🔑 Environment Variables Quick Reference

📦 Backend Environment Variables (Render)

Required Variables

Variable	**Value**	**Description**
`NODE_ENV`	`production`	Set to production mode
`DATABASE_URL`	`postgresql://user:pass@host:5432/db`	PostgreSQL connection string from Render
`JWT_SECRET`	`<random-32-char-hex>`	Secret key for JWT tokens
`JWT_EXPIRES_IN`	`7d`	JWT token expiration time
`JWT_REFRESH_SECRET`	`<random-32-char-hex>`	Secret key for refresh tokens
`JWT_REFRESH_EXPIRES_IN`	`30d`	Refresh token expiration time
`CORS_ORIGIN`	`https://workigom.vercel.app,https://workigom-frontend1.onrender.com`	Comma-separated list of allowed origins

Optional Variables

Variable	Default	Description
`RATE_LIMIT_WINDOW_MS`	`900000`	Rate limit window (15 minutes)
`RATE_LIMIT_MAX_REQUESTS`	`100`	Max requests per window
`MAX_FILE_SIZE`	`5242880`	Max file upload size (5MB)
`ALLOWED_FILE_TYPES`	`image/jpeg,image/png,image/jpg,application/pdf`	Allowed file types
`SMTP_HOST`	-	Email server host (optional)
`SMTP_PORT`	`587`	Email server port
`SMTP_USER`	-	Email username
`SMTP_PASS`	-	Email password
`EMAIL_FROM`	`noreply@workigom.com`	From email address

How to Generate Secrets

JWT_SECRET:

```
node -e "console.log(require('crypto').randomBytes(32).toString('hex'))"
```

JWT_REFRESH_SECRET:

```
node -e "console.log(require('crypto').randomBytes(32).toString('hex'))"
```

Or use an online generator: https://generate-secret.vercel.app/32

🎨 Frontend Environment Variables (Render or Vercel)

Required Variable

Variable	Value	Description
`VITE_BACKEND_URL`	`https://workigom-backend.onrender.com`	Backend API URL (WITHOUT /api suffix)

⚠️ **CRITICAL**:
- Do NOT include `/api` at the end
- The frontend code automatically adds `/api` to construct API endpoints

- Example: `https://workigom-backend.onrender.com` ✅
- NOT: `https://workigom-backend.onrender.com/api` ❌

🔄 Copy-Paste Templates

Backend (Render Dashboard)

```
NODE_ENV=production
DATABASE_URL=<paste-from-render-postgresql>
JWT_SECRET=<generate-using-command-above>
JWT_EXPIRES_IN=7d
JWT_REFRESH_SECRET=<generate-using-command-above>
JWT_REFRESH_EXPIRES_IN=30d
CORS_ORIGIN=https://workigom.vercel.app,https://workigom-frontend1.onrender.com
RATE_LIMIT_WINDOW_MS=900000
RATE_LIMIT_MAX_REQUESTS=100
MAX_FILE_SIZE=5242880
ALLOWED_FILE_TYPES=image/jpeg,image/png,image/jpg,application/pdf
```

Frontend (Render Dashboard)

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

Frontend (Vercel Dashboard)

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

🎯 Setting Variables in Render

1. Go to Render Dashboard (https://dashboard.render.com/)
2. Click on your service (backend or frontend)
3. Go to **Environment** tab on the left
4. Click **Add Environment Variable**
5. Enter **Key** and **Value**
6. Click **Save Changes**
7. Service will automatically redeploy

🎯 Setting Variables in Vercel

1. Go to Vercel Dashboard (https://vercel.com/dashboard)
2. Select your project
3. Go to **Settings** → **Environment Variables**
4. Click **Add New**
5. Enter **Key** and **Value**

6. Select environments: **Production**, **Preview**, **Development**
7. Click **Save**
8. Redeploy your project

✅ Verification Checklist

Backend

- [ ] `DATABASE_URL` is set (from Render PostgreSQL)
- [ ] `JWT_SECRET` is a secure random string (32+ characters)
- [ ] `JWT_REFRESH_SECRET` is a different secure random string
- [ ] `CORS_ORIGIN` includes all your frontend domains
- [ ] No spaces in `CORS_ORIGIN` (only commas)
- [ ] Backend is deployed and running
- [ ] Health check works: `curl https://workigom-backend.onrender.com/api/health`

Frontend

- [ ] `VITE_BACKEND_URL` is set to backend URL
- [ ] `VITE_BACKEND_URL` does NOT end with `/api`
- [ ] Frontend is deployed and accessible
- [ ] Browser console shows correct API URL in logs
- [ ] No CORS errors in browser console
- [ ] Login request goes to correct backend URL

🔍 Quick Debugging

Check Backend Logs

```
# In Render Dashboard → Backend Service → Logs tab
# Look for:
🔒 CORS Allowed Origins: [...]
✅ CORS: Allowing origin https://...
```

Check Frontend Console

```
// In Browser DevTools → Console tab
// Should see:
🔧 API Configuration: {
  VITE_BACKEND_URL: "https://workigom-backend.onrender.com",
  finalApiUrl: "https://workigom-backend.onrender.com/api",
  mode: "production"
}
```

Test Backend API

```
# Health check
curl https://workigom-backend.onrender.com/api/health

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

🚨 Common Mistakes

1. ❌ Adding `/api` to `VITE_BACKEND_URL` → Frontend will request `/api/api/...`
2. ❌ Spaces in `CORS_ORIGIN` → CORS will fail silently
3. ❌ Using same secret for `JWT_SECRET` and `JWT_REFRESH_SECRET` → Security risk
4. ❌ Not redeploying after changing environment variables → Changes won't apply
5. ❌ Wrong `DATABASE_URL` → Backend won't start

📞 Need Help?

If authentication still doesn't work:

1. Check backend logs for CORS errors
2. Check frontend console for API URL
3. Verify all environment variables are set
4. Try hard refresh (Ctrl+F5) in browser
5. Clear browser cache
6. Redeploy both services with "Clear build cache"

Last Updated: November 1, 2025