

# Chapter 14: Forms and Validation

## 1 Introduction

Is chapter mein humne React mein forms aur validation seekhi, jo **EduCampusHustle** MVP ke liye zaroori hain. Humne controlled components, Formik, aur Yup use kiya.

## 2 14.1 Basics of Forms in React

### 2.1 Controlled vs Uncontrolled Components

- **Controlled:** State se manage, onChange se update.
- **Uncontrolled:** DOM se direct access.

### 2.2 Code Example: Controlled Component

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Controlled Form Example</title>
6   <script
7     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js"
8   </script>
9   <script
10    src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.development.js"
11  </script>
12   <script
13    src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js"
14  </script>
15   <script src="https://cdn.tailwindcss.com"></script>
16 </head>
17 <body>
18   <div id="root"></div>
19   <script type="text/babel">
20     function LoginForm() {
21       const [formData, setFormData] = React.useState({
22         email: '',
23         password: ''
24       });
25
26       const handleChange = (e) => {
27         const { name, value } = e.target;
28         setFormData((prevData) => ({
29           ...prevData,
30           [name]: value
31         }));
32       };
33
34       const handleSubmit = (e) => {
35         e.preventDefault();
36         alert('Logged in with Email: ${formData.email},
37           Password: ${formData.password}');
```

```

31     };
32
33     return (
34         <div className="max-w-md mx-auto mt-10 p-6 bg-white
35             rounded-lg shadow-md">
36             <h2 className="text-2xl font-bold text-blue-600
37                 mb-4">Login</h2>
38             <form onSubmit={handleSubmit}>
39                 <div className="mb-4">
40                     <label className="block
41                         text-gray-700">Email</label>
42                     <input
43                         type="email"
44                         name="email"
45                         value={formData.email}
46                         onChange={handleChange}
47                         className="w-full p-2 border rounded"
48                         placeholder="Enter your email"
49                     />
50                 </div>
51                 <div className="mb-4">
52                     <label className="block
53                         text-gray-700">Password</label>
54                     <input
55                         type="password"
56                         name="password"
57                         value={formData.password}
58                         onChange={handleChange}
59                         className="w-full p-2 border rounded"
60                         placeholder="Enter your password"
61                     />
62                 </div>
63                 <button
64                     type="submit"
65                     className="w-full bg-blue-500 text-white p-2
66                         rounded hover:bg-blue-600"
67                 >
68                     Login
69                 </button>
70             </form>
71         </div>
72     );
73 }
74
75 const root =
76     ReactDOM.createRoot(document.getElementById('root'));
77 root.render(<LoginForm />);
78 </script>
79 </body>
80 </html>

```

## 2.3 Output

- Centered box, "Login" heading (blue).
- Email aur Password inputs.
- "Login" button (blue, hover pe darker).
- **Behavior:** Submit pe alert with data.

## 3 14.2 Form Validation with Formik and Yup

### 3.1 Why Formik and Yup?

- **Formik:** Form state aur submission.
- **Yup:** Validation rules.

### 3.2 Code Example: Login with Validation

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Login with Formik and Yup</title>
6   <script
7     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js"
8   </script>
9   <script
10    src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.development.js"
11  </script>
12   <script
13    src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js"
14  </script>
15   <script
16    src="https://cdn.jsdelivr.net/npm/formik@2.2.9/dist/formik.umd.min.js"
17  </script>
18   <script src="https://cdn.tailwindcss.com"></script>
19 </head>
20 <body>
21   <div id="root"></div>
22   <script type="text/babel">
23     const validationSchema = Yup.object({
24       email: Yup.string().email('Invalid email').required('Email
is required'),
25       password: Yup.string().min(6, 'Password must be at least 6
characters').required('Password is required')
26     });
27
28     function LoginForm() {
29       return (
30         <div className="max-w-md mx-auto mt-10 p-6 bg-white
rounded-lg shadow-md">
31           <h2 className="text-2xl font-bold text-blue-600
mb-4">Login</h2>
```

```

25 <Formik
26   initialValues={{ email: '', password: '' }}
27   validationSchema={validationSchema}
28   onSubmit={(values) => {
29     alert('Logged in with Email: ${values.email},
30       Password: ${values.password}');
31   }}
32 >
33   ({ handleSubmit, handleChange, values, errors,
34     touched }) => (
35     <form onSubmit={handleSubmit}>
36       <div className="mb-4">
37         <label className="block
38           text-gray-700">Email</label>
39         <input
40           type="email"
41           name="email"
42           value={values.email}
43           onChange={handleChange}
44           className={`w-full p-2 border rounded
45             ${errors.email && touched.email ?
46               'border-red-500' : 'border-gray-300'}{}`
47           placeholder="Enter your email"
48         />
49         {errors.email && touched.email && <p
50           className="text-red-500
51           text-sm">{errors.email}</p>}
52       </div>
53       <div className="mb-4">
54         <label className="block
55           text-gray-700">Password</label>
56         <input
57           type="password"
58           name="password"
59           value={values.password}
60           onChange={handleChange}
61           className={`w-full p-2 border rounded
62             ${errors.password && touched.password ?
63               'border-red-500' : 'border-gray-300'}{}`
64           placeholder="Enter your password"
65         />
66         {errors.password && touched.password && <p
67           className="text-red-500
68           text-sm">{errors.password}</p>}
69       </div>
70       <button
71         type="submit"
72         className="w-full bg-blue-500 text-white p-2
73           rounded hover:bg-blue-600"
74       >
75         Login

```

```

63         </button>
64     </form>
65     })
66 </Formik>
67 </div>
68 );
69 }
70
71 const root =
72     ReactDOM.createRoot(document.getElementById('root'));
73     root.render(<LoginForm />);
74 </script>
75 </body>
76 </html>

```

### 3.3 Output

- Centered box, "Login" heading.
- Inputs with red border on error, error messages.
- "Login" button.
- **Behavior:** Valid submit pe alert, invalid pe errors.

## 4 14.3 Real-World Example: EduCampusHustle Login

### 4.1 Code Example: EduCampusHustle Login

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>EduCampusHustle Login</title>
6   <script
7     src="https://cdn.jsdelivr.net/npm/react@18.2.0/umd/react.development.js"
8   </script>
9   <script
10    src="https://cdn.jsdelivr.net/npm/react-dom@18.2.0/umd/react-dom.development.js"
11  </script>
12   <script
13    src="https://cdn.jsdelivr.net/npm/@babel/standalone@7.20.6/babel.min.js"
14  </script>
15   <script
16    src="https://cdn.jsdelivr.net/npm/formik@2.2.9/dist/formik.umd.min.js"
17  </script>
18   <script
19    src="https://cdn.jsdelivr.net/npm/yup@0.32.11/+esm"
20  </script>
21   <script src="https://cdn.tailwindcss.com"></script>
22 </head>
23 <body>
24   <div id="root"></div>
25   <script type="text/babel">
26     const validationSchema = Yup.object({
27       email: Yup.string().email('Invalid email').required('Email
28         is required'),

```

```

18     password: Yup.string().min(6, 'Password must be at least 6
19         characters').required('Password is required')
20 });
21
22 function LoginForm() {
23     const handleSubmit = (values) => {
24         alert('Welcome to EduCampusHustle, ${values.email}!');
25     };
26
27     return (
28         <div className="max-w-md mx-auto mt-10 p-6
29             bg-gradient-to-br from-blue-500 to-purple-600
30             rounded-lg shadow-lg text-white">
31             <h2 className="text-3xl font-bold mb-6">Login to
32                 EduCampusHustle</h2>
33             <Formik
34                 initialValues={{ email: '', password: '' }}
35                 validationSchema={validationSchema}
36                 onSubmit={handleSubmit}
37             >
38                 ({({ handleSubmit, handleChange, values, errors,
39                     touched }) => (
40                     <form onSubmit={handleSubmit}
41                         className="space-y-4">
42                         <div>
43                             <label className="block
44                                 text-white">Email</label>
45                             <input
46                                 type="email"
47                                 name="email"
48                                 value={values.email}
49                                 onChange={handleChange}
50                                 className={`w-full p-2 rounded
51                                     ${errors.email && touched.email ?
52                                         'border-red-500' : 'border-transparent'}
53                                 bg-white text-gray-800`}
54                                 placeholder="Enter your email"
55                             />
56                             {errors.email && touched.email && <p
57                                 className="text-red-200
58                                     text-sm">{errors.email}</p>}
59                         </div>
60                         <div>
61                             <label className="block
62                                 text-white">Password</label>
63                             <input
64                                 type="password"
65                                 name="password"
66                                 value={values.password}
67                                 onChange={handleChange}

```

```

55         className={ `w-full p-2 rounded
                    ${errors.password && touched.password ?
                      'border-red-500' : 'border-transparent'}
                    bg-white text-gray-800` }
56         placeholder="Enter your password"
57       />
58       {errors.password && touched.password && <p
        className="text-red-200
        text-sm">{errors.password}</p>}
59     </div>
60     <button
61       type="submit"
62       className="w-full bg-green-500 text-white p-2
        rounded hover:bg-green-600"
63     >
64       Login
65     </button>
66   </form>
67   )}
68 </Formik>
69 </div>
70 );
71 }
72
73 const root =
74   ReactDOM.createRoot(document.getElementById('root'));
75   root.render(<LoginForm />);
76 </script>
77 </body>
</html>

```

## 4.2 Output

- Gradient box, "Login to EduCampusHustle" heading.
- Inputs with white background, red border on error.
- "Login" button (green).
- **Behavior:** Valid submit pe welcome alert.

## 5 Common Mistakes

- No validation.
- Overcomplicating forms.
- Ignoring UX.

## 6 Interview Tips

- **Forms kaise handle karte ho?** Controlled components aur Formik.

- **Validation kaise add karte ho?** Yup ke saath.
- **Form optimization?** useMemo/useCallback.

## **7 Assignment: Practice Time**

### **7.1 Task 1: Signup Form**

- Fields: Name, Email, Password, Confirm Password.
- Validation: Email valid, password 6+ chars, confirm match.
- Formik aur Yup use karo.
- Tailwind se gradient background, green button.

### **7.2 Task 2: Course Enrollment Form**

- Fields: Course Name, Student Name, Payment Amount.
- Validation: All required, amount  $> 0$ .
- On submit: Alert with data.
- Tailwind se responsive design.

### **7.3 Task 3: Enhance Login Form**

- Add "Remember Me" checkbox.
- Add "Forgot Password?" link.
- Button color purple (9333EA) on hover.