

# Lenis Horizontal Scroll Setup (VS Code Agent + SSR)

## Goal Recap

You're using VS Code Web/Agent, so all client-side animation libraries must be safely initialized in a way that won't break SSR.

This guide will show you how to implement a smooth horizontal scrolling logo section using Lenis and GSAP inside a Next.js app.

## Step 1: Install Packages

In your VS Code terminal (agent or local), run:

```
npm install gsap @studio-freight/lenis
```

## Step 2: Create LenisProvider Component

Create a file at components/LenisProvider.tsx with the following content:

```
'use client'
import { useEffect } from 'react'
import Lenis from '@studio-freight/lenis'

export default function LenisProvider({ children }) {
  useEffect(() => {
    const lenis = new Lenis({ smooth: true, direction: 'vertical' })
    const raf = (time) => {
      lenis.raf(time)
      requestAnimationFrame(raf)
    }
    requestAnimationFrame(raf)
  }, [])

  return <>{children}</>
}
```

## Step 3: Use LenisProvider in Layout

Wrap your app layout in the LenisProvider:

```
import LenisProvider from '@components/LenisProvider'

export default function RootLayout({ children }) {
  return (
    <html>
```

## Lenis Horizontal Scroll Setup (VS Code Agent + SSR)

```
<body>
  <LenisProvider>{children}</LenisProvider>
</body>
</html>
)
}
```

### Step 4: HorizontalScrollLogos Component

Create a component at components/HorizontalScrollLogos.tsx:

```
'use client'
import { useRef, useEffect } from 'react'
import gsap from 'gsap'
import ScrollTrigger from 'gsap/ScrollTrigger'
gsap.registerPlugin(ScrollTrigger)

export default function HorizontalScrollLogos() {
  const containerRef = useRef(null)
  const trackRef = useRef(null)

  useEffect(() => {
    const container = containerRef.current
    const track = trackRef.current
    if (!container || !track) return

    const ctx = gsap.context(() => {
      const totalScroll = track.scrollWidth - container.offsetWidth

      gsap.to(track, {
        x: -totalScroll,
        ease: 'none',
        scrollTrigger: {
          trigger: container,
          start: 'top top',
          end: () => `+=${track.scrollWidth}`,
          scrub: true,
          pin: true,
          anticipatePin: 1,
        },
      })
    }, container)

    return () => ctx.revert()
  }, [])

  const logos = [
    '/logos/angular.svg',
    '/logos/react.svg',
```

## Lenis Horizontal Scroll Setup (VS Code Agent + SSR)

```
    '/logos/nextjs.svg',  
    '/logos/figma.svg',  
    '/logos/openai.svg',  
    '/logos/ionic.svg',  
    '/logos/adobe.svg',  
  ]  
  
  return (  
    <section ref={containerRef} className="relative h-screen overflow-hidden bg-white">  
      <div ref={trackRef} className="flex items-center gap-16 px-8 py-20 whitespace-nowrap">  
        {logos.map((src, i) => (  
          <img key={i} src={src} alt="" className="h-20 w-auto" />  
        ))}  
      </div>  
    </section>  
  )  
}
```

### SSR-Safe Practices

- Use 'use client' in all interactive components
- Avoid using window or document outside of useEffect
- Wrap DOM-dependent logic inside useEffect
- Register GSAP plugins inside useEffect
- Always check for element refs before using them