Name :- Manav Pahilwani
Roll No :- 37
Class :- D11AD

Experiment 4d

Aim :- To create Controlled and Uncontrolled forms using react.js

Theory :-

In a controlled component form data is handled by a react component. The alternative is controlled components, where form data is handled by the DOM itself.

To write uncontrolled components, instead of writing an event handler for every handler for every state update, you can use a ref to get form values from the DOM.

Hook is a special function that lets you “hook into” react features. For eg useState is a hock that lets you add React state to function components
useState
It returns a pair of values:- the current state and a function that updates it. This is similar to this.state.count and this.setState in a class except you get them in a pair
useRet

useRet hook allows you to persist values between renders. It can be used to store a mutable value that does not cause a rereader when updated. It can be used to access a DOM element directly.

Conclusion :- Thus we have learned and successfully implemented controlled and uncontrolled form using useState and useRet hook.

Code :-

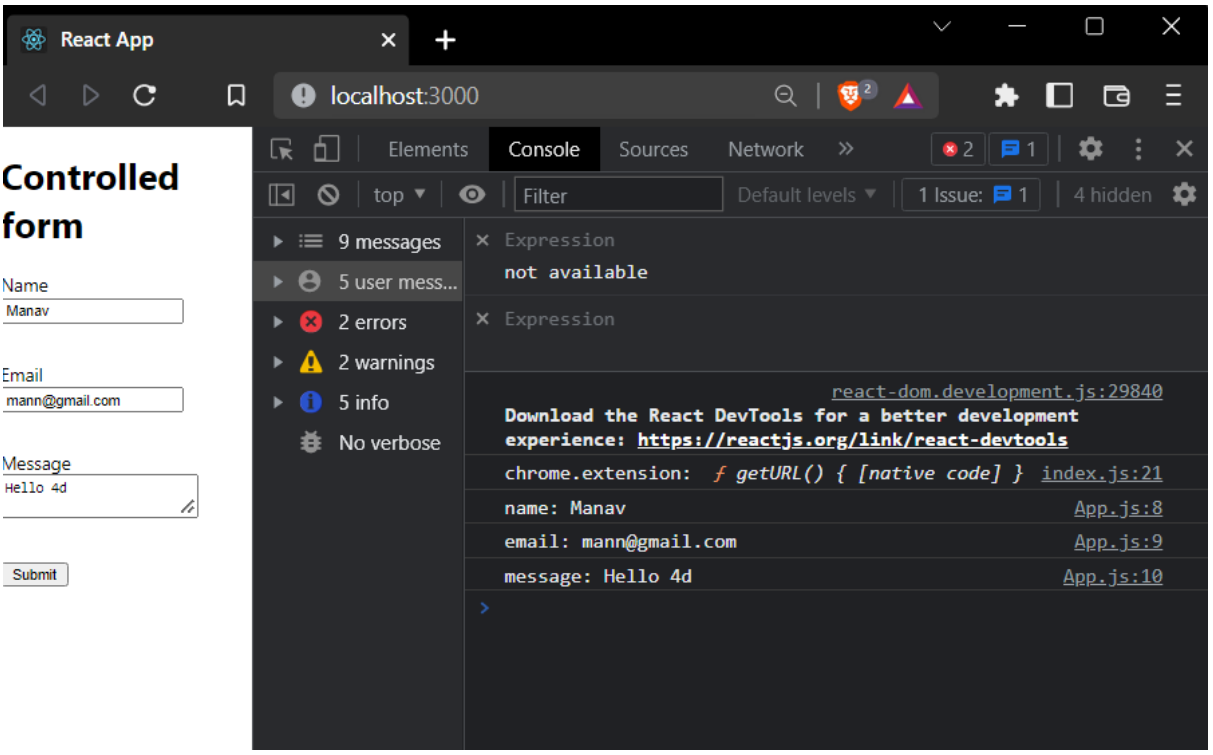
```
import {useState} from 'react';
const Form_1 = () => {
  const [name, setname] = useState('');
  const [email, setEmail] = useState('');
  const [message, setMessage] = useState('');
  function handleSubmit(event) {
    event.preventDefault();
    console.log('name:', name);
    console.log('email:', email);
    console.log('message:', message);
  }
  return (
    <div className="form1" style={{height: "73vh"}}>
      <h1>Controlled form</h1>
      <form onSubmit={handleSubmit}>
        <div style={{ marginBottom: "25px"}}>
          <label htmlFor="name" style={{display:"block"}}>Name</label>
          <input id="name" type="text" value={name} onChange={(e) =>
setname(e.target.value)}/>
        </div>
        <div style={{ marginBottom: "25px"}}>
          <label htmlFor="email"
style={{display:"block"}}>Email</label>
          <input id="email" type="email" value={email}onChange={(e) =>
setEmail(e.target.value)}/>
        </div>
        <div style={{ marginBottom: "25px"}}>
          <label htmlFor="message"
style={{display:"block"}}>Message</label>
            <textarea id="message" value={message} onChange={(e) =>
setMessage(e.target.value)}/>
        </div>
        <button type="submit">Submit</button>
      </form>
    </div>
  );
}
  export default Form_1;
```

2nd :-

```
import {useRef} from 'react'
  const Form_2 = () => {
  const nameRef = useRef();
  const emailRef = useRef();
  const messageRef = useRef();
  function handleSubmit(event) {
    event.preventDefault();
    console.log('name:', nameRef.current.value);
    console.log('email:', emailRef.current.value);
    console.log('message:', messageRef.current.value);
  }
  return (
    <div clasName="form2" style={{height: "73vh"}}>
      <h1>Uncontrolled form</h1>
      <form onSubmit={handleSubmit}>
    <div style={{ marginBottom: "25px"}}>
    <label htmlFor="name" style={{display: "block"}}>Name</label>
    <input id="name" type="text" ref={nameRef}/>
    </div>
    <div style={{ marginBottom: "25px"}}>
    <label htmlFor="email" style={{display:"block"}}>Email</label>
    <input id="email" type="email" ref={emailRef}/>
    </div>
    <div style={{ marginBottom: "25px"}}>
    <label htmlFor="message" style={{display: "block"}}>Message</label>
    <textarea id="message" ref={messageRef}/>
    </div>
    <button type="submit">Submit</button>
    </form>
    </div>
  );
}
export default Form_2;
```

Output :-

Controlled Form using useState Hook



Uncontrolled form using useRef Hook