Sentry Complete Setup - January 3, 2026

✅ All Configuration Steps Completed

This document summarizes the comprehensive Sentry setup completed on January 3, 2026, addressing all items from the user’s checklist.

🎯 Changes Made

1. Fixed DSN Configuration

File: `.env`
- **OLD DSN:** `https://d649b9c85c145427fcfb62cecdeaa2d9e@o4510110703216128.ingest.us.sentry.io/4510154442089472`
- **NEW DSN:** `https://4649b9c85c1454217cfb62cecdea2d9e@o4510119703216128.ingest.us.sentry.io/4510154426089472`
- **Impact:** Now using the correct Sentry project credentials

2. Installed Profiling Package

Package: `@sentry/profiling-node`
- Added to `package.json` dependencies
- Required for server-side profiling integration

3. Updated Client Configuration

File: `sentry.client.config.ts`

Added the following features:
- ✅ `enableLogs: true` - Enables log sending to Sentry
- ✅ `Sentry.browserTracingIntegration()` - Performance monitoring
- ✅ `Sentry.browserProfilingIntegration()` - Client-side profiling
- ✅ `Sentry.consoleLoggingIntegration({ levels: ["log", "warn", "error"] })` - Console capture
- ✅ `profilesSampleRate: 0.1 (production) / 1.0 (dev)` - Profiling sample rate

4. Updated Server Configuration

File: `sentry.server.config.ts`

Added the following features:
- ✅ `enableLogs: true` - Enables log sending to Sentry
- ✅ `import { nodeProfilingIntegration } from "@sentry/profiling-node"` - Import profiling
- ✅ `integrations: [nodeProfilingIntegration()]` - Server-side profiling
- ✅ `profilesSampleRate: 0.1 (production) / 1.0 (dev)` - Profiling sample rate

5. Updated Edge Configuration

File: `sentry.edge.config.ts`

Added the following features:
- ✅ `enableLogs: true` - Enables log sending to Sentry

6. Added Document-Policy Header

File: `next.config.js`

Added to headers configuration:

```
{
  key: 'Document-Policy',
  value: 'js-profiling',
}
```

This header is required for browser profiling to work properly.

7. Fixed Tunnel Endpoint

File: `src/app/api/sentry-tunnel/route.ts`

Critical fixes:
1. **Updated host:** `o4510119703216128.ingest.us.sentry.io` (was using old host)
2. **Updated project ID:** `4510154426089472` (was using old project)
3. **Added authentication header:**
``` ``typescript ```
`   const publicKey = SENTRY_DSN.split('@')[0].split('//')[1];`
`   const sentryAuth = [` Sentry sentry_version=7 ` , `sentry_key=${publicKey} ` , `sentry_client=sentry.javascript.nextjs/10.32.1`,
].join(', ');

headers: {
'Content-Type': 'application/x-sentry-envelope',
'X-Sentry-Auth': sentryAuth, // <– CRITICAL FIX
}
```

Why this matters: The missing `X-Sentry-Auth` header was causing 401 Unauthorized errors when the tunnel tried to forward events to Sentry.

📋 Checklist Status

Item	Status	Details
Run Sentry wizard	⚠️ Skipped	Wizard timed out (interactive). Manual config completed instead.
Install @sentry/nextjs	✅ Done	Already installed (v10.32.1)
Enable logs	✅ Done	Added `enableLogs: true` to all 3 config files
Console logging integration	✅ Done	Added to client config with log/warn/error levels
Install @sentry/profiling-node	✅ Done	Installed and added to package.json
Node profiling integration	✅ Done	Added to server config with import and integration
Browser profiling integration	✅ Done	Added to client config
Document-Policy header	✅ Done	Added `js-profiling` header to next.config.js
Fix DSN	✅ Done	Updated from `d649...` to `4649...` in .env
Fix tunnel auth	✅ Done	Added `X-Sentry-Auth` header with proper public key

---

🚀 Deployment Instructions

For Render.com:

1. **Update Environment Variables:**

```
NEXT_PUBLIC_SENTRY_DSN=https://4649b9c85c1454217cfb62cecdea2d9e@o4510119703216128.ingest.us.sentry.io/4510154426089472
   SENTRY_DSN=https://4649b9c85c1454217cfb62cecdea2d9e@o4510119703216128.ingest.us.sentry.io/4510154426089472
```

⚠️ **CRITICAL:** Make sure to update BOTH variables in Render’s environment variables!

1. **Push to GitHub:**

```
bash
   git add .
   git commit -m "fix: Complete Sentry setup with all missing configuration"
   git push origin main
```

2. **Monitor Render Logs:**
   - Watch for Sentry initialization messages
   - Check for any errors during build/deploy
   - Look for successful event forwarding

Verification Steps:

1. **Test Error Capture:**
   - Trigger a test error in the application
   - Check Sentry dashboard for the error event
   - Verify it appears within 1-2 minutes

2. **Check Logs:**
   Look for these success messages:
```
[Sentry] ✅ Client-side initialization successful
   [Sentry] ✅ Server-side initialization successful
   [Sentry] ✅ Edge initialization successful
   [Sentry Tunnel] ✅ Event forwarded successfully
```

3. **Verify Tunnel:**
```
bash
   curl https://getcarelinkai.com/api/sentry-tunnel
```
   Should return: `{"status":"ok","message":"Sentry tunnel is operational",...}`

4. **Test Profiling:**
   - Navigate to various pages
   - Check Sentry Performance tab for profile data
   - Verify browser and server profiles appear

🐛 Troubleshooting

If Events Still Don’t Appear:

1. **Check Environment Variables:**
   - Verify DSN is set correctly in Render
   - Ensure no typos in the DSN
   - Confirm both SENTRY_DSN and NEXT_PUBLIC_SENTRY_DSN are set

2. **Check Logs:**
```
bash
   # Look for these errors:
   - "SENTRY_DSN is not set"
   - "DSN mismatch"
   - "401 Unauthorized"
   - "403 Forbidden"
```

3. **Verify Tunnel:**
   - Check that tunnel endpoint is accessible

- Verify authentication header is being sent
- Confirm public key extraction is working

4. **Check Sentry Dashboard:**
   - Verify project exists
   - Check DSN matches exactly
   - Ensure rate limits aren’t exceeded

Common Issues:

Issue	Solution
401 Unauthorized	Fixed! We added X-Sentry-Auth header
Events not appearing	Check DSN is correct in Render env vars
Profiling not working	Check Document-Policy header is set
Build failures	Verify @sentry/profiling-node is installed

📊 What to Expect

After deployment, you should see:

1. **Errors:** Captured and sent to Sentry dashboard
2. **Performance:** Transaction traces for page loads and API calls
3. **Profiling:** CPU profiles for slow operations
4. **Logs:** Console output captured and searchable
5. **Session Replay:** Video-like playback of user sessions

✨ Summary

Before:
- ❌ Wrong DSN (d649… instead of 4649…)
- ❌ Missing enableLogs configuration
- ❌ Missing console logging integration
- ❌ Missing profiling packages and integrations
- ❌ Missing Document-Policy header
- ❌ Tunnel missing authentication header

After:
- ✅ Correct DSN configured
- ✅ enableLogs enabled in all configs
- ✅ Console logging integration added
- ✅ Profiling packages installed and configured
- ✅ Document-Policy header added
- ✅ Tunnel authentication header fixed

Result: Sentry should now be fully functional! 🎉

📝 Files Changed

1. `.env` - Updated DSN
2. `package.json` - Added @sentry/profiling-node
3. `package-lock.json` - Dependencies updated
4. `sentry.client.config.ts` - Added enableLogs, console logging, browser profiling
5. `sentry.server.config.ts` - Added enableLogs, node profiling
6. `sentry.edge.config.ts` - Added enableLogs
7. `next.config.js` - Added Document-Policy header
8. `src/app/api/sentry-tunnel/route.ts` - Fixed DSN values and added auth header

Date: January 3, 2026
Status: ✅ Complete and ready for deployment
Next Step: Push to GitHub and deploy to Render