

# New Mexico Reggae Calendar - Deployment Guide

---

## Prerequisites

---

1. **Vercel Account:** Sign up at [vercel.com](https://vercel.com) (<https://vercel.com>)
2. **GitHub Account:** For code repository management
3. **Database:** PostgreSQL database (recommended: Neon, PlanetScale, or Railway)

## Step 1: Database Setup

---

### Option 1: Using Neon (Recommended)

1. Go to [neon.tech](https://neon.tech) (<https://neon.tech>) and create an account
2. Create a new project
3. Copy the database URL (starts with `postgresql://`)

### Option 2: Using PlanetScale

1. Go to [planetscale.com](https://planetscale.com) (<https://planetscale.com>) and create an account
2. Create a new database
3. Get the connection string

### Option 3: Using Railway

1. Go to [railway.app](https://railway.app) (<https://railway.app>) and create an account
2. Create a PostgreSQL database
3. Copy the database URL

## Step 2: Prepare Your Code

---

### 1. Push to GitHub:

```
bash
cd /home/ubuntu/nm-reggae-calendar
git init
git add .
git commit -m "Initial commit - NM Reggae Calendar"
git branch -M main
git remote add origin https://github.com/yourusername/nm-reggae-calendar.git
git push -u origin main
```

### 2. Verify Environment Variables:

- Copy `.env.example` to `.env.local` locally if needed
- Never commit actual `.env` files to git

## Step 3: Deploy to Vercel

---

### Method 1: Vercel Dashboard (Recommended)

1. Go to [vercel.com/dashboard](https://vercel.com/dashboard) (<https://vercel.com/dashboard>)

2. Click "New Project"
3. Import your GitHub repository
4. Configure project:
  - **Framework Preset:** Next.js
  - **Build Command:** `next build` (auto-detected)
  - **Install Command:** `yarn install`
  - **Output Directory:** `.next` (auto-detected)

## Method 2: Vercel CLI

```
npm i -g vercel
cd /home/ubuntu/nm-reggae-calendar/app
vercel --prod
```

## Step 4: Configure Environment Variables

In your Vercel project dashboard:

1. Go to **Settings** → **Environment Variables**
2. Add the following variables:

Variable	Value	Environment
DATABASE_URL	Your PostgreSQL connection string	Production, Preview, Development
ADMIN_PASSWORD	Your chosen admin password	Production, Preview, Development

1. **Important:** Use Vercel's environment variable encryption for sensitive data

## Step 5: Initial Database Setup

After deployment, you need to initialize your database:

1. **Using Vercel CLI:**

```
bash
vercel env pull .env.local
cd /home/ubuntu/nm-reggae-calendar/app
npx prisma generate
npx prisma db push
npx prisma db seed
```

2. **Or via your production environment:**

- Trigger a redeploy to run migrations automatically
- Check deployment logs for any issues

## Step 6: Custom Domain (Optional)

1. In Vercel dashboard, go to **Settings** → **Domains**

2. Add your custom domain: `nmreggaecalendar.com`
3. Configure DNS settings as instructed by Vercel
4. SSL certificates are automatically provisioned

## Step 7: Verify Deployment

---

1. **Check the main calendar:** `https://your-app.vercel.app`
2. **Test the admin interface:** `https://your-app.vercel.app/admin`
3. **Verify API endpoints:**
  - `https://your-app.vercel.app/api/events`
  - `https://your-app.vercel.app/api/admin/backup?password=yourpassword`

## Troubleshooting

---

### Common Issues:

1. **Database Connection Errors:**
  - Verify `DATABASE_URL` is correct
  - Check if database allows external connections
  - Ensure Prisma schema is pushed: `npx prisma db push`
2. **Build Failures:**
  - Check build logs in Vercel dashboard
  - Verify all dependencies are in `package.json`
  - Run `yarn build` locally to test
3. **API Route Errors:**
  - Check function logs in Vercel dashboard
  - Verify environment variables are set
  - Test API routes locally first
4. **Authentication Issues:**
  - Verify `ADMIN_PASSWORD` environment variable
  - Check browser console for errors

### Performance Optimization:

1. **Enable Vercel Analytics:**

```
bash
yarn add @vercel/analytics
```

Add to your `layout.tsx`:

```
typescript
import { Analytics } from '@vercel/analytics/react'
// Add <Analytics /> to your component
```
2. **Configure Edge Runtime (optional):**

Add to API routes that don't use Prisma:

```
typescript
export const runtime = 'edge'
```

## Monitoring & Maintenance

---

### 1. Set up monitoring:

- Enable Vercel Analytics
- Monitor function duration and invocations
- Set up error tracking

### 2. Regular maintenance:

- Monitor database usage
- Update dependencies regularly
- Backup data monthly using admin interface

### 3. Scaling considerations:

- Vercel automatically scales functions
- Database connections are pooled
- Consider upgrading database plan if needed

## Security Notes

---

### 1. Environment Variables:

- Never expose DATABASE\_URL in client code
- Use strong admin passwords
- Rotate passwords periodically

### 2. Database Security:

- Use connection pooling
- Enable SSL connections
- Restrict database access by IP if possible

### 3. Admin Access:

- Consider implementing proper OAuth
- Log admin actions
- Use HTTPS only

## Support & Updates

---

- **Documentation:** This guide and maintenance guide
  - **Issues:** Check Vercel deployment logs
  - **Updates:** Pull latest changes and redeploy
  - **Backup:** Use admin interface to download CSV backups
- 

### Deployment Checklist:

- [ ] Database created and accessible
- [ ] Code pushed to GitHub
- [ ] Vercel project created and connected
- [ ] Environment variables configured
- [ ] Database schema pushed
- [ ] Initial data seeded
- [ ] Main calendar accessible

- ☐ Admin interface working
- ☐ Custom domain configured (if applicable)
- ☐ SSL certificate active
- ☐ Monitoring enabled