Fix Login Redirect Loop - QUICK SOLUTION

Tell the AI Agent:

"The app redirects back to login after OAuth callback. This is a middleware/session issue. Please fix:"

1. Check Middleware (MOST LIKELY ISSUE)

Look for `middleware.ts` or auth protection that's too aggressive:

```typescript
// middleware.ts - WRONG
export async function middleware(req: NextRequest) {
  const { data: { session } } = await supabase.auth.getSession()

  // This might be failing and redirecting
  if (!session) {
    return NextResponse.redirect('/login')
  }
}

// FIX: Add callback to excluded paths
export const config = {
  matcher: [
    '/((?!auth|_next/static|_next/image|favicon.ico).*)',
  ]
}
```

2. Fix Auth Callback Route

The callback might be redirecting before session is established:

typescript

```
// app/auth/callback/route.ts
export async function GET(request: Request) {
  const requestUrl = new URL(request.url)
  const code = requestUrl.searchParams.get('code')

  if (code) {
    const supabase = createRouteHandlerClient({ cookies })

    // Exchange code for session
    const { error } = await supabase.auth.exchangeCodeForSession(code)

    if (error) {
      console.error('Session exchange failed:', error)
      return NextResponse.redirect(`${requestUrl.origin}/login`)
    }

    // IMPORTANT: Wait for session to be established
    await new Promise(resolve => setTimeout(resolve, 1000))

    // Then redirect
    return NextResponse.redirect(`${requestUrl.origin}/dashboard`)
  }

  return NextResponse.redirect(`${requestUrl.origin}/login`)
}
```

3. Check Protected Routes

Your dashboard might be checking auth wrong:

typescript

```
// app/dashboard/page.tsx - WRONG
const session = await supabase.auth.getSession()
if (!session) redirect('/login')

// CORRECT
const { data: { session } } = await supabase.auth.getSession()
if (!session) redirect('/login')
```

4. Add Debug Logging

Add this temporarily to see what's happening:

typescript

```
// In your auth callback
console.log('Auth Callback Debug:', {
  hasCode: !!code,
  url: request.url,
  cookies: request.headers.get('cookie')
})

// In your dashboard
console.log('Dashboard Debug:', {
  hasSession: !!session,
  user: session?.user?.email
})
```

5. Quick Fix - Disable Auth Check Temporarily

To confirm it's an auth issue, temporarily comment out auth checks:

typescript

```
// In dashboard/page.tsx
// if (!session) redirect('/login')  // Comment this out
```

If the page loads, it's definitely the auth check.

The Real Issue:

Since it worked with NEON, the problem is likely:

1. Supabase cookies aren't being set properly
2. Middleware is checking auth before session is ready
3. Different cookie domains between dev/prod

Immediate Fix:

Add `/auth/callback` to your middleware exclusions and add a delay in the callback before redirecting.