**Experiment-8: Form in react.**

**Task 1: Create a form and log the input:**

App.js

**import { useState } from 'react';**

**function MyForm() {**

**const [name, setName] = useState("");**

**const handleSubmit = (event) => {**

**event.preventDefault();**

**alert(`The name you entered was: ${name}`)**

**console.log(`name: ${name}`)**

**}**

**return (**

**<form onSubmit={handleSubmit}>**

**<label>Enter your name:**

**<input**

**type="text"**

**value={name}**

**onChange={(e) => setName(e.target.value)}**

**/>**

**</label>**

**<input type="submit" />**

**</form>**

**)}**

**export default MyForm**

**Task 2: Handle dropdowns and checkboxes**

**import { useState } from "react";**

**export default function Dropdown() {**

**const [selectedOption, setSelectedOption] = useState("option1");**

**const handleDropdownChange = (event) => {**

**setSelectedOption(event.target.value);**

**};**

**const handleSubmit = (event) => {**

**event.preventDefault();**

**alert(`The OPTION you entered was: ${selectedOption}`)**

**console.log(`OPTION: ${selectedOption}`)**

**}**

**return (**

**<div>**

**<form onSubmit={handleSubmit}>**

**<label>**

**Select an option:**

**<select value={selectedOption} onChange={handleDropdownChange}>**

**<option value="option1">Option 1</option>**

**<option value="option2">Option 2</option>**

**<option value="option3">Option 3</option>**

**</select>**

**<input type="submit" />**

**</label>**

**<p>Selected option: {selectedOption}</p>**

**</form>**

**</div>**

**);**

**}**

**Task 3: Multiple of fields.**

**import { useState } from "react";**

**export default function Multiple() {**

**const [formData, setFormData] = useState({name: "",email: "",message: ""});**

**const handleChange = (event) => {**

**const { name, value } = event.target;**

**setFormData((prevFormData) => ({ ...prevFormData, [name]: value }));**

**};**

**const handleSubmit = (event) => {**

**event.preventDefault();**

**alert(`Name: ${formData.name}, Email: ${formData.email}, Message: ${formData.message}`**

**);**

**console.log(`Name: ${formData.name}, Email: ${formData.email}, Message: ${formData.message} `)**

**};**

**return (**

**<form onSubmit={handleSubmit}>**

**<label>Name:</label>**

**<input type="text" id="name" name="name" value={formData.name} onChange={handleChange}/>**

**<label >Email:</label>**

**<input type="email" id="email" name="email" value={formData.email} onChange={handleChange}/>**

**<label >Message:</label>**

**<textarea id="message" name="message" value={formData.message} onChange={handleChange}/>**

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

**</form>**

**);**

**}**

**Task 4: Validation**

**import { useState } from "react";**

**export default function Multiple() {**

**const [formData, setFormData] = useState({name: "",email: "",message: ""});**

**const [inputError, setInputError] = useState(null);**

**const handleChange = (event) => {**

**const { name, value } = event.target;**

**setFormData((prevFormData) => ({ ...prevFormData, [name]: value }));**

**if (formData.name.length <= 5) {**

**setInputError('Input must be at least 5 characters');**

**} else {**

**setInputError(null);**

**}**

**};**

**const handleSubmit = (event) => {**

**if (formData.name.length >= 5) {**

**event.preventDefault();**

**alert(`Name: ${formData.name}, Email: ${formData.email}, Message: ${formData.message}`**

**);**

**console.log(`Name: ${formData.name}, Email: ${formData.email}, Message: ${formData.message} `)**

**} else {**

**setInputError('Input must be at least 5 characters');**

**}**

**};**

**return (**

**<form onSubmit={handleSubmit}>**

**<label>Name:</label>**

**<input type="text" id="name" name="name" value={formData.name} onChange={handleChange}/>**

**<label >Email:</label>**

**<input type="email" id="email" name="email" value={formData.email} onChange={handleChange}/>**

**<label >Message:</label>**

**<textarea id="message" name="message" value={formData.message} onChange={handleChange}/>**

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

**{inputError && <div style={{ color: 'red' }}>{inputError}</div>}**

**</form>**

**);**

**}**

**Task 5: using Useref().**

**import { useRef } from "react";**

**export default function Uncontrolled() {**

**const selectRef = useRef(null);**

**const checkboxRef = useRef(null);**

**const inputRef = useRef(null);**

**function handleSubmit(event) {**

**event.preventDefault();**

**console.log("Input value:", inputRef.current.value);**

**console.log("Select value:", selectRef.current.value);**

**console.log("Checkbox value:", checkboxRef.current.checked);**

**}**

**return (**

**<form onSubmit={handleSubmit}>**

**<label>**

**<p>Name:</p>**

**<input ref={inputRef} type="text" />**

**</label>**

**<label>**

**<p>Favorite color:</p>**

**<select ref={selectRef}>**

**<option value="red">Red</option>**

**<option value="green">Green</option>**

**<option value="blue">Blue</option>**

**</select>**

**</label>**

**<label>**

**Do you like React?**

**<input type="checkbox" ref={checkboxRef} />**

**</label>**

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

**</form>**

**);**

**}**