Our Buy Button is more complex than it needs to be — and that's exactly why GitHub Copilot (and sometimes TypeScript) complains. Stripe's Buy Button is intentionally simple, and the cleanest implementation avoids:

- manually creating DOM elements
- manually injecting scripts
- refs that mutate innerHTML
- race conditions where the script loads after the component renders

Let's turn this into a **minimal, stable, Next.js-friendly component** that deploys cleanly and avoids Copilot warnings.

⭐ The cleanest possible Stripe Buy Button component (Next.js)

Stripe recommends **embedding the script once globally**, not per-component. Then your component only needs to render the `<stripe-buy-button>` tag.

Step 1 — Add the script globally (layout.tsx)

```
// app/layout.tsx
export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <head>
        <script
          async
          src="https://js.stripe.com/v3/buy-button.js"
        ></script>
      </head>
      <body>{children}</body>
    </html>
  );
}
```

⭐ Step 2 — A clean, simple Buy Button component

```
"use client";
interface StripeBuyButtonProps {
  buyButtonId: string;
  className?: string;
}

export function StripeBuyButton({ buyButtonId, className }:
StripeBuyButtonProps) {
  return (
    <stripe-buy-button
      buy-button-id={buyButtonId}
      publishable-
key="pk_live_51RcPVICs6GUQsp1IaEnIePJEsDVsET7sakCSd6ruWgSPX2eKOU1NTJYuRQR6Cxmmql
za2BAcxeLziwLUxbnUJsHh00XgIR3bub"
      class={className}
    />
  );
}
```

Why this version is better:

- No useEffect
- No DOM manipulation
- No script injection
- No refs
- No hydration mismatch risk
- No Copilot warnings
- Fully compatible with Next.js App Router

Stripe’s custom element automatically initializes itself once the script is loaded.