🚀 Vercel Deployment Guide - The Mindful Musicpreneur App

This guide will walk you through deploying your Next.js application to Vercel step-by-step.

📋 Prerequisites

Before you begin, make sure you have:

- ✅ A GitHub account (Sign up here (https://github.com/signup))
- ✅ A Vercel account (Sign up here (https://vercel.com/signup))
- ✅ A PostgreSQL database (we recommend Neon (https://neon.tech) or Supabase (https://supabase.com))
- ✅ A Stripe account (Sign up here (https://dashboard.stripe.com/register))
- ✅ A Resend account for emails (Sign up here (https://resend.com/signup))

📁 Step 1: Push Code to GitHub

Your code is already committed locally. Now you need to push it to GitHub.

1.1 Create a New GitHub Repository

1. Go to GitHub (https://github.com) and log in
2. Click the **"+"** icon in the top right corner
3. Select **"New repository"**
4. Fill in the details:
   - **Repository name**: `mindful-musicpreneur-app` (or your preferred name)
   - **Description**: "The 360° Mindful Musicpreneur System - E-commerce platform for female musicians"
   - **Visibility**: Choose **Private** (recommended) or Public
   - **DO NOT** initialize with README, .gitignore, or license (we already have these)
5. Click **"Create repository"**

1.2 Push Your Code

After creating the repository, GitHub will show you commands. Run these in your terminal:

```
# Navigate to your project
cd /home/ubuntu/mindful_musicpreneur_app/nextjs_space

# Add the remote repository (replace YOUR_USERNAME with your GitHub username)
git remote add origin https://github.com/YOUR_USERNAME/mindful-musicpreneur-app.git

# Push your code
git push -u origin master
```

Note: If you're using a different branch name (like `main`), use that instead of `master`.

If prompted for credentials, you may need to:
- Use a Personal Access Token (https://github.com/settings/tokens) instead of your password
- Or set up SSH keys (https://docs.github.com/en/authentication/connecting-to-github-with-ssh)

🗄 Step 2: Set Up PostgreSQL Database

You need a production PostgreSQL database. We recommend **Neon** (free tier available).

Option A: Neon (Recommended)

1. Go to Neon (https://neon.tech) and sign up
2. Create a new project
3. Copy your **connection string** (it looks like: `postgresql://user:password@ep-xxx.region.aws.neon.tech/neondb?sslmode=require`)
4. Save this for Step 4 - you'll need it as `DATABASE_URL`

Option B: Supabase

1. Go to Supabase (https://supabase.com) and sign up
2. Create a new project
3. Go to **Settings → Database**
4. Copy the **Connection String** (in "Connection pooling" mode)
5. Save this for Step 4

Option C: Other PostgreSQL Providers

You can also use:
- Railway (https://railway.app)
- Render (https://render.com)
- AWS RDS (https://aws.amazon.com/rds/)
- DigitalOcean (https://www.digitalocean.com/products/managed-databases)

💳 Step 3: Configure Stripe

3.1 Get Your Stripe API Keys

1. Log in to your Stripe Dashboard (https://dashboard.stripe.com)
2. Click **Developers → API keys**
3. You'll see two keys:
   - **Publishable key** (starts with `pk_test_...` in test mode)
   - **Secret key** (starts with `sk_test_...` in test mode - click "Reveal")
4. Save both keys for Step 4

3.2 Set Up Webhook

Stripe needs to send payment confirmations to your app.

1. In Stripe Dashboard, go to **Developers → Webhooks**
2. Click **"Add endpoint"**

3. For the endpoint URL, use: `https://your-app-name.vercel.app/api/stripe/webhook`
   - **Note**: You'll need to come back and update this after deploying to Vercel
   - For now, you can use a placeholder and update it later
4. Click **"Select events"** and choose:
   - `checkout.session.completed`
   - `payment_intent.succeeded`
5. Click **"Add endpoint"**
6. Copy the **Signing secret** (starts with `whsec_...`)
7. Save this for Step 4

✉️ Step 4: Set Up Resend for Emails

1. Go to Resend (https://resend.com) and sign up
2. Go to **API Keys**
3. Click **"Create API Key"**
4. Give it a name (e.g., "Production")
5. Copy the API key (starts with `re_...`)
6. Save this for the next step

☁️ Step 5: Deploy to Vercel

5.1 Import Your GitHub Repository

1. Go to Vercel (https://vercel.com) and log in
2. Click **"Add New"** → **"Project"**
3. If this is your first time:
   - Click **"Import Git Repository"**
   - Authorize Vercel to access your GitHub account
4. Find and select your `mindful-musicpreneur-app` repository
5. Click **"Import"**

5.2 Configure Your Project

Vercel will detect it's a Next.js app automatically.

1. **Framework Preset**: Should auto-detect as "Next.js"
2. **Root Directory**: Leave as `./` (or select `nextjs_space` if needed)
3. **Build Command**: `npm run build` (should be auto-filled)
4. **Output Directory**: `.next` (should be auto-filled)

5.3 Add Environment Variables

Click on **"Environment Variables"** and add the following:

Name	Value	Where to Get It
`DATABASE_URL`	Your PostgreSQL connection string	From Step 2 (Neon/Supabase)
`NEXTAUTH_SECRET`	Random 32-character string	Generate: `openssl rand -base64 32`
`NEXTAUTH_URL`	`https://your-app-name.vercel.app`	Will be provided after first deploy
`NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY`	Your Stripe publishable key	From Step 3 (starts with `pk_test_...`)
`STRIPE_SECRET_KEY`	Your Stripe secret key	From Step 3 (starts with `sk_test_...`)
`STRIPE_WEBHOOK_SECRET`	Your Stripe webhook secret	From Step 3 (starts with `whsec_...`)
`RESEND_API_KEY`	Your Resend API key	From Step 4 (starts with `re_...`)
`ADMIN_EMAIL`	Your admin email	Your email address
`NEXT_PUBLIC_APP_URL`	`https://your-app-name.vercel.app`	Will be provided after first deploy

Important Notes:
- For `NEXTAUTH_SECRET`, run this in your terminal to generate a secure random string:
```bash
  openssl rand -base64 32
```
- For `NEXTAUTH_URL` and `NEXT_PUBLIC_APP_URL`, you'll need to update these after the first deployment with your actual Vercel URL

5.4 Deploy!

1. Click **"Deploy"**
2. Wait for the deployment to complete (usually 2-5 minutes)
3. Once done, you'll see a success screen with your live URL!

🔄 Step 6: Post-Deployment Configuration

After your first successful deployment:

6.1 Update Environment Variables

1. Go to your Vercel project dashboard
2. Click **"Settings"** → **"Environment Variables"**

3. Update the following variables with your actual Vercel URL:
   - `NEXTAUTH_URL` → `https://your-actual-url.vercel.app`
   - `NEXT_PUBLIC_APP_URL` → `https://your-actual-url.vercel.app`
4. Click **"Save"**
5. Go to **"Deployments"** tab
6. Click the three dots on the latest deployment → **"Redeploy"**

6.2 Update Stripe Webhook URL

1. Go back to Stripe Dashboard (https://dashboard.stripe.com/webhooks)
2. Edit your webhook endpoint
3. Update the URL to: `https://your-actual-url.vercel.app/api/stripe/webhook`
4. Save the changes

6.3 Run Database Migrations

Your database needs to have the correct schema. Run migrations:

Option A: From Your Local Machine

```
# Make sure DATABASE_URL in your .env points to your production database
# Then run:
npx prisma migrate deploy
npx prisma db seed
```

Option B: From Vercel Terminal (if available)

1. Go to your Vercel project
2. Click **"Settings"** → **"Functions"**
3. Find the terminal option
4. Run the same commands

✅ Step 7: Test Your Deployment

7.1 Access Your App

1. Visit your Vercel URL: `https://your-app-name.vercel.app`
2. You should see the homepage

7.2 Test Admin Login

1. Go to: `https://your-app-name.vercel.app/admin/login`
2. Use the admin credentials from your seed file:
   - **Email**: `admin@themindfulmusicpreneur.com`
   - **Password**: `Admin123!` (or whatever you set in your seed file)
3. You should be able to access the admin dashboard

7.3 Test Stripe Payment (Test Mode)

1. Browse the products on your site
2. Add a product to cart
3. Go to checkout

4. Use Stripe's test card:
   - **Card Number**: `4242 4242 4242 4242`
   - **Expiry**: Any future date (e.g., `12/25`)
   - **CVC**: Any 3 digits (e.g., `123`)
   - **ZIP**: Any 5 digits (e.g., `12345`)
5. Complete the payment
6. Check if you receive a confirmation email

🔐 Step 8: Security Checklist

Before going live with real customers:

- [ ] Change all default passwords
- [ ] Update admin email to your real email
- [ ] Switch Stripe from test mode to live mode
- [ ] Set up proper domain (optional but recommended)
- [ ] Enable Vercel's password protection (Settings → Deployment Protection)
- [ ] Review and test all payment flows
- [ ] Test email delivery
- [ ] Set up monitoring and error tracking (consider Sentry (https://sentry.io))

🌐 Step 9: Custom Domain (Optional)

To use your own domain instead of `.vercel.app`:

1. Buy a domain from a registrar (e.g., Namecheap (https://namecheap.com), GoDaddy (https://godaddy.com))
2. In Vercel dashboard, go to **"Settings" → "Domains"**
3. Click **"Add"**
4. Enter your domain name
5. Follow Vercel's instructions to update your DNS settings
6. After DNS propagates (can take up to 48 hours), update:
   - `NEXTAUTH_URL` to your custom domain
   - `NEXT_PUBLIC_APP_URL` to your custom domain
   - Stripe webhook URL to your custom domain

🚨 Troubleshooting

Build Fails on Vercel

Error: "Cannot find module…"
- **Solution**: Make sure all dependencies are in `package.json`
- Run `npm install` locally first to verify

Error: "Prisma Client not generated"

- **Solution**: Add a build script in `package.json` :

`json`

```
  "scripts": {
    "vercel-build": "prisma generate && next build"
  }
```

- Update Vercel build command to `npm run vercel-build`

Database Connection Issues

Error: "Can't reach database server"

- **Solution**: Check your `DATABASE_URL` is correct

- Ensure your database allows connections from Vercel's IP addresses

- For Neon/Supabase, make sure SSL mode is enabled

Stripe Webhook Not Working

Error: Payments succeed but orders don't save

- **Solution**:

- Verify webhook URL is correct: `https://your-url.vercel.app/api/stripe/webhook`

- Check webhook secret is correct in environment variables

- Look at Vercel function logs for errors

Email Not Sending

Error: Resend API returns error

- **Solution**:

- Verify your Resend API key is correct

- Check if you've verified your sender domain in Resend

- Check Vercel function logs for detailed errors

Authentication Issues

Error: "Cannot read properties of null"

- **Solution**:

- Verify `NEXTAUTH_SECRET` is set and is a long random string

- Verify `NEXTAUTH_URL` matches your actual deployed URL

- Clear browser cookies and try again

📊 Monitoring Your App

After deployment, monitor your app's health:

1. **Vercel Analytics**:
   - Go to **"Analytics"** tab in your Vercel project
   - View page visits, performance metrics

2. **Vercel Logs**:
   - Go to **"Functions"** tab
   - View real-time logs of API routes and errors

3. **Stripe Dashboard**:
   - Monitor payments in real-time
   - Check webhook delivery status

4. **Database Usage**:
   - Monitor your database usage in Neon/Supabase dashboard
   - Set up alerts for approaching limits

🎉 You're Live!

Congratulations! Your app is now deployed and accessible worldwide.

Next Steps:

1. **Share your URL** with your first users
2. **Monitor the logs** for any issues
3. **Test all features** thoroughly
4. **Set up a staging environment** for testing changes (create a new Vercel project)
5. **Keep your dependencies updated**: Run `npm outdated` regularly

Making Updates:

Vercel automatically redeploys when you push to GitHub:

```
# Make your changes
git add .
git commit -m "Your update message"
git push origin master
```

Vercel will automatically detect the push and redeploy your app!

📚 Additional Resources

- Vercel Documentation (https://vercel.com/docs)
- Next.js Deployment (https://nextjs.org/docs/deployment)
- Prisma with Vercel (https://www.prisma.io/docs/guides/deployment/deployment-guides/deploying-to-vercel)
- Stripe Testing (https://stripe.com/docs/testing)
- NextAuth.js Documentation (https://next-auth.js.org/getting-started/introduction)

💬 Need Help?

If you encounter any issues during deployment:

1. Check the Vercel build logs (in the deployment interface)
2. Review the function logs (Settings → Functions)
3. Consult the troubleshooting section above

4. Reach out to Vercel Support (https://vercel.com/support)

Good luck with your deployment! 🚀