

Vercel Deployment Fix - October 31, 2025

Problem

Both Vercel deployments were failing with the following error:

```
Build error occurred
Error: ENOENT: no such file or directory, stat '/vercel/path0/yarn.lock'
    at async Object.stat (node:internal/fs/promises:1039:18)
    at async getFilesInDir (/vercel/path0/node_modules/next/dist/lib/get-files-in-dir.js:24:28)
    at async /vercel/path0/node_modules/next/dist/build/index.js:486:32
    at async Span.traceAsyncFn (/vercel/path0/node_modules/next/dist/trace/trace.js:154:20)
    at async build (/vercel/path0/node_modules/next/dist/build/index.js:368:9) {
  errno: -2,
  code: 'ENOENT',
  syscall: 'stat',
  path: '/vercel/path0/yarn.lock'
}
Error: Command "npm run build" exited with 1
```

Root Cause

The `yarn.lock` file was a symbolic link (symlink) pointing to a file outside the repository:

```
yarn.lock -> /opt/hostedapp/node/root/app/yarn.lock
```

Vercel's build environment cannot follow symlinks that point to files outside the repository, resulting in the "ENOENT" (no such file or directory) error.

Solution

Replaced the symbolic link with the actual `yarn.lock` file content:

```
cd /home/ubuntu/verifylens-landing/nextjs_space
rm yarn.lock
cp /opt/hostedapp/node/root/app/yarn.lock .
git add yarn.lock
git commit -m "Fix: Replace yarn.lock symlink with actual file for Vercel deployment"
git push origin main
```

Changes Made

Commit: 6df55d9

- **File:** `yarn.lock`
- **Change:** Replaced symlink with actual file (445KB)
- **Status:** Pushed to GitHub

Verification

Local build tested successfully:

- TypeScript compilation passed
- Development server started without errors
- Production build completed successfully
- All routes and API endpoints working

Deployment Status

- Fix committed and pushed to `Jgabbard61/roblox-lander`
- Vercel should now be able to find and use the `yarn.lock` file
-  Vercel deployments should succeed on next trigger

Why This Happened

The development environment uses a shared `yarn.lock` via symlink to optimize disk space and ensure consistency across projects. However, this approach doesn't work with cloud deployment platforms like Vercel that require all files to be present in the repository.

Best Practice

For deployable projects, always use actual files instead of symlinks for critical dependency files like:

- `yarn.lock`
- `package-lock.json`
- `pnpm-lock.yaml`

Testing the Fix

To verify the fix works on Vercel:

1. Push a new commit or trigger a manual redeploy in Vercel
2. Monitor the build logs
3. Confirm the build completes without “ENOENT: `yarn.lock`” errors

Additional Notes

- The `yarn.lock` file is now properly tracked in git
- File size: 445,184 bytes (12,323 lines)
- Mode changed from symlink (120000) to regular file (100644)
- This fix applies to both your deployment environments

Status: Issue resolved and deployed to GitHub

Next Step: Vercel should automatically detect the new commit and retry the deployment. The build should now succeed.