MoveInSync Assignment Visitor Management System Documentation

Contents

[Overview 1](#_Toc200064898)

[System Architecture 1](#_Toc200064899)

[Frontend 1](#_Toc200064900)

[Backend 1](#_Toc200064901)

[Portal Descriptions 1](#_Toc200064902)

[Admin Portal 1](#_Toc200064903)

[Security Guard Portal 2](#_Toc200064904)

[Employee Portal 2](#_Toc200064905)

[Key Components 2](#_Toc200064906)

[QR Scanner 2](#_Toc200064907)

[Authentication System 3](#_Toc200064908)

[Database Schema 3](#_Toc200064909)

[Security Features 3](#_Toc200064910)

[Authentication 3](#_Toc200064911)

[Data Protection 3](#_Toc200064912)

[Access Control 3](#_Toc200064913)

[Installation and Setup 3](#_Toc200064914)

[Development Guidelines 4](#_Toc200064915)

[Best Practices 4](#_Toc200064916)

[API Documentation 5](#_Toc200064917)

[Authentication Endpoints 5](#_Toc200064918)

[Visitor Management Endpoints 5](#_Toc200064919)

[Check-in/Check-out Endpoints 5](#_Toc200064920)

[Troubleshooting 5](#_Toc200064921)

[Future Enhancements 6](#_Toc200064922)

[Demonstration 6](#_Toc200064923)

# Overview

The MoveInSync Visitor Management System is a comprehensive solution designed to streamline and secure the visitor management process. The system consists of three main portals: Admin Portal, Security Guard Portal, and Employee Portal, each serving specific roles in the visitor management workflow.

# System Architecture

## Frontend

Technology Stack:

* React.js
* TypeScript
* Material-UI (MUI)
* HTML5QrCode for QR scanning
* React Router for navigation
* Axios for API calls

## Backend

Technology Stack:

* Node.js
* Express.js
* MongoDB
* JWT for authentication
* Bcrypt for password hashing

# Portal Descriptions

## Admin Portal

Purpose: System administration and management

Key Features:

* User management (employees and security guards)
* System configuration
* Analytics and reporting
* Access control management

## Security Guard Portal

Purpose: On-ground visitor management

Key Features:

* QR code scanning for visitor check-in/out
* Visitor registration
* Real-time visitor tracking
* Emergency alerts
* Visitor history

## Employee Portal

Purpose: Employee self-service and visitor management

Key Features:

* Visitor invitation
* Visitor approval
* Check-in/out status tracking
* Visitor history

# Key Components

## QR Scanner

Technology: HTML5QrCode

Features:

* Real-time QR code detection
* Camera integration
* Error handling
* Status feedback

Configuration:

* Environment camera support
* 10 FPS scanning rate
* 250x250px scan area
* Aspect ratio maintenance

## Authentication System

* JWT-based authentication
* Role-based access control
* Secure password handling
* Session management

# Database Schema

Collections:

* Users (Employees, Security Guards, Admins)
* Visitors
* Check-ins/Check-outs
* Invitations
* System Logs

# Security Features

## Authentication

* JWT token-based authentication
* Password encryption
* Session management
* Role-based access control

## Data Protection

* Encrypted data transmission
* Secure password storage
* Input validation
* XSS protection

## Access Control

* Role-based permissions
* Portal-specific access
* Feature-level authorization

# Installation and Setup

Prerequisites

* Node.js (v14 or higher)
* MongoDB
* Modern web browser
* Camera access for QR scanning

Installation Steps

1. Clone the repository
2. Install dependencies:  
   npm install
3. Set up environment variables
4. Start the development server:  
   npm start

# Development Guidelines

Code Structure

* Component-based architecture
* Modular design
* TypeScript for type safety
* Consistent coding standards

## Best Practices

* Code Quality
* TypeScript for type safety
* ESLint for code linting
* Prettier for code formatting
* Component documentation
* Performance
* Lazy loading
* Code splitting
* Optimized assets
* Efficient state management
* Testing
* Unit tests
* Integration tests
* End-to-end testing
* Performance testing

# API Documentation

## Authentication Endpoints

* POST /api/auth/login
* POST /api/auth/logout
* POST /api/auth/refresh-token

## Visitor Management Endpoints

* POST /api/visitors/register
* GET /api/visitors/:id
* PUT /api/visitors/:id
* DELETE /api/visitors/:id

## Check-in/Check-out Endpoints

* POST /api/checkin
* POST /api/checkout
* GET /api/visits/history

# Troubleshooting

Common Issues

* QR Scanner Issues
* Camera access denied
* Scanner not detecting codes
* Performance issues
* Authentication Issues
* Token expiration
* Login failures
* Session timeouts
* Performance Issues
* Slow loading times
* Scanner lag
* API response delays

Solutions

* QR Scanner
* Check camera permissions
* Ensure good lighting
* Verify QR code quality
* Clear browser cache
* Authentication
* Clear browser cache
* Check token validity
* Verify credentials
* Reset password if needed

# Future Enhancements

Planned Features

* Mobile app integration
* Advanced analytics
* Automated visitor notifications
* Integration with access control systems

Technical Improvements

* Performance optimization
* Enhanced security measures
* Better error handling
* Improved user experience

# Demonstration

The portal that is present at the security guard at the gate.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

After filling the form we submit it for processing the data

A white and green background

AI-generated content may be incorrect.

After submitting we get an acknowledgement.

A screenshot of a computer

AI-generated content may be incorrect.

The employee the visitor mentioned will get an email like this!

A screenshot of a computer

AI-generated content may be incorrect.

He gets two options! To approve or to reject the visit based on the picture of the visitor!

After approval the employee gets an interface like this

A close-up of a message

AI-generated content may be incorrect.

And the visitor will get a qr code like this through the organization email! In this case it is my University email!

A close-up of a document

AI-generated content may be incorrect.

As I cant use my email twice or thrice and it becomes clumsy if I do, I borrowed my friends university email and registered them as the employee at moveinsync!

This is the scanner that security guards use to verify the qr code!

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

This is the Admin portal to add employees and remove them!

A screenshot of a computer

AI-generated content may be incorrect.

This is the form to add employees.

A screenshot of a phone

AI-generated content may be incorrect.

This is the employee portal in which the employees can pre approve the visitors days before!

A screenshot of a login screen

AI-generated content may be incorrect.

This is the prebook form!

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

After filling details and submitting we get the waiting indication!

A screenshot of a computer

AI-generated content may be incorrect.

After successfully completing the visitor registration we get an interface like this!

A screenshot of a computer

AI-generated content may be incorrect.

The visitor gets a qr code for the slot of visit!

A screenshot of a computer

AI-generated content may be incorrect.

This can be shown at the security to verify the visitor and confirm the validity.