Vercel Deployment Workflow

This document explains how to safely push code to Vercel without build errors.

Quick Commands

Option 1: Use the Safe Push Script

```
cd nextjs_space
./scripts/push-to-vercel.sh "your commit message"
```

This script:
1. Validates your code against the Prisma schema
2. Fixes known issues (like next.config.js)
3. Commits and pushes to GitHub
4. Vercel auto-deploys

Option 2: Manual Validation

```
# Validate changed files
node scripts/validate-before-push.js

# Validate specific files against schema
node scripts/schema-validator.js app/api/some-route/route.ts

# Validate all changed files
node scripts/schema-validator.js --changed
```

Common Errors & Fixes

1. Invalid Enum Value

Error: `Invalid enum value 'REFSBOSource.REALTOR_COM'`
Fix: Check `prisma/schema.prisma` for valid enum values

2. Missing Field

Error: `Property 'userId' does not exist on type 'REExpiredListing'`
Fix: The field name might be different (e.g., `assignedUserId`)

3. path0/path0 Error

Error: `ENOENT: no such file or directory, lstat '/vercel/path0/path0/.next'`
Fix: Remove `outputFileTracingRoot` from `next.config.js`

Checkpoints

We use GitHub + Vercel instead of Abacus checkpoints:
- **GitHub commits** = version history

- **Vercel deployments** = rollback points
- No need for `build_and_save_nextjs_project_checkpoint`

Pre-Push Checklist

- [ ] Run `node scripts/validate-before-push.js`
- [ ] Verify enum values match `schema.prisma`
- [ ] Check that `next.config.js` has no `outputFileTracingRoot`
- [ ] Test imports resolve correctly (`@/lib/...` not `../../`)