CORS Configuration Fix Summary

Date: October 25, 2025

Problem

The frontend deployed on Vercel at `https://workigom-frontend.vercel.app` was unable to make API calls to the backend on Render at `https://workigom.onrender.com` due to CORS (Cross-Origin Resource Sharing) errors.

Error Messages Observed:

- `ERR_CONNECTION_REFUSED`
- `Failed to fetch Jobs`
- `Failed to load resource: net::ERR_CONNECTION_REFUSED`
- Network error in browser console when trying to fetch from `localhost:3001/api/jobs`

Root Cause

The backend CORS configuration did not include the correct Vercel frontend URL (`https://workigom-frontend.vercel.app`) in its allowed origins list.

Solution Applied

1. Updated CORS Configuration in `backend/src/app.ts`

Added the Vercel frontend URL to the allowed origins array:

```
// CORS configuration
const allowedOrigins = [
  'http://localhost:5173',
  'https://workigom.vercel.app',
  'https://workigom-frontend.vercel.app',  // ✅ Added this line
  process.env.CORS_ORIGIN
].filter(Boolean);
```

2. Updated Local Environment Variable

Updated `.env` file with the correct CORS_ORIGIN value:

```
# CORS Configuration
CORS_ORIGIN=https://workigom-frontend.vercel.app
```

Note: The `.env` file is not committed to Git (it's in `.gitignore`), so you'll need to update the environment variable in your Render dashboard.

3. Git Commit and Push

- Committed changes with message: "Fix CORS configuration for Vercel frontend deployment"

- Successfully pushed to GitHub repository: `volkanakbulut73/workigom`

Next Steps

On Render Dashboard:

1. Go to your Render dashboard: https://dashboard.render.com
2. Select your `workigom` backend service
3. Go to "Environment" tab
4. Add or update the environment variable:
   - **Key:** `CORS_ORIGIN`
   - **Value:** `https://workigom-frontend.vercel.app`
5. Click "Save Changes"
6. Render will automatically redeploy your backend with the new environment variable

On Vercel Dashboard:

1. Ensure your frontend environment variable `VITE_BACKEND_URL` is set to:
   `https://workigom.onrender.com`
2. If you need to redeploy, go to the Deployments tab and click "Redeploy"

Verification

After updating the environment variables on Render:

1. **Test the backend directly:**
   ```bash
   curl https://workigom.onrender.com/api/health
   ```
   Should return a successful response.

2. **Test from frontend:**
   - Visit `https://workigom-frontend.vercel.app`
   - Open browser DevTools (F12) → Console tab
   - Check for CORS errors (should be gone)
   - Verify that API calls to `/api/jobs` and other endpoints work

3. **Check the Network tab:**
   - Open DevTools → Network tab
   - Filter by "XHR" or "Fetch"
   - API calls to `https://workigom.onrender.com` should show successful responses (200 OK)

Technical Details

CORS Configuration Explained:

- **Purpose:** Allows the frontend (running on one domain) to make requests to the backend (running on another domain)
- **Security:** Only explicitly allowed origins can access the API
- **Credentials:** The configuration includes `credentials: true` to allow cookies and authentication headers

Current Allowed Origins:

1. `http://localhost:5173` - Local development
2. `https://workigom.vercel.app` - Alternative Vercel deployment
3. `https://workigom-frontend.vercel.app` - Main Vercel frontend
4. Any origin specified in `CORS_ORIGIN` environment variable

Files Modified

- ✅ `backend/src/app.ts` - Added Vercel frontend URL to allowed origins
- ✅ `.env` - Updated CORS_ORIGIN (local only, not in Git)
- ✅ Committed and pushed to GitHub

Status

✅ **COMPLETED** - All code changes have been made and pushed to GitHub.

⚠️ **ACTION REQUIRED** - You need to update the `CORS_ORIGIN` environment variable on Render dashboard for the changes to take effect in production.

Additional Notes

If Issues Persist:

1. **Check Backend Logs on Render:**
   - Look for any startup errors
   - Verify the environment variable is loaded
   - Check for CORS-related log messages

2. **Verify Backend Environment Variable:**
   javascript
   ```
   console.log('CORS_ORIGIN:', process.env.CORS_ORIGIN);
   ```

3. **Test Backend API Endpoint:**
   bash
   ```
   curl -H "Origin: https://workigom-frontend.vercel.app" \
        -H "Access-Control-Request-Method: GET" \
        -H "Access-Control-Request-Headers: Content-Type" \
        -X OPTIONS \
        https://workigom.onrender.com/api/jobs
   ```
   Should return CORS headers in response.

4. **Check Frontend Environment:**
   - Verify `VITE_BACKEND_URL` is correctly set on Vercel
   - Ensure no hardcoded `localhost` URLs in frontend code

Production Checklist:

- [ ] Update `CORS_ORIGIN` on Render dashboard
- [ ] Verify backend redeploys successfully
- [ ] Test API calls from Vercel frontend

- [ ] Check browser console for errors
- [ ] Verify authentication works (if applicable)
- [ ] Test all main features (jobs listing, applications, etc.)

Need Help?

If you encounter any issues after updating the environment variables, please share:

1. Render deployment logs
2. Browser console errors
3. Network tab screenshots showing the failed requests