FIELD MAINTENANCE TRACKING SYSTEM - INSTALLATION MANUAL

SYSTEM OVERVIEW

The Field Maintenance Tracking System is a React-based Progressive Web App (PWA) designed for field maintenance teams. It uses Supabase as the backend database and can be deployed on Vercel or run locally for development.

TECHNICAL STACK

- Frontend: React 18 + TypeScript + Vite
- Backend: Supabase (PostgreSQL + Auth + Real-time)
- UI Framework: Tailwind CSS + Radix UI
- Deployment: Vercel (recommended) or local development
- PWA: Service Worker + Offline Storage

PREREQUISITES

- Node.js 18+ installed
- npm or yarn package manager
- Git (for version control)
- Supabase account (free tier available)
- Vercel account (free tier available) - for production deployment

INSTALLATION METHODS

METHOD 1: QUICK LOCAL DEVELOPMENT SETUP

Step 1: Clone and Install

---

1. Open terminal/command prompt
2. Navigate to your desired directory
3. Run: git clone [repository-url]
4. cd quadgen
5. Run: npm install

Step 2: Environment Setup

---

1. The .env file is already configured with Supabase credentials
2. No additional environment variables needed for local development

Step 3: Start Development Server

---

1. Run: npm run dev
2. Open browser to: http://localhost:3000
3. Login with: admin@fieldmaintenance.com / admin123

METHOD 2: SUPABASE + VERCEL DEPLOYMENT

Step 1: Supabase Database Setup

---

1. Go to: https://supabase.com/dashboard
2. Create new project or use existing
3. Go to SQL Editor
4. Run the database schema from setup-auth-trigger.sql

5. Create admin user in Authentication > Users

6. Set user role to 'admin' in users table

Step 2: Vercel Deployment

---

1. Install Vercel CLI: npm install -g vercel

2. Login: vercel login

3. Deploy: vercel

4. Set environment variables in Vercel dashboard:

   - VITE_SUPABASE_URL: [your-supabase-url]

   - VITE_SUPABASE_ANON_KEY: [your-supabase-anon-key]

Step 3: Production Configuration

---

1. Update Supabase RLS policies

2. Configure authentication settings

3. Set up backup and monitoring

4. Test all functionality

METHOD 3: LOCAL PRODUCTION BUILD

Step 1: Build the Application

---

1. Run: npm run build

2. This creates a 'dist' folder with production files

3. All files are optimized and ready for deployment

Step 2: Serve Locally

---

1. Install a local server: npm install -g serve

2. Navigate to dist folder: cd dist

3. Start server: serve -s .

4. Access via: http://localhost:3000

Step 3: Configure for Local Use

---

1. Update Supabase settings for local testing

2. Configure environment variables

3. Test all functionality locally

4. Use for development or testing purposes

DETAILED CONFIGURATION

SUPABASE CONFIGURATION

Database Schema Setup:

---

1. Run the complete SQL script from setup-auth-trigger.sql

2. This creates all necessary tables:

   - users (authentication and roles)

   - daily_punch_in (attendance tracking)

   - corrective_maintenance (issue reporting)

   - preventive_maintenance (scheduled maintenance)

   - change_requests (infrastructure changes)

   - gp_live_checks (ground point verification)

   - patroller_tasks (field patrol documentation)

Authentication Setup:

---

1. Enable email authentication in Supabase
2. Configure email templates
3. Set up password policies
4. Configure session management

Row Level Security (RLS):

---

1. Enable RLS on all tables
2. Create policies for user data access
3. Admin users can access all data
4. Regular users can only access their own data

VERCEL DEPLOYMENT CONFIGURATION

Environment Variables:

- VITE_SUPABASE_URL: Your Supabase project URL
- VITE_SUPABASE_ANON_KEY: Your Supabase anonymous key
- VITE_APP_URL: Your deployed app URL (for PWA)

Build Configuration:

- Build command: npm run build
- Output directory: dist
- Install command: npm install
- Node.js version: 18.x

Domain Configuration:

---

1. Add custom domain in Vercel dashboard
2. Configure SSL certificate
3. Set up redirects for PWA
4. Configure CDN settings

PWA CONFIGURATION

Service Worker Setup:

---

1. Service worker is automatically registered
2. Handles offline data storage
3. Manages background sync
4. Caches app resources

Manifest Configuration:

---

1. Update public/manifest.json
2. Configure app name and icons
3. Set display mode and theme
4. Configure install prompts

Offline Storage:

---

1. Uses IndexedDB for local storage
2. Syncs data when online
3. Handles photo storage
4. Manages GPS coordinates

SECURITY CONFIGURATION

Authentication Security:

---

1. Enable email verification
2. Set strong password requirements
3. Configure session timeouts
4. Enable two-factor authentication (optional)

Database Security:

---

1. Enable RLS on all tables
2. Create appropriate policies
3. Regular security audits
4. Backup and recovery procedures

API Security:

---

1. Rate limiting on API endpoints
2. Input validation and sanitization
3. CORS configuration
4. HTTPS enforcement

TESTING AND VALIDATION

Local Testing:

---

1. Test all forms and functionality
2. Verify offline capabilities
3. Test photo capture and GPS
4. Validate data synchronization

Production Testing:

---

1. Load testing with multiple users
2. Mobile device compatibility
3. Network connectivity testing
4. Performance optimization

User Acceptance Testing:

---

1. Admin dashboard functionality
2. User management features
3. All maintenance forms
4. PWA installation and usage

MAINTENANCE AND UPDATES

Regular Maintenance:

---

1. Monitor system performance
2. Update dependencies regularly
3. Backup database weekly
4. Review security logs

System Updates:

---

1. Test updates in development
2. Deploy during maintenance windows
3. Notify users of changes
4. Provide training for new features

Backup Procedures:

---

1. Daily automated backups
2. Weekly manual verification
3. Monthly disaster recovery testing
4. Quarterly security audits

TROUBLESHOOTING

Common Issues:

Database Connection Errors:

- Check Supabase project status
- Verify API keys and URLs
- Test network connectivity
- Review RLS policies

Authentication Issues:

- Verify user exists in Supabase
- Check email verification status
- Review password requirements

- Clear browser cache

PWA Installation Issues:

- Check HTTPS configuration
- Verify manifest.json
- Test service worker registration
- Clear browser data

Performance Issues:

- Monitor database queries
- Check CDN configuration
- Optimize images and assets
- Review caching strategies

SUPPORT AND DOCUMENTATION

Technical Support:

- Check system logs in Supabase
- Review Vercel deployment logs
- Monitor browser console errors
- Test in different browsers

Documentation:

- User manual: COMPANY_INSTRUCTION_MANUAL.md
- Quick setup: QUICK_SETUP_GUIDE.md

- Development: LOCAL_DEVELOPMENT.md
- Deployment: DEPLOYMENT_GUIDE.md

Contact Information:

- IT Administrator: [Your contact info]
- System Status: Check dashboard indicators
- Emergency Support: [Emergency contact]

SYSTEM REQUIREMENTS

Minimum Requirements:

- Node.js 18.0 or higher
- Modern web browser (Chrome, Firefox, Safari, Edge)
- 4GB RAM minimum
- 2GB free disk space
- Internet connection for initial setup

Recommended Requirements:

- Node.js 20.0 or higher
- 8GB RAM or more
- SSD storage
- High-speed internet connection
- Mobile device for field testing

Browser Compatibility:

- Chrome 90+
- Firefox 88+
- Safari 14+
- Edge 90+
- Mobile browsers (iOS Safari, Chrome Mobile)

PERFORMANCE OPTIMIZATION

Database Optimization:

- Index frequently queried columns
- Optimize query performance
- Regular database maintenance
- Monitor query execution times

Frontend Optimization:

- Code splitting and lazy loading
- Image optimization
- Bundle size optimization
- Caching strategies

CDN Configuration:

- Static asset delivery
- Global content distribution
- Cache optimization
- Performance monitoring

MONITORING AND ANALYTICS

System Monitoring:

- Uptime monitoring
- Performance metrics
- Error tracking
- User activity logs

Analytics:

- User engagement metrics
- Form completion rates
- Feature usage statistics
- Performance benchmarks

Alerting:

- System downtime alerts
- Error rate monitoring
- Performance degradation alerts
- Security incident notifications

FINAL CHECKLIST

Pre-Deployment:

□ Database schema created

□ Authentication configured

□ RLS policies enabled

□ Environment variables set

□ Build process tested

□ PWA configuration verified

Post-Deployment:

□ All features tested

□ User accounts created

□ Admin dashboard functional

□ Offline capabilities working

□ Mobile app installation tested

□ Performance monitoring active

User Training:

□ Admin users trained

□ Field workers trained

□ Documentation provided

□ Support contacts shared

□ Backup procedures established

□ Maintenance schedule created

This installation manual provides comprehensive guidance for setting up the Field Maintenance Tracking System. Follow the appropriate method based on your deployment needs and technical requirements.

For additional support or questions, refer to the other documentation files or contact your system administrator.