React.js

Regards: Hassan Bilal

TABLE OF CONTENTS

01 > Custom Hooks

02 > Forms in React

01

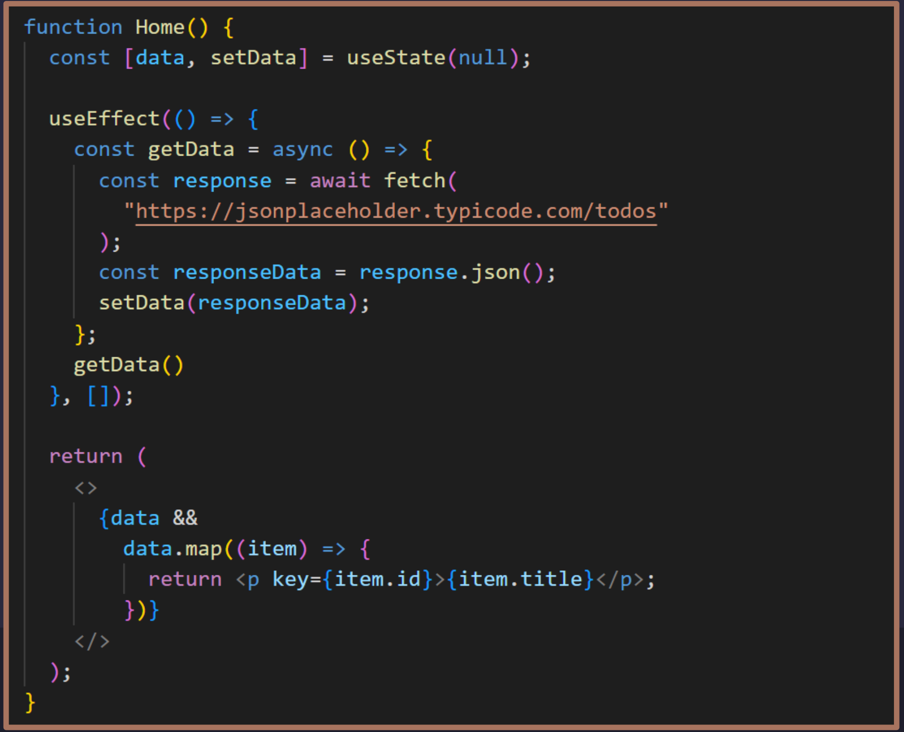
Custom Hooks

Custom Hooks

- Hooks are reusable functions.
- When you have component logic that needs to be used by multiple components.
- We can extract that logic to a custom Hook.
- Custom Hooks start with "use".
- Example: useFetch.

Custom Hooks

- Fetching data in our Home component and displaying it.



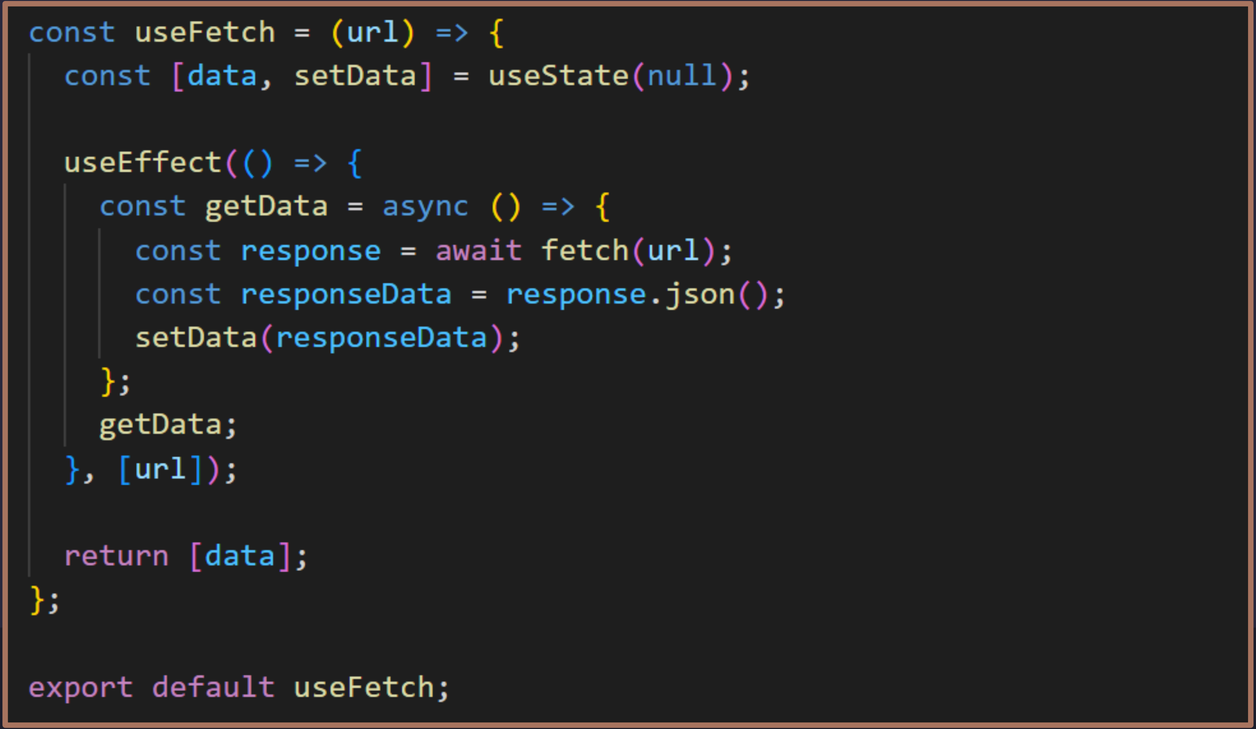
```
function Home() {
  const [data, setData] = useState(null);

  useEffect(() => {
    const getData = async () => {
      const response = await fetch(
        "https://jsonplaceholder.typicode.com/todos"
      );
      const responseData = response.json();
      setData(responseData);
    };
    getData()
  }, []);

  return (
    <>
      {data &&
        data.map((item) => {
          return <p key={item.id}>{item.title}</p>;
        })}
    </>
  );
}
```

Custom Hooks

- The fetch logic may be needed in other components as well.
- So we will extract that into a custom Hook.

Custom Hooks

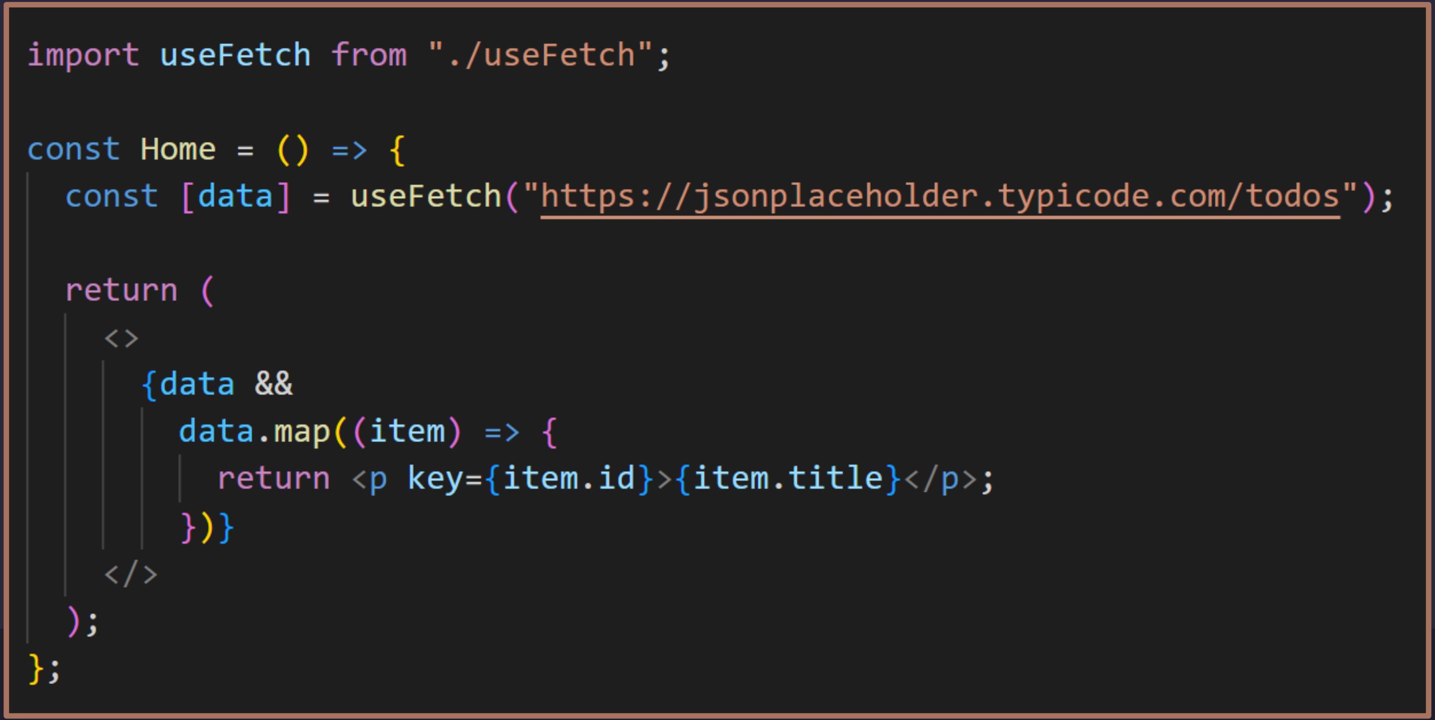
- Now we have our custom hook in place.



```
const useFetch = (url) => {
  const [data, setData] = useState(null);

  useEffect(() => {
    const getData = async () => {
      const response = await fetch(url);
      const responseData = response.json();
      setData(responseData);
    };
    getData;
  }, [url]);

  return [data];
};

export default useFetch;
```

Custom Hooks

- Now we have our custom hook in place.

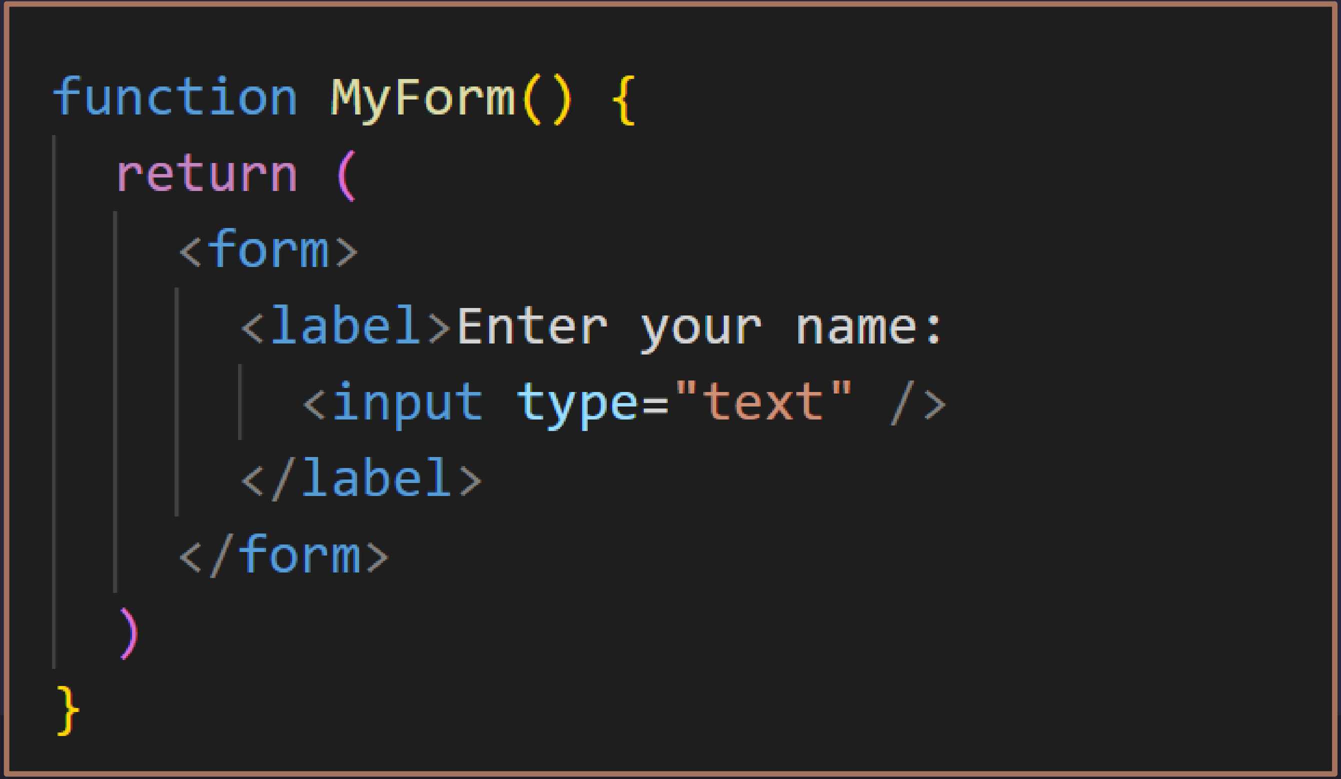


```
import useFetch from "./useFetch";

const Home = () => {
  const [data] = useFetch("https://jsonplaceholder.typicode.com/todos");

  return (
    <>
      {data &&
        data.map((item) => {
          return <p key={item.id}>{item.title}</p>;
        })}
    </>
  );
};
```

02

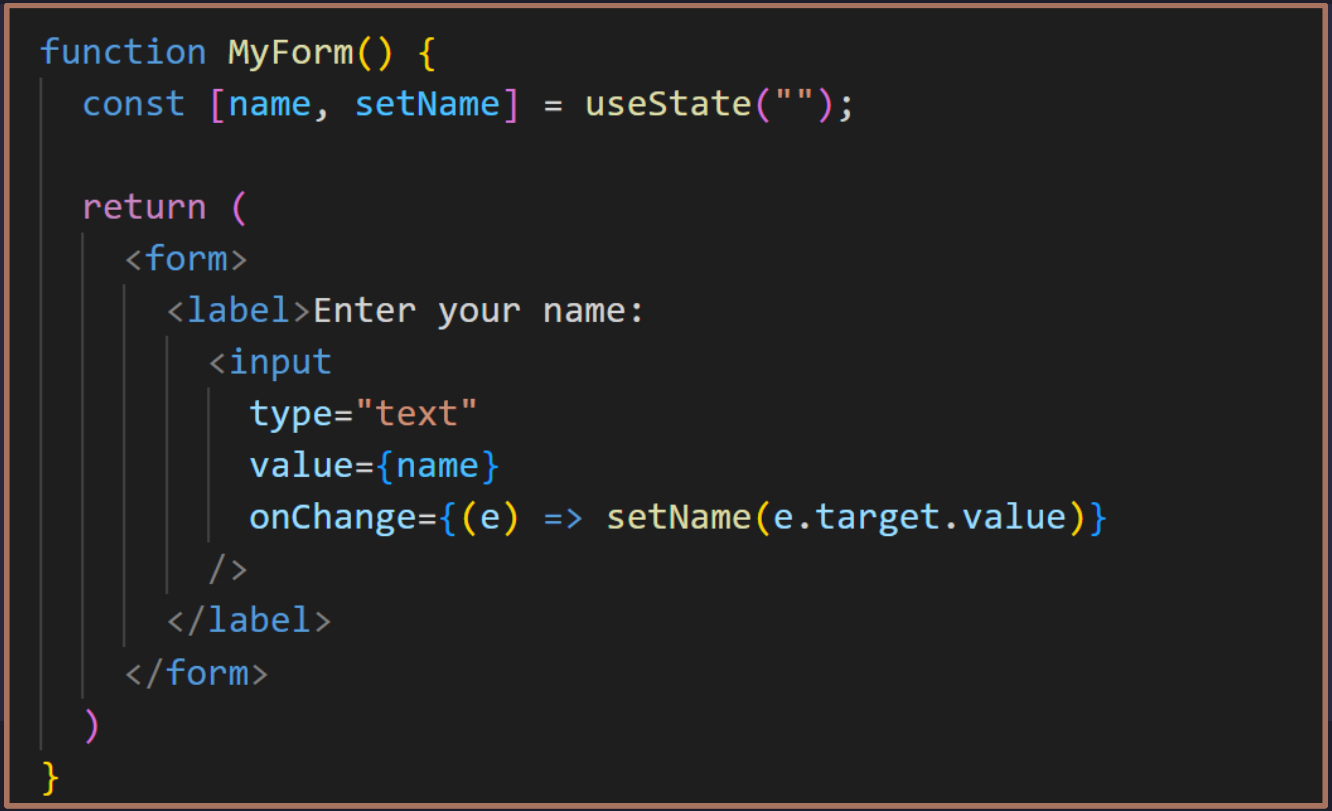
Forms in React

Forms in React



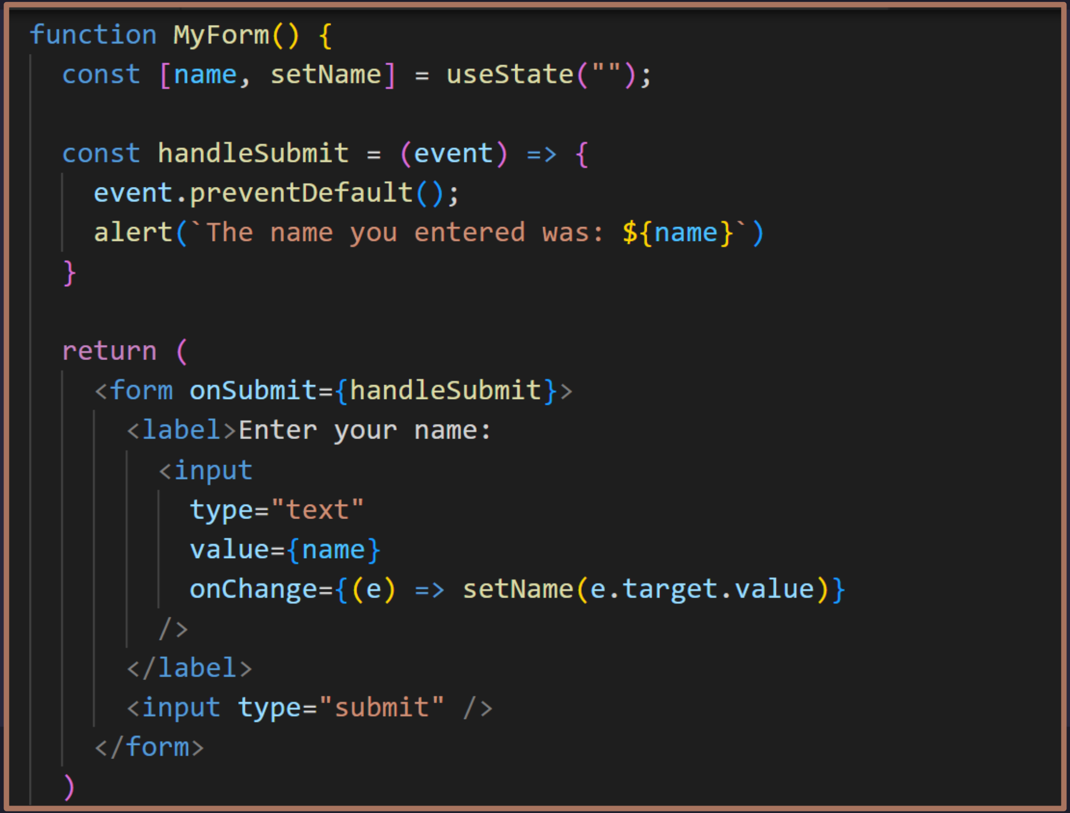
```
function MyForm() {
  return (
    <form>
      <label>Enter your name:
        <input type="text" />
      </label>
    </form>
  )
}
```

Forms in React



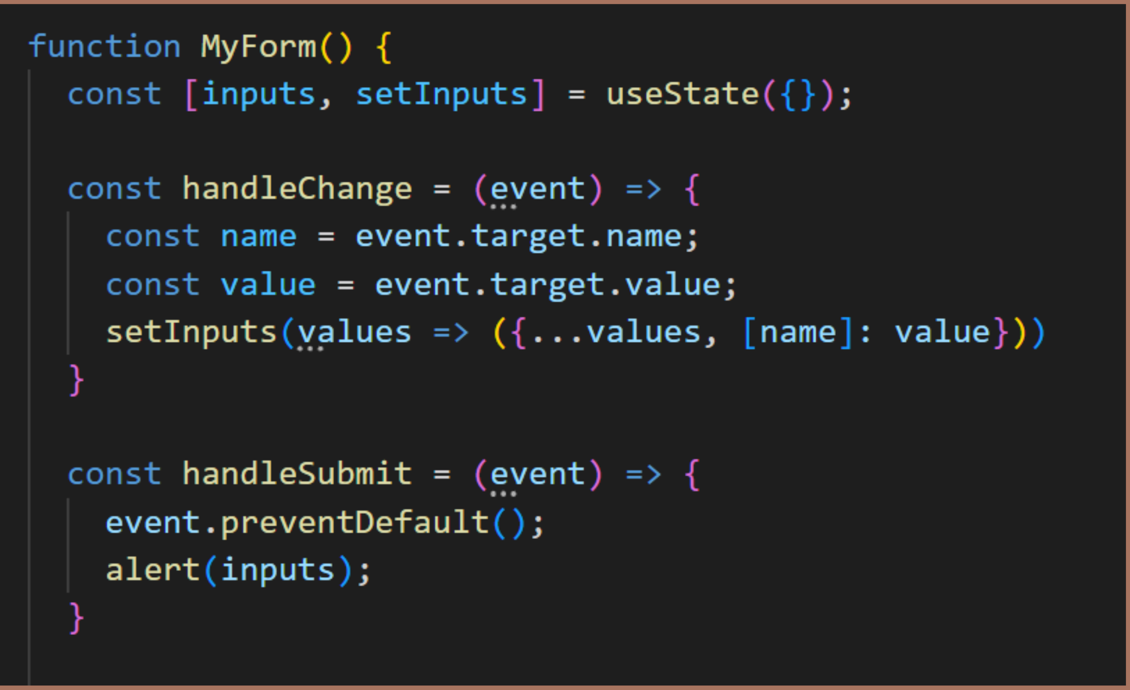
```
function MyForm() {
  const [name, setName] = useState("");

  return (
    <form>
      <label>Enter your name:
        <input
          type="text"
          value={name}
          onChange={(e) => setName(e.target.value)}
        />
      </label>
    </form>
  )
}
```

Forms in React

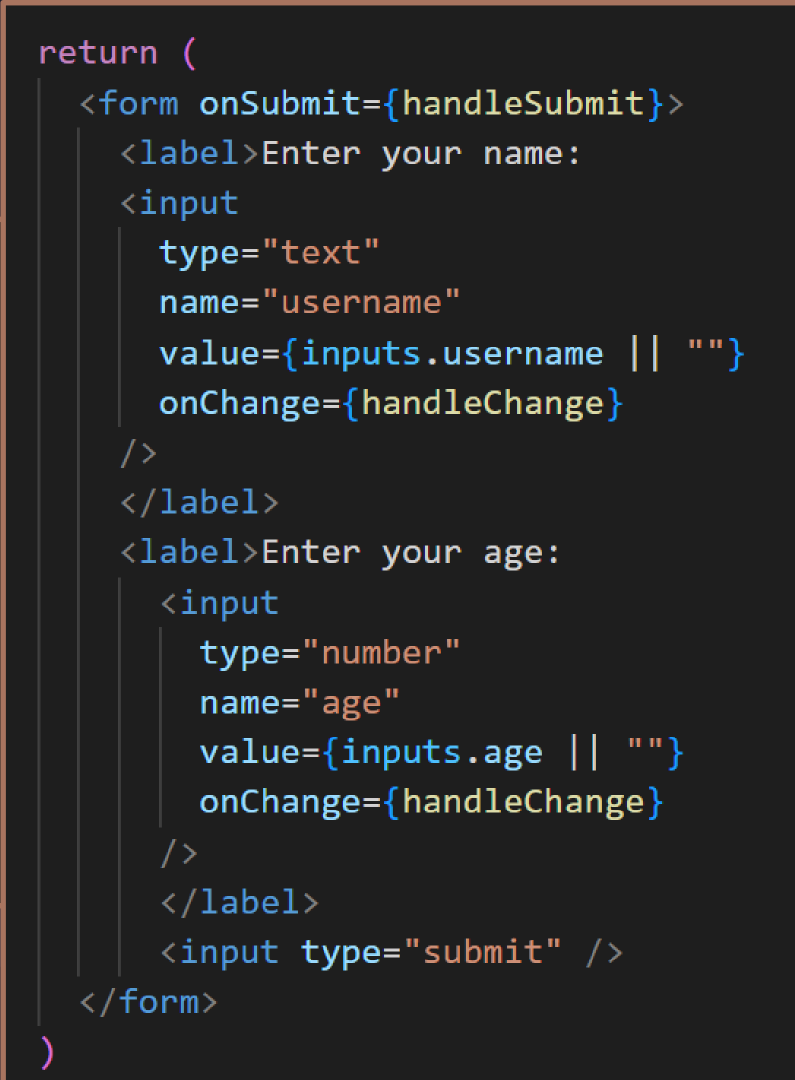


```
function MyForm() {
  const [name, setName] = useState("");

  const handleSubmit = (event) => {
    event.preventDefault();
    alert(`The name you entered was: ${name}`)
  }

  return (
    <form onSubmit={handleSubmit}>
      <label>Enter your name:
        <input
          type="text"
          value={name}
          onChange={(e) => setName(e.target.value)}
        />
      </label>
      <input type="submit" />
    </form>
  )
```

Forms in React – Multiple input



```
function MyForm() {
  const [inputs, setInputs] = useState({});

  const handleChange = (event) => {
    const name = event.target.name;
    const value = event.target.value;
    setInputs(values => ({...values, [name]: value}))
  }

  const handleSubmit = (event) => {
    event.preventDefault();
    alert(inputs);
  }
```



```
return (
  <form onSubmit={handleSubmit}>
    <label>Enter your name:
    <input
      type="text"
      name="username"
      value={inputs.username || ""}
      onChange={handleChange}
    />
    </label>
    <label>Enter your age:
      <input
        type="number"
        name="age"
        value={inputs.age || ""}
        onChange={handleChange}
      />
      </label>
      <input type="submit" />
  </form>
)
```

<QnA>

Thanks!

Regards: Hassan Bilal