



**Department of Artificial Intelligence and Machine Learning**

**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

**Experiment 3**

**Name: Shubham Mourya**

**SAP ID: 60017230110**

Name: Shubham Mourya  
AIML  
60017230110 A065

Experiment No. 3

Aim - Implementing Forms, Events, Reactives, Props, Keys

Theory:-

\* Forms in React:

- Forms are crucial in web applications for user interaction & data collection.
- React manages form inputs through controlled components, where form data is handled by the React component state.

```
import React, {useState} from 'React';
```

```
function SimpleForm() {
```

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

```
  const handleChange = (e) => {
```

```
    setName(e.target.value);
```

```
  };
```

```
  const handleSubmit = (e) => {
```

```
    e.preventDefault();
```

```
    alert('Form submitted with name: ' + name);
```

```
  }
```

\* Routing in React

- React Router is a standard library for routing in React.
- It enables navigation between different components, maintaining a single-page application feel.

**Department of Artificial Intelligence and Machine Learning****B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

- The 'App Router' example shows how to set up routing with 'BrowserRouter', 'Route', 'Switch', & 'Link' components to navigate.

```
import { BrowserRouter as Router, Route, Switch, Link } from 'react-router-dom';  
function AppRouter() {  
  return (  
    <Router>  
      <nav> <Link to = "/"> Home </Link>  
    </nav>  
    <Switch>  
      <Route exact path = "/" component = { Home } />  
    </Switch>  
  </Router>);  
}
```

export default AppRouter.

### \* Using Refs in React

- Refs provide a way to access DOM nodes directly within React components.
- Used for managing focus, text selection, or triggering animations.

```
function FocusInput() {  
  const inputRef = useRef(null);
```

### \* Using key in Lists

- Keys are important in React for identifying which items in a list have changed, been added, or removed.
- Keys help optimize rendering by providing a unique identifier.

• Conclusion:- This experiment helped us to implement Forms, Routers, Refs & Keys in React.



**Department of Artificial Intelligence and Machine Learning**  
**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

<b>Date:</b>	<b>13/8/2024</b>
<b>Aim</b>	<b>Implementing Forms, Events, Routers, Refs, Keys</b>
<b>Software</b>	
<b>Pre-requisite</b>	Active internet connection
<b>Theory</b>	<div><b>1. Forms in React</b>  Forms in React are usually handled as "controlled components," meaning that form elements are controlled by the React component's state rather than by the DOM itself. This approach allows React to manage the form data and ensures that the form's behavior is predictable.  <b>Example: Controlled Form</b></div> <pre>import React, { useState } from 'react';  function SimpleForm() {   const [name, setName] = useState("");    const handleChange = (e) =&gt; {     setName(e.target.value);   };    const handleSubmit = (e) =&gt; {     e.preventDefault();     alert('Form submitted with name: ' + name);   };    return (     &lt;form onSubmit={handleSubmit}&gt;       &lt;input         type="text"         value={name}         onChange={handleChange}         placeholder="Enter your name"       /&gt;       &lt;button type="submit"&gt;Submit&lt;/button&gt;     &lt;/form&gt;   ); }  export default SimpleForm;</pre>



**Department of Artificial Intelligence and Machine Learning**  
**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

**Explanation:**

- **State Management:** The form data is stored in the component's state using the ``useState`` hook. The state is an object with ``name``, ``email``, and ``message`` fields.
- **Event Handling:** The ``handleChange`` function updates the state whenever the user types in any of the form fields. The ``handleSubmit`` function handles the form submission and prevents the default form behavior using ``e.preventDefault()``.
- **Controlled Components:** Each form field's value is controlled by React through the ``value`` attribute, which is set to the corresponding state value. This ensures that React is in charge of the form's data.

**2. Events in React**

React uses a synthetic event system, which is a wrapper around the browser's native event system. This system ensures that events behave consistently across different browsers. Event handlers in React are written as functions and are passed to elements as props.

**Example: Handling Click Events**

```
import React from 'react';

function ClickButton() {
  const handleClick = () => {
    alert('Button clicked!');
  };

  return (
    <button onClick={handleClick}>Click Me</button>
  );
}

export default ClickButton;
```

**Explanation:**

- **Event Handler:** The ``handleClick`` function is an event handler that is triggered when the button is clicked.
- **Event Prop:** The ``onClick`` prop is used to attach the ``handleClick`` function to the button. When the button is clicked, the function is called, and an alert is displayed.
- **Synthetic Events:** React's synthetic event system ensures that the ``onClick`` event works consistently across different browsers.





**Department of Artificial Intelligence and Machine Learning**

**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

### 3. Routers in React

React Router is a popular library used to manage routing in React applications. It allows you to define routes in your app and handle navigation between different components based on the URL.

#### Example: Basic Routing

```
import React from 'react';
import { BrowserRouter as Router, Route, Switch, Link } from 'react-router-dom';

function Home() {
  return <h2>Home Page</h2>;
}

function About() {
  return <h2>About Page</h2>;
}

function Contact() {
  return <h2>Contact Page</h2>;
}

function AppRouter() {
  return (
    <Router>
      <nav>
        <Link to="/">Home</Link> |
        <Link to="/about">About</Link> |
        <Link to="/contact">Contact</Link>
      </nav>

      <Switch>
        <Route exact path="/" component={Home} />
        <Route path="/about" component={About} />
        <Route path="/contact" component={Contact} />
      </Switch>
    </Router>
  );
}

export default AppRouter;
```



**Department of Artificial Intelligence and Machine Learning**  
**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

**Explanation:**

- **Router Component:** The `Router` component wraps your entire application. It listens to changes in the browser's URL and renders the corresponding components.
- **Route Component:** The `Route` component is used to define different routes in your app. The `path` prop specifies the URL path, and the `component` prop specifies the component to render for that path.
- **Switch Component:** The `Switch` component ensures that only one `Route` is rendered at a time. It renders the first route that matches the current URL.
- **Link Component:** The `Link` component is used to create navigational links. Clicking a `Link` changes the URL and renders the corresponding component without reloading the page.

**4. Refs in React**

Refs provide a way to access DOM elements or React elements created in the render method. They are often used when you need to interact with the DOM directly, such as focusing an input field or managing animations.

**Example: Using Refs**

```
import React, { useRef } from 'react';

function FocusInput() {
  const inputRef = useRef(null);

  const handleFocus = () => {
    inputRef.current.focus();
  };

  return (
    <div>
      <input ref={inputRef} type="text" placeholder="Focus on me" />
      <button onClick={handleFocus}>Focus Input</button>
    </div>
  );
}

export default FocusInput;
```



**Department of Artificial Intelligence and Machine Learning**

**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

**Explanation:**

- **useRef Hook:** The `useRef` hook is used to create a reference to a DOM element. In this case, it creates a reference to the input field.
- **Accessing the DOM:** The `inputRef.current` gives you access to the DOM node of the input field. You can then call methods like `focus()` on this DOM node.
- **Event Handler:** The `handleFocus` function is an event handler that focuses the input field when the button is clicked.

**5. Keys in React**

Keys help React identify which items have changed, been added, or removed. They are important when rendering lists of elements because they help React optimize the rendering process by efficiently updating the DOM.

**Example: Using Keys in a List**

```
import React from 'react';

function FruitList() {
  const fruits = ['Apple', 'Banana', 'Cherry'];

  return (
    <ul>
      {fruits.map((fruit, index) => (
        <li key={index}>{fruit}</li>
      ))}
    </ul>
  );
}

export default FruitList;
```

**Explanation:**

- **Unique Keys:** The `key` prop is used to assign a unique identifier to each list item. This helps React differentiate between items when rendering and updating the list.
- **Array Index:** In this example, the array index (`index`) is used as the key. While this works for simple lists, it's better to use a unique ID when possible, especially if the list can change or be reordered.
- **Efficient Rendering:** Keys allow React to efficiently re-render only the items that have changed, instead of re-rendering the entire list.



**Department of Artificial Intelligence and Machine Learning**

**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

	<p><b>Conclusion</b></p> <ul style="list-style-type: none"><li>• <b>Forms:</b> Managed using controlled components where React controls the form's data through state.</li><li>• <b>Events:</b> Handled using React's synthetic event system, ensuring consistent behavior across browsers.</li><li>• <b>Routers:</b> Managed using React Router to handle navigation and route changes in a single-page application.</li><li>• <b>Refs:</b> Used to directly access and interact with DOM elements, such as focusing an input field.</li><li>• <b>Keys:</b> Provide unique identifiers for list items, helping React optimize the rendering process.</li></ul>
Code	<pre>import React, { useState, useRef } from 'react'; import { useNavigate } from 'react-router-dom';  const CustomerServiceForm = () =&gt; {   const [formData, setFormData] = useState({     name: "",     email: "",     question: "",     topic: 'General Inquiry'   });    const navigate = useNavigate();   const nameInputRef = useRef(null); // Create a ref for the name input    const topics = [     'General Inquiry',     'Order Status',     'Product Inquiry',     'Feedback'   ];    const handleChange = (e) =&gt; {     const { name, value } = e.target;     setFormData((prevFormData) =&gt; ({       ...prevFormData,       [name]: value     }));   };    const handleSubmit = (e) =&gt; {     e.preventDefault();     navigate('/thank-you', { state: { ...formData } });     nameInputRef.current.focus(); // Focus on the "Name" input after     submission   }; }</pre>





**Department of Artificial Intelligence and Machine Learning**  
**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

```
return (  
  <div className='max-w-md mx-auto p-6 bg-white shadow-md  
rounded-md'>  
    <h2 className='text-2xl font-bold mb-4'>Customer Service  
Form</h2>  
  
    <form onSubmit={handleSubmit} className='space-y-4'>  
      <div>  
        <label className='block text-sm font-medium'>Name</label>  
        <input  
          type='text'  
          name='name'  
          value={formData.name}  
          onChange={handleChange}  
          ref={nameInputRef} // Attach the ref to the input  
          className="w-full px-3 py-2 border border-gray-300  
rounded-md"  
          required  
        />  
      </div>  
  
      <div>  
        <label className="block text-sm font-medium">Email</label>  
        <input  
          type='email'  
          name='email'  
          value={formData.email}  
          onChange={handleChange}  
          className='w-full px-3 py-2 border border-gray-300  
rounded-md'  
        />  
      </div>  
  
      <div>  
        <label className="block text-sm font-  
medium">Question</label>  
        <textarea  
          name="question"  
          value={formData.question}  
          onChange={handleChange}  
          className="w-full px-3 py-2 border border-gray-300  
rounded-md"  
          required  
        ></textarea>  
      </div>  
  
      <div>
```



**Department of Artificial Intelligence and Machine Learning**

**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

```
<label className="block text-sm font-medium">Topic</label>
<select
  name='topic'
  value={formData.topic}
  onChange={handleChange}
  className='w-full px-3 py-2 border border-gray-300
rounded-md'
>
  {topics.map((topic, index) => (
    <option key={index} value={topic}>
      {topic}
    </option>
  ))}
</select>
</div>

<button
  type='submit'
  className='w-full bg-blue-500 text-white px-4 py-2 rounded-
md hover:bg-blue-600'
>
  Submit
</button>
</form>
</div>
);
};

export default CustomerServiceForm;

import CustomerServiceForm from './CustomerServiceForm';
import ThankYou from './ThankYou';
import { Routes, Route } from 'react-router-dom';

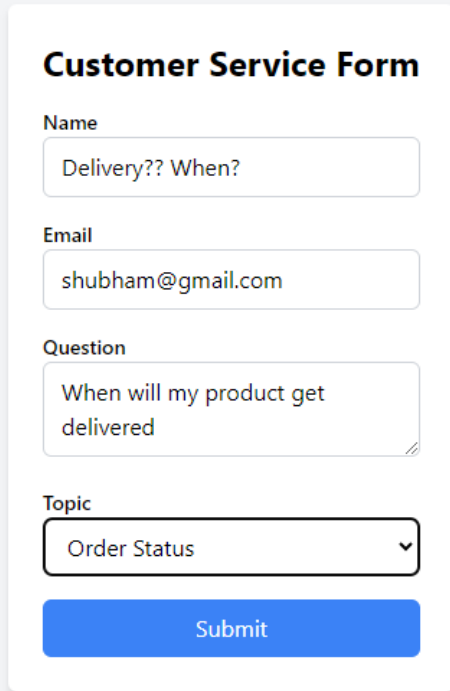
function App() {
  return (
    <div className="min-h-screen bg-gray-100 flex items-center justify-
center">
      <Routes>
        <Route path="/" element={ <CustomerServiceForm /> } />
        <Route path="/thank-you" element={ <ThankYou /> } />
      </Routes>
    </div>
  );
}

export default App;
```



**Department of Artificial Intelligence and Machine Learning**

**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

	<pre>import React from 'react'; import { useLocation } from 'react-router-dom';  const ThankYou = () =&gt; {   const { state } = useLocation();    return (     &lt;div className="max-w-md mx-auto p-6 bg-white shadow-md rounded- md text-center"&gt;       &lt;h2 className="text-3xl font-bold mb-4"&gt;Thank You!&lt;/h2&gt;       &lt;p className="text-lg mb-6"&gt;We have received your message. Here are the details you provided:&lt;/p&gt;       &lt;div className="text-left"&gt;         &lt;p&gt;&lt;strong&gt;Name:&lt;/strong&gt; {state.name}&lt;/p&gt;         &lt;p&gt;&lt;strong&gt;Email:&lt;/strong&gt; {state.email}&lt;/p&gt;         &lt;p&gt;&lt;strong&gt;Question:&lt;/strong&gt; {state.question}&lt;/p&gt;         &lt;p&gt;&lt;strong&gt;Topic:&lt;/strong&gt; {state.topic}&lt;/p&gt;       &lt;/div&gt;       &lt;p className="mt-6"&gt;We will get back to you shortly.&lt;/p&gt;     &lt;/div&gt;   ); };  export default ThankYou;</pre>
Result	



**Department of Artificial Intelligence and Machine Learning**  
**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

	<div><h2>Thank You!</h2><p>We have received your message. Here are the details you provided:</p><p><b>Name:</b> Delivery?? When? <b>Email:</b> shubham@gmail.com <b>Question:</b> When will my product get delivered <b>Topic:</b> Order Status</p><p>We will get back to you shortly.</p></div>
Code	<pre>import { useState, useCallback, useEffect, useRef } from 'react';  const PasswordGenerator = () =&gt; {   const [length, setLength] = useState(8);   const [numberAllowed, setNumberAllowed] = useState(false);   const [charAllowed, setCharAllowed] = useState(false);   const [password, setPassword] = useState("");    const passwordRef = useRef(null);    const passwordGenerator = useCallback(() =&gt; {     let pass = "";     let str =       "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";     if (numberAllowed) str += "0123456789";     if (charAllowed) str += "!@#\$%^&amp;*-_+={}~`";      for (let i = 0; i &lt; length; i++) {       let char = Math.floor(Math.random() * str.length);       pass += str.charAt(char);     }      setPassword(pass);   }, [length, numberAllowed, charAllowed]);    const copyPasswordToClipboard = useCallback(() =&gt; {     passwordRef.current?.select();     passwordRef.current?.setSelectionRange(0, 99999);   }, [passwordRef]);</pre>



**Department of Artificial Intelligence and Machine Learning**

**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

```
window.navigator.clipboard.writeText(password);
}, [password]);

useEffect(() => {
  passwordGenerator();
}, [length, numberAllowed, charAllowed, passwordGenerator]);

return (
  <div className="w-full max-w-md mx-auto shadow-lg rounded-lg px-4
py-5 my-8 bg-gray-800 text-orange-500">
    <h1 className="text-white text-2xl font-bold text-center mb-
4">Password Generator</h1>
    <div className="flex mb-4">
      <input
        type="text"
        value={password}
        className="outline-none w-full py-2 px-3 rounded-l-lg"
        placeholder="Password"
        readOnly
        ref={passwordRef}
      />
      <button
        onClick={copyPasswordToClipboard}
        className="bg-blue-700 text-white px-4 py-2 rounded-r-lg hover:bg-
blue-800"
      >
        Copy
      </button>
    </div>
    <div className="flex flex-col gap-y-4">
      <div className="flex items-center gap-x-2">
        <input
          type="range"
          min={6}
          max={100}
          value={length}
          className="cursor-pointer"
          onChange={(e) => setLength(parseInt(e.target.value))}
        />
        <span className="text-gray-300">Length: {length}</span>
      </div>
      <div className="flex items-center gap-x-2">
        <input
          type="checkbox"
          checked={numberAllowed}
        />
      </div>
    </div>
  </div>
```





**Department of Artificial Intelligence and Machine Learning**

**B.Tech. Sem: V Subject: Full Stack Development Laboratory (DJS22AML504)**

	<pre>id="numberInput" onChange={() =&gt; setNumberAllowed(prev =&gt; !prev)} /&gt; &lt;label htmlFor="numberInput" className="text-gray- 300"&gt;Numbers&lt;/label&gt; &lt;/div&gt; &lt;div className="flex items-center gap-x-2"&gt; &lt;input type="checkbox" checked={charAllowed} id="characterInput" onChange={() =&gt; setCharAllowed(prev =&gt; !prev)} /&gt; &lt;label htmlFor="characterInput" className="text-gray- 300"&gt;Characters&lt;/label&gt; &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; ); };  export default PasswordGenerator;</pre>
	<div><div>Home About Contact</div><div>Welcome to the Password Generator App</div><div><div><div>Password Generator</div><div>&amp;\$A{7RkbpG7D} Copy</div><div>Length: 12</div><div><input checked="" type="checkbox"/> Numbers</div><div><input checked="" type="checkbox"/> Characters</div></div></div></div>
Conclusion	<p>This experiment demonstrated React's core features: form handling, event management, routing with React Router, and using refs and keys. We built a simple form, a customer service form with dynamic topics, and navigated between pages using React Router. These elements highlight React's ability to create interactive and dynamic web applications efficiently.</p>