🚀 Biospark Health AI - Production Deployment Guide

🎯 Enterprise-Grade Deployment with 11/10 Rigor

🏗️ Pre-Deployment Checklist

✅ Environment Preparation

- [ ] Vercel account setup and CLI installed
- [ ] GitHub repository access configured
- [ ] Database (PostgreSQL/Supabase) provisioned
- [ ] Sentry project created and DSN obtained
- [ ] All API keys and credentials secured

✅ Configuration Validation

- [ ] `.env.local` configured with all required variables
- [ ] `vercel.json` deployment configuration verified
- [ ] Database schema migrations ready
- [ ] Sentry integration tested locally
- [ ] Build process validated (`npm run build`)

🌍 Environment Variables Setup

Create these environment variables in your Vercel dashboard:

```
# Core Application
NEXT_PUBLIC_APP_ENV=production
NEXTAUTH_SECRET=your_secure_random_string
NEXTAUTH_URL=https://your-domain.vercel.app

# Database Configuration
DATABASE_URL=postgresql://user:password@host:port/database
DIRECT_URL=postgresql://user:password@host:port/database

# Sentry Monitoring (CRITICAL for production)
SENTRY_DSN=https://your-sentry-dsn@sentry.io/project-id
SENTRY_ORG=your-sentry-organization
SENTRY_PROJECT=your-sentry-project
SENTRY_AUTH_TOKEN=your-sentry-auth-token

# External Integrations
OPENAI_API_KEY=sk-your-openai-api-key
ZEP_API_KEY=your-zep-api-key
ZEP_API_URL=https://your-zep-api-url

# Optional: Payment Processing
STRIPE_SECRET_KEY=sk_live_your-stripe-secret
STRIPE_PUBLISHABLE_KEY=pk_live_your-stripe-publishable
```

🚀 Deployment Methods

Method 1: Vercel CLI (Recommended)

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

# Login to Vercel
vercel login

# Deploy to production
vercel --prod

# Set environment variables
vercel env add SENTRY_DSN
vercel env add DATABASE_URL
# ... add all required variables
```

Method 2: GitHub Integration

1. Connect repository to Vercel dashboard
2. Configure environment variables in Vercel settings
3. Enable automatic deployments from main branch
4. Trigger deployment with git push

Method 3: Vercel Dashboard

1. Import project from GitHub
2. Configure build settings:
   - Framework: Next.js
   - Build Command: `npm run build`
   - Output Directory: `.next`
3. Add all environment variables
4. Deploy

🔧 Build Configuration

The project includes optimized build settings:

next.config.js:
- TypeScript error handling for deployment
- Performance optimizations
- External package configurations
- Security headers

vercel.json:
- Function timeout settings
- Regional deployment configuration
- Security headers
- API route rewrites

📊 Post-Deployment Validation

✅ Health Checks

```
# Test application endpoints
curl https://your-domain.vercel.app/api/health
curl https://your-domain.vercel.app/api/auth/session

# Verify database connectivity
curl https://your-domain.vercel.app/api/db/status

# Check Sentry integration
curl https://your-domain.vercel.app/api/sentry/test
```

✅ Performance Validation

- Lighthouse score > 95
- Core Web Vitals within thresholds
- API response times < 200ms
- Database query performance optimized

✅ Security Validation

- HTTPS enforcement active
- Security headers properly configured
- CORS policies implemented
- Rate limiting functional

🔍 Monitoring Setup

Sentry Configuration

1. **Error Tracking:** Automatic error capture and reporting
2. **Performance Monitoring:** Real-time performance metrics
3. **Release Tracking:** Deployment and version monitoring
4. **User Context:** Session and user activity tracking

Vercel Analytics

1. **Web Analytics:** Page views and user interactions
2. **Speed Insights:** Core Web Vitals monitoring
3. **Function Logs:** Serverless function performance
4. **Deployment Logs:** Build and deployment tracking

🚨 Troubleshooting

Common Deployment Issues

Build Failures:

```
# Clear Next.js cache
rm -rf .next
npm run build

# Check TypeScript errors
npx tsc --noEmit

# Validate dependencies
npm audit fix
```

Environment Variable Issues:

```
# Verify variables in Vercel
vercel env ls

# Test locally with production env
vercel env pull .env.local
npm run build
```

Database Connection Issues:

```
# Test database connectivity
npx prisma db push
npx prisma generate

# Run migrations
npx prisma migrate deploy
```

🔄 Continuous Deployment

GitHub Actions (Optional)

```yaml
name: Deploy to Vercel
on:
  push:
    branches: [main]
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Deploy to Vercel
        uses: amondnet/vercel-action@v20
        with:
          vercel-token: ${{ secrets.VERCEL_TOKEN }}
          vercel-org-id: ${{ secrets.ORG_ID }}
          vercel-project-id: ${{ secrets.PROJECT_ID }}
```

📈 Performance Optimization

Production Optimizations

- **Image Optimization:** Next.js automatic image optimization
- **Code Splitting:** Automatic route-based code splitting
- **Static Generation:** ISR for dynamic content
- **CDN Distribution:** Vercel Edge Network

- **Caching Strategy:** Optimized cache headers

Database Optimization

- **Connection Pooling:** Prisma connection management
- **Query Optimization:** Efficient database queries
- **Index Strategy:** Proper database indexing
- **Migration Management:** Automated schema updates

🎯 Success Metrics

Deployment Success Criteria

- ✅ Build completes without errors
- ✅ All environment variables configured
- ✅ Database connectivity established
- ✅ Sentry monitoring active
- ✅ Performance metrics within targets
- ✅ Security headers properly configured
- ✅ HIPAA compliance validated

Production Readiness Checklist

- [ ] SSL certificate active
- [ ] Custom domain configured (if applicable)
- [ ] Monitoring and alerting setup
- [ ] Backup and recovery procedures
- [ ] Documentation updated
- [ ] Team access and permissions configured

🎭 Built with BMAD Method | Enterprise-Grade | Production-Ready

For additional support, refer to:
- Vercel Documentation (https://vercel.com/docs)
- Next.js Deployment Guide (https://nextjs.org/docs/deployment)
- Sentry Integration Guide (./SENTRY_INTEGRATION.md)