# comprehensive\_nextjs\_build\_fix

Successfully implemented a comprehensive 7-step fix for the persistent Next.js build errors. The root cause was identified as a TypeScript compilation conflict with an Express server file using NextFunction.

**Key Accomplishments:** 1. **Excluded Express Server from Compilation** - Updated tsconfig.json to exclude src/ai-training/api/\*\* which eliminates the recurring NextFunction namespace error 2. **Replaced Next.js Configuration** - Comprehensive CSP headers, static export settings, and image optimization for proper deployment 3. **Complete Service Worker Removal** - Deleted SW files and added cleanup component to clear existing registrations 4. **Fixed Orders Page** - Converted from client-side useParams to server-side params prop for Next.js 15 compatibility 5. **Added Runtime Declarations** - Added export const runtime = "nodejs"; to 8 API routes and server files to suppress Supabase Edge Runtime warnings 6. **Generated Icon Assets** - Created required favicon and apple-touch-icon files 7. **Deployed Changes** - Committed comprehensive fixes in commit c05ee52 and triggered new Vercel build

**Expected Results:** - Build should now pass without TypeScript errors - No more service worker interference - Proper CSP security headers - Supabase Edge Runtime warnings suppressed - All missing assets provided

This systematic approach addressed both the immediate build failures and underlying architectural issues that were causing recurring problems.

## Key Files

* max-out-website/next.config.js: Comprehensive Next.js configuration with CSP headers and static export settings
* max-out-website/src/app/\_kill-sw.tsx: Service worker cleanup component to remove existing SW registrations and caches
* max-out-website/tsconfig.json: Updated TypeScript configuration excluding problematic Express server from compilation
* max-out-website/src/app/(shop)/orders/[orderId]/page.tsx: Fixed orders page converted to server component with proper params prop