**Create Project using Create React App**

1. Need node.js, install and check:

C:\xampp\htdocs\react>node -v

v18.17.1

C:\xampp\htdocs\react>npm -v

9.6.7

C:\xampp\htdocs\react>npx create-react-app myreactapp

Chrome Extension for react projects:

Search “react developer tools for chrome” in google chrome

<https://chromewebstore.google.com/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi?hl=en&pli=1>

Extensions for Visual Studio Code:

**React Native Tools**

**ES7+ React/Redux/React-Native snippets**

React using camelcase like backgroundColor.

Delete files in src folder except index.css and index.js.

Index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

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

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

App.js:

import { Component } from "react";

class App extends Component{

    render(){

        return (

        <div className="App">

            Hello World

            </div>

            )

    }

}

export default App

In terminal, type this command to run react project:

>npm start

You can now view myreactapp in the browser.

Local: http://localhost:3000

On Your Network: http://172.21.16.1:3000

Note that the development build is not optimized.

To create a production build, use npm run build.

webpack compiled successfully

Component - Functional and Class Component:

Table.js:

import { Component } from "react";

const TableHeader = ()=>{

    return (

                <thead>

                    <tr>

                        <th>Name</th>

                        <th>Role</th>

                    </tr>

                </thead>

    )

}

const TableBody = ()=>{

    return(

                <tbody>

                    <tr>

                        <td>Vignesh</td>

                        <td>Developer</td>

                    </tr>

                    <tr>

                        <td>Vengadesh</td>

                        <td>Finance</td>

                    </tr>

                </tbody>

    )

}

class Table extends Component{

    render() {

        return(

            <table>

                <TableHeader/>

                <TableBody/>

            </table>

        )

    }

}

export default Table;

App.js:

import { Component } from "react";

import Table from "./Table";

class App extends Component{

    render(){

        return (

        <div className="App">

            <h1>Table</h1>

            <Table/>

            </div>

            )

    }

}

export default App

Props (Properties):

In React, "props" (short for properties) are a mechanism for passing data from a parent component to a child component. They are a fundamental concept in React for building reusable and dynamic user interfaces.

Props in Functional Component:

App.js:

import { Component } from "react";

import Garage from "./components/Garage";

class App extends Component{

    render(){

        return (

        <Garage/>

            )

    }

}

export default App

Car.js:

function Car(props){

    const {carInfo} = props;

    const {brand,color} = carInfo;

    const text = `Hi, this is my ${color} ${brand} car`;

    return(

        <h2>{text}</h2>

    )

}

export default Car;

Garage.js:

import Car from "./Car"

function Garage(){

    const carInfo = { brand:"Reynauld", color:"Black"}

    return(

        <Car carInfo={carInfo}/>

    )

}

export default Garage;

Output:

## Hi, this is my Black Reynauld car

Props in Class Component:

Apple.js:

import { Component } from "react";

class Apple extends Component{

    render(){

        const {appleInfo} = this.props;

        const {type,color} = appleInfo;

        return(

            <h2>Im {color} {type} Apple</h2>

        )

    }

}

export default Apple;

Garage.js:

import Apple from "./Apple";

import Car from "./Car"

function Garage(){

    const carInfo = { brand:"Reynauld", color:"Black"}

    const appleInfo = {type:"Fuji",color:"Red"}

    return(

        <>

        <Car carInfo={carInfo}/>

        <Apple appleInfo={appleInfo}/>

        </>

    )

}

export default Garage;

App.js:

import { Component } from "react";

import Garage from "./components/Garage";

class App extends Component{

    render(){

        return (

        <Garage/>

            )

    }

}

export default App

Conditional Rendering:

import Apple from "./Apple";

import Car from "./Car"

function Garage(){

    const carInfo = { brand:"Reynauld", color:"Black"}

    // const appleInfo = {type:"Fuji",color:"Red"}

    const appleInfo = {}

    return(

        <>

        <Car carInfo={carInfo}/>

//Conditional Rendering

        {

//Ternary operator

            appleInfo.type !== undefined && appleInfo.color !== undefined ?

            <Apple appleInfo={appleInfo}/> : null

        }

        </>

    )

}

export default Garage;

import Apple from "./Apple";

import Car from "./Car"

function Garage(){

    const carInfo = { brand:"Reynauld", color:"Black"}

    const appleInfo = {type:"Fuji",color:"Red"}

    // const appleInfo = {};

    const showappleInfo = appleInfo.type !== undefined && appleInfo.color !== undefined;

    return(

        <>

        <Car carInfo={carInfo}/>

        {showappleInfo && <Apple appleInfo={appleInfo}/>}

        </>

    )

}

export default Garage;

List:

import Apple from "./Apple";

import Car from "./Car"

function Garage(){

    const carInfo = { brand:"Reynauld", color:"Black", model:"Logan"}

    const appleInfo = {type:"Fuji",color:"Red"}

    const carList = [

        {brand:"Mahindra", color:"White", model:"Scorpio"},

        {brand:"BMW", color:"Black", model:"X6"},

        {brand:"Tesla", color:"Green", model:"Electric"}

    ]

    // const appleInfo = {};

    const showappleInfo = appleInfo.type !== undefined && appleInfo.color !== undefined;

    return(

        <>

        <Car carInfo={carInfo}/>

        {showappleInfo && <Apple appleInfo={appleInfo}/>}

        <ul>

        {carList.map((car)=><li><Car carInfo={car}/></li>)}

        </ul>

        </>

    )

}

export default Garage;

List Keys:

import Apple from "./Apple";

import Car from "./Car"

function Garage(){

    const carInfo = { brand:"Reynauld", color:"Black", model:"Logan"}

    const appleInfo = {type:"Fuji",color:"Red"}

    const carList = [

        {brand:"Mahindra", color:"White", model:"Scorpio"},

        {brand:"BMW", color:"Black", model:"X6"},

        {brand:"Tesla", color:"Green", model:"Electric"}

    ]

    // const appleInfo = {};

    const showappleInfo = appleInfo.type !== undefined && appleInfo.color !== undefined;

    return(

        <>

        <Car carInfo={carInfo}/>

        {showappleInfo && <Apple appleInfo={appleInfo}/>}

        <ul>

        {carList.map((car,index)=><li key={index}><Car carInfo={car}/></li>)}

        </ul>

        </>

    )

}

export default Garage;

Inline CSS Styling:

Header.js:

function Header(){

    const paragraphStyle = {

        color:"green",

        backgroundColor:"black",

        fontFamily:"sans-serif",

        padding:"10px"

    }

    return(

        <>

        <h1 style={{color:"red",backgroundColor:"lightblue"}}>Heading</h1>

        <p style={paragraphStyle}>Paragraph</p>

        </>

    )

}

export default Header;

App.js:

import { Component } from "react";

import Header from "./components/Header";

class App extends Component{

    render(){

        return (

        <Header/>

            )

    }

}

export default App

CSS Stylesheet:

Header.js:

import './Header.css';

import styles from './Header.module.css';

function Header(){

    const paragraphStyle = {

        color:"green",

        backgroundColor:"black",

        fontFamily:"sans-serif",

        padding:"10px"

    }

    return(

        <>

        {/\* <h1 style={{color:"red",backgroundColor:"lightblue"}}>Heading</h1> \*/}

        <h1 className={styles.bigFont}>Heading style</h1>

        <p style={paragraphStyle}>Paragraph</p>

        </>

    )

}

export default Header;

Header.css:

body{

    color: aqua;

    background-color: brown;

    font-family: sans-serif;

    padding: 10px;

    text-align: center;

}

Header.module.css:

.bigFont{

    font-size: large;

    color: darkturquoise;

}

React State:

Create state in Functional Component:

ChangeColor.js

import { useState } from "react";

function ChangeColor(){

    const [color,setColor] = useState('Red');

    return(

        <>

        <h1>This is {color} color</h1>

        <button onClick={()=>{setColor('Green')}}>Change Color</button>

        </>

    )

}

export default ChangeColor;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import ChangeColor from './components/ChangeColor';

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

root.render(

  <React.StrictMode>

    <ChangeColor />

  </React.StrictMode>

);

Create state in Class Component:

FavoriteColor.js

import React from "react";

class FavoriteColor extends React.Component{

    constructor(){

        super();

        this.state = {color:"Red"};

    }

    render(){

        return(

        <>

        <h1>This is {this.state.color} color.</h1>

        <button onClick={()=>{this.setState({color:"Green"})}}>Change Color</button>

        </>

        )

    }

}

export default FavoriteColor;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import ChangeColor from './components/ChangeColor';

import FavoriteColor from './components/FavoriteColor';

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

root.render(

  <React.StrictMode>

    <ChangeColor />

    <FavoriteColor />

  </React.StrictMode>

);

Multiple State in Hooks:

Scooter.js:

import {useState } from "react"

function Scooter() {

const [color, setColor] = useState("Red")

const [brand, setBrand] = useState("Honda")

const [model, setModel] = useState("Activa 6G")

const [year, setYear] = useState("2020")

return <>

<h1>My Scooter</h1>

<p>Color: {color}</p>

<p>Brand: {brand}</p>

<p>Model: {model}</p>

<p>Year: {year}</p>

</>

}

export default Scooter;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import ChangeColor from './components/ChangeColor';

import FavoriteColor from './components/FavoriteColor';

import Scooter from './components/Scooter';

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

root.render(

  <React.StrictMode>

    <ChangeColor />

    <FavoriteColor />

    <Scooter/>

  </React.StrictMode>

);

State with Object and Updating state in Object in Function:

… spread operator copy previous state value of object.

Scooter.js:

import { useState } from "react"

function Scooter() {

const [scooter, setScooter] = useState(

{color: "Red",

brand: "Honda",

model: "Activa 6G",

year: "2021"

}

);

function updateColor(){

    setScooter(previousState=>{

        return {...previousState,color:"Blue"};

    })

}

console.log("current state:", {scooter});

return <>

<h1>My Scooter</h1>

<p>Color: {scooter.color}</p>

<p>Brand: {scooter.brand}</p>

<p>Model: {scooter.model}</p>

<p>Year: {scooter.year}</p>

<button onClick={updateColor}>Change Color</button>

</>

}

export default Scooter;

Updating Object in Class Component:

import React from "react";

class FavoriteColor extends React.Component{

    constructor(){

        super();

        this.state = {color:"Red", model:"Logan"};

    }

    render(){

        return(

        <>

        <h1>This is {this.state.color} color and model is {this.state.model}.</h1>

        <button onClick={()=>{this.setState((previousState)=> {return {...previousState,color:"Green"}})}}>Change Color</button>

        </>

        )

    }

}

export default FavoriteColor;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import ChangeColor from './components/ChangeColor';

import FavoriteColor from './components/FavoriteColor';

import Scooter from './components/Scooter';

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

root.render(

  <React.StrictMode>

    <ChangeColor />

    <FavoriteColor />

    <Scooter/>

  </React.StrictMode>

);

Updating Arrays in State:

List.js:

import { useState } from "react";

function List(){

    const [list,setList] = useState([]);

    const [count,setCount] = useState(1);

    function addItem(){

        const item = 'Item'+count;

        setList((previousState)=>{return [...previousState,item]})

        setCount((previousCount)=>{return previousCount+1})

    }

    return (

        <>

        <h1>List</h1>

        <button onClick={addItem}>Add Item</button>

        <ul>

            {

            list.map((el,index)=><li key={index}>{el}</li>)

            }

        </ul>

        </>

    )

}

export default List;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import ChangeColor from './components/ChangeColor';

import FavoriteColor from './components/FavoriteColor';

import Scooter from './components/Scooter';

import List from './components/List';

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

root.render(

  <>

    <ChangeColor />

    <FavoriteColor />

    <Scooter/>

    <List/>

    </>

);

UseEffect:

Based on property changed then callback function is executed.

useEffect in React is a built-in Hook that allows you to perform "side effects" in functional components. Side effects are operations that affect something outside of the component's rendering logic, such as:

* **Data Fetching:** Making API calls to retrieve data from a server.
* **DOM Manipulation:** Directly interacting with the Document Object Model (e.g., changing the document title, adding/removing event listeners).
* **Subscriptions:** Setting up and cleaning up subscriptions to external services.
* **Timers:** Implementing setTimeout or setInterval.

useEffect takes two arguments:

* **A function (the "effect"):** This function contains the code for your side effect.
* **A dependency array (optional):** This array specifies the values that the effect depends on. The effect will re-run only when one of these dependencies changes.

Timer.js:

import { useEffect, useState } from "react";

function Timer(){

    const [count,setCount] = useState(1);

    useEffect(()=>{

        console.log("Screen Rendered");

        checkCount()

    },[count])

    function checkCount(){

        if(count>10){

            setCount(1);

        }

    }

    function updateTimer(){

        setCount((previousState)=>previousState+1)

    }

    return (

        <>

        <h1>I have rendered {count} times</h1>

        <button onClick={updateTimer}>Change Timer</button>

        </>

    )

}

export default Timer;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import ChangeColor from './components/ChangeColor';

import Timer from './components/Timer';

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

root.render(

  <>

    <ChangeColor />

    <Timer/>

    </>

);

Using useEffect in Different ways:

* + Not use like this callback function use in useEffect:
* import { useEffect, useState } from "react";
* function Timer(){
* const [count,setCount] = useState(1);
* useEffect(()=>{
* console.log("Screen Rendered");
* setTimeout(()=>{
* setCount((previousState)=>previousState+1)
* },1000)
* })
* function updateTimer(){
* setCount((previousState)=>previousState+1)
* }
* return (
* <>
* <h1>I have rendered {count} times</h1>
* <button onClick={updateTimer}>Change Timer</button>
* </>
* )
* }
* export default Timer;

React Forms:

Creating Form:

MyForm.js:

import { useState } from "react";

function MyForm(){

    const [name,setName] = useState('');

    console.log("Current name:",name);

    return(

        <>

        <form>

            <label>Enter your name:<input type="text" onChange={(e)=>{setName(e.target.value)}} /></label>

        </form>

        </>

    )

}

export default MyForm;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import MyForm from './components/MyForm';

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

root.render(

  <>

  <MyForm/>

  </>

);

Submitting Form and Handling Multiple Input Values:

MyForm.js:

import { useState } from "react";

function MyForm(){

    const [form,setForm] = useState({});

    function handleSubmit(e){

        e.preventDefault();

        console.log("Form Submitted");

        console.log("Current State: ",form);

    }

    return(

        <>

        <form onSubmit={handleSubmit}>

            <label>Enter your name:<input type="text" onChange={(e)=>{setForm((previousState)=>{return {...previousState,name:e.target.value}})}} /></label><br/>

            <label>Enter your age:<input type="text" onChange={(e)=>{setForm((previousState)=>{return{...previousState,age:e.target.value}})}}  /></label><br/>

            <label>Enter your email:<input type="email" onChange={(e)=>{setForm((previousState)=>{return{...previousState,email:e.target.value}})}}  /></label><br/>

            <input type="submit" value="Submit"/>

        </form>

        </>

    )

}

export default MyForm;

Simplify OnChange Handler:

import { useState } from "react";

function MyForm(){

    const [form,setForm] = useState({});

    function handleSubmit(e){

        e.preventDefault();

        console.log("Form Submitted");

        console.log("Current State: ",form);

    }

    function handleChange(e){

        setForm((previousState)=>{return {...previousState,[e.target.name]:e.target.value}})

    }

    return(

        <>

        <form onSubmit={handleSubmit}>

            <label>Enter your name:<input type="text" name="name" onChange={handleChange} /></label><br/>

            <label>Enter your age:<input type="text" name="age" onChange={handleChange}  /></label><br/>

            <label>Enter your email:<input type="email" name="email" onChange={handleChange}  /></label><br/>

            <input type="submit" value="Submit"/>

        </form>

        </>

    )

}

export default MyForm;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import MyForm from './components/MyForm';

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

root.render(

  <>

  <MyForm/>

  </>

);

Setting Initial Form Values and <select> tag in react:

In select tag, in option selected is not as select, based on state value take value for selected.

import { useState } from "react";

function MyForm(){

    const [form,setForm] = useState({phone:"+91",email:"@gmail.com",country:"India",name:"",age:""});

    function handleSubmit(e){

        e.preventDefault();

        console.log("Form Submitted");

        console.log("Current State: ",form);

        setForm({phone:"+91",email:"@gmail.com",country:"India",name:"",age:""})

    }

    function handleChange(e){

        setForm((previousState)=>{return {...previousState,[e.target.name]:e.target.value}})

    }

    return(

        <>

        <form onSubmit={handleSubmit}>

            <label>Enter your name:<input type="text" name="name" onChange={handleChange} value={form.name} /></label><br/>

            <label>Enter your age:<input type="text" name="age" onChange={handleChange} value={form.age}  /></label><br/>

            <label>Enter your email:<input type="email" name="email" onChange={handleChange} value={form.email}  /></label><br/>

            <label>Enter your phone:<input type="text" name="phone" onChange={handleChange} value={form.phone}  /></label><br/>

            <label>

                Select your Country:

                <select name="country" onChange={handleChange} value={form.country}>

                    <option value="">Select</option>

                    <option value="Australia">Australia</option>

                    <option value="India">India</option>

                    <option value="United States">United States</option>

                </select>

            </label><br/>

            <input type="submit" value="Submit"/>

        </form>

        </>

    )

}

export default MyForm;

Textarea in React:

Textarea is not like working in html, here can add name, value, onChange attributes can add and use it like self-closing tag.

MyForm.js:

import { useState } from "react";

function MyForm(){

    const [form,setForm] = useState({phone:"+91",email:"@gmail.com",country:"India",name:"",age:"",about:"I'm a Student"});

    function handleSubmit(e){

        e.preventDefault();

        console.log("Form Submitted");

        console.log("Current State: ",form);

        setForm({phone:"+91",email:"@gmail.com",country:"India",name:"",age:"",about:"I'm a Student"})

    }

    function handleChange(e){

        setForm((previousState)=>{return {...previousState,[e.target.name]:e.target.value}})

    }

    return(

        <>

        <form onSubmit={handleSubmit}>

            <label>Enter your name:<input type="text" name="name" onChange={handleChange} value={form.name} /></label><br/>

            <label>Enter your age:<input type="text" name="age" onChange={handleChange} value={form.age}  /></label><br/>

            <label>Enter your email:<input type="email" name="email" onChange={handleChange} value={form.email}  /></label><br/>

            <label>Enter your phone:<input type="text" name="phone" onChange={handleChange} value={form.phone}  /></label><br/>

            <label>

                Select your Country:

                <select name="country" onChange={handleChange} value={form.country}>

                    <option value="">Select</option>

                    <option value="Australia">Australia</option>

                    <option value="India">India</option>

                    <option value="United States">United States</option>

                </select>

            </label><br/>

            <textarea name="about" value={form.about} onChange={handleChange}/><br/>

            <input type="submit" value="Submit"/>

        </form>

        </>

    )

}

export default MyForm;

index.js:

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import MyForm from './components/MyForm';

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

root.render(

  <>

  <MyForm/>

  </>

);