🚀 DiscHound Deployment Guide

Complete step-by-step deployment instructions for your DiscHound e-commerce store.

📋 Pre-Deployment Checklist

- [ ] GitHub repository created and code pushed
- [ ] Stripe account created (test keys obtained)
- [ ] Domain name registered (if using custom domain)
- [ ] Production database prepared
- [ ] Environment variables documented

🔧 Vercel Deployment (Recommended)

Step 1: Prepare Repository

```
# Initialize git repository
git init

# Add all files
git add .

# Initial commit
git commit -m "Initial DiscHound store deployment"

# Add remote repository
git remote add origin https://github.com/yourusername/dischound-store.git

# Push to GitHub
git push -u origin main
```

Step 2: Deploy to Vercel

1. **Sign up/Login to Vercel**
   - Visit vercel.com (https://vercel.com)
   - Sign up with GitHub account

2. **Import Project**
   - Click “New Project”
   - Select your GitHub repository
   - Vercel auto-detects Next.js configuration

3. **Configure Build Settings** (Usually auto-detected)
   `Framework Preset: Next.js`
   `Root Directory: ./nextjs_space`
   `Build Command: yarn build`
   `Output Directory: .next`
   `Install Command: yarn install`

4. **Deploy**
   - Click “Deploy”
   - Wait for build to complete (2-3 minutes)

Step 3: Configure Environment Variables

In your Vercel project dashboard:

1. **Go to Settings → Environment Variables**

2. **Add Required Variables:**
   `DATABASE_URL → your_postgresql_connection_string`
   `STRIPE_PUBLISHABLE_KEY → pk_test_your_publishable_key`
   `STRIPE_SECRET_KEY → sk_test_your_secret_key`
   `STRIPE_WEBHOOK_SECRET → whsec_your_webhook_secret`
   `NEXTAUTH_URL → https://your-app.vercel.app`
   `NODE_ENV → production`

3. **Redeploy**
   - Go to Deployments tab
   - Click “Redeploy” to apply environment variables

🗄️ Database Setup

Option 1: Vercel Postgres (Recommended)

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

# Login to Vercel
vercel login

# Create Postgres database
vercel postgres create dischound-db

# Get connection string
vercel postgres connect dischound-db
```

Option 2: External PostgreSQL

Popular providers:
- **Neon**: neon.tech (https://neon.tech) - Free tier available
- **Supabase**: supabase.com (https://supabase.com) - Free tier available
- **PlanetScale**: planetscale.com (https://planetscale.com) - MySQL alternative
- **Railway**: railway.app (https://railway.app) - PostgreSQL hosting

Run Database Migrations

```
# In your local environment
DATABASE_URL="your_production_connection_string" yarn prisma db push
DATABASE_URL="your_production_connection_string" yarn prisma db seed
```

🌐 Custom Domain Configuration

Step 1: Add Domain in Vercel

1. **Go to Project Settings → Domains**
2. **Add Domain**: Enter `dischound.com` (or your domain)
3. **Vercel provides DNS instructions**

Step 2: Configure DNS Records

At your domain registrar (GoDaddy, Namecheap, Cloudflare, etc.):

For Apex Domain (dischound.com):

```
Type: A
Name: @
Value: 76.76.19.61
TTL: 300 (or Auto)
```

For WWW Subdomain (www.dischound.com):

```
Type: CNAME
Name: www
Value: cname.vercel-dns.com
TTL: 300 (or Auto)
```

Step 3: Update Environment Variables

```
NEXTAUTH_URL=https://dischound.com
```

Step 4: SSL Certificate

- Vercel automatically provides SSL certificates
- Usually active within 24-48 hours
- Check at SSL Checker (https://www.sslshopper.com/ssl-checker.html)

💳 Stripe Configuration

Step 1: Create Stripe Account

1. Visit stripe.com (https://stripe.com)
2. Create account and verify business information
3. Complete account setup

Step 2: Get API Keys

Test Mode (For Testing):
1. Go to Stripe Dashboard (https://dashboard.stripe.com/test/apikeys)
2. Copy Publishable Key (pk_test_...)
3. Copy Secret Key (sk_test_...)

Live Mode (For Production):
1. Activate your account (complete verification)

2. Switch to Live mode in dashboard
3. Copy Live Keys (pk_live_..., sk_live_...)

Step 3: Configure Webhooks

1. **Go to Stripe Dashboard → Developers → Webhooks**
2. **Click "Add Endpoint"**
3. **Endpoint URL**: `https://dischound.com/api/stripe/webhook`
4. **Select Events**:
   - `payment_intent.succeeded`
   - `payment_intent.payment_failed`
5. **Click "Add Endpoint"**
6. **Copy Webhook Signing Secret** (whsec_...)

Step 4: Test Payments

Test Card Numbers:

```
Success: 4242 4242 4242 4242
Decline: 4000 0000 0000 0002
Requires Authentication: 4000 0025 0000 3155
```

📊 Analytics & Monitoring

Google Analytics 4 (Optional)

```
# Install GA4
yarn add @next/third-parties

# Add to app/layout.tsx
import { GoogleAnalytics } from '@next/third-parties/google'

<GoogleAnalytics gaId="GA_MEASUREMENT_ID" />
```

Error Monitoring (Optional)

- **Sentry**: sentry.io (https://sentry.io) - Error tracking
- **LogRocket**: logrocket.com (https://logrocket.com) - Session replay
- **Vercel Analytics**: Built-in performance monitoring

🔍 Testing Deployment

Step 1: Functionality Test

- [ ] Homepage loads correctly
- [ ] Product pages display properly
- [ ] Cart functionality works
- [ ] Checkout form submits
- [ ] Order confirmation appears

Step 2: Payment Test

- [ ] Use Stripe test card: 4242 4242 4242 4242

- [ ] Complete test purchase
- [ ] Verify order appears in database
- [ ] Check Stripe dashboard for payment

Step 3: Performance Test

- [ ] PageSpeed Insights (https://pagespeed.web.dev/)
- [ ] GTmetrix (https://gtmetrix.com/)
- [ ] Mobile responsiveness test

🚨 Troubleshooting

Common Issues

Build Failures:

```
# Check build logs in Vercel dashboard
# Common fixes:
- Ensure all dependencies in package.json
- Check TypeScript errors
- Verify environment variables
```

Database Connection:

```
# Test connection locally
yarn prisma db push

# Check connection string format
# Ensure database is accessible from Vercel
```

Stripe Integration:

```
# Verify webhook endpoint
# Check webhook signing secret
# Test with Stripe CLI
stripe listen --forward-to localhost:3000/api/stripe/webhook
```

Getting Help

1. **Vercel Support**: vercel.com/help (https://vercel.com/help)
2. **Stripe Support**: support.stripe.com (https://support.stripe.com)
3. **Next.js Docs**: nextjs.org/docs (https://nextjs.org/docs)

🎉 Go Live Checklist

Before switching to live mode:

- [ ] Complete Stripe account verification
- [ ] Update to live Stripe keys
- [ ] Test with real payment (small amount)
- [ ] Verify order processing
- [ ] Check email confirmations

- [ ] Update webhook endpoints
- [ ] Set up monitoring/alerts
- [ ] Backup database
- [ ] Document admin processes

🔄 Maintenance

Regular Tasks

- Monitor order volume and payments
- Update product inventory
- Check error logs weekly
- Backup database monthly
- Update dependencies quarterly
- Review analytics monthly

Scaling Considerations

- Database performance optimization
- CDN for static assets
- Caching strategies
- Rate limiting implementation
- Load balancing for high traffic

🎯 Success Metrics

Week 1 Targets:
- Site loads in < 3 seconds
- 0 payment processing errors
- Mobile responsiveness: 100%
- SSL certificate active

Month 1 Targets:
- Conversion rate: > 2%
- Average order value: > $30
- Cart abandonment: < 70%
- Customer support tickets: < 5/week

Ready to launch your DiscHound store! 🚀