



# React Forms

Tessema Mengistu (Ph.D.)

Department of Computer Science

Virginia Tech

[mengistu@vt.edu](mailto:mengistu@vt.edu)

# Outline

- HTML Forms
- React Forms
  - Controlled Forms
  - Uncontrolled Forms

# HTML Forms

- Forms allow users to input and submit information
- The HTML **<form>** element is used to create an HTML form for user input:

`<form>`

`// form elements`

`</form>`

# HTML Forms

- The `<form>` element is a container for different types of **input** elements:
  - text fields
  - checkboxes
  - radio buttons
  - submit buttons
  - etc.

# React Forms

- Very similar to HTML forms
  - Uses the basic HTML form elements
- Can be:
  - Uncontrolled
    - Form data is usually handled by the DOM
    - **useRef** hook
  - Controlled
    - Form data is usually handled by the component
    - `useState` hook
- Third party libraries
  - Formik
  - Redux Form
  - React Hook Form
  - yup
  - etc.

# Controlled Components

- In React, form data is usually handled by the components
  - Mutable state is typically kept in the **state** property of components
    - Managed using **useState** hook
  - We need to use:
    - The **value** prop to set the value of form elements
      - Takes the initial state value
    - The **onChange** event to handle changes made to the value
      - The setter method should be called

# Controlled Components

```
function Login() {  
  const [userName, setUsername] = useState('');  
  . . .  
  const handleChange = (event) => {  
    . . .  
    setUsername(event.target.value);  
    . . .  
  };  
  return (  
    <form>  
      <label>Input Value:  
        <input type="text" value={userName} onChange =  
{handleChange} />  
      </label>  
    . . .  
  );  
}
```

# Controlled Components

- Labels provide information about their associated elements and are important for accessibility
- Label Binding

```
    . . .  
<h2>Checkout</h2>  
    <form>  
      <label htmlFor="name">  
        Name  
        <input  
          id="name"  
          type="text"  
        />  
      </label>  
    </form>  
    . . .
```



# Common Events & Event Attributes of Tags

## *Event*

blur

change

click

focus

load

...

select

submit

unload

## *Tag Attribute*

onblur

onchange

onclick

onfocus

onload

...

onselect

onsubmit

onunload

## *React Attribute*

onBlur

onChange

onClick

onFocus

onLoad

...

onSelect

onSubmit

onUnload

# Controlled Components

- onSubmit
  - By default, the browser submits the form values as a new GET request to the current URL, then displays the result
  - Can be controlled by preventing the default behavior

```
event.preventDefault();
```

# Controlled Components

- Advantages:
  - Gives full control over the form elements
    - Easier form validation
    - Easier to apply formatting
    - Provides uniformity across components

# Uncontrolled Components

- Form data is handled (controlled) by the DOM itself
- Needs an access to the DOM element in order to get values
  - useRef hook

# Uncontrolled Components

- useRef
  - A specialized hook
    - The useRef hook creates and stores a mutable JavaScript object that persists through re-renders
    - Store values that are not intended to trigger a re-render of the component

```
function Login() {
  const username = useRef();
  . . .
  const HandleSubmit = (event) => {
    . . .
    console.log(username.current.value);
    . . .
  };
  return (
    <form onSubmit = {HandleSubmit}>
    <label>Input Value:
    <input type="text"
value={username.current} ref = {username}/>
    </label>
```

# References

- <https://react.dev/reference/react-dom/components>