Form Handeling

```
import { useState } from "react";

function App() {

    const [firstName, setFirstName] = useState("");
    const [lastName, setLastName] = useState("");

    function FirstchangeHandler(event) {
        setFirstName(event.target.value);
    }
    function LastchangeHandler(event) {
        setLastName(event.target.value);
    }

    return (
      <div>
        <form>
          <br/>
          <input
            type = "text"
            placeholder = "First name"
            onChange = {FirstchangeHandler}
          />

        <br/>
```

```
        <input
          type="text"
          placeholder="last name"
          onChange={LastchangeHandler}
        />
      </form>
    </div>
  );
}
export default App;
```

The problem with above code is that for every input field we are defining different onChange handler functions and multiple useState hooks.

Let say we have input field and other form components in a very large number then writing onChange handler for every component is not the best way to create a form.

The more optimized code is given below.

Code

```
function App() {
```

```
const [formData, setFormData] = useState({
    firstName: "",
    lastName: "",
    email: "",
    isVisible: true
});

console.log(formData);

function changeHandler(event) {
    const {name, value, checked, type} = event.target
    setFormData( (prevFormData) => {
        return {
            ...prevFormData,
            [name]: type === "checkbox"
            ? checked : value
        };
    });
}

return (
    <div>
        <form>
```

name attribute is provided in every element

// Here the whole data of the form is being tracked i.e. in formData

Here it is storing the previous state of the form with the help of ...spread operator.

```
<input
    type="text"
    placeholder="first name"
    onChange={changeHandler}
    name="firstName"
    value={formData.firstName}
/>
<br/>
<input
    type="text"
    placeholder="last name"
    onChange={changeHandler}
    name="lastName"
    value={formData.lastName}
/>

<br/>

<input
    type="email"
    placeholder="enter your email"
    onChange={changeHandler}
    name="email"
    value={formData.email}
/>
```

// With the help of value you can track the ~~value~~ state of individual element.

Controlled components

```
<br/>

<input
    type = "checkbox"
    onChange={ changeHandler }
    name = "isVisible"
    id = "isVisible"
    checked = {formData.isVisible}
/>
<label htmlFor="isVisible">
    Am I visible?
</label>
</form>
</div>
<br/>
```

'htmlFor' attribute is used to attach the label with checkbox with the help of 'id' attribute i.e. isVisible.

// Controlled Components

// It maintains state inside the component

```
<fieldset>
    <legend> Mode: </legend>
    <input
    type = "radio"
    onChange = { changeHandler }
    name = "mode"
```

```
            value="Online-mode"
            id = "Online-mode"
            checked={formData.mode === "Online-Mode"}
        />
        <label htmlFor='Online-mode'>Online Mode</label>

        <input
            type="radio"
            onChange={changeHandler}
            name="mode"
            value = "Offline-mode"
            id = "Offline-mode"
            checked={formData.mode === "Offline-Mode"}
        />
        <label htmlFor='Offline-mode'>Offline Mode</label>
    </fieldset>
    <label htmlFor='favCar'>Favorite car</label>
    <select
            name = "favCar"
            id = "favCar"
            value = {formData.favCar}
            onChange = {changeHandler}
    >
    <option value="scarpio">Scarpio</option>
    <option value="tharr">Tharr</option>
    </select>
    <button>Submit</button>
</form>
</div>
```