Vercel Deployment Guide for Placid Asia

Critical Files to Change

1. next.config.js

Current (Abacus AI optimized):

```
const path = require('path');

/** @type {import('next').NextConfig} */
const nextConfig = {
  distDir: process.env.NEXT_DIST_DIR || '.next',
  output: process.env.NEXT_OUTPUT_MODE,
  experimental: {
    outputFileTracingRoot: path.join(__dirname, '../'),
  },
  eslint: {
    ignoreDuringBuilds: true,
  },
  typescript: {
    ignoreBuildErrors: false,
  },
  images: { unoptimized: true },
};

module.exports = nextConfig;
```

Replace with (Vercel optimized):

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  eslint: {
    ignoreDuringBuilds: true,
  },
  typescript: {
    ignoreBuildErrors: false,
  },
  images: {
    remotePatterns: [
      {
        protocol: 'https',
        hostname: 'placid.asia',
      },
      {
        protocol: 'https',
        hostname: 'cdn.abacus.ai',
      },
    ],
  },
};

module.exports = nextConfig;
```

Vercel Project Settings

Root Directory

Set to: `nextjs_space`

Build Settings

- **Framework Preset:** Next.js
- **Build Command:** `yarn build`
- **Output Directory:** `.next`
- **Install Command:** `yarn install`

Environment Variables (Required)

Add these in Vercel Dashboard → Settings → Environment Variables:

```
# Database (Use your Vercel Postgres or external DB)
DATABASE_URL="postgresql://user:password@host:5432/database?
schema=public&sslmode=require"

# NextAuth.js
NEXTAUTH_URL="https://your-vercel-app.vercel.app"
NEXTAUTH_SECRET="generate-a-random-32-char-string"

# Brevo Email API
BREVO_API_KEY="your-brevo-api-key"

# Google Analytics (Optional)
NEXT_PUBLIC_GA_MEASUREMENT_ID="G-XXXXXXXXXX"

# Abacus AI Chatbot (Optional)
ABACUSAI_API_KEY="your-abacusai-key"
```

Generate NEXTAUTH_SECRET

```
openssl rand -base64 32
```

Deployment Steps

Via GitHub Desktop

1. **Create a new branch for Vercel:**
   - Branch → New Branch → Name it `vercel-deployment`

2. **Make the change:**
   - Open `nextjs_space/next.config.js` in your editor
   - Replace content with the “Vercel optimized” version above
   - Save the file

3. **Commit changes:**
   - GitHub Desktop will show the change
   - Add commit message: “Configure for Vercel deployment”
   - Click “Commit to vercel-deployment”

4. **Push to GitHub:**
   - Click “Publish branch” or “Push origin”

5. **On Vercel:**
   - Go to vercel.com
   - Click “Add New Project”
   - Import your GitHub repository
   - **IMPORTANT:** Set Root Directory to `nextjs_space`
   - Add all environment variables
   - Deploy!

Database Options for Vercel

Option 1: Vercel Postgres (Recommended for trial)

- Vercel → Storage → Create Database → Postgres
- Copy the `DATABASE_URL` to environment variables
- Run: `npx prisma db push` to create tables

Option 2: Use existing database

- Just use your current `DATABASE_URL`
- Make sure it’s accessible from Vercel’s servers

After Deployment

1. **Seed the database** (if using new database):

   ```
   bash
      vercel env pull .env.local
      npx prisma db push
      npx prisma db seed
   ```

2. **Test the deployment:**
   - Visit your Vercel URL
   - Check admin login: `info@placid.asia`
   - Verify product pages load

Troubleshooting

Build fails with “DATABASE_URL not found”

- Make sure `DATABASE_URL` is added in Vercel environment variables
- Redeploy after adding variables

Images not loading

- Check that image URLs are using `https://`
- Verify `remotePatterns` in `next.config.js`

Prisma errors

- Run `npx prisma generate` in Vercel build
- Check that `prisma/schema.prisma` has no hardcoded paths

Keep Both Deployments

- **Abacus AI (main):** Use `main` branch with original `next.config.js`
- **Vercel (trial):** Use `vercel-deployment` branch with modified config

This way you can test on Vercel without affecting your live site!