Frontend Network Error (ERR_NETWORK) - Diagnosis and Solution

Date: October 25, 2025
Issue: Frontend deployed at https://workigom.vercel.app cannot connect to backend
Error: `ERR_NETWORK: Failed to load resource: net::ERR_CONNECTION_REFUSED`

🔍 Root Cause Analysis

1. Current Configuration Issue

The frontend is trying to connect to the **WRONG backend URL**:

Current Configuration in `.env.production`:

```
VITE_BACKEND_URL=https://workigom.onrender.com
```

Actual Backend URL:

```
https://workigom-backend.onrender.com
```

Notice the difference: The actual backend has `-backend` in the URL, but the configuration doesn't.

2. How the Frontend API Configuration Works

Looking at `/home/ubuntu/workigom/src-frontend/lib/api.ts`:

```
const getApiUrl = () => {
  const backendUrl = import.meta.env.VITE_BACKEND_URL || 'http://localhost:3001';
  const baseUrl = backendUrl.endsWith('/api') ? backendUrl : `${backendUrl}/api`;
  return baseUrl;
};
```

This means:
- The frontend reads `VITE_BACKEND_URL` environment variable
- It automatically appends `/api` to create the full API URL
- Current wrong URL: `https://workigom.onrender.com/api` ❌
- Correct URL should be: `https://workigom-backend.onrender.com/api` ✅

3. Backend Health Check Verification

The backend at `https://workigom-backend.onrender.com` is **healthy and working**:

```
{
  "success": true,
  "message": "Workigom API is running",
  "timestamp": "2025-10-25T00:23:49.341Z",
  "database": "connected"
}
```


4. CORS Configuration

The backend CORS is **correctly configured** and already allows:
- `https://workigom.vercel.app` ✅
- `https://workigom-frontend.vercel.app` ✅
- `http://localhost:5173` ✅

So CORS is NOT the problem. The problem is the frontend is trying to connect to a non-existent backend URL.

✅ Solution: Fix Vercel Environment Variables

Step 1: Update Vercel Environment Variable

1. Go to **Vercel Dashboard**: https://vercel.com/dashboard
2. Select your **workigom** project (currently at https://workigom.vercel.app)
3. Navigate to **Settings → Environment Variables**
4. Find the `VITE_BACKEND_URL` variable (or create it if it doesn't exist)
5. Set the value to:
   `https://workigom-backend.onrender.com`
6. Make sure it's enabled for:
   - ✅ Production
   - ✅ Preview
   - ✅ Development (optional)

Step 2: Trigger a New Deployment

After setting the environment variable:

1. Go to **Deployments** tab in Vercel
2. Click on the three dots menu (...) on the latest deployment
3. Select **Redeploy**
4. OR simply push a small change to trigger a new deployment

Step 3: Verify the Fix

After redeployment, open the browser console at https://workigom.vercel.app and you should see:

```
🔧 API Configuration: {
  VITE_BACKEND_URL: "https://workigom-backend.onrender.com",
  finalApiUrl: "https://workigom-backend.onrender.com/api",
  mode: "production"
}
```

Then test the API calls - they should now work successfully!

📋 Alternative Solution: Update .env.production in Code

If you prefer to fix this in the code itself:

Option A: Update .env.production file

Edit `/home/ubuntu/workigom/src-frontend/.env.production`:

```
# Change from:
VITE_BACKEND_URL=https://workigom.onrender.com

# To:
VITE_BACKEND_URL=https://workigom-backend.onrender.com
```

Then commit and push:

```
cd /home/ubuntu/workigom
git add src-frontend/.env.production
git commit -m "fix: Update backend URL to workigom-backend.onrender.com"
git push origin master
```

Vercel will automatically deploy with the correct URL.

Option B: Update .env.example for documentation

Also update `.env.example` to reflect the correct production URL:

```
# Edit /home/ubuntu/workigom/src-frontend/.env.example
VITE_BACKEND_URL=http://localhost:3001

# Add comment for production:
# For production (Vercel):
# VITE_BACKEND_URL=https://workigom-backend.onrender.com
```

🎯 Quick Fix Command (If you want to update via code)

```
# Navigate to the project
cd /home/ubuntu/workigom

# Update .env.production
cat > src-frontend/.env.production << 'EOF'
# Production environment variables
# Note: In Vercel, set VITE_BACKEND_URL in the dashboard under Settings > Environment
Variables
VITE_BACKEND_URL=https://workigom-backend.onrender.com
EOF

# Commit and push
git add src-frontend/.env.production
git commit -m "fix: Correct backend URL to workigom-backend.onrender.com"
git push origin master
```

🔧 Debugging Tips

Check Current Environment Variable in Browser

Open browser console at https://workigom.vercel.app and run:

```
console.log(import.meta.env.VITE_BACKEND_URL);
```

Test Backend Directly

Test the backend API directly in browser:

```
https://workigom-backend.onrender.com/api/health
```

Should return:

```
{
  "success": true,
  "message": "Workigom API is running",
  "database": "connected"
}
```

Check Network Tab in DevTools

1. Open DevTools (F12)
2. Go to Network tab
3. Try to use the app (e.g., login, fetch jobs)
4. Look at the failed request
5. Check the Request URL - it should be pointing to `workigom-backend.onrender.com`

📊 Summary

Issue	Current State	Required State
Frontend URL	✅ https://workigom.vercel.app	✅ Correct
Backend URL	✅ https://workigom-backend.onrender.com	✅ Correct
CORS Config	✅ Allows workigom.vercel.app	✅ Correct
Environment Variable	❌ Points to workigom.onrender.com	❌ **NEEDS FIX**

The ONLY issue is the environment variable value in Vercel!

🚀 Recommended Action

I recommend using the Vercel Dashboard method (Step 1 above) because:

1. ✅ Faster - no code commit needed
2. ✅ Can be done immediately
3. ✅ No git history pollution
4. ✅ Can test different URLs easily
5. ✅ Follows best practices (environment variables should be in platform settings, not in code)

⚠️ Important Notes

1. **DO NOT commit `.env` files with production secrets to git** (currently `.env.production` is in the repo, which is okay since it only has the URL, but be careful)
2. **The `.env.production` file** is used during the build process by Vite, but Vercel environment variables override it
3. **After setting the environment variable in Vercel**, you MUST redeploy for the changes to take effect
4. **The `/api` suffix** is automatically added by the frontend code, so don't include it in the environment variable

---

📞 Next Steps

1. Set the environment variable in Vercel Dashboard
2. Redeploy the application
3. Test the application at https://workigom.vercel.app
4. Verify API calls are working in browser console
5. ✅ Problem solved!

Need help? If you encounter any issues after following these steps, check:
- Vercel deployment logs for build errors
- Browser console for any new errors
- Network tab to verify the correct URL is being called