**Understanding React useState and Event Handling**

**What is useState?**

1. **React Basics (Starting Point):**
   * In React, every website or application **needs to handle data** that can change over time. This is where **state** comes in.
   * Example: Mu gihe ufite umubare ugomba kwiyongera cyangwa kugabanuka (like a counter), uwo mubare urahinduka igihe umuntu akanda button.
2. **What is a Hook?**
   * "Hooks" ni uburyo React idufasha gukoresha ibintu bikomeye (nk'**state**) mu buryo bworoshye.
   * Don't worry too much about the word "hook"—it's just a tool React gives us.
3. **What Does useState Do?**
   * **useState** ni imwe muri izo tools. Idutegurira:
     + **Ahantu ho kubika amakuru ahinduka (state).**
     + **Uburyo bwo guhindura ayo makuru igihe cyose bishobotse.**
   * *Example:* Niba dushaka kubika umubare utangirana na zero:
   * const [count, setCount] = useState(0);
     + **count**: Aho umubare ubu uri kubikwa.
     + **setCount**: Ni function izakoreshwa guhindura uwo mubare.
     + **0**: Umubare wa mbere w'itangiriro.

**What is Event Handling?**

1. **What is an Event?**
   * **Event** ni igikorwa umuntu akora mu website:
     + Gukanda ku button (click).
     + Kwandika muri input box.
     + Gukora scroll ku page.
2. **Why Do We Need Event Handling?**
   * Iyo abantu bakoze ibintu muri website, tugomba **kumva ibyo bakoze** no kubikosora.
   * *Example:* Iyo umuntu akanze kuri button yo "Increment", counter igomba kwiyongera.
3. **How Do We Handle Events in React?**
   * Tugira **functions** zishinzwe kumva ibyo bikorwa. Izo functions twita **event handlers**.
   * Zikoreshwa nko muri:
   * <button onClick={handleClick}>Click Me</button>

**Step-by-Step Guide to Creating a Counter App**

Now let’s build a **simple counter app** where we’ll use:

* **useState** to store and update the counter value.
* **Event handling** to increment, decrement, and reset the counter.

**Step 1: Start a New React App**

1. Make sure your environment is set up with Node.js and create a new React app:
2. npx create-react-app counter-app
3. cd counter-app
4. npm start

**Step 2: Initialize State**

Add this code to your App.js file:

import React, { useState } from "react";

function App() {

// Create state to store the counter value

const [count, setCount] = useState(0); // Starting at 0

return (

<div>

<h1>Counter: {count}</h1>

</div>

);

}

export default App;

* **What’s Happening Here?**
  + Tugize umwanya witwa count aho tuzabika umubare uri kwiyongera.
  + Dukoresheje setCount guhindura agaciro ka count.

**Step 3: Add Buttons with Event Handling**

Now let’s add three buttons: **Increment**, **Decrement**, and **Reset**.

function App() {

const [count, setCount] = useState(0);

// Event Handlers

const increment = () => {

setCount(count + 1);

};

const decrement = () => {

setCount(count - 1);

};

const reset = () => {

setCount(0);

};

return (

<div style={{ textAlign: "center", marginTop: "50px" }}>

<h1>Counter: {count}</h1>

<button onClick={increment}>Increment</button>

<button onClick={decrement}>Decrement</button>

<button onClick={reset}>Reset</button>

</div>

);

}

* **Event Handlers Explained:**
  + Iyo umuntu akanze kuri button ya **Increment**:
    - increment function ikora kandi ikongera count kuri 1.
  + Iyo akanze kuri **Decrement**, bigabanya kuri 1.
  + Iyo akanze kuri **Reset**, bigarura kuri zero.

**Step 4: Add Styling for Better Display**

We can add some simple styles to make it look nicer:

<div style={{ textAlign: "center", marginTop: "50px" }}>

<h1>Counter: {count}</h1>

<button

style={{

padding: "10px",

margin: "5px",

backgroundColor: "green",

color: "white",

border: "none",

borderRadius: "5px",

}}

onClick={increment}

>

Increment

</button>

<button

style={{

padding: "10px",

margin: "5px",

backgroundColor: "red",

color: "white",

border: "none",

borderRadius: "5px",

}}

onClick={decrement}

>

Decrement

</button>

<button

style={{

padding: "10px",

margin: "5px",

backgroundColor: "blue",

color: "white",

border: "none",

borderRadius: "5px",

}}

onClick={reset}

>

Reset

</button>

</div>

**Step 5: Key Takeaways**

1. **useState Recap**
   * Dukoresha useState kugira ngo tubike agaciro (value) gashobora guhinduka.
   * Duhindura agaciro dukoresheje setState.
2. **Event Handling Recap**
   * Tugira **functions** zireba ibiba mu events (nko gukanda kuri button).
   * Izo functions tuzihuza na elements nka button zikoresheje attributes nka onClick.

**Challenge for Students**

1. Add a button to double the value of the counter.
2. Add a limit so the counter doesn’t go below zero.