**React JS – Hands on Assignment -1**

**1. Create a Simple React App Display “Hello, World!” inside an Tag.**

//app.jsimport "./App.css";

function App() {

return (

<div>

<h1>Hello, World!</h1>

</div>

);

}

export default App;

//**index.js**

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

****

**2. Write JSX to Display a Paragraph Inside a , display a tag with your name.**

//app.js

import "./App.css";

function App() {

return (

<div>

<p>Mansi shelar</p>

</div>

);

}

export default App;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);



**3. Create a JSX expression that renders the current date and time dynamically**

//app.js

import "./App.css";

function App() {

const currentDate = new Date().toLocaleDateString();

const currentTime = new Date().toLocaleTimeString();

return (

<div>

<p>

{currentDate} - {currentTime}

</p>

</div>

);

}

export default App;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

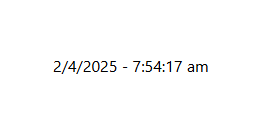
root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);



**4. Create a simple functional component that returns a heading**

//Head.js

import "./Head.css";

function Head() {

return (

<div>

<h2>This is a Heading component.</h2>

</div>

);

}

export default Head;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import Head from "./heading";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

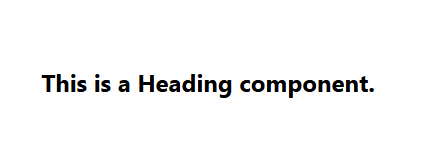
root.render(

<React.StrictMode>

<Head />

</React.StrictMode>

);



**5. Write a program to demonstration of nested functional component.**

//app.js

import "./App.css";

function Outname() {

return (

<div className="App">

<InnerName />

</div>

);

}

function InnerName() {

return <h2>This is an example of inner funtional component in react js</h2>;

}

export default Outname;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App.js";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

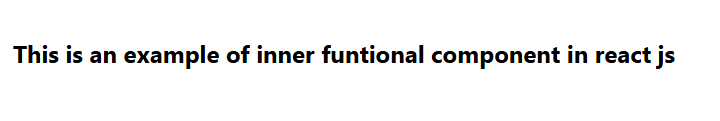
<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();



**6. Create a parent component that passes a name prop to a child component and displays it.**

//child.js

function Child(props) {

return <p>Hello, {props.name}!</p>;

}

export default Child;

//app.js

import "./App.css";

import Child from "./child";

function App() {

return (

<div>

<Child name="Mansi" />

</div>

);

}

export default App;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App.js";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();



**7. Create an array of students which includes roll number, name and age pass it as a props to functional component and display all information.**

//app.js

import "./App.css";

import StudentList from "./Studentlist";

function App() {

const students = [

{ rollNumber: 1, name: "Mansi", age: 20 },

{ rollNumber: 2, name: "Rosh", age: 21 },

{ rollNumber: 3, name: "Tanya", age: 19 },

];

return (

<div>

<StudentList students={students} />

</div>

);

}

export default App;

//Studentlist.js

function StudentList(props) {

return (

<div>

<h3>Student List</h3>

<ul>

{props.students.map((student) => (

<li key={student.rollNumber}>

Roll: {student.rollNumber}, Name: {student.name}, Age: {student.age}

</li>

))}

</ul>

</div>

);

}

export default StudentList;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App.js";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

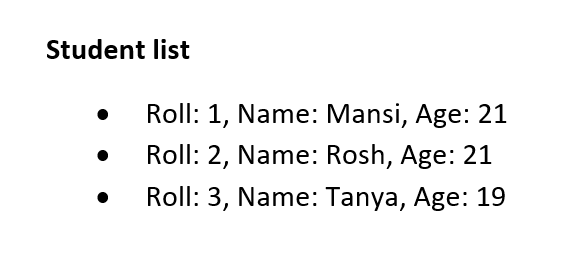
<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();



**8. Pass an array of colors (as props) to a child component and render them as <li> items.**

//Colorlist.js

function ColorList(props) {

return (

<div>

<h3>Colors</h3>

<ul>

{props.colors.map((color, index) => (

<li key={index} style={{ color: color }}>

{color}

</li>

))}

</ul>

</div>

);

}

export default ColorList;

//app.js

import "./App.css";

import ColorList from "./Colorlist";

function App() {

const colors = ["red", "green", "blue", "yellow", "purple"];

return (

<div>

<ColorList colors={colors} />

</div>

);

}

export default App;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App.js";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();



**9. Build a counter with two buttons (+ and -) to increment/decrement a state variable.**

//Counter.js

import { useState } from "react";

function Counter() {

const [count, setCount] = useState(0);

return (

<div>

<h3>Counter: {count}</h3>

<button onClick={() => setCount(count + 1)}>+</button>

<button onClick={() => setCount(count - 1)}>-</button>

</div>

);

}

export default Counter;

//app.js

import "./App.css";

import Counter from "./Counter";

function App() {

return (

<div>

<Counter />

</div>

);

}

export default App;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App.js";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

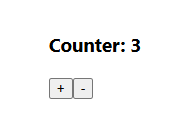
<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();



**10. Create an input field that logs its value to the console on every keystroke (use onChange).**

//Input

function Input() {

const handleChange = (event) => {

console.log(event.target.value);

};

return <input type="text" onChange={handleChange} />;

}

export default Input;

//app.js

import "./App.css";

import Input from "./Input";

function App() {

return (

<div>

<Input />

</div>

);

}

export default App;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App.js";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

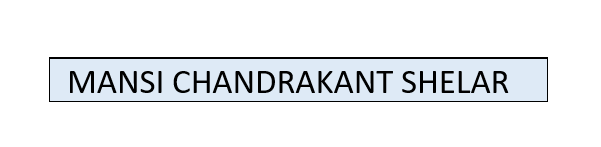
<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();



**11. Create a button that changes its background color when clicked (use onClick and state).**

//ColorButton.js

import { useState } from "react";

function ColorButton() {

const [bgColor, setBgColor] = useState("white");

const colors = ["red", "green", "blue", "yellow", "purple"];

const changeColor = () => {

const randomIndex = Math.floor(Math.random() \* colors.length);

setBgColor(colors[randomIndex]);

};

return (

<button

onClick={changeColor}

style={{ backgroundColor: bgColor, padding: "10px 20px" }}

>

Click me to change color

</button>

);

}

export default ColorButton;

//app.js

import "./App.css";

import ColorButton from "./ColorButton";

function App() {

return (

<div>

<ColorButton />

</div>

);

}

export default App;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App.js";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

<React.StrictMode>

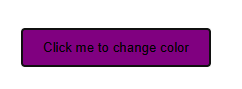
<App />

</React.StrictMode>

);

reportWebVitals();







**12. Build a form with name and email fields that alerts the submitted values (use onSubmit).**

//ContactForm.js

import { useState } from "react";

function ContactForm() {

const [formData, setFormData] = useState({

name: "",

email: "",

});

const handleSubmit = (event) => {

event.preventDefault();

alert(`Name: ${formData.name}\nEmail: ${formData.email}`);

};

const handleChange = (event) => {

setFormData({

...formData,

[event.target.name]: event.target.value,

});

};

return (

<form onSubmit={handleSubmit}>

<div>

<label>Name:</label>

<input

type="text"

name="name"

value={formData.name}

onChange={handleChange}

required

/>

</div>

<div>

<label>Email:</label>

<input

type="email"

name="email"

value={formData.email}

onChange={handleChange}

required

/>

</div>

<button type="submit">Submit</button>

</form>

);

}

export default ContactForm;

//app.js

import "./App.css";

import ContactForm from "./ContactForm";

function App() {

return (

<div>

<ContactForm />

</div>

);

}

export default App;

//index.js

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import App from "./App.js";

import reportWebVitals from "./reportWebVitals";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

reportWebVitals();

