

CSrental AI - Deployment Guide

Production Deployment

Prerequisites

- Node.js 18+ installed
- PostgreSQL database (Supabase recommended)
- OpenAI API key
- Domain with SSL certificate
- VPN/Network access control

Environment Setup

1. Clone Repository

```
git clone <repository-url>
cd csrental-improved
npm install
```

1. Environment Variables

```
cp .env.example .env.production
# Edit .env.production with production values
```

1. Database Setup

```
npx prisma generate
npx prisma db push
```

Deployment Options

Option 1: Vercel (Recommended)

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

# Deploy
vercel --prod

# Configure environment variables in Vercel dashboard
```

Option 2: Docker

```
# Dockerfile
FROM node:18-alpine

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

COPY . .
RUN npm run build

EXPOSE 3000
CMD ["npm", "start"]
```

```
# Build and run
docker build -t csrental-ai .
docker run -p 3000:3000 --env-file .env.production csrental-ai
```

Option 3: Traditional Server

```
# Build application
npm run build

# Start with PM2
npm install -g pm2
pm2 start npm --name "csrental-ai" -- start
pm2 save
pm2 startup
```

Security Configuration

1. IP Whitelisting

```
ALLOWED_IPS=office.ip.address,vpn.ip.range,127.0.0.1
```

2. SSL/TLS Setup

- Use Let's Encrypt or commercial SSL certificate
- Configure HTTPS redirect
- Set secure headers in next.config.js

3. Database Security

- Enable Row Level Security (RLS)
- Use connection pooling
- Regular backups

Monitoring Setup

1. Application Monitoring

```
# Install monitoring tools
npm install @sentry/nextjs
```

2. Database Monitoring

- Enable slow query logging
- Set up connection monitoring
- Configure backup alerts

3. Security Monitoring

- Enable audit logging
- Set up intrusion detection
- Configure security alerts

Performance Optimization

1. Caching

```
# Redis for rate limiting and caching
docker run -d --name redis -p 6379:6379 redis:alpine
```

2. CDN Configuration

- Configure Cloudflare or similar CDN
- Enable static asset caching
- Optimize image delivery

3. Database Optimization

- Add appropriate indexes
- Configure connection pooling
- Enable query optimization

Backup Strategy

1. Database Backups

```
# Daily automated backups
pg_dump $DATABASE_URL > backup_$(date +%Y%m%d).sql
```

2. File Storage Backups

- Configure Supabase Storage backups
- Implement versioning
- Test restore procedures

3. Application Backups

- Git repository backups
- Environment configuration backups
- Documentation backups

Health Checks

1. Application Health

```
// pages/api/health.ts
export default function handler(req, res) {
  res.status(200).json({
    status: 'healthy',
    timestamp: new Date().toISOString(),
    version: process.env.npm_package_version
  })
}
```

2. Database Health

```
# Database connectivity check
npx prisma db pull
```

3. External Services

- OpenAI API connectivity
- Supabase service status
- Email service status

Troubleshooting

Common Issues

1. Database Connection Errors

```
# Check connection string
echo $DATABASE_URL
# Test connection
npx prisma db pull
```

1. Authentication Issues

```
# Verify Supabase configuration
echo $NEXT_PUBLIC_SUPABASE_URL
echo $NEXT_PUBLIC_SUPABASE_ANON_KEY
```

1. Rate Limiting Problems

```
# Check Redis connection
redis-cli ping
# Clear rate limit cache
redis-cli FLUSHDB
```

Rollback Procedures

1. Application Rollback

```
# Vercel rollback
vercel rollback

# Manual rollback
git checkout previous-stable-tag
npm run build
pm2 restart csrental-ai
```

2. Database Rollback

```
# Restore from backup
psql $DATABASE_URL < backup_YYYYMMDD.sql
```

3. Configuration Rollback

```
# Restore environment variables
cp .env.production.backup .env.production
pm2 restart csrental-ai
```

Maintenance

Regular Tasks

1. Weekly

- Review security logs
- Check performance metrics
- Update dependencies (security patches)

2. Monthly

- Full security audit
- Performance optimization review
- Backup verification

3. Quarterly

- Disaster recovery testing
- Security penetration testing
- Capacity planning review

Update Procedures

1. Security Updates

```
npm audit
npm audit fix
npm run test
npm run build
```

1. Feature Updates

```
git pull origin main
npm install
npx prisma generate
npm run build
pm2 restart csrental-ai
```

1. Database Updates

```
# Backup first
pg_dump $DATABASE_URL > backup_pre_update.sql
# Apply migrations
npx prisma db push
```

For additional support, contact the CSrental IT team.