# React-JS

## Ans(1):-

React JS is Java-script libray used to build user interfaces for websites and web application.

→Using react JS create a fast and interactive UI like button, form etc.

## React Different from Other :-

→React JS focused in UI .Other framework focused in Full-featured.

→React is easy to fast and other framework can be complex.

→React JS performance is very fast with Virtual DOM & other is slow some use real DOM.

→React js can be used with other libraries. other framework More opinionated, requires full setup.

## Ans(2):-

## Core principles of React:-

## 1.Virtual DOM:-

React creates a virtual copy of the real DOM.

## 2.Component-Based Architecture:-

A button, a form, or even a full web page can be a component.
Components make the code reusable and easy to maintain.

## 3. One-Way Data Flow:-

React flows data from parent to child components.thats why one way data flow.

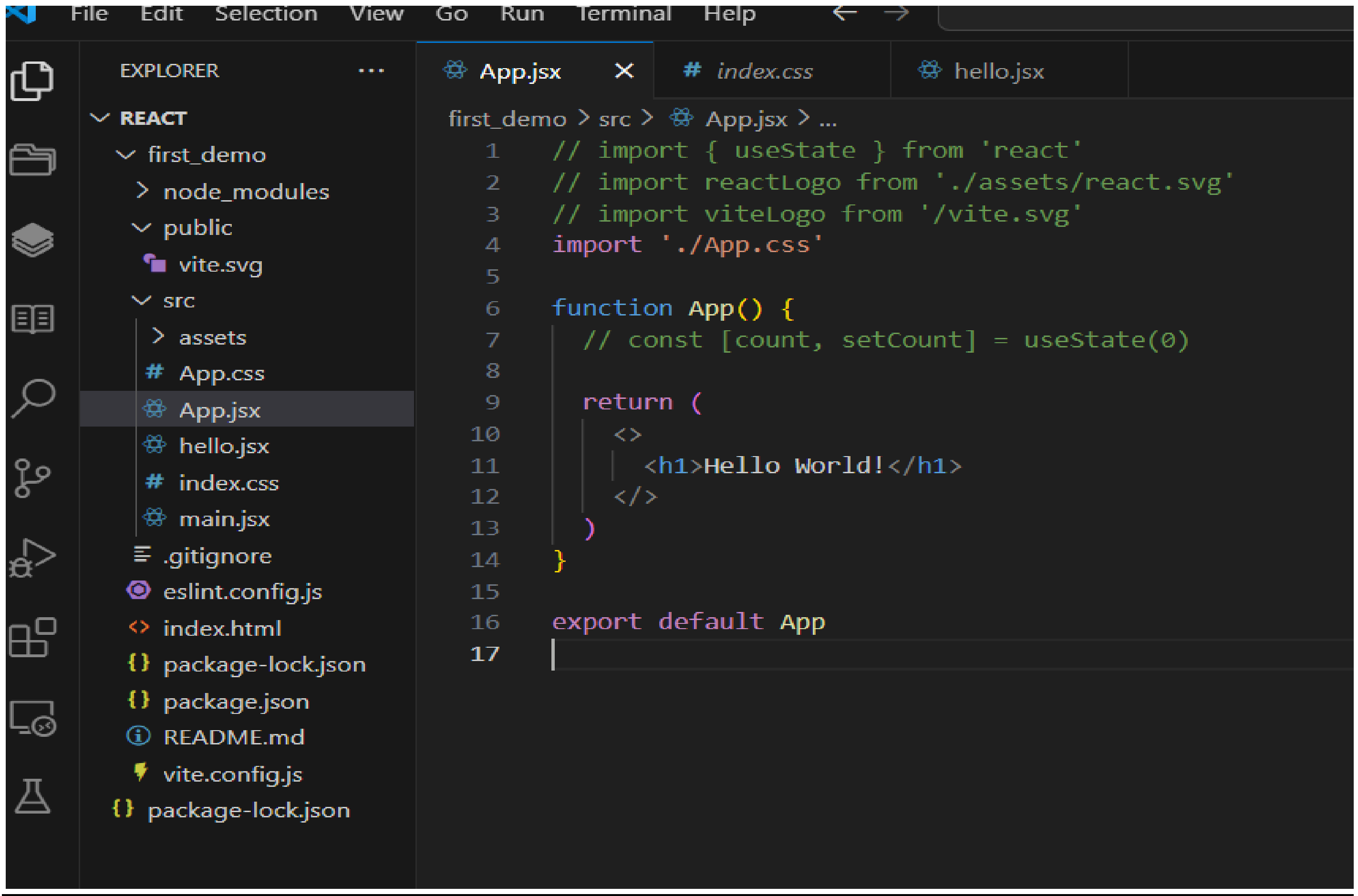
## 4.JSX:-

Jsx makes writing UI.

## 5.State & Props:-

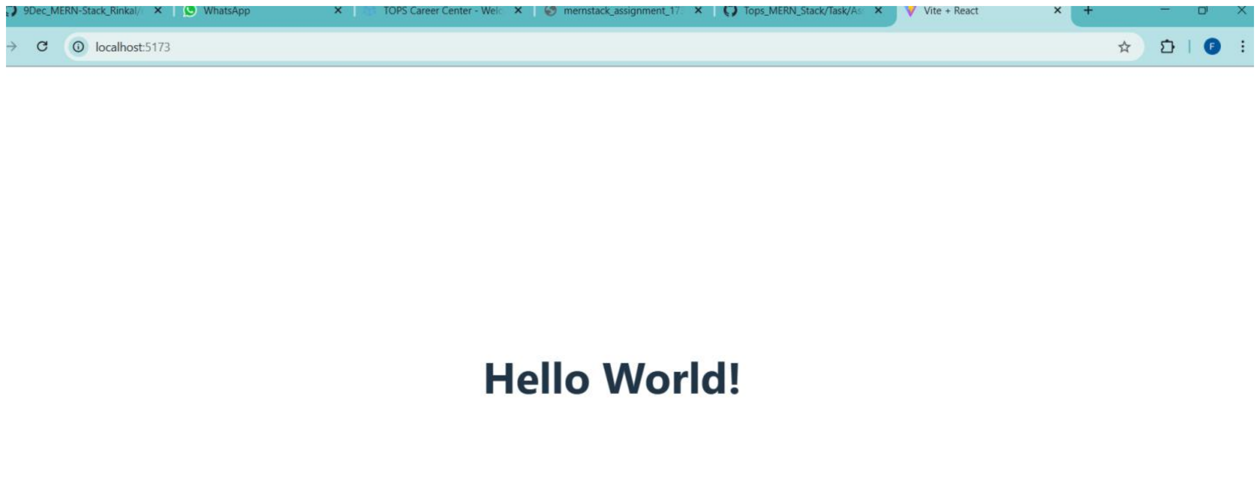
Stores dynamic data inside a component and props used to pass data from one component to another.

## LAB EXERSICE:-



```
// import { useState } from 'react'
// import reactLogo from './assets/react.svg'
// import viteLogo from '/vite.svg'
import './App.css'

function App() {
  // const [count, setCount] = useState(0)

  return (
    <>
      <h1>Hello World!</h1>
    </>
  )
}

export default App
```

**OUTPUT:-**



Hello World!

---

**JSX:-**

**Ans(1):-**

**What is JSX in React.js? Why is it used?:-**

→jsx is special syntax to in react that allows you to write html code inside javascript.

Why:-

→ Makes UI Code Easier to Write ,You can write HTML-like syntax directly in JavaScript.

→ No need to use complex document. createElement() functions.

→ You can use variables, functions, and expressions inside JSX. With javascript

## Ans(2):-

## How is JSX diffrenet from regular Javascript:-

→JSX is looks like HTML inside Java-script .regular Java-script user pure JS.

→JSX used in react for UL building. Regular JS is used for logic and DOM.

→JSX is faster .regular js is slower manipulating DOM directly.

## Ans(3):-

## Curly braces {}in JSX expressions:-

Curly braces {} are very important because they allow us to insert dynamic values inside JSX.

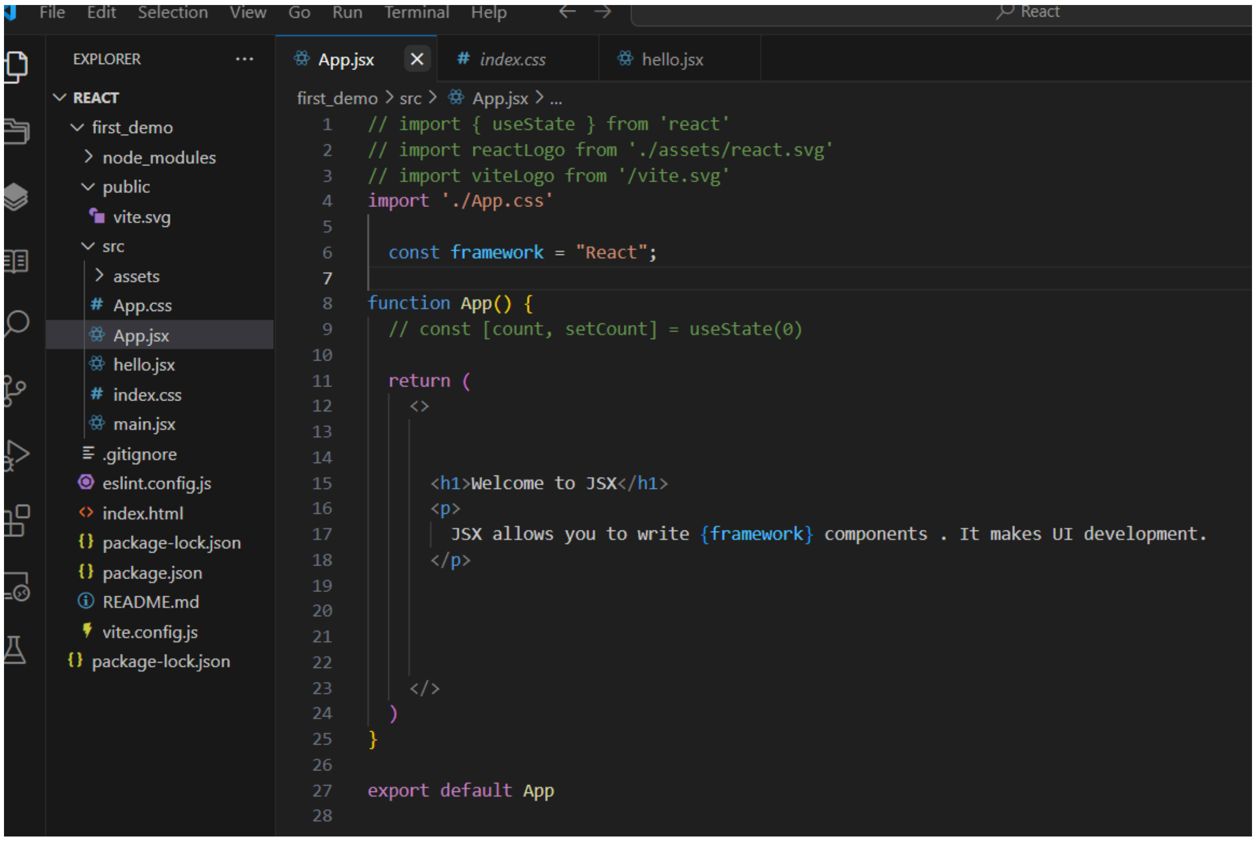
## Example:-

```
const name = "Foram";

<h1>Hello, {name}!</h1>
```

## Outputs:

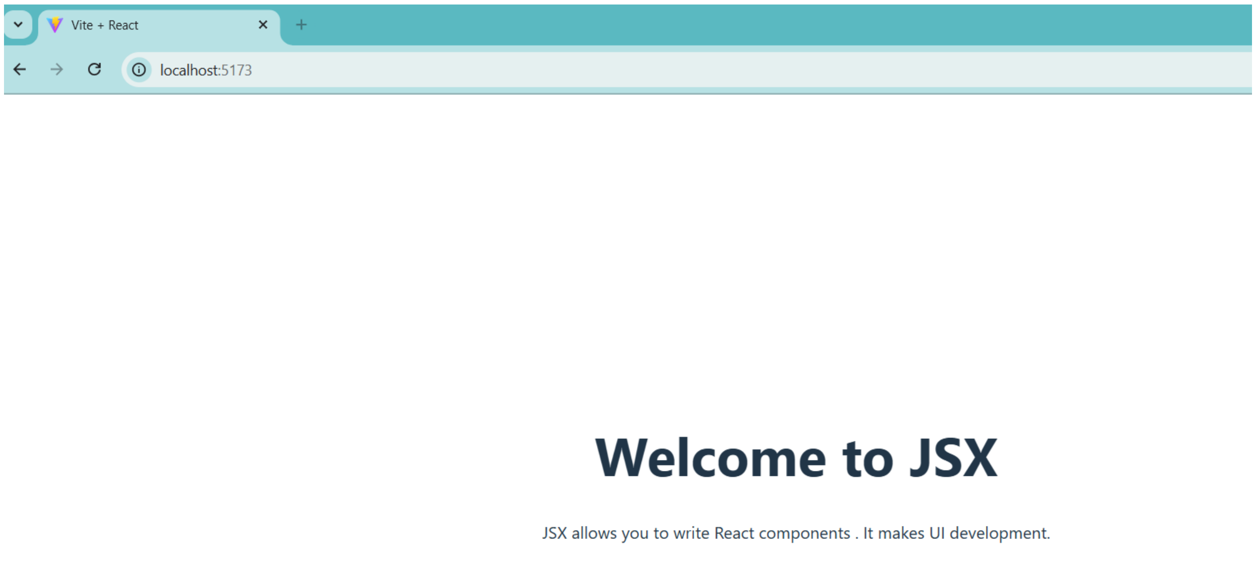
```
Hello, Foram!
```

## LAB EXERCISE:-



```
// import { useState } from 'react'
// import reactLogo from './assets/react.svg'
// import viteLogo from '/vite.svg'
import './App.css'

  const framework = "React";

function App() {
  // const [count, setCount] = useState(0)

  return (
    <>


      <h1>Welcome to JSX</h1>
      <p>
        JSX allows you to write {framework} components . It makes UI development.
      </p>





    </>
  )
}

export default App

```

## OUTPUT:-



Welcome to JSX

JSX allows you to write React components . It makes UI development.

## Components:-

### Ans (1):-

### What are React Component:-

Component a bulding a block of react application. Each component is like a UI piece that can have its own logic and behavior.

### Difference between functionalcomponents and class components:-

| **Function Components** | **Class Components** |
|---|---|
| Function components is javascript function that return UI. | Class components that extends React.Component |
| State management uses hooks like useState. | State management uses this.state. |
| Life-cycle method useEffect. | Life-cycle method like componentDidmount. |
| Function component performance is Faster. | Class component performance is more complex. |

### Ans(2):-

### Pass data to a component using props:-

React, props allow us to pass data from a parent component to a child component.

**Example:-**

Parent components:-

```
function App() {
  return <UserInfo name="Foram" age={22} />;
}
```

Child Components:-

```
function UserInfo(props) {
  return (
    <div>
      <h1>Name: {props.name}</h1>
      <h2>Age: {props.age}</h2>
    </div>
  );}
```

## Output:-

```
Name: Foram
Age : 22
```

## Ans(3):-

## Role of render():-

→Class component react it tells react what to display on the screen.

→must be include in every class components.

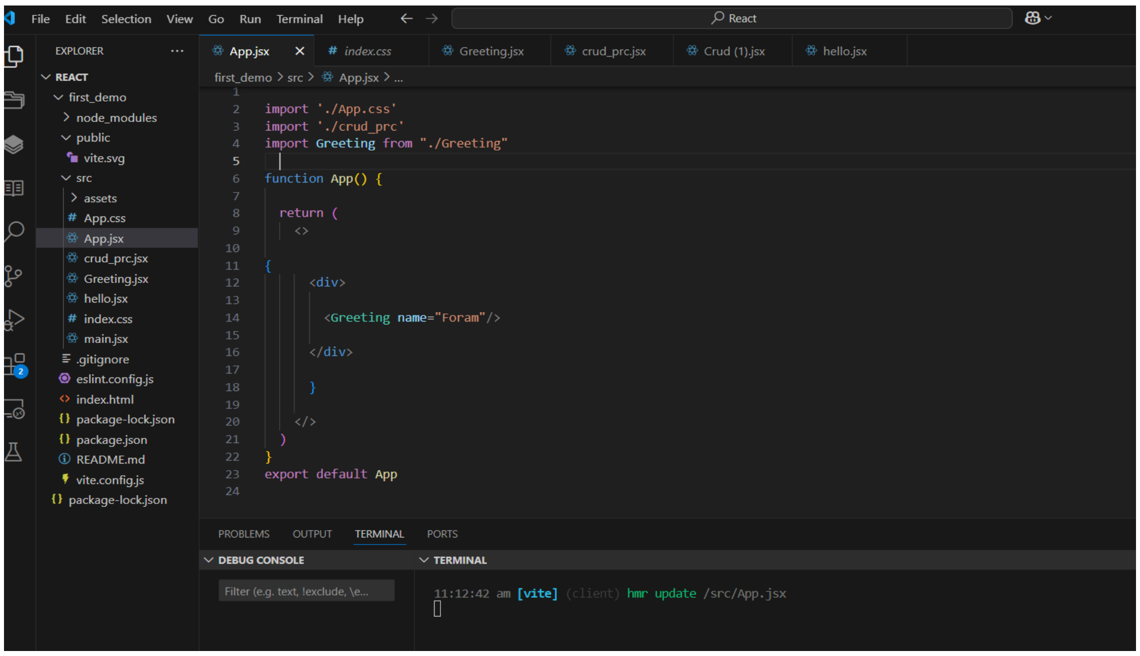
→Retrun jsx. runs automatically .

Example:-

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

class Print extends Component {

  render() {

    return <h1>Hello, World!</h1>;

  }

}

export default Print;
```

## LAB EXERCISE:-

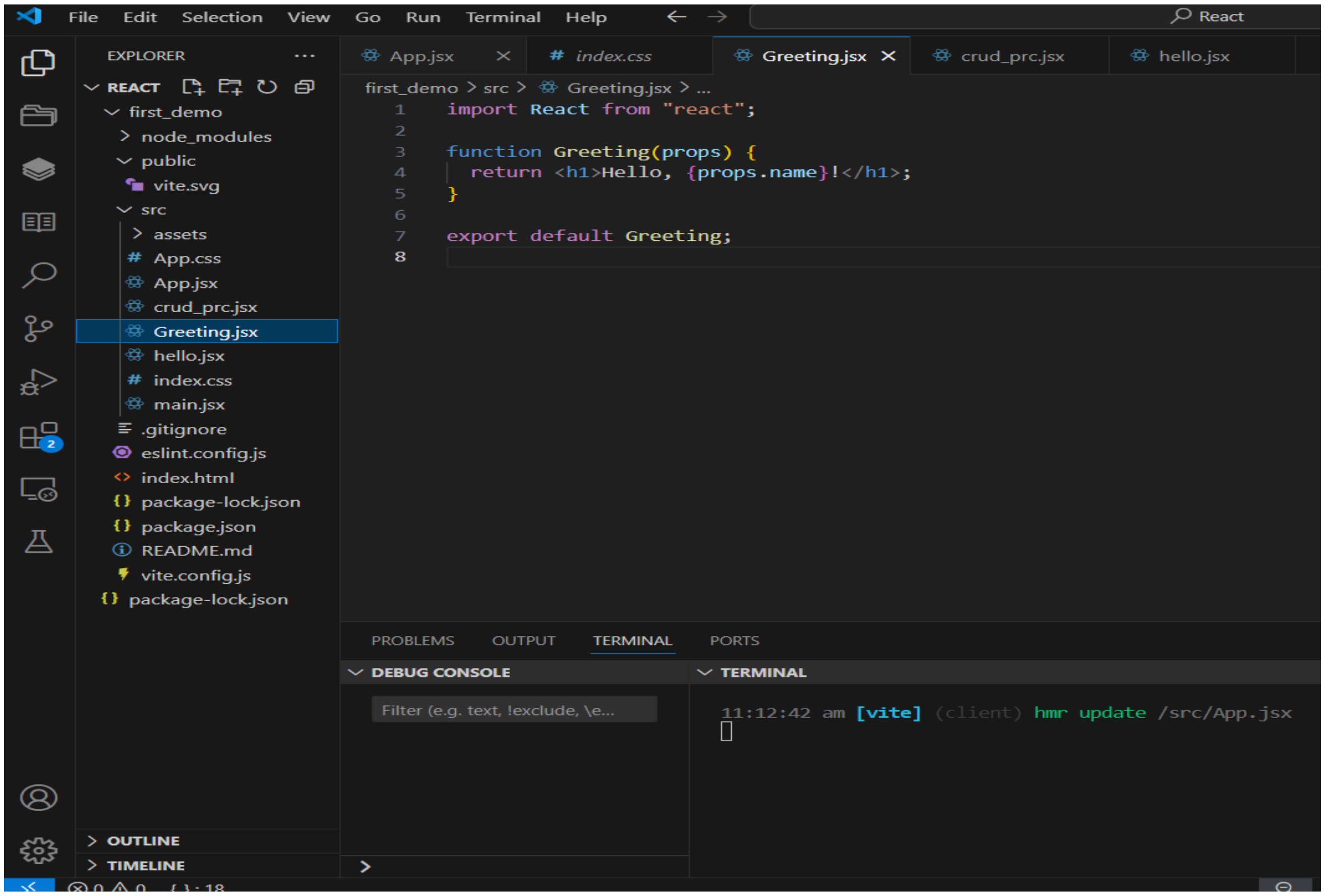
App.jsx:



```

import './App.css'
import './crud_prc'
import Greeting from "./Greeting"

function App() {

  return (
    <>

  {

      <div>

        <Greeting name="Foram"/>

      </div>

      }

    </>
  )
}
export default App

```

```
11:12:42 am [vite] (client) hmr update /src/App.jsx
```

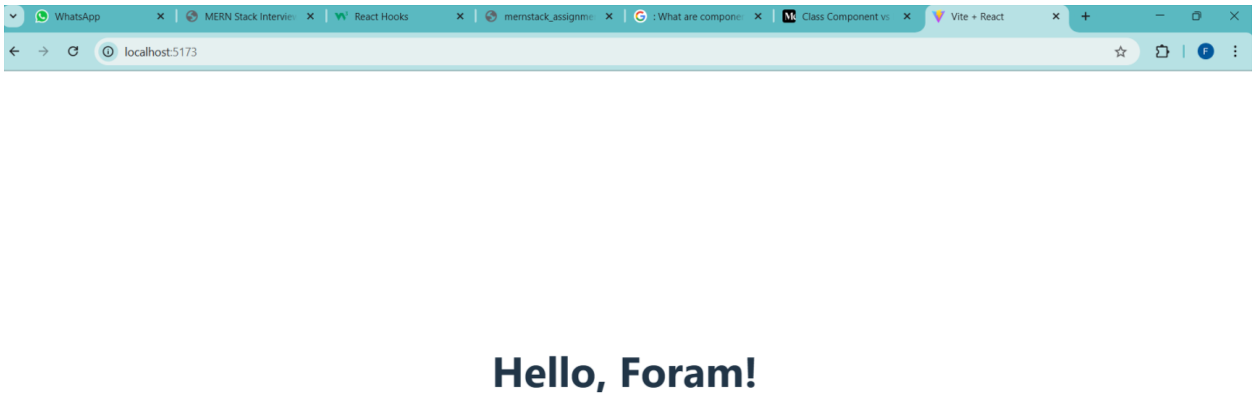
## Greeting.jsx:-



```jsx
import React from "react";

function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}

export default Greeting;
```

```
11:12:42 am [vite] (client) hmr update /src/App.jsx
```

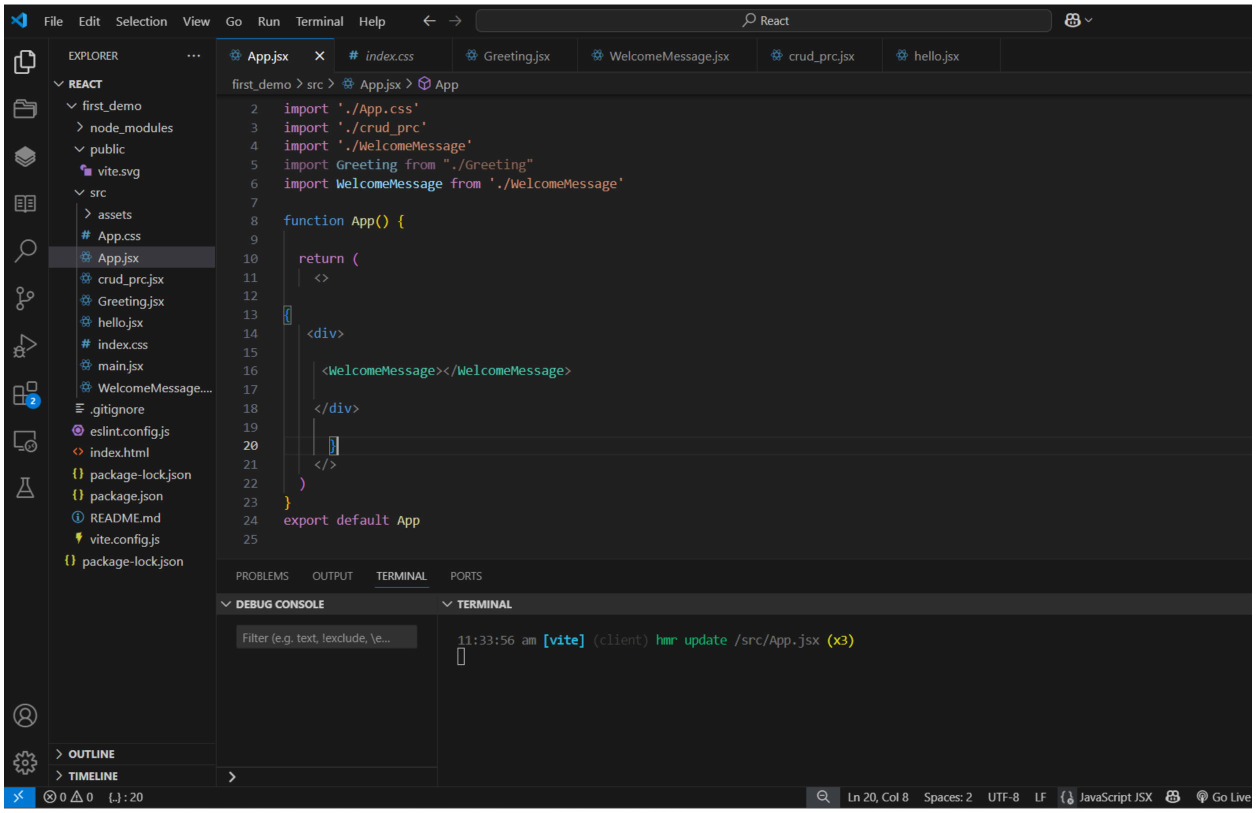
## <u>**OUTPUT:-**</u>



Hello, Foram!

## Task – 2 :-

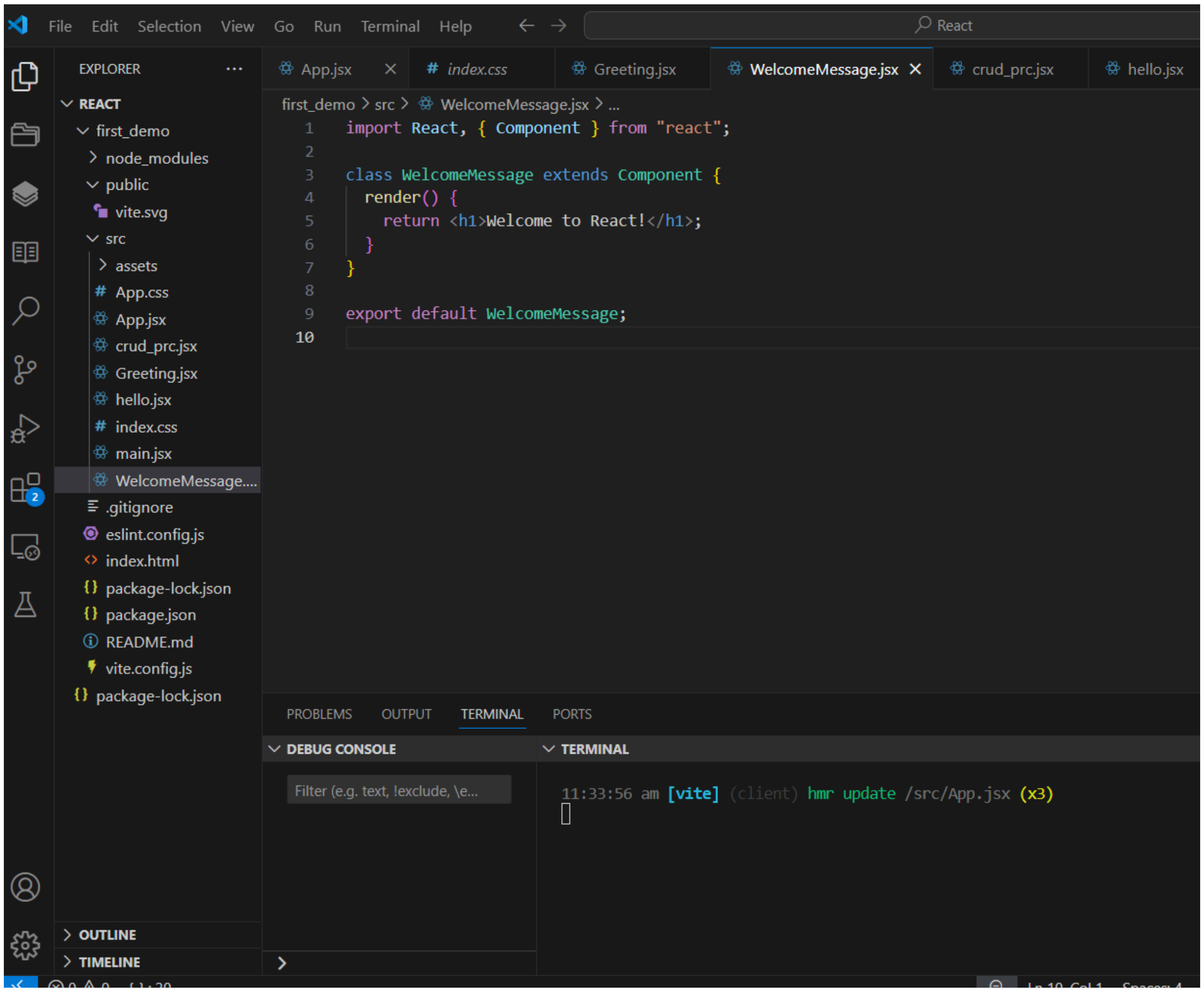
## App.jsx



```
import './App.css'
import './crud_prc'
import './WelcomeMessage'
import Greeting from "./Greeting"
import WelcomeMessage from './WelcomeMessage'

function App() {

  return (
    <>

{

   <div>

     <WelcomeMessage></WelcomeMessage>

    </div>

      }
    </>
  )
}
export default App
```

```
11:33:56 am [vite] (client) hmr update /src/App.jsx (x3)
```

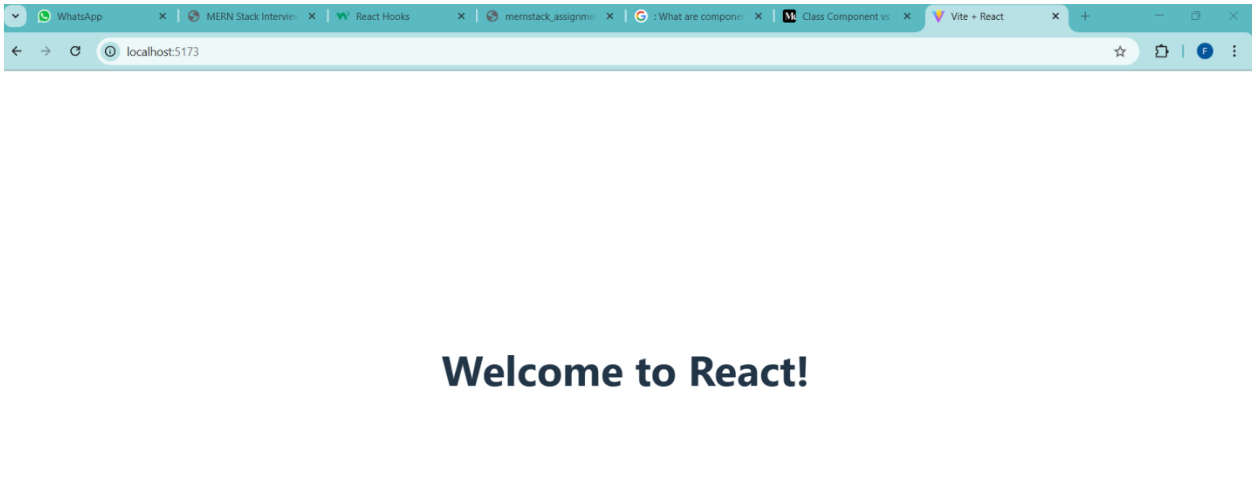
WelcomeMessage.jsx:-



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

class WelcomeMessage extends Component {
  render() {
    return <h1>Welcome to React!</h1>;
  }
}

export default WelcomeMessage;
```

```
11:33:56 am [vite] (client) hmr update /src/App.jsx (x3)
```

**OUTPUT:-**



localhost:5173

Welcome to React!

## Props and state:-

## ANs(1):-

## What are props in React.js:-

Props are used to pass data from a parent component to a child component in React.props are like function arguments.

→Propas Data passed from parent to child. state data managed within a components.

→Popas can't be change. state can change.

→Propas are used to send data . state are used to store data.

## Ans(2):-

## state in React :-

→ State in React is a built-in object that stores data inside a component.

→ React provides a special function called useState to manage state.

Example:-

→State can mutable.

Example:-

Button clicks.

## Ans(3):-

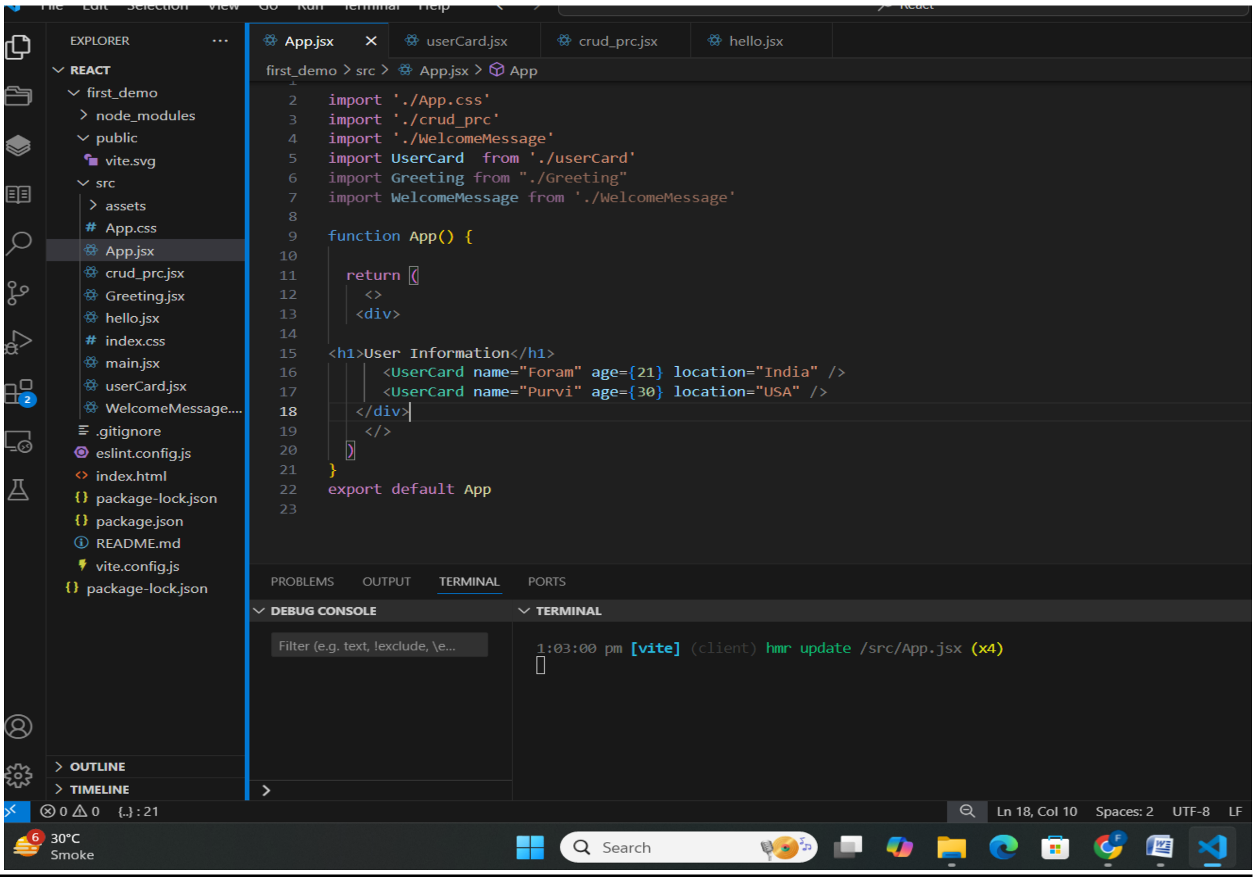
## this.setState()used:-

→Class components this.state is used to store data, but you cannot update state directly by writing this.state = newValue.

→ It merges the new state with the existing state.
→ It triggers a re-render, updating the UI with the new state value.
→It does not change state immediately.

## LAB EXERCISE:-

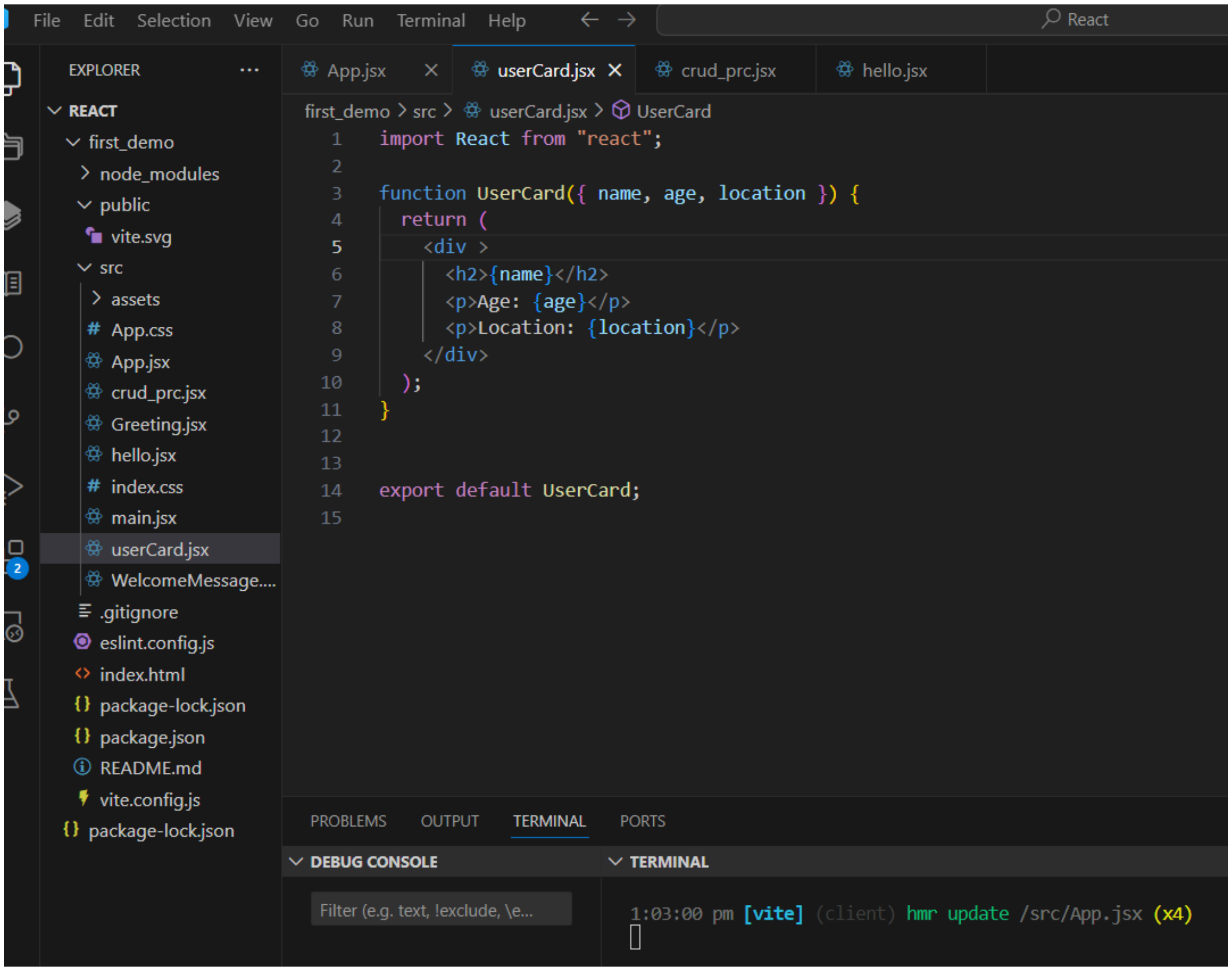
## App.jsx



```
import './App.css'
import './crud_prc'
import './WelcomeMessage'
import UserCard  from './userCard'
import Greeting from "./Greeting"
import WelcomeMessage from './WelcomeMessage'

function App() {

  return (
    <>
   <div>

 <h1>User Information</h1>
      <UserCard name="Foram" age={21} location="India" />
      <UserCard name="Purvi" age={30} location="USA" />
   </div>
    </>
  )
}
export default App
```

```
1:03:00 pm [vite] (client) hmr update /src/App.jsx (x4)
```

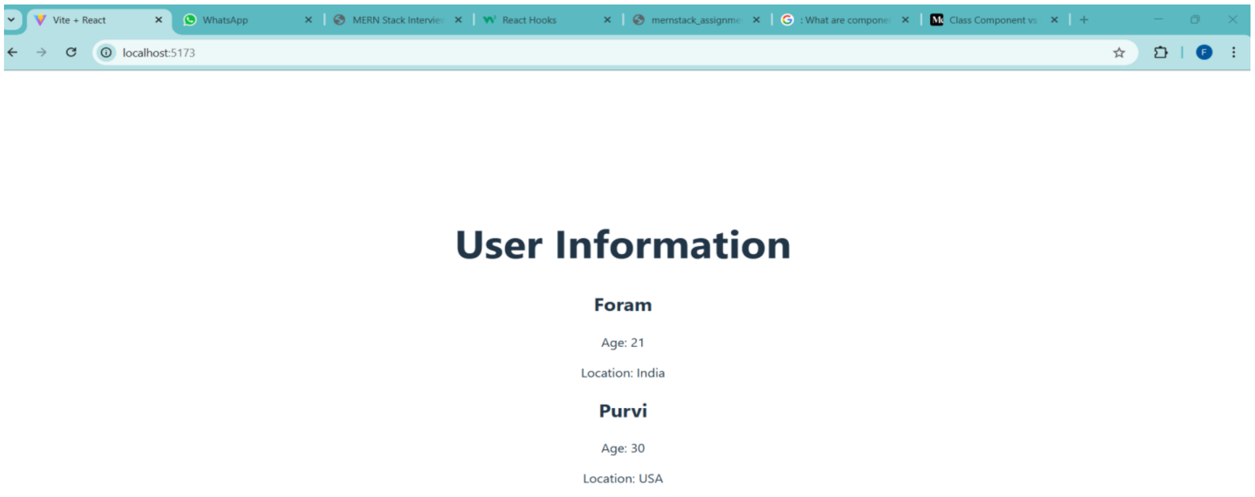
## userCard.jsx



```jsx
import React from "react";

function UserCard({ name, age, location }) {
  return (
    <div >
      <h2>{name}</h2>
      <p>Age: {age}</p>
      <p>Location: {location}</p>
    </div>
  );
}


export default UserCard;
```

```
1:03:00 pm [vite] (client) hmr update /src/App.jsx (x4)
```

## **OUTPUT:-**



User Information

Foram

Age: 21

Location: India

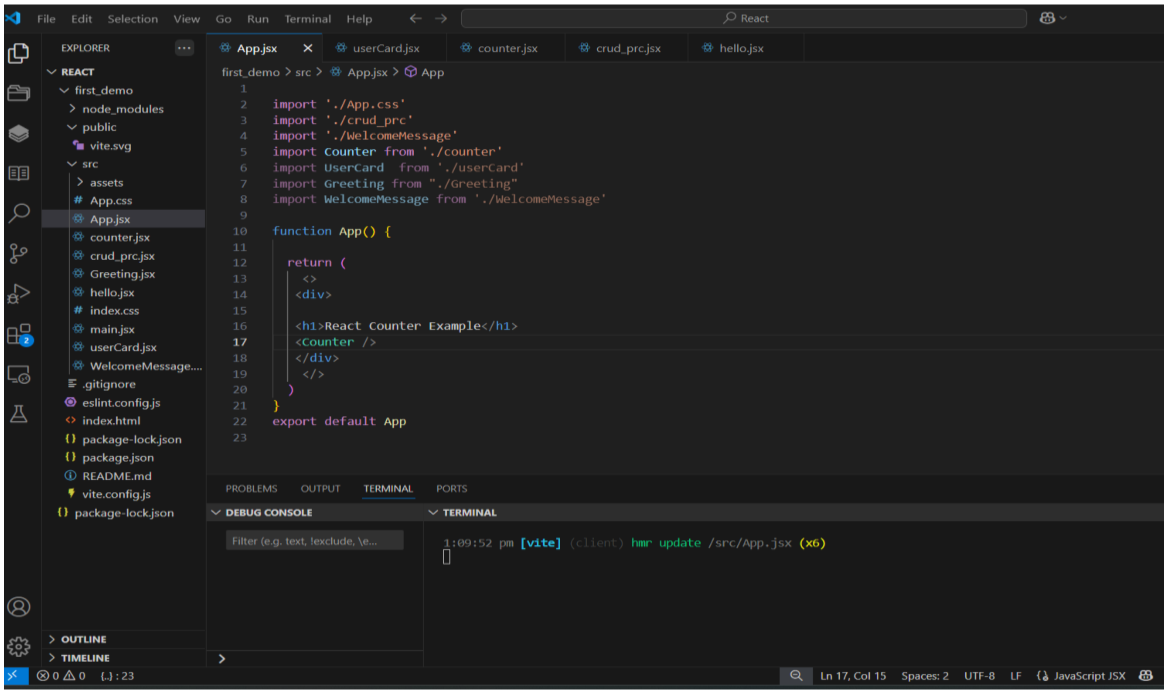
Purvi

Age: 30

Location: USA

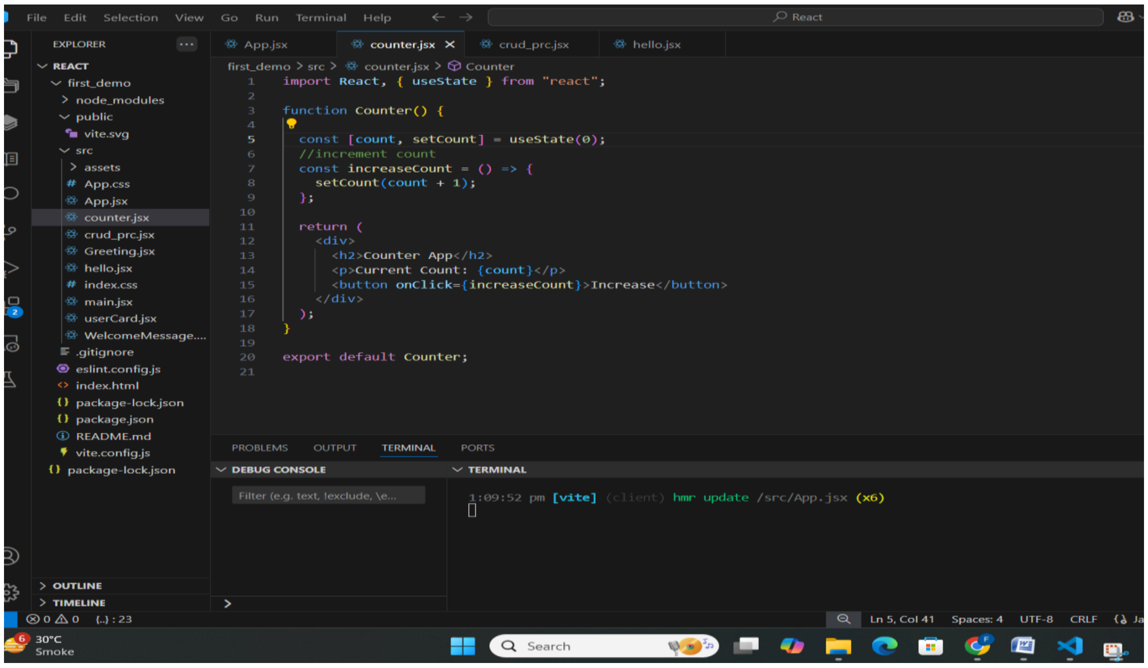
## Task – 2:-

## App.jsx


```
import './App.css'
import './crud_prc'
import './WelcomeMessage'
import Counter from './counter'
import UserCard  from './userCard'
import Greeting from "./Greeting"
import WelcomeMessage from './WelcomeMessage'

function App() {

  return (
   <>
   <div>

   <h1>React Counter Example</h1>
   <Counter />
   </div>
    </>
  )
}
export default App
```

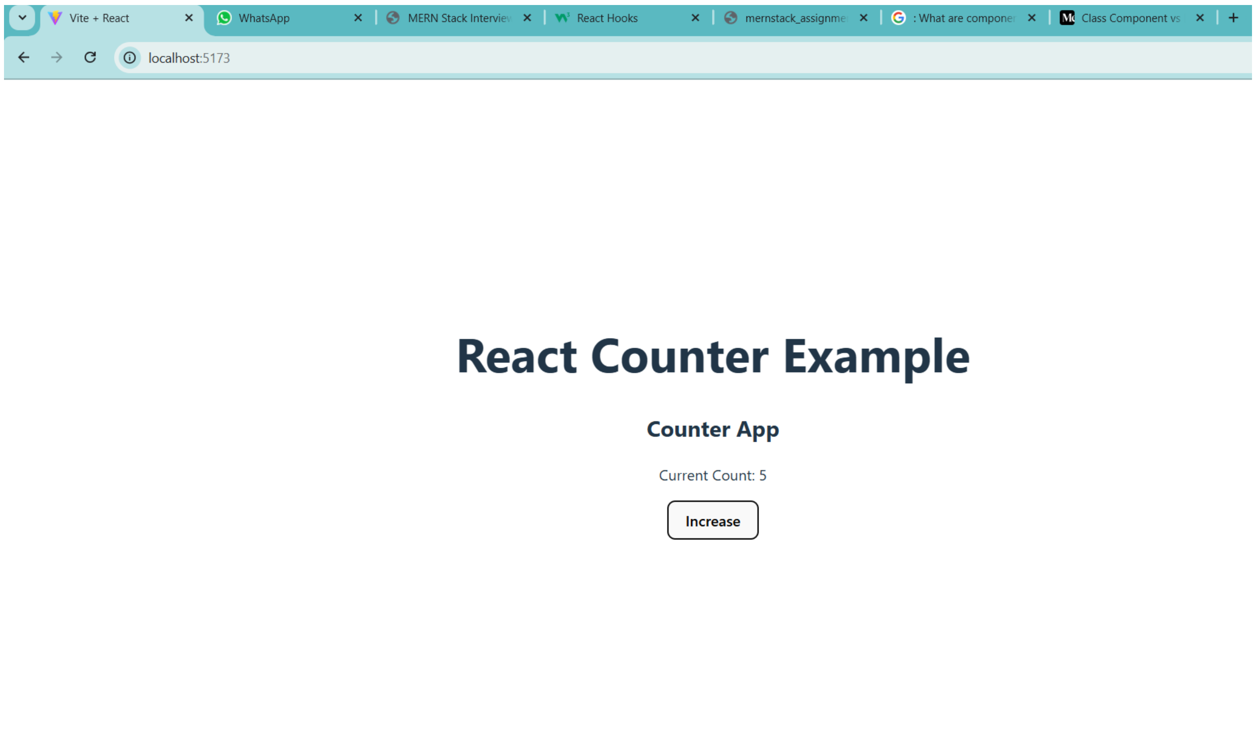
```
1:09:52 pm [vite] (client) hmr update /src/App.jsx (x6)
```

## Counter.jsx


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

function Counter() {

  const [count, setCount] = useState(0);
  //increment count
  const increaseCount = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <h2>Counter App</h2>
      <p>Current Count: {count}</p>
      <button onClick={increaseCount}>Increase</button>
    </div>
  );
}

export default Counter;
```

```
1:09:52 pm [vite] (client) hmr update /src/App.jsx (x6)
```

## **OUTPUT:-**



React Counter Example

Counter App

Current Count: 5