**3.Important concepts of React. Js**

**a.Write a React program to implement a counter button using React use State hook**

**App.js**

Import ‘./App.css’;

import React, { useState } from "react";

function App() {

let[counter, setCounter]=useState(0);

let stock = 10;

return(

<div class=”wrapper’>

<button class=”plus’onClick = { () => {

If(counter<stock) {

setCount(counter+1)

}

}}>INCREAMENT </button>

<p class=”counter”> {counter} </p>

<button class= “minus” onClick = { () => {

If(counter>0) {

setCounter(counter-1);

}

}}> DECREAMENT</button>

</div>

);

}

Export default App;

**App.css**

.wrapper[

Display: flex;

Justify-content: center;

gap: 30px;

}

.wrapper button{

border: none;

outline: none;

padding: 10px 15px;

font-size: 20px;

border-radius: 5px;

color: #fff;

cursor: pointer;

}

.counter{

Font-size: 16px;

}

Button.plus{

Background-color: blue;

}

**b.Write a React program to fetch the data from an API using React use Effect hook**

**App.js**import React, { useState, useEffect } from "react";

import ProductData from "./ProductData";

const App = () => {

const [products, setProducts] = useState([]);

useEffect(() => {

constgetData = async () => {

try {

constdbData = await fetch("https://fakestoreapi.com/products");

const result = await dbData.json();

setProducts(result);

} catch (error) {

alert("Failed to fetch the data: " + error.message);

}

};

getData();

}, []);

constproductsInfo = products.map((product, index) => (

<ProductData key={index} product={product} />

));

return (

<div>

<h1>Making a GET request with an API</h1>

<div>{productsInfo}</div>

</div>

);

};

export default App;

**ProductData.js**import React from "react";

constProductData = (props) => {

const{ title, image, price, description, rating } = props.product;

return (

<div style={{ border: "1px solid gray", margin: "10px", padding: "10px" }}>

<img height={100} src={image} alt="product" />

<h3>Product: {title}</h3>

<h4>Price: ₹{price}</h4>

<h4>Rating: {rating?.rate}</h4>

<p>About Product: {description}</p>

<button>View More</button>

<button>Add To Cart</button>

</div>

);

};

export default ProductData;

**c. Write a React program with two react components sharing data using Props.**

**import React from "react";**

import React from "react";

// Way 1: Basic Props (without destructuring)

function Student1(props) {

return (

<div style={boxStyle}>

<h2>Way 1: Basic Props</h2>

<p>Name: {props.name}</p>

<p>Age: {props.age}</p>

</div>

);

}

// Way 2: Using Props Destructuring

function Student2({ name, age }) {

return (

<div style={boxStyle}>

<h2>Way 2: Props Destructuring</h2>

<p>Name: {name}</p>

<p>Age: {age}</p>

</div>

);

}

// Way 3: Passing an Object as Prop

function Student3({ details }) {

return (

<div style={boxStyle}>

<h2>Way 3: Object as Prop</h2>

<p>Name: {details.name}</p>

<p>Age: {details.age}</p>

</div>

);

}

// Way 4: Passing Function as Prop

function Student4({ name, greet }) {

return (

<div style={boxStyle}>

<h2>Way 4: Function as Prop</h2>

<p>Name: {name}</p>

<button onClick={greet}>Click to Greet</button>

</div>

);

}

// Way 5: Multiple Child Components Sharing Props

function Student5({ name, age }) {

return (

<div style={boxStyle}>

<h3>Student Info</h3>

<p>Name: {name}</p>

<p>Age: {age}</p>

</div>

);

}

function Course({ course, duration }) {

return (

<div style={boxStyle}>

<h3>Course Info</h3>

<p>Course: {course}</p>

<p>Duration: {duration}</p>

</div>

);

}

// Parent Component (App)

function App() {

conststudentData = { name: "Mounika", age: 21 };

constsayHello = () => {

alert("Hello from Parent Component!");

};

return (

<div style={{ textAlign: "center", fontFamily: "Arial" }}>

<h1>React Props - All Ways in One Program</h1>

{/\* Way 1 \*/}

<Student1 name="Anjali" age={21} />

{/\* Way 2 \*/}

<Student2 name="Bhuvana" age={23} />

{/\* Way 3 \*/}

<Student3 details={studentData} />

{/\* Way 4 \*/}

<Student4 name="Chaitrika" greet={sayHello} />

{/\* Way 5 \*/}

<div style={{ display: "flex", justifyContent: "center", gap: "20px" }}>

<Student5 name="Dhamini" age={26} />

<Course course="React Development" duration="3 Months" />

</div>

</div>

);

}

// Small style for better view

constboxStyle = {

border: "2px solid #ccc",

padding: "10px",

margin: "10px",

borderRadius: "8px",

width: "250px",

display: "inline-block",

verticalAlign: "top",

};

export default App;

**d. Write a React program to implement the forms in react**

**App.js**

import React, { useState } from "react";

function App() {

const [formData, setFormData] = useState({

name: "",

email: "",

password: "",

});

// Handle input changes

consthandleChange = (e) => {

setFormData({ ...formData, [e.target.name]: e.target.value });

};

// Handle form submission

consthandleSubmit = (e) => {

e.preventDefault();

alert(`Form Submitted!\nName: ${formData.name}\nEmail: ${formData.email}`);

};

return (

<div style={{ margin: "20px", textAlign: "center" }}>

<h2>React Form Example</h2>

<form onSubmit={handleSubmit} style={{ display: "inline-block", textAlign: "left" }}>

<div style={{ marginBottom: "10px" }}>

<label>Name: </label><br />

<input

type="text"

name="name"

value={formData.name}

onChange={handleChange}

required

/>

</div>

<div style={{ marginBottom: "10px" }}>

<label>Email: </label><br />

<input

type="email"

name="email"

value={formData.email}

onChange={handleChange}

required

/>

</div>

<div style={{ marginBottom: "10px" }}>

<label>Password: </label><br />

<input

type="password"

name="password"

value={formData.password}

onChange={handleChange}

required

/>

</div>

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

</form>

<h3>Form Data Preview:</h3>

<p><strong>Name:</strong> {formData.name}</p>

<p><strong>Email:</strong> {formData.email}</p>

</div>

);

}

export default App;

**e. Write a React program to implement the iterative rendering using map() function.**

**ExampleMapRenderings.js**

import React from 'react';

import './ExampleMapRenderings.css'; // Import the CSS file

// Child component for demonstration

function FruitItem({ name }) {

return <li className="fruit-item">{name}</li>;

}

function ExampleMapRenderings() {

// Data arrays

const fruits = ['Apple', 'Banana', 'Cherry'];

const items = [

{ id: 1, name: 'Carrot' },

{ id: 2, name: 'Tomato' },

{ id: 3, name: 'Brinjal' }

];

consttodos = [

{ id: 1, text: 'Learn React', completed: true },

{ id: 2, text: 'Try map()', completed: false }

];

return (

<div className="container">

{/\* 1. Basic Array of Strings \*/}

<h2 className="section-title">Basic Array of Strings</h2>

<ulclassName="list">

{fruits.map((fruit, index) => (

<li key={index} className="list-item">{fruit}</li>

))}

</ul>

{/\* 2. Array of Objects \*/}

<h2 className="section-title">Array of Objects</h2>

<ulclassName="list">

{items.map(item => (

<li key={item.id} className="list-item">{item.name}</li>

))}

</ul>

{/\* 3. Child Component Rendering \*/}

<h2 className="section-title">Child Component Rendering</h2>

<ulclassName="list">

{fruits.map((fruit, index) => (

<FruitItem key={index} name={fruit} />

))}

</ul>

{/\* 4. Conditional Rendering \*/}

<h2 className="section-title">Conditional Rendering</h2>

<ulclassName="list">

{todos.map(todo => (

<li key={todo.id} className="list-item">

{todo.completed ? <s>{todo.text}</s> : todo.text}

</li>

))}

</ul>

{/\* 5. Event Handling during map \*/}

<h2 className="section-title">Event Handling</h2>

<ulclassName="list">

{fruits.map((fruit, index) => (

<li

key={index}

className="list-item clickable"

onClick={() => alert(`Clicked: ${fruit}`)}

>

{fruit}

</li>

))}

</ul>

</div>

);

}

export default ExampleMapRenderings;

**ExampleMapRenderings.css**

.container {

font-family: Arial, sans-serif;

max-width: 600px;

margin: 20px auto;

padding: 15px;

background-color: #f9f9f9;

border-radius: 8px;

}

.section-title {

color: #2c3e50;

border-bottom: 2px solid #2980b9;

padding-bottom: 6px;

margin-top: 30px;

}

.list {

list-style-type: disc;

padding-left: 20px;

margin-top: 10px;

}

.list-item {

font-size: 16px;

margin: 6px 0;

color: #34495e;

}

.fruit-item {

color: #27ae60;

font-weight: bold;

}

.clickable {

cursor: pointer;

transition: background-color 0.2s ease;

}

.clickable:hover {

background-color: #dff0ea;

}