

# Deployment Guide - Environment Variables

## Purpose

This guide walks you through deploying the Mindful Champion application with proper environment variable configuration on various platforms.



## Platform-Specific Instructions

### Vercel (Recommended)

Vercel is the recommended platform for Next.js applications.

#### Method 1: Vercel CLI (Fastest)

```
# Install Vercel CLI
npm install -g vercel

# Login to Vercel
vercel login

# Add environment variables
vercel env add DATABASE_URL
vercel env add NEXTAUTH_SECRET
vercel env add NEXTAUTH_URL
vercel env add GMAIL_USER
vercel env add GMAIL_APP_PASSWORD
vercel env add CRON_SECRET
vercel env add ABACUSAI_API_KEY
vercel env add NEXT_PUBLIC_APP_URL

# Add all other variables...
# (You'll be prompted to enter values for each)

# Deploy
vercel --prod
```

#### Tips:

- Use `vercel env pull` to download current environment variables
- Use `vercel env ls` to list all environment variables
- Use `vercel env rm VARIABLE_NAME` to remove a variable

#### Method 2: Vercel Dashboard (Visual)

##### 1. Go to Vercel Dashboard

- Visit: <https://vercel.com/dashboard>
- Select your project

##### 2. Navigate to Environment Variables

- Settings → Environment Variables

### 3. Add Each Variable

- Click “Add New”
- Enter name (e.g., DATABASE\_URL )
- Enter value
- Select environments (Production, Preview, Development)
- Click “Save”

### 4. Redeploy

- Deployments → Latest Deployment → “Redeploy”
- Or push a new commit

## Method 3: Bulk Import (Fastest for many variables)

```
# Create a .env.production file (NOT .env)
# Add all your production values

# Import to Vercel
vercel env pull .env.production --environment production
```

## Vercel-Specific Variables

### Automatic Variables:

- VERCEL : Always 1
- VERCEL\_URL : Deployment URL (auto-generated)
- VERCEL\_ENV : production , preview , or development

### Cron Jobs Configuration:

Create `vercel.json` in project root:

```
{
  "crons": [
    {
      "path": "/api/cron/send-goal-reminders",
      "schedule": "0 9 * * *"
    },
    {
      "path": "/api/cron/cleanup-expired-trials",
      "schedule": "0 0 * * *"
    }
  ]
}
```

**Note:** Vercel automatically includes `CRON_SECRET` in cron job requests.

## Railway

Railway is a simple, modern platform for full-stack applications.

### Setup Steps

#### 1. Install Railway CLI

```
bash
npm install -g @railway/cli
```

## 2. Login to Railway

```
bash
  railway login
```

## 3. Initialize Project

```
bash
  railway init
```

## 4. Add Environment Variables

```
```bash
# Method 1: One by one
railway variables set DATABASE_URL="postgresql://..."
railway variables set NEXTAUTH_SECRET="..."

# Method 2: From file
railway variables set -file .env.production
```

```

## 1. Deploy

```
bash
  railway up
```

## Railway Dashboard Method

### 1. Go to Railway Dashboard

- Visit: <https://railway.app/dashboard>
- Select your project

### 2. Add Variables

- Variables tab
- Click “New Variable”
- Enter name and value
- Click “Add”

### 3. Deploy

- Railway auto-deploys on variable changes

## Netlify

Netlify is great for static sites and serverless functions.

## Setup Steps

### 1. Install Netlify CLI

```
bash
  npm install -g netlify-cli
```

### 2. Login

```
bash
  netlify login
```

### 3. Link Project

```
bash
  netlify link
```

#### 4. Add Environment Variables

```
bash
  netlify env:set DATABASE_URL "postgresql://..."
  netlify env:set NEXTAUTH_SECRET "..."
# ... add all other variables
```

#### 5. Deploy

```
bash
  netlify deploy --prod
```

### Netlify Dashboard Method

#### 1. Go to Netlify Dashboard

- Visit: <https://app.netlify.com/>
- Select your site

#### 2. Add Variables

- Site settings → Build & deploy → Environment
- Click “Edit variables”
- Add each variable
- Click “Save”

#### 3. Trigger Deploy

- Deploys → Trigger deploy

### Docker

For self-hosted or container-based deployments.

#### Option 1: Environment File

Create `.env.production` (NOT committed):

```
DATABASE_URL=postgresql://...
NEXTAUTH_SECRET=...
# ... all other variables
```

#### Dockerfile:

```
FROM node:18-alpine

WORKDIR /app

COPY package*.json .
RUN npm ci --only=production

COPY .

# Build app
RUN npm run build

EXPOSE 3000

CMD [ "npm", "start" ]
```

### Run with env file:

```
docker build -t mindful-champion .
docker run -p 3000:3000 --env-file .env.production mindful-champion
```

### Option 2: Docker Secrets (Recommended for Production)

```
# Create secrets
echo "postgresql://..." | docker secret create db_url -
echo "..." | docker secret create nextauth_secret -

# Use in docker-compose.yml
version: '3.8'
services:
  app:
    image: mindful-champion
    secrets:
      - db_url
      - nextauth_secret
    environment:
      DATABASE_URL: /run/secrets/db_url
      NEXTAUTH_SECRET: /run/secrets/nextauth_secret

secrets:
  db_url:
    external: true
  nextauth_secret:
    external: true
```

## AWS (Elastic Beanstalk, ECS)

### Elastic Beanstalk

```
# Install EB CLI
pip install awsebcli

# Initialize
eb init

# Set environment variables
eb setenv DATABASE_URL="postgresql://..." \
  NEXTAUTH_SECRET="..." \
  NEXTAUTH_URL="https://..." \
  # ... all other variables

# Deploy
eb deploy
```

### ECS (Fargate)

#### 1. Create Task Definition

- Use AWS Systems Manager Parameter Store or Secrets Manager

#### 2. Reference in task definition:

```
json
{
```

```

    "containerDefinitions": [
      {
        "name": "mindful-champion",
        "secrets": [
          {
            "name": "DATABASE_URL",
            "valueFrom": "arn:aws:secretsmanager:region:account:secret:db_url"
          }
        ]
      }
    ]
  }
}

```

---

## Heroku

```

# Install Heroku CLI
brew tap heroku/brew && brew install heroku

# Login
heroku login

# Create app
heroku create mindful-champion

# Add environment variables
heroku config:set DATABASE_URL="postgresql://..."
heroku config:set NEXTAUTH_SECRET="..."
# ... add all other variables

# Deploy
git push heroku main

```

### Dashboard Method:

1. Go to app dashboard
2. Settings → Config Vars
3. Click “Reveal Config Vars”
4. Add each variable

---

## Pre-Deployment Checklist

### Environment Variables

- All CRITICAL variables set
- All IMPORTANT variables set
- All values are production values (not test/dev)
- No placeholder values remaining
- `NEXT_PUBLIC_*` variables are public-safe
- Secrets are strong and random

## Security

- .env file NOT in git
- Different secrets for prod vs dev
- API keys have restrictions (domain, IP)
- Database uses SSL
- S3 bucket has proper IAM policies

## Testing

- Run `npm run verify-env` locally
- Test database connection
- Test email sending
- Test S3 uploads
- Test Stripe webhooks
- Test cron jobs

## Application

- `npm run build` succeeds
- `npm start` works locally
- All features tested
- No console errors
- Monitoring enabled

## Post-Deployment Verification

### 1. Check Application Health

```
# Check if app is responding
curl https://your-domain.com/api/health

# Expected: {"status": "ok"}
```

### 2. Test Cron Endpoints (with CRON\_SECRET)

```
# Test goal reminders
curl -X GET https://your-domain.com/api/cron/send-goal-reminders \
-H "Authorization: Bearer jYEyBMds0rDgrmmAPi8yuUisg2C0zDfqnf/8/1VkJCo="

# Should return success response
```

### 3. Test Email Sending

```
# Trigger a test email through the app
# Or use the test script
node scripts/test-email.js
```

## 4. Check Database Connection

```
# Check database connection
npx prisma studio

# Or check in app logs
# Should see "Database connected successfully"
```

## 5. Monitor Logs

**Vercel:**

```
vercel logs --follow
```

**Railway:**

```
railway logs
```

**Netlify:**

```
netlify logs
```

## 6. Test Key Features

- User registration
- User login
- Goal creation
- Video upload
- Coach Kai chat
- Stripe payment
- Email notifications



## Troubleshooting

### Error: “Environment variable not set”

**Solution:**

1. Verify variable is set in platform
2. Check spelling (exact match)
3. Redeploy after adding
4. Clear platform cache if needed

### Error: “Database connection failed”

**Causes:**

1. Wrong DATABASE\_URL
2. Database not accessible from platform
3. Firewall blocking connection
4. SSL certificate issue

**Solutions:**

1. Verify DATABASE\_URL format
2. Whitelist platform IPs in database
3. Add ?sslmode=require to URL if needed
4. Check database is running

**Error: “NEXTAUTH\_URL mismatch”****Solution:**

```
# Ensure NEXTAUTH_URL matches your actual domain
NEXTAUTH_URL=https://mindful-champion-2hzb4j.abacusai.app

# NOT localhost in production
```

**Error: “Cron job unauthorized”****Solution:**

1. Verify CRON\_SECRET is set
2. Check for typos/spaces
3. Ensure format: Authorization: Bearer <secret>
4. Regenerate if needed

**Error: “Email sending failed”****Causes:**

1. Wrong Gmail credentials
2. App Password not generated correctly
3. 2FA not enabled
4. Gmail daily limit exceeded

**Solutions:**

1. Verify GMAIL\_USER and GMAIL\_APP\_PASSWORD
2. Regenerate App Password
3. Enable 2FA on Gmail
4. Wait 24 hours if limit exceeded



## Updating Environment Variables

### Vercel

```
# Update via CLI
vercel env rm DATABASE_URL
vercel env add DATABASE_URL

# Or update in dashboard
# Settings → Environment Variables → Edit

# Redeploy
vercel --prod
```

## Railway

```
# Update via CLI
railway variables set DATABASE_URL="new-value"

# Or in dashboard (auto-redeploys)
```

## Netlify

```
# Update via CLI
netlify env:set DATABASE_URL "new-value"

# Trigger deploy
netlify deploy --prod
```



## Monitoring & Alerts

### Set Up Monitoring

#### 1. Error Tracking (Sentry)

```
# Install Sentry
npm install @sentry/nextjs

# Add SENTRY_DSN to environment
vercel env add SENTRY_DSN
```

#### 2. Uptime Monitoring

Use services like:

- UptimeRobot: <https://uptimerobot.com/>
- Pingdom: <https://www.pingdom.com/>
- StatusCake: <https://www.statuscake.com/>

#### 3. Log Aggregation

Consider:

- LogRocket: <https://logrocket.com/>
- DataDog: <https://www.datadoghq.com/>
- New Relic: <https://newrelic.com/>



## Quick Deploy Commands

### Vercel

```
vercel --prod
```

## Railway

```
railway up
```

## Netlify

```
netlify deploy --prod
```

## Docker

```
docker build -t mindful-champion .
docker run -p 3000:3000 --env-file .env.production mindful-champion
```

---

**Last Updated:** December 3, 2025

**Status:** Production Ready

**Support:** For deployment issues, check platform-specific documentation