Name : Kunal Rajesh Kumbhare
Prn : 120A3024
Branch : IT
3nd Year (5th semester)

Experiment No: 7

Aim: - Installation of react and create class and function component

Theory :-

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies.

- Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

- Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

- A Simple Component

React components implement a render() method that takes input data and returns what to display. This example uses an XML-like syntax called JSX. Input data that is passed into the component can be accessed by render() via this.props.

- A Stateful Component

In addition to taking input data (accessed via this.props), a component can maintain internal state data (accessed via this.state). When a component's state data changes, the rendered markup will be updated by re-invoking render().

Installation:

Create your React app

To install the full React toolchain on WSL, we recommend using create-react-app:

1. Open a terminal(Windows Command Prompt or PowerShell).
2. Create a new project folder: mkdir my-apps and enter that directory:

cd my-app.

3. Install React using create-react-app, a tool that installs all of the dependencies to build and run a full React.js application:

```
PowerShell                                Copy

npx create-react-app my-app
```

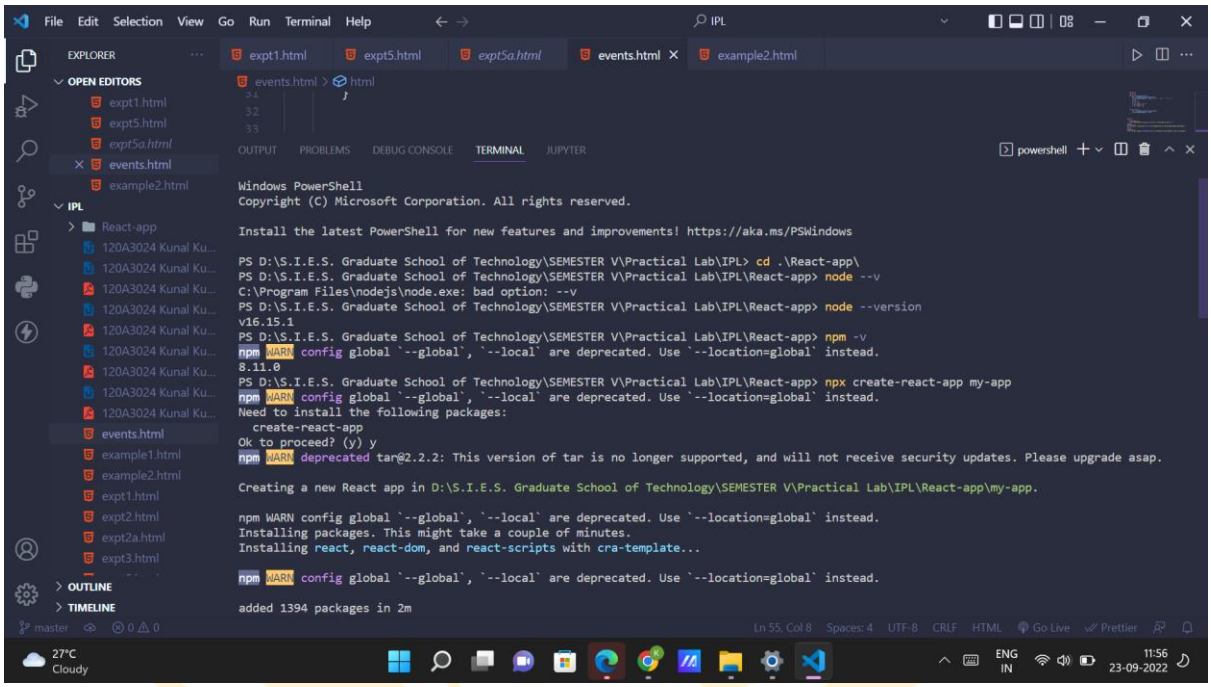
npx create-react-app my-app

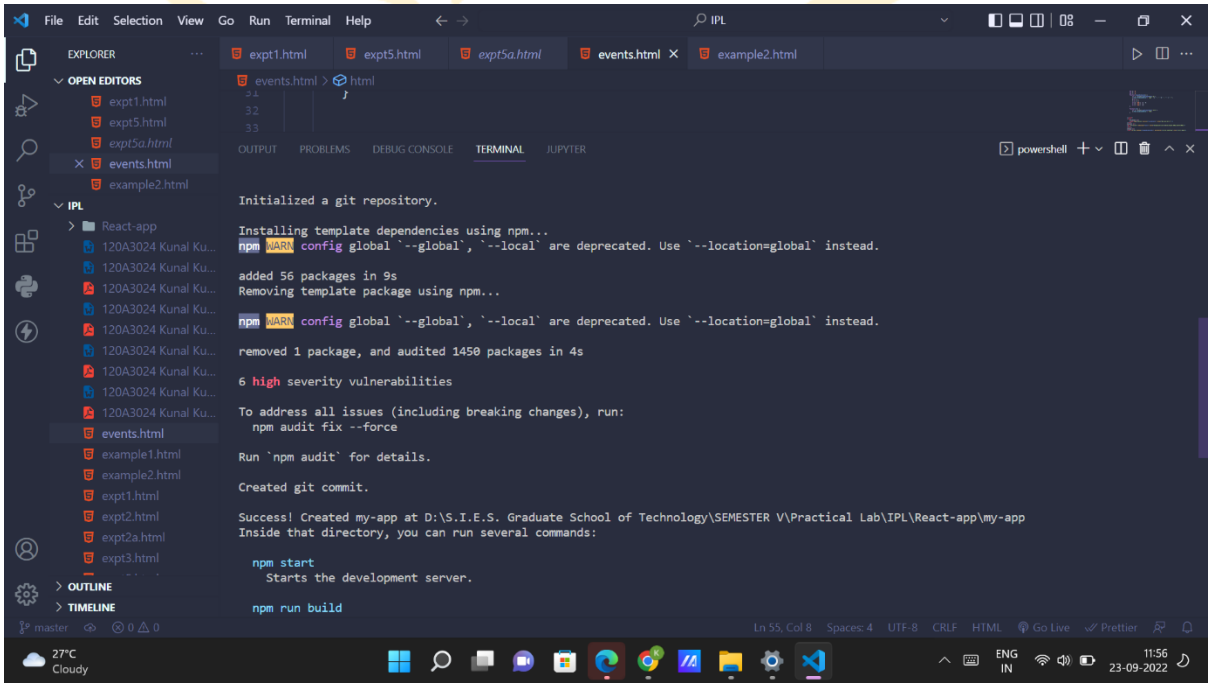
4. This will first ask for your permission to temporarily install create-react-app and it's associated packages. Once completed, change directories into your new app ("my-app" or whatever you've chosen to call it): cd my-app.
5. Start your new React app:

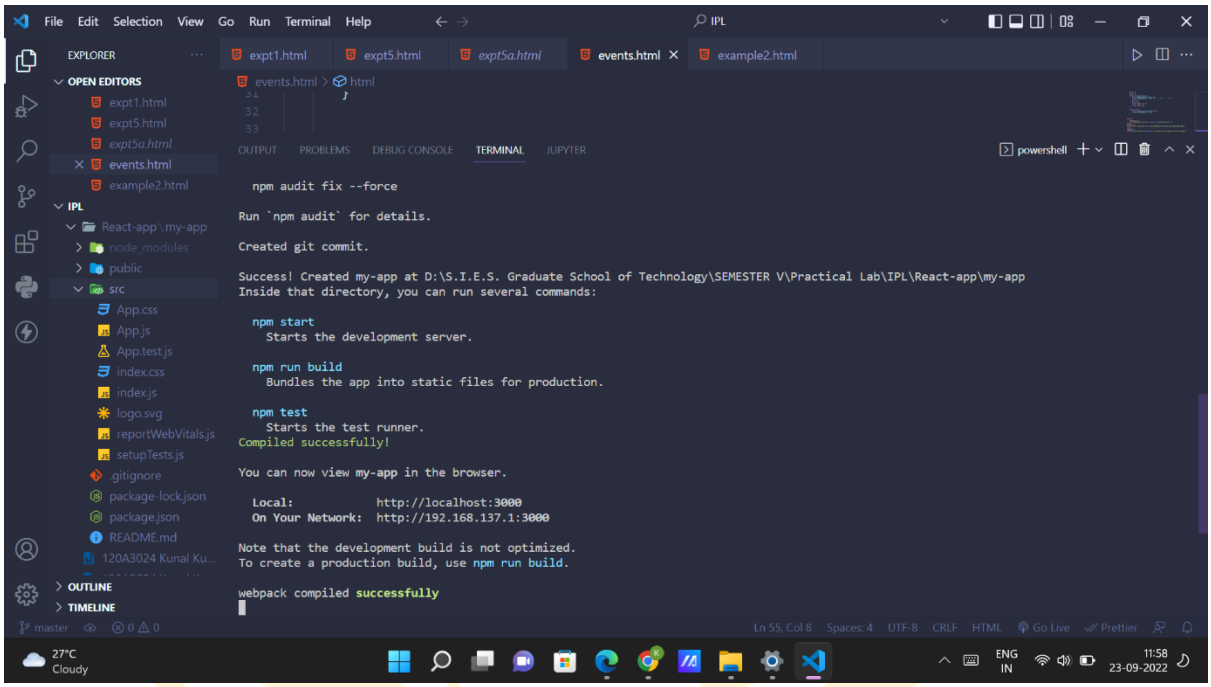
This command will start up the Node.js server and launch a new browser window displaying your app. You can use Ctrl + c to stop running the React app in your command line.

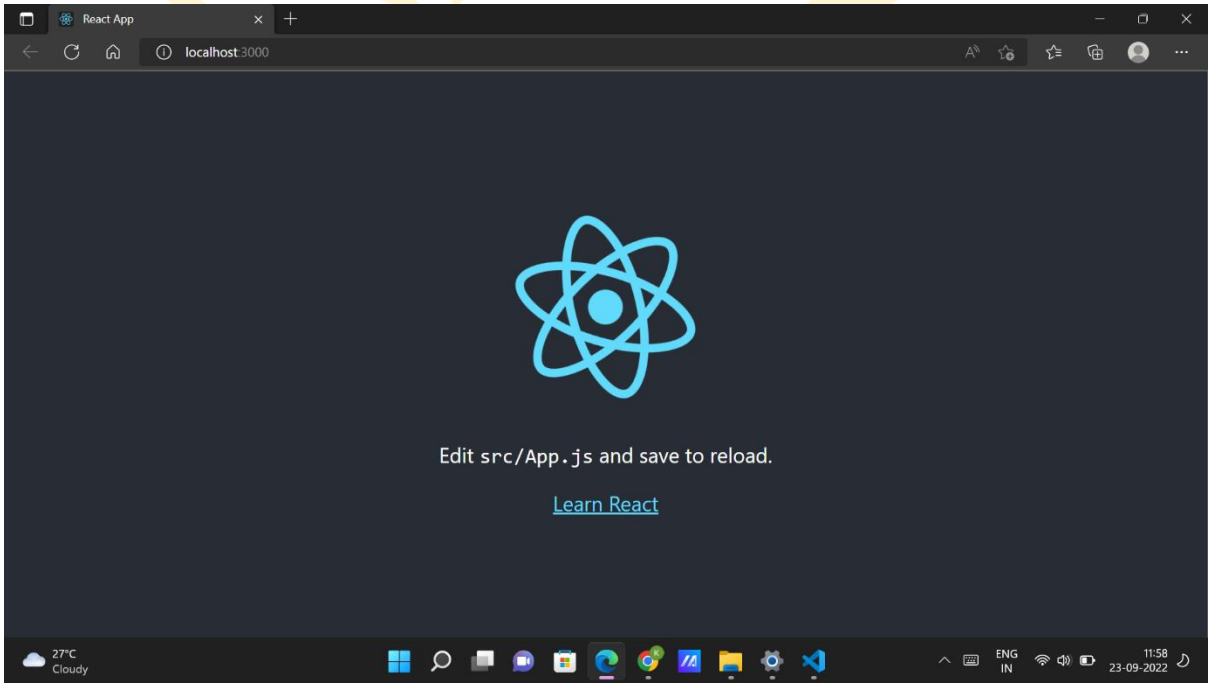
```
PowerShell                                Copy

npm start
```

Output :


File Edit Selection View Go Run Terminal Help
IPL
EXPLORER
OPEN EDITORS
expt1.html
expt5.html
expt5a.html
events.html
example2.html
IPL
React-app
OUTLINE
TIMELINE
OUTPUT PROBLEMS DEBUG CONSOLE TERMINAL JUPYTER
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL> cd .\React-app\
PS D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL\React-app> node --v
C:\Program Files\nodejs\node.exe: bad option: --v
PS D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL\React-app> node --version
v16.15.1
PS D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL\React-app> npm -v
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
8.11.0
PS D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL\React-app> npx create-react-app my-app
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
Need to install the following packages:
create-react-app
Ok to proceed? (y) y
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.
Creating a new React app in D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL\React-app\my-app.
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
added 1394 packages in 2m
27°C Cloudy
ENG IN
11:56 23-09-2022


File Edit Selection View Go Run Terminal Help
IPL
EXPLORER
OPEN EDITORS
OUTPUT PROBLEMS DEBUG CONSOLE TERMINAL JUPYTER
powershell
Initialized a git repository.
Installing template dependencies using npm...
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
added 56 packages in 9s
Removing template package using npm...
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
removed 1 package, and audited 1450 packages in 4s
6 high severity vulnerabilities
To address all issues (including breaking changes), run:
npm audit fix --force
Run `npm audit` for details.
Created git commit.
Success! Created my-app at D:\S.I.E.S. Graduate School of Technology\SEMESTER V\Practical Lab\IPL\React-app\my-app
Inside that directory, you can run several commands:
npm start
Starts the development server.
npm run build
27°C Cloudy
ENG IN
11:56 23-09-2022





Components:

Theory :

React Components

Earlier, the developers write more than thousands of lines of code for developing a single page application. These applications follow the traditional DOM structure, and making

changes in them was a very challenging task. If any mistake found, it manually searches the entire application and update accordingly. The component-based approach was introduced to overcome an issue. In this approach, the entire application is divided into a small logical group of code, which is known as components.

A Component is considered as the core building blocks of a React application. It makes the task of building UIs much easier. Each component exists in the same space, but they work independently from one another and merge all in a parent component, which will be the final UI of your application.

Every React component have their own structure, methods as well as APIs. They can be reusable as per your need. For better understanding, consider the entire UI as a tree. Here, the root is the starting component, and each of the other pieces becomes branches, which are further divided into sub-branches.

1. Functional Components
2. Class Components

Functional Components

In React, function components are a way to write components that only contain a render method and don't have their own state. They are simply JavaScript functions that may or may not receive data as parameters. We can create a function that takes props(properties) as input and returns what should be rendered.

Class Components

Class components are more complex than functional components. It requires you to extend from React. Component and create a render function which returns a React element. You can pass data from one class to other class components. You can create a class by defining a class that extends Component and has a render function.

Code:

Output :

Function Component

src > MyComponents > Myfunction.js > default

```
import React from "react";

function Myfunction () {
    return (
        <div>
        <p><h1> This is function component List</h1></p>

    <ul>
        <li>Apple</li>
        <li>Banana</li>
        <li>mango</li>

    </ul>
    </div>
    )
}

export default Myfunction;
```

Class component:

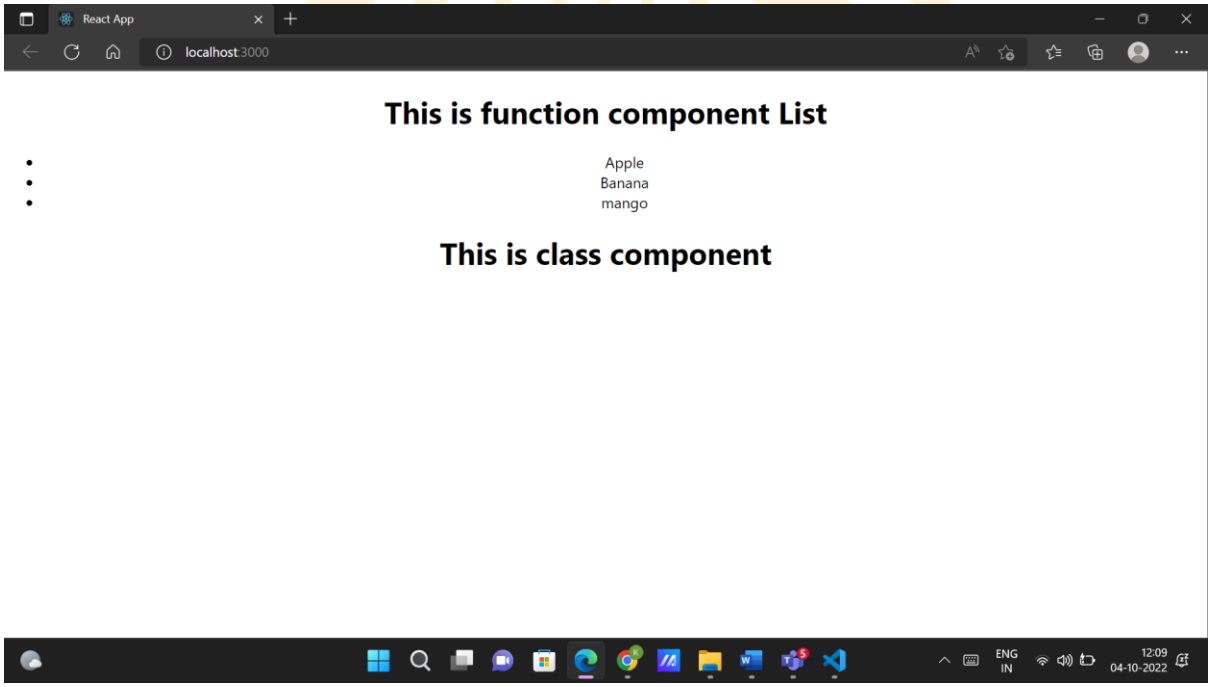
src > MyComponents > Classcomp.js > default

```
import React from "react";

class Classcomp extends React.Component {
render(){
    return <h1> This is class component</h1>

}
}

export default Classcomp;
```

App

```
src > JS App.js > ...
1   import logo from './logo.svg';
2   import './App.css';
3   import Myfunction from './MyComponents/Myfunction';
4   import Classcomp from './MyComponents/Classcomp';
5   import Hookusestate from './MyComponents/Hookusestate';
6
7   function App() {
8     return (
9       <div className="App">
10        <Myfunction/>
11        <Classcomp/>
12        <Hookusestate/>
13      </div>
14    );
15  }
16
17  export default App;
```

Output :



Conclusion :

Thus, we have successfully able to install react.js and also able to create class and function component in react js.