



Experiment No. 08

Title: Design a web page using React JS.



Batch:B1**Roll No:16010420133****Experiment No:8****Aim:** To design a web page using React JS.

Resources needed: Notepad, any Web Browser and Internet.

Theory:

React (also known as React.js or ReactJS) is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. 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(Document Object Model), so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality. ReactJS is JavaScript library used for building reusable UI components.

Features of React

- **JSX** – JSX is JavaScript syntax extension. It isn't necessary to use JSX in React development, but it is recommended.
- **Components** – React is all about components. You need to think of everything as a component. This will help you maintain the code when working on larger scale projects.
- **Unidirectional data flow and Flux** – React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional.
- **License** – React is licensed under the Facebook Inc. Documentation is licensed under CC BY 4.0.

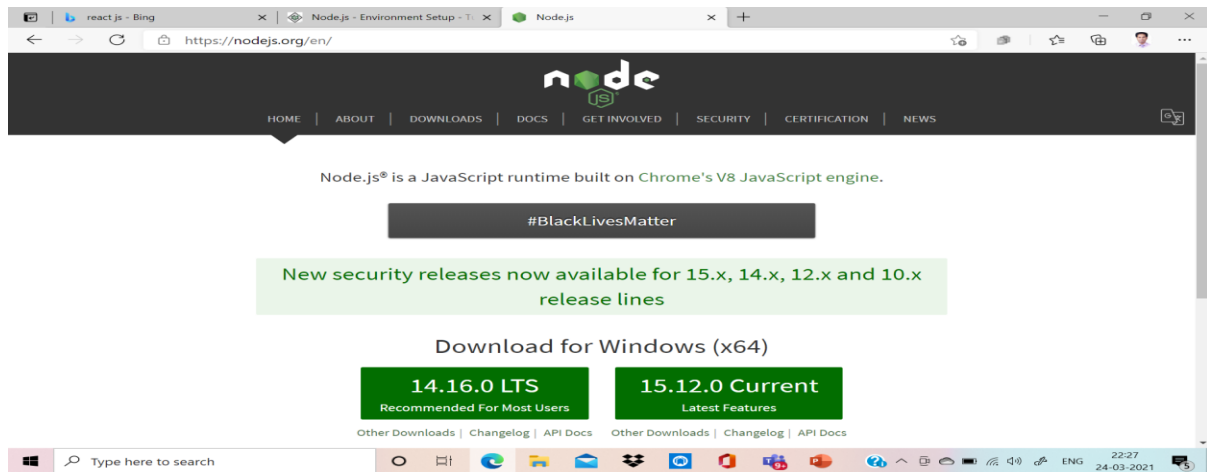
ReactJS - Environment Setup

1. First you need to install NodeJS
2. Second install ReactJS

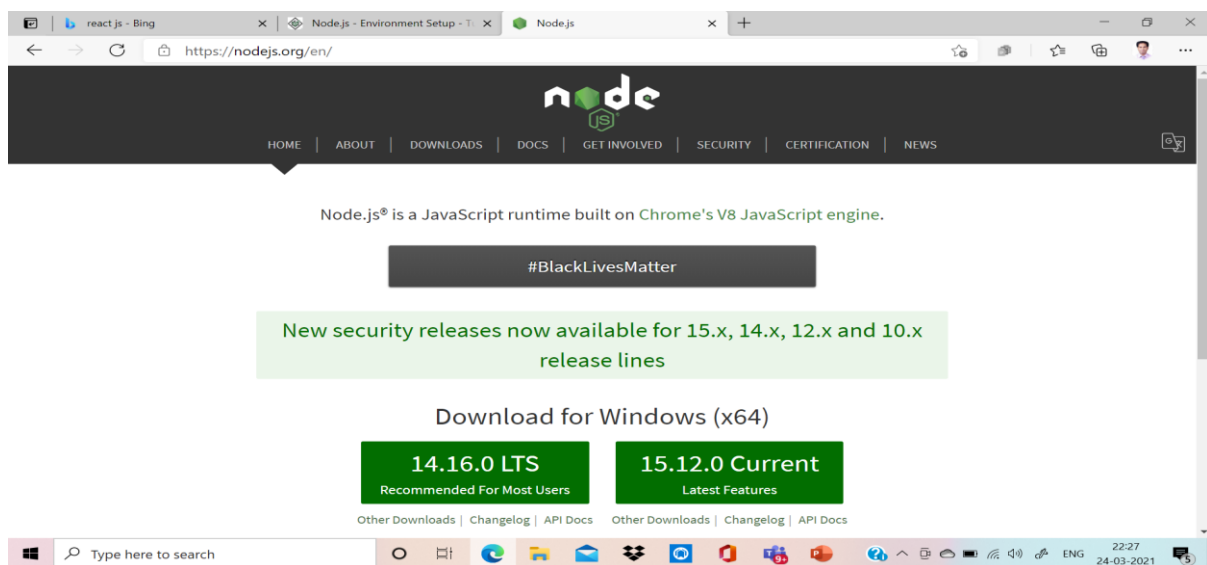
Install NodeJS:**Step 1:**

Visit the website www.Nodejs.org/en/. For installation on Windows ,you use the MSI file and follow the prompts to install the Node.js. By default, the installer uses the Node.js distribution in C:\Program Files\nodejs. The installer should set the C:\Program Files\nodejs\bin directory in window's PATH environment variable. Restart any open command prompts for the change to take effect. The source code written in source file is simply javascript. The Node.js interpreter will be used to interpret and execute your javascript code. Node.js distribution comes as a binary installable for SunOS , Linux, Mac OS X, and Windows operating systems with the 32-bit (386) and 64-bit (amd64) x86 processor architectures. Next step will guide to install Node.js binary distribution on windows OS.

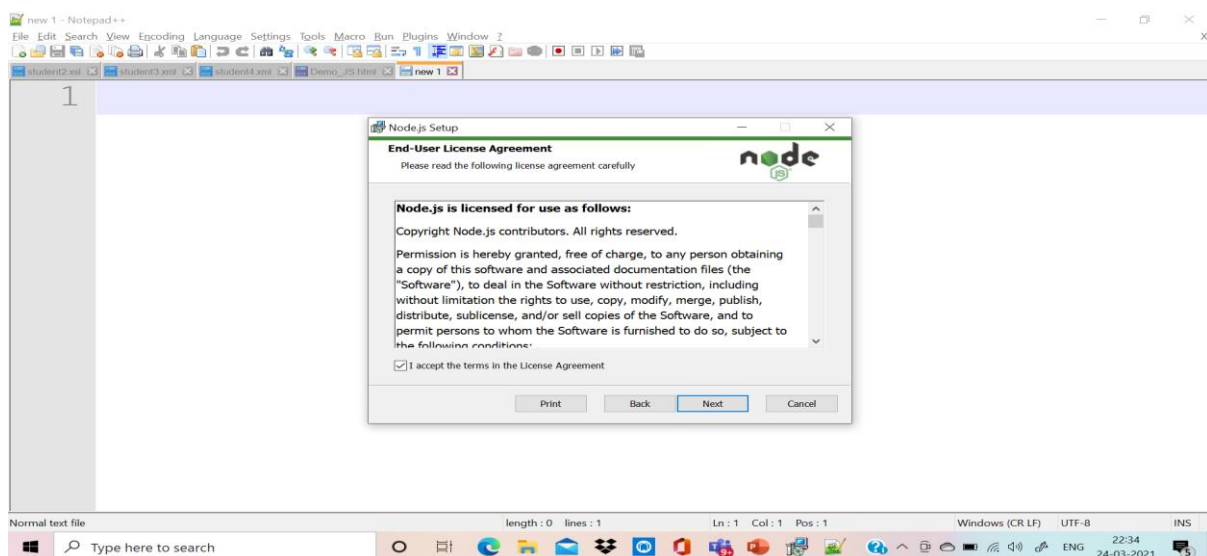
Step 2:



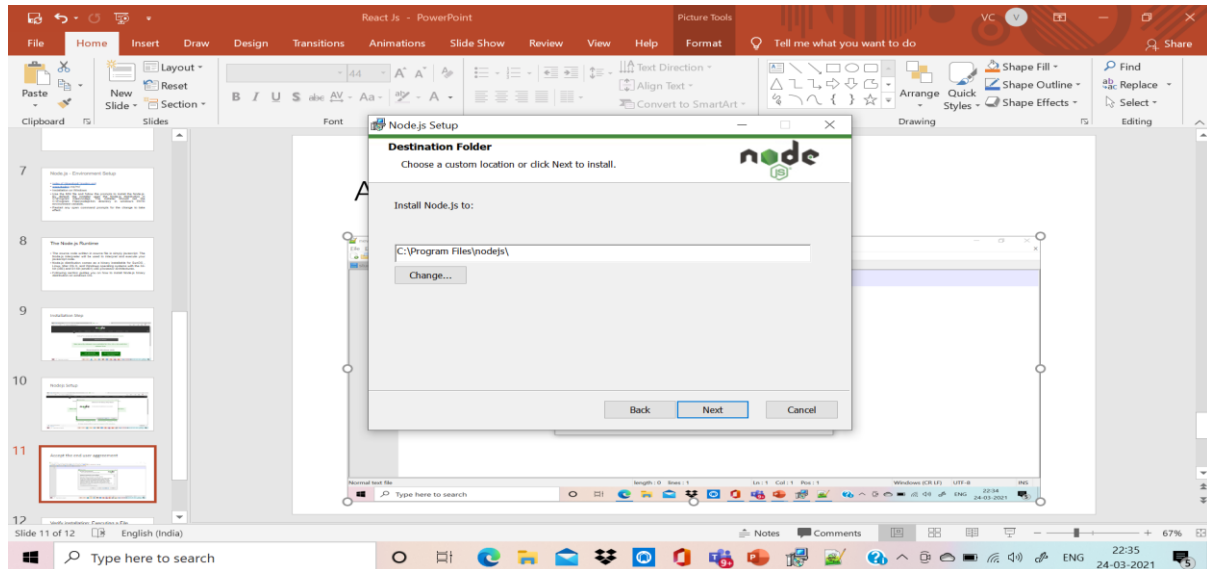
Step3:



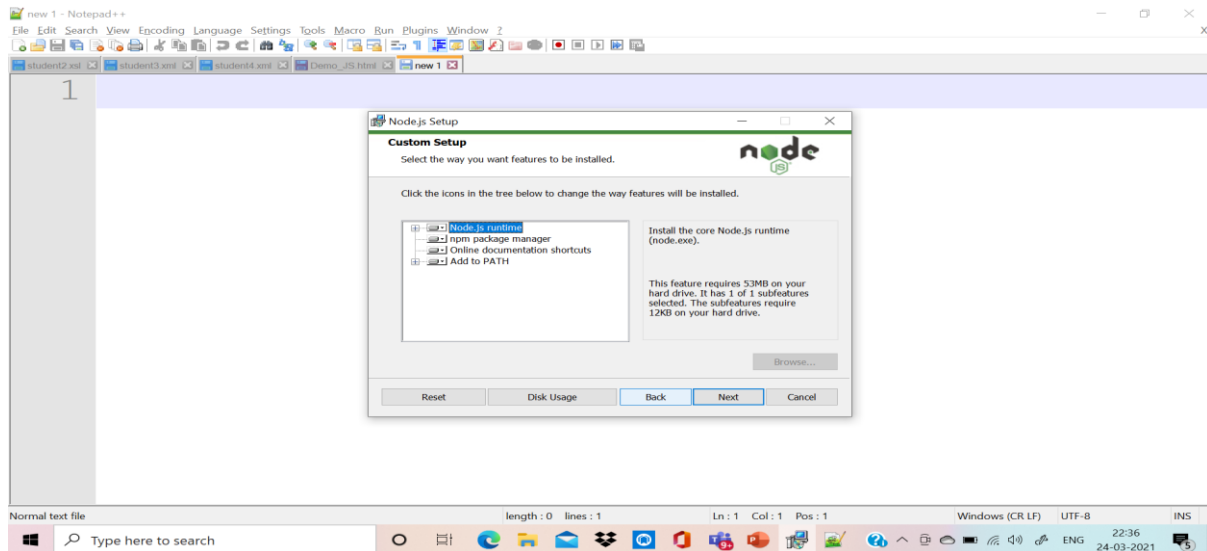
Step4: Accept the Agreement



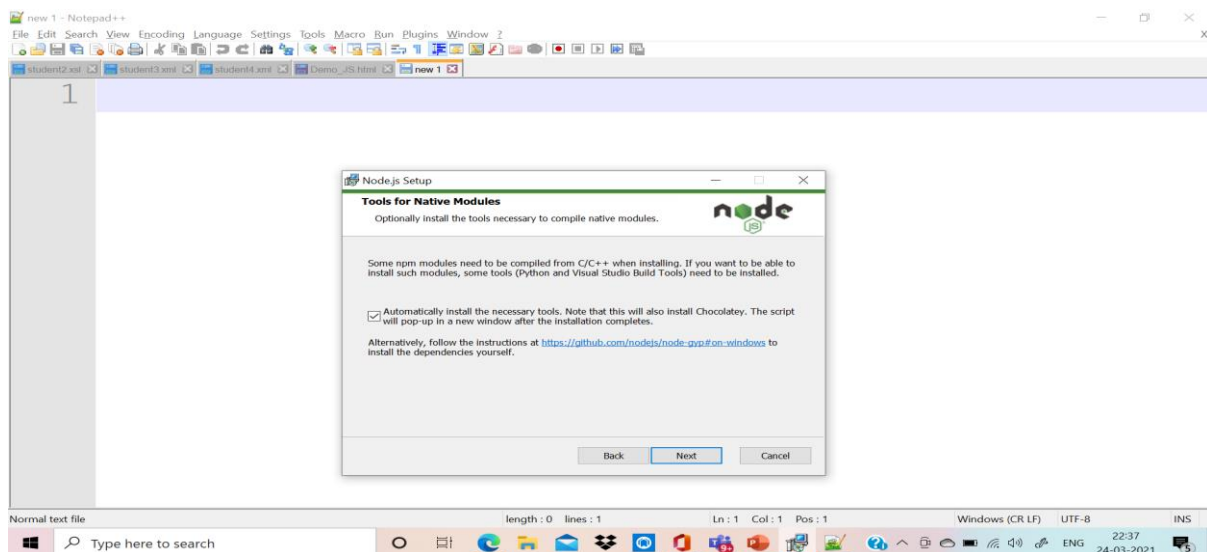
Step5: Choose Destination Folder

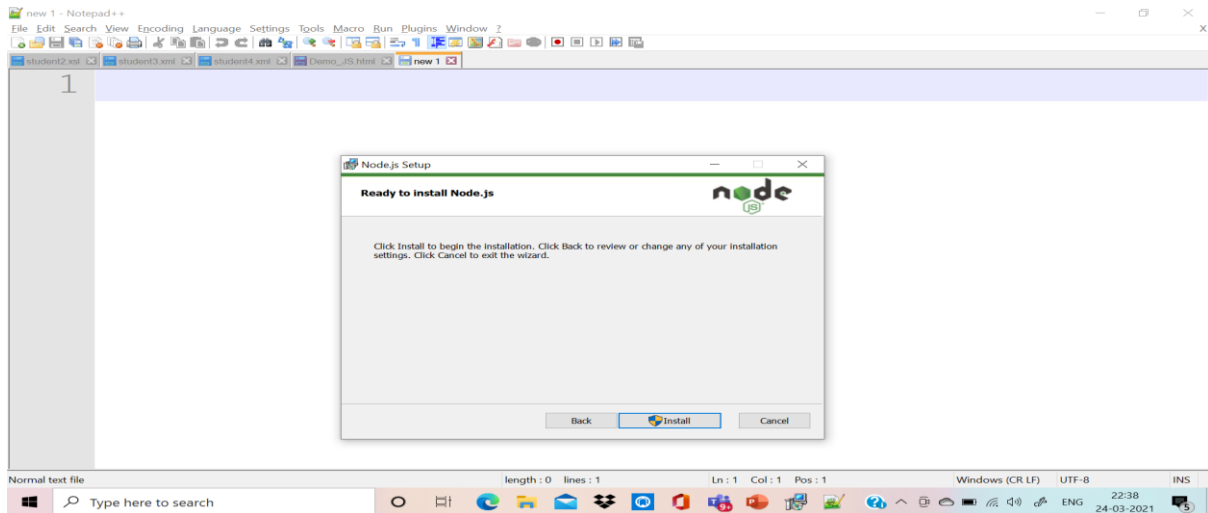
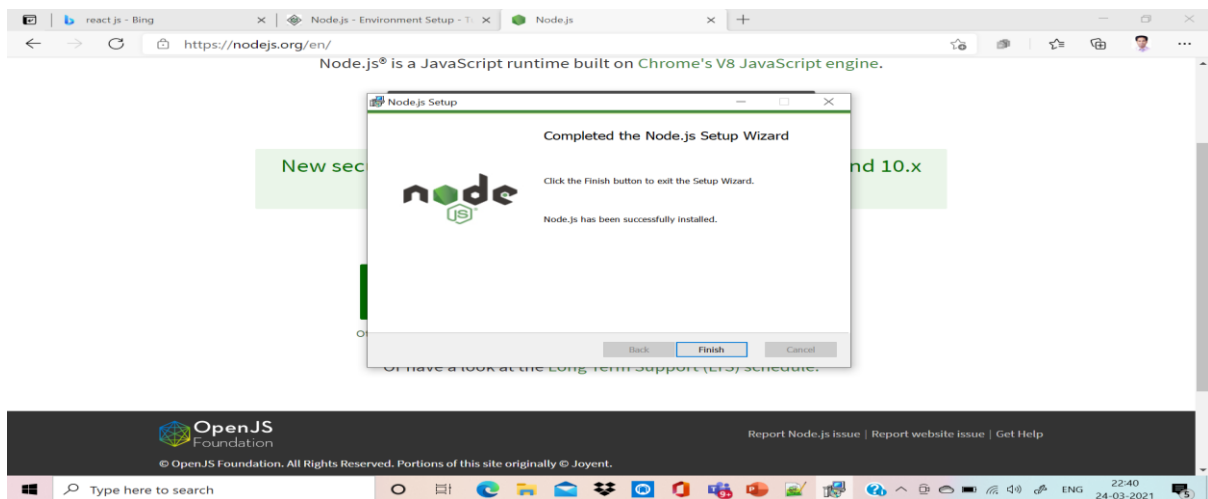
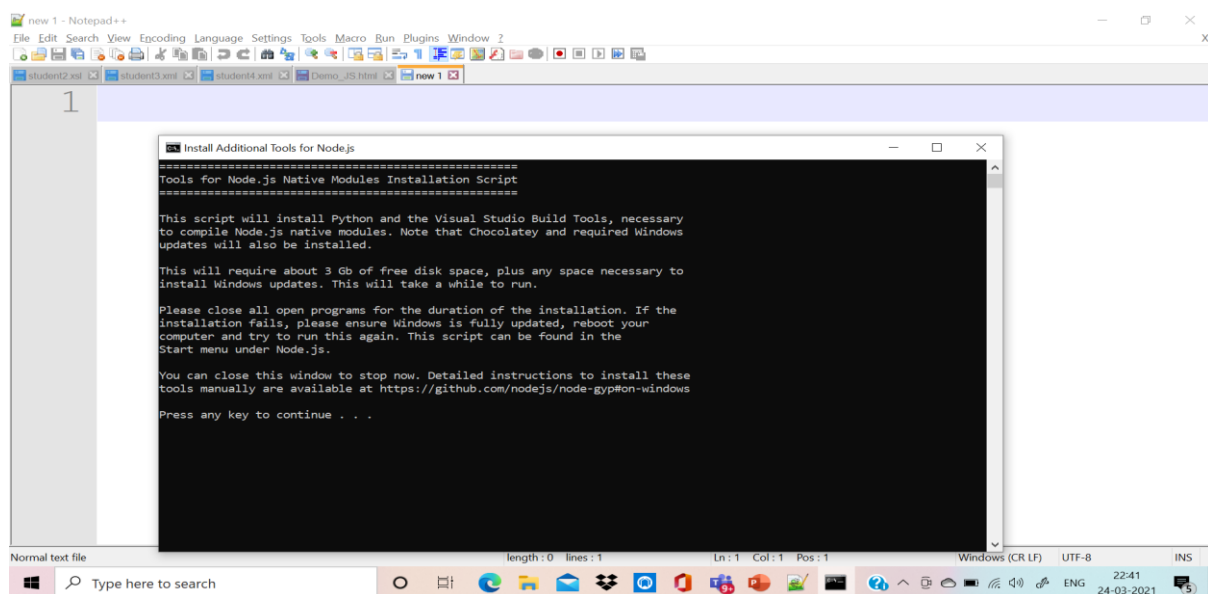


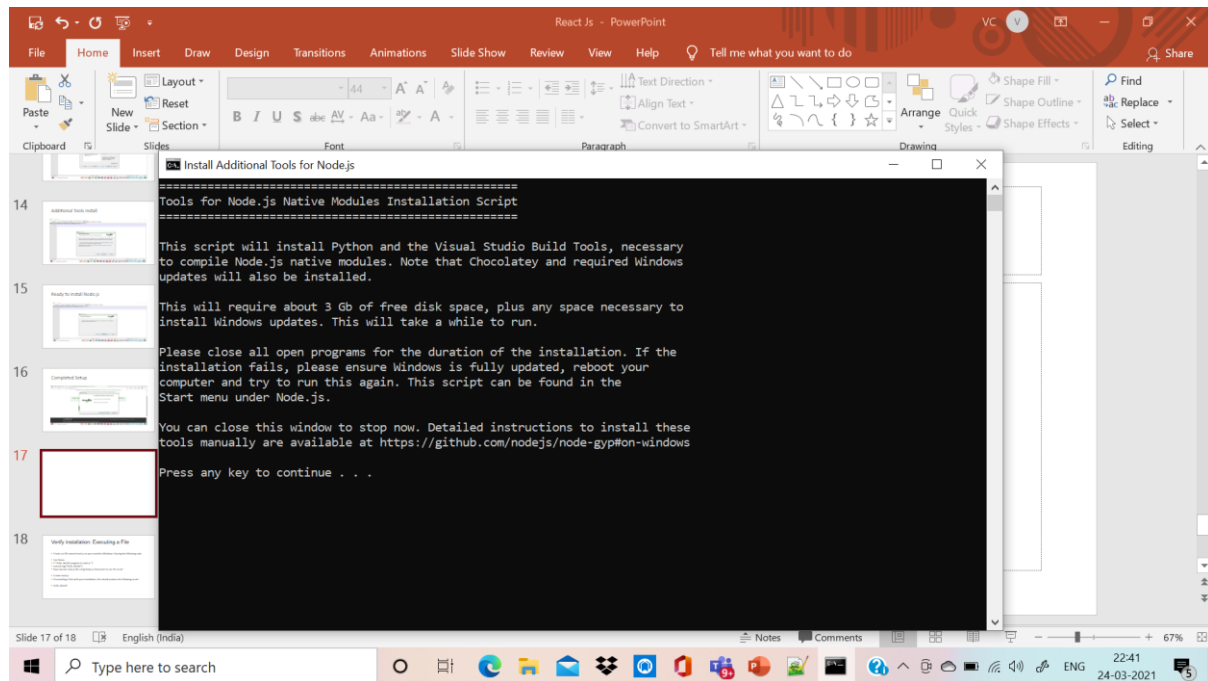
Step6: Custom Installation setup



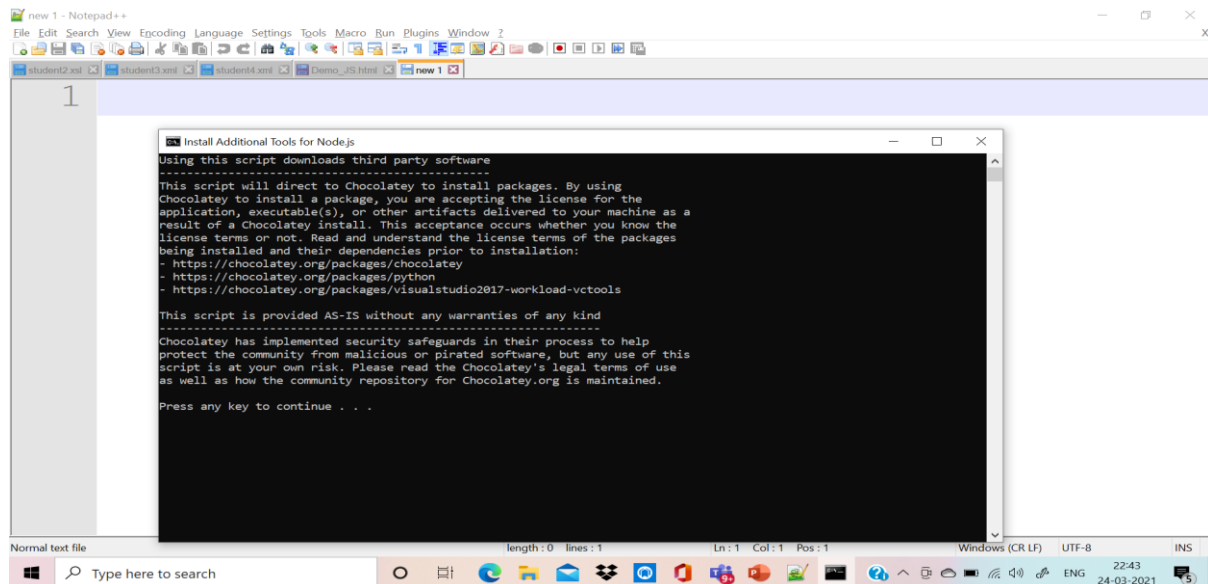
Step 7: Additional Tools Setup



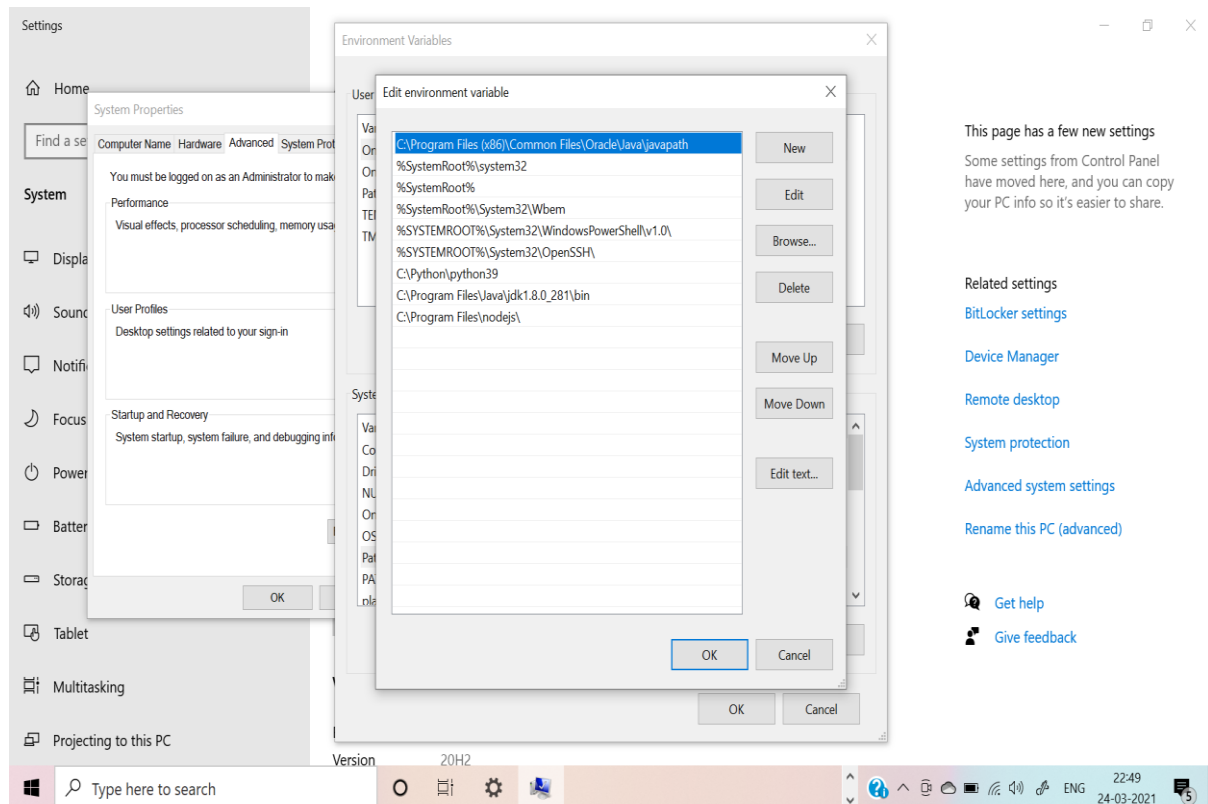
Step8: Ready to install Nodejs**Step9: Completed Setup****Step 10: Ready to start****Step11: Install Native Module**



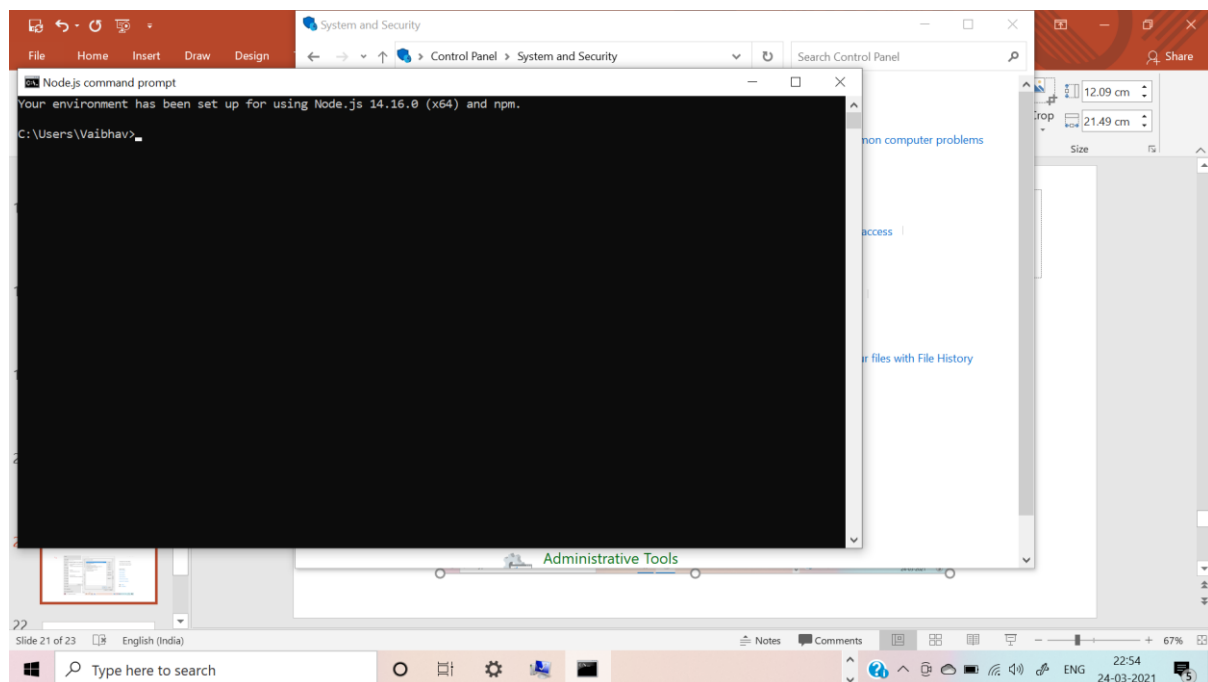
Step 12: Additional Installation



Step 13: Verify the setup



Step 14: Setup Message on Command Prompt: For this message go start menu and click the button you will find command prompt menu available with cmd .



Step 15: Check the version

•C:\>node -v

•➔v14.16.0

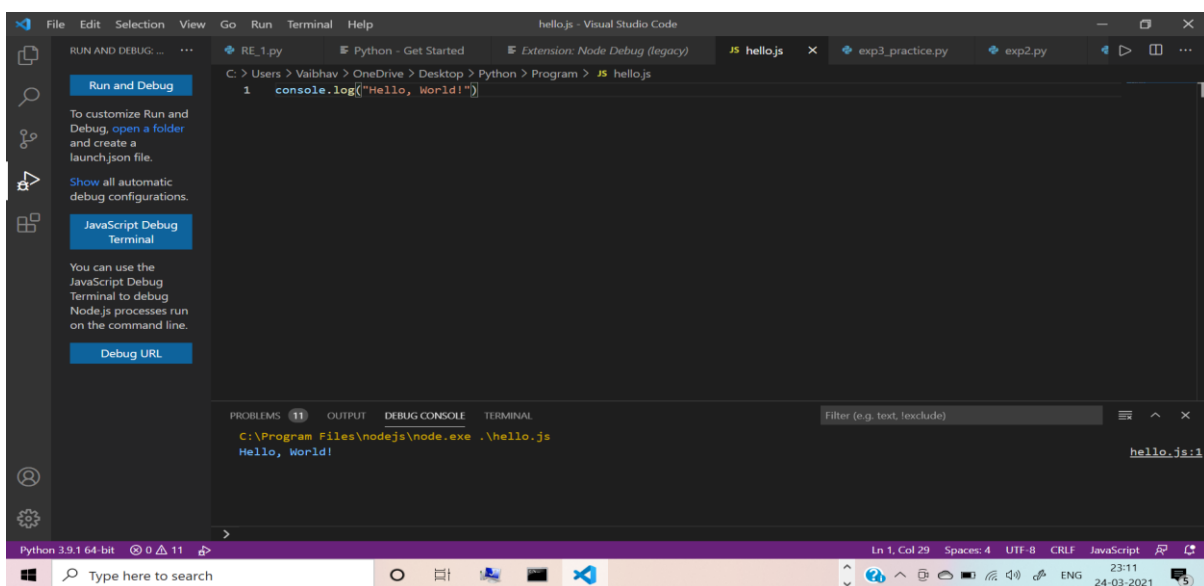
Step 16: Type the Hello World and execute

Type the command and install gyrescript :

C:\>npm install -g typescript

- Create a js file named main.js on your machine (Windows) having the following code.
- Live Demo
- `/* Hello, World! program in node.js */`
- `console.log("Hello, World!")`
- Now execute main.js file using Node.js interpreter to see the result –
- `$ node main.js`
- If everything is fine with your installation, this should produce the following result –
- Hello, World!

Step 17: Install VS Code: Editor Tool



Install ReactJS:

There are 3 ways to install ReactJS :

- 1) webpack
- 2) Babel
- 3) create-react-app

Webpack:

Webpack is a module bundler (manages and loads independent modules). It takes dependent modules and compiles them to a single (file) bundle. You can use this bundle while developing apps using command line or, by configuring it using webpack.config file.

Steps:

- 1) Since we are using webpack to generate bundler install webpack, webpack-dev-server and webpack-cli.
- 2) `C:\Users\username\Desktop\reactApp>npm install webpack --save`

- 3) C:\Users\username\Desktop\reactApp>npm install webpack-dev-server --save
- 4) C:\Users\username\Desktop\reactApp>npm install webpack-cli --save
- 5) Or, you can install all of them in single command as –
- 6) C:\Users\username\Desktop\reactApp>npm install webpack webpack-dev-server --save

Babel:

Babel is a JavaScript compiler and transpiler. It is used to convert one source code to other. Using this you will be able to use the new ES6 features in your code where, babel converts it into plain old ES5 which can be run on all browsers.

- 1) Install babel, and its plugins babel-core, babel-loader, babel-preset-env, babel-preset-react and, html-webpack-plugin
- 2) C:\Users\username\Desktop\reactApp>npm install babel-core --save-dev
- 3) C:\Users\username\Desktop\reactApp>npm install babel-loader --save-dev
- 4) C:\Users\username\Desktop\reactApp>npm install babel-preset-env --save-dev
- 5) C:\Users\username\Desktop\reactApp>npm install babel-preset-react --save-dev
- 6) C:\Users\username\Desktop\reactApp>npm install html-webpack-plugin --save-dev
- 7) Or, you can install all of them in single command as –
- 8) C:\Users\username\Desktop\reactApp>npm install babel-core babel-loader babel-preset-env
- 9) babel-preset-react html-webpack-plugin --save-dev

Working with ReactJS

In this experiment , we are going to use this steps of create react app

Steps are as follows:

- 1) Create a folder with name reactApp on the desktop to install all the required files, using the mkdir command.

```
C:\Users\username\Desktop>mkdir reactApp
```

- 2) Change the directory:

```
C:\Users\username\Desktop>cd reactApp
```

```
C:\Users\vaibhav>cd C:\Users\vaibhav\reactapp\
```

- 3) Install ReactJS:

```
C:\Users\vaibhav\reactapp>npx create-react-app my-app
```

This will create a folder named my-app on the desktop and installs all the required files in it.

npm (node package manager) is the dependency/package manager you get out of the box when you install Node.js. It provides a way for developers to install packages both globally and locally

npx: The npx stands for Node Package Execute and it comes with the npm, when you installed npm above 5.2.0 version then automatically npx will installed. It is an npm package runner that can execute any package that you want from the npm registry without even installing that package. The npx is useful during a single time use package. If you have installed npm below 5.2.0 then npx is not installed in your system.

4) Delete all source files

- Browse through the src folder in the generated my-app folder and remove all the files in it as shown below –
- `C:\Users\Desktop>cd my-app/src`
- `C:\Users\Desktop\my-app\src>del *`
- `C:\Users\Desktop\my-app\src*, Are you sure (Y/N)? y`

5) Add files with names index.css and index.js in the src folder as –









- `C:\Users\Desktop\my-app\src>type nul > index.css`
- `C:\Users\Tutorialspoint\Desktop\my-app\src>type nul > index.js`

6) Skip step 4 and 5 and only delete the file called App.js

7) Locate your code into these folder as event1 as shown below:

This PC > Windows (C:) > Users > Vaibhav > reactapp > event1				
	Name	Date modified	Type	Size
	node_modules	08-04-2021 11:54	File folder	
	public	08-04-2021 11:49	File folder	
	src	08-04-2021 11:49	File folder	
	.gitignore	01-04-2021 11:20	GITIGNORE File	1 KB
	package.json	08-04-2021 11:50	JSON File	1 KB
	README.md	08-04-2021 11:49	MD File	4 KB
	yarn.lock	08-04-2021 11:50	LOCK File	496 KB

8) Open the src folder as shown below:

This PC > Windows (C:) > Users > Vaibhav > reactapp > event1 > src				
Name	Date modified	Type	Size	
 App	01-04-2021 11:20	Cascading Style Shee...	1 KB	
 App	08-04-2021 11:53	JavaScript File	1 KB	
 App.test	01-04-2021 11:20	JavaScript File	1 KB	
 index	01-04-2021 11:20	Cascading Style Shee...	1 KB	
 index	01-04-2021 11:20	JavaScript File	1 KB	
 logo	01-04-2021 11:20	SVG Document	3 KB	
 reportWebVitals	01-04-2021 11:20	JavaScript File	1 KB	
 setupTests	01-04-2021 11:20	JavaScript File	1 KB	

9) Choose the App.js file and indite code which is given below.

```
import React, {Component} from 'react';
```

```
class App extends React.Component {
```

```
  constructor(props) {
```

```
    super(props);
```

```
    this.state = {
```

```
      companyName: "
```

```
    };
```

```
  }
```

```
  changeText(event) {
```

```
    this.setState({
```

```
      companyName: event.target.value
```

```
    });
```

```
  }
```

```
  render() {
```

```
    return (
```

```
      <div>
```

```
        <h2>Simple Event Example</h2>
```

```
        <label htmlFor="name">Enter company name: </label>
```

```
<input type="text" id="companyName" onChange={this.changeText.bind(this)} />
```

```
<h4>You entered: { this.state.companyName }</h4>
```

```
</div>
```

```
);
```

```
}
```

```
} export default App;
```

10 .Execute the React Code:

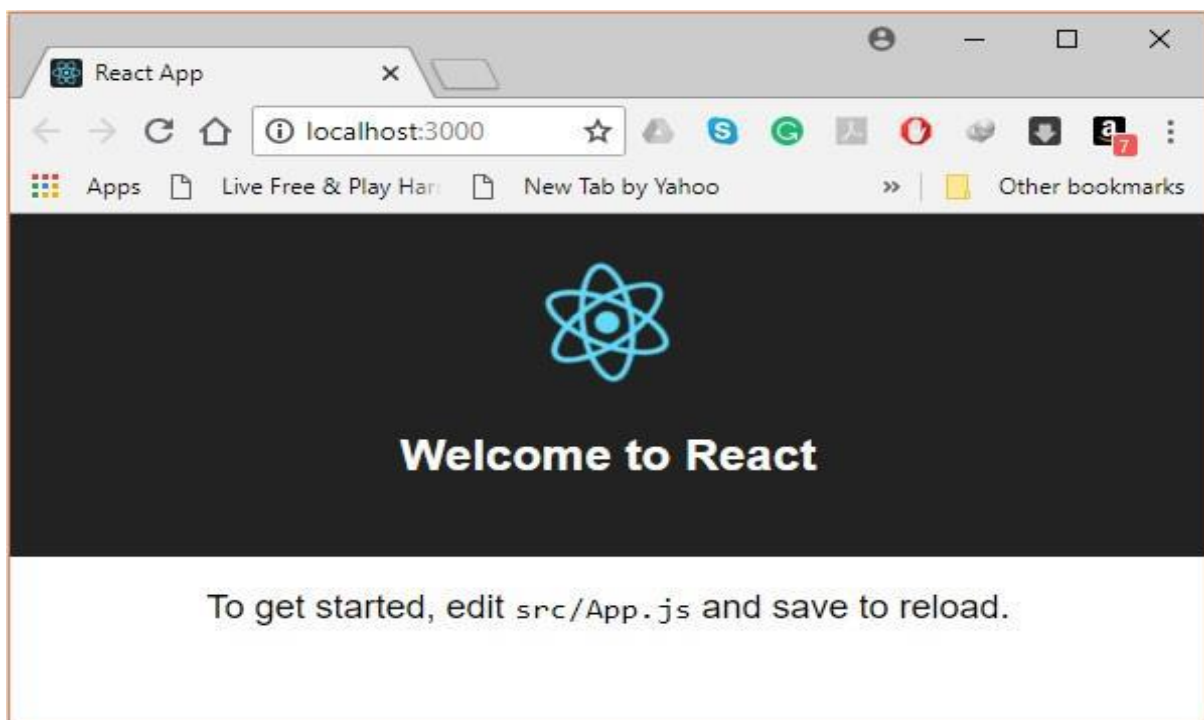
To run the react code you have to type the command on Vs Code Terminal as it is:

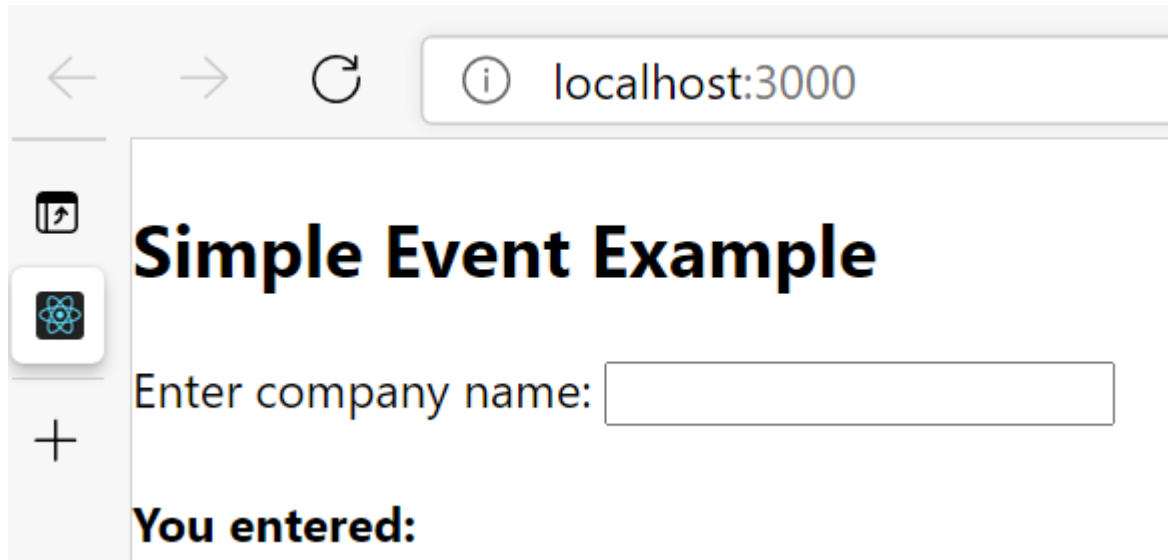
- npm start
- yarn start(if you install yarn utility)

To do this on, terminal of VS CODE,

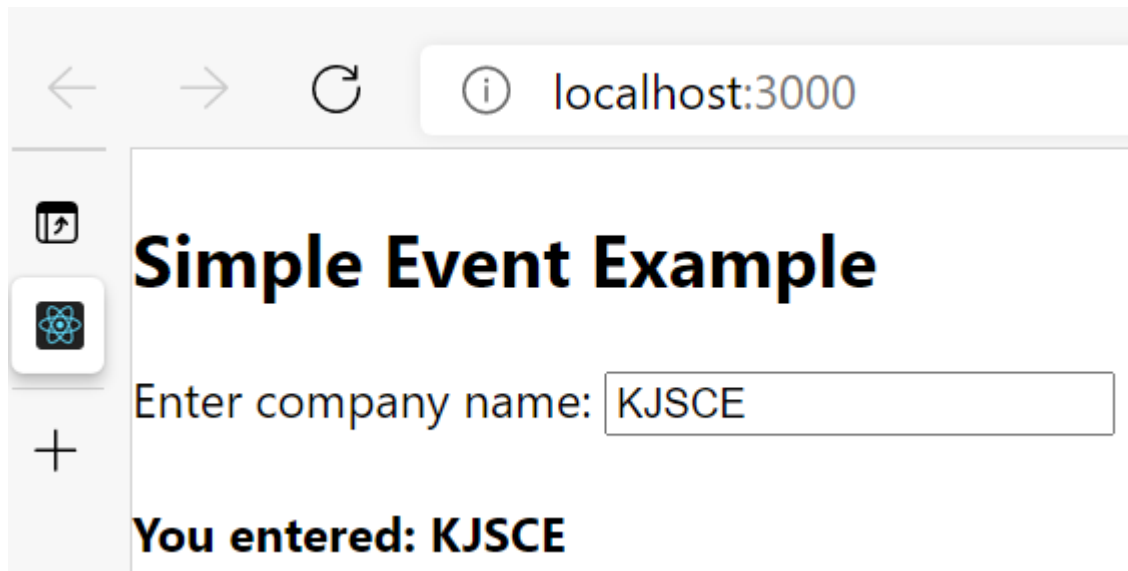
C:\Users\Vaibhav\reactapp\myevent\npm start.

By default port number 3000 will start on web browser with the code execution.





11. After running the event the outlook:

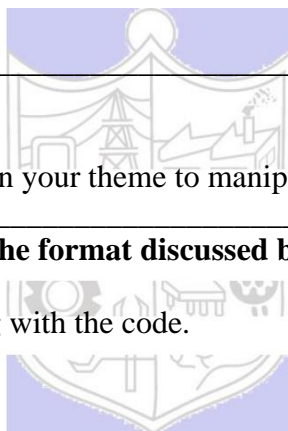


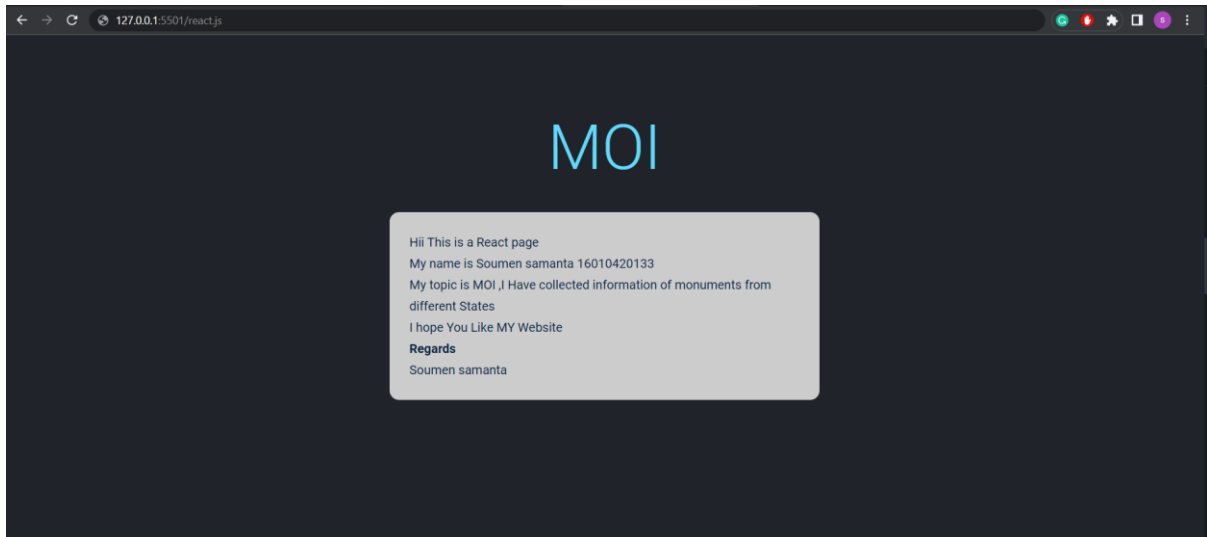
Activities:

- To design a web page using React JS on your theme to manipulate the DOM elements.

Results: (Document printout as per the format discussed by the faculty t)

Display of the designed webpage along with the code.





```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <title>React</title>
    <style>@import
url("https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;700&display=
swap");
    * {
      font-family: "Roboto", sans-serif;
    }
    body {
      background: #20232a;
    }
    .container h1 {
      color: rgb(97, 218, 251);
      font-size: 5em;
      font-weight: lighter;
      text-align: center;
    }
    .container {
      max-width: 600px;
      margin: 0 auto;
      padding: 25px;
    }
    .container > p {
      color: #fff;
    }
    .welcome-back {
      margin: 35px auto;
      background: #ccc;
      border-radius: 12px;
      padding: 25px;
    }
    .welcome-back h2 {
      margin: 0 0 35px;
```

```

    text-align: center;
    font-size: 2.5em;
    font-weight: normal;
  }
  .welcome-back button {
    font-size: 2em;
    background: #20232a;
    color: rgb(97, 218, 251);
    border: none;
    padding: 15px;
    width: 100%;
    max-width: 300px;
    margin: 0 auto;
    cursor: pointer;
    display: block;
    border-radius: 12px;
    transition: 0.3s;
  }
  .welcome-back button:hover {
    color: #fff;
    background: #000;
  }
  .welcome-back > p {
    font-size: 1.2em;
    text-align: center;
    margin: 0 0 25px;
    color: #980000;
  }
  .welcome-back p.downloading {
    font-size: 2em;
    color: #333;
  }
</style>
</head>
<body>
  <div class="container">
    <h1>MOI</h1>
    <div class="welcome-back">
      Hii This is a React page<br>
      My name is Soumen samanta 16010420133 <br>
      My topic is MOI ,I Have collected information of monuments from
different States<br>

      I hope You Like MY Website <br>

      <strong>Regards</strong><br>
      Soumen samanta<br>
    </div>
  </div>

  <script
    src="https://unpkg.com/react@16/umd/react.production.min.js"
    crossorigin
  ></script>

```

```

<script
  src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"
  crossorigin
></script>
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

<script type="text/babel" src="WelcomeBack.js"></script>
<script type="text/babel" src="CoolButton.js"></script>

<script type="text/babel">
  ReactDOM.render(<WelcomeBack />, document.querySelector(".welcome-
back"));
</script>
</body>
</html>

```

Questions:

1.What are the different components of ReactJS?

1] Functional Components:

In simple words, Functional components are javascript functions. By writing a javascript function, we can create a functional component in React Apps. To make React app efficient, we use functional component only when we are sure that our component does not require to interact with any other components. Functional components do not require data from other components. Below shows an example of functional component in React:

```

function Title()
{
  return <h1>I am Title</h1>;
}

```

2] Class Components

The class components are similar to the functional component but has some additional features that makes class component a little more complex than the functional components. The functional components do not care about the other components in your app whereas the class components can work with each other. We can pass data from one class

component to other class component. Below shows an example of class component in React:

```
class Title extends React.Component
{
  render() {
    return <h1>I am Title</h1>;
  }
}
```

3] Higher-Order Components

Higher order components, or known under the abbreviation HOCs are the component which takes a component as input and returns the component as output but with extended functionalities. React Higher-Order Components are popular advanced React pattern to deploy reusable logic and functionality across React components. Below shows an example higher-order component that transforms and returns usernames in uppercase:

```
const hoc = (MyComponent) => (props) => {
  return (
    <div>
      <MyComponent {...props}>
        {props.children.toUpperCase()}
      </MyComponent>
    </div>
  )
}
```

4] Dumb Components

A Dumb Component can very easily be defined as a stateless functional component. A stateless component is much more efficient than a stateful one, because it doesn't require as many computer resources to render (memory, CPU, and GPU in terms of graphic-intensive apps). Below shows an example of dumb component in React:

```
export default Title => () => (
  <h1>I am Title</h1>
);
```

5] Smart Components

A Smart Component is any class component that manages its own state. Smart Components are stateful components and works similar to class components. When working with Babel

or ES6-style React, we've come to know this as any class-like object that extends Component.

This includes either React.Component or in our case

Other.Component. Below shows an example of smart

component in React:

```
export default title => class MyComponent extends
Other.Component {
  render() {
    return (
      <h1>I am Title</h1>
    );
  }
}
```

6] Presentational Components

The presentation component is often called as stateless functional component that takes props and renders UI. A stateless functional components are plain JavaScript functions that do not have states. The components that receive state from the higher-order component will function as presentational components. State gets passed to them and they conditionally render UI based on it. They do not bother with the management of state. Presentational Component mainly concerned with how things look. Below shows an example of presentational component in React:

```
const List = props =>
  (<ul>
    {props.list.map(user =>
      (<li>{items}</li>    ))
    }
  </ul>)
```

7] Container components

Container component is a class component that provides the data and behavior to presentational or other container components. A container does data fetching and then renders its corresponding sub-component. This component mainly concerned with how things work. Container components call flux actions and provides these as callbackss to the

presentational component. Below shows an example of

presentational component in React:

```
class ListContainer extends React.Component{
  constructor()
  {
    this.state = {
      items: []
    }
  }
  componentDidMount() {
    axios.get('/list').then(
      items=>this.setState({ list: items}))
  )
}
render() {
  return<Usersusers={this.state.items} />
  }}
}
```

2.What is Virtual DOM? How virtual DOM Works? What is the purpose of render of react DOM?

The virtual DOM (VDOM) is a programming concept where an ideal, or “virtual”, representation of a UI is kept in memory and synced with the “real” DOM by a library such as ReactDOM. This process is called reconciliation.

WORKING

So when there is a update in the virtual DOM, react compares the virtual DOM with a snapshot of the virtual DOM taken right before the update of the virtual DOM.

With the help of this comparison React figures out which components in the UI needs to be updated. This process is called diffing. The algorithm that is used for the diffing process is called as the diffing algorithm.

Once React knows which components has been updated, then it replaces the original DOM nodes with the updated DOM node.

PURPOSE

render() controls the contents of the container node you pass in. Any existing DOM elements inside are replaced when first called. Later calls use React’s DOM diffing algorithm for efficient updates.

Outcomes:

Conclusion:

(Conclusion to be based on objectives and outcomes achieved)

Grade: AA/AB/BB/BC/CC/CD/DD/FF

Signature of faculty in-charge with date

References: Books/ Journals/ Websites:

1. React – A JavaScript library for building user interfaces (reactjs.org)
2. "React - A JavaScript library for building user interfaces". *React*. Retrieved 7 April 2018.
3. *Krill, Paul (May 15, 2014)*. "React: Making faster, smoother UIs for data-driven Web apps". InfoWorld.
4. *Hemel, Zef (June 3, 2013)*. "Facebook's React JavaScript User Interfaces Library Receives Mixed Reviews". *InfoQ*.
5. *Dawson, Chris (July 25, 2014)*. "JavaScript's History and How it Led To ReactJS". *The New Stack*.
6. *Dere, Mohan (2018-02-19)*. "How to integrate create-react-app with all the libraries you need to make a great app". *freeCodeCamp*. Retrieved 2018-06-14.
7. React Tutorial (w3schools.com)
8. ReactJS - Overview - Tutorialspoint
9. ReactJS Tutorials - GeeksforGeeks