Contents

Installation Process:

Step-1:

Download Node Js: https://nodejs.org/en/download/
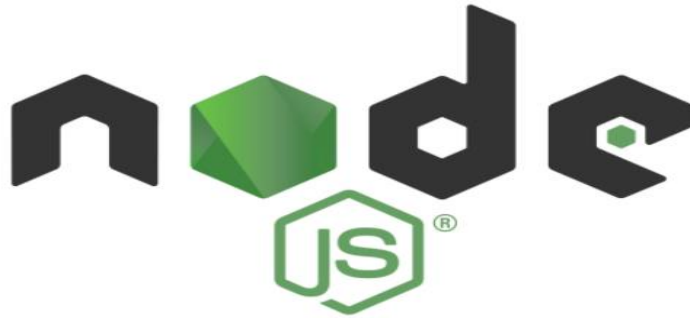
Note: Use node version latest then npx will work

   After Installing Check those commands:

   1.npm –version

   2.node -v

   3.npx --version

```
C:\Users\satya>npm --version
7.18.1
node -v
v12.20.1
npx--version
'npx--version' is not recognized as an internal or external command,
operable program or batch file.
npx --version
7.18.1
```

Step-2 :

Then Create your First React App. First Choose your folder then click   SHIFT +Right Click then Poweshell window will be open .See the below image

```
Windows PowerShell                                              —   □   ×
PS C:\Workspace\SoumyaDELL\New folder>
```
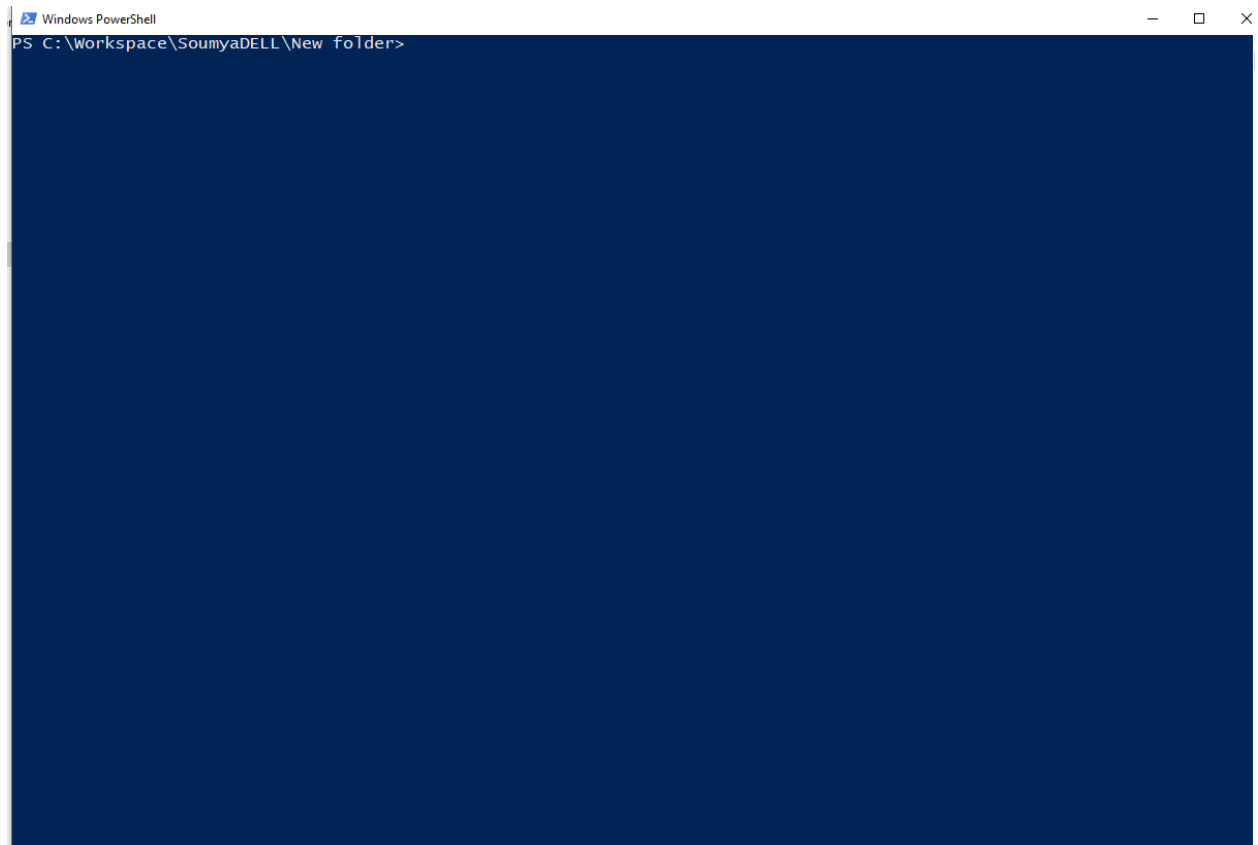
Step-3 :

Use this command to create your skeleton react app: npx create-react-app <mark>my-app</mark> (App Name)

After installing all the packages it will show like this. See below images

Error :

If any error comes like this :

You are running `create-react-app` 5.0.0, which is behind the latest release (5.0.1).

We no longer support global installation of Create React App.

Please remove any global installs with one of the following commands:

- npm uninstall -g create-react-app

- yarn global remove create-react-app

The latest instructions for creating a new app can be found here:

https://create-react-app.dev/docs/getting-started/

Then you need to follow this command to resolve the issue

1. npm uninstall -g create-react-app

2. npm install -g create-react-app

3. npm cache clean -f

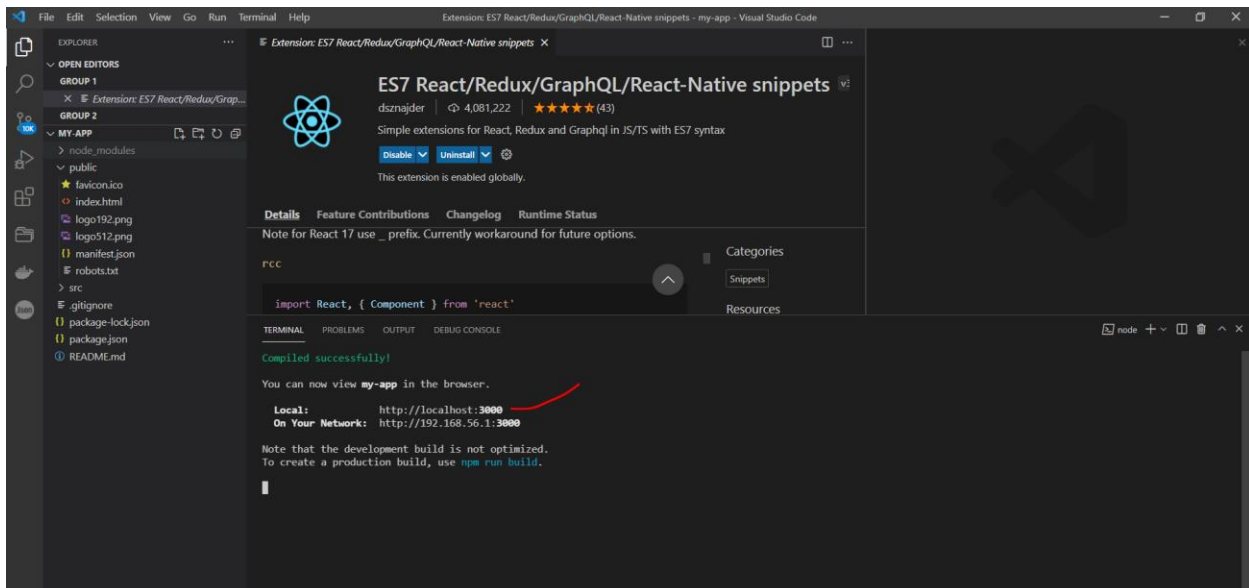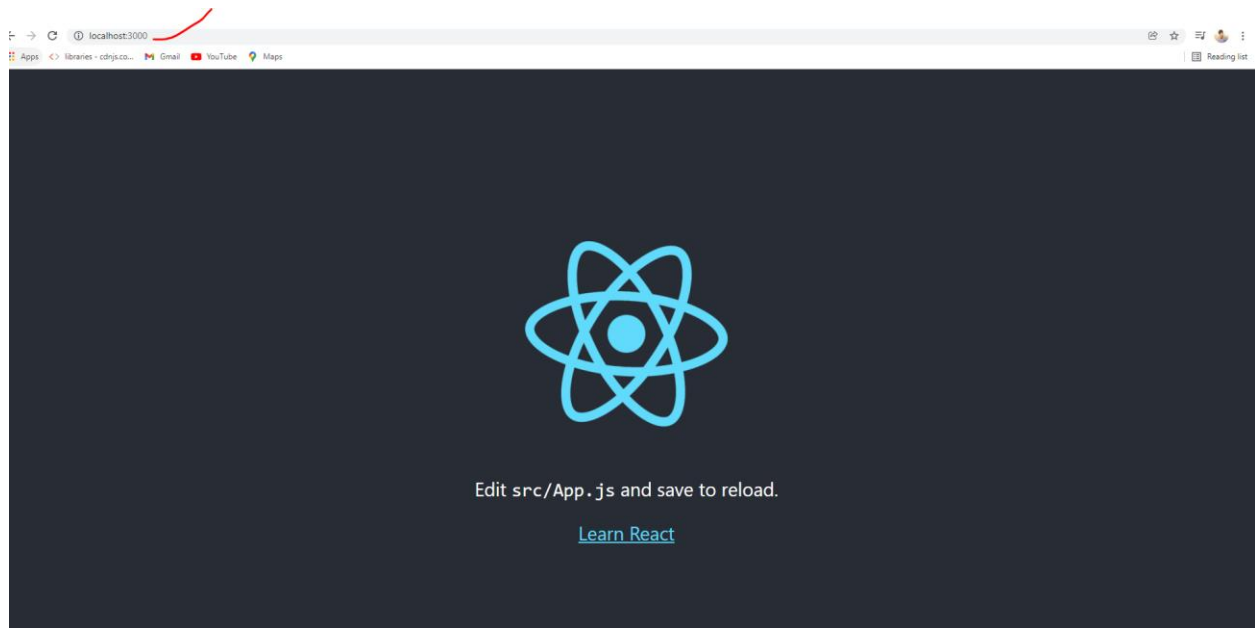*4. npx create-react-app my-app*

See the below images:

Step-4:

Needs to start (build) the application Use this command : npm start.

After successfully build it will open automatically and run on http://localhost:3000 PORT.

Edit src/App.js and save to reload.

Learn React

What is JSX?

JSX stands for JavaScript XML.

JSX allows us to write HTML in React.

JSX makes it easier to write and add HTML in React.
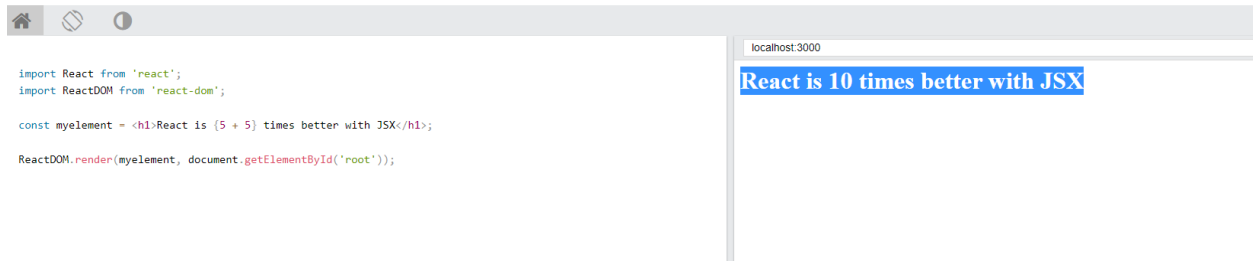
Coding JSX

JSX allows us to write HTML elements in JavaScript and place them in the DOM .

JSX converts HTML tags into react elements.

Expressions in JSX

With JSX you can write expressions inside curly braces { }.

The expression can be a React variable, or property, or any other valid JavaScript expression. JSX will execute the expression and return the result:

```
import React from 'react';
import ReactDOM from 'react-dom';

const myelement = <h1>React is {5 + 5} times better with JSX</h1>;

ReactDOM.render(myelement, document.getElementById('root'));
```

localhost:3000

React is 10 times better with JSX

Elements Must be Closed

JSX follows XML rules, and therefore HTML elements must be properly closed.

Example

Close empty elements with `/>`

```
import React from 'react';
import ReactDOM from 'react-dom';

const myelement = <input type="text" />;

ReactDOM.render(myelement, document.getElementById('root'));
```

React Components

When creating a React component, the component's name MUST start with an upper case letter.

Class Component

A class component must include the `extends React.Component` statement. This statement creates an inheritance to React.Component, and gives your component access to React.Component's functions.

The component also requires a `render()` method, this method returns HTML.

```
class Car extends React.Component {
  render() {
    return <h2>Hi, I am a Car!</h2>;
  }
}
```

Function Component

Here is the same example as above, but created using a Function component instead.

A Function component also returns HTML, and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand, and will be preferred in this tutorial.

Example

Create a Function component called `Car`

```
function Car() {
  return <h2>Hi, I am a Car!</h2>;
}
```

Rendering a Component

Now your React application has a component called Car, which returns an `<h2>` element.

*To use this component in your application, use similar syntax as normal HTML: `<Car />`*

Routing:

What is React-Router?

Many modern websites are actually made up of a single page, they just look like multiple pages because they contain components which render like separate pages.

*These are usually referred to as SPAs - **s**ingle-**p**age **a**pplications. At its core, what React Router does is conditionally render certain components to display depending on the route being used in the URL (`/` for the home page, `/about` for the about page, etc.).*

React Lifecycle:
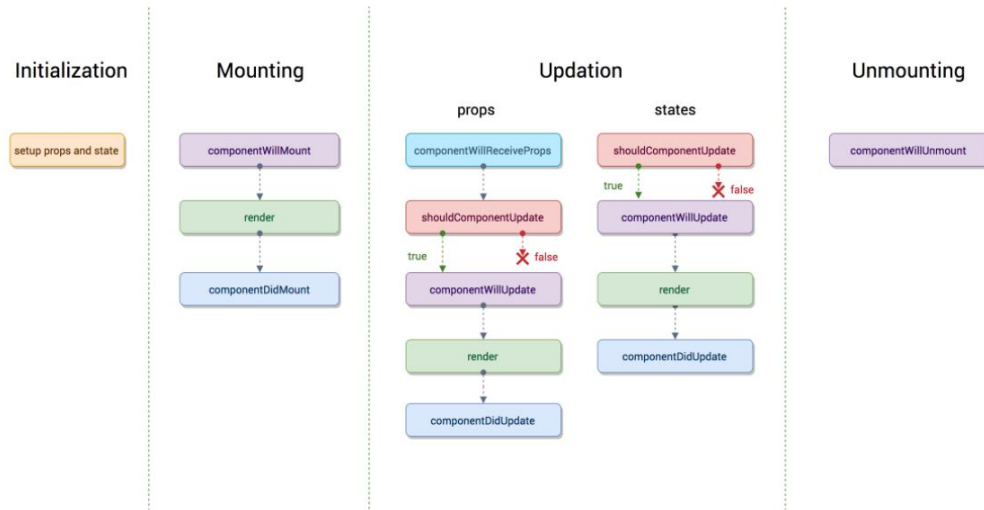
ReactJs lifecycle phases and methods