🔍 WORKIGOM RAILWAY DEPLOYMENT DIAGNOSIS

Generated: October 23, 2025
Repository: volkanakbulut73/workigom
Branch: master

📊 INVESTIGATION RESULTS

✅ What's Working

1. **Git Tracking - CONFIRMED**
   ```bash
   # All Prisma files ARE tracked in Git:
   backend/prisma.config.ts
   backend/prisma/migrations/20251022231535_exit/migration.sql
   backend/prisma/migrations/migration_lock.toml
   backend/prisma/schema.prisma
   backend/prisma/seed.ts
   ```

2. **Local Files - EXIST**
   ```bash
   backend/prisma/
   ├── migrations/
   │   ├── 20251022231535_exit/
   │   └── migration_lock.toml
   ├── schema.prisma (4,788 bytes)
   └── seed.ts (6,493 bytes)
   ```

3. **Git Status - CLEAN**
   - All changes committed
   - No untracked files affecting prisma

🚨 ROOT CAUSE IDENTIFIED

Problem #1: .dockerignore is Blocking Backend Directory

File: `.dockerignore`

```
node_modules
npm-debug.log
dist
.env
.git
.gitignore
README.md
.vscode
.idea
backend          ⚠️ THIS IS THE CULPRIT!
*.swp
*.swo
```

Impact: When Docker builds the image, the `.dockerignore` file tells Docker to **completely ignore the `backend` directory**, which means:
- `COPY backend/package*.json ./` → FAILS (no backend directory in build context)
- `COPY backend/prisma ./prisma/` → FAILS (no backend directory in build context)
- `COPY backend/ .` → FAILS (no backend directory in build context)

Problem #2: Railway is Building the Wrong Dockerfile

From the logs (`logs.1761239968123.json`), Railway is attempting to build the **FRONTEND Dockerfile** (at root `/Dockerfile`) instead of the backend Dockerfile (`backend/Dockerfile`).

Evidence from logs:

```
ERROR: "/nginx.conf": not found
```

This error only appears in the frontend Dockerfile, not the backend one.

📁 CURRENT PROJECT STRUCTURE

```
workigom/
├── .dockerignore              ❌ Contains "backend" - blocking entire directory
├── Dockerfile                 📄 Frontend Dockerfile (nginx)
├── package.json               📦 Frontend dependencies
├── src/                       📁 Frontend React code
├── backend/
│   ├── Dockerfile              📄 Backend Dockerfile (Node.js)
│   ├── package.json            📦 Backend dependencies
│   ├── prisma/                ✅ EXISTS and tracked in Git
│   │   ├── schema.prisma
│   │   ├── seed.ts
│   │   └── migrations/
│   └── src/                   📁 Backend TypeScript code
└── .git/                     ✅ All files committed
```

🎯 SOLUTION OPTIONS

Option A: Quick Fix - Fix .dockerignore (RECOMMENDED)

Pros:
- Minimal changes
- Keeps organized structure
- Fast to implement

Cons:
- Still requires Railway to use correct Dockerfile path

Steps:
1. Remove `backend` from `.dockerignore`
2. Configure Railway to use `backend/Dockerfile` as the build path
3. Ensure Railway's root directory is set to `/` (not `/backend`)

Option B: Restructure - Move Backend to Root (USER SUGGESTED)

Pros:
- Simplifies Docker context
- Railway can easily find all files
- No .dockerignore conflicts
- Clearer deployment path

Cons:
- Requires reorganizing entire project
- More files to move and test
- Potential for breaking imports/paths

Steps:
1. Move all `backend/` files to root
2. Update package.json scripts
3. Update import paths if needed
4. Create new Dockerfile at root
5. Remove old backend directory
6. Test locally
7. Commit and push

🔧 RECOMMENDED ACTION PLAN

PREFERRED: Option A - Quick Fix

```
# 1. Fix .dockerignore
# Remove "backend" line from .dockerignore

# 2. Update Railway settings:
# - Go to Railway service settings
# - Set "Dockerfile Path" to: backend/Dockerfile
# - Set "Root Directory" to: /
# - Or use railway.json to configure:
{
  "build": {
    "builder": "DOCKERFILE",
    "dockerfilePath": "backend/Dockerfile"
  }
}

# 3. Push changes
git add .dockerignore
git commit -m "fix: remove backend from dockerignore"
git push origin master
```

ALTERNATIVE: Option B - Restructure

If Option A doesn't work (due to Railway context issues), then restructure:

```
# This will move backend to root
# Detailed steps will be provided if you choose this option
```

📋 .gitignore vs .dockerignore Analysis

`.gitignore` (NO ISSUES)

```
node_modules/
dist/
build/
.env
.env.local
.DS_Store
*.log
```

✅ Not blocking any source files

`.dockerignore` (HAS ISSUES)

```
backend  ← BLOCKING ENTIRE BACKEND DIRECTORY!
```

❌ This prevents Docker from seeing backend files during build

---

🧪 NEXT STEPS

To Proceed with Quick Fix (Option A):

1. Say: **“Fix the .dockerignore file”**

To Proceed with Restructure (Option B):

1. Say: **“Restructure the project - move backend to root”**

To Investigate Further:

1. Say: **“Show me more details about [specific aspect]”**

📝 NOTES

- **All Prisma files are in Git** ✅
- **Backend directory exists locally** ✅
- **The problem is Docker build context**, not Git tracking
- **Railway appears to be building frontend Dockerfile**, not backend
- **.dockerignore is actively blocking the backend directory** ❌