






Vercel Deployment Guide - AI-Powered Résumé Builder

This comprehensive guide will walk you through deploying your AI-powered résumé builder from GitHub to Vercel, ensuring a smooth production deployment with all features working correctly.

Prerequisites

Before starting the deployment process, ensure you have:

-  **GitHub Repository:** <https://github.com/thomasperdana/resume-builder.git>
-  **Vercel Account:** Free or Pro account at vercel.com
-  **Supabase Project:** Active Supabase project with database configured
-  **OpenRouter API Key:** Valid API key from OpenRouter
-  **Domain (Optional):** Custom domain if you want to use one

Step 1: Vercel Account Setup

Create or Access Vercel Account

1. Visit vercel.com
2. Sign up for a new account or log into existing account
3. **Recommended:** Sign up using GitHub for seamless integration
4. Verify your email address if required

Install Vercel CLI (Optional)

```
# Install globally for advanced features
npm install -g vercel

# Login to your account
vercel login
```

Step 2: Connect GitHub Repository

Method 1: Vercel Dashboard (Recommended)

1. **Access Dashboard:** Go to vercel.com/dashboard
2. **Import Project:** Click "Add New..." → "Project"
3. **Connect GitHub:**
 - If not connected, click "Continue with GitHub"
 - Authorize Vercel to access your GitHub repositories
4. **Select Repository:**
 - Find `thomasperdana/resume-builder` in the list
 - Click "Import" next to the repository

Method 2: Direct Import URL

1. Go to: `https://vercel.com/new/git/external?repository-url=https://github.com/thomasperdana/resume-builder.git`
2. Click "Continue" to import the repository

Step 3: Configure Project Settings

Framework Detection

Vercel should automatically detect:

- **Framework:** Next.js
- **Build Command:** `pnpm build` or `npm run build`
- **Output Directory:** `.next`
- **Install Command:** `pnpm install` or `npm install`

Manual Configuration (if needed)

If auto-detection fails:

```
# Project Settings
Framework Preset: Next.js
Build Command: pnpm build
Output Directory: .next
Install Command: pnpm install
Development Command: pnpm dev
```

Step 4: Environment Variables Configuration

Required Environment Variables

Add these environment variables in the Vercel dashboard:

Supabase Configuration

```
NEXT_PUBLIC_SUPABASE_URL=https://your-project-id.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
SUPABASE_SERVICE_ROLE_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

AI Service Configuration

```
OPENROUTER_API_KEY=sk-or-v1-your-api-key-here
```

Application Configuration

```
NEXT_PUBLIC_APP_URL=https://your-domain.vercel.app
NEXT_PUBLIC_APP_ENV=production
NEXTAUTH_SECRET=your-nextauth-secret-key
NEXTAUTH_URL=https://your-domain.vercel.app
```

How to Add Environment Variables

Via Vercel Dashboard

1. **Project Settings:** Go to your project dashboard
2. **Environment Variables:** Click "Settings" → "Environment Variables"
3. **Add Variables:** For each variable:
 - **Name:** Enter the variable name (e.g., `NEXT_PUBLIC_SUPABASE_URL`)
 - **Value:** Enter the variable value
 - **Environments:** Select "Production", "Preview", and "Development"
 - **Add:** Click "Add" to save

Via Vercel CLI

```
# Navigate to your project directory
cd resume-builder

# Add environment variables
vercel env add NEXT_PUBLIC_SUPABASE_URL production
vercel env add NEXT_PUBLIC_SUPABASE_ANON_KEY production
vercel env add SUPABASE_SERVICE_ROLE_KEY production
vercel env add OPENROUTER_API_KEY production
vercel env add NEXT_PUBLIC_APP_URL production
vercel env add NEXT_PUBLIC_APP_ENV production
```

Environment Variables Checklist

- ☐ `NEXT_PUBLIC_SUPABASE_URL`
- ☐ `NEXT_PUBLIC_SUPABASE_ANON_KEY`
- ☐ `SUPABASE_SERVICE_ROLE_KEY`
- ☐ `OPENROUTER_API_KEY`
- ☐ `NEXT_PUBLIC_APP_URL`
- ☐ `NEXT_PUBLIC_APP_ENV`
- ☐ `NEXTAUTH_SECRET` (if using NextAuth)
- ☐ `NEXTAUTH_URL` (if using NextAuth)



Step 5: Build Configuration

Vercel Configuration File (Optional)

Create `vercel.json` in your repository root for advanced configuration:

```

{
  "framework": "nextjs",
  "buildCommand": "pnpm build",
  "devCommand": "pnpm dev",
  "installCommand": "pnpm install",
  "functions": {
    "app/api/**/*.ts": {
      "maxDuration": 30
    }
  },
  "headers": [
    {
      "source": "/(.*)",
      "headers": [
        {
          "key": "X-Frame-Options",
          "value": "DENY"
        },
        {
          "key": "X-Content-Type-Options",
          "value": "nosniff"
        },
        {
          "key": "Referrer-Policy",
          "value": "strict-origin-when-cross-origin"
        }
      ]
    }
  ],
  "redirects": [
    {
      "source": "/home",
      "destination": "/",
      "permanent": true
    }
  ]
}

```

```
]
}
```

Next.js Configuration

Ensure your `next.config.js` is production-ready:

```

/** @type {import('next').NextConfig} */
const nextConfig = {
  experimental: {
    appDir: true
  },
  images: {
    domains: [
      'your-supabase-project-id.supabase.co',
      'avatars.githubusercontent.com'
    ]
  },
  async headers() {
    return [
      {
        source: '/(.*)',
        headers: [
          {
            key: 'X-Frame-Options',
            value: 'DENY'
          },
          {
            key: 'X-Content-Type-Options',
            value: 'nosniff'
          },
          {
            key: 'Referrer-Policy',
            value: 'strict-origin-when-cross-origin'
          }
        ]
      }
    ]
  },
  env: {
    NEXT_PUBLIC_APP_URL: process.env.NEXT_PUBLIC_APP_URL
  }
}

```



```
}
```

```
module.exports = nextConfig
```

Step 6: Deploy to Production

Automatic Deployment

1. **Trigger Deployment:** After configuring environment variables, click "Deploy"
2. **Monitor Build:** Watch the build process in real-time
3. **Build Duration:** Typically takes 3-5 minutes
4. **Success Confirmation:** Look for "Build Completed" message

Manual Deployment via CLI



```
# Navigate to project directory
cd resume-builder




# Deploy to production
vercel --prod

# Follow the prompts:
# - Confirm project linking
# - Wait for build completion
# - Get deployment URL
```

Build Process Monitoring

During deployment, monitor for:

-  **Dependencies Installation:** `pnpm install` success
-  **TypeScript Compilation:** No type errors

-  **Next.js Build:** Static generation completion
-  **Environment Variables:** Proper loading
-  **Function Creation:** API routes deployment

Step 7: Domain Configuration

Using Vercel Subdomain

By default, your app will be available at:

- `https://resume-builder-git-master-thomasperdana.vercel.app`
- `https://resume-builder-thomasperdana.vercel.app`

Custom Domain Setup

Add Custom Domain

1. **Domain Settings:** Go to Project → Settings → Domains
2. **Add Domain:** Click "Add" and enter your domain
3. **DNS Configuration:** Follow Vercel's DNS instructions

DNS Configuration Examples

For Apex Domain (example.com):

```
Type: A
Name: @
Value: 76.76.19.61
```

For Subdomain (www.example.com):

Type: CNAME

Name: www

Value: cname.vercel-dns.com

For Custom Subdomain (resume.example.com):

Type: CNAME

Name: resume

Value: cname.vercel-dns.com

SSL Certificate

- **Automatic SSL:** Vercel provides free SSL certificates
- **Certificate Provisioning:** Usually takes 5-10 minutes
- **HTTPS Redirect:** Automatically enabled



Step 8: Continuous Deployment

Automatic Deployments

Vercel automatically deploys when you:

- **Push to Main:** Deploys to production
- **Push to Other Branches:** Creates preview deployments
- **Create Pull Requests:** Generates preview URLs

Branch Configuration

```
# Production Branch: master or main  
# Preview Branches: feature/*, develop, staging
```

Deployment Hooks

Set up webhooks for external triggers:

1. **Webhook URL:** Get from Project → Settings → Git
2. **Trigger Events:** Configure in GitHub repository settings
3. **Deploy Hooks:** Use for content management systems



Step 9: Performance Optimization

Vercel Analytics

1. **Enable Analytics:** Project → Analytics → Enable
2. **Real User Monitoring:** Track Core Web Vitals
3. **Performance Insights:** Monitor page load times

Speed Insights

```
# Install Vercel Speed Insights  
npm install @vercel/speed-insights
```

Add to your app:

```
// app/layout.tsx
import { SpeedInsights } from '@vercel/speed-insights/next'

export default function RootLayout({
  children,
}: {
  children: React.ReactNode
}) {
  return (
    <html>
      <body>
        {children}
        <SpeedInsights />
      </body>
    </html>
  )
}
```

Edge Functions

For optimal performance, Vercel automatically:

- **Edge Caching:** Static assets cached globally
- **API Routes:** Deployed as serverless functions
- **ISR:** Incremental Static Regeneration for dynamic content



Step 10: Monitoring & Debugging

Function Logs

1. **Access Logs:** Project → Functions → View Function Logs
2. **Real-time Monitoring:** Watch live function executions
3. **Error Tracking:** Monitor function errors and timeouts

Build Logs

1. **Deployment History:** Project → Deployments
2. **Build Details:** Click on any deployment for detailed logs
3. **Debug Information:** Check for build warnings and errors

Health Checks

Monitor your deployment:

```
# Check deployment status
curl -I https://your-domain.vercel.app/api/health

# Expected response: 200 OK
```

Troubleshooting Guide

Common Issues and Solutions

Build Failures

Issue: Error: Command "pnpm build" exited with 1

```
# Solution: Check package.json scripts
{
  "scripts": {
    "build": "next build",
    "start": "next start",
    "dev": "next dev"
  }
}
```

Issue: TypeScript compilation errors

```
# Solution: Fix type errors locally first  
npm run type-check  
# Fix all errors, then redeploy
```

Environment Variable Issues

Issue: Error: Missing environment variables

```
# Solution: Verify all required env vars are set  
# Check Settings → Environment Variables  
# Ensure values are correct and saved
```

Issue: Supabase connection failed

```
# Solution: Verify Supabase URLs and keys  
# Test connection: https://app.supabase.com  
# Check project status and API keys
```

Runtime Errors

Issue: 500 Internal Server Error

```
# Solution: Check function logs  
# Project → Functions → View Logs  
# Look for specific error messages
```

Issue: API routes not working

```
# Solution: Verify file structure
# app/api/route-name/route.ts
# Check export syntax: export async function GET()
```

Performance Issues

Issue: Slow page loads

```
# Solution: Enable analytics and optimize
# Check Core Web Vitals
# Optimize images and reduce bundle size
```

Issue: Function timeouts

```
# Solution: Optimize function code
# Reduce processing time
# Consider upgrading to Pro plan for longer timeouts
```

Debug Commands

```
# Local debugging
vercel dev # Test locally with production settings
vercel logs # View recent function logs
vercel env ls # List environment variables

# Production debugging
vercel inspect <deployment-url>
vercel --debug # Verbose deployment output
```




Step 11: Post-Deployment Checklist

Functionality Testing

- ☐ Homepage loads correctly
- ☐ User authentication works
- ☐ Theme switching functions
- ☐ Resume builder preview updates
- ☐ AI tools generate content
- ☐ Database operations successful
- ☐ File uploads/downloads work
- ☐ All API endpoints respond

Performance Verification

- ☐ Page load time < 3 seconds
- ☐ Core Web Vitals passing
- ☐ Images loading optimized
- ☐ Mobile responsiveness
- ☐ Cross-browser compatibility

Security Checks

- ☐ HTTPS enabled and working
- ☐ Environment variables secure
- ☐ API rate limiting active
- ☐ Authentication functioning
- ☐ CORS policies configured

SEO and Analytics

- ☐ Meta tags present
- ☐ Sitemap accessible
- ☐ Analytics tracking
- ☐ Error monitoring setup



Step 12: Ongoing Maintenance

Regular Updates

```
# Update dependencies monthly
npm update

# Security updates
npm audit fix

# Redeploy after updates
git push origin master
```

Monitoring Setup

1. **Uptime Monitoring:** Use Vercel Analytics or external services
2. **Error Tracking:** Implement Sentry or similar service
3. **Performance Monitoring:** Regular Core Web Vitals checks
4. **Security Scanning:** Periodic vulnerability assessments

Backup Strategy

1. **Code Backup:** GitHub repository
2. **Database Backup:** Supabase automatic backups

3. **Environment Config:** Document all environment variables
4. **Deployment Config:** Version control vercel.json

Support and Resources

Vercel Documentation

- **Official Docs:** vercel.com/docs
- **Next.js Guide:** vercel.com/docs/frameworks/nextjs
- **Environment Variables:** vercel.com/docs/concepts/projects/environment-variables

Community Support

- **Vercel Discord:** vercel.com/discord
- **GitHub Discussions:** Project repository discussions
- **Stack Overflow:** Tag questions with `vercel` and `nextjs`

Professional Support

- **Vercel Pro:** Advanced features and priority support
 - **Enterprise:** Dedicated support team
 - **Consulting:** Vercel-certified partners
-

Deployment Complete!








Congratulations! Your AI-powered résumé builder is now live on Vercel. Users can access your platform at your deployed URL and start creating professional résumés with AI assistance.

Next Steps:

1. Share the deployment URL with stakeholders

2. Monitor initial user feedback
3. Set up ongoing monitoring and maintenance
4. Plan for future feature enhancements

Your deployed application includes:

-  Dual theme system with smooth transitions
-  AI-powered résumé building tools
-  Real-time preview and editing
-  ATS optimization features
-  Interview preparation assistance
-  Secure user authentication
-  Production-ready performance

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

GitHub Repository: `https://github.com/thomasperdana/resume-builder.git`

Last Updated: August 2025