

# Deployment Guide - Advies N Consultancy BV Website

---

## Pre-Deployment Checklist

---

### 1. Environment Setup

```
# Production environment variables
DATABASE_URL=postgresql://user:password@host:5432/database
NEXTAUTH_URL=https://adviesnconsultancy.nl
NEXTAUTH_SECRET=secure-random-string
GOOGLE_ANALYTICS_ID=G-XXXXXXXXXX
```

### 2. Database Preparation

```
# Apply schema to production database
npx prisma db push

# Seed initial data (optional)
npx prisma db seed
```

### 3. Build Process

```
# Install dependencies
yarn install --frozen-lockfile

# Type checking
yarn tsc --noEmit

# Production build
yarn build

# Test build
yarn start
```

## SSL/TLS Configuration Fix

---

### Critical Issue Resolution

**Problem:** ERR\_SSL\_VERSION\_OR\_CIPHER\_MISMATCH bij externe toegang

### Immediate Actions Required

#### 1. Hosting Provider Configuration

Contact hosting provider om de volgende SSL/TLS instellingen te verifiëren:

**Required TLS Configuration:**

- TLS Version: 1.2 & 1.3 support
- Cipher Suites: Modern cipher suites enabled
- Certificate Chain: Complete chain including intermediates
- HSTS: Compatible with application headers

**2. Cloudflare Settings (if applicable)****SSL/TLS Tab Settings:**

- Encryption Mode: "Full (strict)"
- TLS Version: Minimum TLS 1.2
- Cipher Suites: Modern only
- HSTS: Disable or configure to match app

**3. Application-Level Fixes****Option A: Disable HSTS Temporarily**

In `middleware.ts`, comment out HSTS:

```
// TEMPORARY FIX - Comment out HSTS
// if (request.url.startsWith('https://')) {
//   response.headers.set('Strict-Transport-Security', 'max-age=3600')
// }
```

**Option B: Relaxed Security Headers**

```
// More permissive CSP for SSL troubleshooting
const cspPolicy = [
  "default-src 'self' https: data:",
  "script-src 'self' 'unsafe-inline' 'unsafe-eval' https:",
  "style-src 'self' 'unsafe-inline' https:",
  "font-src 'self' https: data:",
  "img-src 'self' data: https: blob:",
  "connect-src 'self' https: wss: ws:",
  "frame-src 'self' https:",
  "object-src 'none'",
  "base-uri 'self'",
  "form-action 'self' https:"
]
```

**4. Testing SSL Configuration**

```
# Test various TLS versions
curl -v --tlsv1.2 https://www.adviesnconsultancy.nl
curl -v --tlsv1.3 https://www.adviesnconsultancy.nl

# Check certificate chain
openssl s_client -connect www.adviesnconsultancy.nl:443 -showcerts

# Test cipher suites
nmap --script ssl-enum-ciphers -p 443 www.adviesnconsultancy.nl

# Browser compatibility test
curl -H "User-Agent: Mozilla/5.0" https://www.adviesnconsultancy.nl
```

# Platform-Specific Deployment

---

## Vercel Deployment

```
# Install Vercel CLI
npm i -g vercel

# Deploy
vercel --prod
```

**vercel.json configuration:**

```
{
  "functions": {
    "app/api/**/*.ts": {
      "maxDuration": 30
    }
  },
  "headers": [
    {
      "source": "/*",
      "headers": [
        {
          "key": "X-SSL-Protocol",
          "value": "TLSv1.2,TLSv1.3"
        }
      ]
    }
  ]
}
```

## Netlify Deployment

**netlify.toml:**

```
[build]
command = "cd app && yarn build"
publish = "app/.next"

[[headers]]
for = "/*"
[headers.values]
X-SSL-Debug = "enabled"
Strict-Transport-Security = "max-age=31536000; includeSubDomains"
```

## Docker Deployment

```
FROM node:18-alpine

WORKDIR /app
COPY app/package.json app/yarn.lock ./
RUN yarn install --frozen-lockfile

COPY app/ ./
RUN yarn build

EXPOSE 3000
CMD ["yarn", "start"]
```

## Post-Deployment Monitoring

### Health Checks

```
# Application health
curl https://adviesnconsultancy.nl/api/health

# SSL verification
curl -I https://adviesnconsultancy.nl

# Performance test
curl -o /dev/null -s -w "%{time_total}\n" https://adviesnconsultancy.nl
```

### Monitoring Dashboard

1. **SSL Certificate Expiry:** Monitor via tools like SSL Labs
2. **Performance Metrics:** Core Web Vitals via Google Analytics
3. **Error Monitoring:** Application errors via logs
4. **Uptime Monitoring:** External service monitoring

## Rollback Procedure

### Quick Rollback

```
# Revert to previous deployment
git revert <commit-hash>
git push origin main

# Or restore from backup
git checkout <previous-stable-commit>
git push origin main --force
```

### Database Rollback (if needed)

```
# Create backup before deployment
pg_dump $DATABASE_URL > backup.sql

# Restore if needed
psql $DATABASE_URL < backup.sql
```

# Security Hardening Post-Deployment

---

## 1. Security Headers Verification

```
# Check security headers
curl -I https://adviesnconsultancy.nl | grep -E "(Strict-Transport|Content-Security|X-Frame|X-XSS)"
```

## 2. Performance Optimization

```
# Analyze bundle
yarn build && yarn analyze

# Check lighthouse score
npx lighthouse https://adviesnconsultancy.nl --output=json
```

## 3. SEO Verification

- [ ] Sitemap accessible: <https://adviesnconsultancy.nl/sitemap.xml>
- [ ] Robots.txt: <https://adviesnconsultancy.nl/robots.txt>
- [ ] Meta tags present on all pages
- [ ] Structured data valid

# Support & Troubleshooting

---

## Common Deployment Issues

### Issue 1: Build Failures

```
# Clear cache and rebuild
rm -rf .next node_modules
yarn install
yarn build
```

### Issue 2: Database Connection

```
# Test database connection
npx prisma studio
npx prisma db push --force-reset
```

### Issue 3: SSL Handshake Failures

- Check hosting provider SSL configuration
- Verify certificate chain completeness
- Test with different TLS versions
- Disable strict security headers temporarily

## Emergency Contacts

- **Hosting Provider Support:** [Contact details]
- **SSL Certificate Provider:** [Contact details]
- **DNS Provider:** [Contact details]

---

**Deployment Status:** ⚠️ Ready (SSL issue requires resolution)

**Last Updated:** July 12, 2025