Name: Sufiyan Khan D.O.P: 4-08-23

Roll No: A12 class: TE-A AI&DS

Experiment No. 5

Aim:

React: Installation and Configuration. JSX, Components, Props, State, Forms, Events, Routers, Refs, Keys.

A. Theory:

1. What is React?

React is a JavaScript-based UI development library. Facebook and an open-source developer community run it. Although React is a library rather than a language, it is widely used in web development. The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web development.

React manages the view layer and is used for the development of both web and mobile applications.

2. How does React Work?

React creates a VIRTUAL DOM in memory.

Instead of manipulating the browser's DOM directly, React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM.

React only changes what needs to be changed!

React finds out what changes have been made, and changes only what needs to be changed.

React Vs React-JS

React JS is used to build the user interface of web applications (that is, apps that run on a web browser) React Native is used to build applications that run on both iOS and Android devices (that is, cross-platform mobile applications). React uses HTML, CSS and JavaScript to create interactive user interfaces.

3. Why to use React?

Compared to other frontend frameworks, the React code is easier to maintain and is flexible due to its modular structure. This flexibility, in turn, saves a huge amount of time and cost to businesses. It is used for building interactive user interfaces and web applications quickly and efficiently with significantly less code. React allows us to create reusable UI components.

4. What is JSX?

JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.

5. Installation and Configuration of React

To use React in production, you need npm which is included with Node.js.

To get an overview of what React is, you can write React code directly in HTML.

But in order to use React in production, you need npm and Node.js installed.

A. React Directly in HTML

The quickest way start learning React is to write React directly in your HTML files.

Start by including three scripts, the first two let us write React code in our JavaScripts, and the third, Babel, allows us to write JSX syntax and ES6 in older browsers.

(JavaScript ES6 also known as European Computer Manufacturer's Association (ECMAScript 2015 or ECMAScript 6) is the newer version of JavaScript that was introduced in 2015.

```
<head>
```

```
   <script     src="https://unpkg.com/react@18/umd/react.development.js"
crossorigin></script>
    <script src="https://unpkg.com/react-dom@18/umd/react
dom.development.js" crossorigin></script>
   <script
src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
</head>
```

B. Setting up a React Environment :

1. Install Node.js and NPM

Node.js is a JavaScript runtime environment that is required to run React applications. NPM is a package manager that is used to install and manage JavaScript packages.

To install Node.js and NPM, you can follow these steps:

1. Go to the Node.js website: https://nodejs.org/en/download/ and download the installer for your operating system.
2. Run the installer and follow the on-screen instructions.
3. Once the installation is complete, verify that Node.js and NPM are installed correctly by opening a command prompt and typing the following commands:

```
node -v
npm -v
```

These commands should output the versions of Node.js and NPM that are installed on your computer.

2. Create a React project

Once Node.js and NPM are installed, you can create a React project using the create-react-app command. This command will create a new directory with all of the necessary files to start a React project.

To create a React project, open a command prompt and navigate to the directory where you want to create the project. Then, type the following command:

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

This command will create a new directory called my-app with all of the necessary files to start a React project.

3. Start the React development server

Once you have created a React project, you can start the React development server by typing the following command in the project directory:

```
npm start
```

This command will start a development server on port 3000. You can access your React application at *http://localhost:3000*.

The development server will watch your files for changes and automatically rebuild your application when you make changes. This makes it easy to develop and debug React applications.

4. Install additional packages

You may need to install additional packages to use certain features in React. You can install packages using the *npm install* command.

For example, to install the *react-router* package, which is used to create routing in React applications, you would type the following command:

```
npm install react-router
```

5. Start coding!

Once you have set up your React environment, you can start coding your React application. You can use any text editor to write your React code.

The React documentation is a great resource for learning how to use React. You can also find many tutorials and examples online.

B. Program:

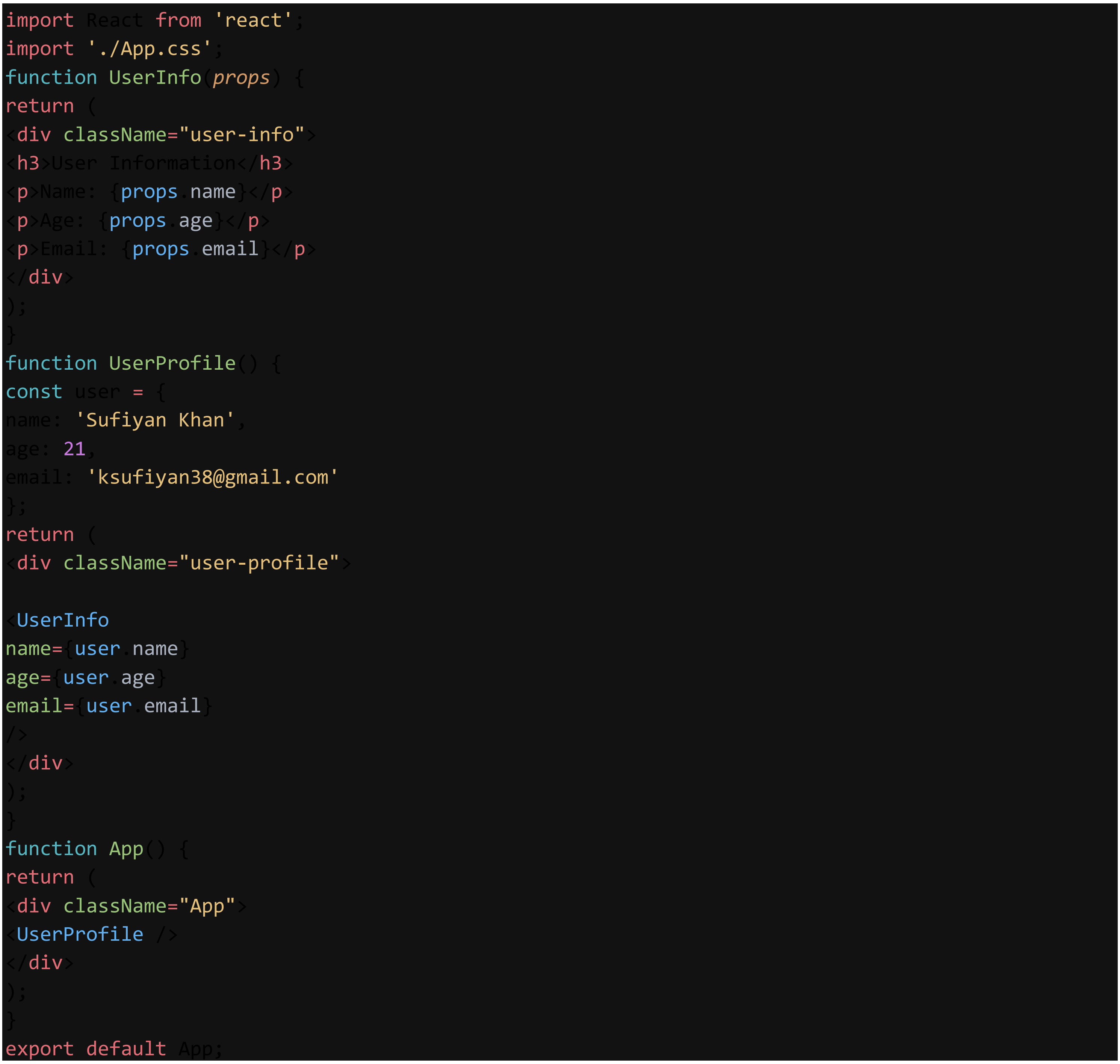
1. JSX:

```
import React from 'react'; import './App.css';

function App() {
const greeting = 'Hello, Sufiyan'; const isUserLoggedIn = true;

return (
<div className="App">
<header className="App-header">
<h1>{greeting}</h1>
{isUserLoggedIn ? <p>Welcome back, User!</p> : <p>Please log in.</p>}
</header>
</div>
);
}

export default App;
```

2. Components:

```
import React, { useState } from 'react'; import './App.css';
function Counter() {
const [count, setCount] = useState(0);

const increment = () => { setCount(count + 1);
};

const decrement = () => { setCount(count - 1);
};

return (
<div className="counter">
<h2>Counter App</h2>
<p>Count: {count}</p>
<button onClick={increment}>Increment</button>
<button onClick={decrement}>Decrement</button>
</div>
);
}

function App() { return (
<div className="App">
<Counter />
</div>
);
}

export default App;
```

3. Props:



```
import React from 'react';
import './App.css';
function UserInfo(props) {
return (
<div className="user-info">
<h3>User Information</h3>
<p>Name: {props.name}</p>
<p>Age: {props.age}</p>
<p>Email: {props.email}</p>
</div>
);
}
function UserProfile() {
const user = {
name: 'Sufiyan Khan',
age: 21,
email: 'ksufiyan38@gmail.com'
};
return (
<div className="user-profile">

<UserInfo
name={user.name}
age={user.age}
email={user.email}
/>
</div>
);
}
function App() {
return (
<div className="App">
<UserProfile />
</div>
);
}
export default App;
```

4. State:

```
import React, { Component } from 'react'; import './App.css';

class Toggle extends Component { constructor(props) {
super(props); this.state = {
showContent: false
};
}

toggleContent = () => { this.setState({
showContent: !this.state.showContent
});
};

render() { return (
<div className="toggle">
<h2>Toggle App with State</h2>
<button onClick={this.toggleContent}>
{this.state.showContent ? 'Hide Content' : 'Show Content'}
</button>
{this.state.showContent && (
<div className="content">
<p>This is the hidden content that can be toggled.</p>
</div>
)}
</div>
);
}
}

function App() { return (

<div className="App">
<Toggle />
</div>
);
}

export default App;
```

5. Forms:

```
import React, { Component } from 'react'; import './App.css';

class ContactForm extends Component { constructor(props) {
super(props); this.state = {
name: '',
email: '', message: ''
};
}

handleChange = event => {
const { name, value } = event.target; this.setState({
[name]: value
});
};

handleSubmit = event => { event.preventDefault();
console.log('Form submitted:', this.state);
};

render() { return (
<div className="contact-form">
<h2>Contact Us</h2>
<form onSubmit={this.handleSubmit}>
<div>
<label>Name:</label>
<input type="text" name="name"
value={this.state.name}

onChange={this.handleChange}
/>
</div>
<div>
<label>Email:</label>
<input type="email" name="email"
value={this.state.email} onChange={this.handleChange}
/>
</div>
<div>
<label>Message:</label>
<textarea name="message"
value={this.state.message} onChange={this.handleChange}
/>
</div>
<button type="submit">Submit</button>
</form>
</div>
);
}
}
```

```
function App() { return (
<div className="App">
<ContactForm />
</div>
);
}

export default App;
```

6. Events:

```
import React, { Component } from 'react'; import './App.css';

class ClickCounter extends Component { constructor(props) {
super(props); this.state = {

count: 0
};
}

handleIncrement = () => { this.setState({
count: this.state.count + 1
});
};

handleReset = () => { this.setState({
count: 0
});
};

render() { return (
<div className="click-counter">
<h2>Click Counter</h2>
<p>Count: {this.state.count}</p>
<button onClick={this.handleIncrement}>Increment</button>
<button onClick={this.handleReset}>Reset</button>
</div>
);
}
}

function App() { return (
<div className="App">
<ClickCounter />
</div>
);
}

export default App;
```

7. Routers:

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route, Link } from 'react- router-dom';
import './App.css'; function Home() {

return <h2>Welcome to Recipe Book!</h2>;
}

function Recipes() { return (
<div>
<h2>Recipes</h2>
<ul>
<li>
<Link to="/recipes/pasta">Pasta</Link>
</li>
<li>
<Link to="/recipes/pizza">Pizza</Link>
</li>
<li>
<Link to="/recipes/salad">Salad</Link>
</li>
</ul>
</div>
);
}

function RecipeDetails({ match }) {
const recipeName = match.params.recipeName; return <h3>Recipe details for {recipeName}</h3>;
}

function App() { return (
<Router>
<div className="App">
<nav>
<ul>
<li>
<Link to="/">Home</Link>
</li>
<li>
<Link to="/recipes">Recipes</Link>
</li>
</ul>
</nav>

<Routes>
<Route path="/" element={<Home />} />
<Route path="/recipes" element={<Recipes />} />
<Route path="/recipes/:recipeName" element={<RecipeDetails />}
/>

</Routes>
</div>
</Router>
);
}

export default App;
```

8. Refs:

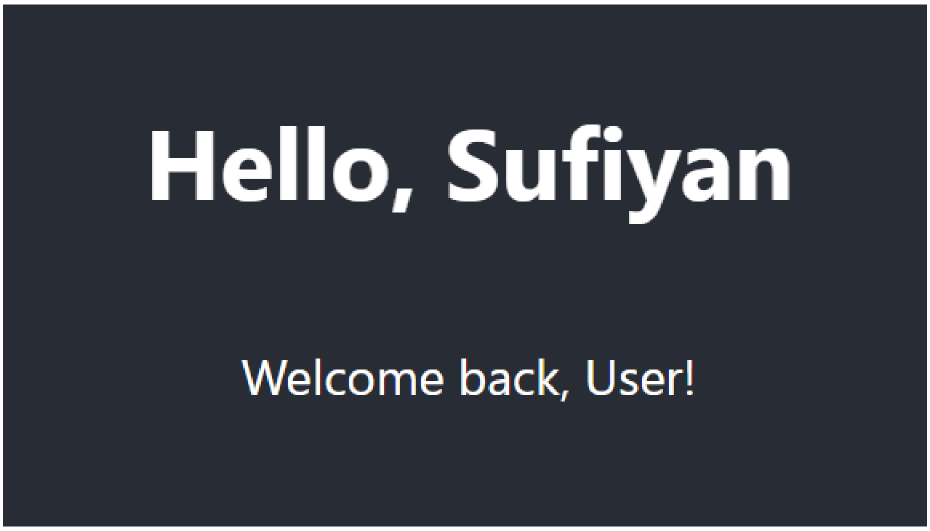
```
import React, { Component } from 'react'; import './App.css';

class FocusInput extends Component { constructor(props) {
super(props);
this.inputRef = React.createRef();
}

focusInput = () => { this.inputRef.current.focus();
};

render() { return (
<div className="focus-input">
<h2>Focus Input Example</h2>
<input ref={this.inputRef} type="text" />
<button onClick={this.focusInput}>Focus Input</button>
</div>
);
}
}

function App() { return (
<div className="App">
<FocusInput />
</div>
);
}

export default App;
```

9. Keys:

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

import './App.css';

class TaskList extends Component { constructor(props) {
super(props); this.state = {
tasks: [
{ id: 1, text: 'Complete assignment' },
{ id: 2, text: 'Go to the gym' },
{ id: 3, text: 'Buy groceries' }
]
};
}

render() {
const { tasks } = this.state; return (
<div className="task-list">
<h2>Task List</h2>
<ul>
{tasks.map(task => (
<li key={task.id}>{task.text}</li>
))}
</ul>
</div>
);
}
}

function App() { return (
<div className="App">
<TaskList />
</div>
);
}

export default App;
```

C. Output and findings:

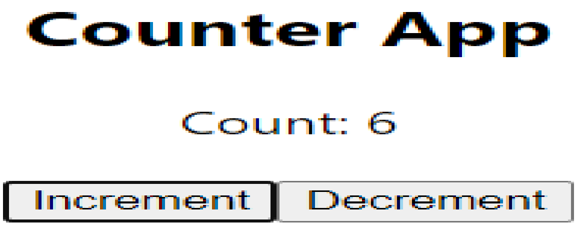
1. JSX:



Hello, Sufiyan

Welcome back, User!

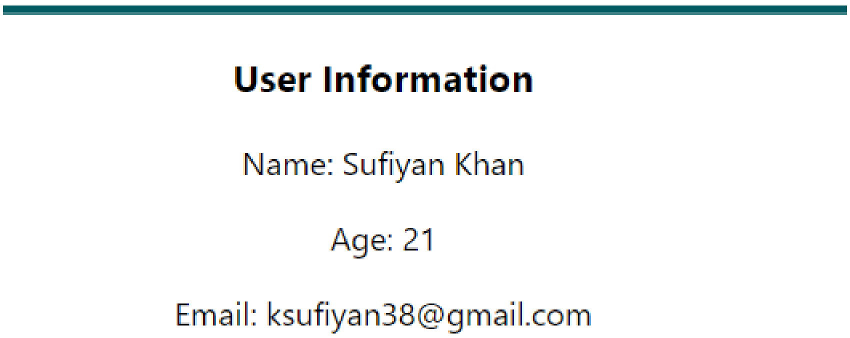
2. Components:



Counter App

Count: 6

3. Components:



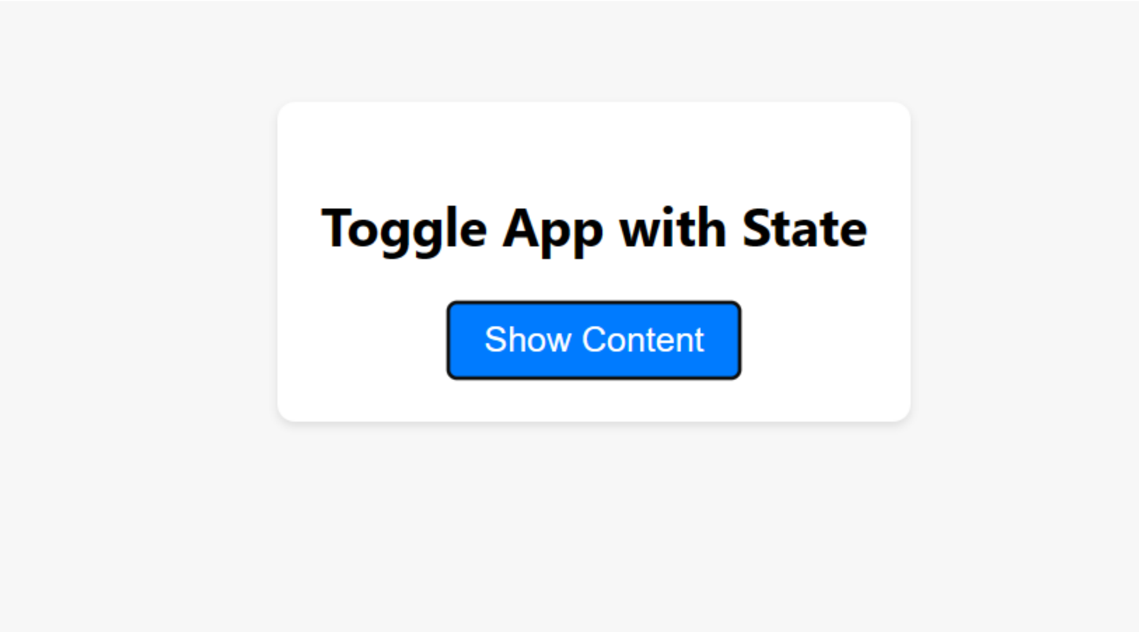
User Information

Name: Sufiyan Khan

Age: 21

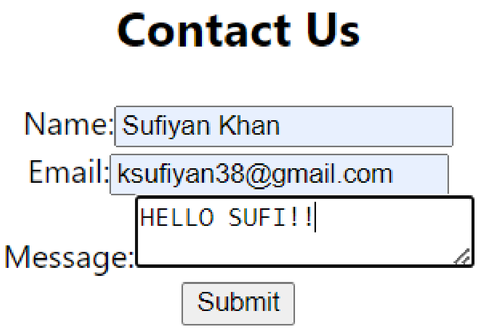
Email: ksufiyan38@gmail.com

4. State:



Toggle App with State

5. Forms:



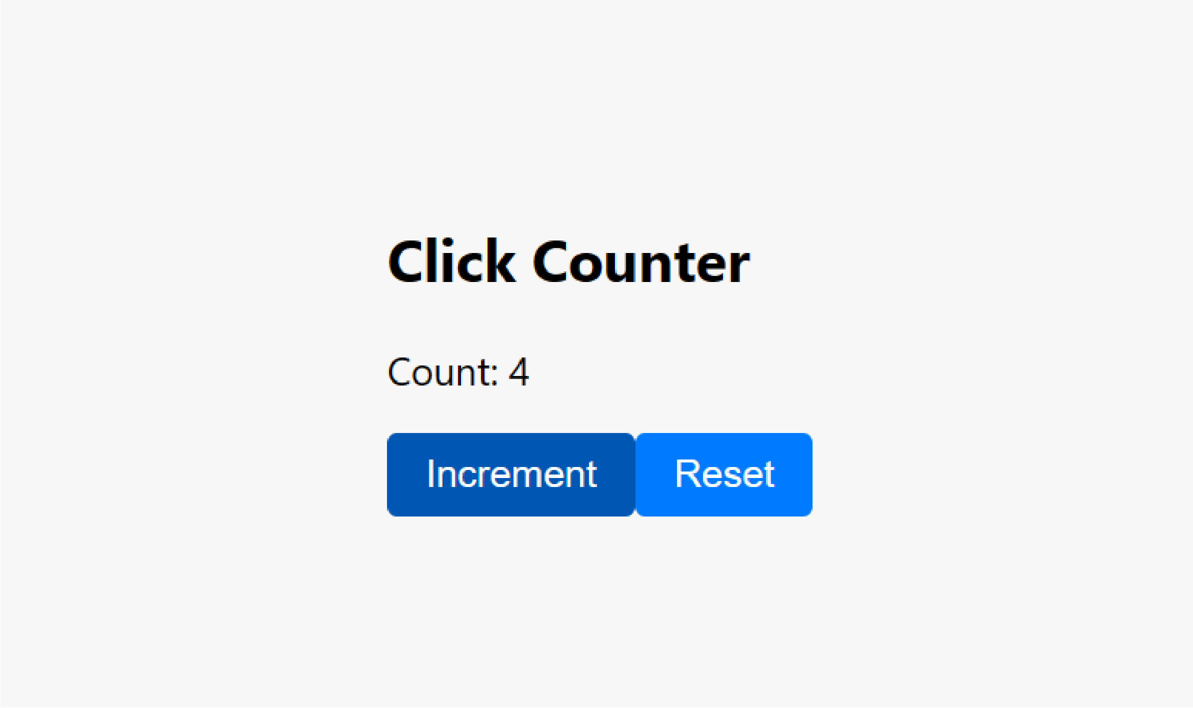
Contact Us

Name: Sufiyan Khan

Email: ksufiyan38@gmail.com

Message: HELLO SUFI!!

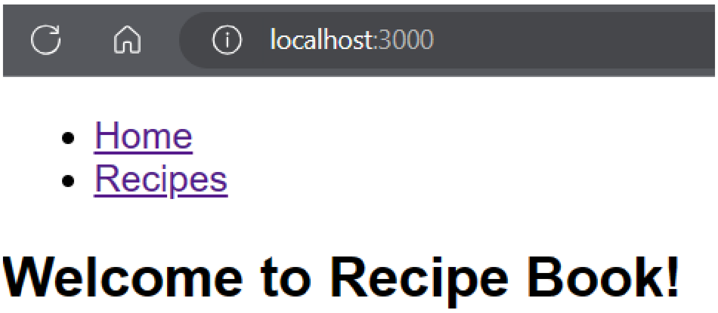
6. Events:



Click Counter

Count: 4

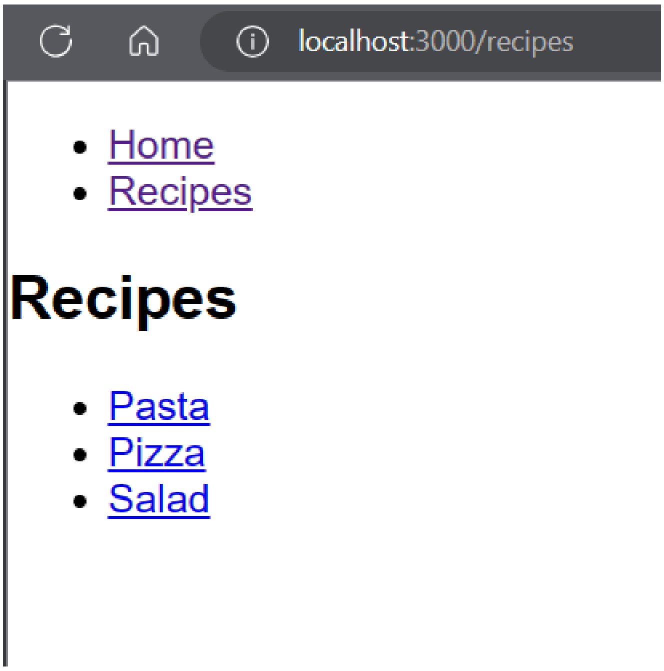
7. Routers



localhost:3000

- Home
- Recipes

Welcome to Recipe Book!



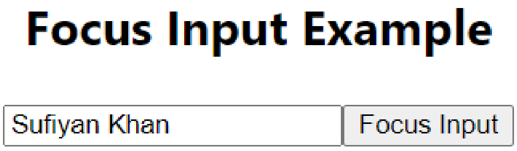
localhost:3000/recipes

- Home
- Recipes

Recipes

- Pasta
- Pizza
- Salad

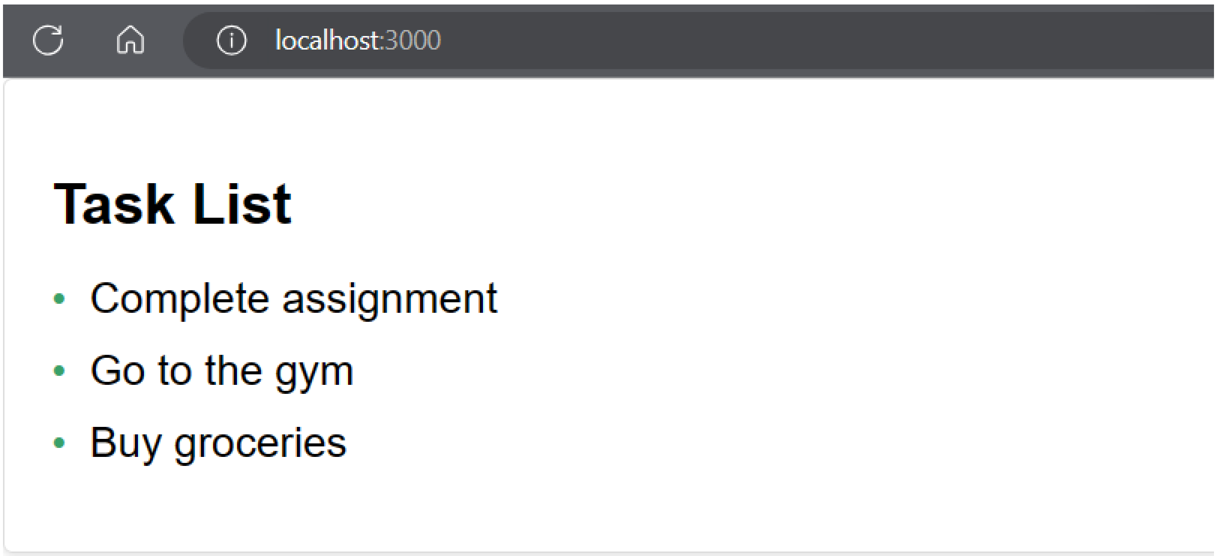
8. Refs:



Focus Input Example

Sufiyan Khan

9. Keys:



localhost:3000

Task List

- Complete assignment
- Go to the gym
- Buy groceries

D. Conclusion

Hence in this experiment, we studies the different operations and aspects of React after installing it with the help of Node.js and successfully implemented each of the functions of React.