



Vercel Deployment Guide for DiscHound

This guide walks you through deploying your DiscHound e-commerce store to Vercel.

Prerequisites Checklist

Before starting, make sure you have:

- [] GitHub account
- [] Vercel account (free tier is fine)
- [] Stripe API keys (already configured ✓)
- [] Domain purchased: diskhound.com (already done ✓)

Step 1: Push Code to GitHub

1.1 Create a New GitHub Repository

1. Go to **<https://github.com/new>**
2. **Repository name:** `dischound-store`
3. **Description:** "DiscHound - NFC Disc Golf Tracking Store"
4. **Visibility:** Private (recommended) or Public
5. Click **"Create repository"**

1.2 Push Your Code

Run these commands in your terminal:

```
cd /home/ubuntu/dischound_store/nextjs_space

# Initialize git (if not already done)
git init

# Add all files
git add .

# Make first commit
git commit -m "Initial DiscHound store - ready for deployment"

# Add your GitHub repository as remote
# Replace YOUR_USERNAME with your actual GitHub username
git remote add origin https://github.com/YOUR_USERNAME/dischound-store.git

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

Note: GitHub may ask for credentials. Use a Personal Access Token (not password):

- Go to: <https://github.com/settings/tokens>
- Click "Generate new token (classic)"

- Select scopes: `repo` (full control)
 - Copy the token and use it as your password
-

Step 2: Create PostgreSQL Database on Vercel

2.1 Create Database

1. Go to <https://vercel.com/dashboard>
2. Click **"Storage"** in the top navigation
3. Click **"Create Database"**
4. Select **"Postgres"**
5. **Database name:** `dischound-production`
6. **Region:** Choose closest to your customers (e.g., US East for USA)
7. Click **"Create"**

2.2 Get Database URL

1. After creation, click on your database
 2. Go to **".env.local"** tab
 3. Copy the `POSTGRES_PRISMA_URL` value
 4. Save this - you'll need it in the next step
-

Step 3: Deploy to Vercel

3.1 Import Your GitHub Repository

1. Go to <https://vercel.com/new>
2. Click **"Import Git Repository"**
3. Find and select `dischound-store`
4. Click **"Import"**

3.2 Configure Project Settings

Framework Preset: Next.js (should auto-detect)

Root Directory: `./` (default)

Build Command: `yarn build` (default)

Output Directory: `.next` (default)

3.3 Add Environment Variables

Click **"Environment Variables"** and add these:

Name	Value	Notes
DATABASE_URL	postgresql://...	Paste the POSTGRES_PRISMA_URL you copied
STRIPE_PUBLISHABLE_KEY	pk_live... or pk_test...	Your Stripe publishable key
STRIPE_SECRET_KEY	sk_live... or sk_test...	Your Stripe secret key
STRIPE_WEBHOOK_SECRET	Leave blank for now	We'll add this later
NEXTAUTH_URL	https://diskhound.com	Your custom domain
NEXTAUTH_SECRET	Generate random string	See below

To generate NEXTAUTH_SECRET:

```
openssl rand -base64 32
```

Or use: <https://generate-secret.vercel.app/32>

3.4 Deploy

1. Click **“Deploy”**
2. Wait 2-3 minutes for build to complete
3. You'll get a URL like: `dischound-store-abc123.vercel.app`

Step 4: Initialize Database Schema

4.1 Run Prisma Migrations

After deployment, you need to set up the database schema:

1. In Vercel Dashboard, go to your project
2. Click **“Settings”** → **“Functions”**
3. Or use Vercel CLI:

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

# Login
vercel login

# Link project
cd /home/ubuntu/diskhound_store/nextjs_space
vercel link

# Run Prisma migration
vercel env pull .env.production
yarn prisma migrate deploy
yarn prisma db seed
```

Alternatively, you can use the Vercel PostgreSQL web interface:

1. Go to Storage → Your Database → Query
2. Copy and paste the schema from `prisma/schema.prisma` (converted to SQL)

Step 5: Configure Custom Domain (diskhound.com)

5.1 Add Domain in Vercel

1. In your Vercel project dashboard
2. Click **“Settings”** → **“Domains”**
3. Enter: `diskhound.com`
4. Click **“Add”**
5. Also add: `www.diskhound.com` (recommended)

5.2 Vercel Will Show You DNS Records

You’ll see something like:

```
Type: A
Name: @
Value: 76.76.21.21

Type: CNAME
Name: www
Value: cname.vercel-dns.com
```

5.3 Configure DNS in GoDaddy

1. **Log into GoDaddy:** <https://dcc.godaddy.com/>
2. Find **diskhound.com** and click **“DNS”**
3. **Delete existing records** for @ and www (the parking page ones)
4. **Add A Record:**
 - Type: A
 - Name: @
 - Value: 76.76.21.21
 - TTL: 600 seconds

5. Add CNAME Record:

- Type: CNAME
- Name: www (lowercase!)
- Points to: cname.vercel-dns.com
- TTL: 1 Hour

6. Click **“Save”**

5.4 Wait for DNS Propagation

- Takes 10-30 minutes (up to 48 hours max)
- Check status: <https://dnschecker.org/#A/diskhound.com>
- Vercel will auto-provision SSL certificate once DNS is verified

Step 6: Configure Stripe Webhook

6.1 Create Webhook in Stripe

1. Go to <https://dashboard.stripe.com/webhooks>
2. Click **“Add endpoint”**
3. **Endpoint URL:** <https://diskhound.com/api/stripe/webhook>
4. **Events to send:**
 - payment_intent.succeeded
 - payment_intent.payment_failed
5. Click **“Add endpoint”**

6.2 Get Webhook Signing Secret

1. After creating, click on the webhook
2. Click **“Reveal”** next to “Signing secret”
3. Copy the value (starts with whsec_...)

6.3 Add to Vercel

1. Go to Vercel Dashboard → Your Project
2. **Settings** → **“Environment Variables”**
3. Click **“Add New”**
4. Name: STRIPE_WEBHOOK_SECRET
5. Value: whsec_... (paste the signing secret)
6. Click **“Save”**
7. **Redeploy** the project for changes to take effect

Step 7: Test Your Deployment

7.1 Basic Tests

1. Visit <https://diskhound.com>
2. Verify:
 - [] Homepage loads correctly

- [] Products page shows all items
- [] Add to cart works
- [] Cart page displays items
- [] Checkout flow works

7.2 Test Payment Flow (Use Test Mode First!)

If using Stripe Test Mode:

- Test card: 4242 4242 4242 4242
- Expiry: Any future date
- CVC: Any 3 digits
- ZIP: Any 5 digits

Test a purchase:

1. Add items to cart
2. Go to checkout
3. Fill in details with test card
4. Complete purchase
5. Verify order confirmation page loads
6. Check Stripe Dashboard for payment record
7. Verify webhook received in Stripe Dashboard → Webhooks

Step 8: Switch to Live Mode (When Ready)

8.1 Update Environment Variables

1. Go to Vercel → Settings → Environment Variables
2. Update:
 - STRIPE_PUBLISHABLE_KEY → Use pk_live_... key
 - STRIPE_SECRET_KEY → Use sk_live_... key
3. Redeploy

8.2 Update Webhook

1. Create new webhook in Stripe for live mode
2. Use same endpoint: `https://diskhound.com/api/stripe/webhook`
3. Get new signing secret
4. Update STRIPE_WEBHOOK_SECRET in Vercel
5. Redeploy

Troubleshooting

Build Fails

Error: Module not found

- Check that all dependencies are in package.json
- Run yarn install locally first

Error: Prisma Client not generated

- Add this to package.json :

```

json
"scripts": {
  "postinstall": "prisma generate"
}

```

Domain Not Working

Shows 404 or blank page:

- Check NEXTAUTH_URL is set correctly
- Verify DNS records in GoDaddy
- Wait 30 minutes for DNS propagation
- Clear browser cache

SSL Error:

- Wait 10-15 minutes after DNS verification
- Vercel auto-provisions SSL (Let's Encrypt)
- Check Vercel dashboard for SSL status

Stripe Webhook Not Working

Payments succeed but order status not updated:

- Verify STRIPE_WEBHOOK_SECRET is set
- Check webhook endpoint in Stripe Dashboard
- View webhook logs in Stripe → Webhooks → Your endpoint
- Check Vercel Function logs for errors

Database Connection Issues

Error: Can't reach database server

- Verify DATABASE_URL is correct
- Check Vercel Postgres is running
- Ensure Prisma schema is migrated

Monitoring & Maintenance

View Logs

Vercel Logs:

1. Project Dashboard → Deployments
2. Click on latest deployment
3. View "Build Logs" or "Function Logs"

Stripe Logs:

1. Dashboard → Developers → Logs
2. Filter by payment intents or webhooks

Performance Monitoring

Vercel Analytics:

- Enable in Project Settings → Analytics
- Free tier: 100k events/month
- Shows page views, performance, and errors

Update Environment Variables

1. Settings → Environment Variables
 2. Edit any variable
 3. **Must redeploy** for changes to take effect
-

Security Checklist

Before going live:

- [] All environment variables are set
 - [] Using HTTPS (SSL enabled)
 - [] Stripe webhook secret configured
 - [] Test mode disabled (using live keys)
 - [] Database backups enabled (Vercel auto-backups Postgres)
 - [] Error tracking configured (optional: Sentry)
-

Quick Commands Reference

```
# Deploy new version
git add .
git commit -m "Update description"
git push

# Vercel will auto-deploy on push

# Manual deploy via CLI
vercel --prod

# View logs
vercel logs

# Pull environment variables
vercel env pull

# Run Prisma migrations on production
vercel env pull .env.production
yarn prisma migrate deploy
```

Support Resources

- **Vercel Docs:** <https://vercel.com/docs>
 - **Next.js Docs:** <https://nextjs.org/docs>
 - **Stripe Docs:** <https://stripe.com/docs>
 - **Prisma Docs:** <https://www.prisma.io/docs>
-

Deployment Complete!

Your DiscHound store is now live at:

- **<https://diskhound.com>**
- **<https://www.diskhound.com>**

Next steps:

1. Test all functionality end-to-end
2. Switch to live Stripe keys when ready
3. Configure dropshipping with supplier
4. Launch marketing campaigns!

Need help? Check the troubleshooting section or Vercel's support documentation.