

Deployment Guide - Off-Plan Vastgoed Platform

Complete guide for deploying the off-plan real estate platform to Netlify at **spanje-rendement.nl**.



Table of Contents

1. [Prerequisites](#)
2. [Environment Variables](#)
3. [Netlify Setup](#)
4. [Domain Configuration](#)
5. [Data Migration](#)
6. [Testing Checklist](#)
7. [Troubleshooting](#)
8. [Post-Deployment](#)



Prerequisites

Before deploying, ensure you have:

- GitHub account with repository access
- Netlify account (Pro tier for Blobs feature)
- Domain access for **spanje-rendement.nl**
- DeepSeek API key for AI matching
- Generated JWT secret key



Environment Variables

Required Environment Variables

Set these in **Netlify Dashboard → Site Settings → Environment Variables**:

1. **JWT_SECRET (Required)**

- **Purpose:** Secure JWT token signing and verification
- **Generation:**

```
bash
openssl rand -base64 32
```
- **Example:** Xk7mP9qR2tY5vW8zA1bC3dE6fH9jK0lN4oP7qS0tU3v=
- **Security:** Keep this secret! Never commit to Git.

2. **DEEPSEEK_API_KEY (Required)**

- **Purpose:** AI-powered investor matching

- **Get Key:** DeepSeek Platform (<https://platform.deepseek.com>)
- **Example:** sk-1234567890abcdef1234567890abcdef

3. NODE_VERSION (Recommended)

- **Value:** 18 or 20
- **Purpose:** Ensure Node.js 18+ for ES modules support

Setting Environment Variables in Netlify

1. Go to **Site Settings → Environment Variables**
 2. Click **Add a variable**
 3. Add each variable:
 - Variable: JWT_SECRET
 - Value: [your-generated-secret]
 - Scopes: Production, Deploy Previews, Branch Deploys
 4. Click **Create variable**
 5. Repeat for DEEPSEEK_API_KEY and NODE_VERSION
-

Netlify Setup

Option 1: Auto-Deploy from GitHub (Recommended)

1. **Connect Repository**
 - Log in to [Netlify](https://app.netlify.com) (<https://app.netlify.com>)
 - Click **Add new site → Import an existing project**
 - Choose **GitHub** and authorize Netlify
 - Select repository: Jali2587/off-plan-vastgoed-deep-seek
2. **Configure Build Settings**
 - Build command: echo 'No build step required'
 - Publish directory: .
 - Functions directory: netlify/functions
 - **Note:** netlify.toml will override these settings
3. **Deploy Site**
 - Click **Deploy site**
 - Netlify will automatically install dependencies from package.json
 - Initial deployment URL: [https://\[random-name\].netlify.app](https://[random-name].netlify.app)

Option 2: Manual Deploy via Netlify CLI

```
# Install Netlify CLI
npm install -g netlify-cli

# Login to Netlify
netlify login

# Initialize site
netlify init

# Deploy to production
netlify deploy --prod
```

Domain Configuration

Configure spanje-rendement.nl

1. Add Custom Domain in Netlify

- Go to **Domain Settings** → **Add a domain**
- Enter: `spanje-rendement.nl`
- Click **Verify**

2. DNS Configuration

Option A: Use Netlify DNS (Recommended)

- Netlify will provide nameservers (e.g., `dns1.p01.nsone.net`)
- Update nameservers at your domain registrar
- Netlify automatically configures DNS and SSL

Option B: External DNS Provider

- Add these DNS records at your provider:

Type	Name	Value	TTL
A	@	75.2.60.5	3600
CNAME	www	[your-site].netlify.app	3600

1. SSL Certificate

- Netlify automatically provisions Let's Encrypt SSL
- HTTPS enabled by default
- Certificate auto-renewal every 90 days

2. Force HTTPS

- **Domain Settings** → **HTTPS** → Enable **Force HTTPS**

Data Migration

Initial Data Setup

The application uses **Netlify Blobs** for data persistence. Initial data from `/data` directory needs to be migrated.

Automatic Migration on First Run

Functions will automatically create Blobs from existing JSON files when first accessed. Initial data files include:

- `users.json` - User accounts
- `projects.json` - Property projects
- `profiles.json` - Investor profiles
- `favorites.json` - User favorites
- `reservations.json` - Reservations

Manual Data Import (Optional)

To pre-populate Blobs before first use:

```
// Use Netlify CLI
netlify blobs:set data/users.json --input data/users.json
netlify blobs:set data/projects.json --input data/projects.json
netlify blobs:set data/profiles.json --input data/profiles.json
netlify blobs:set data/favorites.json --input data/favorites.json
netlify blobs:set data/reservations.json --input data/reservations.json
```

Testing Checklist

Pre-Deployment Tests

- [] All environment variables configured
- [] `package.json` contains all dependencies
- [] `netlify.toml` properly configured
- [] No hardcoded secrets in code
- [] Git repository clean (no sensitive data)

Post-Deployment Tests

1. Basic Functionality

- [] Site loads at custom domain
- [] HTTPS working (no certificate warnings)
- [] All static assets load correctly

2. Authentication

- [] Register new user (`/auth-register`)
- [] Login existing user (`/auth-login`)
- [] Token verification works (`/auth-verify`)
- [] JWT expiration handled correctly

3. User Features

- [] View projects list
- [] Save favorites
- [] Load favorites
- [] Create investor profile
- [] Update investor profile
- [] Load investor profile

4. AI Matching

- [] AI match endpoint responds (`/ai-match`)
- [] DeepSeek API integration works
- [] Match results returned correctly

5. Admin Features (if admin user exists)

- [] View all users (`/admin-users-list`)
- [] View all profiles (`/admin-profile-list`)

- [] View reservations (/admin-reservations-list)
- [] View investor stats (/admin-investor-stats)
- [] View project stats (/admin-project-stats)
- [] Delete project (/admin-delete-project)

6. Security

- [] CSP headers applied (check browser console)
- [] X-Frame-Options: DENY
- [] HTTPS enforced
- [] Unauthorized requests return 401
- [] Non-admin requests to admin endpoints return 403

Testing Tools

cURL Examples

```
# Test registration
curl -X POST https://spanje-rendement.nl/.netlify/functions/auth-register \
-H "Content-Type: application/json" \
-d '{"email":"test@example.com","role":"investor"}'

# Test login
curl -X POST https://spanje-rendement.nl/.netlify/functions/auth-login \
-H "Content-Type: application/json" \
-d '{"email":"test@example.com"}'

# Test protected endpoint
curl https://spanje-rendement.nl/.netlify/functions/favorites-load \
-H "Authorization: Bearer YOUR_JWT_TOKEN"
```

Browser Console Tests

```
// Test API from browser console
async function testAPI() {
  // Register
  const registerRes = await fetch('/.netlify/functions/auth-register', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ email: 'test@example.com', role: 'investor' })
  });
  const { token } = await registerRes.json();

  // Load favorites
  const favRes = await fetch('/.netlify/functions/favorites-load', {
    headers: { 'Authorization': `Bearer ${token}` }
  });
  console.log(await favRes.json());
}

testAPI();
```



Troubleshooting

Common Issues

1. “Missing JWT_SECRET” Error

Problem: Environment variable not set

Solution: Add `JWT_SECRET` in Netlify Environment Variables

Verify: Redeploy site after adding variable

2. “401 Unauthorized” on Protected Endpoints

Problem: Invalid or expired JWT token

Solution:

- Re-login to get fresh token
- Check token expiration (7 days)
- Verify `JWT_SECRET` hasn't changed

3. “500 Internal Server Error” on Blob Operations

Problem: Netlify Blobs not enabled (requires Pro tier)

Solution:

- Upgrade to Netlify Pro
- Verify Blobs feature is enabled in Site Settings

4. DeepSeek API Errors

Problem: Invalid API key or rate limit

Solution:

- Verify `DEEPSEEK_API_KEY` is correct
- Check API quota at DeepSeek dashboard
- Review function logs for specific error

5. CSP Blocking Resources

Problem: Content Security Policy blocking scripts/styles

Solution:

- Check browser console for CSP violations
- Update `Content-Security-Policy` in `netlify.toml`
- Add allowed domains to appropriate directives

Checking Logs

Netlify Function Logs

1. Go to **Site Dashboard → Functions**
2. Click on specific function
3. View **Function logs** for errors

Real-time Log Streaming

```
# Using Netlify CLI
netlify functions:log [function-name]

# Stream all function logs
netlify functions:log --stream
```

Netlify Blobs Inspection

```
# List all blobs
netlify blobs:list

# Get specific blob
netlify blobs:get data/users.json
```

Post-Deployment

1. Create Admin User

```
# Use Netlify Blobs CLI or function to create admin
curl -X POST https://spanje-rendement.nl/.netlify/functions/auth-register \
-H "Content-Type: application/json" \
-d '{"email":"admin@spanje-rendement.nl","role":"admin"}'
```

2. Monitor Performance

- **Netlify Analytics:** Enable in Site Settings
- **Function Execution:** Monitor in Dashboard
- **Bandwidth Usage:** Check monthly limits

3. Set Up Alerts

- **Downtime Monitoring:** Use UptimeRobot or similar
- **Error Tracking:** Consider Sentry integration
- **Budget Alerts:** Set Netlify bandwidth alerts

4. Regular Maintenance

- **Weekly:** Check function logs for errors
- **Monthly:** Review bandwidth and function usage
- **Quarterly:** Rotate JWT_SECRET for security
- **As Needed:** Update dependencies in package.json

5. Backup Strategy

Netlify Blobs are durable, but consider periodic backups:

```
# Export all blobs
netlify blobs:get data/users.json > backup/users-$(date +%Y%m%d).json
netlify blobs:get data/projects.json > backup/projects-$(date +%Y%m%d).json
netlify blobs:get data/profiles.json > backup/profiles-$(date +%Y%m%d).json
netlify blobs:get data/favorites.json > backup/favorites-$(date +%Y%m%d).json
netlify blobs:get data/reservations.json > backup/reservations-$(date +%Y%m%d).json
```

Additional Resources

- **Netlify Documentation:** <https://docs.netlify.com>

- **Netlify Blobs:** <https://docs.netlify.com/blobs/overview/>
 - **Netlify Functions:** <https://docs.netlify.com/functions/overview/>
 - **DeepSeek API:** <https://platform.deepseek.com/docs>
 - **Let's Encrypt SSL:** <https://letsencrypt.org>
-

Support

Need Help?

1. **Check Netlify Status:** <https://www.netlifystatus.com>
 2. **Community Forums:** <https://answers.netlify.com>
 3. **GitHub Issues:** Report bugs in repository
 4. **Netlify Support:** Available for Pro tier customers
-



Notes

- **Staging Environment:** Consider creating a staging site for testing
 - **Git Workflow:** Use feature branches for changes, deploy to preview first
 - **Version Control:** Never commit `.env` files or secrets
 - **GDPR Compliance:** Ensure data handling complies with privacy laws
-

Last Updated: December 2024

Version: 1.0.0

Deployment Status:  Production-Ready