

Royal Farms Poultry Management System

Production Deployment Guide

Table of Contents

1. [Prerequisites](#)
 2. [Database Setup](#)
 3. [Environment Variables](#)
 4. [Deployment Options](#)
 5. [Domain Configuration](#)
 6. [Post-Deployment Checklist](#)
 7. [Troubleshooting](#)
-

Prerequisites

Before deploying, ensure you have:

- [] Node.js 18+ installed on your local machine
 - [] PostgreSQL database (hosted or self-managed)
 - [] Domain name (e.g., royalfarms.com)
 - [] Hosting account (GoDaddy, Truehost, Vercel, Railway, etc.)
-

Database Setup

Option A: Managed PostgreSQL (Recommended)

Recommended Providers:

- **Neon** (neon.tech) - Free tier available, serverless
- **Supabase** (supabase.com) - Free tier, includes Auth features
- **Railway** (railway.app) - Easy setup, \$5/month
- **PlanetScale** (planetscale.com) - MySQL alternative
- **AWS RDS** - Enterprise-grade

Setup Steps:

1. Create account on chosen provider
2. Create new PostgreSQL database
3. Copy the connection string (looks like):

```
postgresql://username:password@hostname:5432/database_name
```

4. Save this for your `.env` file

Option B: Self-Hosted PostgreSQL

If using GoDaddy/Truehost VPS:

```
# Install PostgreSQL
sudo apt update
sudo apt install postgresql postgresql-contrib

# Create database and user
sudo -u postgres psql
CREATE DATABASE royalfarms;
CREATE USER royalfarms_user WITH ENCRYPTED PASSWORD 'your_secure_password';
GRANT ALL PRIVILEGES ON DATABASE royalfarms TO royalfarms_user;
\q
```

Environment Variables

Create a `.env` file with these variables:

```
# Database
DATABASE_URL="postgresql://username:password@hostname:5432/database_name?
schema=public"

# Authentication (generate a secure random string)
NEXTAUTH_SECRET="your-32-character-secret-key-here"
NEXTAUTH_URL="https://royalfarms.com"

# Optional: Google Analytics
NEXT_PUBLIC_GA_MEASUREMENT_ID="G-XXXXXXXXXX"
```

Generate NEXTAUTH_SECRET:

```
openssl rand -base64 32
```

Deployment Options

Option 1: Vercel (Easiest - Recommended)

Pros: Free tier, automatic SSL, easy setup

Cons: Limited server resources on free tier

Steps:

1. Push code to GitHub/GitLab
2. Go to vercel.com (<https://vercel.com>)
3. Import your repository
4. Add environment variables in Vercel dashboard
5. Deploy!

Custom Domain:

- In Vercel Dashboard → Project → Settings → Domains
- Add `royalfarms.com`
- Follow DNS instructions provided

Option 2: Railway (Simple + Database)

Pros: Includes PostgreSQL, simple deployment

Cons: Paid after free tier (\$5/month)

Steps:

1. Go to railway.app (https://railway.app)
 2. Create new project
 3. Add PostgreSQL service
 4. Add new service → Deploy from GitHub
 5. Railway auto-detects Next.js
 6. Add environment variables
 7. Deploy
-

Option 3: GoDaddy VPS/cPanel

Pros: Full control, existing domain management

Cons: More complex setup, manual SSL

A. Using Node.js on VPS

```
# 1. SSH into your server
ssh user@your-server-ip

# 2. Install Node.js 18+
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs

# 3. Install PM2 (process manager)
sudo npm install -g pm2

# 4. Clone or upload your project
git clone your-repo-url /var/www/royalfarms
cd /var/www/royalfarms/nextjs_space

# 5. Install dependencies
npm install

# 6. Create .env file
nano .env
# Add your environment variables

# 7. Generate Prisma client and push schema
npx prisma generate
npx prisma db push

# 8. Build the application
npm run build

# 9. Start with PM2
PORT=3000 pm2 start npm --name "royalfarms" -- start
pm2 save
pm2 startup
```

B. Nginx Reverse Proxy Configuration

```
# /etc/nginx/sites-available/royalfarms.com
server {
    listen 80;
    server_name royalfarms.com www.royalfarms.com;

    location / {
        proxy_pass http://127.0.0.1:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
    }
}
```

```
# Enable site and get SSL
sudo ln -s /etc/nginx/sites-available/royalfarms.com /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx

# Install Certbot for free SSL
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d royalfarms.com -d www.royalfarms.com
```

Option 4: Truehost Hosting

For VPS/Cloud Hosting:

Follow GoDaddy VPS instructions above.

For Shared Hosting (Node.js support required):

1. Check if Truehost plan supports Node.js
2. Use cPanel's Node.js selector if available
3. Upload files via FTP/File Manager
4. Configure application in cPanel

Domain Configuration

Step 1: DNS Settings

In your domain registrar (GoDaddy/Truehost):

| Type | Name | Value | TTL |
|-------|------|----------------|------|
| A | @ | YOUR_SERVER_IP | 3600 |
| A | www | YOUR_SERVER_IP | 3600 |
| CNAME | www | royalfarms.com | 3600 |

For Vercel/Railway:

They provide specific DNS records - follow their instructions.

Step 2: SSL Certificate

Automatic (Vercel/Railway): SSL included

Manual (VPS):

```
sudo certbot --nginx -d royalfarms.com -d www.royalfarms.com
```

Step 3: Update NEXTAUTH_URL

After domain is live, update:

```
NEXTAUTH_URL="https://royalfarms.com"
```

Post-Deployment Checklist

1. Initialize Database

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

# Create production admin user
npx tsx scripts/clean-database.ts
```

2. Verify Application

- [] Visit <https://royalfarms.com>
- [] Login with admin@royalfarms.com / RoyalFarms2026!
- [] **IMMEDIATELY** change the admin password
- [] Create additional user accounts
- [] Add your farm sites
- [] Add livestock types (if needed beyond defaults)

3. Security Hardening

- [] Change default admin password
- [] Enable 2FA (if implementing)
- [] Review user permissions
- [] Set up database backups

- [] Enable monitoring/logging

4. Backup Strategy

```
# Database backup (run daily via cron)
pg_dump $DATABASE_URL > backup_$(date +%Y%m%d).sql

# Cron job example (daily at 2 AM)
0 2 * * * pg_dump $DATABASE_URL > /backups/royalfarms_$(date +%Y%m%d).sql
```

Troubleshooting

Common Issues

1. “Cannot connect to database”

- Verify DATABASE_URL is correct
- Check database server is running
- Ensure IP is whitelisted (for managed DBs)

2. “NEXTAUTH_URL mismatch”

- Update NEXTAUTH_URL to match your domain
- Include https:// prefix
- No trailing slash

3. “502 Bad Gateway”

- Check if Node.js app is running: `pm2 status`
- Check logs: `pm2 logs royalfarms`
- Verify Nginx config: `sudo nginx -t`

4. “SSL Certificate Error”

- Run Certbot: `sudo certbot --nginx`
- Check certificate: `sudo certbot certificates`

5. “Prisma: Schema out of sync”

```
npx prisma db push
npx prisma generate
```

Support

For technical support:

- Review application logs
- Check database connectivity
- Verify environment variables

Quick Start Commands

```
# Local development
cd nextjs_space
npm install
npm run dev

# Production build
npm run build
npm start

# Database operations
npx prisma db push          # Apply schema
npx prisma studio           # Visual database browser
npx tsx scripts/clean-database.ts # Clean for production

# Process management (PM2)
pm2 start npm --name "royalfarms" -- start
pm2 status
pm2 logs royalfarms
pm2 restart royalfarms
```

Document Version: 2.0.0

Last Updated: January 2026

System: Royal Farms Poultry Management System