|  |  |
| --- | --- |
| Roll No. | A-75 |
| Student Name | Prasad Kashid |
| Subject | Internet Programming |
| Date Of Experiment |  |
| Date Of Submission |  |

# SUBJECT: INTERNET PROGRAMMING EXPERIMENT NO. 6

**AIM:** Installation React and Node JS

**OBJECTIVE:**

To orient students to React for developing front end applications To orient students to Node.js for developing backend applications

**LAB OUTCOMES:** LO5 & LO6

LO5 - Construct front end applications using React.

LO6 - Construct back end applications using Node.js/Express.

**THEORY:**

# React

React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It’s used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components. React was first created by Jordan Walke, a software engineer working for Facebook. React first deployed on Facebook’s newsfeed in 2011 and on Instagram.com in 2012.

React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application. This corresponds to the view in the MVC template. It can be used with a combination of other JavaScript libraries or frameworks, such as Angular JS in MVC.

React JS is also called simply to React or React.js.

There are so many open-source platforms for making the front-end web application development easier, like Angular. Let us take a quick look on the benefits of React over other competitive technologies or frameworks. With the front-end world- changing on a daily basis, it’s hard to devote time to learning a new framework – especially when that framework could ultimately become a dead end. So, if you're

looking for the next best thing but you're feeling a little bit lost in the framework jungle, I suggest checking out React.

## Simplicity

ReactJS is just simpler to grasp right away. The component-based approach, well- defined lifecycle, and use of just plain JavaScript make React very simple to learn, build a professional web (and mobile applications), and support it. React uses a special syntax called JSX which allows you to mix HTML with JavaScript. This is not a requirement; Developer can still write in plain JavaScript but JSX is much easier to use.

## Easy to learn

Anyone with a basic previous knowledge in programming can easily understand React while Angular and Ember are referred to as ‘Domain-specific Language’, implying that it is difficult to learn them. To react, you just need basic knowledge of CSS and HTML.

## Native Approach

React can be used to create mobile applications (React Native). And React is a diehard fan of reusability, meaning extensive code reusability is supported. So at the same time, we can make IOS, Android and Web applications.

## Data Binding

React uses one-way data binding and an application architecture called Flux controls the flow of data to components through one control point – the dispatcher. It's easier to debug self-contained components of large ReactJS apps.

## Performance

React does not offer any concept of a built-in container for dependency. You can use Browserify, Require JS, EcmaScript 6 modules which we can use via Babel, ReactJS-di to inject dependencies automatically.

## Testability

ReactJS applications are super easy to test. React views can be treated as functions of the state, so we can manipulate with the state we pass to the ReactJS view and take a look at the output and triggered actions, events, functions, etc.

# React Advantages

* + Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.
  + Can be used on client and server side as well as with other frameworks.
  + Component and data patterns improve readability, which helps to maintain larger apps.

# React Limitations

* + Covers only the view layer of the app, hence you still need to choose other technologies to get a complete tooling set for development.
  + Uses inline templating and JSX, which might seem awkward to some developers.

# React is popular!

* + Working with the DOM API is hard. React basically gives developers the ability to work with a virtual browser that is friendlier than the real browser.
  + React is just JavaScript. There is a very small API to learn, and after that your JavaScript skills are what make you a better React developer. There are no barriers to entry.
  + Learning React pays off big-time for iOS and Android mobile applications as well. React Native allows you to use your React skills to build native mobile applications. You can even share some logic between your web, iOS, and Android applications.
  + The React team at Facebook tests all improvements and new features that get introduced to React right there on facebook.com, which increases the trust in the library among the community. It’s rare to see big and serious bugs in React releases because they only get released after thorough production testing at Facebook.
  + Most importantly, React enables developers to declaratively describe their User Interfaces and model the state of those interfaces. This means instead of coming up with steps to describe transactions on interfaces, developers just describe the interfaces in terms of a final state (like a function). When transactions happen to that state, React takes care of updating the User Interfaces based on that.

# Node.js

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project! Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant. A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles

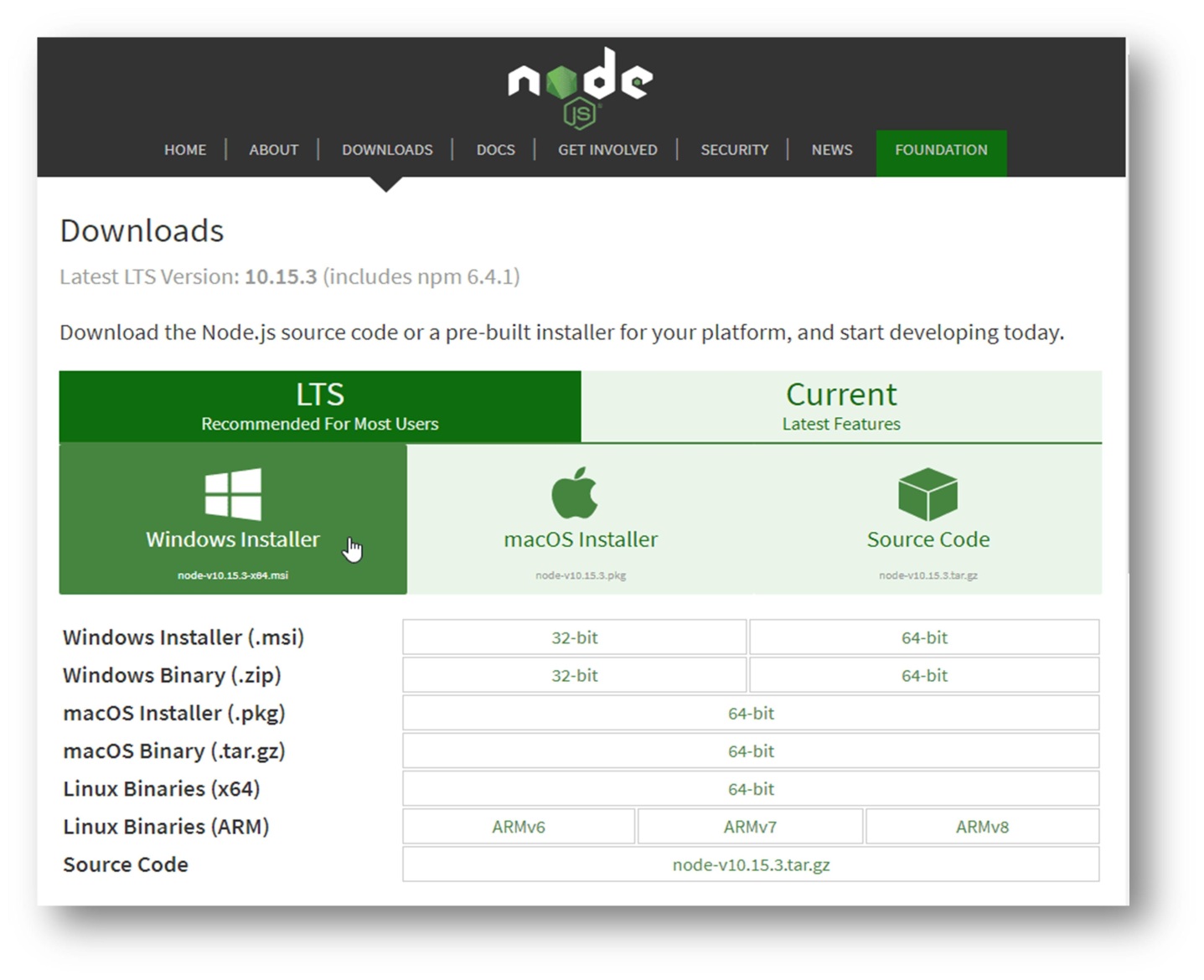
waiting, Node.js will resume the operations when the response comes back. This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to write the server-side code in addition to the client-side code without the need to learn a completely different language.

In Node.js the new ECMAScript standards can be used without problems, as you don't have to wait for all your users to update their browsers - you are in charge of deciding which ECMAScript version to use by changing the Node.js version, and you can also enable specific experimental features by running Node.js with flags.

# Installation of Node.js:

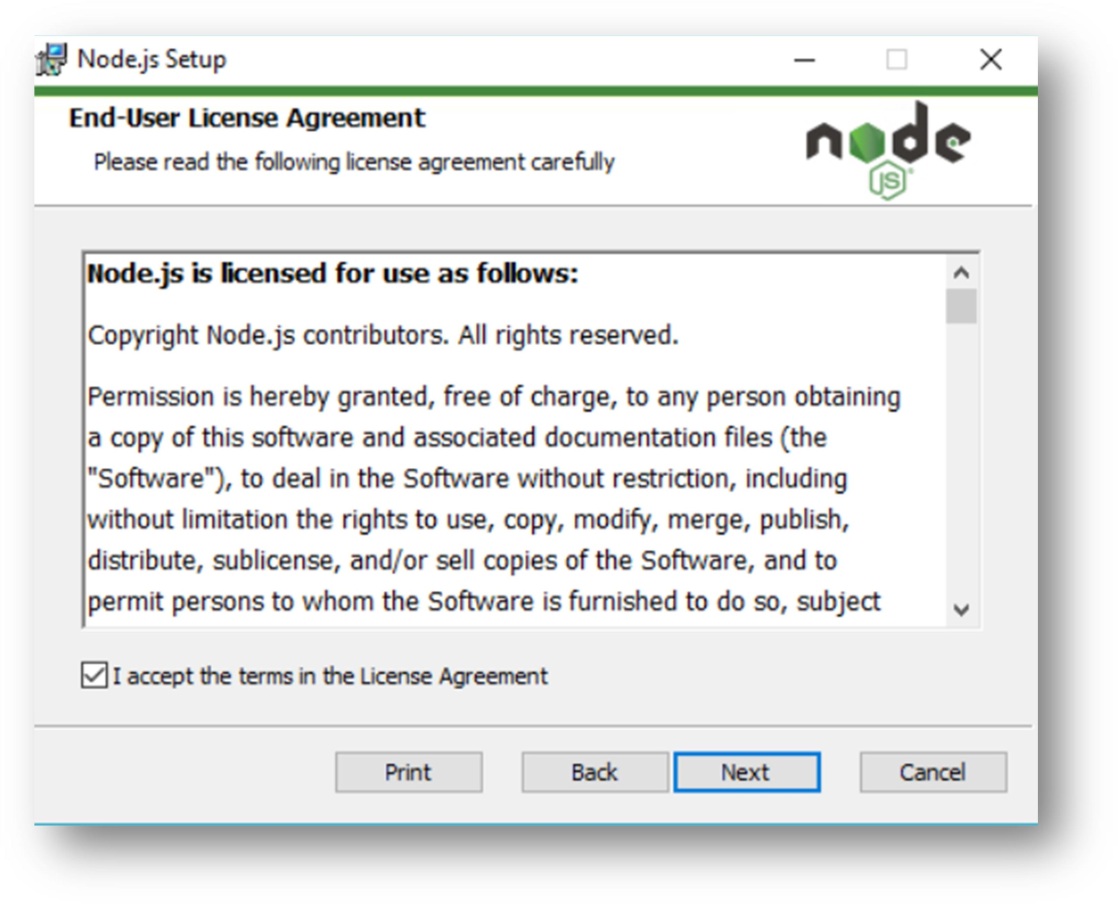
Download Node.js from https://nodejs.org/en/download/.

The first step is to download the Node.js installer for Windows. Let’s use the latest Long Term Support (LTS) version for Windows and choose the 64-bit version, using the Windows Installer icon.

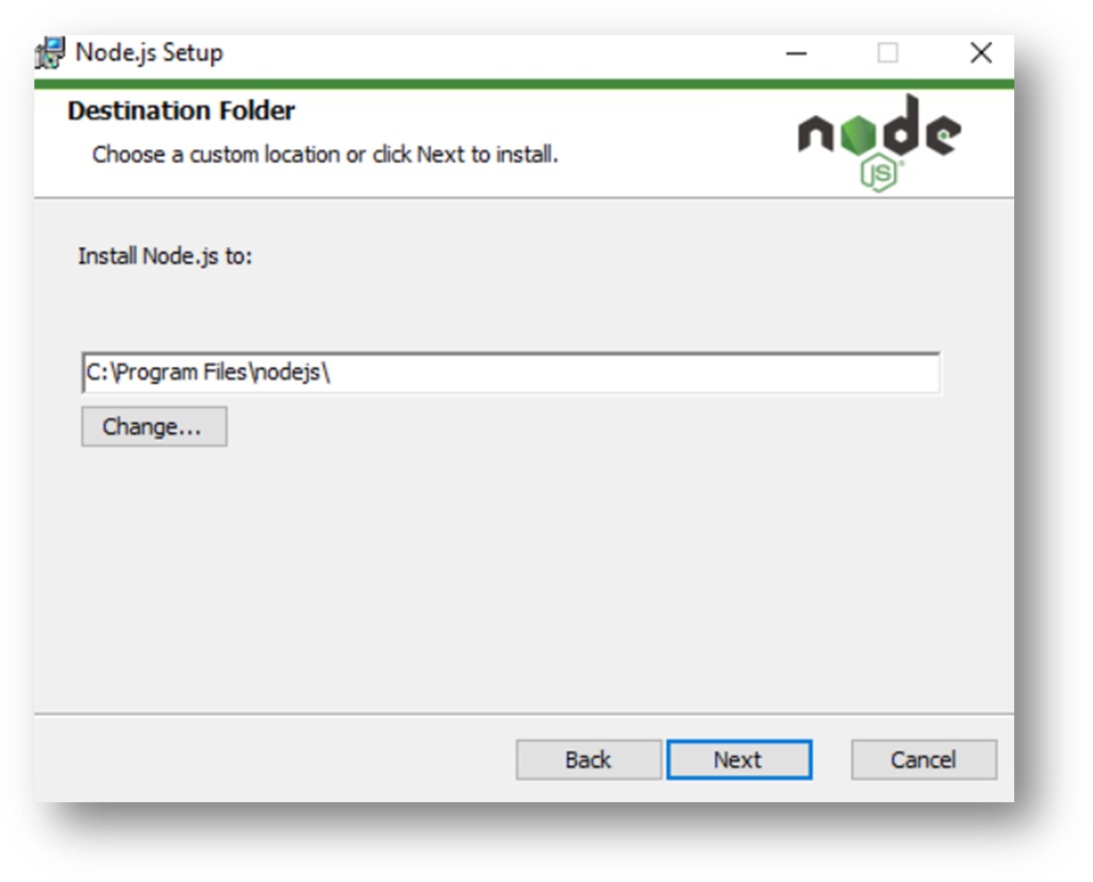
After downloading follow the below steps to install node.js



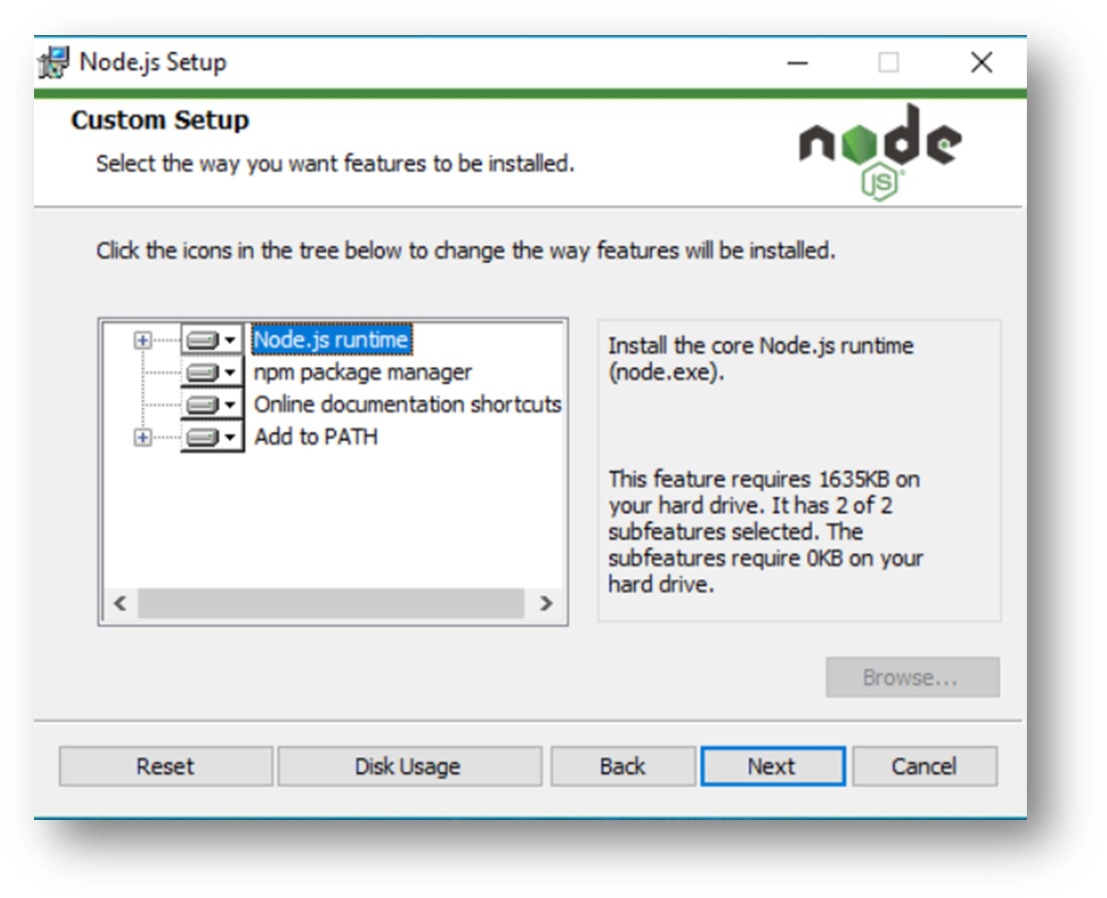
Click Next and proceed further for accepting terms and conditions



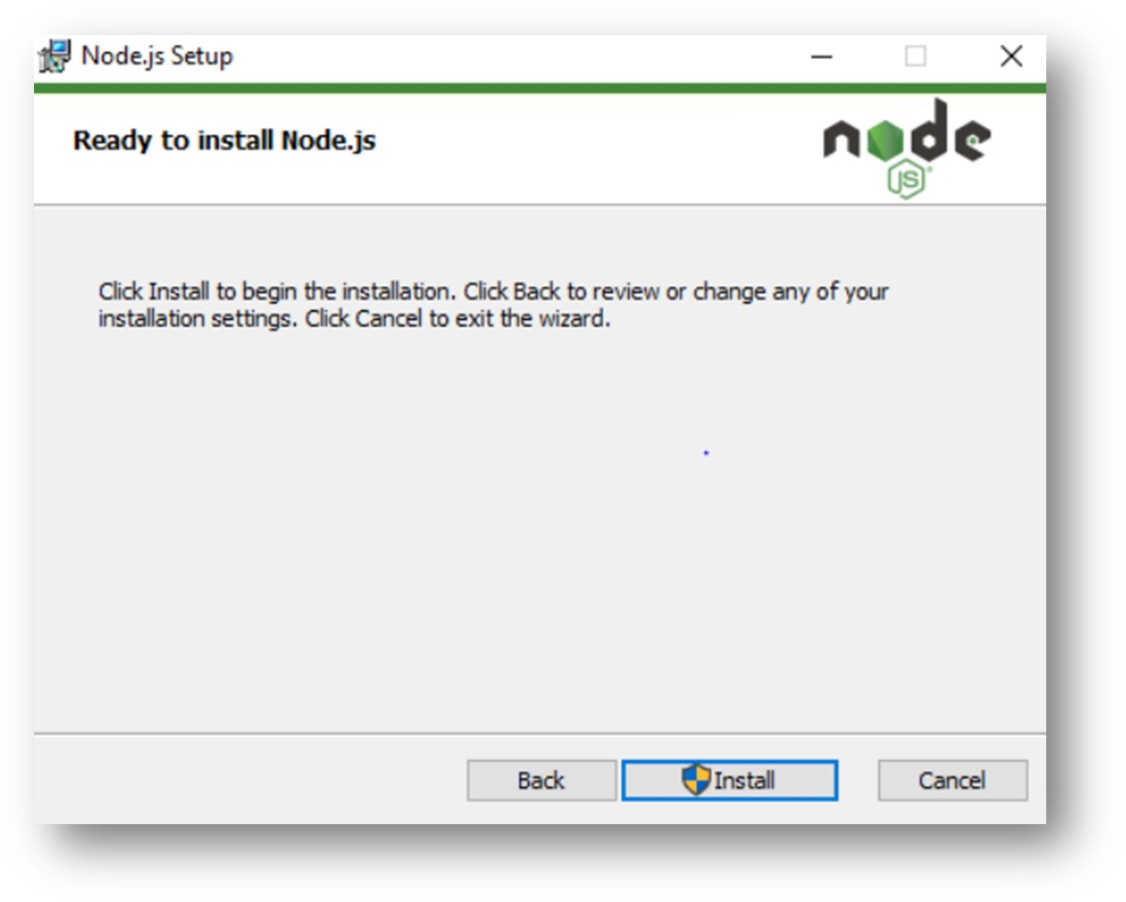
Click Next after checking for terms and conditions in the License Agreement.

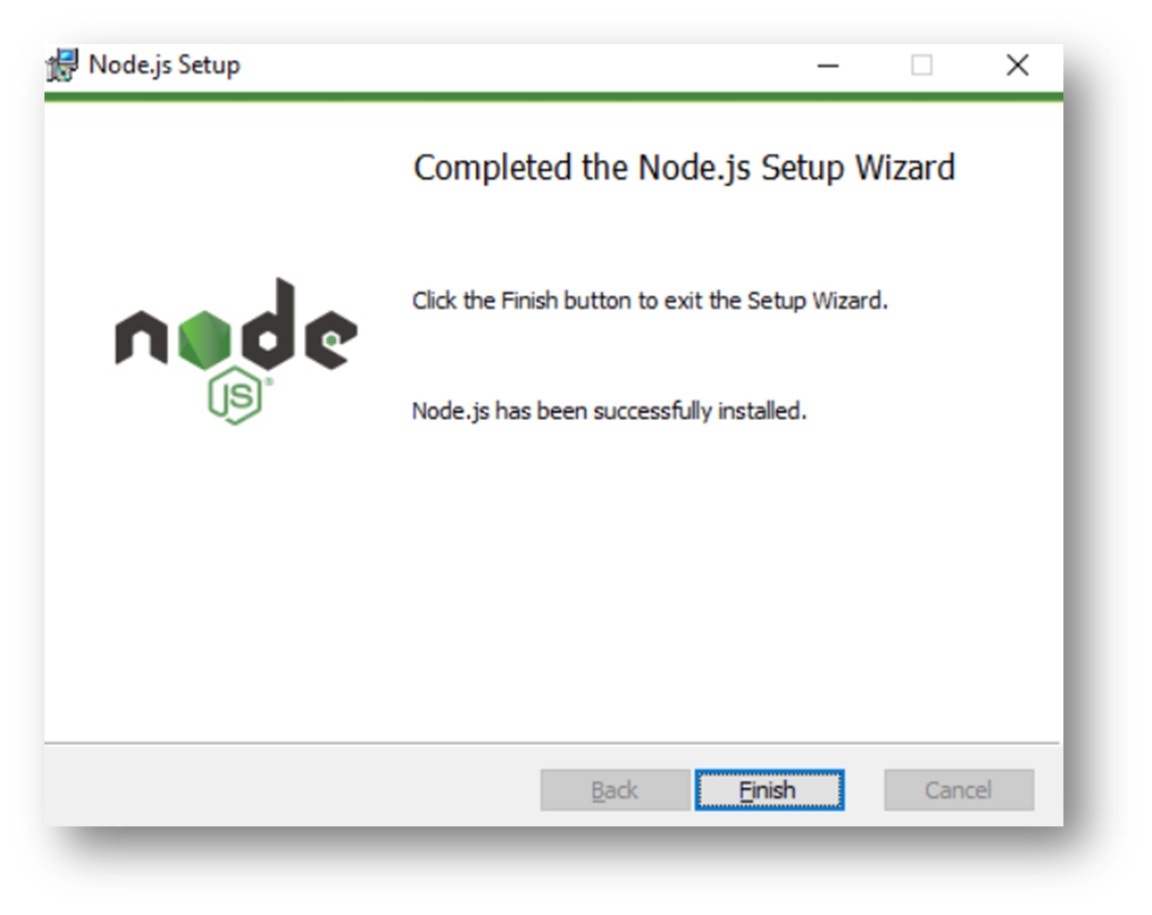


Now browse the location where you want to install the Node



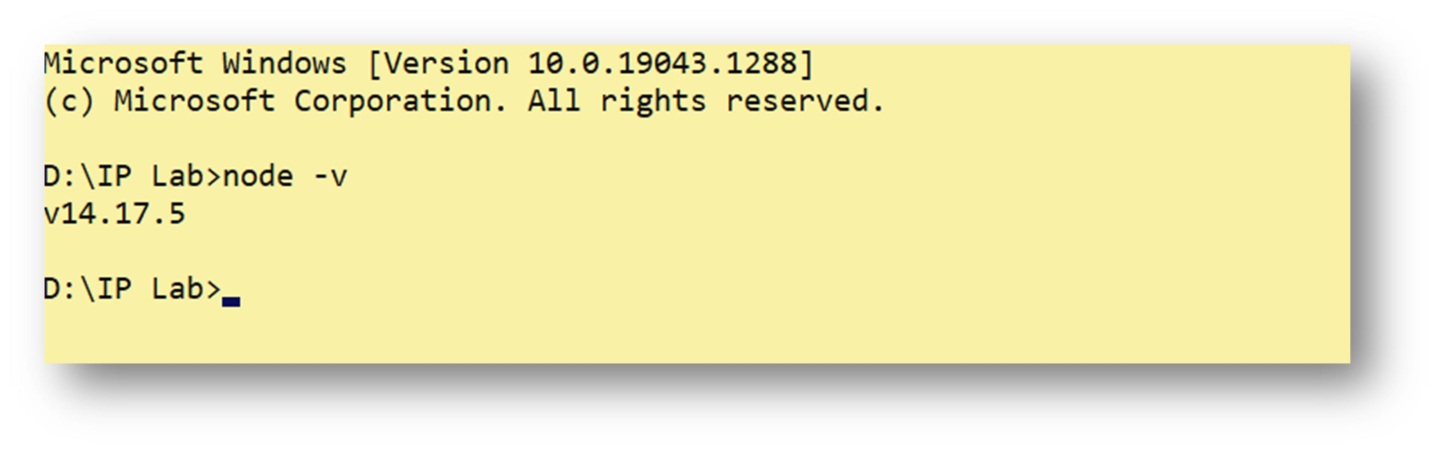
Now select Node.js runtime and then click Next.



Now click the final Install button to Install Node.js

Click on Finish Button

# Now that we have Node.js installed, we can move on to the next step.

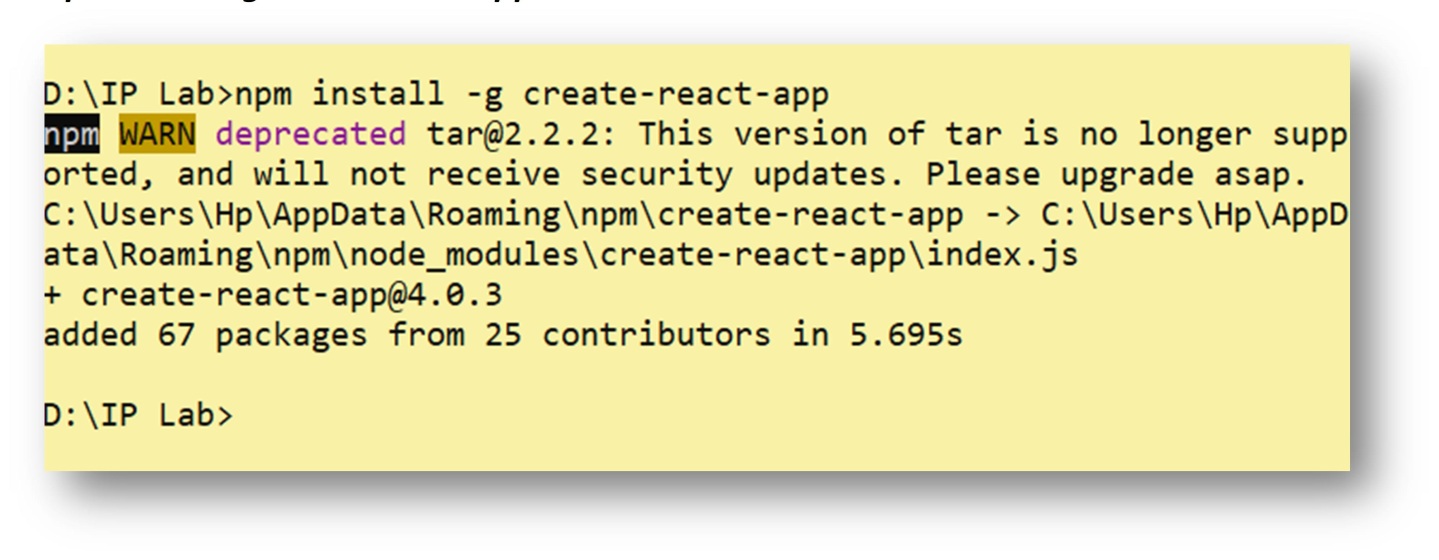
Open command prompt to check whether it is completely installed or not type the command.

If the installation went well it will give you the version you have installed.

# React installation and create project:

**Step 1**: Install Node.js installer for windows. Here install the LTS version which we have already done.

**Step 2**: Now in the terminal run the below command:

***npm install -g create-react-app***

It will globally install react app for you. To check everything went well run the command.

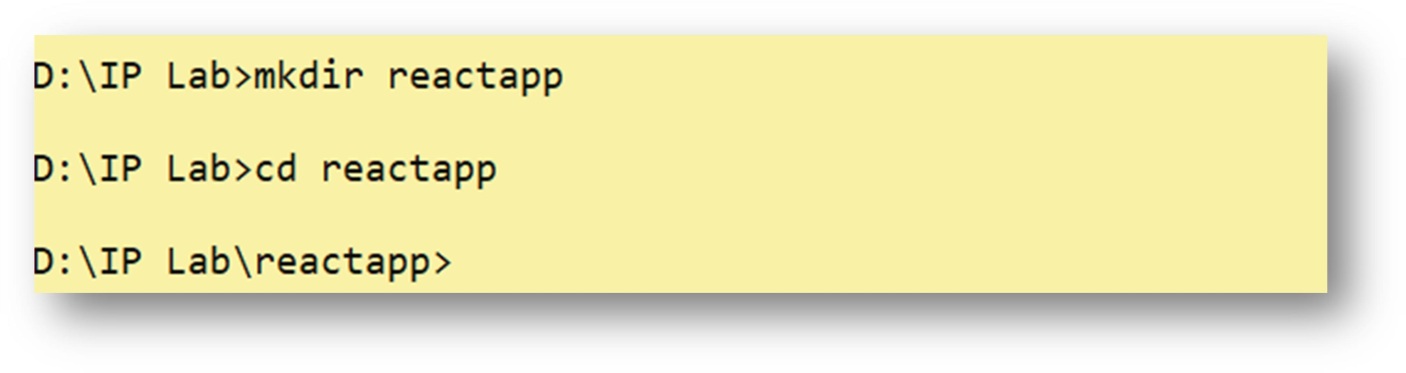
## npm view react version

**Step 3:** Now Create a new folder where you want to make your react app using the below command:

## mkdir reactapp

**Note:** The newfolder in the above command is the name of the folder and can be anyting.

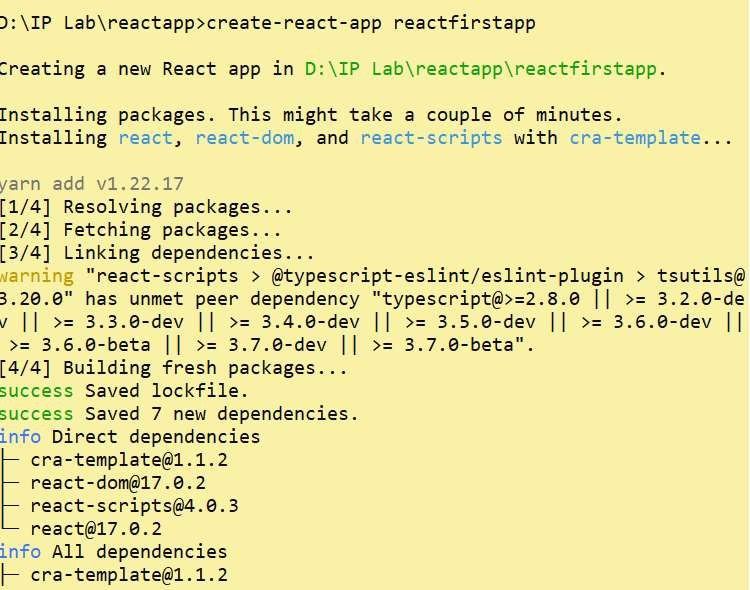
Move inside the same folder using the command: ***cd reactapp1***

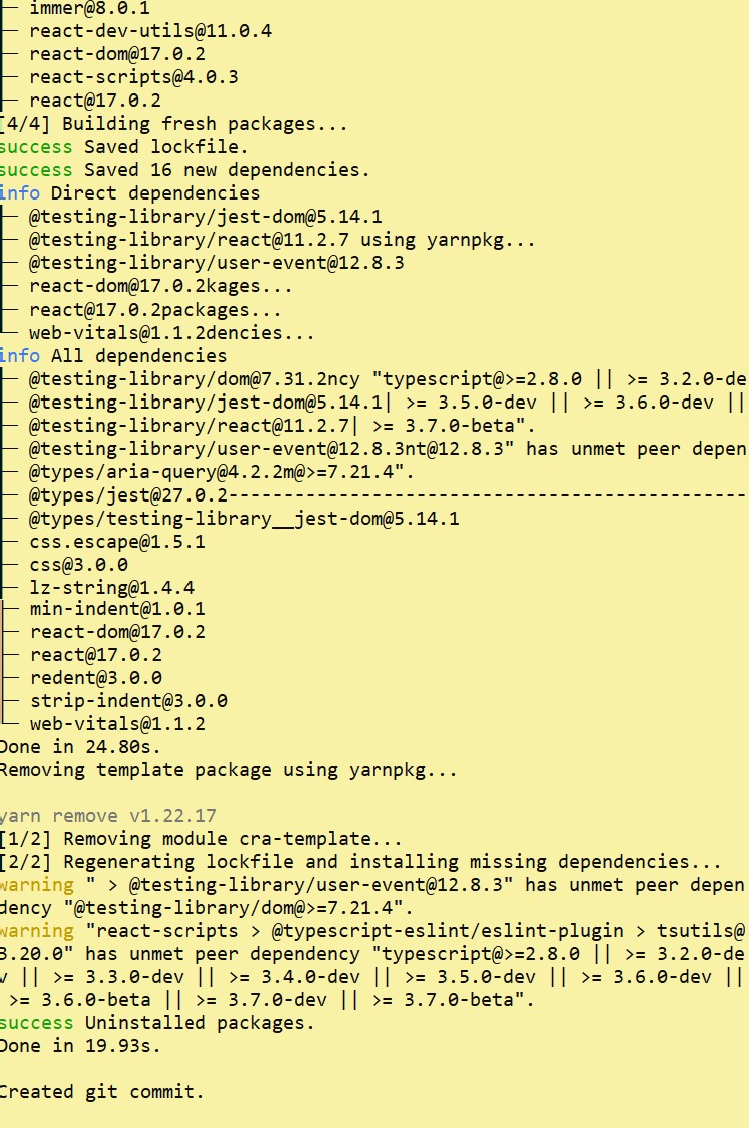


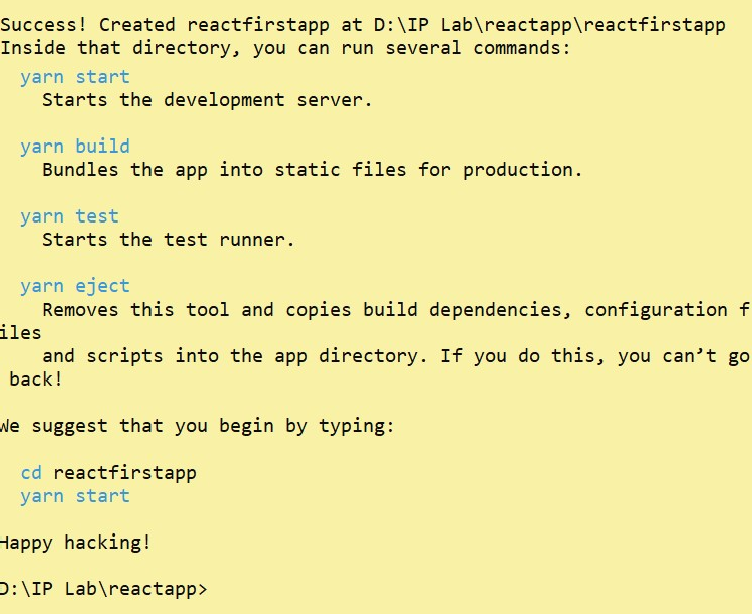
**Step 4:** Now inside this folder run the command

## create-react-app reactfirstapp (Your App Name)

It will take some time to install the required dependencies.

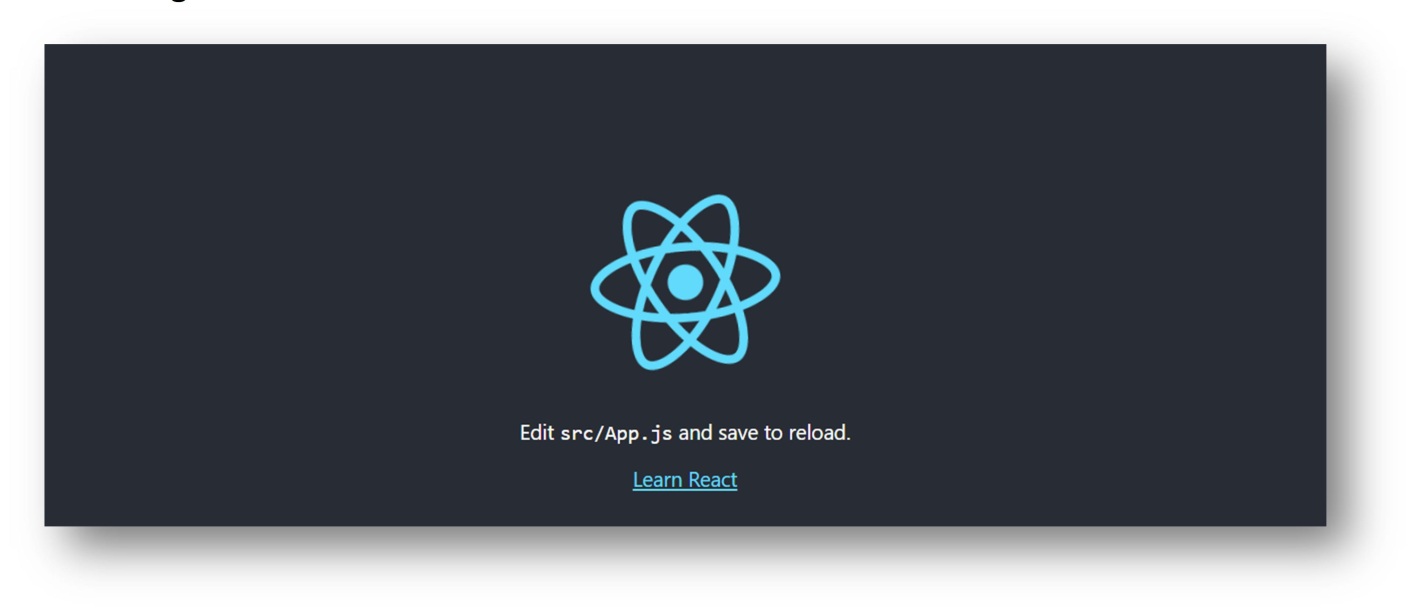






**Step 5**: Now open the IDE of your choice for e.g. Visual studio code, to start your app run the below command.

## npm start

Once you run the above command a new tab will open in your browser showing React logo.

Congratulation you have successfully installed the react-app and are ready to build awesome websites and app.

**CONCLUSION:** In the above experiment, we studied about node.js and react, we also installed it and learned about how to create an app i.e. project in react.