Installing Node js and react js

Node JS

```
sudo tar -xvf node-v16.15.1-linux-x64.tar.xz
sudo cp -r node-v16.15.1-linux-x64/{bin,include,lib,share} /usr/


export PATH=/usr/node-v16.15.1-linux-x64/bin:$PATH


node --version
```

REACT JS

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

create-react-app --version


create-react-app "NAME"

cd  "APPNAME"


npm start
```

Exercise 1 :Create a simple react application (Hello world)

Aim: Create a simple react application (Hello world)

Procedure:

Step 1: Create a react application using the following command

```
create-react-app filename
Example: create-react-app project1
```

Step 2: Once it is done change your directory to the newly created application using the following command

```
cd filename
Example: cd project1
```

Step 3: Now inside **App.js** and write down the following code and save the file then open http:// localhost:8000 and check the output.

App.js

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

function App() {
  return (
    <div className="App">
     <h1> Hello World </h1>
    </div>
  );
}

export default App;
```

App.css

```
.App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

index.css

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);const http = require('http');

const hostname = 'localhost';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World!\n');
});
```

```
server.listen(port, hostname, () => {
  console.log(`Server running at httpconst http = require('http');

const hostname = 'localhost';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});://${hostname}:${port}/`);
});

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

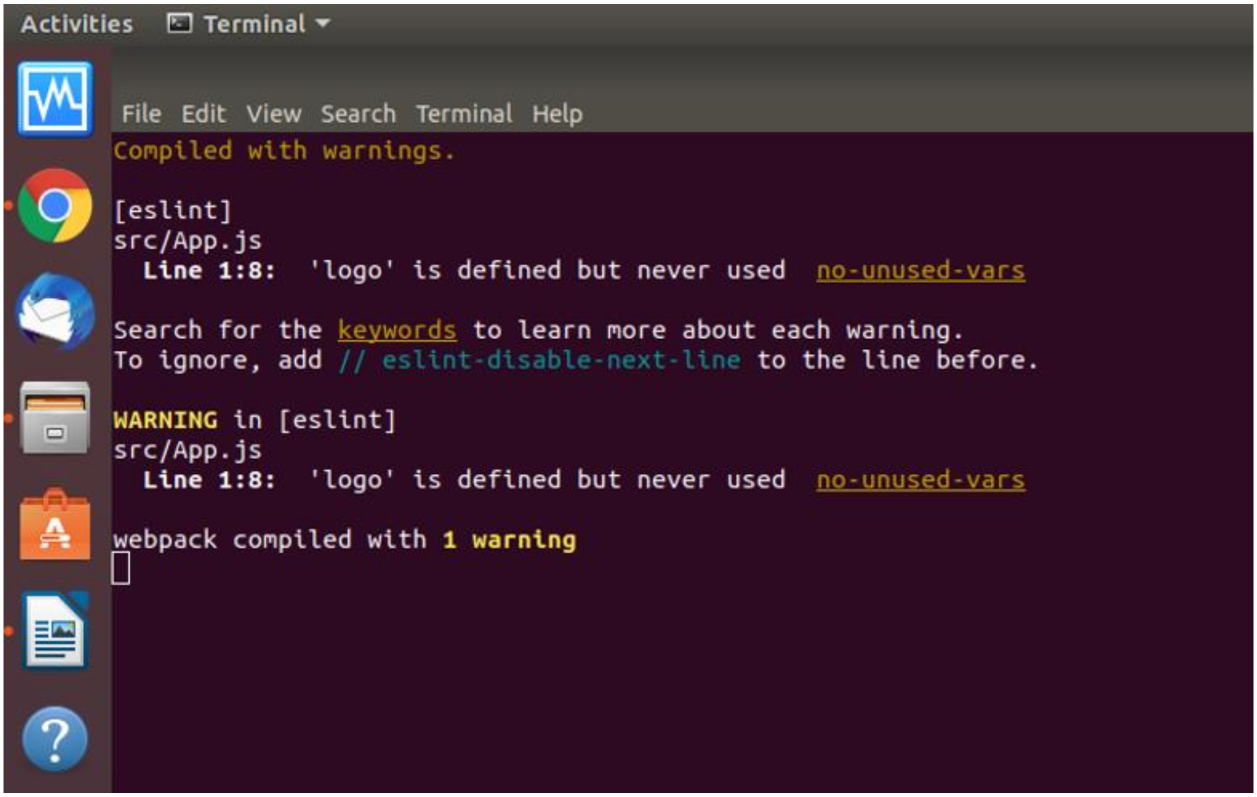
reportWebVitals.js

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;
```

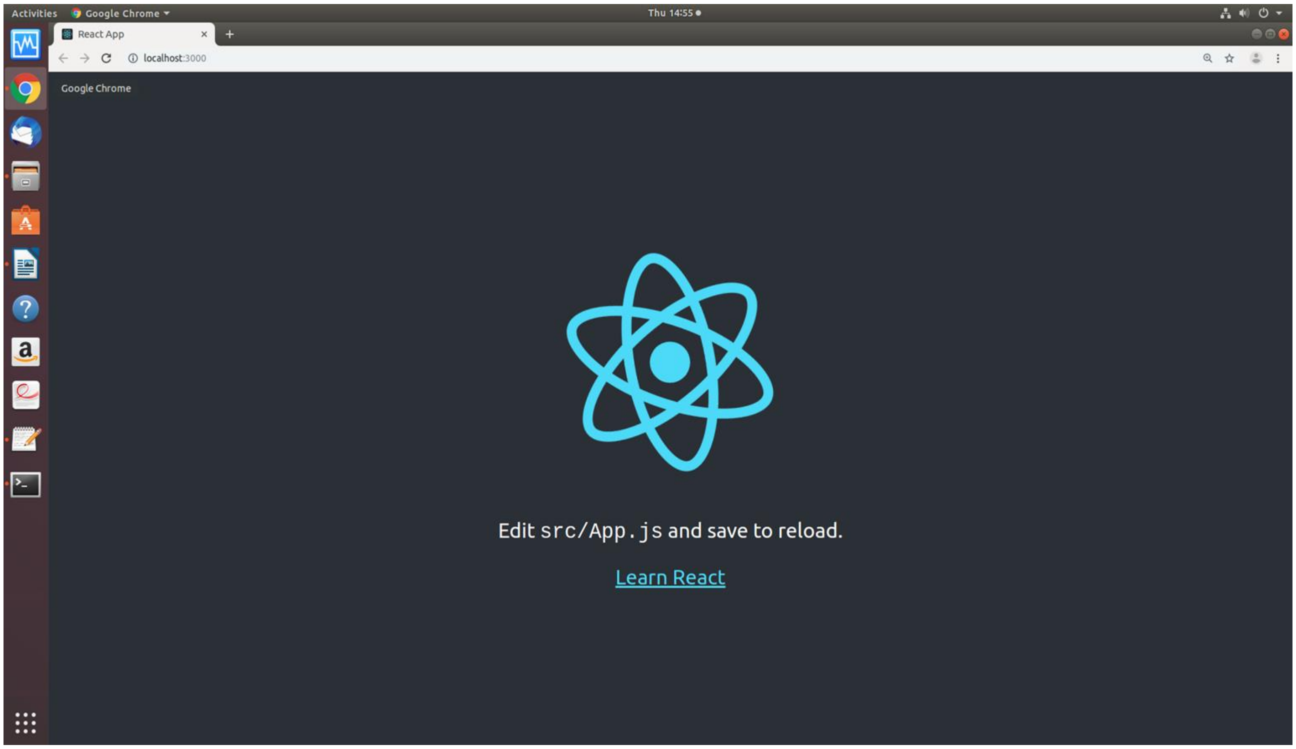
setupTests.js

```
// jest-dom adds custom jest matchers for asserting on DOM nodes.
// allows you to do things like:
// expect(element).toHaveTextContent(/react/i)
// learn more: https://github.com/testing-library/jest-dom
import '@testing-library/jest-dom';
```

Output

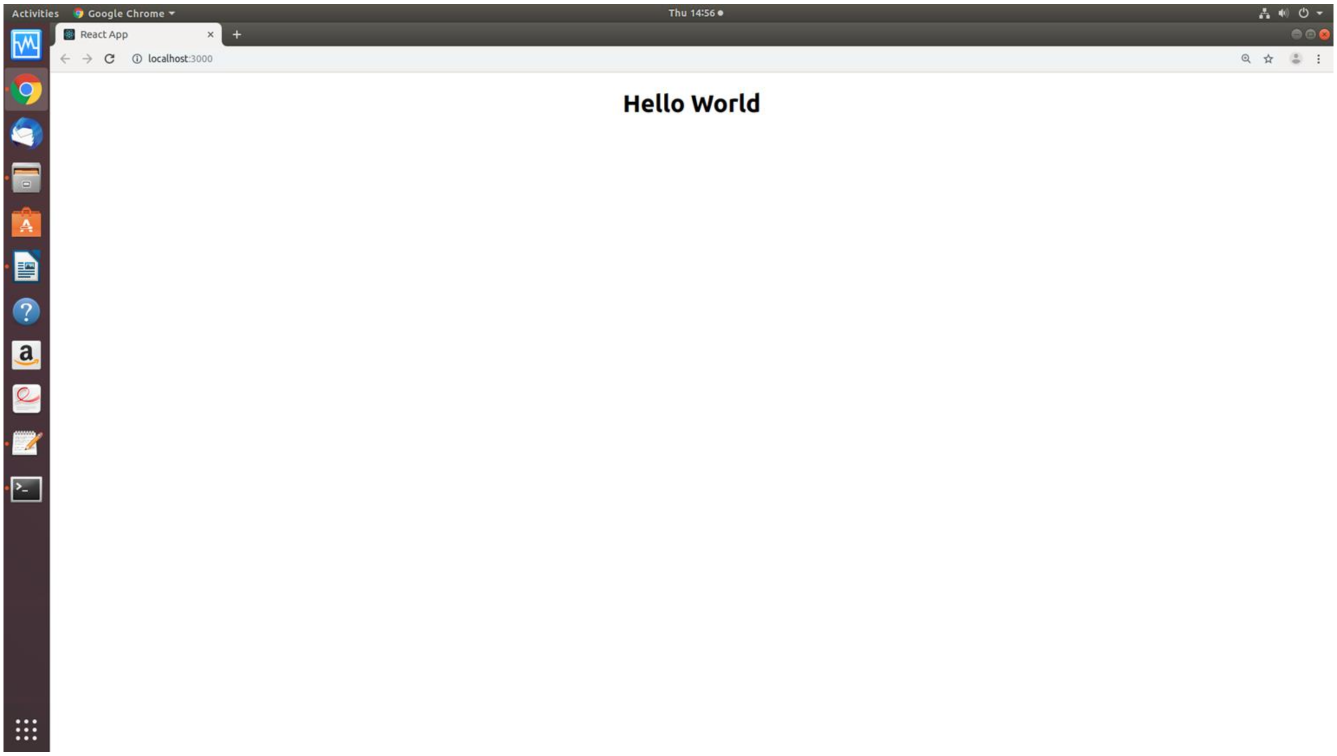


```
Compiled with warnings.

[eslint]
src/App.js
  Line 1:8:  'logo' is defined 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.

WARNING in [eslint]
src/App.js
  Line 1:8:  'logo' is defined but never used  no-unused-vars

webpack compiled with 1 warning
```



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

Learn React



Hello World

Exercise 2: Create simple web server application using Node js

Aim: Create simple web server application using Node js

Procedure

Step 1: First, we need to set up an accessible coding environment to do our exercises, as well as the others in the article. In the terminal, create a folder called first-servers:

```
mkdir first-servers
```

Step 2: Then enter that folder using below comment

```
cd first-servers
```

Step 3: Now, create the file that will house the code:

```
touch hello.js
```

Step 4: Open the file in a text editor. We will use nano as it's available in the terminal:

```
nano hello.js
```

We start by loading the http module that's standard with all Node.js installations. Add the following line to hello.js:

hello.js

```
const http = require('http');

const hostname = 'localhost';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

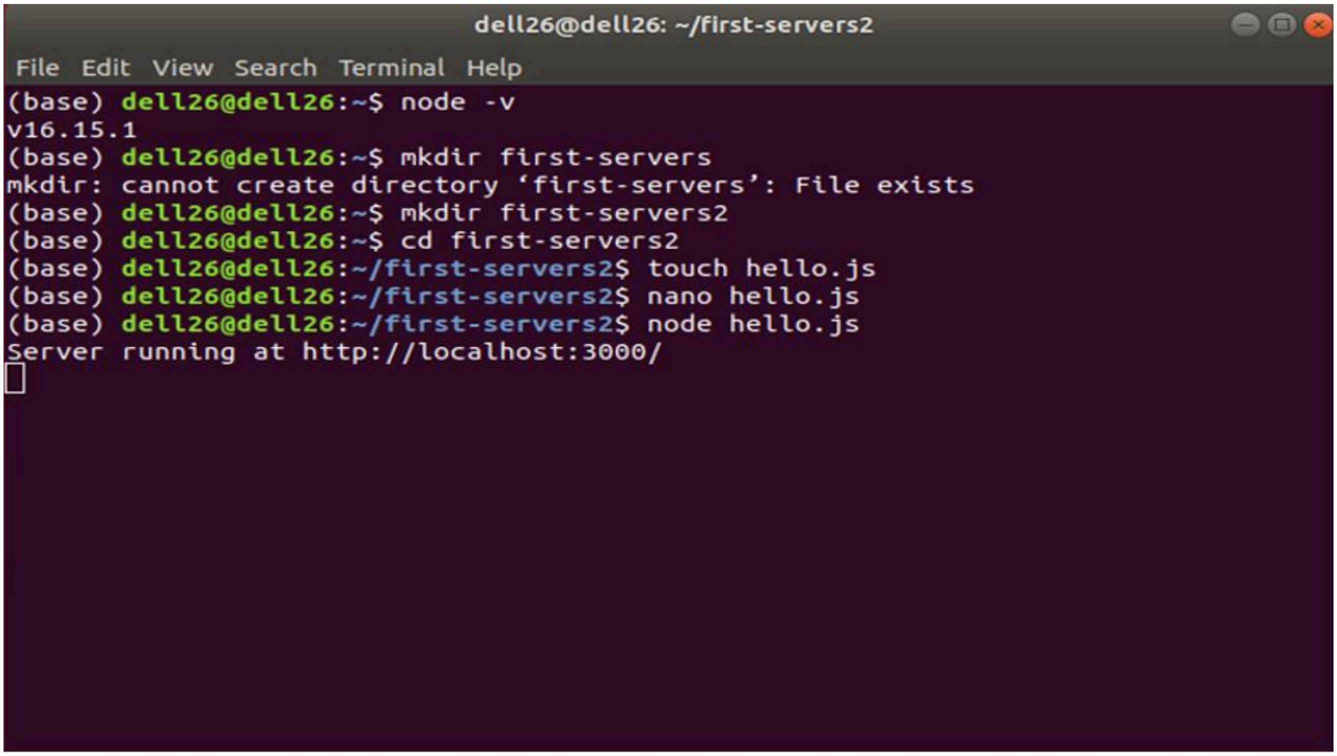
Save the file and exit the editor.

This Node.js application listens on the specified address (localhost) and port (3000), and returns “Hello World!” with a 200 HTTP success code. Since we’re listening on localhost, remote clients won’t be able to connect to our application.

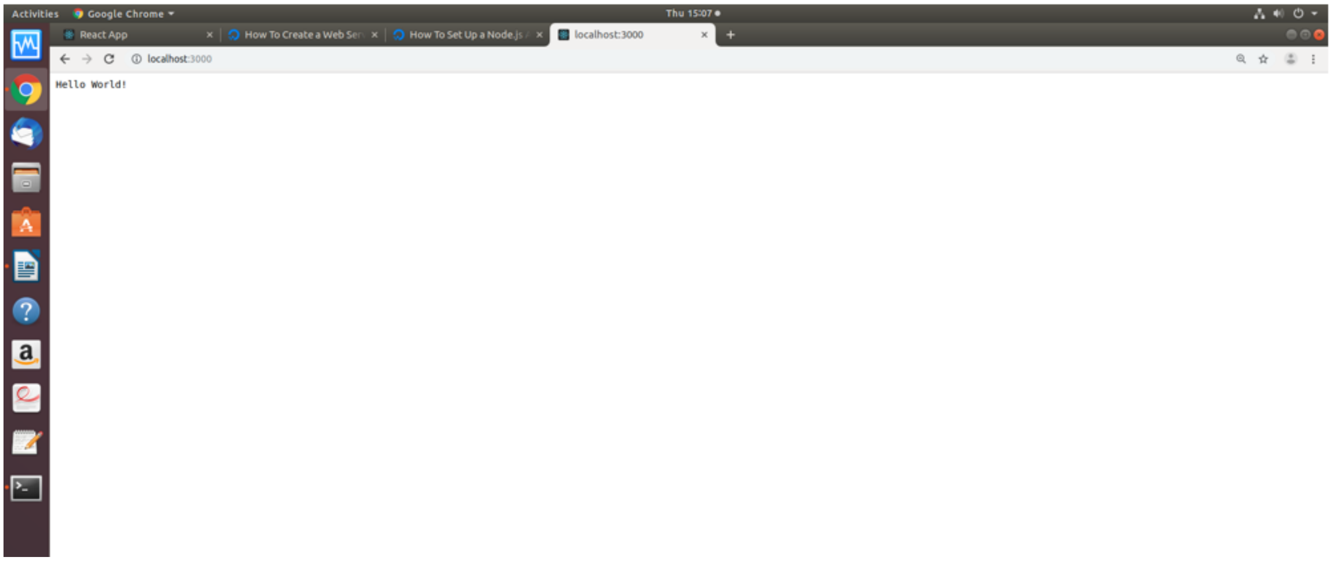
To test your application, type:

node hello.js

Output:
Server running at http://localhost:3000/


dell26@dell26: ~/first-servers2
File Edit View Search Terminal Help

```
(base) dell26@dell26:~$ node -v
v16.15.1
(base) dell26@dell26:~$ mkdir first-servers
mkdir: cannot create directory ‘first-servers’: File exists
(base) dell26@dell26:~$ mkdir first-servers2
(base) dell26@dell26:~$ cd first-servers2
(base) dell26@dell26:~/first-servers2$ touch hello.js
(base) dell26@dell26:~/first-servers2$ nano hello.js
(base) dell26@dell26:~/first-servers2$ node hello.js
Server running at http://localhost:3000/
```


Activities
Google Chrome
Thu 15:07
React App
How To Create a Web Ser
How To Set Up a Node.js
localhost:3000
localhost:3000

```
Hello World!
```