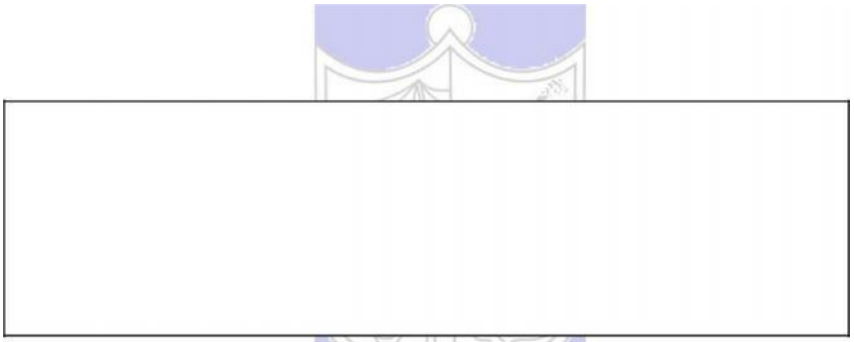


Batch:A2 **Roll No:16010421063** **Experiment No:**

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

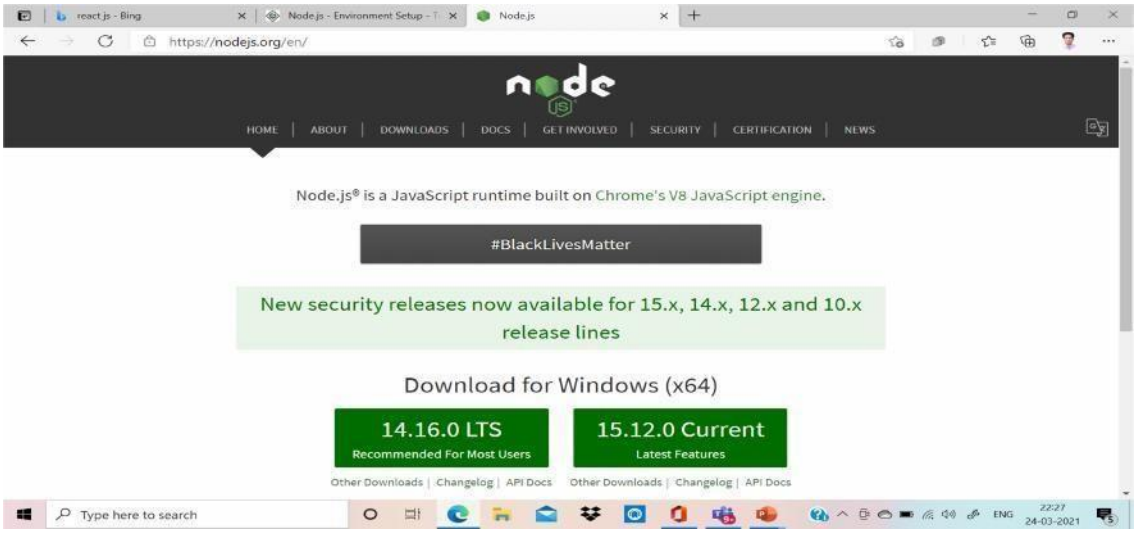
Install NodeJS:

Step 1:

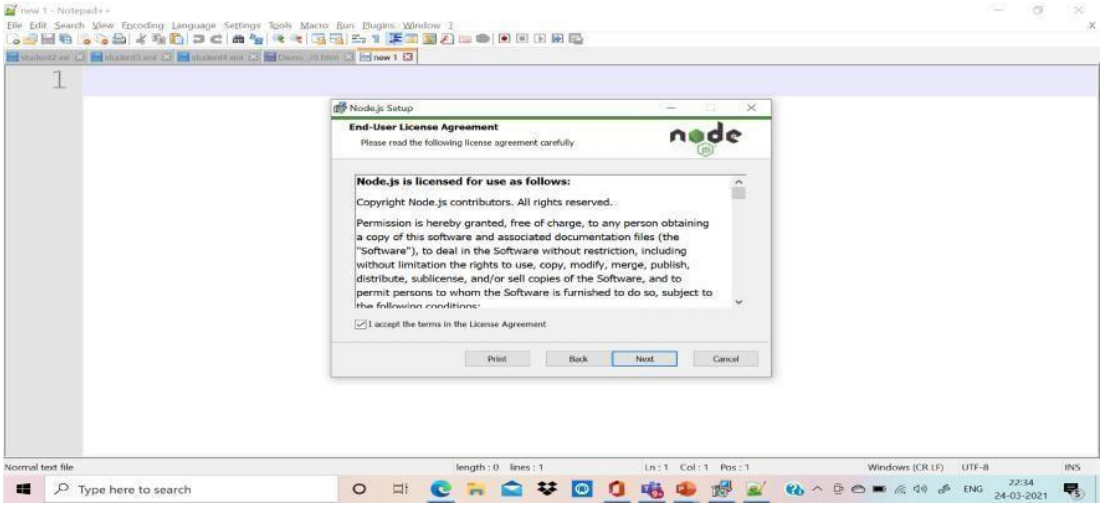
Visit the website www.Nodejs.org/en/. For installation on Windows ,you use the MSI file and follow the prompts to install the Node.js. By default, the installer uses the Node.js distribution in C:\Program Files\nodejs. The installer should set the C:\Program Files\nodejs\bin directory in window's PATH 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:

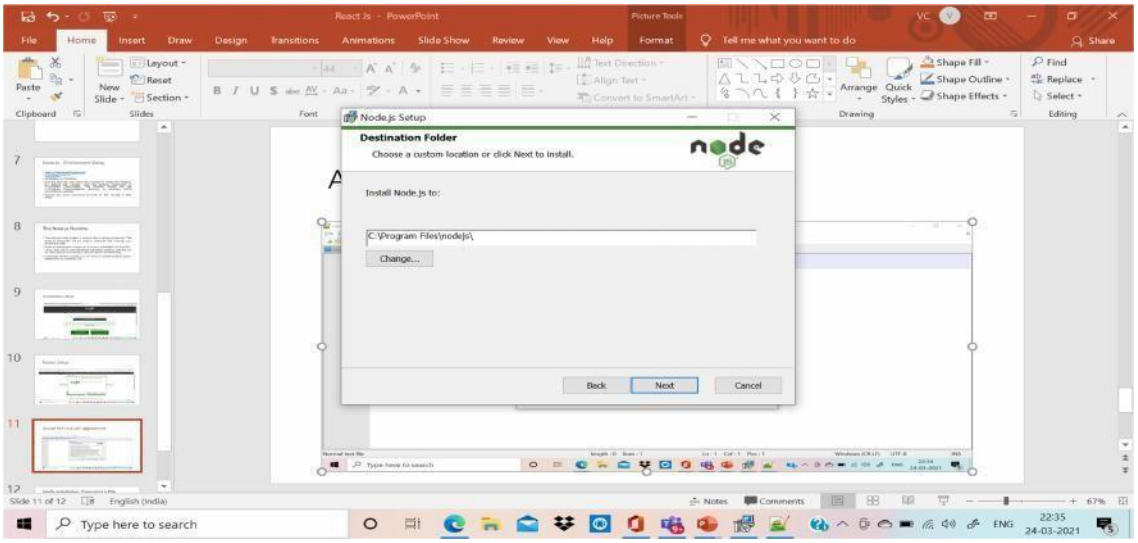
Step3:



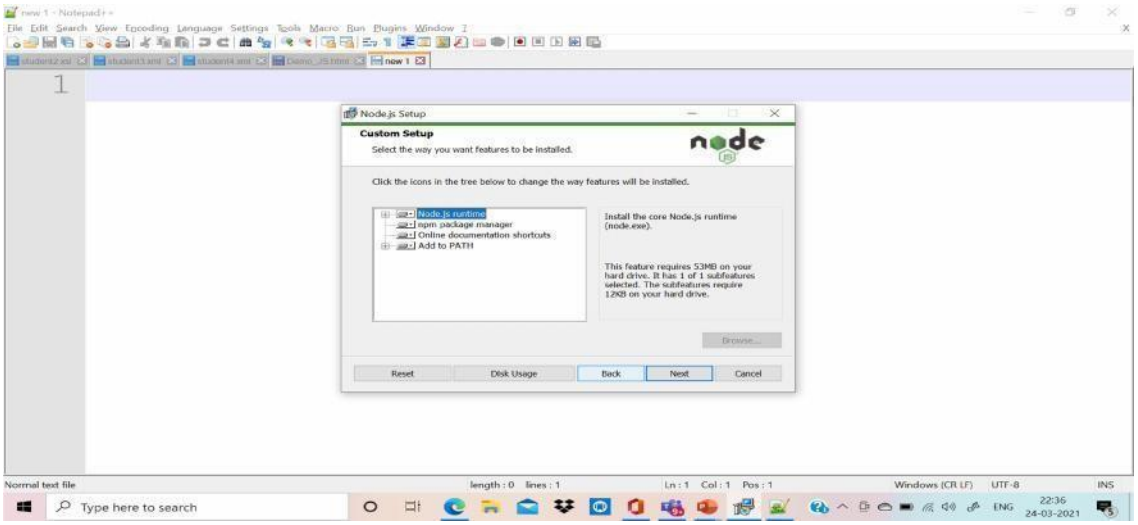
Step4: Accept the Agreement



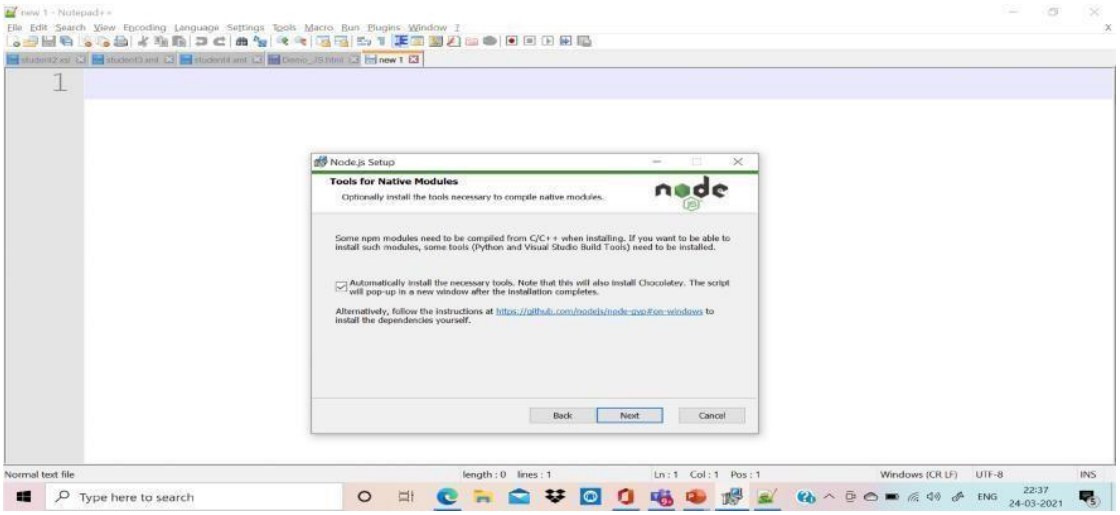
Step5: Choose Destination Folder



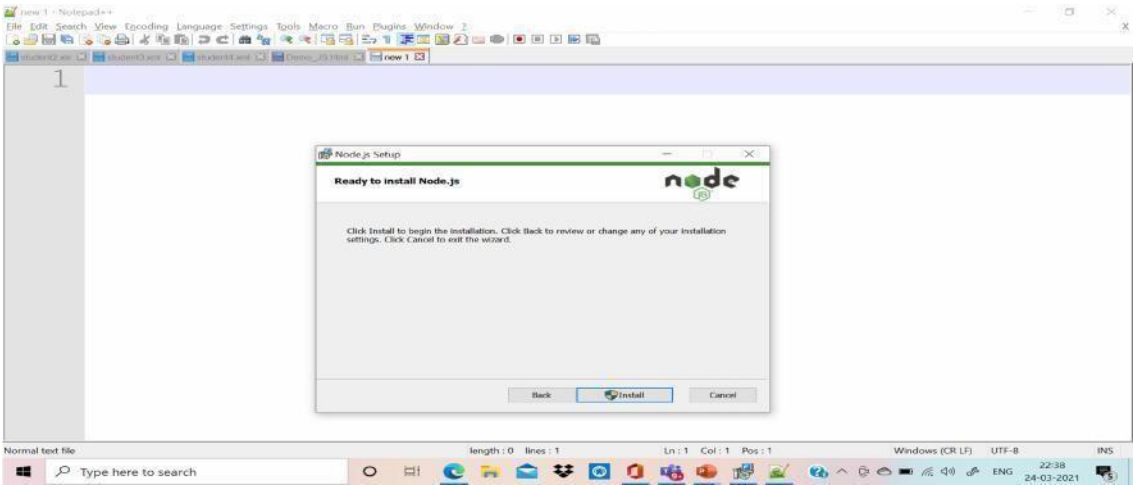
Step6: Custom Installation setup



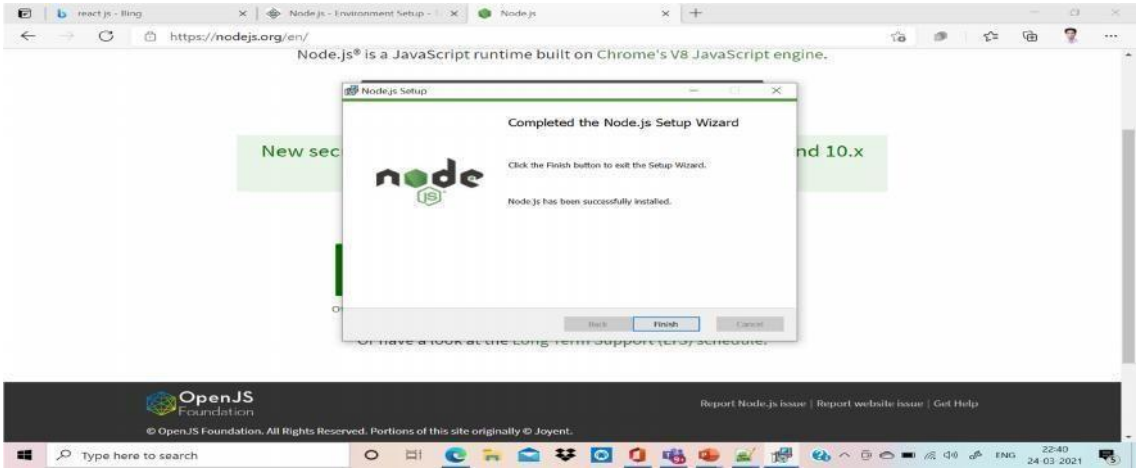
Step 7: Additional Tools Setup



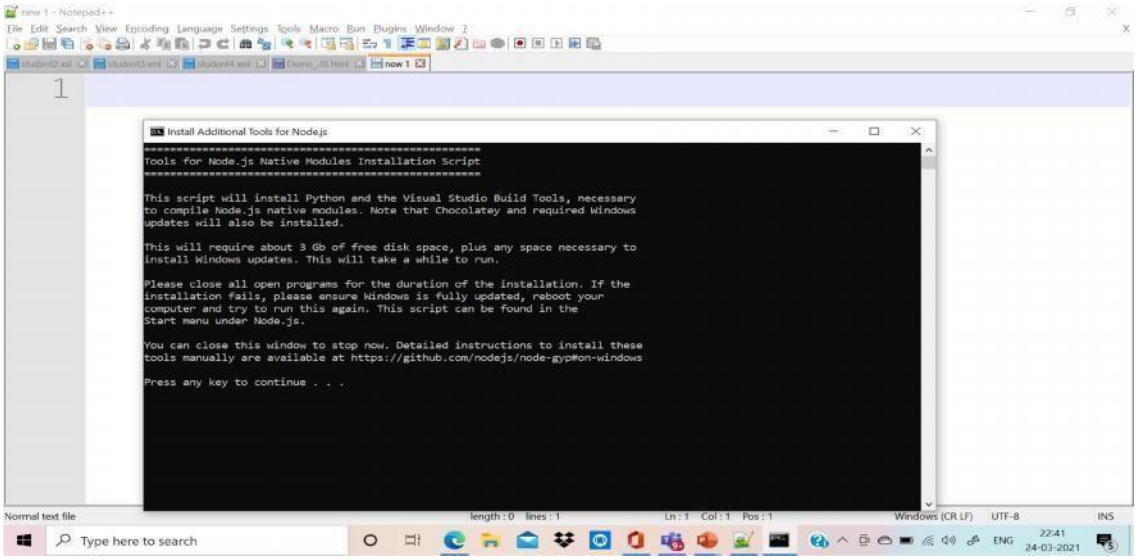
Step8: Ready to install Nodejs



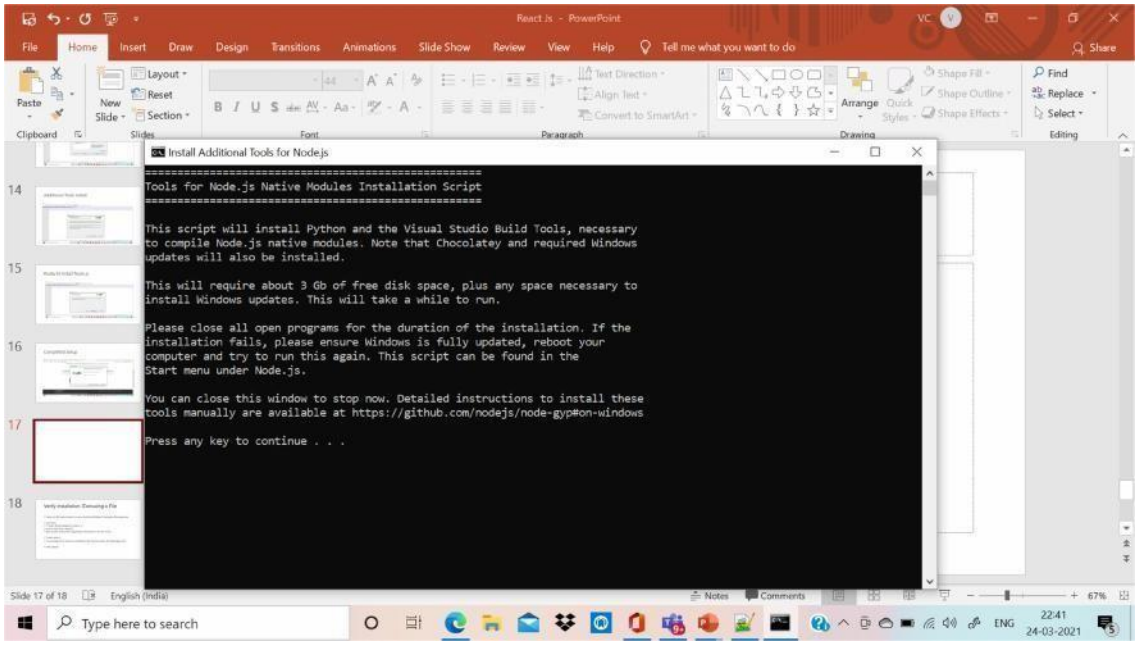
Step9: Completed Setup



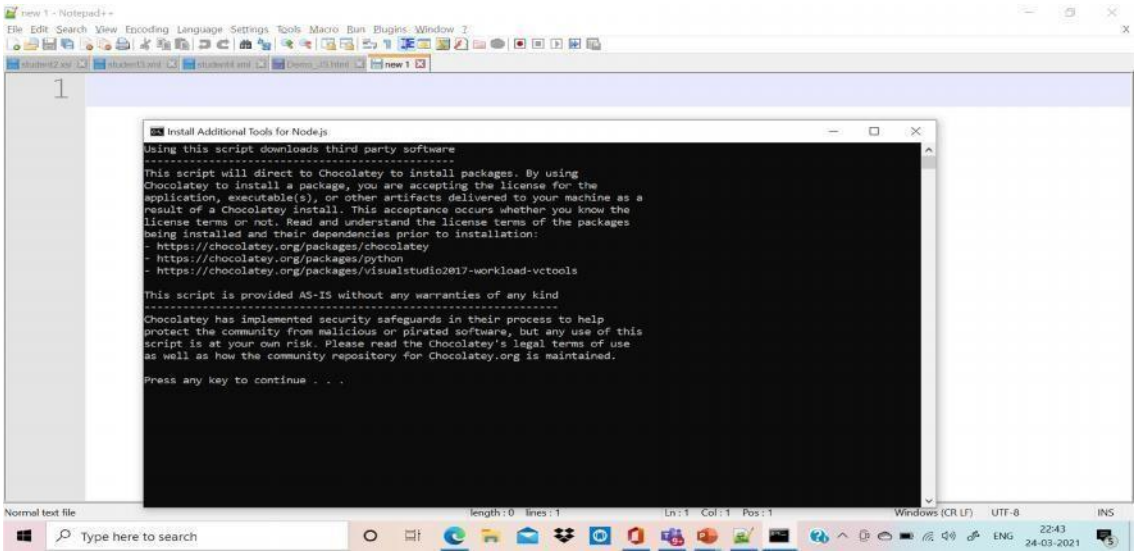
Step 10: Ready to start



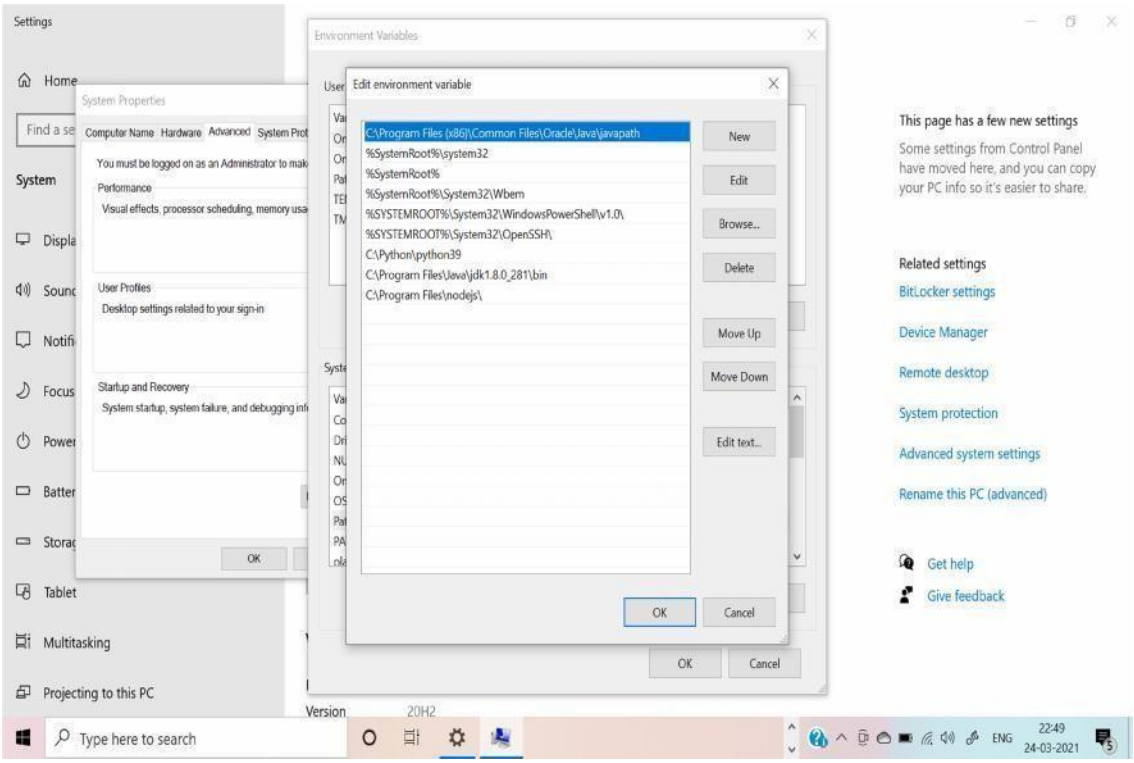
Step11: Install Native Module



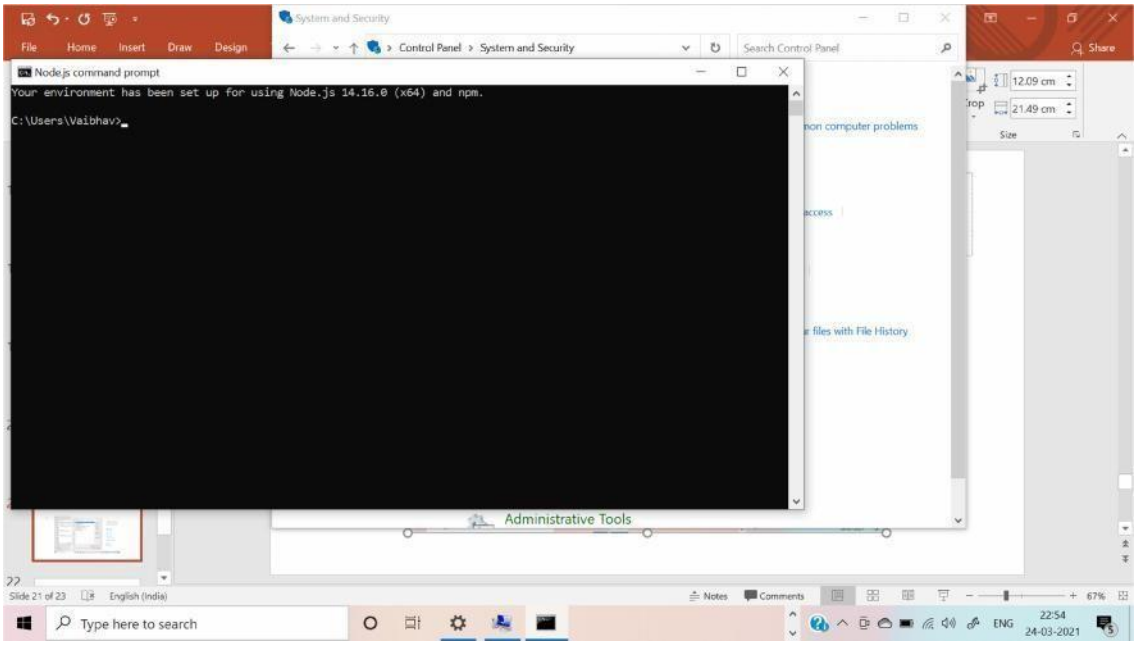
Step 12: Additional Installation



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**

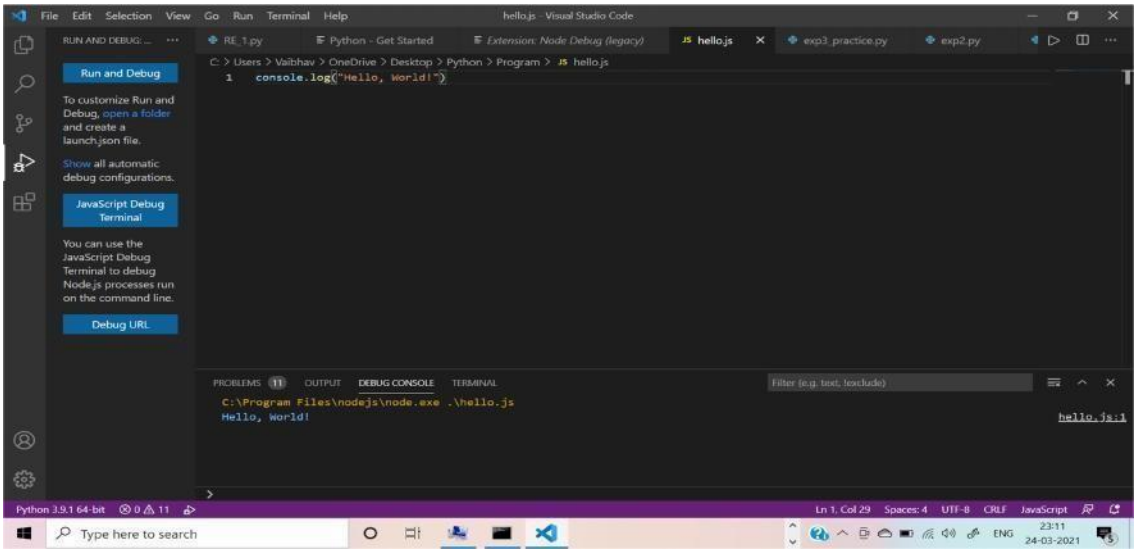
Step 16: Type the Hello World and execute

Type the command and install gyrescipt :

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 theresult −
- $ 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

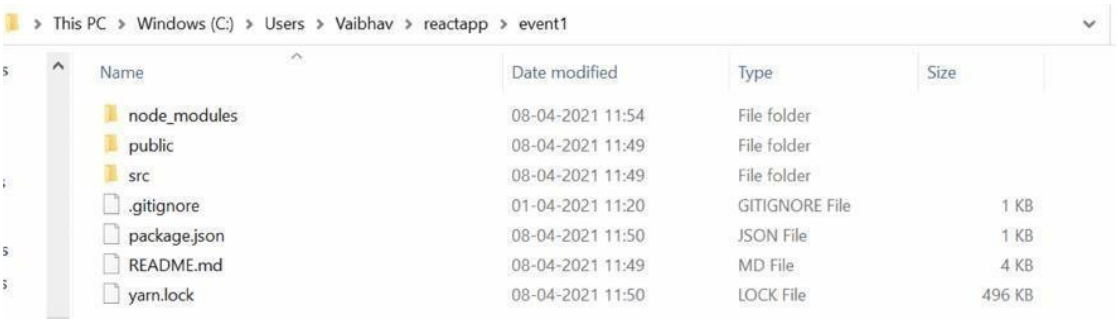
- Browse through the src folder in the generated my-app folder and remove all the files in it as shown below −
- C:\Users\Desktop>cd my-app/src
- C:\Users\Desktop\my-app\src>del *
- C:\Users\Desktop\my-app\src*, Are you sure (Y/N)? y

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

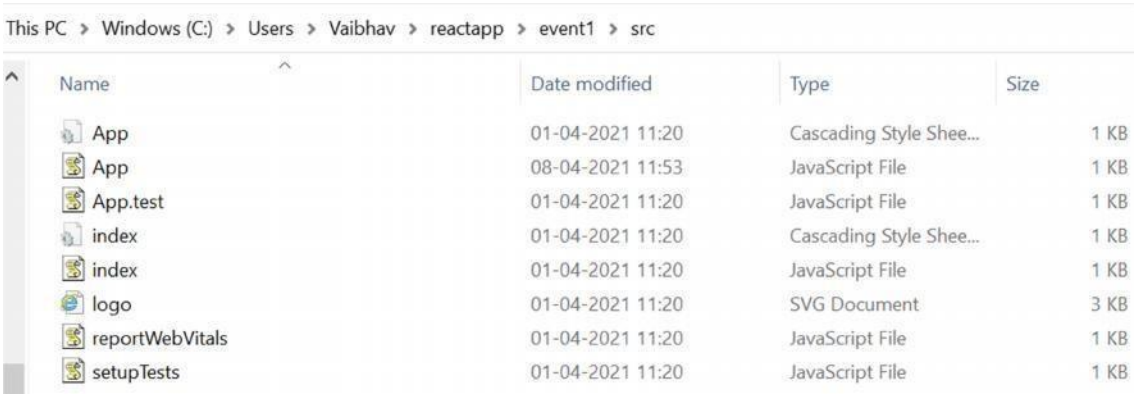
- C:\Users\Desktop\my-app\src>type nul > index.css
- C:\Users\Tutorialspoint\Desktop\my-app\src>type nul > index.js

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:



8) Open the src folder as shown below:



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:

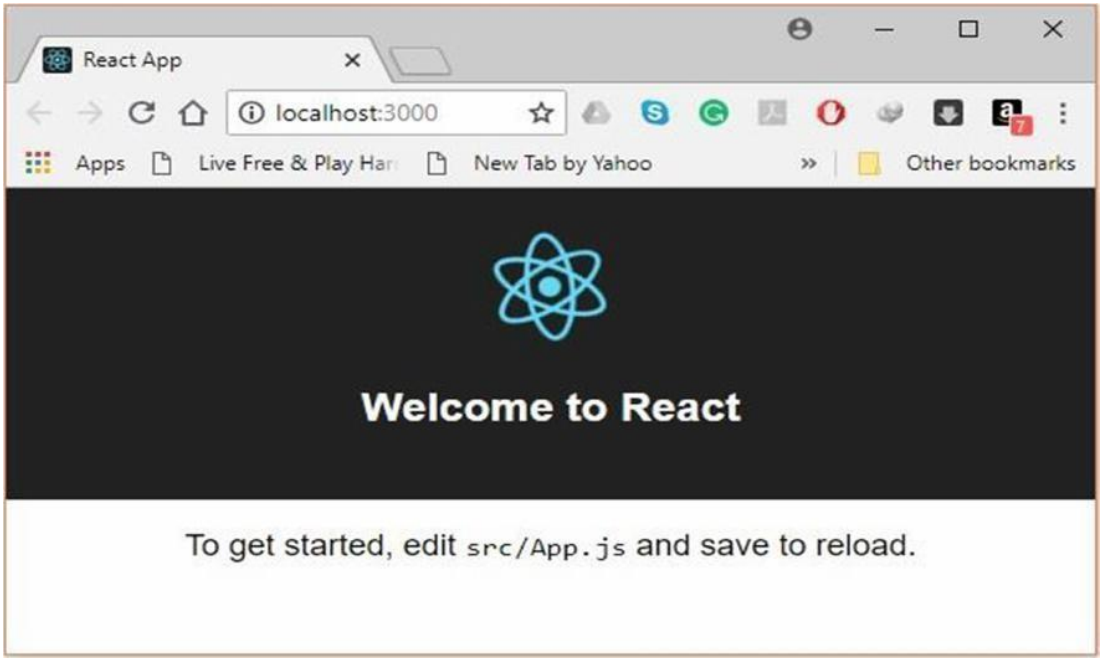
- npm start
- yarn start(if you install yarn

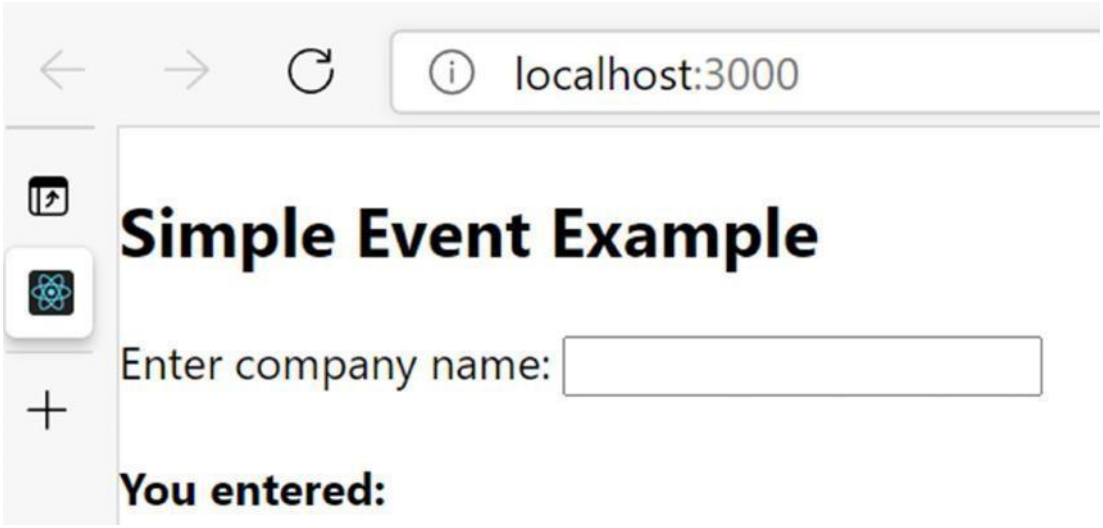
utility) To do this on, terminal of

VSCODE,

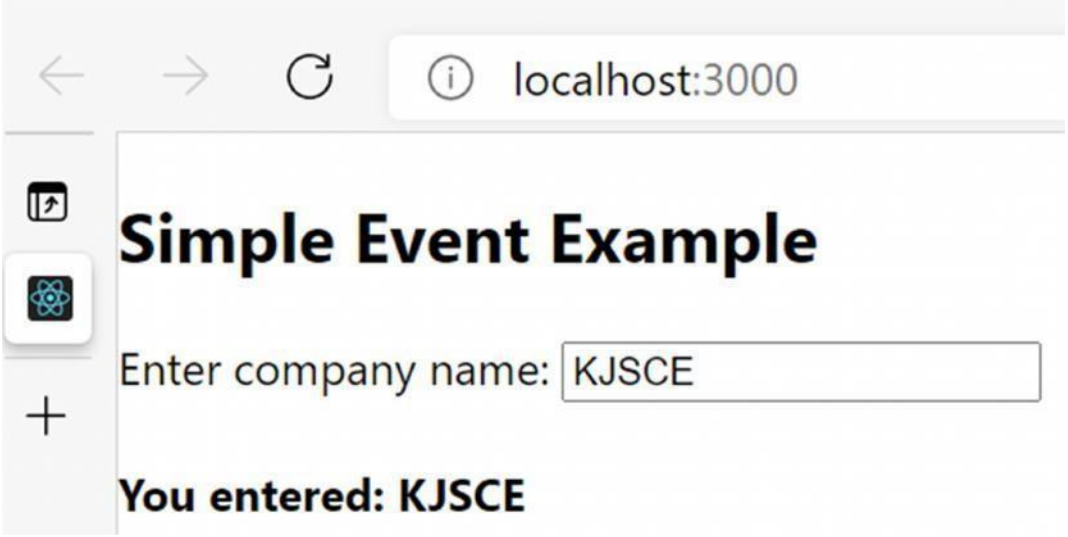
C:\Users\Vaibhav\reactapp\myevent\npm start.

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.

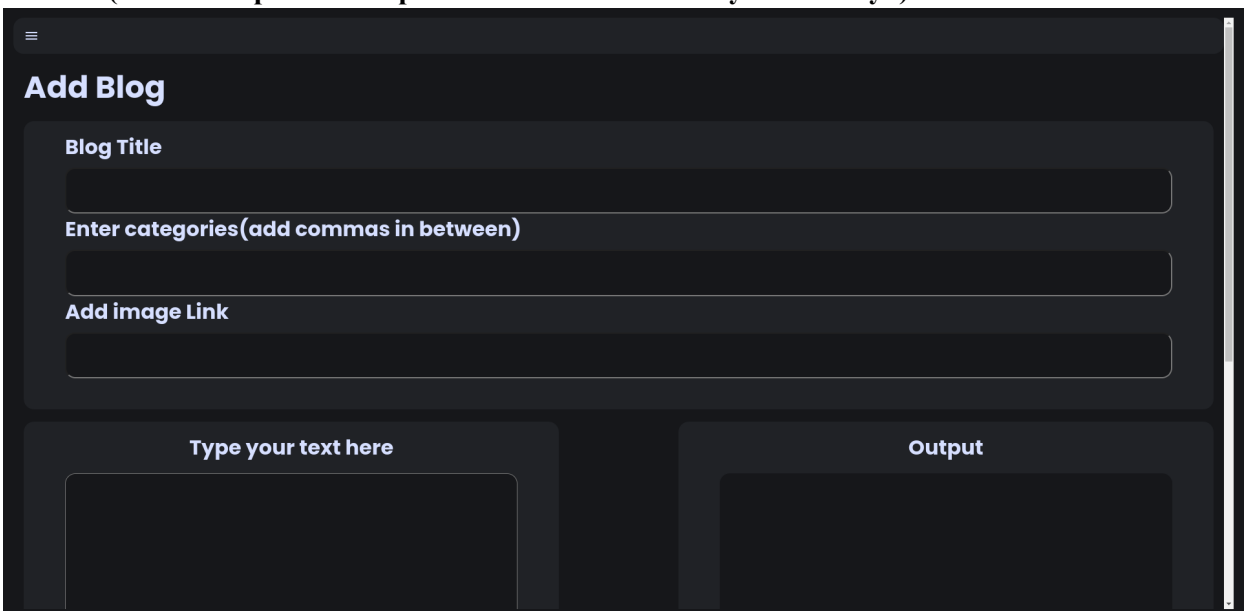
```
import React, { useState } from "react";
```

```
import ReactMarkdown from "react-markdown";
import styles from "./AddBlogScreen.module.css";
import { MenuItem, Select } from "@mui/material";
import { url } from "../../constants/baseUrl";
import { useSnackbar } from "notistack";
import axios from "axios";
import { useNavigate } from "react-router-dom";
import getCommonOptions from "../../helpers/getCommonOptions";

const AddBlogScreen = () => {
   const [markdown, setMarkdown] = useState("");
   const [title, setTitle] = useState("");
   const [categories, setCategories] = useState("");
   const [connected, setConnected] = useState(false);
   const [walletAddress, setWalletAddress] = useState("");
   const { enqueueSnackbar } = useSnackbar();
   const [image, setImage] = useState("");
   const navigate = useNavigate();
   const handleConnect = () => {
       window.ethereum.request({ method: "eth_requestAccounts"
}).then((res) => {
           setWalletAddress(res[0]);
           setConnected(true);
       });
   };
   const handleSubmit = async () => {
       const category = categories.split(", ");
       if (!markdown.length) {
           enqueueSnackbar("Enter something in content", { variant:
"error" });
           return;
       }
       if (!category.length) {
           enqueueSnackbar("No category", { variant: "error" });
           return;
       }
       if (!title.length) {
           enqueueSnackbar("Enter title", { variant: "error" });
           return;
       }
       if (!connected) {
           enqueueSnackbar("Connect Wallet", { variant: "error" });
       }
```

```
        axios.post(
            `${url}blogs/addBlog/`,
            {
                title: title,
                description: markdown.slice(0, 10),
                content: markdown,
                category: category,
                image: image,
            },
            getCommonOptions()
        );
        enqueueSnackbar("Successfully Added", { variant: "success" });
        navigate("/");
    };
    return (
        <div className={styles.container}>
            <div className={styles.pageTitle}>Add Blog</div>
            <div
                style={{
                    display: "flex",
                    justifyContent: "center",
                }}
            >
                <div className={styles.titleBox}>
                    <h1>Blog Title</h1>
                    <input
                        value={title}
                        onChange={(e) => setTitle(e.target.value)}
                        className={styles.titleInput}
                    />
                    <h1>Enter categories(add commas in between)</h1>
                    <input
                        value={categories}
                        onChange={(e) => setCategories(e.target.value)}
                        className={styles.titleInput}
                    />
                    <h1>Add image Link</h1>
                    <input
                        value={image}
                        onChange={(e) => setImage(e.target.value)}
                        className={styles.titleInput}
                    />
```

```
                </div>
            </div>
            <div className={styles.container2}>
                <div className={styles.textAreaContainer}>
                    <h1>Type your text here</h1>
                    <textarea
                        className={styles.textarea}
                        value={markdown}
                        onChange={(e) => setMarkdown(e.target.value)}
                    />
                </div>
                <div className={styles.textAreaContainer}>
                    <h1>Output</h1>
                    <div className={styles.textarea}>
                        <ReactMarkdown>{markdown}</ReactMarkdown>
                    </div>
                </div>
            </div>
            {/* <div className={} */}
            <div
                style={{
                    display: "flex",
                    justifyContent: "center",
                }}
            >
                <div className={styles.buttonGroup}>
                    <button onClick={() => handleConnect()}
className={styles.button}>
                        {connected
                            ? "Wallet Address- " + walletAddress
                            : "Click here to Connect Wallet"}
                    </button>
                    <button onClick={() => handleSubmit()}
className={styles.button}>
                        Submit
                    </button>
                </div>
            </div>
        </div>
    );
};

export default AddBlogScreen;
```

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



Display of the designed webpage along with the code.

Questions:

1. **What are the different components of ReactJS?**
   ReactJS is a JavaScript library for building user interfaces, and it is made up of several key components that work together to create a rich and interactive web application. Some of the main components of ReactJS include JSX, which allows developers to write HTML-like syntax in their JavaScript code, and the virtual DOM, which allows React to efficiently update the user interface by only re-rendering the parts of the page that have changed. Other important components of ReactJS include components, state, and props, which allow developers to build reusable and modular UI elements. Overall, ReactJS offers a powerful and flexible framework for building dynamic and responsive web applications

2. **What is Virtual DOM? How virtual DOM Works? What is the purpose of render of react DOM?**
   Virtual DOM is a programming concept introduced by ReactJS that provides a lightweight representation of the actual DOM. It's a tree-like structure that React uses to keep track of changes in the user interface. The purpose of the Virtual DOM is to reduce the amount of direct manipulation of the actual DOM, which can be slow and inefficient. Instead, React compares changes to the Virtual DOM and only updates the parts of the actual DOM that have changed. This process is called reconciliation and is what makes React so efficient and fast. The render method in React is responsible for returning a description of the user interface as a Virtual DOM tree, which React then uses to update the actual DOM. Overall, the use of the Virtual DOM in React allows for faster and more efficient rendering of web applications, improving the user experience.

Outcomes:CO 4 Implement web application using React JS, Angular JS, JSON and CBOR

Conclusion:
(Conclusion to be based on objectives and outcomes achieved)

Implement reactjs

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