Here's a **complete, runnable JavaScript example** that gets the mouse coordinates **relative to a specific** `<div>` using `getBoundingClientRect()`.

HTML + JavaScript

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Mouse Coordinates Relative to a DIV</title>
<style>
  #myDiv {
    width: 300px;
    height: 200px;
    background-color: lightblue;
    border: 2px solid #333;
    position: relative;
    margin: 50px;
  }
  #coords {
    margin-top: 10px;
    font-family: monospace;
  }
</style>
</head>
<body>
```

```
<div id="myDiv">Move your mouse here</div>
<div id="coords">X: 0, Y: 0</div>

<script>
  const myDiv = document.getElementById('myDiv');
  const coordsDisplay = document.getElementById('coords');

  myDiv.addEventListener('mousemove', function(event) {
    try {
      // Get the bounding rectangle of the div
      const rect = myDiv.getBoundingClientRect();

      // Calculate mouse position relative to the div
      const x = event.clientX - rect.left;
      const y = event.clientY - rect.top;

      // Ensure values are within bounds (optional)
      const relX = Math.max(0, Math.min(x, rect.width));
      const relY = Math.max(0, Math.min(y, rect.height));

      coordsDisplay.textContent = `X: ${relX.toFixed(0)}, Y: ${relY.toFixed(0)}`;
    } catch (err) {
      console.error("Error calculating coordinates:", err);
    }
  });
</script>
```

```
</body>
</html>
```

How it works:

1. `getBoundingClientRect()`
   Returns the position and size of the element relative to the viewport.
2. **Relative coordinates**
   `event.clientX - rect.left` and `event.clientY - rect.top` give the mouse position relative to the top-left corner of the div.
3. **Bounds check**
   Optional clamping ensures coordinates don't go negative or exceed the div's size.

✅ This works even if the page is scrolled or the div is positioned anywhere.
If you want **coordinates relative to the div's content area including scroll inside the div**, you can use `event.offsetX` and `event.offsetY` directly.

If you want, I can also give you a **version that works for nested divs** where `offsetX`/`offsetY` might be misleading.
Do you want me to prepare that?