

# Spark Platform - Production Deployment Guide

---

This guide will walk you through deploying the Spark Platform to production using Vercel and Supabase.

## Prerequisites

---

- GitHub account
- Vercel account (free tier available)
- Supabase account (free tier available)
- Domain name (optional, for custom domain setup)

## Step 1: Supabase Database Setup

---

### 1.1 Create a New Supabase Project

1. Go to [supabase.com](https://supabase.com) (<https://supabase.com>) and sign in
2. Click “New Project”
3. Choose your organization
4. Enter project details:
  - **Name:** `spark-platform`
  - **Database Password:** Generate a strong password (save this!)
  - **Region:** Choose closest to your users
5. Click “Create new project”

### 1.2 Configure Database

1. Wait for project creation to complete
2. Go to **Settings** → **Database**
3. Copy the connection string under “Connection string” → “URI”
4. Note down these values for later:
  - **Project URL:** `https://your-project-id.supabase.co`
  - **Anon Key:** Found in Settings → API
  - **Service Role Key:** Found in Settings → API (keep this secret!)

### 1.3 Run Database Migrations

1. In your Supabase project dashboard, go to **SQL Editor**
2. Copy the contents of `supabase/migrations/20240622000001_init_schema.sql`
3. Paste and run the SQL to create your database schema
4. Optionally, run the seed data from `supabase/seed.sql`

## Step 2: Vercel Deployment

---

### 2.1 Connect GitHub Repository

1. Go to [vercel.com](https://vercel.com) (<https://vercel.com>) and sign in
2. Click “New Project”
3. Import your GitHub repository containing the Spark Platform
4. Vercel will automatically detect it's a Next.js project

## 2.2 Configure Environment Variables

Before deploying, add these environment variables in Vercel:

1. In your Vercel project settings, go to **Environment Variables**
2. Add the following variables:

```
# Database
DATABASE_URL=postgresql://postgres:[YOUR-PASSWORD]@db.[YOUR-PROJECT-REF].supabase.co:5432/postgres

# Supabase
NEXT_PUBLIC_SUPABASE_URL=https://[YOUR-PROJECT-ID].supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=[YOUR-ANON-KEY]
SUPABASE_SERVICE_ROLE_KEY=[YOUR-SERVICE-ROLE-KEY]

# Authentication
JWT_SECRET=[GENERATE-32-CHAR-SECRET]
NEXTAUTH_SECRET=[GENERATE-32-CHAR-SECRET]
NEXTAUTH_URL=https://[YOUR-VERCEL-DOMAIN].vercel.app

# API Keys
ABACUSAI_API_KEY=[YOUR-ABACUS-AI-KEY]

# Environment
NODE_ENV=production
NEXT_PUBLIC_APP_URL=https://[YOUR-VERCEL-DOMAIN].vercel.app
```

## 2.3 Deploy

1. Click “Deploy” in Vercel
2. Wait for the build to complete
3. Your app will be available at `https://[project-name].vercel.app`

## Step 3: Custom Domain Setup (spark.ideanest.net)

### 3.1 Configure Domain in Vercel

1. In your Vercel project, go to **Settings** → **Domains**
2. Add your custom domain: `spark.ideanest.net`
3. Vercel will provide DNS configuration instructions

### 3.2 Configure DNS in SiteGround

1. Log into your SiteGround account
2. Go to **Site Tools** → **Domain** → **DNS Zone Editor**
3. Add the DNS records as instructed by Vercel:
  - **Type:** CNAME
  - **Name:** spark
  - **Value:** cname.vercel-dns.com
4. Save the changes

### 3.3 Update Environment Variables

Update these environment variables in Vercel to use your custom domain:

```
NEXTAUTH_URL=https://spark.ideanest.net
NEXT_PUBLIC_APP_URL=https://spark.ideanest.net
```

## Step 4: Post-Deployment Configuration

---

### 4.1 Update Supabase Auth Settings

1. In Supabase, go to **Authentication** → **URL Configuration**
2. Update **Site URL**: `https://spark.ideanest.net`
3. Add **Redirect URLs**: `https://spark.ideanest.net/auth/callback`

### 4.2 Test the Deployment

1. Visit your deployed application
2. Test user registration and login
3. Verify database connectivity
4. Check all major features work correctly

## Step 5: Monitoring and Maintenance

---

### 5.1 Set Up Monitoring

- Enable Vercel Analytics in your project settings
- Monitor Supabase usage in the Supabase dashboard
- Set up error tracking (optional: Sentry integration)

### 5.2 Regular Maintenance

- Monitor database performance and usage
- Keep dependencies updated
- Regular backups (Supabase handles this automatically)
- Monitor SSL certificate renewal (automatic with Vercel)

## Troubleshooting

---

### Common Issues

1. **Build Failures**: Check build logs in Vercel for specific errors
2. **Database Connection Issues**: Verify DATABASE\_URL format and credentials
3. **Authentication Issues**: Ensure NEXTAUTH\_URL matches your domain
4. **API Errors**: Check environment variables are set correctly

### Getting Help

- Check Vercel documentation: [vercel.com/docs](https://vercel.com/docs) (`https://vercel.com/docs`)
- Check Supabase documentation: [supabase.com/docs](https://supabase.com/docs) (`https://supabase.com/docs`)
- Review application logs in Vercel dashboard

## Security Considerations

---

- Never commit `.env` files to version control
- Regularly rotate API keys and secrets

- Monitor access logs for suspicious activity
- Keep dependencies updated for security patches
- Use strong passwords for all accounts

## Performance Optimization

---

- Enable Vercel Edge Functions for better performance
  - Use Supabase Edge Functions for database operations
  - Implement proper caching strategies
  - Monitor Core Web Vitals in Vercel Analytics
  - Optimize images and assets
- 

**Note:** This deployment guide assumes you're using the production-ready configuration files included in this repository. Make sure all configuration files are properly set up before deployment.