Department of Artificial Intelligence and Machine Learning
B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)

Experiment 1

Name: Shubham Mourya **SAP ID: 60017230110**

Name:- Shubham Mourya
AIML
60017230110 A06T

Experiment No:- 1

Aim:- Installation and Configuration of React.

Theory:-
ReactJS is popular javascript library developed by Facebook for building UI, particularly for single-page applications.

React is based on components, which are independent, reusable pieces of code that manage their own state.

React also emphasizes unidirectional data flow easier to understand & debug. State and props are two components, making the data flow easier to understand & debug.

* Installation of React JS

① Open command prompt & navigate to project directory

② Initialize Node.js using npm init. This will create package.json

③ Install React & ReactDOM
↳ npm install --save react
↳ npm install --save react-dom

④ Alternately you can use the Create React App tool to setup new React Project
npx create-react-app my-app.

⑤ Navigate to project directory cd myapp

⑥ npm start to start deployment server & browser opens your application at http://localhost:3000

Adding a Component

① Create a new file name MyComponent.js in src folder.

```
import React from 'react'
const MyComponent = () => {
    return ( <div>
        <h1> Hello World </h1>
      </div>
    );
  };
export default MyComponent;
```

② Import & use this component in App.js

```
import React from 'react'
import MyComponent from './MyComponent';
const App = () => {
    return ( <div>
        <MyComponent />
      </div>
    );
};  export default App;
```

Conclusion:- This experiment helped in understanding installation & configuration of React on local machine.

Sundaram

23/08

Department of Artificial Intelligence and Machine Learning
B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)

Date:	**1/8/2024**
Aim	**Installation and Configuration of React.**
Software	1) Node.js 2) At least one installed code Editor to test and debug your code e.g. • Atom • Sublime • Visual studio code
Pre-requisite	Active internet connection
Theory	ReactJS is a library written in TypeScript. It utilises the syntax of the modern version of JavaScript as described by ES6 and its higher version. Applications built using ReactJS use the Single reusability principle. This advocates the idea of building web pages and applications using components and unidirectional flow. In React we have the concept of states and the concept of immutability. Components have hierarchy in terms of Parent and Child components. A component in case of React can be thought of as a piece of code which is based on the principle of pure functions. We will look into the pure component later. First, let's understand what a state is. For e.g. To become a member of a service, the user fills his information on the registration page. While filling the details there can be many states of the form, for e.g. When the form field is empty or when the form field has witnessed some error on some particular field, which needs to be corrected; or when after correction, the form data has been validated and is ready for submission. So, at a broad level, the registration form has been through various states. These states represent at which level the application is, in terms of interacting with the end-user. Each level of interaction for this form is represented by the state, from an empty form to being a fully filled form with a display of an error for certain fields and the validated form. In React, we have the component based on the pure function. A pure function can be memorised as a piece of code, which does one task and does it pretty well. For a certain input, it always returns the same output, so this means we are increasing predictability of the code. Since React.js follows a certain code pattern and principles in order to work, it lowers the curve of the knowledge gap; whether you are one-person or a team of developers working mutually.

	Introduction to Node.js and NPM on Windows 10 To run ReactJS we will require Node.js on our system.Node.js is a server which will help us to run the React code. It is based on non-blocking input and output and the Chrome V8 JavaScript engine. The Node.js code is open source. NPM which is an abbreviation of Node package manager, npmjs.com is supported by various developers around the world. It has various node modules, using which developers can host and publish their modules on the open-source community. It hosts modules in private and public visibility. A module carries code which exists to serve high or low level functionalities. In terms of code adoption and availability of various modules it gives an edge and tries to make the developer more productive. We can plug in and plug out the module. Some modules are dependent on other modules; which is defined as dependency among modules. While building an application, a developer can pick the module, tweak and remix it to suit the application needs, and can then release to the open-source community. So, instead of reinventing the wheel, it is like picking a wheel (npm module) from npmjs.com, giving it further momentum and giving it back to the open source community. Download and Install Node.js To install Node.js we need to go to the URL

Depending upon our Windows OS version in terms of 32 Bit or 64 Bit, we can pick the installer and install that version.

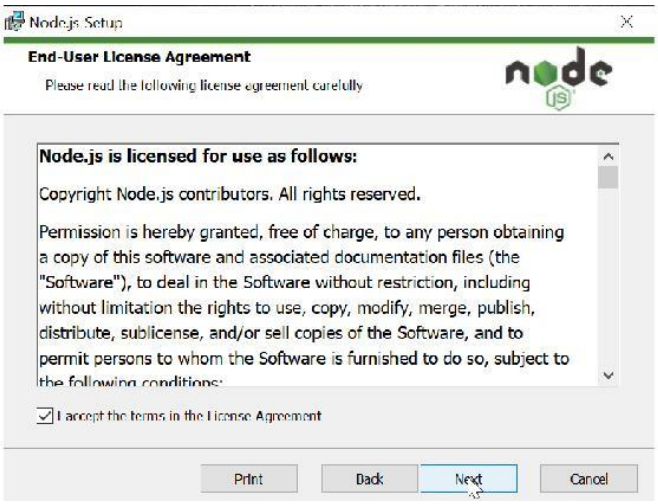
The Node.js installer includes NPM. It is best to install the even numbered version of NPM.

Depending upon your operating system, the Nodejs.org home page will show the Download button and recommended LTS version.

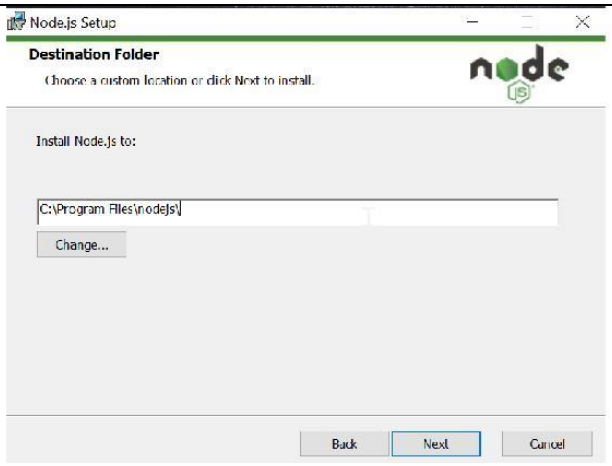
After the download is complete we will go to the downloads folder and run the installer.

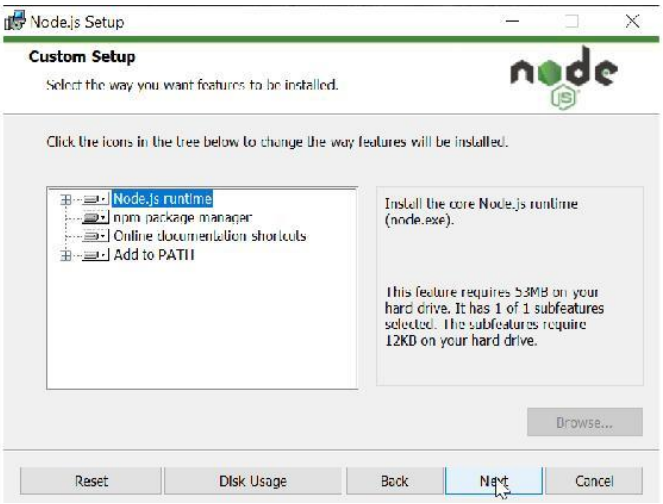
The installer will show the below Setup Wizard. Click next.

The next screen will ask for the End-user License Agreement. Select the checkbox at the bottom right to provide your consent and click on Next to proceed with the installation.



The installer will ask for Destination folder and the default path set by installation is C:\Program Files\nodejs\
Click on Next button

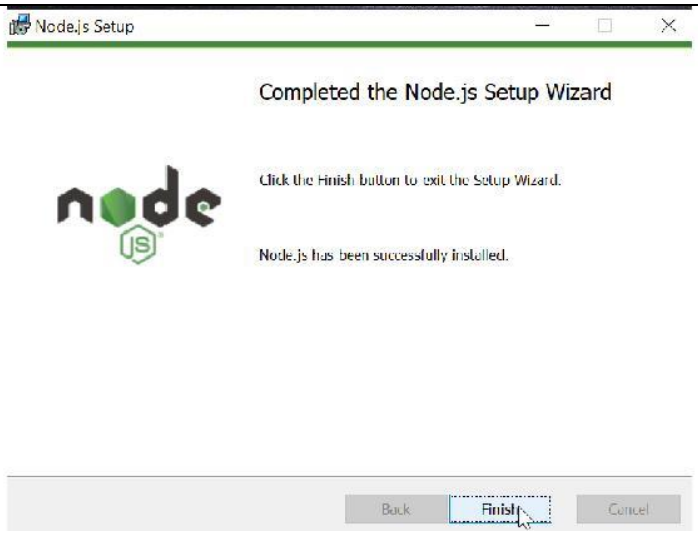




The above screen is an important step in the installation process. And if you see closely it also sets the environmental path variables to command prompt on Windows. To begin your journey in web development.

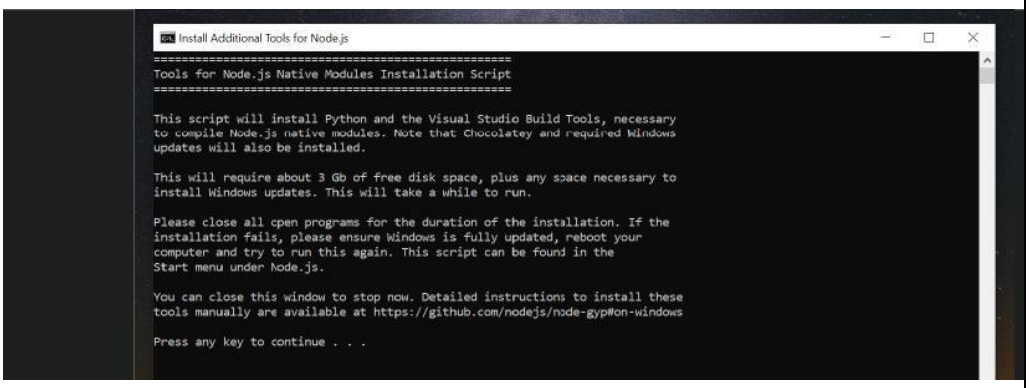
Click on Next to continue with the installation.

The Windows OS may ask you to allow Node.js installation and make changes.Click on Yes button.



During the installation, if you have allowed for Chocolatey and required modules installation for C++ and Python, you will see the UI below in the command prompt. This installation requires 3 Gb of free disk space. In this tutorial this step is not required, so we are skipping this step by closing the window.

If you are interested in installing it, press Enter to continue.



Once the installation is complete you need to verify the Node.js installation.

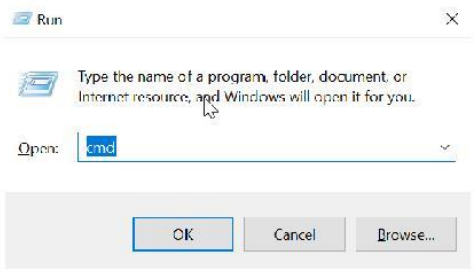
For this, we will use the command prompt.

To run command prompt

Press keys Win+R

And type cmd in the window below.

Next Click on Ok or Press Enter on the keyboard.



Installation of React

After installation of Node.js, we need to install React. To check the Node.js version, open the Windows command prompt.

Press Win+R and type cmd.

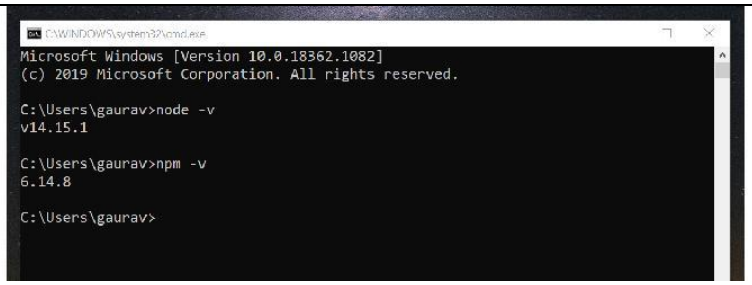
In the command line, type

node -v to see its version.

We can also check for npm version, which is installed with Node.js, with the following command

npm -v

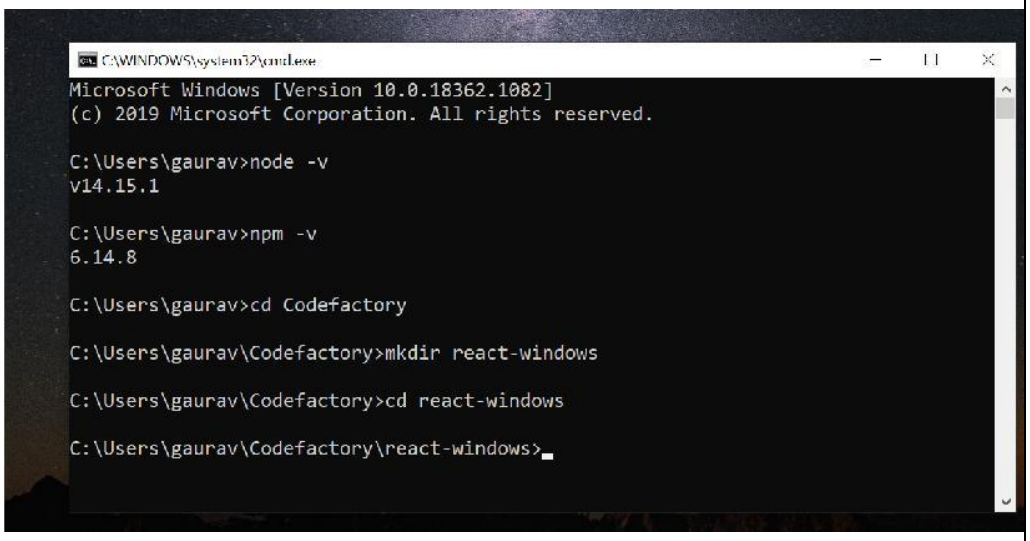
After running these commands, we can check the node version v14.15.1 and npm version 6.14.8



As we have confirmed the Node.js installation we can proceed to the next steps.

While in the command prompt, we have navigated to a folder called Codefactory by following the command cd Codefactory

In this folder, we have created a folder called react-windows by using the command mkdir react-windows.



After the folder react-windows has been created, we will change the directory to react-windows with the command

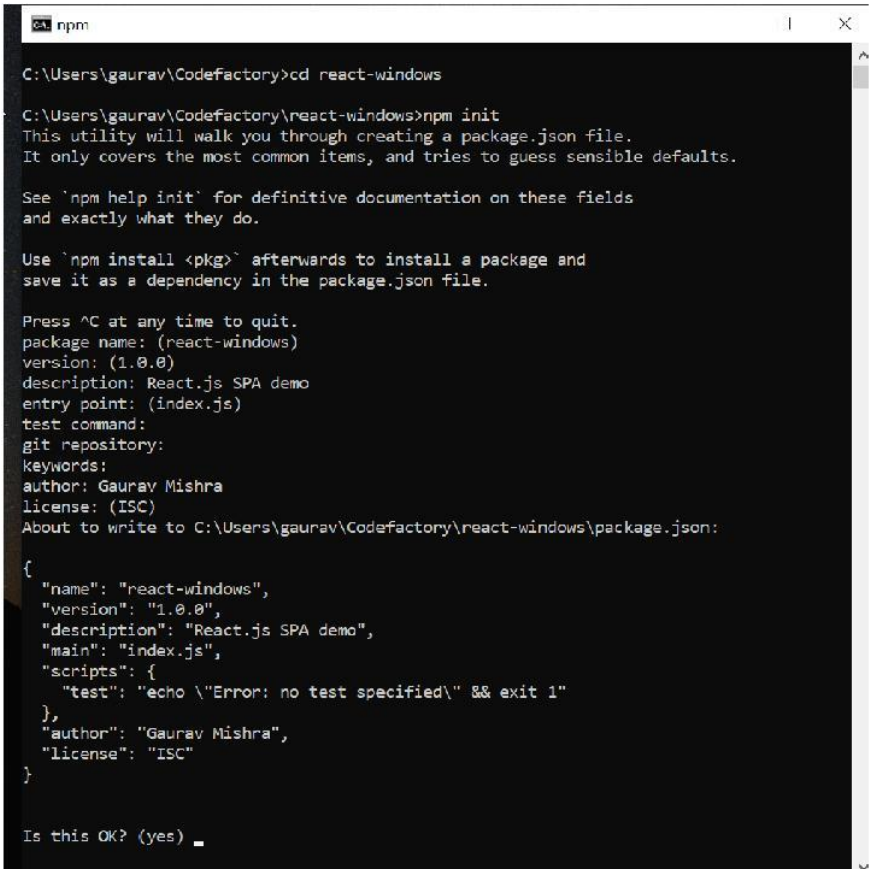
cd react-windows

ReactJS can be installed in various ways.

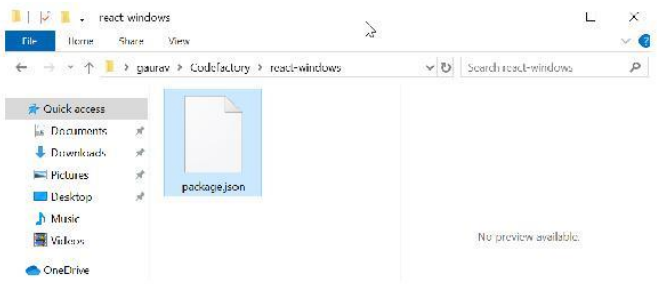
Now, we will type npm init. It will ask for the below configuration line by line.

Insert your input, followed by Enter keypress to proceed with the next configuration.

At the end of the configuration it will confirm for the inputs you have entered. If you are happy with the configuration data, type yes and enter to continue.

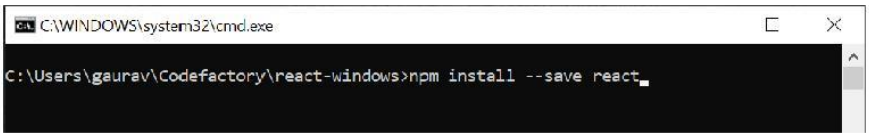


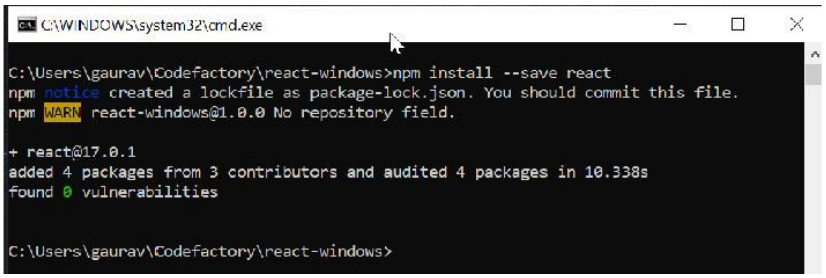
The npm init will help us to create a package.json file.



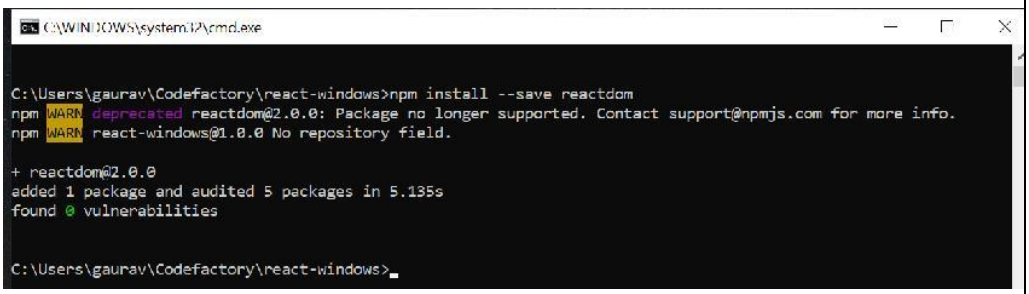
Now, the next step to install React requires us to go to the command prompt and type the following command in the react-windows directory.

npm install --save react



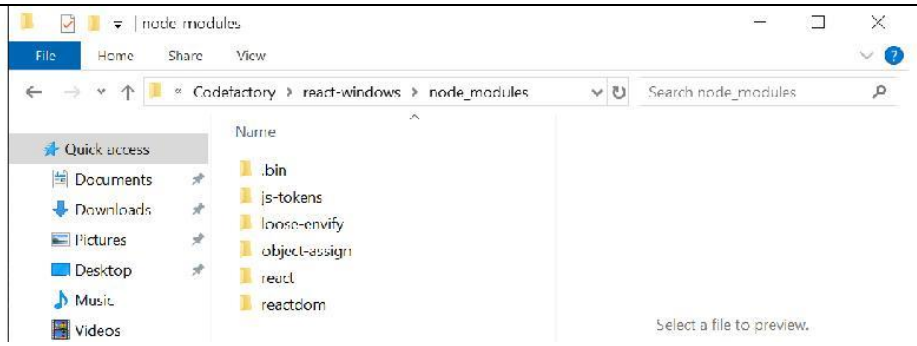


And after the above command **npm install --save react-dom**



Behind the scene, these commands fetch the specified module from npmjs.com and download it in the local codebase.

Let's have a look at the react-windows folder. Here we can see some newly created directories in node_modules.



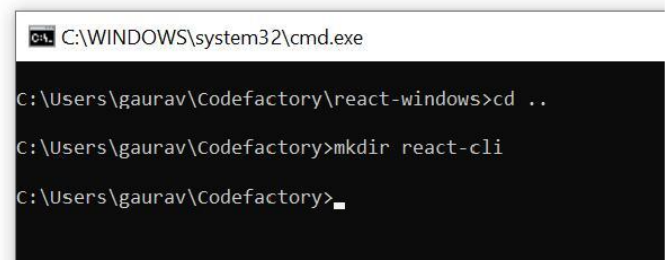
So, in this tutorial, we have learned to install React and reactDOM.

But to see the ReactJS SPA (single page app) there is more work to be done in the above code.

As an alternative and fast approach we can do it via create-react-app

Let us move to Codefactory folder and with the command **cd..** create another folder **react-cli**

Next, type the following command **mkdir react-cli**



Now we will use create-react-app module and type the following command

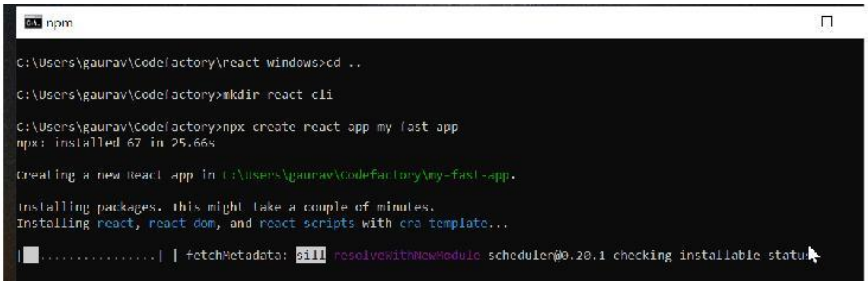
Please note that **my-fast-app** is the name of your app. This is an example and you can be creative in choosing your own name.

npx create-react-app my-fast-app

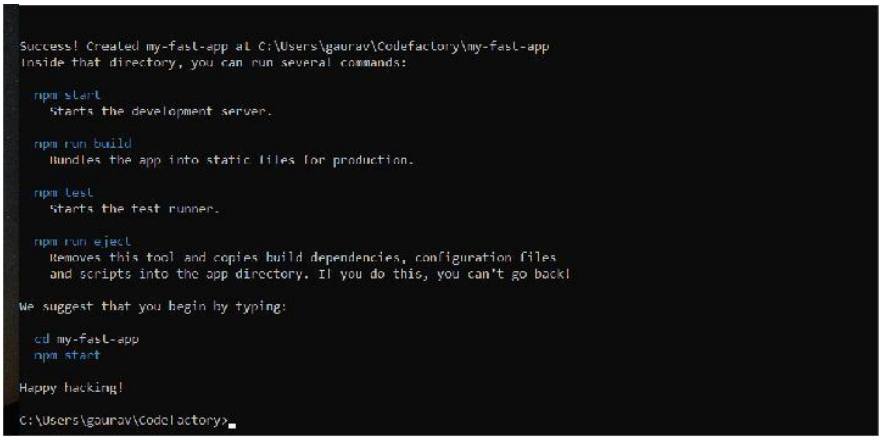
Copy Code

If we see closely it will take care of the rest of the installation steps, and react, react-dom and other related modules are installed automatically.

This process is a little data intensive, so please be patient while the download and installation happens



When the above step gets completed the command prompt displays the below output.

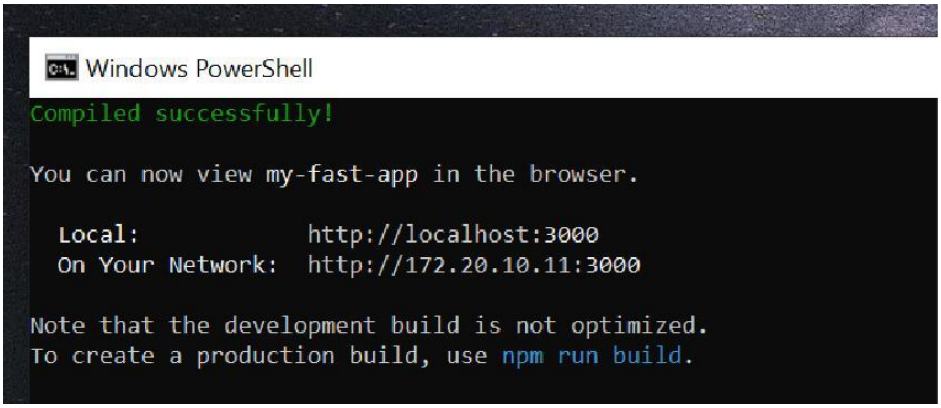


Now, let us run our first react app, by navigating to my-fast-app directory as below

cd my-fast-app

Copy Code

And enter the next command as **npm start**

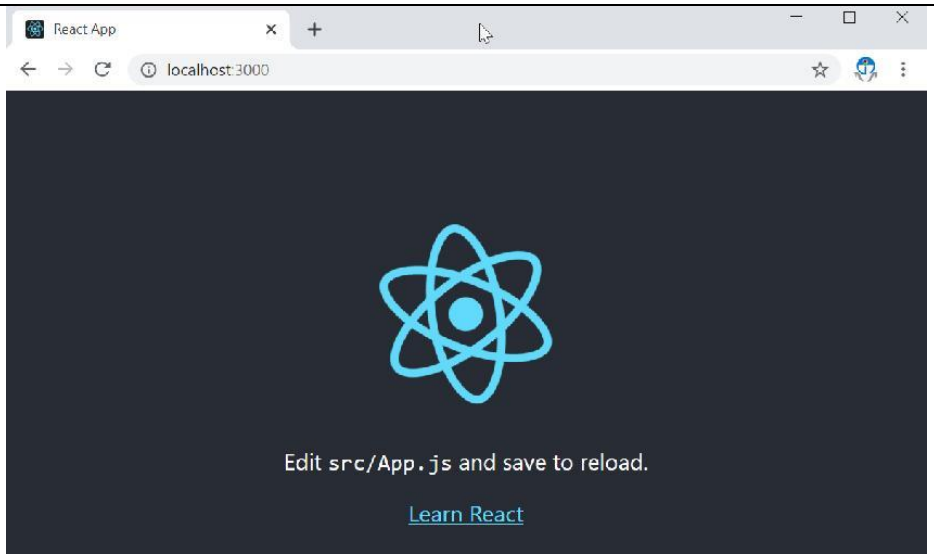


The npm command will show the application in the browser. http://localhost:3000

And if you are running node.js for the first time using npm command, it will ask for permission to allow access and we need to allow access to run.

As we are using a code editor we can have a look at the directory structure and some of the important files, such as index.html in the public folder, in src folder the index.js and App.js. The src folder contains the react component which we can build further on this codebase.

index.js is the js invocation point for react app.This index.js is linked with the App.js, which is responsible for showing the content in the browser. That's what we see on the demo page.

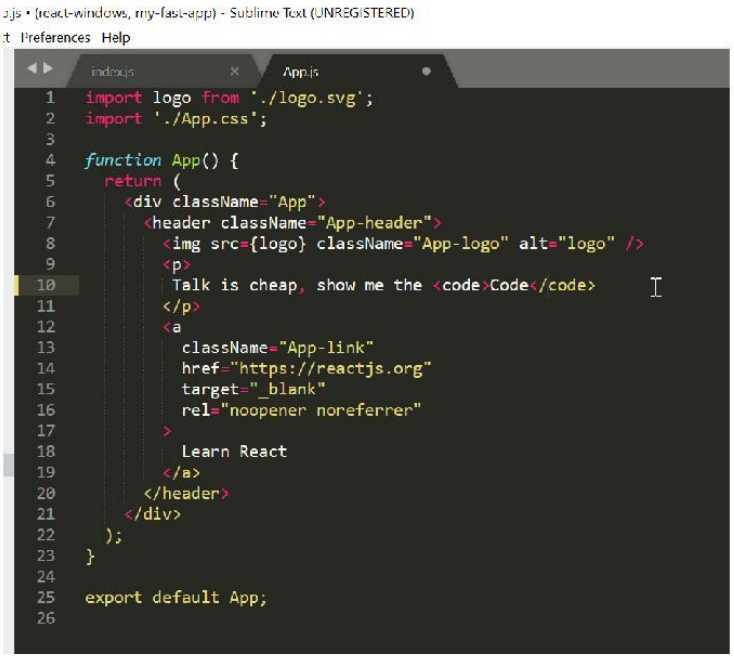


Let's edit the App.js by going to line 10 and adding the following code

Talk is cheap, show me the <code>Code</code>

Once you save the file by Ctrl+S

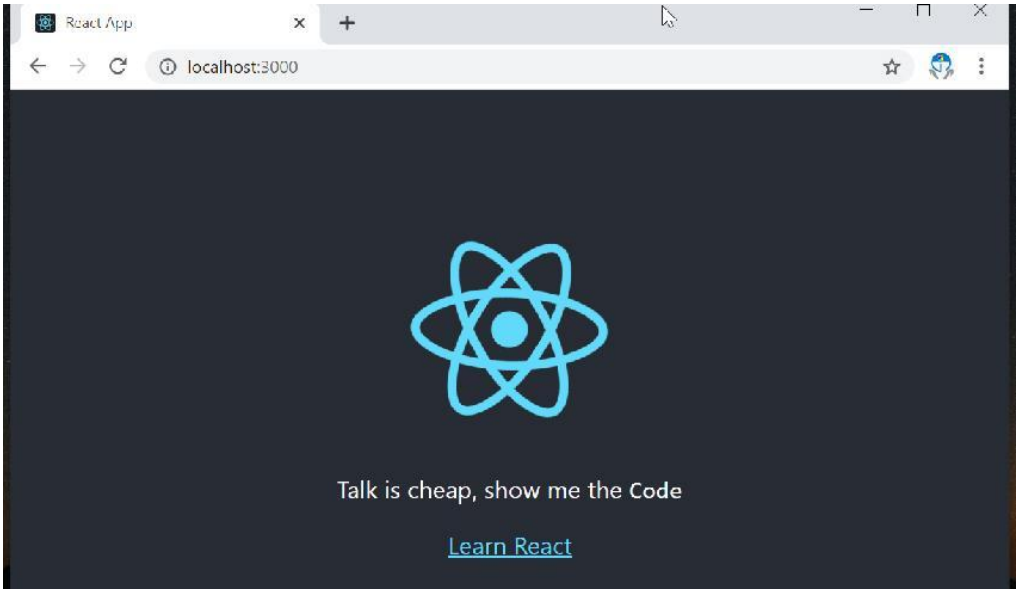
The code will be auto refreshed in the browser, after compiling.



It will show us the following output.

So, now feel free to change messages and alter the page layout and structure for experimentation.

If you are familiar with CSS, you may also change the page style using App.css and tinker with the code as well.



How to Test if React.js Installation is Properly Done?

After successfully completing react installation in windows machine, it's crucial to ensure that the installation is error-free and that the framework functions as expected. Here's a simple yet effective method to test the proper installation of React.js.

Step 1: Open a Command Prompt or Terminal

Ensure you have an open command-line interface, such as Command Prompt or PowerShell on Windows or Terminal on macOS/Linux.

Step 2: Check the React.js Version

Run the following command to display the installed version of React:

```
npx create-react-app --version
```

Copy Code

This command uses create-react-app to check the version. Since create-react-app is a part of the React.js ecosystem, its version corresponds to the installed React.js version.

Step 3: Verify Output

The command will output the version number of create-react-app. This version number is associated with the React.js installation. For example, the output might look like:

```
5.0.1
```

Copy Code

This indicates that React.js version 5.0.1 is successfully installed on your system.

By checking the version of create-react-app, you can easily verify the React.js version on your Windows machine. This straightforward method provides confidence in the correctness of your React.js installation and ensures you're equipped with the latest features and improvements.

How to Use React.js?

Now that React.js is successfully installed on your Windows machine, let's delve into how to use this powerful JavaScript library to build dynamic and interactive user interfaces. Below are essential steps, code snippets, and tips to kickstart your React.js development journey:

Step 1: Create a New React App

Initiate the creation of a new React app using the following command in your preferred command-line interface:

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

Copy Code

Replace "my-react-app" with your desired project name.

Step 2: Navigate to the App Directory

Move into the newly created app directory:

```
cd my-react-app
```

Copy Code

Step 3: Create a Simple React Component

Begin by creating a basic React component. Open your preferred code editor and create a new file, e.g., MyComponent.js. Define a functional component using the following code:

```
// MyComponent.js

import React from 'react';

const MyComponent = () => {
  return (
    <div>
      <h1>Hello, React!</h1>
      <p>This is my first React component.</p>
    </div>
  );
};
```

```
export default MyComponent;
```

Copy Code

Step 4: Use the React Component in App

Next, import and use the created component in your main App.js file:

```
// App.js

import React from 'react';

import MyComponent from './MyComponent';

const App = () => {

  return (

    <div>

<MyComponent />

    </div>

  );

};

export default App;
```

Copy Code

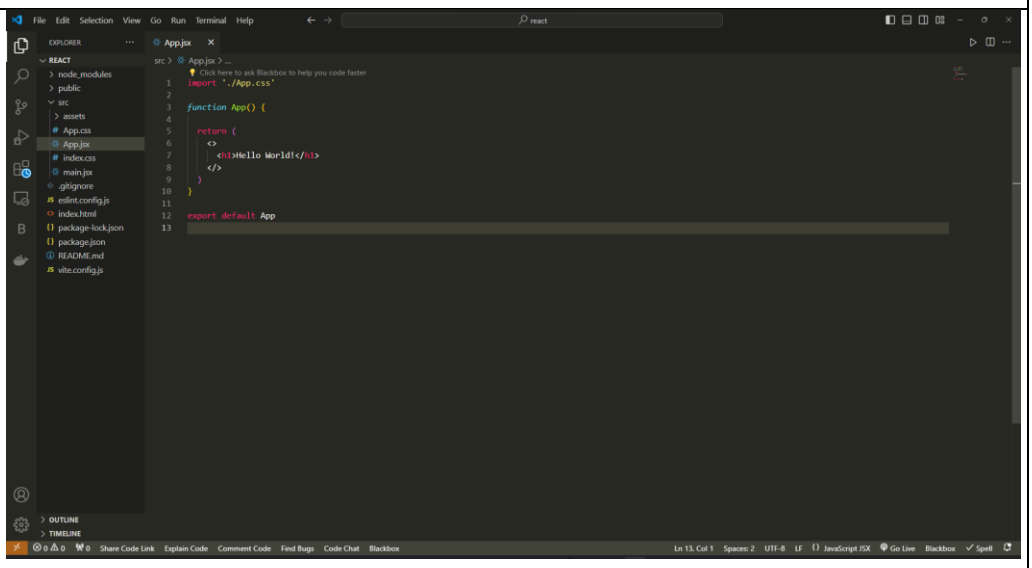
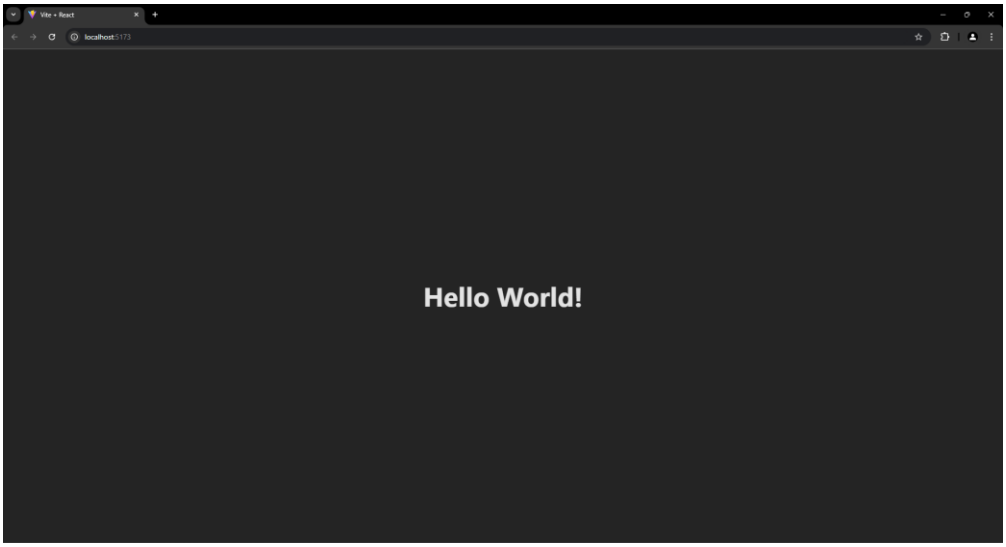
Step 5: Run the React App

Initiate the development server to see your React app in action:

```
npm start
```

Copy Code

This command will start the development server, and you should see a message indicating that the app is running on a specific port (usually 3000).

Result	 
Conclusion	We successfully installed Node.js and React.js on a Windows 10 machine, verified the installation, and created a simple React application. This process involved setting up Node.js and npm, installing React, and using `create-react-app` to start a new project. This setup provides a foundation for building dynamic web applications with React.