

Vercel Deployment Guide for AI Sales-Enablement Platform

Overview

This guide provides step-by-step instructions for deploying the AI Sales-Enablement Platform to Vercel, including environment setup, build configuration, and troubleshooting.

Prerequisites

- GitHub repository: `https://github.com/git-bonda108/ai-sales-platform`
- Vercel account (free tier available)
- Node.js 18+ for local development

Step 1: Connect GitHub Repository to Vercel

1.1 Create Vercel Account

1. Visit vercel.com (`https://vercel.com`)
2. Sign up using your GitHub account for seamless integration
3. Authorize Vercel to access your GitHub repositories

1.2 Import Project

1. Click "New Project" on your Vercel dashboard
2. Select "Import Git Repository"
3. Choose `git-bonda108/ai-sales-platform` from the list
4. Click "Import"

1.3 Configure Project Settings

- **Framework Preset:** Next.js (auto-detected)
- **Root Directory:** `app` (important: the Next.js app is in the `/app` subdirectory)
- **Build Command:** `npm run build` (default)
- **Output Directory:** `.next` (default)
- **Install Command:** `npm install` (default)

Step 2: Environment Variables Setup

2.1 Required Environment Variables

Add these environment variables in Vercel dashboard under Project Settings > Environment Variables:

```
# OpenAI Configuration (when ready to integrate)
OPENAI_API_KEY=your_openai_api_key_here

# Database Configuration (for future CRM integration)
DATABASE_URL=your_database_connection_string

# Authentication (for future user management)
NEXTAUTH_SECRET=your_nextauth_secret_here
NEXTAUTH_URL=https://your-vercel-domain.vercel.app

# API Configuration
API_BASE_URL=https://your-vercel-domain.vercel.app/api

# Feature Flags
ENABLE_CHAT_FEATURE=true
ENABLE_CRM_INTEGRATION=false
ENABLE_ANALYTICS=true
```

2.2 Environment Variable Configuration

1. Go to your project dashboard on Vercel
2. Navigate to Settings > Environment Variables
3. Add each variable with appropriate values for:
 - Production
 - Preview (optional)
 - Development (optional)

Step 3: Build Configuration

3.1 Vercel Configuration File

The project includes a `next.config.mjs` file with optimized settings:

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  experimental: {
    appDir: true,
  },
  images: {
    domains: ['localhost'],
  },
  env: {
    CUSTOM_KEY: process.env.CUSTOM_KEY,
  },
}

export default nextConfig
```

3.2 Build Settings Override (if needed)

If you need to customize build settings:

1. Go to Project Settings > General
2. Modify build settings:
 - **Build Command:** `cd app && npm run build`

- **Output Directory:** `app/.next`
- **Install Command:** `cd app && npm install`

Step 4: Deploy

4.1 Automatic Deployment

- Vercel automatically deploys on every push to the main branch
- Preview deployments are created for pull requests
- Check deployment status in the Vercel dashboard

4.2 Manual Deployment

If needed, trigger manual deployment:

1. Go to your project dashboard
2. Click “Deployments” tab
3. Click “Redeploy” on the latest deployment

Step 5: Custom Domain Setup (Optional)

5.1 Add Custom Domain

1. Go to Project Settings > Domains
2. Add your custom domain
3. Configure DNS records as instructed by Vercel
4. Wait for SSL certificate provisioning (automatic)

5.2 Domain Configuration

- **A Record:** Point to Vercel’s IP (provided in dashboard)
- **CNAME:** Point to `cname.vercel-dns.com`
- **SSL:** Automatically handled by Vercel

Step 6: Performance Optimization

6.1 Recommended Settings

- Enable Edge Functions for API routes
- Configure caching headers for static assets
- Use Vercel Analytics for performance monitoring

6.2 Build Optimization

```
{
  "buildCommand": "npm run build",
  "devCommand": "npm run dev",
  "installCommand": "npm install",
  "framework": "nextjs"
}
```

Troubleshooting

Common Issues and Solutions

1. Build Failures

Problem: Build fails with module not found errors

Solution:

- Ensure all dependencies are in `package.json`
- Check that the root directory is set to `app`
- Verify Node.js version compatibility

2. Environment Variables Not Working

Problem: Environment variables are undefined in production

Solution:

- Double-check variable names (case-sensitive)
- Ensure variables are set for the correct environment
- Restart deployment after adding variables

3. API Routes Not Working

Problem: API endpoints return 404 errors

Solution:

- Verify API routes are in `app/api/` directory
- Check `route.ts` file naming convention
- Ensure proper export of HTTP methods

4. Static Assets Not Loading

Problem: Images or CSS files not loading

Solution:

- Check `next.config.mjs` image domains configuration
- Verify asset paths are relative
- Use Next.js Image component for optimized loading

5. Database Connection Issues

Problem: Cannot connect to database

Solution:

- Verify `DATABASE_URL` format
- Check database provider's IP whitelist
- Ensure connection string includes all required parameters

Debug Commands

```
# Check build logs
vercel logs [deployment-url]

# Local development with Vercel CLI
npx vercel dev

# Pull environment variables locally
npx vercel env pull .env.local
```

Monitoring and Analytics

6.1 Vercel Analytics

- Enable Vercel Analytics in project settings
- Monitor Core Web Vitals
- Track user interactions and performance

6.2 Error Monitoring

- Set up error tracking (Sentry integration available)
- Monitor API endpoint performance
- Set up alerts for deployment failures

Security Considerations

7.1 Environment Variables

- Never commit sensitive keys to repository
- Use Vercel's encrypted environment variables
- Rotate API keys regularly

7.2 CORS Configuration

```
// In API routes
const corsHeaders = {
  'Access-Control-Allow-Origin': process.env.ALLOWED_ORIGINS || '*',
  'Access-Control-Allow-Methods': 'GET, POST, PUT, DELETE, OPTIONS',
  'Access-Control-Allow-Headers': 'Content-Type, Authorization',
}
```

Deployment Checklist

- [] GitHub repository connected
- [] Root directory set to `app`
- [] Environment variables configured
- [] Build settings verified
- [] Custom domain configured (if applicable)
- [] SSL certificate active
- [] Analytics enabled
- [] Error monitoring set up
- [] Performance optimizations applied
- [] Security headers configured

Support and Resources

- [Vercel Documentation](https://vercel.com/docs) (https://vercel.com/docs)
- [Next.js Deployment Guide](https://nextjs.org/docs/deployment) (https://nextjs.org/docs/deployment)
- [Vercel Community](https://github.com/vercel/vercel/discussions) (https://github.com/vercel/vercel/discussions)

Next Steps After Deployment

1. **API Integration:** Add real API endpoints for CRM and AI features
 2. **Authentication:** Implement user authentication system
 3. **Database:** Set up persistent data storage
 4. **Monitoring:** Configure comprehensive logging and monitoring
 5. **Testing:** Set up automated testing pipeline
 6. **CDN:** Optimize asset delivery with Vercel's global CDN
-

Deployment URL: Your app will be available at `https://your-project-name.vercel.app`

Repository: <https://github.com/git-bonda108/ai-sales-platform>