

Vercel Build Fix - Database Connection Issue

Problem

The Vercel build was failing with error:

```
Error: P1001: Can't reach database server at db-5ebca9571.db003.hosteddb.reai.io:5432
```

Root Cause

The build command in `package.json` included `prisma db push --accept-data-loss`, which attempted to connect to the database during the build process. Vercel's build environment cannot reach external databases during the build phase.

Solution Applied

Changed build command from:

```
"build": "prisma generate && prisma db push --accept-data-loss && next build"
```

To:

```
"build": "prisma generate && next build"
```

What This Means

- During Build:** Vercel will now only generate the Prisma client (type definitions) without connecting to the database
- During Runtime:** Your application will connect to the database when it actually runs
- Database Schema:** Must be synced manually before deployment (already done)

Vercel Environment Variables Required

Make sure these environment variables are set in your Vercel project settings:

Required Variables

- DATABASE_URL**
 - Value: Your PostgreSQL connection string
 - Format: `postgresql://user:password@db-5ebca9571.db003.hosteddb.reai.io:5432/5ebca9571?schema=public`
- NEXTAUTH_SECRET**
 - Value: Your NextAuth secret key
 - Should be a long random string (minimum 32 characters)

3. **NEXTAUTH_URL** (Vercel sets this automatically, but you can override)
 - Value: Your deployed URL (e.g., <https://portfolio-intelligence.co.uk>)

How to Set Environment Variables in Vercel

1. Go to your Vercel dashboard
2. Select your project
3. Navigate to **Settings → Environment Variables**
4. Add each variable with appropriate values
5. Make sure to add them for **Production**, **Preview**, and **Development** environments

Database Schema Management

Important: Database changes should be made BEFORE deploying to Vercel

For local development:

```
cd nextjs_space
yarn prisma db push
```

For production database:

- Run the same command locally with your production DATABASE_URL
- Or use Prisma Migrate for better version control:

```
yarn prisma migrate dev --name description_of_changes
yarn prisma migrate deploy # For production
```

Testing the Fix

The fix has been pushed to GitHub. Vercel should automatically:

1. Detect the new commit
2. Trigger a new build
3. Successfully build without database connection errors

Status

- Build command updated
- Changes committed to Git
- Changes pushed to GitHub (commit: e6cadc3)

Next Steps

1.  **DONE:** Build command fixed and pushed to GitHub
2.  **PENDING:** Verify environment variables are set in Vercel
3.  **PENDING:** Vercel will automatically rebuild from the latest commit
4.  **PENDING:** Verify the deployment succeeds

Troubleshooting

If Build Still Fails

1. Check that environment variables are properly set in Vercel
2. Ensure DATABASE_URL uses SSL parameters if required
3. Verify the database is accessible from the internet

If Runtime Errors Occur

1. Check that DATABASE_URL is correct in environment variables
2. Verify database schema is up to date
3. Check application logs in Vercel dashboard

Support

If you encounter any issues after this fix:

1. Check Vercel build logs for specific error messages
2. Verify all environment variables are set correctly
3. Ensure your database allows connections from Vercel's IP ranges

Update: Additional Fix Applied (December 10, 2024)

Second Build Error Encountered

After fixing the database connection issue, Vercel encountered another error:

```
Error: ENOENT: no such file or directory, stat '/vercel/path0/nextjs_space/yarn.lock'
```

Root Cause

The `yarn.lock` file was a **symlink** (symbolic link) pointing to a local system path:

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

When code is pushed to GitHub, Git preserves the symlink structure, but the target path doesn't exist in Vercel's build environment, causing the build to fail.

Solution Applied

Replaced the symlink with the actual file:

```
cd nextjs_space
rm yarn.lock
cp /opt/hostedapp/node/root/app/yarn.lock yarn.lock
```

This creates a real `yarn.lock` file (515KB, 14,557 lines) that can be properly committed to Git and used by Vercel.

Changes Pushed

- Commit:** "Fix Vercel build: replace yarn.lock symlink with actual file"
- File Size:** 515KB (proper lock file)
- Status:** Pushed to GitHub (commit: 7963a58)

What This Means

- `yarn.lock` is now a real file that travels with your repository
- Vercel can properly read the lock file during builds
- Dependency versions are now properly locked across all environments

Prevention

Important: Never use `npm` or create symlinks for `node_modules` or lock files. Always use:

- `yarn add <package>` to add dependencies
- `yarn install` to install dependencies
- Let `yarn` manage the lock file naturally

Complete Fix Summary

Two issues were resolved:

1. **Database Connection During Build**
 - Removed `prisma db push` from build command
 - Build now only generates Prisma client
2. **Yarn Lock File Symlink**
 - Replaced symlink with actual file
 - Ensures lock file availability in Vercel

Both fixes have been committed and pushed to GitHub. Your Vercel deployment should now build successfully! 🎉