

## Note:

This is **one** method of completing the following exercises, not the only method. How you choose to answer will depend on how you interpret the question, but the logic should remain roughly the same.

---

## Real-Time Character Countdown (Tweet Composer)

### Task:

\*As an extra CSS exercise, quickly compose CSS to recreate a 'tweet' style composition input box.

Create a **tweet composer** that:

1. Allows the user to **input their tweet** in an input box, with a '***Tweet***' button at the bottom.
2. Limits tweets to **280 characters**.
3. Shows the **remaining character count** dynamically.
4. Disables the submit button if the limit is exceeded.
5. Allows users to **reset** the tweet.

### For an Extra Challenge:

1. Add **word count tracking**.
2. Show a **warning when 20 or fewer characters are left**.

```
// TweetCompoer.jsx
//This is a possible solution, there are all manner of ways in which you, individually, can accomplish it.
//I have excluded the App.js import statement needed to render this component.

//////////
import React, { useState } from "react";

function TweetComposer() {
  const maxChars = 280;
  const [tweet, setTweet] = useState("");

  function handleChange(event) {
```

```

    setTweet(event.target.value);
  }

  function resetTweet() {
    setTweet("");
  }

  return (
    <div>
      <h2>Compose Tweet</h2>

      <textarea
        value={tweet}
        onChange={handleChange}
        placeholder="What's happening?"
        rows="4"
      ></textarea>

      <div>
        <span>{maxChars - tweet.length} characters left</span>
        <button onClick={resetTweet}>Reset</button>
      </div>

      <button disabled={tweet.length > maxChars}>Tweet</button>
    </div>
  );
}

export default TweetComposer;

```