**Custom Events** → Create and dispatch your own events for modular apps.

**Example Code:**

<!--Custom Events → Create and dispatch your own events for modular apps.-->

<button id="btn">Click Me</button>

<script>

const btn = document.getElementById("btn");

// Listen for custom event

btn.addEventListener("helloEvent", (e) => {

console.log("Custom Event Fired!", e.detail);

});

// Create and dispatch custom event

const event = new CustomEvent("helloEvent", {

detail: { message: "Hi from custom event" }

});

btn.dispatchEvent(event);

</script>

**1. What Are Custom Events?**

* Custom Events are user-defined events in JavaScript (or similar environments) that allow you to create and trigger your own events beyond the standard browser events (e.g., click, submit).
* They are useful for modular applications where different components need to communicate without tight coupling.
* For reporting, Custom Events can be used to track specific user interactions, application states, or metrics that are not covered by default browser events.

**2. Key Concepts for Creating and Dispatching Custom Events**

* **Creating a Custom Event**:
  + Use the CustomEvent constructor in JavaScript to create an event.
  + Example:

javascript

const myEvent = new CustomEvent('userAction', {

detail: { userId: 123, action: 'clicked\_feature' }, *// Custom data*

bubbles: true, *// Allows the event to bubble up the DOM*

cancelable: true *// Allows the event to be canceled*

});

* + The detail property holds custom data you want to pass with the event.
* **Dispatching the Event**:
  + Use the dispatchEvent method to trigger the event on a specific DOM element or globally.
  + Example:

javascript

document.dispatchEvent(myEvent);

* **Listening for Custom Events**:
  + Use addEventListener to listen for your custom event and handle the data.
  + Example:

javascript

document.addEventListener('userAction', (event) => {

console.log('User Action:', event.detail); *// Access custom data*

*// Send to analytics or reporting system*

});

**Real-Life Example:**

**⭐ Rating Widget**

- Custom Event: onRatingChange

- Triggered Behavior: Updates the user's feedback score when they select a rating (e.g., 4 stars).

- Why It Matters: Helps apps track user satisfaction or product reviews in real time.

**📅 Calendar Picker**

- Custom Event: onDateSelect

- Triggered Behavior: Loads tasks or events for the selected date.

- Why It Matters: Enables dynamic scheduling features—great for productivity apps or booking systems.

**🔔 Notification Box**

- Custom Event: onToastDismissed

- Triggered Behavior: Logs user interaction or resets the app state when a notification is closed.

- Why It Matters: Useful for tracking engagement or cleaning up temporary UI elements.