**1. Explain React Forms Validation**

**React form validation** ensures that the user inputs correct and complete data before submission. It can be:

* **Client-side**: Using conditions in onChange or onSubmit handlers.
* **Custom validation**: Checking length, format (e.g., email), etc.
* **Using libraries**: Like Formik, Yup, or React Hook Form for advanced validation.

**Example:**

if (email === "" || !email.includes("@")) {

alert("Please enter a valid email");

}

**2. Identify the Differences Between React Form and HTML Form**

| **Feature** | **HTML Form** | **React Form** |
| --- | --- | --- |
| Control | Browser handles input | React controls input using state |
| Validation | Built-in HTML attributes (required) | Custom validation in JS |
| Data handling | Automatic form submission | Manual data collection via state |
| Flexibility | Limited | Highly flexible with dynamic behaviors |
| Event handling | Traditional HTML events | Synthetic events (onChange, onSubmit) |

**3. Explain About Controlled Components**

**Controlled components** in React are form elements whose values are controlled by React state. Input changes are handled by state updates.

**Example:**

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

<input type="text" value={name} onChange={(e) => setName(e.target.value)} />

This ensures full control over the form input behavior and validation.

**4. Identify Various React Form Input Controls**

React supports all basic HTML input types as **controlled inputs**:

* <input type="text" />
* <input type="password" />
* <input type="checkbox" />
* <input type="radio" />
* <textarea />
* <select />
* <input type="file" />

Each of these can be linked to state for dynamic behavior.

**5. Explain How to Handle React Forms**

**Handling React forms** involves:

* Creating state variables for input values
* Using onChange handlers to update the state
* Managing multiple inputs with a common handler (using name attribute)

**Example:**

const handleChange = (e) => {

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

};

**6. Explain About Submitting Forms in React**

Form submission in React is handled using the onSubmit event:

**Steps:**

1. Use onSubmit in <form>.
2. Prevent default reload with e.preventDefault().
3. Access form data from state.
4. Perform actions like validation or API call.

**Example:**

const handleSubmit = (e) => {

e.preventDefault();

console.log("Submitted data:", formData);

};