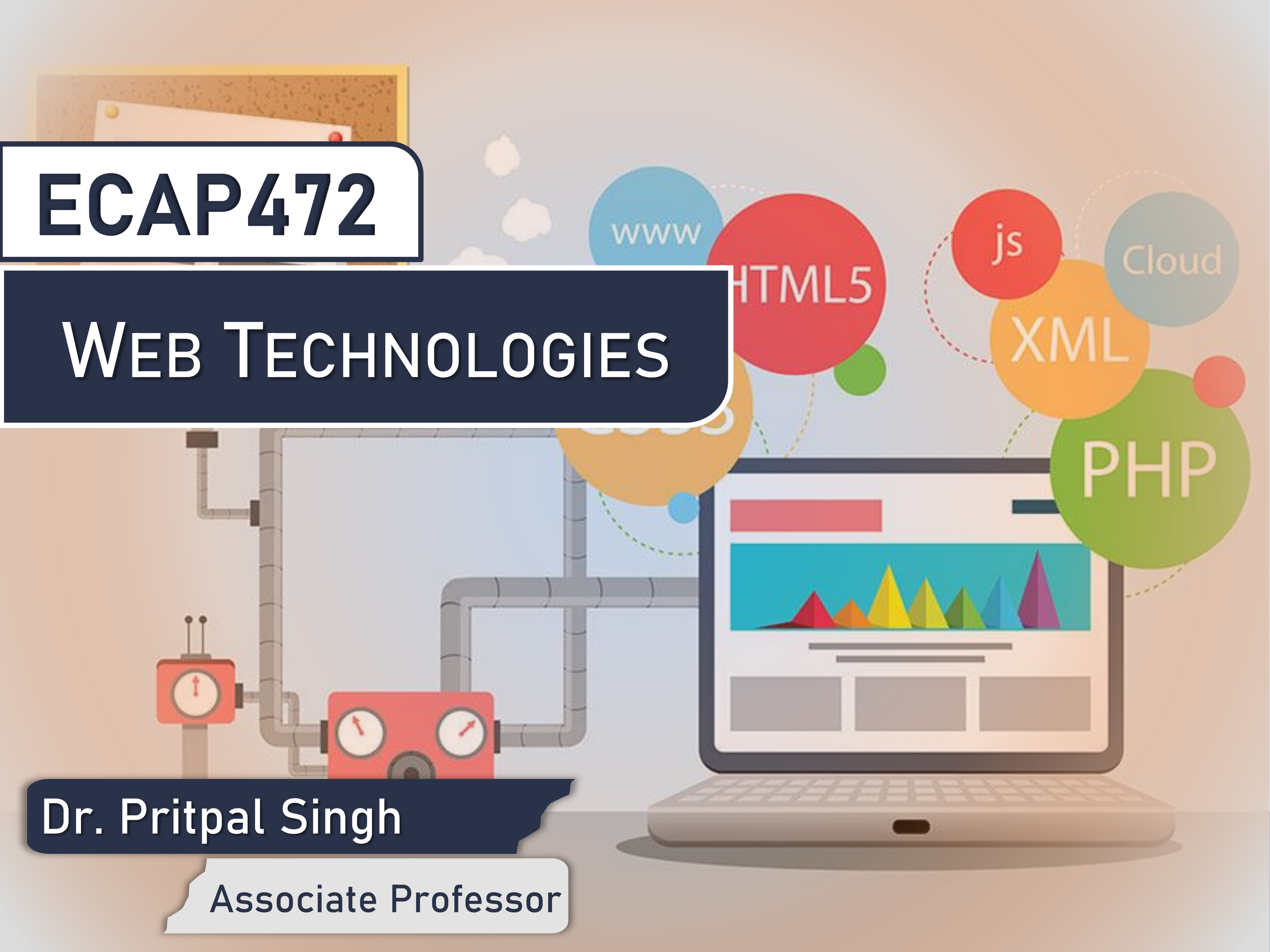
ECAP472

Web Technologies

Dr. Pritpal Singh

Associate Professor



Learning Outcomes

After this lecture, you will be able to

- go over and summarize various concepts of working with react.
- Understand basics of router in react .
- understand folder structure in react.

React

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.

React (JavaScript library)

- React can be used as a base in the development of single-page or mobile applications.
- However, React is only concerned with state management and rendering that state to the DOM.
- So creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Components

- React code is made of entities called components. These components are reusable and must be formed in the SRC folder following the Pascal Case as its naming conversion.
- Components can be rendered to a particular element in the DOM using the React DOM library

Virtual DOM

- Another notable feature is the use of a virtual Document Object Model, or virtual DOM.
- React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently. This process is called reconciliation.

React Installation

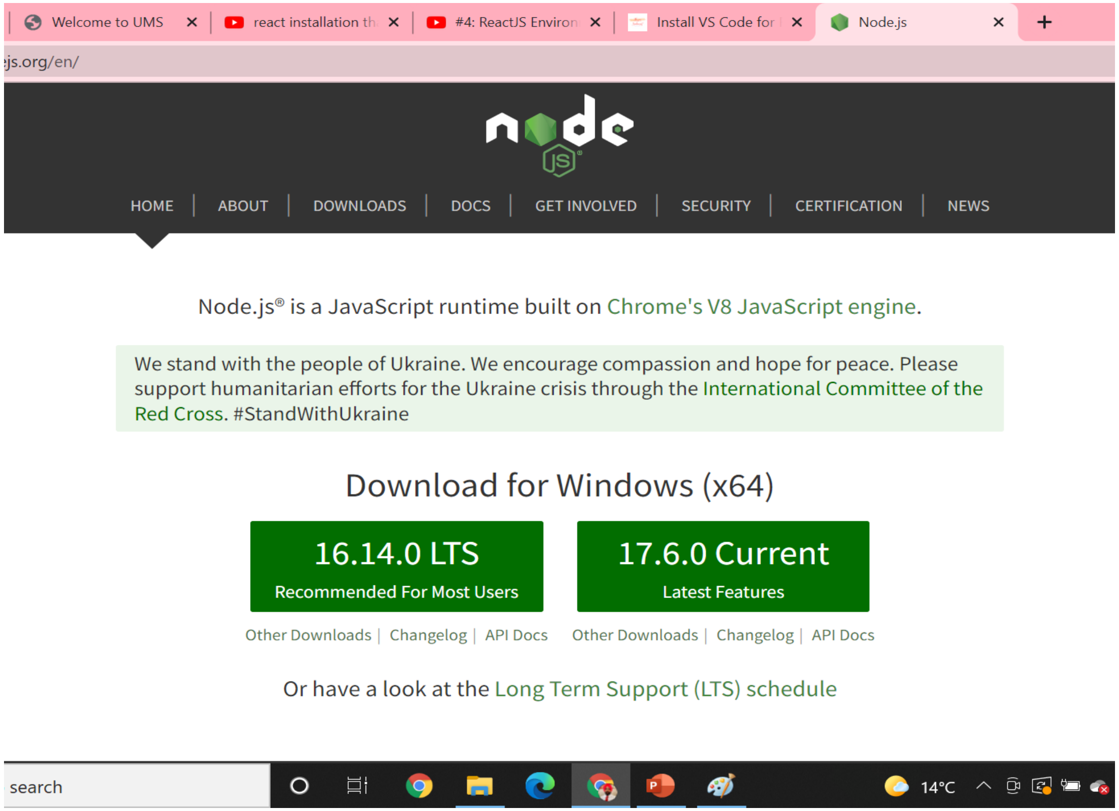
Install NodeJS and NPM.

Install Visual Studio Code.

Install React from terminal .

```
npm install -g create-react-app
```

Install NodeJS and NPM.



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

We stand with the people of Ukraine. We encourage compassion and hope for peace. Please support humanitarian efforts for the Ukraine crisis through the International Committee of the Red Cross. #StandWithUkraine

Download for Windows (x64)

16.14.0 LTS
Recommended For Most Users

17.6.0 Current
Latest Features

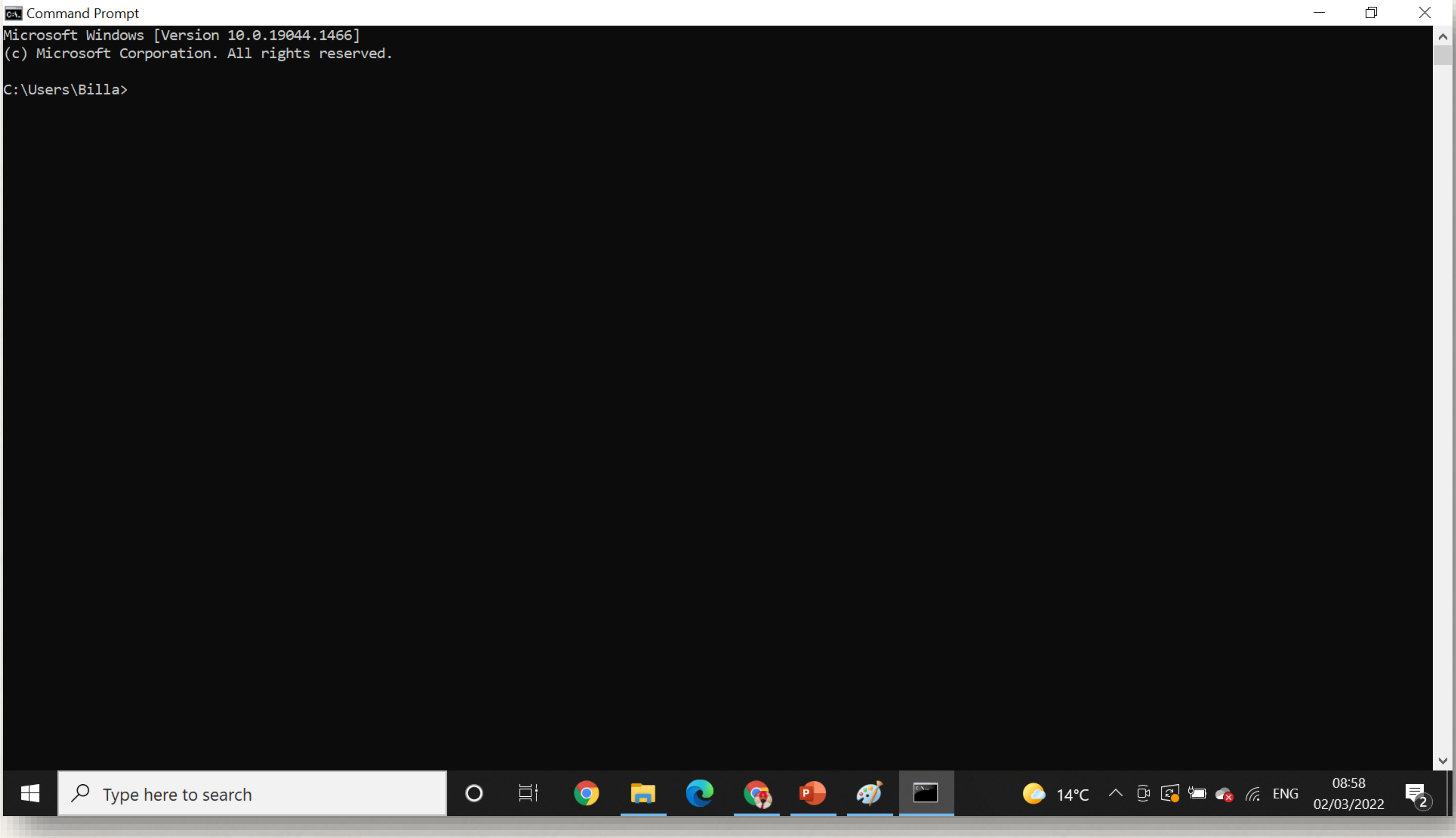
Other Downloads | Changelog | API Docs

Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule

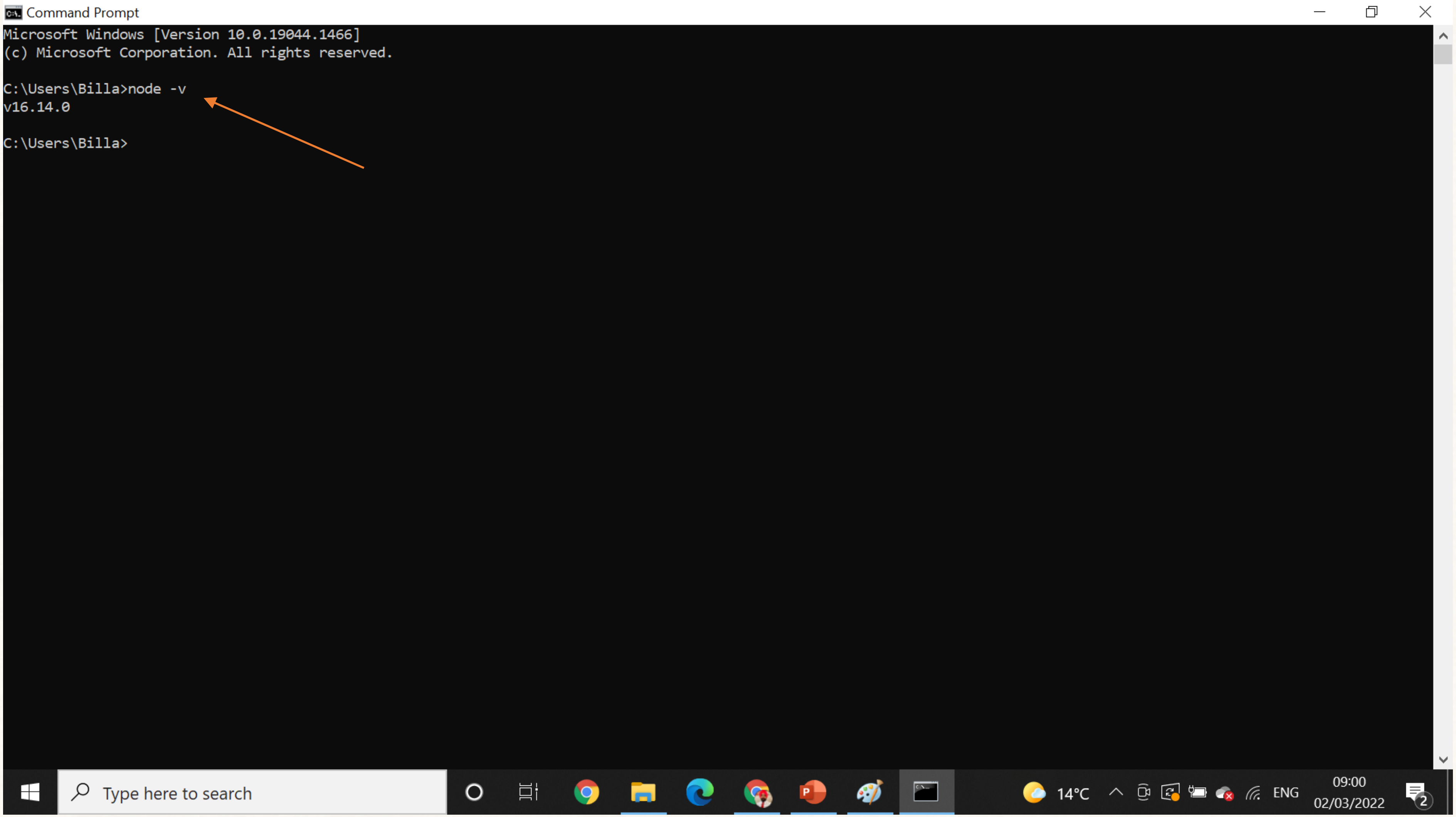
- Go to https://nodejs.org/en/
- Download the latest version of NodeJS LTS with Administrator privileges.

Open cmd *cmd



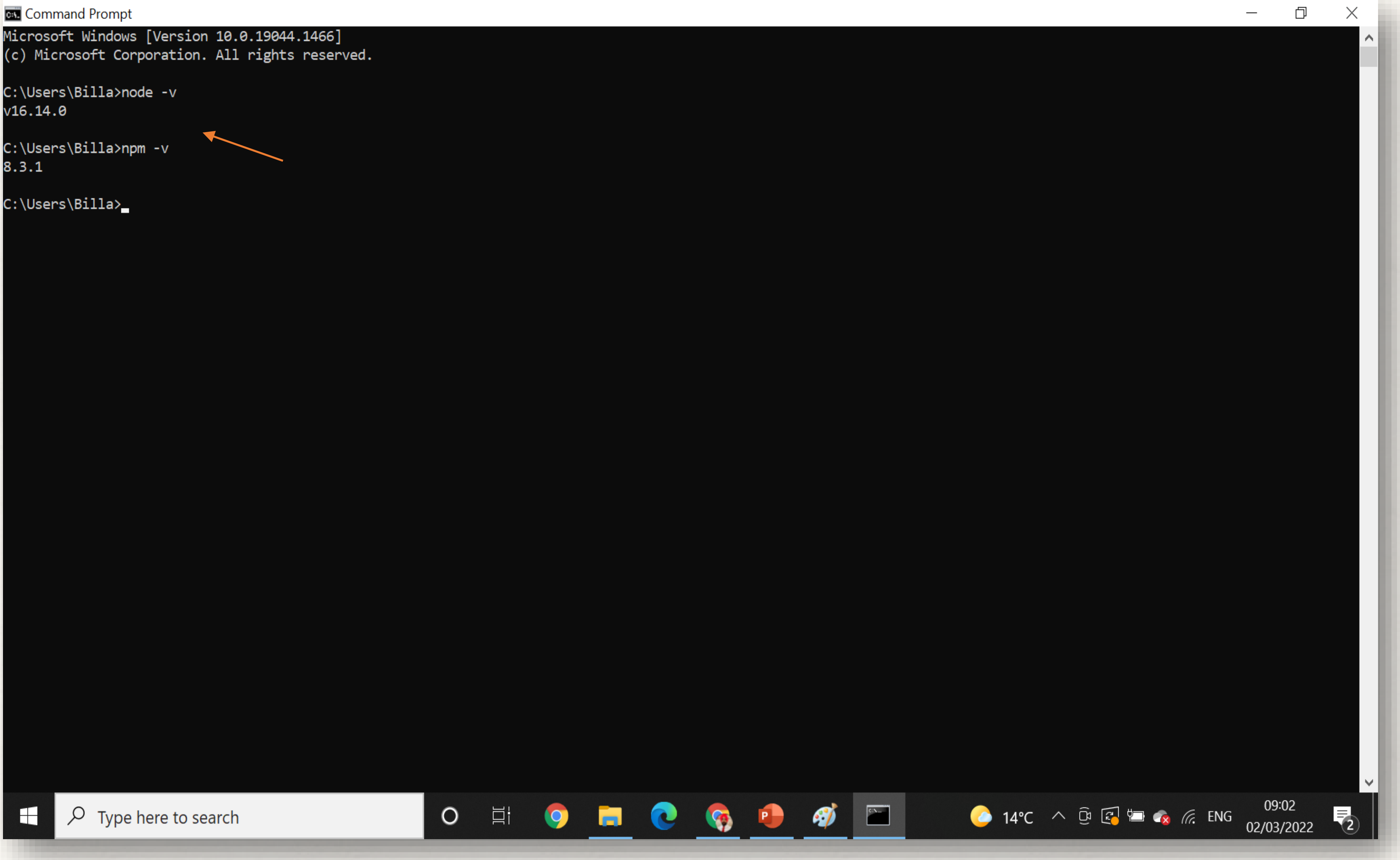
```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>
```

To check version* node -v



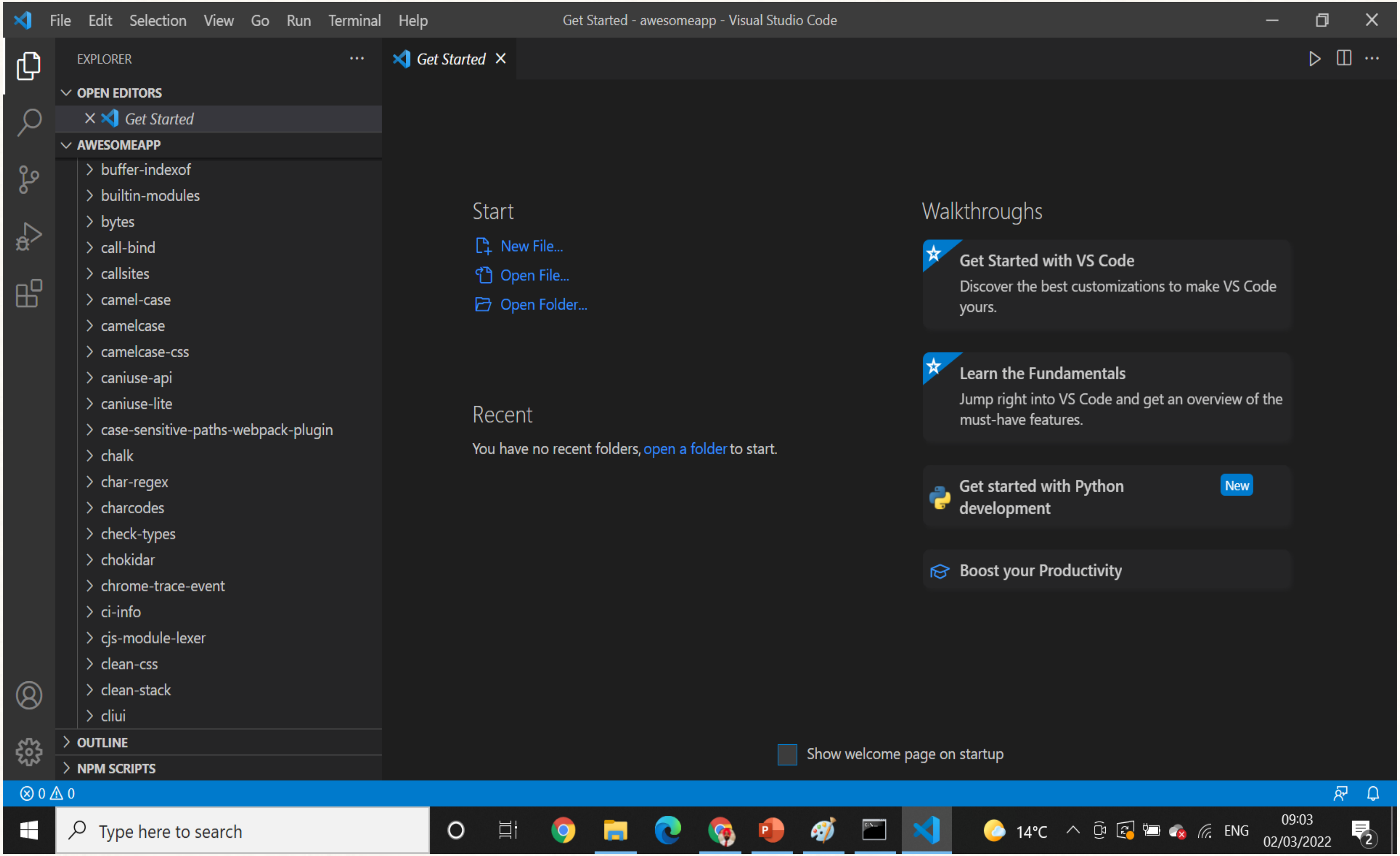
```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>
```

Npm installation* npm -v



```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>_
```

Install Visual Studio Code



Get Started - awesomeapp - Visual Studio Code

Start

- New File...
- Open File...
- Open Folder...

Recent

You have no recent folders, open a folder to start.

Walkthroughs

Get Started with VS Code
Discover the best customizations to make VS Code yours.

Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.

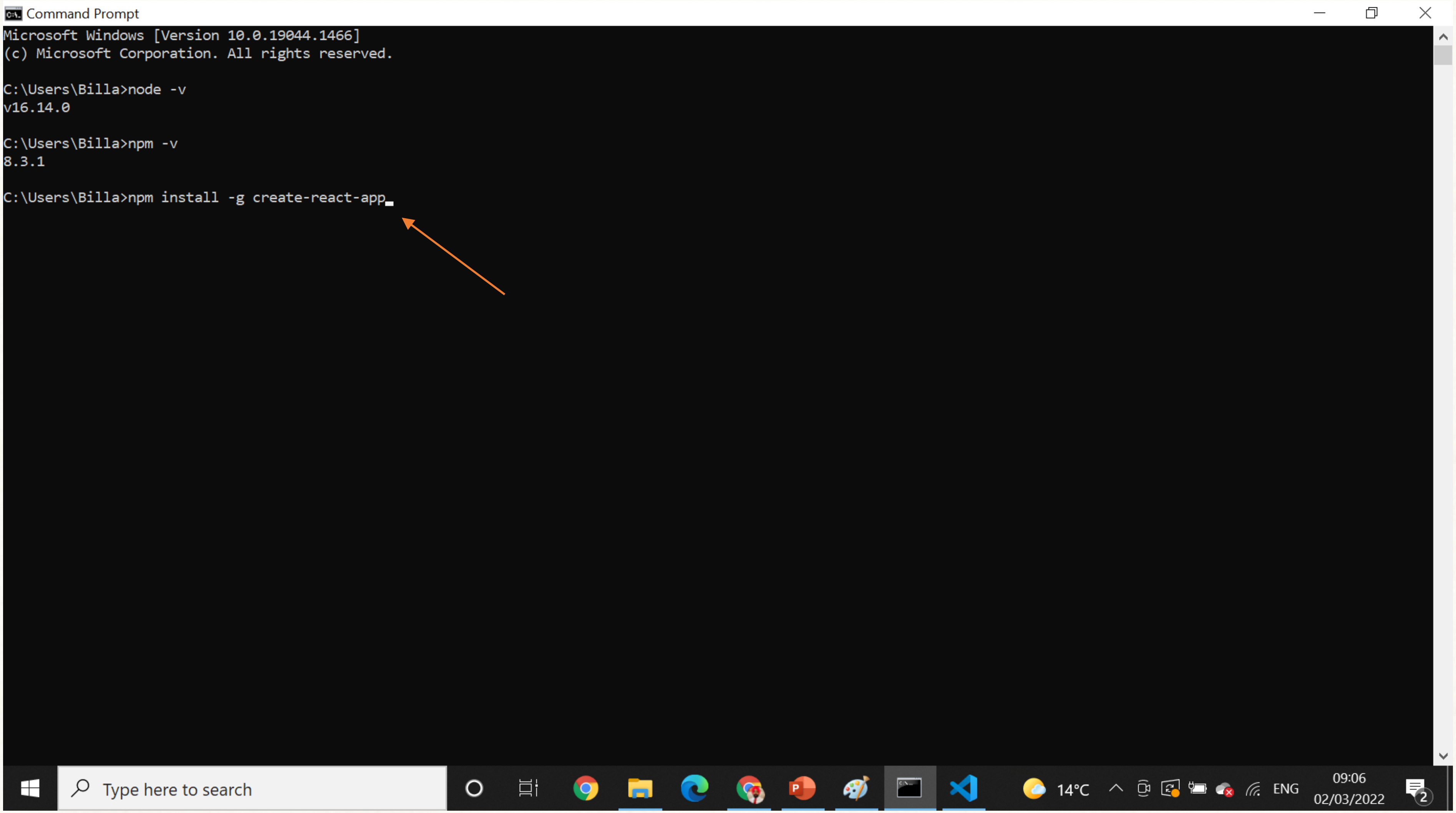
Get started with Python development New

Boost your Productivity

□ Show welcome page on startup

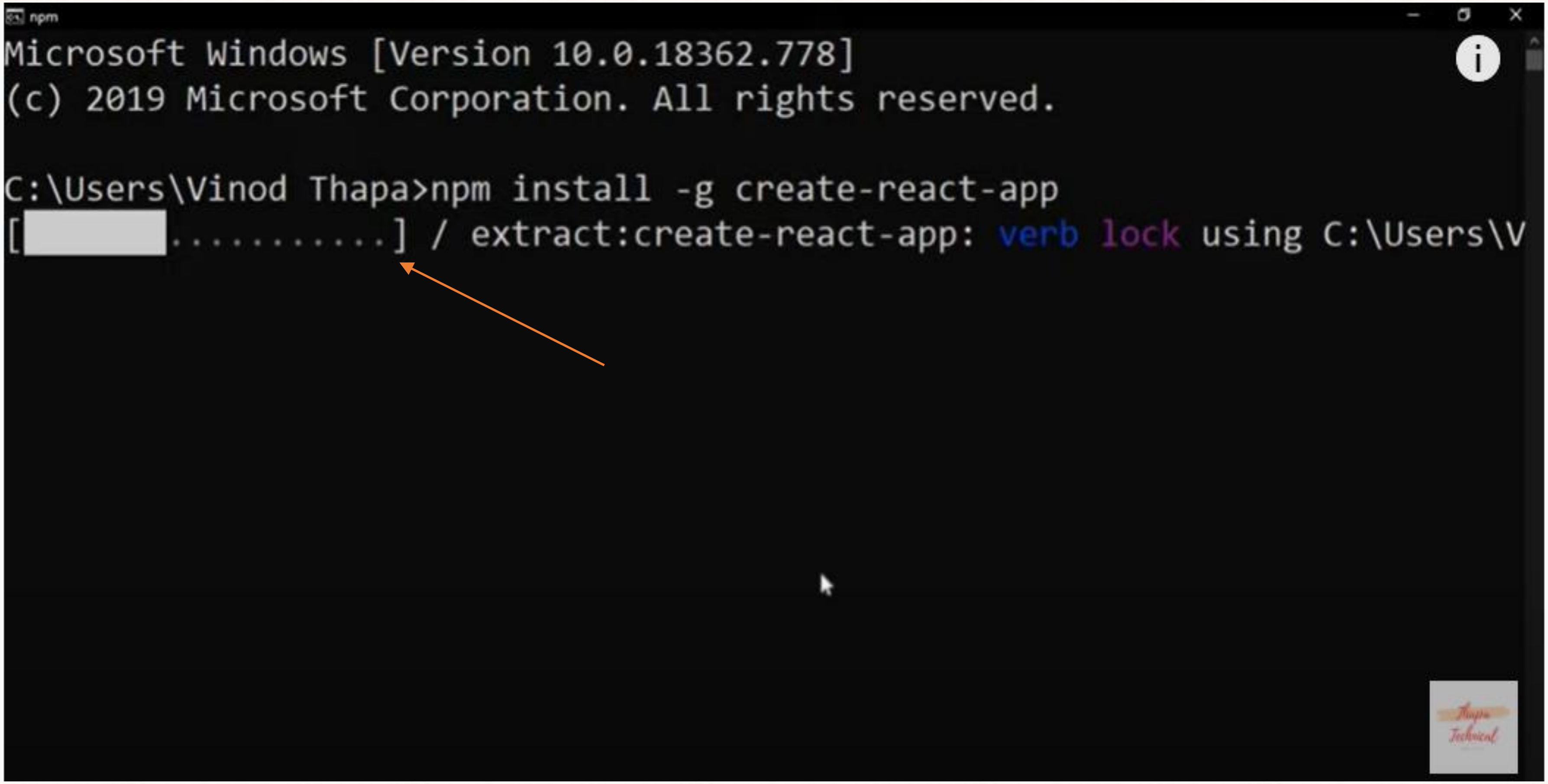
Install react from terminal

npm install -g create-react-app



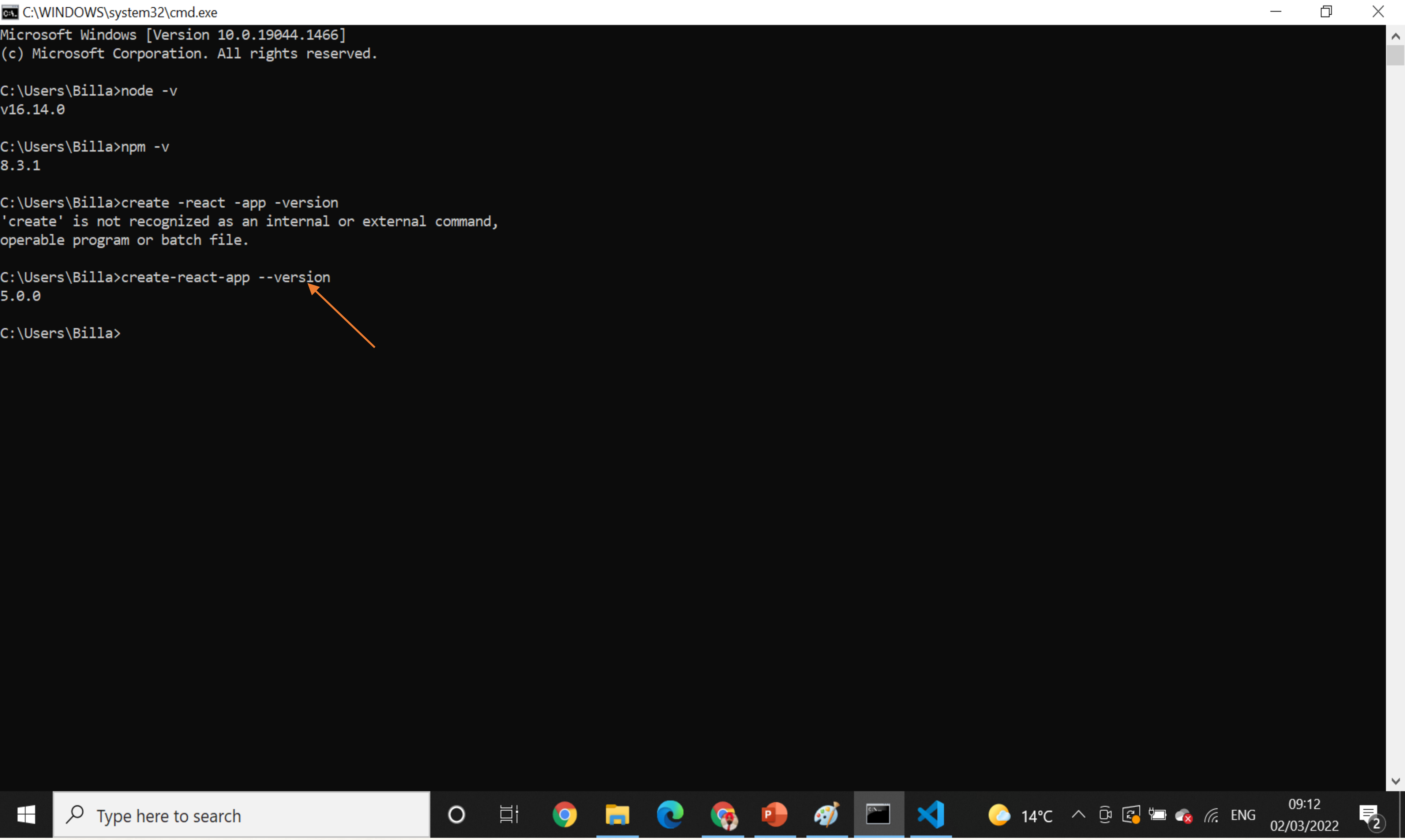
```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>npm install -g create-react-app
```

Installation ---



```
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Vinod Thapa>npm install -g create-react-app
[████████...........] / extract:create-react-app: verb lock using C:\Users\V
```

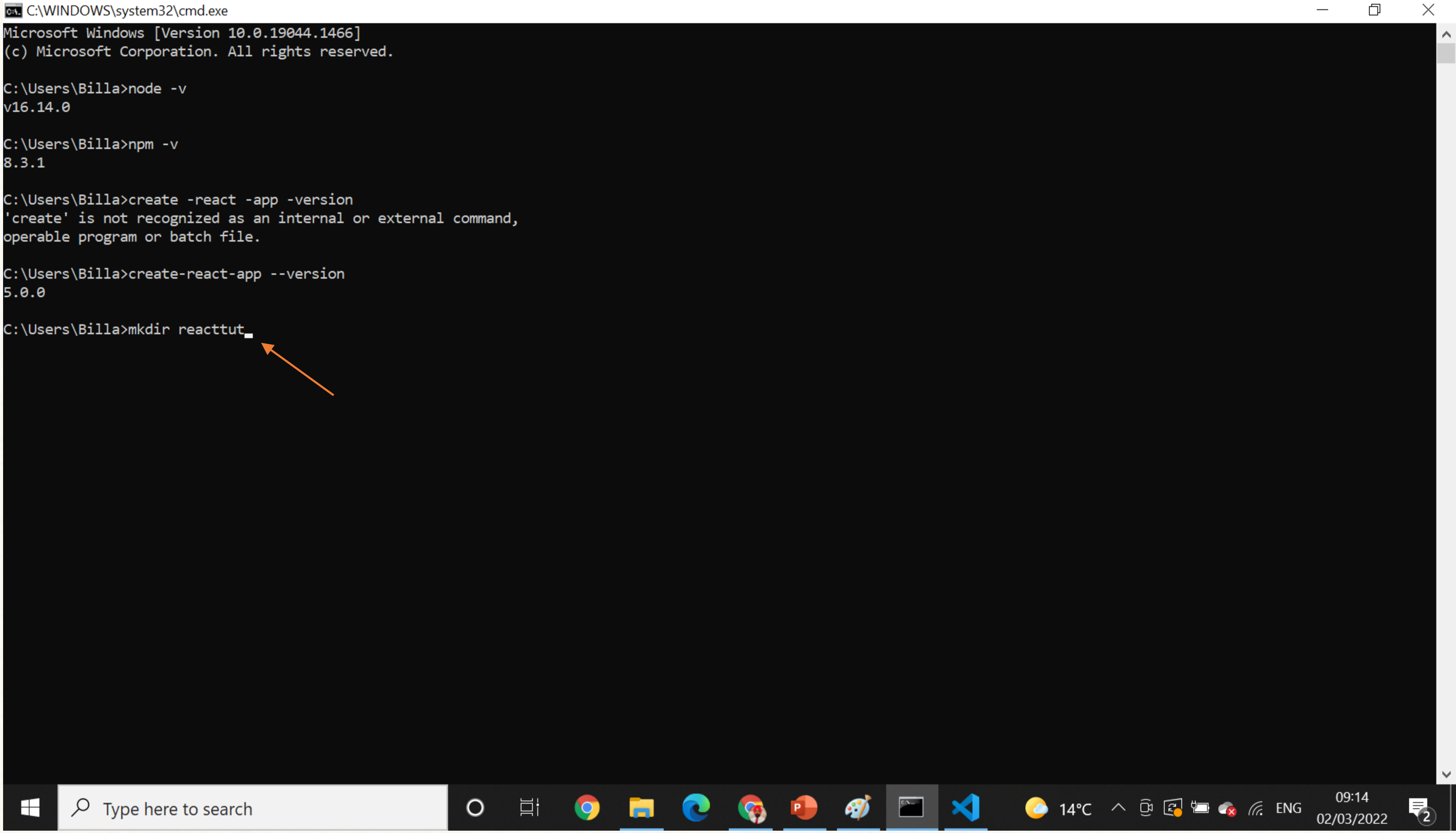
React Version* Create-react-app --Version



C:\WINDOWS\system32\cmd.exe

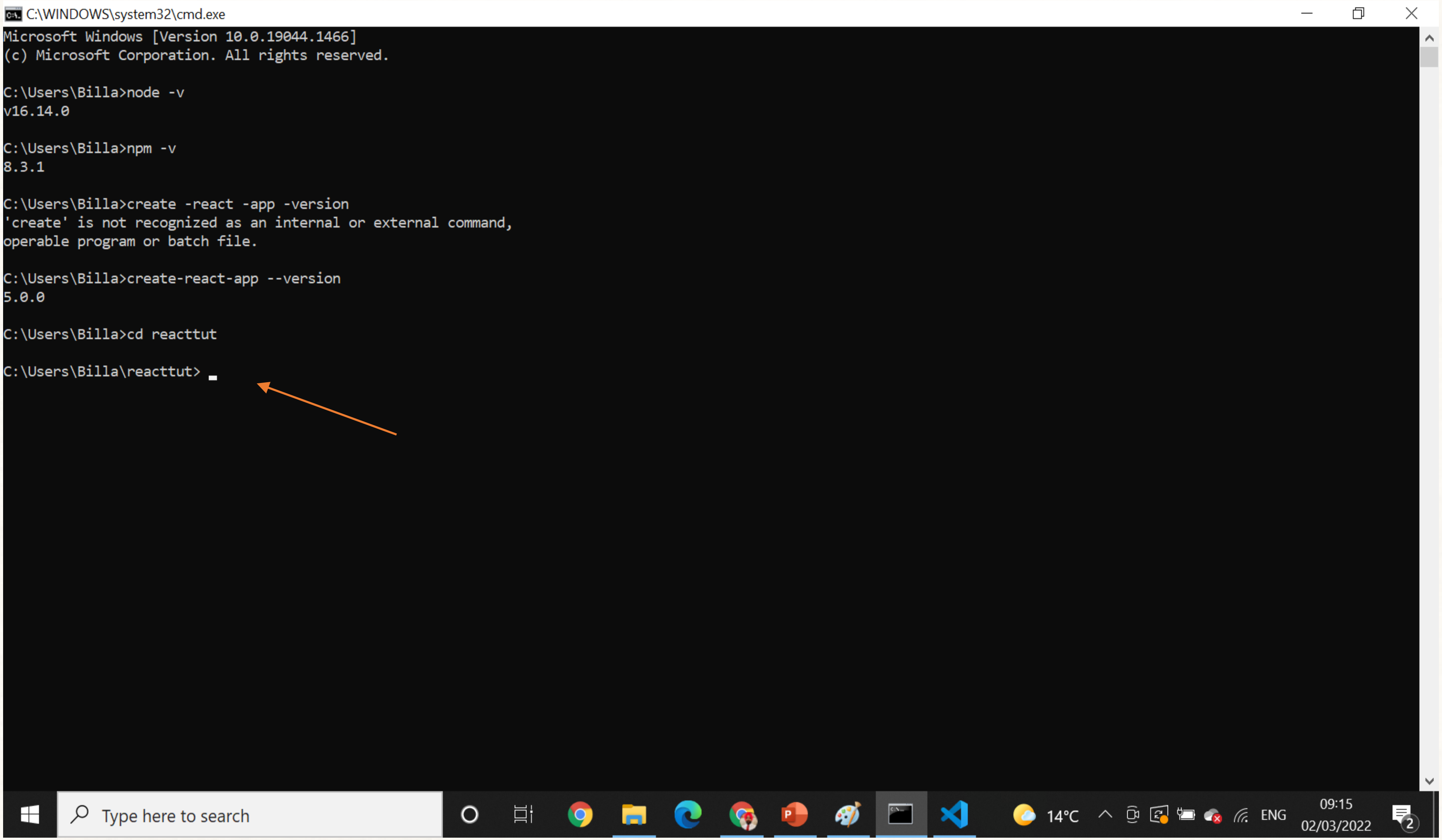
```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>create -react -app -version
'create' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Billa>create-react-app --version
5.0.0

C:\Users\Billa>
```

Creating directory* mkdir



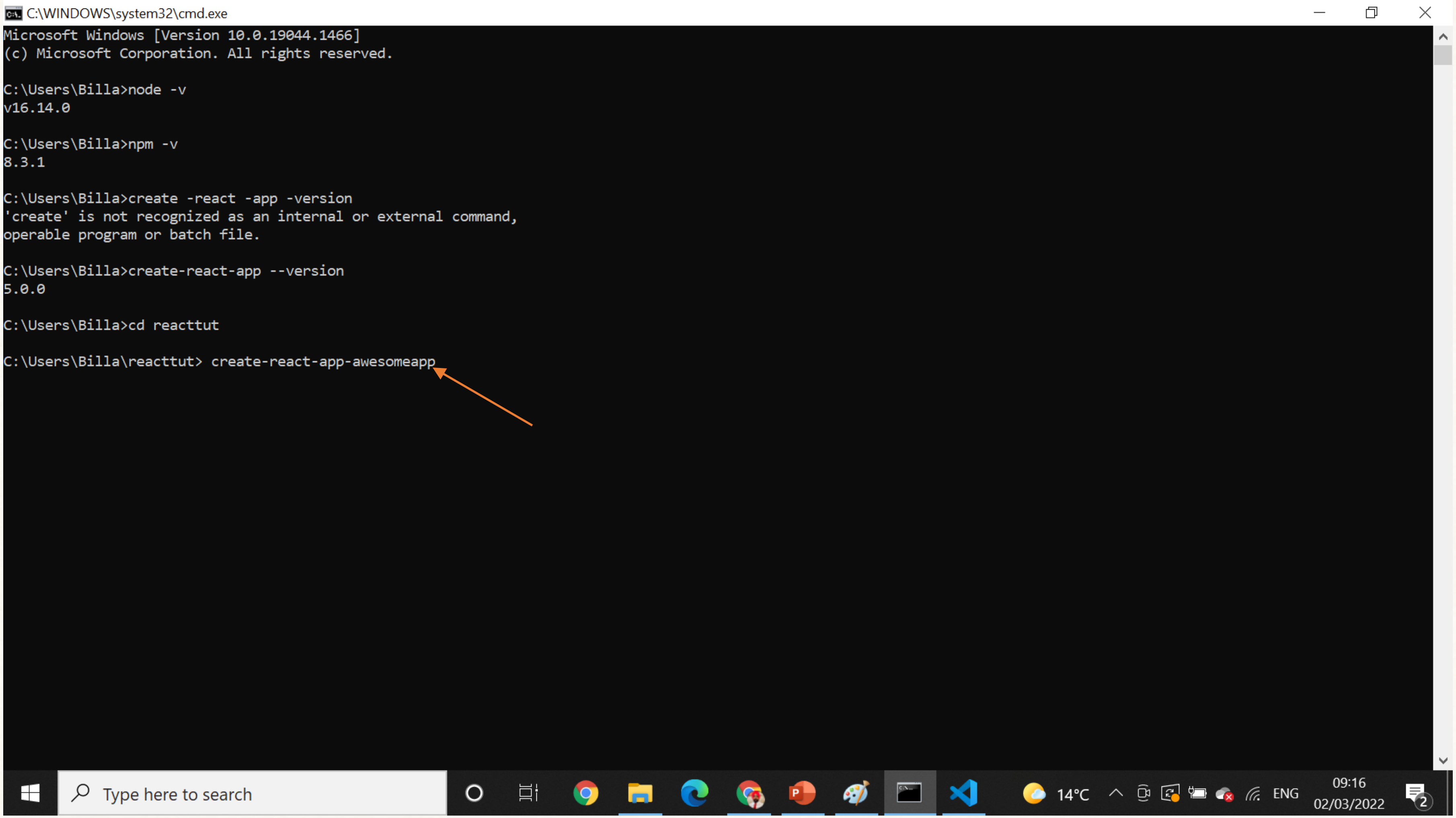
```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>create -react -app -version
'create' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Billa>create-react-app --version
5.0.0

C:\Users\Billa>mkdir reacttut
```

Change working directory*   CD



```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>create -react -app -version
'create' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Billa>create-react-app --version
5.0.0

C:\Users\Billa>cd reacttut

C:\Users\Billa\reacttut> _
```

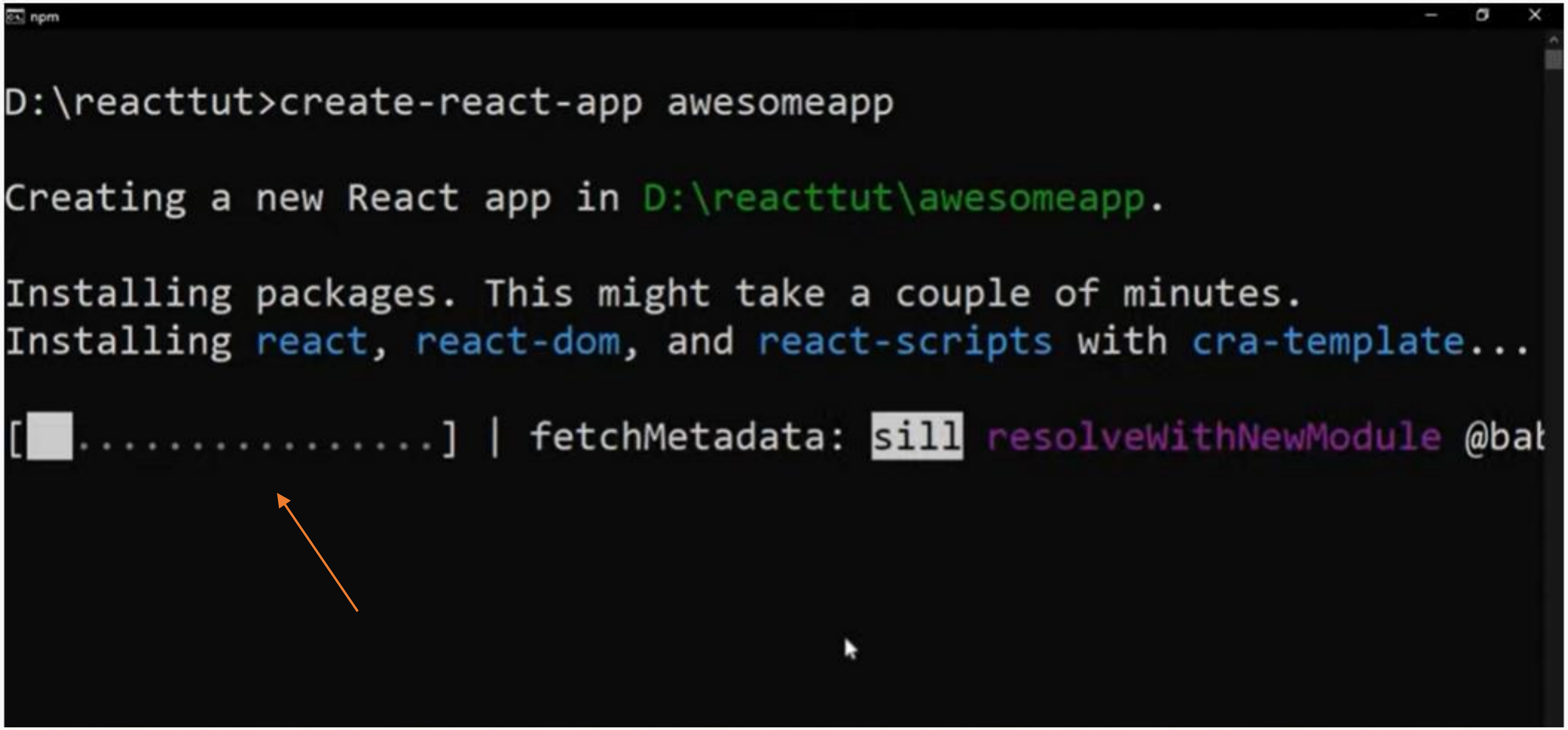
Create app inside directory * create-react-app-awesomeapp



C:\WINDOWS\system32\cmd.exe

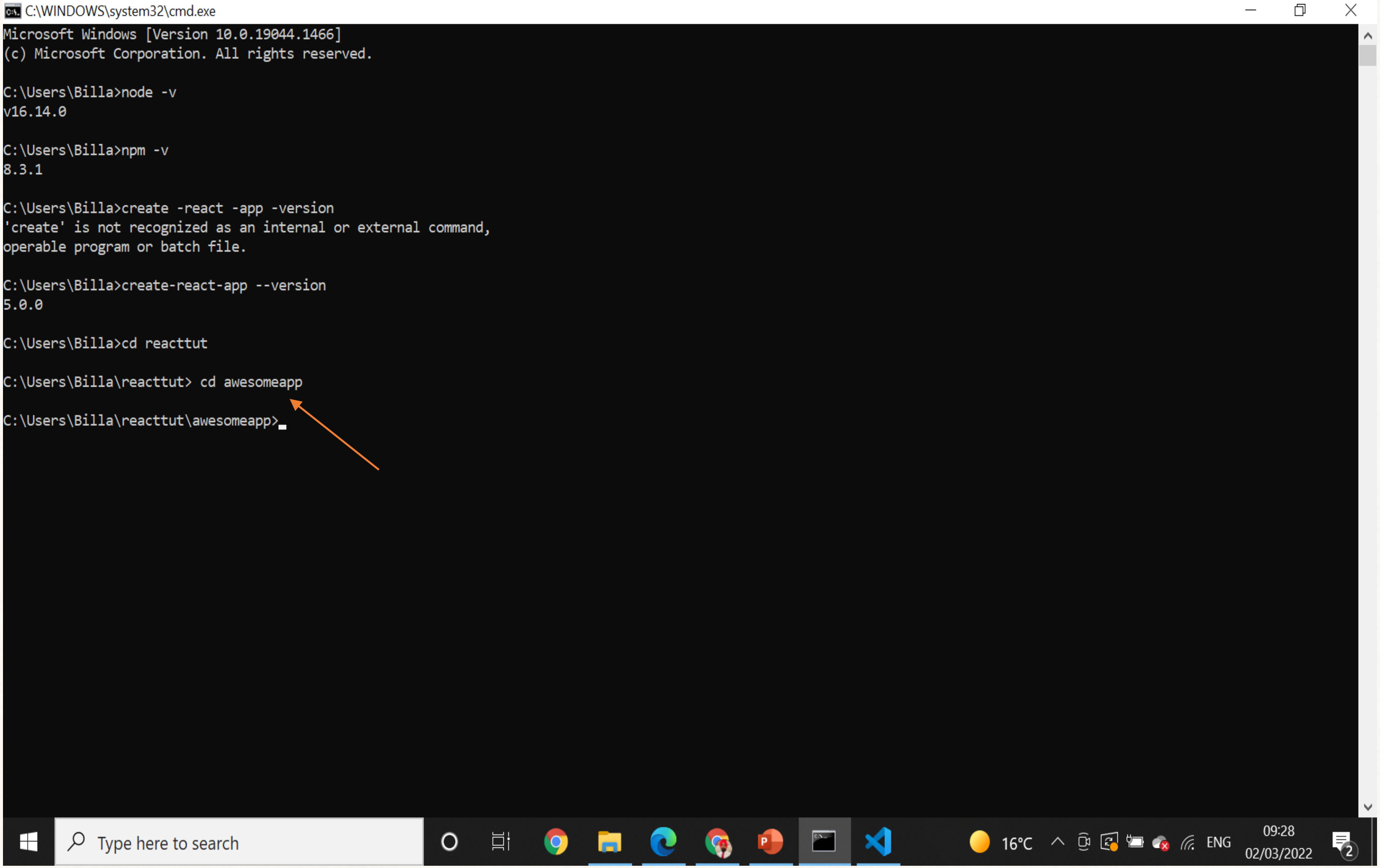
```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>create -react -app -version
'create' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Billa>create-react-app --version
5.0.0

C:\Users\Billa>cd reacttut

C:\Users\Billa\reacttut> create-react-app-awesomeapp
```

Installation going on



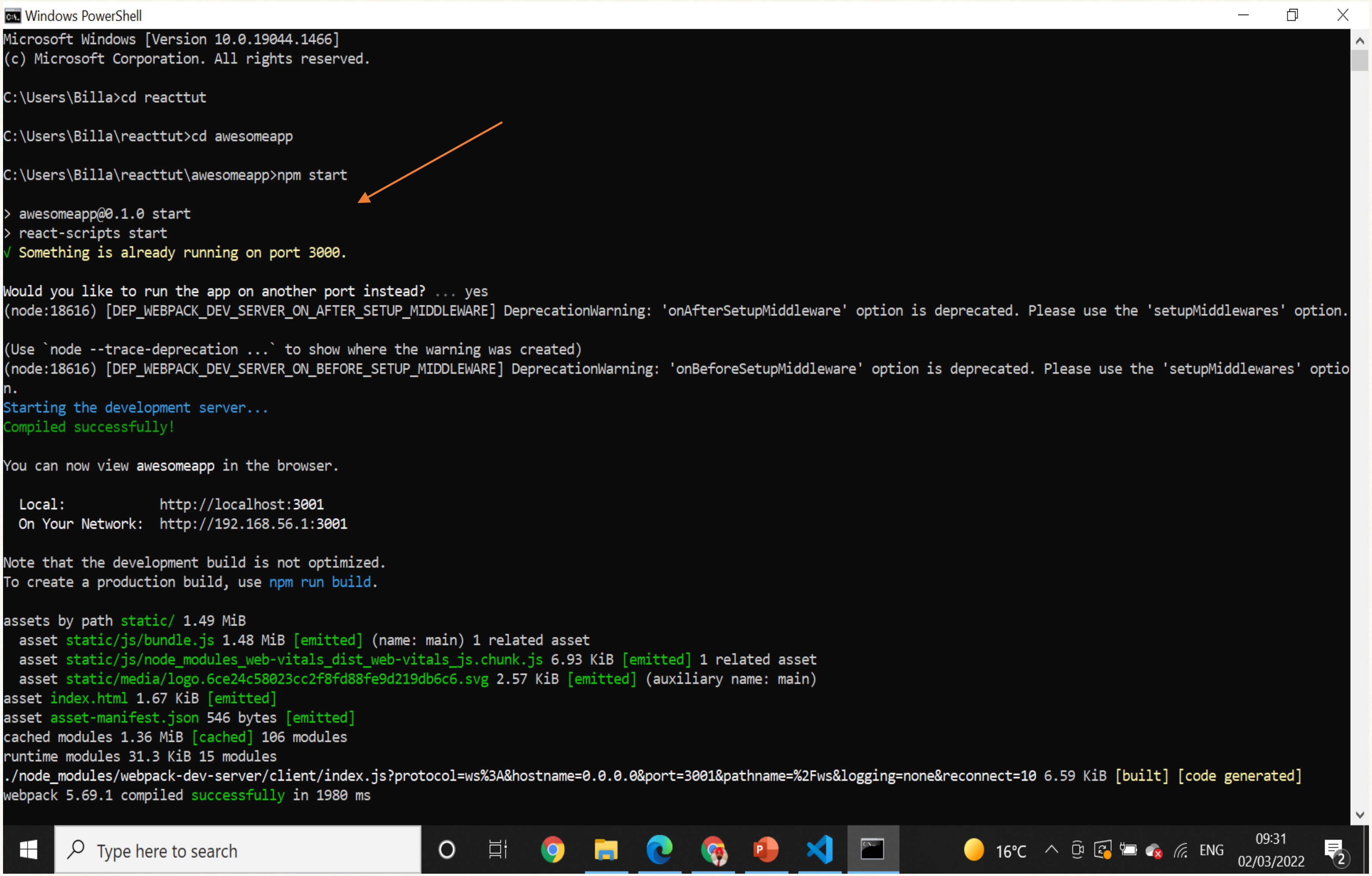
```
D:\reacttut>create-react-app awesomeapp

Creating a new React app in D:\reacttut\awesomeapp.

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

[█.................] | fetchMetadata: sill resolveWithNewModule @bat
```

Change directory *   CD



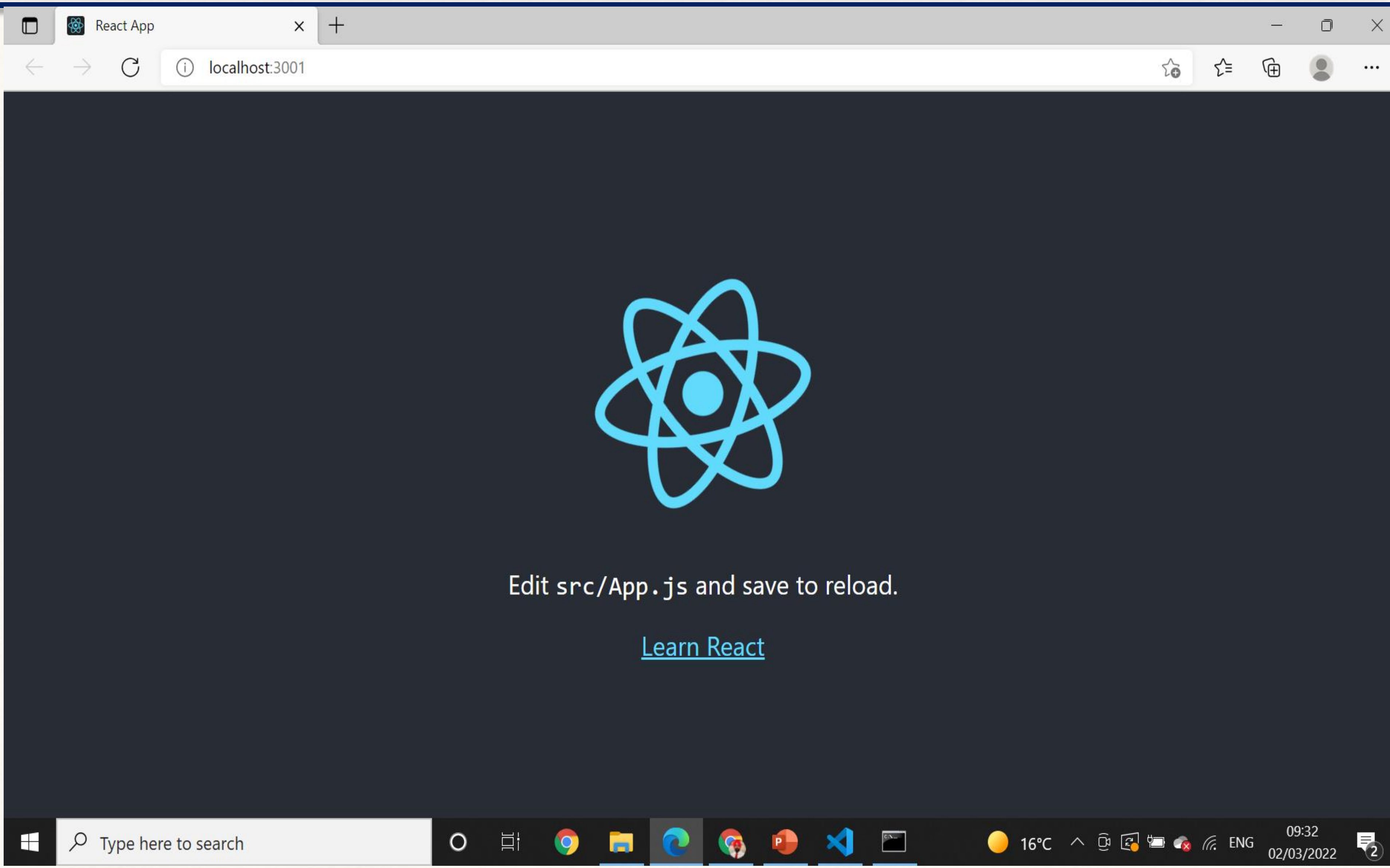
```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>node -v
v16.14.0

C:\Users\Billa>npm -v
8.3.1

C:\Users\Billa>create -react -app -version
'create' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Billa>create-react-app --version
5.0.0

C:\Users\Billa>cd reacttut

C:\Users\Billa\reacttut> cd awesomeapp

C:\Users\Billa\reacttut\awesomeapp>_
```

Npm start*   npm start



```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Billa>cd reacttut

C:\Users\Billa\reacttut>cd awesomeapp

C:\Users\Billa\reacttut\awesomeapp>npm start

> awesomeapp@0.1.0 start
> react-scripts start
√ Something is already running on port 3000.

Would you like to run the app on another port instead? ... yes
(node:18616) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.

(Use `node --trace-deprecation ...` to show where the warning was created)
(node:18616) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' optio
n.
Starting the development server...
Compiled successfully!

You can now view awesomeapp in the browser.

  Local:            http://localhost:3001
  On Your Network:  http://192.168.56.1:3001

Note that the development build is not optimized.
To create a production build, use npm run build.

assets by path static/ 1.49 MiB
  asset static/js/bundle.js 1.48 MiB [emitted] (name: main) 1 related asset
  asset static/js/node_modules_web-vitals_dist_web-vitals_js.chunk.js 6.93 KiB [emitted] 1 related asset
  asset static/media/logo.6ce24c58023cc2f8fd88fe9d219db6c6.svg 2.57 KiB [emitted] (auxiliary name: main)
asset index.html 1.67 KiB [emitted]
asset asset-manifest.json 546 bytes [emitted]
cached modules 1.36 MiB [cached] 106 modules
runtime modules 31.3 KiB 15 modules
./node_modules/webpack-dev-server/client/index.js?protocol=ws%3A&hostname=0.0.0.0&port=3001&pathname=%2Fws&logging=none&reconnect=10 6.59 KiB [built] [code generated]
webpack 5.69.1 compiled successfully in 1980 ms
```

Local Host.



Edit `src/App.js` and save to reload.

Learn React

What is npm for?

- npm is the package manager for the Node JavaScript platform.
- It puts modules in place so that node can find them, and manages dependency conflicts intelligently.
- It is extremely configurable to support a wide variety of use cases.
- Most commonly, it is used to publish, discover, install, and develop node programs.

What is Gitignore?

- The . gitignore file tells Git which files to ignore when committing your project to the GitHub repository. gitignore is located in the root directory of your repo. / will ignore directories with the name.
- Ignored files are usually build artifacts and machine generated files that can be derived from your repository source or should otherwise not be committed

package. json file

- The package. json file is the heart of any Node project.
- It records important metadata about a project which is required before publishing to NPM.
- Also defines functional attributes of a project that npm uses to install dependencies, run scripts, and identify the entry point to our package.

What is a package lock json file?

- In version 5, npm introduced the package-lock. … json file is to keep track of the exact version of every package that is installed so that a product is 100% reproducible in the same way even if packages are updated by their maintainers. This solves a very specific problem that package.

What is ES6?

- ES6 stands for ECMAScript 6.
- ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript, it was published in 2015, and is also known as ECMAScript 2015.
- ECMAScript (European Computer Manufacturers Association Script) is a scripting language based on JavaScript. Invented by Brendan Eich at Netscape, ECMAScript made its first appearance in the Navigator 2.0 browser

What is ES6?

- ES6 allows you to write the code in such a way that makes your code more modern and readable.
- By using ES6 features, we write less and do more, so the term 'Write less, do more' suits ES6.
- ES6 introduces you many great features such as scope variable, arrow functions, template strings, class destructions, modules, etc.

React ES6 Classes

- Classes are an essential part of object-oriented programming (OOP). Classes are used to define the blueprint for real-world object modeling and organize the code into reusable and logical parts.
- Before ES6, it was hard to create a class in JavaScript. But in ES6, we can create the class by using the class keyword. We can include classes in our code either by class expression or by using a class declaration.

React ES6 Classes

A class is a type of function, but instead of using the keyword function to initiate it, we use the keyword class, and the properties are assigned inside a constructor() method.

Arrow functions

- Arrow functions are introduced in ES6, which provides you a more accurate way to write the functions in JavaScript. They allow us to write smaller function syntax. Arrow functions make your code more readable and structured.
- Arrow functions are anonymous functions (the functions without a name and not bound with an identifier). They don't return any value and can declare without the function keyword. Arrow functions cannot be used as the constructors.

React ES6 Arrow Functions

Arrow functions allow us to write shorter function

syntax:

Before:

```
hello = function()

{

  return "Hello World!";

}
```

With Arrow Function:

```
hello = () =>
 {
  return "Hello World!";
}
```

It gets shorter! If the function has only one statement, and the statement returns a value, you can remove the brackets and the return keyword.

Arrow Functions Return Value by Default:

```
hello = () => "Hello World!";
```

Advantages of Arrow Function

The advantages of the arrow function are listed below:

- It reduces code size.
- The return statement is optional for a single line function.
- Lexically bind the context.
- Functional braces are optional for a single-line statement.

React ES6 Array Methods

- There are many JavaScript array methods.
- One of the most useful in React is the .map() array method.
- The .map() method allows you to run a function on each item in the array, returning a new array as the result.
- In React, map() can be used to generate lists.

Example

- Generate a list of items from an array:
- const myArray = ['apple', 'banana', 'orange'];

```
const myList = myArray.map((item) =>
<p>{item}</p>)
```

Export

- You can export a function or variable from any file.
- Let us create a file named person.js, and fill it with the things we want to export.
- There are two types of exports:
- Named and
- Default.

Import

You can import modules into a file in two ways, based on if they are named exports or default exports.

Import From Named Exports

- Import named exports from the file person.js:
- import { name, age } from "./person.js";

React ES6 Ternary Operator

- The ternary operator is a simplified conditional operator like if / else.

Syntax: condition ? <expression if true> : <expression if false>

- Here is an example using if / else.

Before:

```
if (authenticated) {
  renderApp();
} else {
  renderLogin();
}
```

Here is the same example using a ternary operator:

```
authenticated ? renderApp() : renderLogin();
```

Advantages of REACTJS

Intuitive

ReactJS is extremely intuitive to work with and provides interactivity to the layout of any UI. Plus, it enables fast and quality assured application development that in turn saves tome for both - clients and developers.

Advantages of REACTJS

Declarative

ReactJS enables significant data changes that result in automatic alteration in the selected parts of user interfaces. Owing to this progressive functionality, there is no additional function that you need to perform to update your user interface.

Provides Reusable Components

- ReactJS provides reusable components that developers have the authority to reuse and create a new application .
- Reusability is exactly like a remedy for developers.
- This platform gives the developers the authority to reuse the components build for some other application having the same functionality.
- Thereby, reducing the development effort and ensuring a flawless performance.

Components Support

- ReactJS is a perfect combination of JavaScript and HTML tags.
- The usage of the HTML tags and JS codes, make it easy to deal with a vast set of data containing the document object model.
- During this time, ReactJS works as a mediator which represents the DOM and assists to decide which component needs changes to get the exact results

SEO-friendly

- React JS was introduced after immense research and improvements by Facebook.
- Naturally, it stands out from the crowd and allows developers to build amazing, SEO-friendly user interfaces across browsers and engines

Some Additional Advantages Of REACTJS:

- Makes JavaScript coding easier
- Extremely competent
- Excellent cross-platform support
- Handles dependencies
- Template designing made easy
- Provides amazing developer tools
- UI focused designs
- Easy to adopt

The Render Function

React's goal is in many ways to render HTML in a web page.

React renders HTML to the web page by using a function called ReactDOM.render().

The Render Function

The ReactDOM.render() function takes two arguments:

- HTML code and an
- HTML element.

The purpose of the function is to display the specified HTML code inside the specified HTML element.

But render where?

- There is another folder in the root directory of your React project, named "public". In this folder, there is an index.html file.
- You'll notice a single <div> in the body of this file. This is where our React application will be rendered.

Example

- Display a paragraph inside an element with the id of "root":
- ReactDOM.render(<p>Hello</p>, document.getElementById('root'));
- The result is displayed in the <div id="root"> element:
- <body>
-   <div id="root"></div>
- </body>

The Root Node

- The root node is the HTML element where you want to display the result.
- It is like a container for content managed by React.
- It does NOT have to be a <div> element and it does NOT have to have the id='root':

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 without any createElement() and/or appendChild() methods.
- JSX converts HTML tags into react elements.
- You are not required to use JSX, but JSX makes it easier to write React applications.

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

React Components

- Components are like functions that return HTML elements.
- Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.
- Components come in two types, Class components and Function components.

Create Your First Component

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

Example

Create a Class component called Car

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

Function Component

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.

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 />

Why Should You Use React?

- React is Flexible:
- React is remarkably flexible. Once you have learned it, you can use it on a vast variety of platforms to build quality user interfaces. React is a library, NOT a framework. Its library approach has allowed React to evolve into such a remarkable tool.
- React was created with a single focus: to create components for web applications. A React component can be anything in your web application like a Button, Text, Label, or Grid.

React Has a Great Developer Experience

- You will fall in love with React when they start coding in it. Rapid development and React's small API combined creates a fantastic developer experience.
- React's API is very simple to learn. It has very few concepts to learn.

React Has Facebook's Support/Resources

React is heavily used in the Facebook app, website, and in Instagram. That's why Facebook is deeply committed to it. They use over 50k React components in their production environment. The top four React contributors on GitHub are full-time Facebook employees.

React Has Facebook's Support/Resources

- Also, the React team maintains a blog that consistently gives you details for each release.
- Because of the deep commitment by Facebook to React in production, when breaking change occur in React, Facebook consistently provides Codemod that automates the change.

React Has Broader Community Support, Too

Since 2015, React's popularity has grown steadily. It has a massive active community and its GitHub Repository has over 164k Stars. It is one of the Top 5 Repositories on GitHub.

React Has Great Performance

- The React team realized that JavaScript is fast, but updating the DOM makes it slow. React minimizes DOM changes. And it has figured out the most efficient and intelligent way to update DOM.
- Before React, most frameworks and libraries would update the DOM unintelligently to reflect a new state. This resulted in changes to a significant portion of the page.

React is Easy to Test

- React's design is very user friendly for testing.
- Traditional UI browser testing is a hassle to setup. On the other hand, you require very little or no configuration for testing in React.
- Traditional UI browser requires browsers for testing, but you can test React components quickly and easily using the node command-line.

React is Easy to Test

- Traditional UI browser testing is slow. But command-line testing is fast, and you can run a considerable amount of test suites at a time.
- Traditional UI browser testing is often time consuming and challenging to maintain. React test can be written quickly using tools like Jest & Enzyme.

What is Routing?

- Routing is the capacity to show different pages to the user. That means the user can move between different parts of an application by entering a URL or clicking on an element.

- React comes without routing. And to enable it in our project, we need to add a library named react-router.

Setting up the router

- To enable routing in our React app, we first need to import BrowserRouter from react-router-dom.
- import { BrowserRouter as Router } from "react-router-dom";
- This should hold everything in our app where routing is needed. That means, if we need routing in our entire app, we must wrap our higher component with BrowserRouter.

Rendering routes

- To render routes, we have to import the Route component from the router package.
- import { BrowserRouter as Router, Route } from "react-router-dom";

Using links to switch pages

- To add links to our project, we will use the React Router again.
- import { BrowserRouter as Router, Route, Link } from "react-router-dom";
- After importing Link, we have to update our navigation bar a bit. Now, instead of using a tag and href, React Router uses Link and to to, well, be able to switch between pages without reloading it.

That's all for now…