**Explain React Forms validation**  
React form validation is the process of ensuring that user inputs meet specific requirements before allowing form submission. Validation can be handled manually using state and conditional logic or with the help of third-party libraries like Formik or React Hook Form. Typical validations include checking for required fields, matching input patterns, or limiting input length. React’s controlled components make it easy to perform validations dynamically as the user types by checking the input values in the onChange handler and updating error messages in the state.

**Identify the differences between React Form and HTML Form**  
The main difference between React and HTML forms lies in how data is managed. In HTML forms, the browser maintains the form data, and submission is typically handled through a page reload. In React forms, the component manages the form data using state (controlled components), allowing more control over input validation, error handling, and submission logic. React forms offer a dynamic and interactive user experience, while HTML forms follow a more static, traditional approach.

**Explain about controlled components**  
Controlled components are form inputs whose values are controlled by React through the component's state. Instead of relying on the DOM to manage input values, React sets the input value via a state variable and updates it through an onChange handler. This approach allows the developer to have complete control over the form’s data, enabling dynamic validation, conditional rendering, and integration with APIs or other logic.

**Identify various React Form input controls**  
React supports several types of form input controls:

* Text input (<input type="text">) for single-line text
* Textarea (<textarea>) for multi-line input
* Password input (<input type="password">) for masked text
* Checkbox (<input type="checkbox">) for boolean selections
* Radio buttons (<input type="radio">) for selecting one option in a group
* Select dropdown (<select>) for choosing one or more options
* File input (<input type="file">) for uploading files  
  Each control is typically linked to state in a controlled component setup.

**Explain how to handle React Forms**  
Handling React forms involves initializing state to store form data, binding each input control to the state using the value or checked attribute, and updating the state through onChange event handlers. This enables real-time updates and validations. The state can then be used to display error messages, control the submit button’s availability, or pass form data to a backend service.

**Explain about submitting forms in React**  
Form submission in React is managed by attaching an onSubmit handler to the <form> element. The handler function typically calls event.preventDefault() to stop the default form submission behavior. After that, the component can access the current form data from the state, perform validations, and optionally send the data to a server or perform other actions. This method provides full control over the submission process and avoids page reloads.