Studio 2 - Vercel Deployment Guide

Complete step-by-step guide to deploy your Studio 2 NextJS application to Vercel using GitHub.

📋 Prerequisites

Before you begin, ensure you have:

- [x] GitHub account
- [x] Vercel account (sign up at vercel.com (https://vercel.com))
- [x] Your app code pushed to GitHub repository: `https://github.com/Stefanbotes/Studiov2.git`
- [x] A PostgreSQL database (Vercel Postgres or external provider)

🚀 Quick Start Deployment

Step 1: Push Code to GitHub

If you haven’t already pushed your code to GitHub, follow these steps:

```
# Navigate to your project directory
cd /home/ubuntu/studio_2_app/nextjs_space

# Check git status
git status

# Add all files
git add .

# Commit changes
git commit -m "Prepare for Vercel deployment: Update configs and add deployment docs"

# Push to GitHub (you'll need to authenticate)
git push -u origin master
```

Note: You may need to authenticate with GitHub. Use a Personal Access Token (https://github.com/settings/tokens) as your password.

Step 2: Import Project to Vercel

1. **Go to Vercel Dashboard**
   - Visit vercel.com/dashboard (https://vercel.com/dashboard)
   - Click **“Add New…”** → **“Project”**

2. **Import Git Repository**
   - Click **“Import”** next to your GitHub repository
   - If you don’t see your repository:

   ◦ Click **“Adjust GitHub App Permissions”**

   ◦ Grant Vercel access to your `Studiov2` repository

   ◦ Return to the import page

3. **Configure Project**
   - **Project Name**: `studio2-app` (or your preferred name)
   - **Framework Preset**: Next.js (should be detected automatically)
   - **Root Directory**: `./` (leave as default if your Next.js app is in the repo root)

IMPORTANT: If your Next.js app is in a subdirectory (like `nextjs_space`):
- Set **Root Directory** to `nextjs_space`

- **Build and Output Settings**: (Leave as default)
  - Build Command: `next build` (automatically uses vercel.json config)
  - Output Directory: `.next`
  - Install Command: `npm install`

1. **Skip Environment Variables for Now**
   - Click **“Deploy”** without adding environment variables
   - The first deployment will fail (expected), but we’ll fix this next

---

Step 3: Set Up Database

Option A: Vercel Postgres (Easiest)

1. In your Vercel project dashboard:
   - Click **“Storage”** tab
   - Click **“Create Database”**
   - Select **“Postgres”**
   - Choose a database name: `studio2-db`
   - Click **“Create”**

2. Vercel automatically adds `DATABASE_URL` to your environment variables!

3. Connect to database and run migrations:
   ```bash
   # Install Vercel CLI
   npm i -g vercel

# Login to Vercel
vercel login

# Link your project
vercel link

# Pull environment variables
vercel env pull .env.local

# Run migrations
npx prisma migrate deploy
   ```

```
# (Optional) Seed the database
npm run prisma:seed
```

Option B: External PostgreSQL Provider

Choose one of these providers:

Supabase (Recommended for free tier)
1. Go to supabase.com (https://supabase.com)
2. Create a new project
3. Go to Settings → Database
4. Copy the "URI" connection string
5. Add to Vercel (see Step 4)

Railway (Good for scalability)
1. Go to railway.app (https://railway.app)
2. Create new project → Add PostgreSQL
3. Copy "Postgres Connection URL"
4. Add to Vercel (see Step 4)

Neon (Serverless Postgres)
1. Go to neon.tech (https://neon.tech)
2. Create a new project
3. Copy connection string
4. Add to Vercel (see Step 4)

Step 4: Configure Environment Variables

1. In your Vercel project dashboard:
   - Click **"Settings"** tab
   - Click **"Environment Variables"** in the sidebar

2. Add the following environment variables:

Required Variables:

Variable Name	Value	Description
`DATABASE_URL`	Your PostgreSQL connection string	From Step 3 (if not using Vercel Postgres)
`NEXTAUTH_SECRET`	Generate with `openssl rand -base64 32`	Secret for NextAuth.js
`NEXTAUTH_URL`	`https://your-app.vercel.app`	Your deployment URL (update after first deploy)

Optional Variables (AWS S3):

Variable Name	Example Value	Description
`AWS_PROFILE`	`hosted_storage`	AWS profile name
`AWS_REGION`	`us-west-2`	AWS region
`AWS_BUCKET_NAME`	`your-bucket-name`	S3 bucket name
`AWS_FOLDER_PREFIX`	`uploads/`	S3 folder prefix
`AWS_ACCESS_KEY_ID`	Your AWS access key	AWS credentials
`AWS_SECRET_ACCESS_KEY`	Your AWS secret key	AWS credentials

1. For each variable:
   - Enter **Key** (variable name)
   - Enter **Value**
   - Select environments:
     - ✅ Production
     - ✅ Preview
     - ✅ Development
     - Click **"Save"**

Step 5: Generate NEXTAUTH_SECRET

If you don't have a `NEXTAUTH_SECRET`, generate one:

```
# On Mac/Linux
openssl rand -base64 32

# Or use this online tool
# https://generate-secret.vercel.app/32
```

Copy the generated secret and add it to Vercel environment variables.

Step 6: Deploy

1. **Trigger Deployment**
   - Go to **"Deployments"** tab
   - Click on the failed deployment
   - Click **"Redeploy"**

Or push a new commit to GitHub:
`bash`

```
git commit --allow-empty -m "Trigger Vercel deployment"
git push
```

1. **Monitor Build**
   - Watch the build logs in real-time
   - Check for any errors
   - Build should complete in 2-5 minutes

2. **Update NEXTAUTH_URL**
   - Once deployed, copy your deployment URL (e.g., `https://studio2-app.vercel.app`)
   - Update `NEXTAUTH_URL` environment variable with this URL
   - Redeploy one more time

Step 7: Run Database Migrations

After successful deployment, run Prisma migrations:

```
# Using Vercel CLI
vercel env pull .env.local
npx prisma migrate deploy

# Optional: Seed initial data
npm run prisma:seed
```

Or use the Vercel dashboard:
- Go to your project
- Click **"Settings"** → **"Functions"**
- Enable **"Serverless Function"** for migrations
- Create a migration API endpoint (if needed)

Step 8: Verify Deployment

1. **Visit Your App**
   - Open your deployment URL: `https://your-app.vercel.app`
   - Test login functionality
   - Check database connectivity

2. **Check Logs**
   - Go to **"Deployments"** tab
   - Click on your deployment
   - View **"Function Logs"** for any runtime errors

3. **Test Features**
   - User authentication
   - Client profile management
   - Coaching sessions
   - Data import/export
   - JSON import functionality

🔧 Post-Deployment Configuration

Custom Domain (Optional)

1. Go to **“Settings”** → **“Domains”**
2. Click **“Add”**
3. Enter your custom domain: `studio2.yourdomain.com`
4. Follow DNS configuration instructions
5. Update `NEXTAUTH_URL` to your custom domain
6. Redeploy

Enable Preview Deployments

Preview deployments are automatically created for:
- Pull requests
- Branches pushed to GitHub

Each preview gets a unique URL like:
`https://studio2-app-git-feature-branch.vercel.app`

Set Up Monitoring

1. Enable **Vercel Analytics**:
   - Go to **“Analytics”** tab
   - Click **“Enable Analytics”**

2. Enable **Vercel Speed Insights**:
   - Go to **“Speed Insights”** tab
   - Click **“Enable Speed Insights”**

🎯 Deployment Checklist

Before going live, verify:

- [ ] All environment variables are set correctly
- [ ] Database migrations have been run successfully
- [ ] Authentication works (login/logout)
- [ ] Client profiles can be created and viewed
- [ ] Coaching sessions functionality works
- [ ] JSON import/export works
- [ ] Custom domain configured (if applicable)
- [ ] HTTPS is working (automatic with Vercel)
- [ ] No console errors in browser
- [ ] Mobile responsive design works
- [ ] Performance is acceptable (check Speed Insights)

🔒 Security Considerations

1. **Environment Variables**
   - Never commit `.env` files to git
   - Use different secrets for production vs preview
   - Rotate `NEXTAUTH_SECRET` periodically

2. **Database**
   - Enable SSL connections (default with Vercel Postgres)
   - Restrict database access to Vercel IP ranges
   - Regular backups (automatic with Vercel Postgres)

3. **Authentication**
   - Use strong password requirements
   - Enable rate limiting (consider using middleware)
   - Monitor failed login attempts

4. **API Routes**
   - Validate all inputs
   - Use authentication middleware
   - Rate limit API endpoints

📊 Monitoring and Maintenance

View Logs

```
# Real-time logs
vercel logs --follow

# Recent logs
vercel logs
```

Monitor Performance

- Check **Speed Insights** in Vercel dashboard
- Monitor **Function Duration** and **Invocations**
- Set up alerts for errors

Database Maintenance

- Regular backups
- Monitor connection pool usage
- Optimize slow queries
- Run `VACUUM` periodically

🆘 Troubleshooting

Build Failures

Error: “Module not found”

```
# Solution: Ensure all dependencies are in package.json
npm install <missing-package> --save
git commit -am "Add missing dependency"
git push
```

Error: “Prisma Client not found”

```
# Solution: Add postinstall script in package.json
"scripts": {
  "postinstall": "npx prisma generate"
}
```

Runtime Errors

Error: “Invalid `prisma.client` invocation”
- Check `DATABASE_URL` is set correctly
- Verify database is accessible
- Run migrations: `npx prisma migrate deploy`

Error: “NEXTAUTH_URL is required”
- Add `NEXTAUTH_URL` environment variable
- Must match your actual deployment URL

Database Issues

Connection timeout
- Add `?connect_timeout=15` to DATABASE_URL
- Check database provider status
- Verify connection pool settings

Migration failed
- Check database permissions
- Verify DATABASE_URL format
- Run migrations manually with Vercel CLI

📚 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)
- NextAuth.js Configuration (https://next-auth.js.org/configuration/options)
- Vercel CLI Reference (https://vercel.com/docs/cli)

🎉 Success!

Your Studio 2 app should now be live on Vercel!

Deployment URL: `https://your-app.vercel.app`

Share the link with your users and start coaching! 🚀

📞 Support

If you encounter any issues:

1. Check the Troubleshooting section
2. Review Vercel documentation (https://vercel.com/docs)
3. Check build logs in Vercel dashboard
4. Review environment variables configuration
5. Verify database connectivity

🔄 Updating Your Deployment

To deploy updates:

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

# Vercel automatically deploys on push to master
# Preview deployments are created for other branches
```

Last Updated: October 2025
Version: 2.0