, index.html, logo192.png, logo512.png, manifest.json, robots.txt, and subfolders src and MyComponents.
- Code Editor:** Displays the content of Header.js, which contains a React component definition.
- Bottom Status Bar:** Shows the file path (todos-list > src > MyComponents > Header.js), line numbers (Ln 10, Col 1), character count (Spaces: 4), encoding (UTF-8), line endings (CRLF), language (JavaScript), and a save icon.

```

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>
  )
}

```

```

        </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;

```

```

File Edit Selection View Go Run Terminal Help
App.js - MY REACT - Visual Studio Code
EXPLORER OPEN EDITORS MY REACT
favicon.ico index.html logo192.png logo512.png manifest.json robots.txt
src MyComponents Header.js App.css App.js App.test.js
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
webpack 5.65.0 compiled with 1 warning in 238 ms
Ln 14, Col 1 Spaces: 2 UTF-8 LF () JavaScript

```



File Edit Selection View Go Run Terminal Help Footer.js - MY REACT - Visual Studio Code todos-list > src > MyComponents > Footer.js

```
1 rafc
  □ rafc          reactArrowFunctionComponent
  □ rafce         reactArrowFunctionExportComponent
  □ rafcp         reactArrowFunctionComponentWithPropTypes
  □ _rafc         _reactArrowFunctionComponent
  □ _rafce        _reactArrowFunctionExportComponent
  □ _rafcp        _reactArrowFunctionComponentWithPropTypes
  [o] ReadableStreamDefaultController
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL webpack 5.65.0 compiled with 1 warning in 559 ms

Ln 1, Col 5 Spaces: 4 UTF-8 CRLF () JavaScript

Type here to search

Let's have RAFC arrow function based components to use

File Edit Selection View Go Run Terminal Help Footer.js - MY REACT robots.txt

src

MyComponents

- Footer.js
- Header.js
- Todo.js
- Todos.js
- App.css
- App.js
- App.test.js

File Edit Selection View Go Run Terminal Help Footer.js - MY REACT todos-list > src > MyComponents > Footer.js

```
1 import React from 'react'
2
3 export const Footer = () => {
4   return (
5     <div>
6       | Footer Works !!
7     </div>
8   )
9 }
10
```

Header.js App.js Footer.js Todos.js Todo.js App.css App.js

```

File Edit Selection View Go Run Terminal Help Todo.js - MY REACT - Visual Studio Code
EXPLORER ... JS Header.js U JS App.js M JS Footer.js U JS Todos.js U JS Todo.js U X
todos-list > src > MyComponents > JS Todo.js > ...
1 import React from 'react'
2
3 export const Todo = () => {
4   return (
5     <div>
6       | Todo Works!!
7     </div>
8   )
9 }
10

File Edit Selection View Go Run Terminal Help Todos.js - MY REACT - Visual Studio Code
EXPLORER ... JS Header.js U JS App.js M JS Footer.js U JS Todos.js U JS Todo.js U X
todos-list > src > MyComponents > JS Todos.js > ...
1 import React from 'react'
2
3 export const Todos = () => {
4   return (
5     <div>
6       | Todos Works!!
7     </div>
8   )
9 }
10

```

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 />
    </>
  );
}

export default App;

```

```

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

export default App;

```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the project structure. Under "src/MyComponents", there are Header.js, Footer.js, Todos.js, and Todo.js.
- Code Editor:** The main editor window displays the `App.js` file content. The code defines a function `App()` which returns a component structure containing `<Header/>`, `<Todos/>`, `<Todo/>`, and `<Footer/>`.
- Terminal:** At the bottom, it shows the output of the webpack compilation: "webpack 5.65.0 compiled with 1 warning in 265 ms".
- Browser Preview:** Below the terminal, a browser window shows the application running at `localhost:3000/#`. The page title is "Todos List". The content area displays the message "Todos Works!!" followed by "Todo Works!!" and "Footer Works !!".

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.

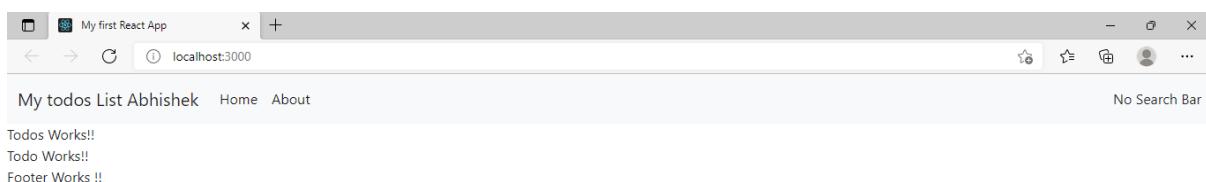
You can apply Boolean values also, as follows

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 (

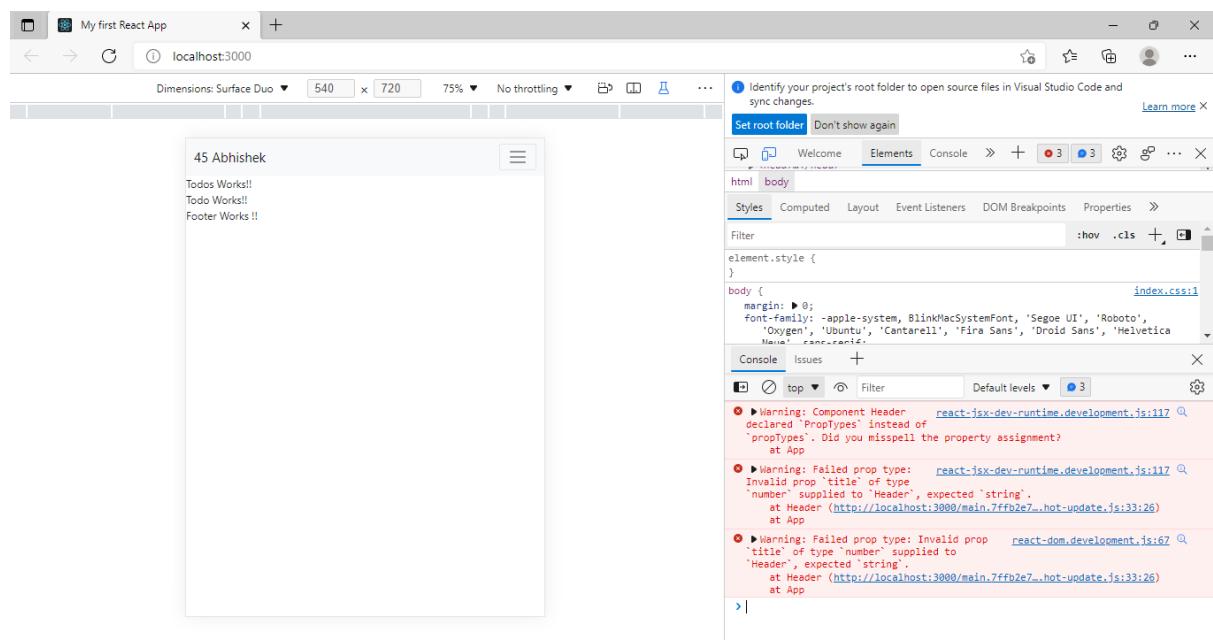
```

```

    <>
      <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';

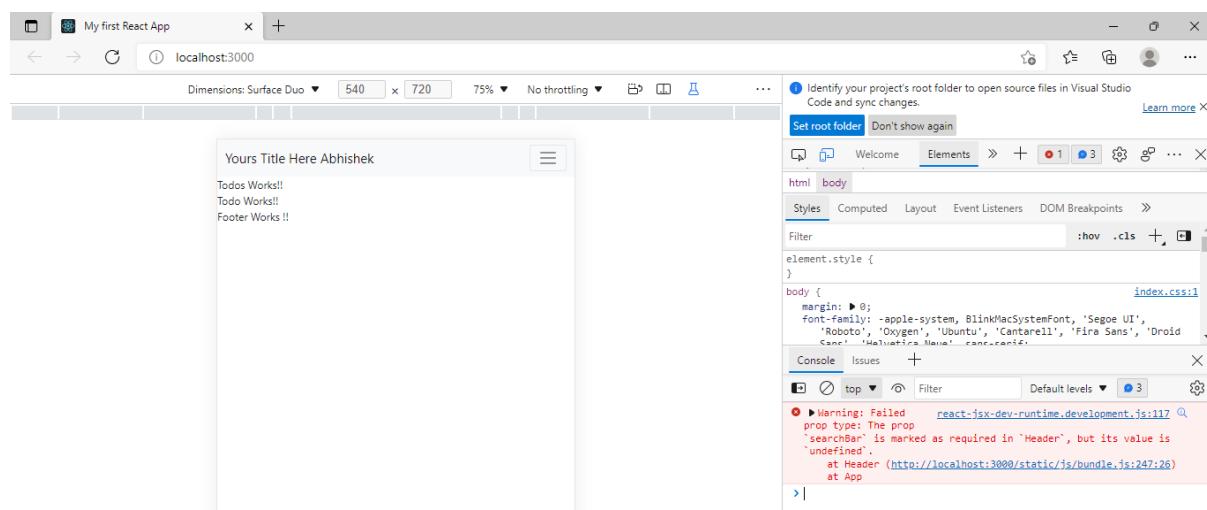
import './App.css';
import Header from './MyComponents/Header';
import { Footer } from './MyComponents/Footer';
import { TodoItem } from './MyComponents/TodoItem';
import { Todos } from './MyComponents/Todos';

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

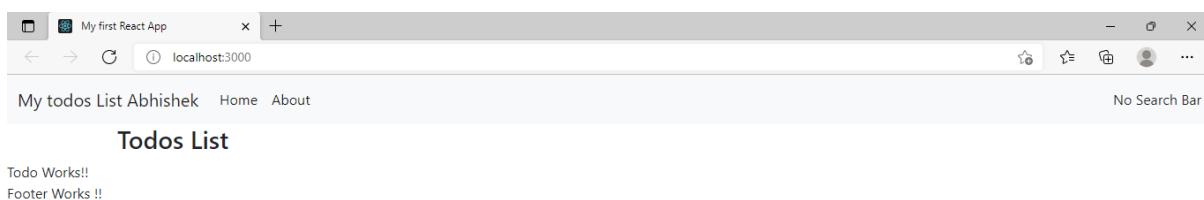


Now let's have some changes in the Todos.js, this time I have to display list of todos, as follows.

Todos.js

```
import React from 'react'

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



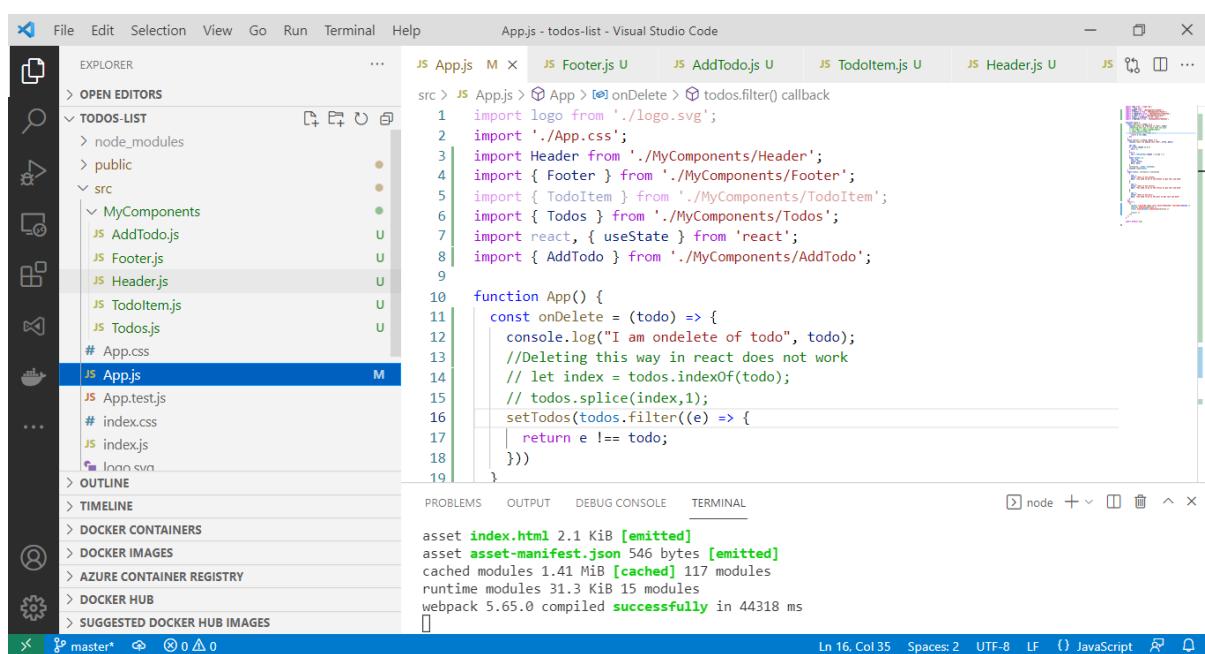
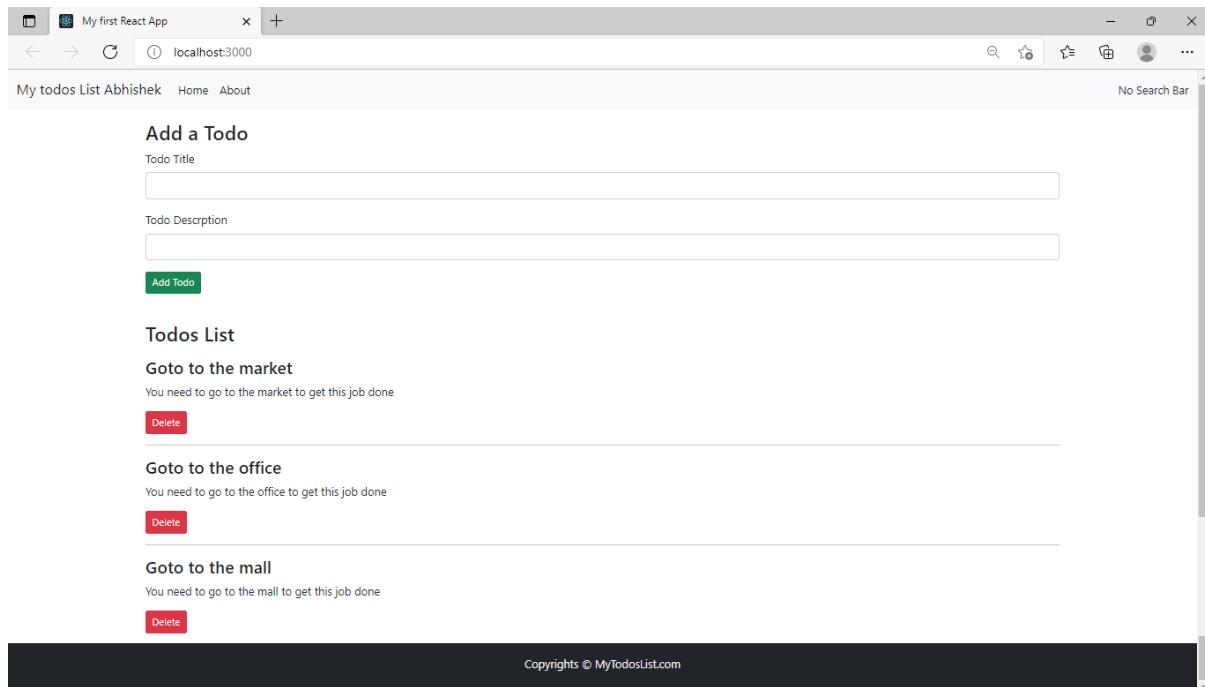
Now let's create some todo list in the app.js file which we are going to display in the todos.js

```
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);
  ]
}
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Tab Bar:** JS App.js M, JS Todos.js U, Footer.js U, JS AddTodo.js U X, JS TodoItem.js U.
- Editor Area:** The code for `AddTodo.js` is displayed. The code uses React components like `<Form>`, `<Label>`, `<Input type="email">`, and `<Button type="submit">`. It also includes CSS-in-JS classes such as `form-label`, `form-control`, and `form-check`.
- Sidebar:** On the left, there are icons for file operations (New, Open, Save, Find, Replace, etc.) and a search bar.
- Status Bar:** At the bottom, it says "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';

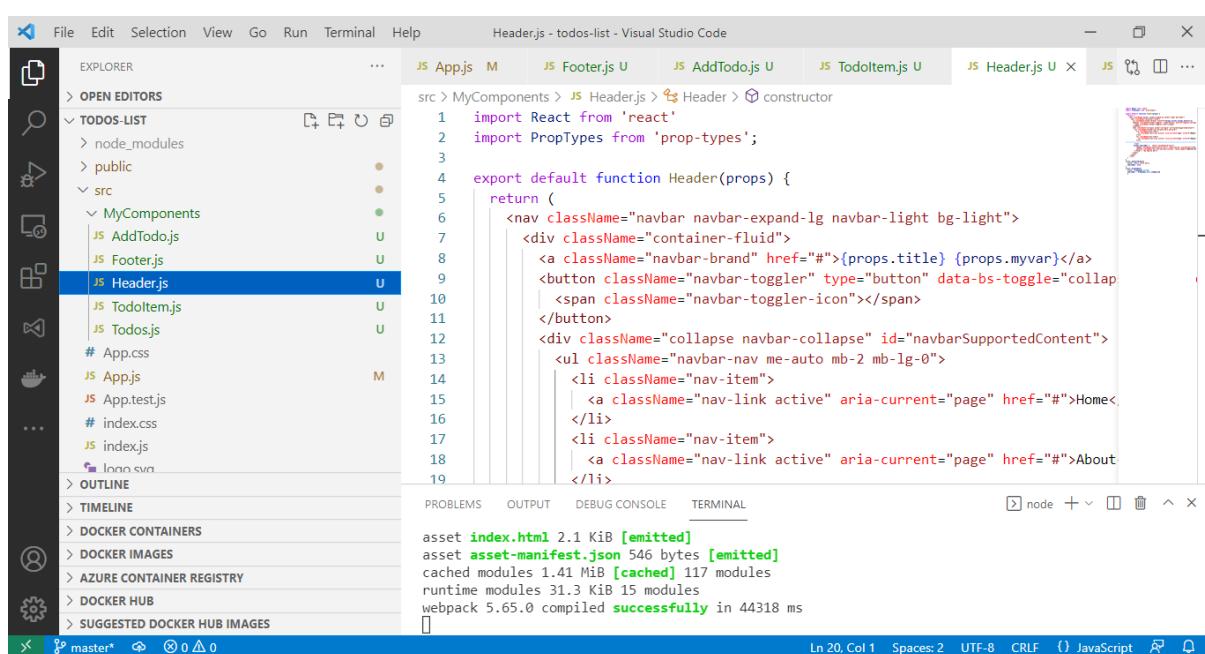
function App() {
  const onDelete = (todo) => {
    console.log("I am ondelete of todo", todo);
    //Deleting this way in react does not work
    // let index = todos.indexOf(todo);
    // todos.splice(index,1);
    setTodos(todos.filter((e) => {
      return e !== todo;
    }))
  }
  const addTodo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

```

File Edit Selection View Go Run Terminal Help
Footer.js - todos-list - Visual Studio Code
EXPLORER ... JS App.js M JS Footer.js U X JS AddTodo.js U JS TodoItem.js U JS Header.js U JS Footer.js ...
src > MyComponents > JS Footer.js > Footer > footerStyle
1 import React from 'react'
2
3 export const Footer = () => {
4   let footerStyle = [
5     position: "relative",
6     top: "70vh",
7     width: "100%"
8   ]
9
10  return (
11    // <footer className="bg-dark text-light py-3" style={footerStyle}>
12    <footer className="bg-dark text-light py-3">
13      <p className="text-center">
14        Copyrights &copy; MyTodosList.com
15      </p>
16    </footer>
17  )
18}
19

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

asset index.html 2.1 KiB [emitted]
asset asset-manifest.json 546 bytes [emitted]
cached modules 1.41 MiB [cached] 117 modules
runtime modules 31.3 KiB 15 modules
webpack 5.65.0 compiled successfully in 44318 ms

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>
  )
}

```

```

File Edit Selection View Go Run Terminal Help
AddTodo.js - todos-list - Visual Studio Code
EXPLORER ... JS Footer.js U JS AddTodo.js U X JS TodoItem.js U JS Header.js U JS Todos.js U ⌂ ⌂ ...
TODOS-LIST node_modules public src MyComponents JS AddTodo.js U JS Footer.js U JS Header.js U JS TodoItem.js U JS Todos.js U # App.css JS App.js JS App.test.js # index.css JS index.js .icon.svg OUTLINE PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
src > MyComponents > JS AddTodo.js > [e] AddTodo > [e] submit
1 import React, { useState } from 'react'
2
3 export const AddTodo = ({addTodo}) => {
4   const [title, setTitle] = useState("");
5   const [desc, setDesc] = useState("");
6
7   const submit = (e)=> {
8     e.preventDefault(); //Page will not reload
9     if (!title || !desc) {
10       alert("Title or Description cannot be blank")
11     }
12     addTodo(title, desc);
13   }
14
15   return (
16     <div className="container my-3">
17       <h3>Add a Todo</h3>
18       <form onSubmit={submit}>
19         <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">

```

In 13, Col 6 Spaces: 4 UTF-8 CRLF () JavaScript ⌂ ⌂

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
Description</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>
)
}

```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under 'src' with files: Footer.js, AddTodo.js, TodoItem.js, Header.js, and Todos.js.
- Code Editor:** The 'Todos.js' file is open, displaying the provided React code.
- Terminal:** Shows the command 'node' and other terminal logs.
- Status Bar:** Displays file statistics: master*, 0△0, Ln 19, Col 35, Spaces 4, UTF-8, CRLF, JavaScript.

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 display" :
                props.todos.map((todo) => {

```

```

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

    </div>
  )
}

```

```

File Edit Selection View Go Run Terminal Help TodoItem.js - todos-list - Visual Studio Code
EXPLORER ... JS Footer.js U JS AddTodo.js U JS TodoItem.js U X JS Header.js U JS Todos.js U index.html
src > MyComponents > JS TodoItem.js ...
1 import React from 'react'
2
3 export const TodoItem = ({ todo, onDelete }) => {
4   return (
5     // <div>
6     //   Todo Works!!
7     // </div>
8     <div>
9       <h4>{todo.title}</h4>
10      <p>{todo.desc}</p>
11      <button className="btn btn-sm btn-danger" onClick={()=>{onDelete(todo)}}>Delete</button>
12    )
13  }
14
15

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

asset index.html 2.1 KiB [emitted]
asset asset-manifest.json 546 bytes [emitted]
cached modules 1.41 MiB [cached] 117 modules
runtime modules 31.3 KIB 15 modules
webpack 5.65.0 compiled successfully in 44318 ms

Ln 15, Col 1 Spaces: 4 UTF-8 CRLF () JavaScript ⚡ ⚡

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>
  )
}

```

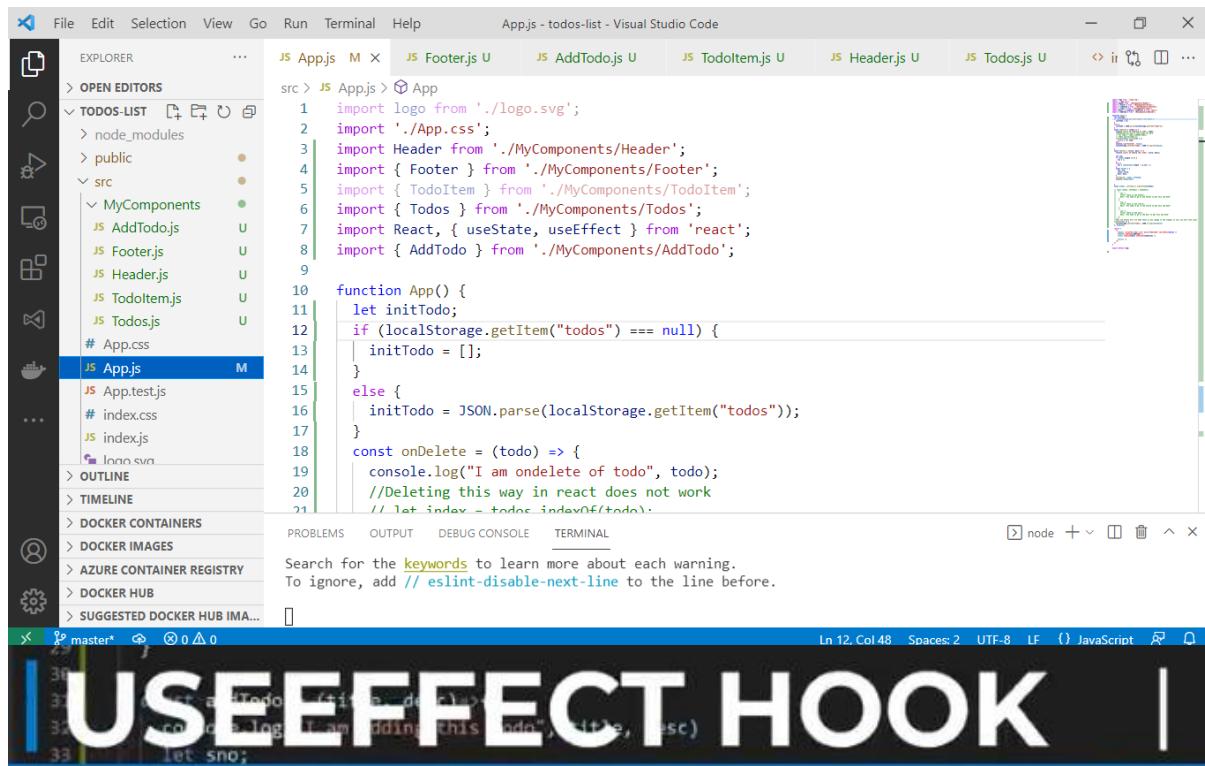
```

        )
}

```

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.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like App.js, Footer.js, AddTodo.js, TodoItem.js, Header.js, and Todos.js.
- Code Editor:** The current file is App.js, displaying the following code:


```

1 import logo from './logo.svg';
2 import './App.css';
3 import Header from './MyComponents/Header';
4 import { Footer } from './MyComponents/Footer';
5 import { TodoItem } from './MyComponents/TodoItem';
6 import { Todos } from './MyComponents/Todos';
7 import React, { useState, useEffect } from 'react';
8 import { AddTodo } from './MyComponents/AddTodo';
9
10 function App() {
11   let initTodo;
12   if (localStorage.getItem("todos") === null) {
13     initTodo = [];
14   }
15   else {
16     initTodo = JSON.parse(localStorage.getItem("todos"));
17   }
18   const onDelete = (todo) => {
19     console.log("I am onDelete of todo", todo);
20     //Deleting this way in react does not work
21     //let index = todos.indexOf(todo);
22   }
23 }
24
25 export default App;
      
```
- Terminal:** Shows the command `node`.
- Bottom Status Bar:** Shows the line number (Ln 12), column (Col 48), spaces (Spaces: 2), encoding (UTF-8), line feed (LF), and file type (JavaScript).

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);
    localStorage.removeItem("todos");
    let sno;
    let todos = [...initTodo];
    todos.splice(index, 1);
    localStorage.setItem("todos", JSON.stringify(todos));
  }
}

export default App;
      
```

```

        }
    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;

```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like App.js, Footer.js, Header.js, TodoItem.js, Todos.js, App.css, App.test.js, index.css, and index.js.
- Code Editor:** Displays the AddTodo.js file containing the following code:

```

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>

```

- Status Bar:** Shows the file is in the master branch, has 0 changes, and 0 issues.

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
                    Description</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

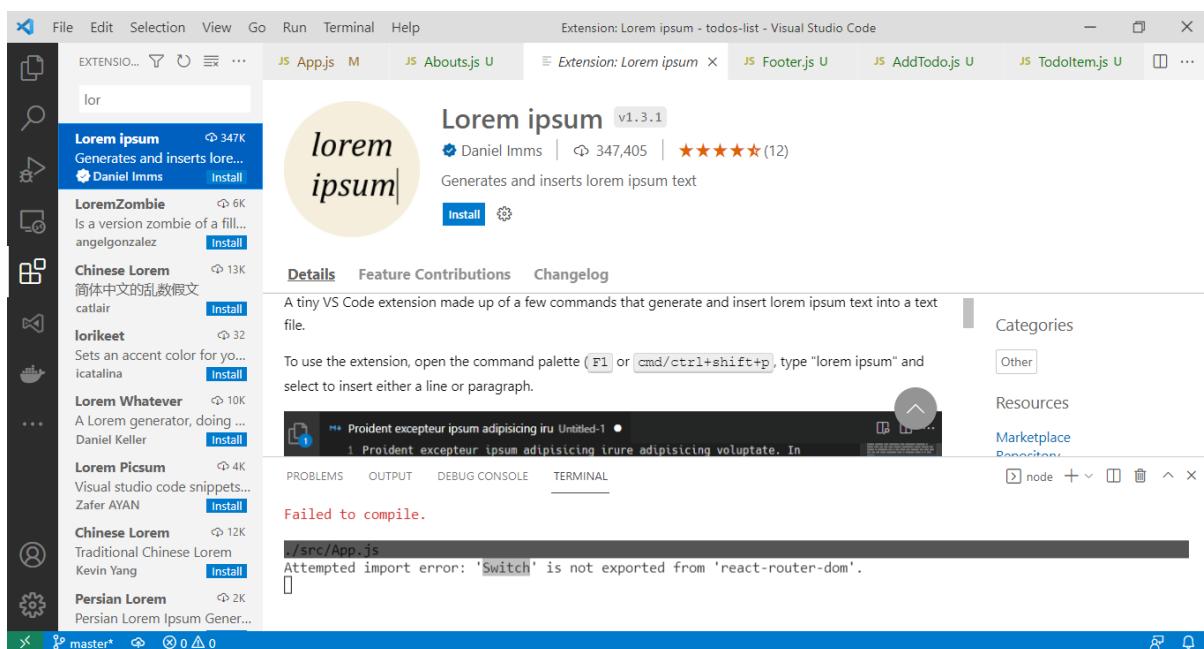
npm uninstall react-router-dom

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**

```

File Edit Selection View Go Run Terminal Help
About.js - todos-list - Visual Studio Code
EXPLORER ... JS App.js M JS Header.js U JS Todos.js U JS TodoItem.js U JS About.js U X
src > MyComponents > JS About.js > [e] About
1 import React from 'react'
2
3 export const About = () => {
4   return (
5     <div>
6       This id an about component
7       <p>
8         Velit pariatur eiusmod cillum deserunt aliquip quis eu proident esse sint duis
9           Cupidatat tempor eiusmod dolor mollit labore. Cillum eu non amet enim pariatur
10          Veniam dolore ullamco nisi pariatur Lorem voluptate adipisicing labore aliqua.
11          Tempor cillum consectetur et sint. Dolor fugiat cupidatat labore veniam sunt con
12        </p>
13      </div>
14    )
15  )
16)
17
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
./src/MyComponents/Footer.js
Line 4:9: 'footerStyle' is assigned a value but never used no-unused-vars
Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

```

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

Google search results for "react router dom". The search bar shows "react router dom". The "All" tab is selected. The results list includes:

- [Quick Start - React Router: Declarative Routing for React.js](https://reactrouter.com/web/guides/quick-start)
- [Overview - React Router](https://reactrouter.com/web/guides/overview)

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

The screenshot shows the official React Router documentation for the 'WEB' version. The 'Quick Start' guide is displayed, which includes instructions for installing the library via npm or yarn and provides a basic example of routing. A terminal window on the right side of the page shows the commands to set up a new React application and install the required dependencies.

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.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Tab Bar:** App.js M X, Header.js U, Todos.js U, TodoItem.js U, About.js U, # App.css
- Explorer:** Shows the project structure with files like index.js, App.js, App.test.js, and index.css.
- Code Editor:** Displays the following code snippet for 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) => {
    ...
  }
  return (
    <div>
      <Header todos={initTodo}></Header>
      <BrowserRouter>
        <Switch>
          <Route exact path="/" component={Todos} />
          <Route exact path="/about" component={About} />
        </Switch>
      </BrowserRouter>
    </div>
  );
}

export default App;
      
```
- Bottom Status Bar:** master*, 0 △ 0, Ln 12, Col 10, Spaces: 2, UTF-8, LF, JavaScript

```

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) => {
    ...
  }
  return (
    <div>
      <Header todos={initTodo}></Header>
      <BrowserRouter>
        <Switch>
          <Route exact path="/" component={Todos} />
          <Route exact path="/about" component={About} />
        </Switch>
      </BrowserRouter>
    </div>
  );
}

export default App;
      
```

```

        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

```

File Edit Selection View Go Run Terminal Help Header.js - todos-list - Visual Studio Code
EXPLORER ... JS App.js M JS Header.js U X JS Todos.js U JS TodoItem.js U JS About.js U # App.css
src > MyComponents > JS Header.js > Header.js > constructor
1 import React from 'react'
2 import PropTypes from 'prop-types';
3 import {Link} from "react-router-dom";
4 export default function Header(props) {
5   return (
6     <nav className="navbar navbar-expand-lg navbar-light bg-light">
7       <div className="container-fluid">
8         <Link className="navbar-brand" to="/">{props.title} {props.myvar}</Link>
9         <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
10          <span className="navbar-toggler-icon"></span>
11        </button>
12        <div className="collapse navbar-collapse" id="navbarSupportedContent">
13          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
14            <li className="nav-item">
15              <a href="#" className="nav-link active" aria-current="page" to="/">Home</a>
16            </li>
17            <li className="nav-item">
18              <a href="#" className="nav-link" aria-current="page" to="/about">About</a>
19            </li>
</ul>
</div>
</div>
</nav>
)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

./src/MyComponents/Footer.js
Line 4:9: 'footerStyle' is assigned a value but never used [no-unused-vars](#)

Search for the [keywords](#) to learn more about each warning.
To ignore, add `// eslint-disable-next-line` to the line before.

Ln 17, Col 38 Spaces: 2 UTF-8 CRLF () JavaScript

```

File Edit Selection View Go Run Terminal Help Header.js - todos-list - Visual Studio Code
EXPLORER ... JS App.js M JS Header.js U X JS Todos.js U JS TodoItem.js U JS About.js U # App.css
src > MyComponents > JS Header.js > Header.js > constructor
3 import {Link} from "react-router-dom";
4 export default function Header(props) {
5   return (
6     <nav className="navbar navbar-expand-lg navbar-light bg-light">
7       <div className="container-fluid">
8         <Link className="navbar-brand" href="#">{props.title} {props.myvar}</Link>
9         <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
10          <span className="navbar-toggler-icon"></span>
11        </button>
12        <div className="collapse navbar-collapse" id="navbarSupportedContent">
13          <ul className="navbar-nav me-auto mb-2 mb-lg-0">
14            <li className="nav-item">
15              <a href="#" className="nav-link active" aria-current="page" href="#">Home</a>
16            </li>
17            <li className="nav-item">
18              <a href="#" className="nav-link active" aria-current="page" href="#">About</a>
19            </li>
</ul>
</div>
</div>
</nav>
)

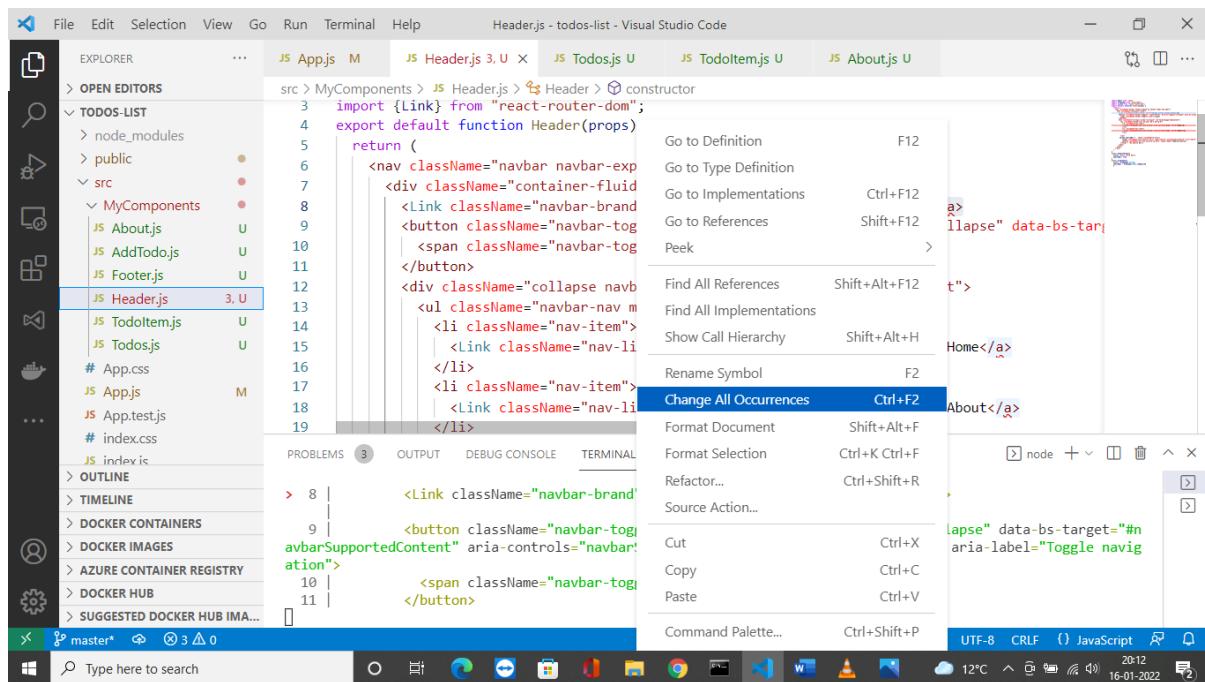
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

./src/MyComponents/Footer.js
Line 4:9: 'footerStyle' is assigned a value but never used [no-unused-vars](#)

Search for the [keywords](#) to learn more about each warning.
To ignore, add `// eslint-disable-next-line` to the line before.

Ln 15, Col 71 (5 selected) Spaces: 2 UTF-8 CRLF () JavaScript



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}</Link>
                {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>
                </div>
            </div>
        </nav>
    )
}

```

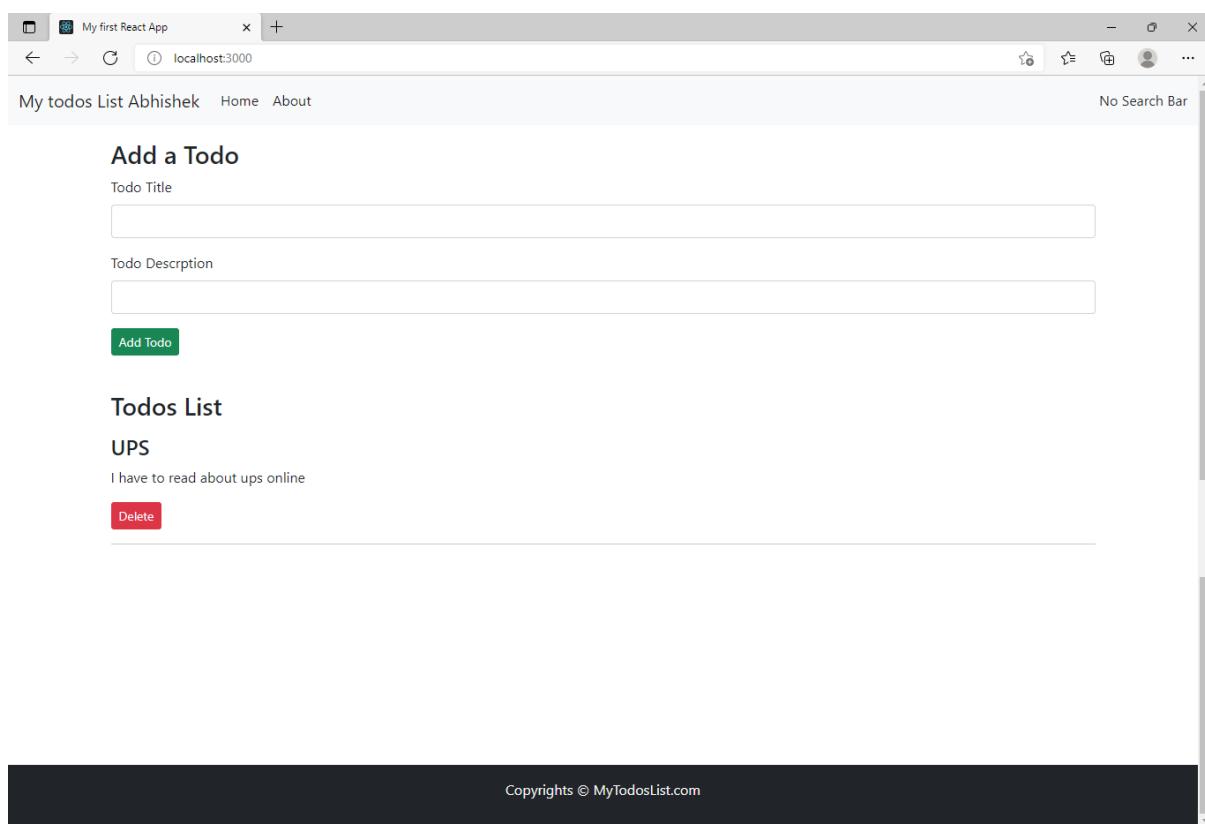
```

        {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

This is 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. Animi dolor proident qui excepteur officia. Officia pariatur excepteur culpa do ullamco exercitation do mollit laborum.

Copyrights © MyTodosList.com

Now you can also use one extension

Home > Extensions > React Developer Tools

React Developer Tools
Offered by: Facebook

★★★★★ 1,360 | Developer Tools | 3,000,000+ users

Add to Chrome

Overview Privacy practices Reviews Support Related

Now we have to BUILD PROJECT for production

npm run build

It will Creating an optimized production build..

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** App.js - todos-list - Visual Studio Code.
- Explorer:**
 - OPEN EDITORS: JS App.js (M), JS Header.js U, JS Todos.js U, JS TodoItem.js U, JS About.js U, # App.css.
 - TODOS-LIST: node_modules, public, src, MyComponents (About.js, AddTodo.js, Footer.js, Header.js, TodoItem.js, Todos.js).
 - OTHER: index.css, App.css, App.test.js, index.html.
- Code Editor:** The code for `App.js` is displayed, showing imports from `./App.css`, `./MyComponents/Header`, `./MyComponents/Footer`, `./MyComponents/TodoItem`, `./MyComponents/Todos`, `react`, `AddTodo`, and `About`. It also includes imports for `BrowserRouter`, `Switch`, and `Route` from `react-router-dom`.
- Terminal:**

```
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS F:\MY REACT\todos-list> npm run build
> todos-list@0.1.0 build
> react-scripts build
```

The screenshot shows the Visual Studio Code interface with the following details:

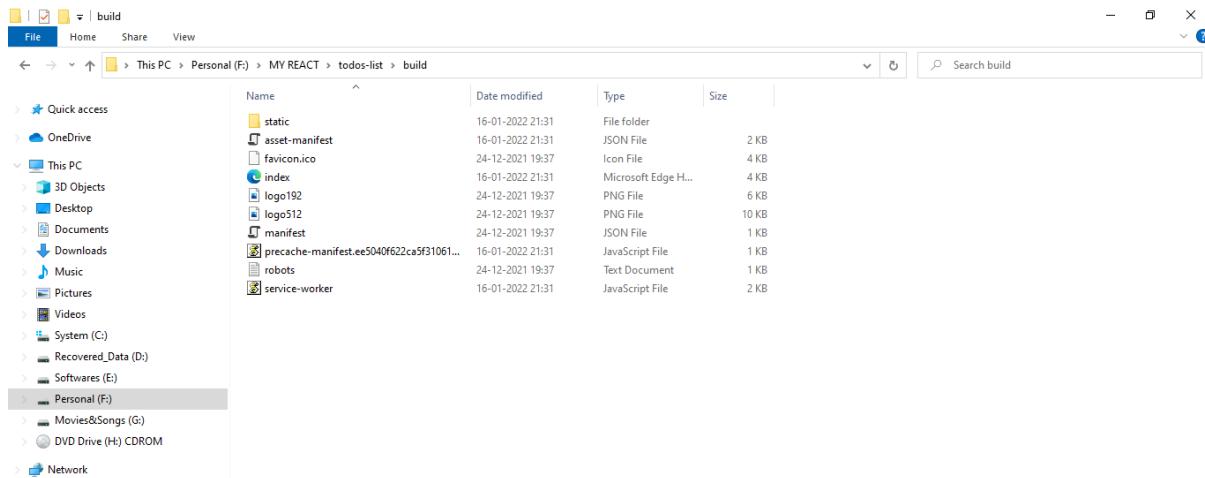
- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** App.js - todos-list - Visual Studio Code.
- Explorer:**
 - OPEN EDITORS: JS App.js (M), JS Header.js U, JS Todos.js U, JS TodoItem.js U, JS About.js U, # App.css.
 - TODOS-LIST: build (selected), static, asset-manifest.json, favicon.ico, index.html, logo192.png, logo512.png, manifest.json, precache-manifest.ee50..., robots.txt, service-worker.js.
 - public, src, MyComponents.
- Code Editor:** The code for `App.js` is displayed, identical to the previous screenshot.
- Terminal:**

```
1.17 KB build\static\js\runtime-main.8bc26d28.js
556 B build\static\css\main.d1b05096.chunk.css
```

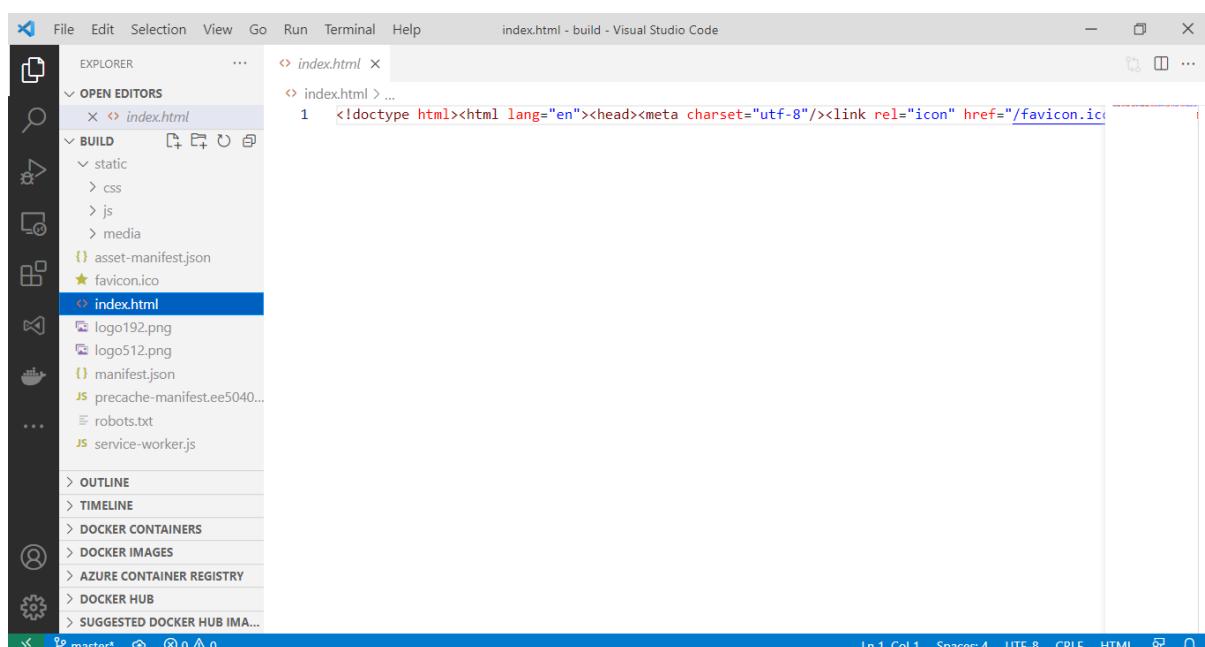
The project was built assuming it is hosted at `/`. You can control this with the `homepage` field in your `package.json`.

The `build` folder is ready to be deployed. You may serve it with a static server:

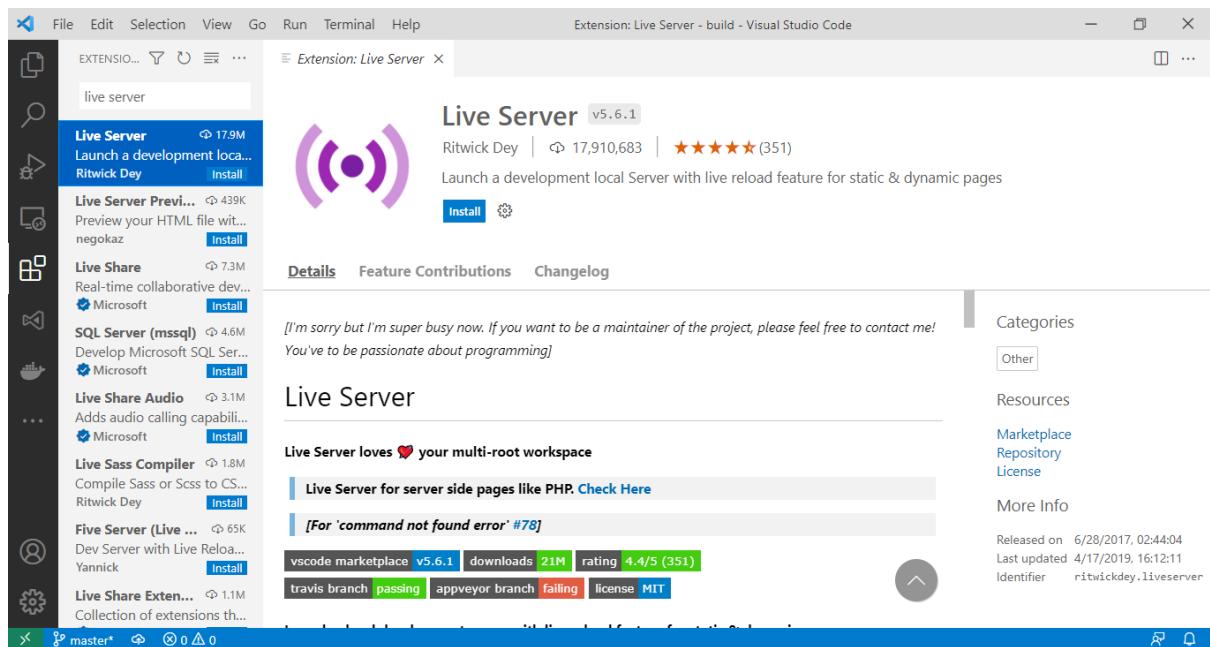
Now open the build folder in windows explorer



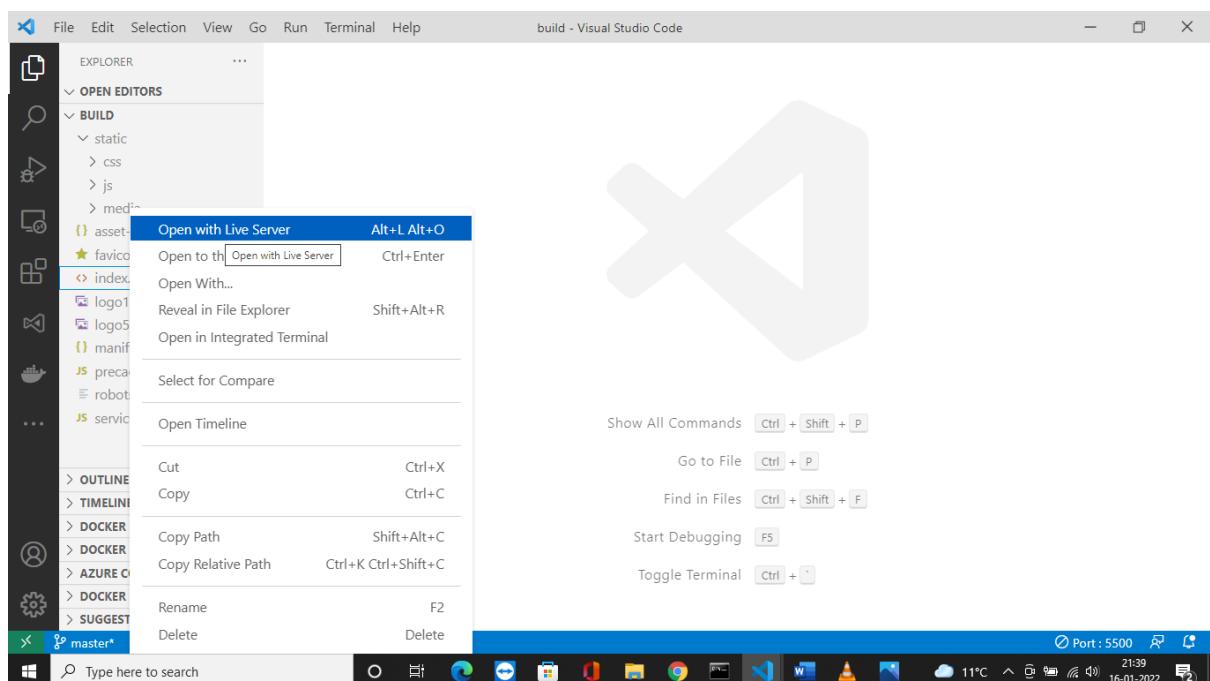
Now open this folder with VS Code



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



My todos List Abhishek Home About

Add a Todo

Todo Title

Todo Description

Add Todo

Todos List

UPS

I have to read about ups online

Delete

My todos List Abhishek Home About

This id an about component

Velit pariatur eiusmod cillum deserunt aliquip quis eu proident esse sint quis 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 volutate et nulla veniam exercitation nisi deserunt. Veniam dolore ullamco nisi pariatur Lorem volutate 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 volutate excepteur qui excepteur adipisicing laboris consectetur. Ea irure quis 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