

PR Title

Fix Prisma Shim Bugs + Architecture Analysis

PR Description



Bug Fixes

Critical Prisma Shim Bugs Fixed

Fixed multiple critical bugs in `lib/prisma-shim.ts` that were causing database queries to fail:

- 1. Missing Query Assignment:** In `findUnique`, `findFirst`, and `findMany` functions, the query builder wasn't being reassigned after adding `.eq()` conditions. This caused WHERE clauses to be completely ignored!

Before:

```
typescript
query.eq(key, value); // ❌ query not updated
```

After:

```
typescript
query = query.eq(key, value); // ✅ query properly updated
```

- 1. Missing `_count` Support:** Added basic support for Prisma's `include: { _count: { ... } }` syntax to prevent API errors. Currently returns placeholder values - full implementation with actual counting can be added later if needed.

Impact

These bugs likely caused:

- Client creation issues (500 errors)
- Incorrect query results throughout the app
- WHERE conditions being silently ignored



Architecture Analysis

Added comprehensive analysis document: `ARCHITECTURE_ANALYSIS.md`

Current Issue

The app has a **dual structure** that doesn't match the Writgo business model:

- **Client-level:** Platforms connected directly to client (Facebook, Instagram, etc.)
- **Project-level:** Separate projects with duplicate settings (WordPress, keywords, etc.)

This creates confusion because:

- A Client already has platforms connected
- Projects don't have their own platform integrations

- Settings are duplicated
- Too complex for local business owners

Writgo Business Model

- Target: Local Dutch service providers (dentists, cleaners, etc.)
- 4 packages: Instapper (€197), Starter (€297), Groei (€497), Dominant (€797)
- Difference: Number of platforms + posts per month
- Promise: Fully autonomous, zero-touch solution

Should be: 1 Client = 1 Account = 1 Package = X Platforms = Y Posts/month

Proposal: Simplified Architecture

Option 1 (Recommended):

- Remove Project layer from new workflow
- 1 Client = direct platform connections + content
- Projects remain for backward compatibility but hidden for new clients
- Simplified admin flow: Client → Package → Platforms → Done

Option 2 (Middle ground):

- Keep Projects but auto-create default project per client
 - Hide project complexity in UI
 - More flexible for future edge cases
-

User Report

User reported 500 error when creating new client with console showing:

```
Failed to load resource: /api/admin/clients[1]
```

The prisma-shim bugs likely contributed to this issue.

Testing Needed

After merge:

1. Test client creation end-to-end
 2. Verify WHERE clauses work correctly in all queries
 3. Test client listing with search/filters
-

Decision Needed

User feedback requested on architecture simplification:

- Should we implement Option 1 (remove Projects from workflow)?
- Or Option 2 (auto-create default Projects)?
- Or keep current structure?

See `ARCHITECTURE_ANALYSIS.md` for full details and comparison.



Files Changed

- `nextjs_space/lib/prisma-shim.ts` : Bug fixes
 - `ARCHITECTURE_ANALYSIS.md` : New documentation
-



Links

- Branch: `feature/fix-prisma-shim-and-client-architecture`
- Base: `main`