

Skills
Network

Implementing Forms

What you will learn



Explain the purpose of forms

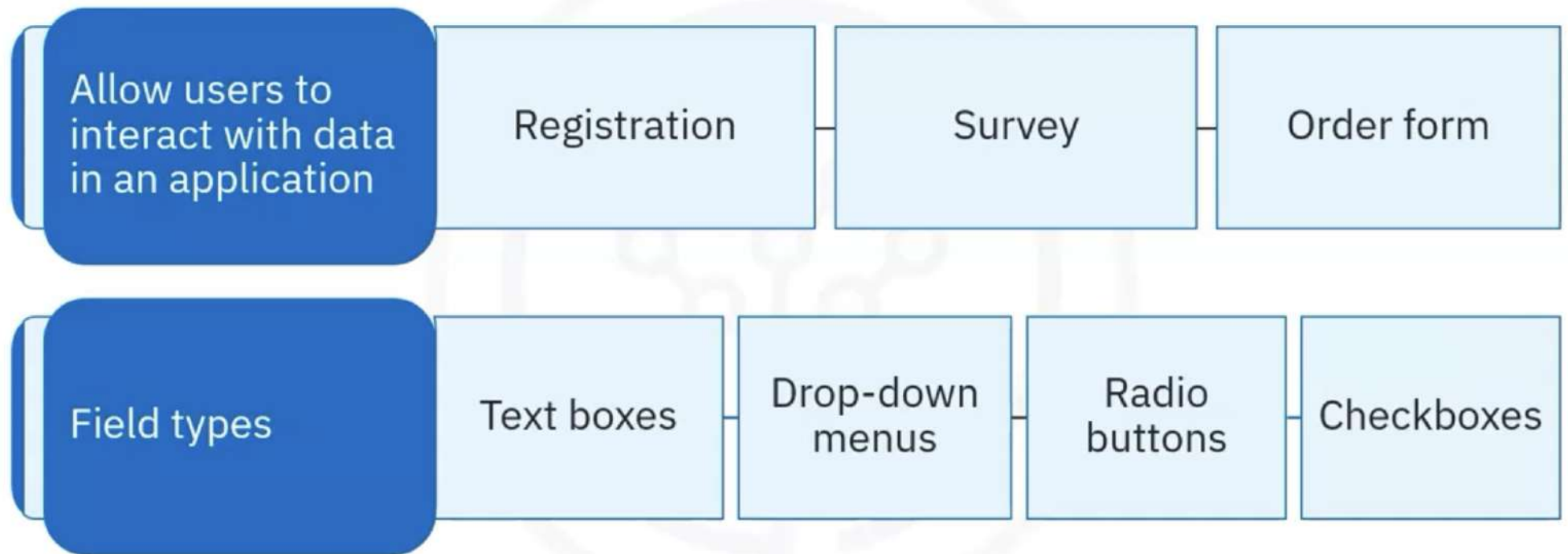


Compare and contrast controlled and uncontrolled form components

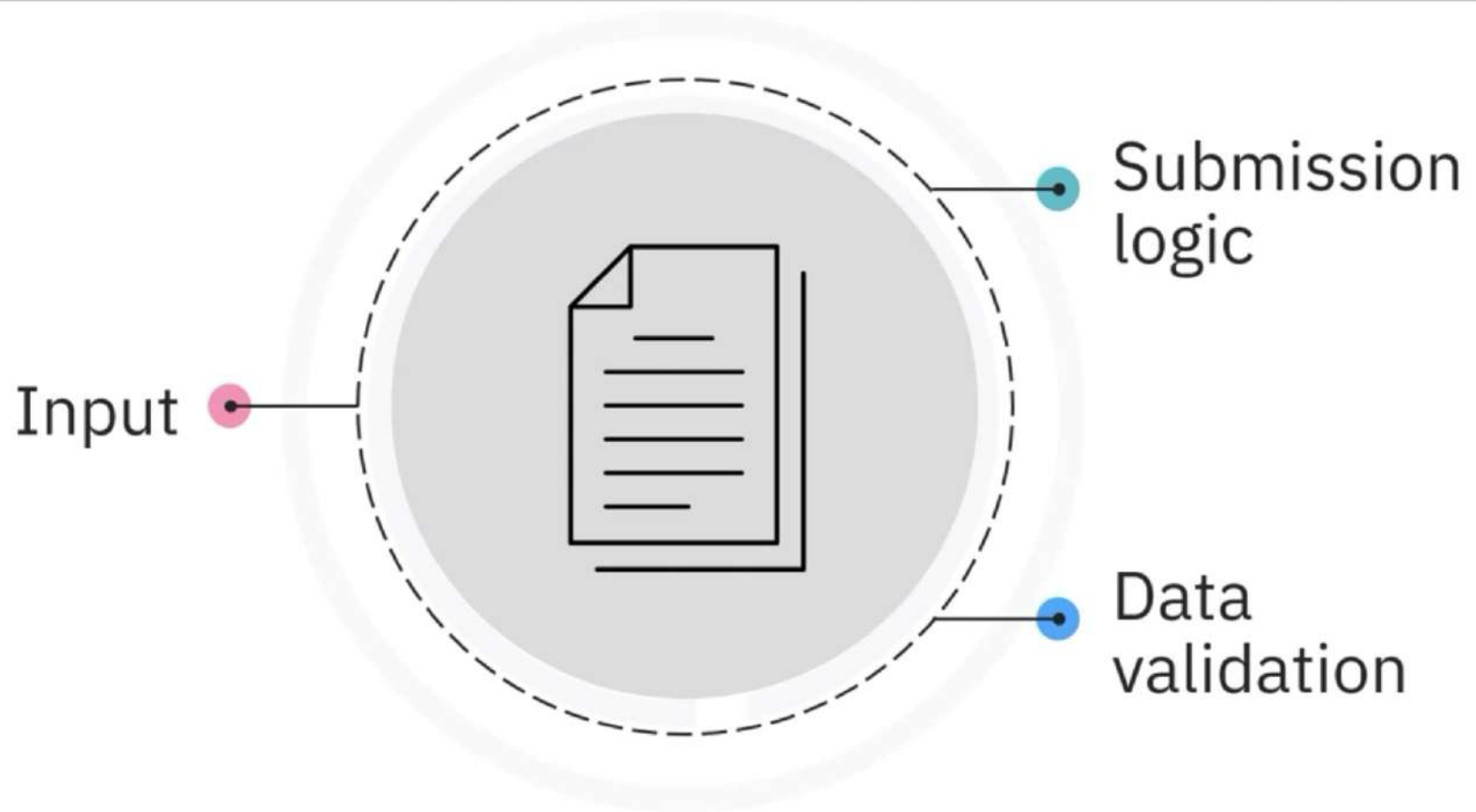


Summarize the benefits of the React hook library for form development

Form introduction



Handling forms



HTML versus React

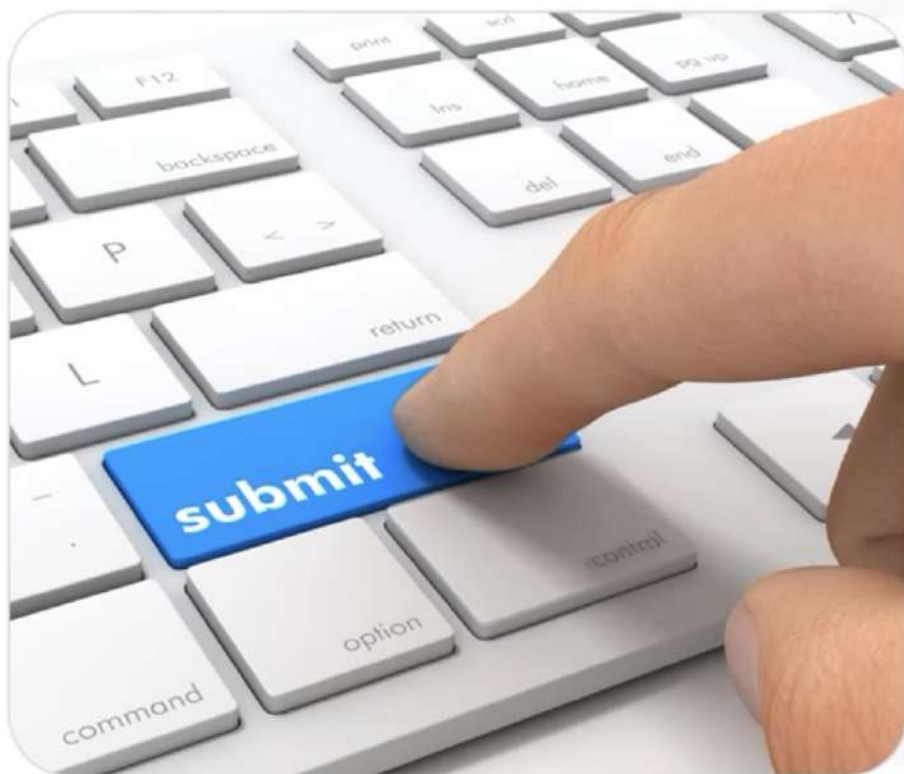
HTML form elements

`<input>`
`<textarea>`
`<select>`

React forms

Components
State stores input
`setState()` to update

Submission



Code contains submission logic

Password



Submit

Validation

Enter your password

Retype your password

Submit

Display: Password must have a minimum of 8 characters and a combination of lower and uppercase letters.

Validation tasks

- Get the input
- Manage and update state
- Validate input
- Display errors

Uncontrolled versus controlled

Uncontrolled

- Browser keeps track of data
- Requires less coding
- Simple behavior only

Controlled

- State keeps track of data
- Write code logic for updating
- Robust behaviors

Uncontrolled components

Browser handles form elements

Data stored by the DOM

Input updates automatically

Requires “ref” functions
to get form input

No event handlers are needed
for state changes

Controlled components

You write logic to handle state changes

The state stores the data

onChange events
setState() for input
Event handlers

Example form

```
import React, { Component } from "react";
export default function App() {
  const [email, setEmail] = React.useState("");
  const handleSubmit = (event) => {
    console.log(`Email: ${email}`);
    event.preventDefault();
  }
  return (
    <form onSubmit = { handleSubmit }>
      <h1>Registration</h1>
      <label>
        Email:<input name="email" value={ email }
          onChange={ e => setEmail(e.target.value) } required />
      </label>
      <button>Submit</button>
    </form>
  );
}
```

React hook form library

For applications
with forms
State management
Data validation

Reduces code
Improves
performance

`install react
hook form`

Recap

In this video, you learned that:

- Forms provide a way for users to interact with data in a web application
- Forms usually require input fields, submission logic, and validation rules to handle user interactions
- The browser keeps track of data and the state of elements with the DOM in uncontrolled components
- Controlled components require you to explicitly manage the UI input data with the component's state
- The React hook form library optimizes performance by reducing the number of times a component renders