Vercel CLI EPIPE Error Troubleshooting Guide

Problem

The `vercel login` command fails with an EPIPE (broken pipe) error, preventing authentication and environment variable pulling.

Root Causes

1. **Node.js version compatibility issues**
2. **Network/proxy interference**
3. **TLS certificate verification problems**
4. **Stdin/stdout piping issues**

Solutions (in order of preference)

Solution 1: Token-Based Authentication (Recommended)

```
# Get token from https://vercel.com/account/tokens
export VERCEL_TOKEN=your_token_here
npx vercel login --token $VERCEL_TOKEN
```

Solution 2: Update Node.js (if using older version)

```
# Check current version
node -v

# Update to latest LTS (if needed)
nvm install --lts
nvm use --lts
```

Solution 3: Fix TLS Issues (if behind corporate proxy)

```
# Temporary workaround (USE WITH CAUTION - only for debugging)
export NODE_TLS_REJECT_UNAUTHORIZED=0
npx vercel login

# Better solution: Install corporate certificates or update Node.js
```

Solution 4: Use Different Terminal/Shell

```
# Try with different shell
bash -c "npx vercel login"

# Or try with explicit stdio handling
npx vercel login < /dev/null
```

Solution 5: Manual Environment Setup

If all else fails, manually create .env.local with Neon credentials from Vercel dashboard.

Verification Steps

1. Test authentication: `npx vercel whoami`
2. Pull environment variables: `npx vercel env pull .env.local`
3. Test database connection: `node test-neon-connection.js`

Environment Variables Needed

- `DATABASE_URL` (primary connection string)
- `DATABASE_URL_UNPOOLED` (direct connection for migrations)

Next Steps After Authentication

1. Update .env.local with actual Neon credentials
2. Run database migrations: `npx prisma migrate deploy`
3. Test local development: `npm run dev`
4. Deploy to Vercel: `npx vercel --prod`