**SIRTIS - SAYWHAT Integrated Real-Time Information System**

A comprehensive AI-powered enterprise platform designed to facilitate real-time collaboration, data analytics, performance management, and advanced reporting across SAYWHAT's operational departments.

**🚀 Features**

**Core Modules**

* **Home Dashboard** - Real-time operational analytics with predictive insights
* **Programs Management** - Dynamic project management with visual Gantt charts and GIS mapping
* **Call Centre Management** - text transcription with sentiment analysis and AI-driven case prioritization
* **HR Management** - Comprehensive Employee Self-Service portal with AI-driven performance appraisals
* **Inventory & Asset Management** - RFID/barcode tracking with automated depreciation calculations
* **Document Repository** - AI-driven document management with automatic summarization
* **AI Chatbot (SIRTIS Copilot)** - Context-aware assistance across all modules

**Advanced Capabilities**

* **Multi-factor Authentication** - Biometric login support and role-based access control
* **Real-time Synchronization** - Offline-first capabilities with automatic data sync
* **Advanced Analytics** - Predictive insights and machine learning-based risk assessment
* **Security & Compliance** - AES-256 encryption, comprehensive audit trails, WCAG 2.1 accessibility
* **Progressive Web App** - Mobile-first responsive design with dark mode support

**🛠 Technology Stack**

* **Frontend**: Next.js 15, TypeScript, Tailwind CSS, React Query
* **Backend**: Node.js, Express.js, Prisma ORM
* **Database**: PostgreSQL
* **Authentication**: NextAuth.js with MFA support
* **Real-time**: Socket.io
* **AI Integration**: OpenAI API
* **UI Components**: Headless UI, Heroicons
* **File Storage**: AWS S3 / Azure Blob Storage / Local storage

**📋 Prerequisites**

* Node.js 18.x or higher
* PostgreSQL database
* npm or yarn package manager

**🚀 Quick Start**

1. **Clone and Install Dependencies**
2. cd SaywhatSirtis

npm install

1. **Environment Setup**

cp .env.example .env.local

Update the following variables in .env.local:

DATABASE\_URL="postgresql://username:password@localhost:5432/sirtis\_db"

NEXTAUTH\_SECRET="your-secret-key"

OPENAI\_API\_KEY="your-openai-api-key"

1. **Database Setup**
2. # Create and migrate database
3. npx prisma migrate dev --name init
4. # Generate Prisma client
5. npx prisma generate
6. # (Optional) Seed with initial data

npx prisma db seed

1. **Start Development Server**

npm run dev

1. **Access the Application**
   * Open [http://localhost:3000](http://localhost:3000/)
   * Use demo credentials:
     + Email: admin@saywhat.org
     + Password: admin123

**📊 Database Schema**

The system includes comprehensive database models for:

* **User Management**: Users, Roles, Permissions with RBAC
* **Programs**: Projects, Indicators, Progress tracking, Document management
* **Call Centre**: Call records, Case management, Sentiment analysis
* **HR**: Performance plans, Appraisals, Deliverable tracking
* **Inventory**: Asset management, Maintenance scheduling
* **Documents**: Repository with AI-powered categorization
* **Audit**: Comprehensive logging and security tracking

**🔐 Authentication & Security**

* **Multi-factor Authentication** with biometric support
* **Role-Based Access Control** (RBAC) with granular permissions
* **AES-256 encryption** for sensitive data
* **Comprehensive audit logging** for all user actions
* **Session management** with secure token handling

**📱 User Interface**

**Navigation Structure**

Based on user roles and permissions:

* **Home Dashboard** - Overview with real-time analytics
* **Programs** - Project management (M&E personnel access)
* **Call Centre** - Call and case management (restricted access)
* **My HR** - Performance management and employee self-service
* **Inventory Tracking** - Asset management (Finance & Admin access)
* **Document Repository** - Organizational document management
* **About SIRTIS** - System information and help

**AI Chatbot Integration**

* Positioned in the bottom-right corner on all pages
* Context-aware responses based on current module
* Integration with system data for analytical insights
* Support for frequently asked questions

**🔧 Development**

**Available Scripts**

# Development

npm run dev # Start development server with Turbopack

npm run build # Build for production

npm run start # Start production server

npm run lint # Run ESLint

# Database

npx prisma studio # Open Prisma Studio

npx prisma migrate dev # Create and apply migrations

npx prisma generate # Generate Prisma client

**Project Structure**

src/

├── app/ # Next.js App Router pages

│ ├── api/ # API routes

│ ├── auth/ # Authentication pages

│ └── (protected)/ # Protected application pages

├── components/ # React components

│ ├── auth/ # Authentication components

│ ├── dashboard/ # Dashboard components

│ ├── layout/ # Layout components

│ ├── providers/ # Context providers

│ └── ui/ # Reusable UI components

├── lib/ # Utility libraries

│ ├── auth.ts # NextAuth configuration

│ ├── prisma.ts # Prisma client

│ └── utils.ts # Utility functions

└── types/ # TypeScript type definitions

**🚀 Deployment**

**Environment Variables**

Required for production:

DATABASE\_URL= # PostgreSQL connection string

NEXTAUTH\_SECRET= # Random string for JWT signing

NEXTAUTH\_URL= # Your domain URL

EMAIL\_SERVER\_HOST= # SMTP server

EMAIL\_SERVER\_USER= # Email credentials

OPENAI\_API\_KEY= # OpenAI API key for AI features

AWS\_ACCESS\_KEY\_ID= # AWS S3 (if using cloud storage)

AWS\_SECRET\_ACCESS\_KEY= # AWS S3 secret

**Deployment Steps**

1. **Database Setup**
   * Set up PostgreSQL database
   * Run migrations: npx prisma migrate deploy
2. **Build Application**

npm run build

1. **Deploy to Platform**
   * Vercel: Connect GitHub repository
   * AWS/Azure: Use Docker container
   * VPS: Use PM2 for process management

**📝 API Documentation**

**Authentication Endpoints**

* POST /api/auth/signin - User authentication
* POST /api/auth/signout - User logout
* GET /api/auth/session - Get current session

**Core API Routes**

* /api/projects - Project management
* /api/calls - Call centre operations
* /api/hr - Human resources management
* /api/inventory - Asset and inventory management
* /api/documents - Document repository
* /api/analytics - Dashboard analytics

**🔍 Key Requirements Implementation**

**Role-Based Access Control**

* **M&E Personnel**: Full project management access
* **Call Centre Officers**: Call and case management
* **HR Department**: Complete personnel file access
* **Finance & Administration**: Inventory tracking access
* **Management**: Document repository access

**Real-time Features**

* Live dashboard updates
* Real-time notifications
* Concurrent user collaboration
* Instant data synchronization

**AI-Powered Features**

* Chatbot with contextual responses
* Predictive analytics for projects
* Sentiment analysis for calls
* Document summarization and keyword extraction

**🤝 Contributing**

1. Fork the repository
2. Create a feature branch (git checkout -b feature/amazing-feature)
3. Commit changes (git commit -m 'Add amazing feature')
4. Push to branch (git push origin feature/amazing-feature)
5. Open a Pull Request

**📞 Support**

For technical support and questions:

* Email: [saywhatorganization@gmail.com](mailto:saywhatorganization@gmail.com)
* Project Documentation: Available in /docs directory
* Training Materials: Provided during handover process

**📄 License**

This project is proprietary software developed for SAYWHAT organization. All rights reserved.

**SIRTIS** - Empowering SAYWHAT with intelligent, real-time operational management.

HOSTING for Frontend: Vercel Platform , with Postgress from Superbase

Storage: AWS/S3