**ReactJS**

**What is React js?**

* React.js is an open source JavaScript library that is used for building user interfaces especially for single-page applications.
* It is used for handling the view layer for web and mobile apps. React also allows us to create reusable components.
* React was first created by Jordan Walke, a software engineer working for Facebook.
* Reactjs was developed by Facebook.
* React.JS was first used in 2011 for Facebook's Newsfeed feature.
* Initial Release to the Public (V0.3.0) was in July 2013.
* Current version of React.JS is V18.0.0/ React 18.

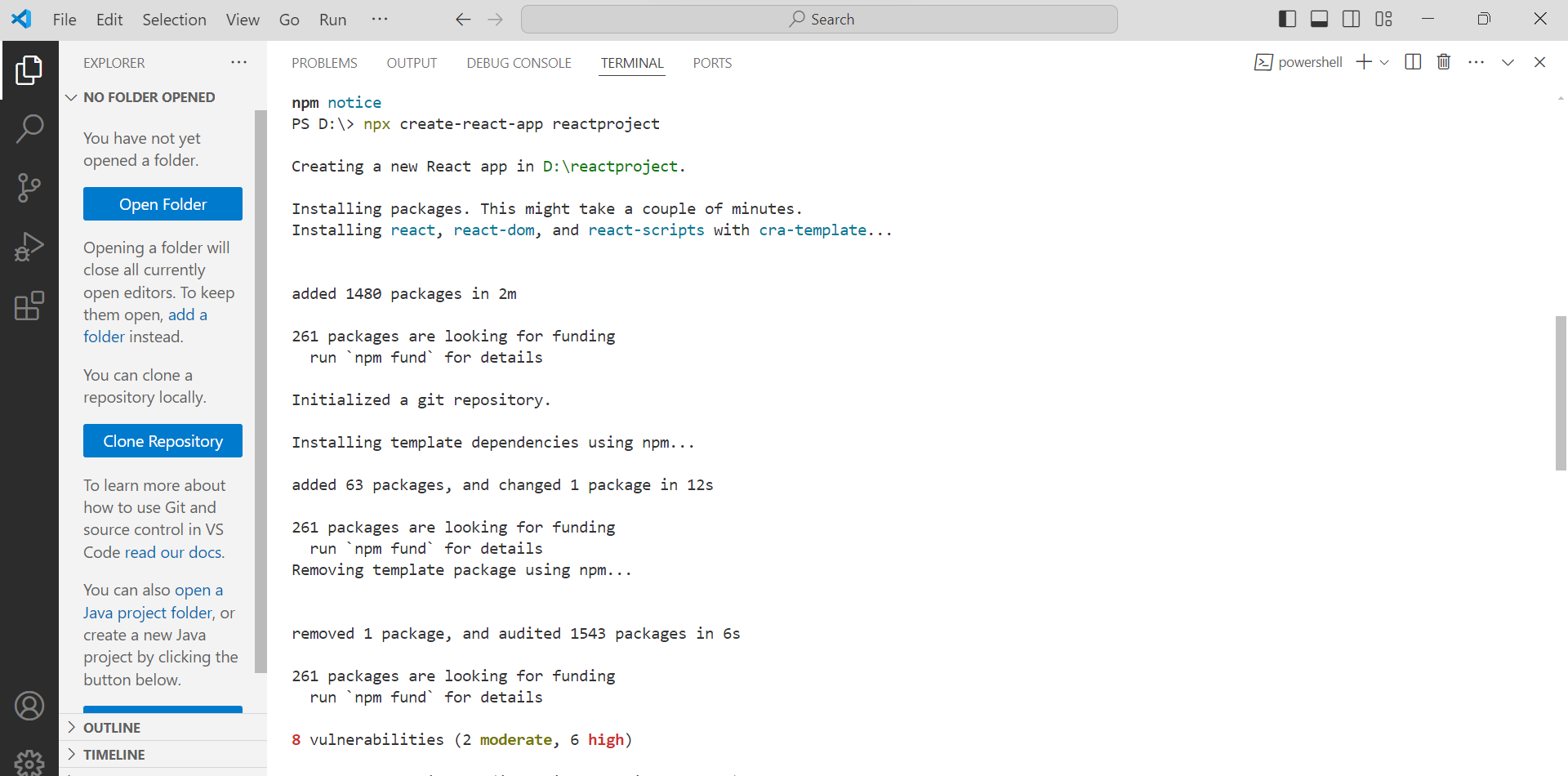
**Why we use ReactJS**

* ReactJs is component based Architecture.
* React Uses virtual DOM for fast rendering . The virtual DOM provides mechanism that abstracts manual DOM manipulations away from the developer, helping us to write more predicable code. It does so by comparing two render trees to determine exactly what has changed, only updating what is necessary on the DOM.
* React js is easy to learn.

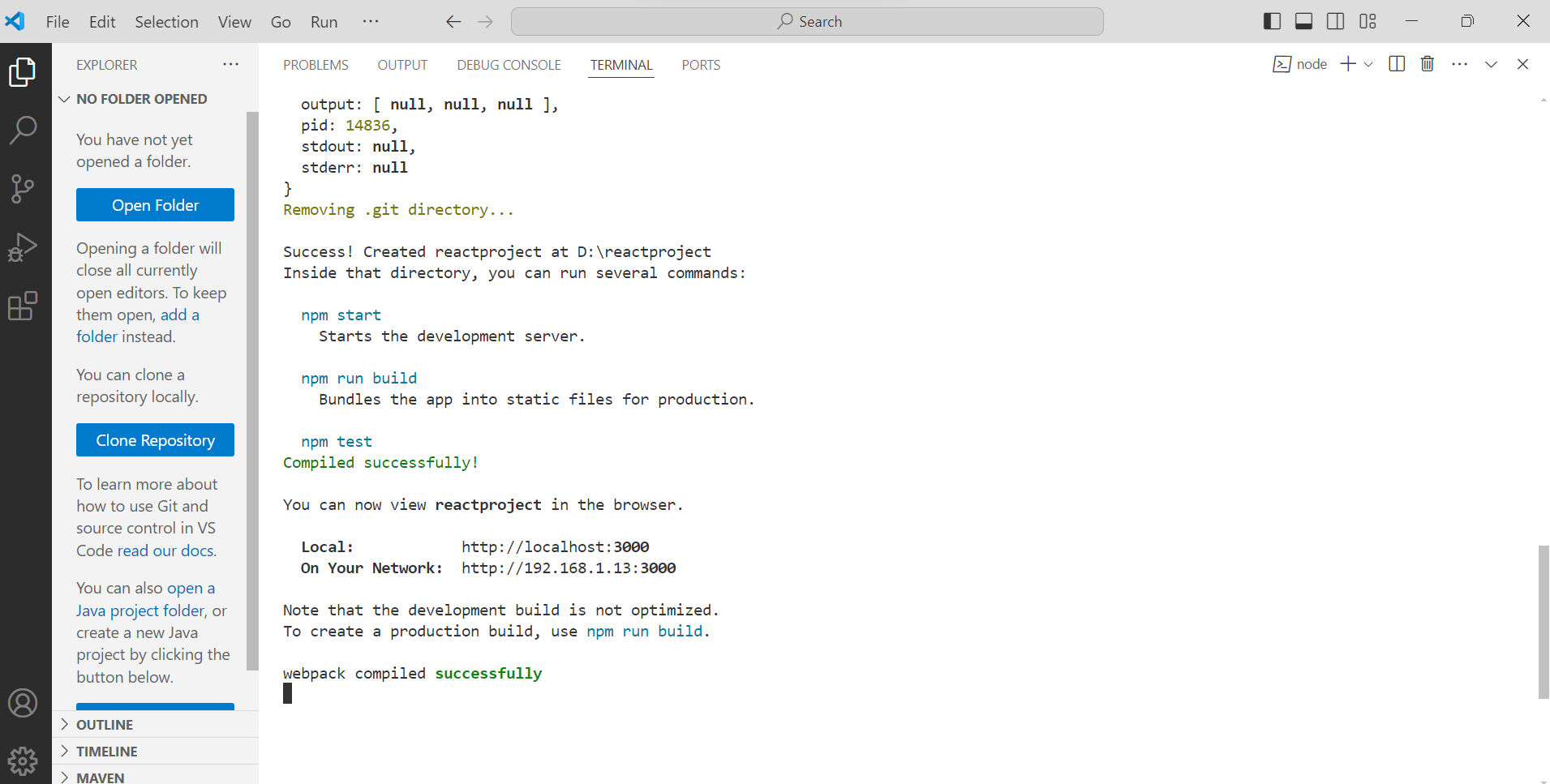
**Create React app**

1. Install node Js
2. Install Visual Studio editor .
3. To create react project 🡪 open visual studio editor 🡪 open terminal

npx create-react-app projectname

****

1. To start our program 🡪 npm start

****

**Component**

In React Component represents the part of the user interface.  
Components describes a part of the user interface.

They are re-usable and can be nested inside other components.

|  |
| --- |
| Header  Main Content  Side Nav Side Nav    Footer |

Root(App) Component

**Component Types**

1. Stateless Functional component Type
2. Stateful Class Component Type
3. Stateless Functional Component:

Properties

JavaScript Function

(props) HTML(JSX)

|  |
| --- |
| import React from "react";  function Greet(){      return (      <h1>Welcome K7IT</h1>  );  }  //export const Greet = ()=> <h1>Welcome K7IT</h1>  export default Greet; |

1. Stateful Class Component

Properties (props) HTML(JSX)

JavaScript Class

|  |
| --- |
| Class Welcome extends Component {  render(){  return <h1>Class Component<h1>  }  }  Export default Welcome; |

**Difference between Functional and Class Components**

|  |  |
| --- | --- |
| Functional Components | Class Components |
| Simple Functions  Use Function components as much as possible  Absence of ‘this’ keyword  Mainly responsible for the UI  Statless/Dumb/Presentational | More Feature rich  Maintain their own private data- state  Complex UI logic  Stateful/Smart/Container |

**JSX**

JSX (Java Script XML)- Extension to the JavaScript language syntax.

Write XML – like code for elements and components

JSX tags have the tag name, attributes, and children.

JSX is not necessary to write react applications.

JSX makes your react code simpler and elegant.

**React with JSX**

|  |
| --- |
| function Greet(){      return (      <h1>Welcome K7IT</h1>  );  }  //export const Greet = ()=> <h1>Welcome K7IT</h1>  export default Greet; |

**React without JSX**

|  |
| --- |
| function Greet(){      return React.createElement(‘div’, null, ‘Welcome K7IT’); or  return React.createElement(  ‘div’,  {id:’hello’, className:’welcome’},  React.createElement(‘h1’,null,’Welcome K7IT’));  }  //export const Greet = ()=> <h1>Welcome K7IT</h1>  export default Greet; |

**PROPS**

Props means properties added in the functional components.

|  |
| --- |
| function App() {    return (      <div className="App">          <Welcome name="K7IT"></Welcome>          <ClassComponent></ClassComponent>          <Menu/>        </div>    );  } |

|  |
| --- |
| import React, { Component } from "react";  const Welcome =(props)=>{          console.log(props)          return <h1> Welcome to {props.name}</h1>;    }  export default Welcome |

**State**

**State** means nothing but an object that is privately maintain inside a class component

State can influence what is render in the browser.

State can be changed within the component.

|  |
| --- |
| import { Component } from "react";  class ClassComponent extends Component{      constructor(){          super()          this.state={              message:'Welcome Class Component'          }      }      changeMessage(){          this.setState({              message:"Thank you for subscribing"          })      }      render(){          return(          <div id="component">          <h1>{this.state.message}</h1>          <button onClick={() => this.changeMessage()}>Subscribe</button>          </div>          );      }  }  export default ClassComponent; |

**React Snippet**

Install the ES7 React/Redux/GraphQL from Visual Studio Extension.

rce- to create class component

rconst – to create constructor inside the component.

**setState method**

|  |
| --- |
| import React, { Component } from 'react'  class Counter extends Component {      constructor(props) {        super(props)          this.state = {           count:0        }      }      increment(){          this.setState(              {              count:this.state.count + 1              },              ()=>{                  console.log('callback value', this.state.count)              }          )          console.log( this.state.count)      }    render() {      return (        <div>         <div> Count -{this.state.count}</div>          <button onClick={()=> this.increment()}>Increment</button>        </div>      )    }  }  export default Counter |

|  |
| --- |
| import React, { Component } from 'react'  class Counter extends Component {      constructor(props) {        super(props)          this.state = {           count:0        }      }      increment(){            // Here passing function as argument, instead of in and object, within the body changing prevoius state          // here instead of using current state we are using previous state          this.setState(              prevState =>({                  count:prevState.count+1              })          )          console.log( this.state.count)      }      incrementFive(){          this.increment()          this.increment()          this.increment()          this.increment()          this.increment()      }    render() {      return (        <div>         <div> Count -{this.state.count}</div>          <button onClick={()=> this.incrementFive()}>Increment</button>        </div>      )    }  }  export default Counter |

**Summary of SetState**Always make use of setState, never modify the state directly.

Code has to be executed after the state has been updated, place that code in the callback function which is the second argument to the setState Method.

We have to update the state based on the previous state value, pass in a function as an argument instead of regular object.

**Event Handling**

|  |
| --- |
| import React from "react";  function ClickHandler(){      function clickHandle(){         console.log("Button clicked")      }      return (      <div>      <button onClick={clickHandle}>Click me</button>      </div>      );  }  export default ClickHandler; |