

SafeGuard 플랫폼 — 개발 스펙 정리

프로젝트명: 스마트건설안전플랫폼

레포지토리: <https://github.com/dajimaya/safeguard-platform>

서비스 URL: <https://safeguard-platform-sigma.vercel.app>

작성일: 2026-05-11

프로젝트 개요

건설현장 근로자 출역관리, CCTV 영상관제, 위치관제, 안전이벤트 등 통합 관리 플랫폼

언어 비율 (GitHub 기준)

언어	비율
TypeScript	97.0%
Shell	2.0%
기타 (Python)	1.0%

디렉토리 구조

```
safeguard-platform/  
├── backend/           # Express.js 백엔드 API  
├── frontend/         # Next.js 프론트엔드  
├── nginx/            # Nginx 리버스 프록시 설정  
├── prisma/           # DB 스키마 및 마이그레이션  
├── scripts/          # 운영 자동화 스크립트  
├── streaming/        # FFmpeg CCTV HLS 스트리밍  
├── .env.example      # 환경변수 템플릿  
├── Makefile          # 운영 명령어 단축키  
├── docker-compose.yml # 개발 환경  
├── docker-compose.dev.yml # 개발 환경 (상세)  
└── docker-compose.prod.yml # 운영 환경
```

기술 스택

프론트엔드

항목	스펙	버전
프레임워크	Next.js (App Router)	최신
언어	TypeScript	6.x

항목	스펙	버전
스타일	Tailwind CSS	-
실시간 통신	Socket.IO Client	-
지도	카카오맵 JavaScript API	-
배포 플랫폼	Vercel (Hobby)	-
서비스 URL	https://safeguard-platform-sigma.vercel.app	-

백엔드

항목	스펙	버전
런타임	Node.js	-
프레임워크	Express.js	-
언어	TypeScript	6.x
ORM	Prisma	6.4.1
실시간 통신	Socket.IO Server	-
유효성 검사	Zod	-
인증	JWT (Access Token + Refresh Token)	-
2FA	OTP (Gmail SMTP 발송)	-
비밀번호	bcrypt	6.x
배포 플랫폼	Render	-
서비스 URL	https://safeguard-platform.onrender.com	-

데이터베이스

항목	스펙	버전
DBMS	PostgreSQL	15 (Alpine)
호스팅	Supabase (현재) → 현장 서버 전환 예정	-
ORM	Prisma	6.4.1
Prisma 어댑터	@prisma/adapters-pg	7.8.0
마이그레이션	prisma migrate deploy	-
Seed	ts-node prisma/seed.ts	-

캐시 / 메시지

항목	스펙	버전
엔진	Redis	7 (Alpine)
주요 용도	실시간 위치 캐시 (TTL 10초)	-
	SOS 활성화 상태 (TTL 1시간)	-
	Socket.IO 세션	-
	날씨 API 캐시 (TTL 5분)	-

CCTV 스트리밍

항목	스펙
입력	RTSP (다후아 카메라)
변환 엔진	FFmpeg
출력 포맷	HLS (HTTP Live Streaming)
세그먼트	10초 단위 · 최신 3개 유지 후 삭제
저장 경로	/var/www/hls/{채널ID}/index.m3u8
서빙 포트	8080
재연결	끊김 감지 시 5초 후 자동 재시작
컨테이너	Alpine Linux + FFmpeg + Python3

인프라 (Docker Compose)

컨테이너	이미지	포트	역할
safeguard-postgres	postgres:15-alpine	5432	데이터베이스
safeguard-redis	redis:7-alpine	6379	캐시 / 세션
safeguard-express	자체 빌드 (backend/Dockerfile)	4000	백엔드 API
safeguard-next	자체 빌드 (frontend/Dockerfile)	3000	프론트엔드
safeguard-nginx	nginx:alpine	80 / 443	리버스 프록시
safeguard-streaming	alpine:latest	8080	CCTV HLS 변환

네트워크: [safeguard-net](#) (Bridge)

볼륨: [pg_data](#) (DB 데이터) · [streaming_data](#) (HLS 파일)

현재 배포 환경 (온라인)

서비스	플랫폼	플랜	비고
프론트엔드	Vercel	Hobby	자동 배포 (main 브랜치)
백엔드	Render	무료	15분 미사용 시 슬립
데이터베이스	Supabase	Free	PostgreSQL 15
슬립 방지	UptimeRobot	무료	5분 주기 /health 호출

현장 서버 환경 (전환 예정)

항목	스펙
OS	Ubuntu 22.04 LTS
컨테이너	Docker + Docker Compose
배포 방식	docker-compose up -d
백업	backup.sh (매일 새벽 2시 cron · 30일 보존)
SSL	Let's Encrypt (scripts/ssl-setup.sh)

현장 서버 권장 사양

항목	권장
CPU	4코어 이상
RAM	16GB 이상 (AI 분석 포함 시 32GB)
저장장치	SSD 500GB 이상
네트워크	유선 1Gbps
동시접속	5~10인 (LH 가이드 기준)
CCTV 채널	최대 16채널 (LH 가이드 기준)

사용자 권한 체계

권한	역할	주요 기능
SUPER_ADMIN	시스템 최고 관리자	현장 등록, 전체 관리, OTP 2FA 필수
SITE_MANAGER	현장 소장	현장 설정, 사용자 관리, 출역 강제처리
SAFETY_MANAGER	안전 관리자	안전 이벤트, 문서 관리, 보고서
CONTROLLER	관제사	CCTV 관제, 업무일지 작성, 이벤트 처리
VIEWER	열람자	읽기 전용

외부 연동 서비스

서비스	용도	상태
카카오맵 JavaScript API	현장 지도 · 위치 표출 · SafetyZone 폴리곤	✅ 완료
OpenWeatherMap API	현장 실시간 날씨	✅ 키 발급 완료
Gmail SMTP	SUPER_ADMIN OTP 이메일 발송	✅ 완료
CoolSMS	SOS · 생체이상 · 충돌 SMS 알림	⌚ 발신번호 등록 중
LH 출역 API	출역 데이터 일일 자동 전송	⌚ 실 연동 협의 중
LH 늘봄 A-Eye	CCTV 이벤트 → LH 시스템 연동	⌚ API 스펙 협의 중

DB 주요 테이블

테이블	용도
User	사용자 계정 · 권한
Site	현장 기본 정보
Worker	근로자 정보
Attendance	출역 체크인/아웃
AttendanceSummary	LH 전송용 출역 집계
TBMRecord	TBM 교육 이수 이력
SafetyEducationRecord	안전교육 이력
CctvChannel	CCTV 채널 정보
CctvEvent	AI 감지 이벤트 (헬멧미착용·쓰러짐·화재·연기·구역침입·구조신호)
Machinery	중장비 정보
MachineryLocation	중장비 GPS 위치 이력
MachineryCollisionEvent	협착 감지 이벤트
WorkerLocation	근로자 비콘 위치 이력
BiometricData	생체데이터 (체온·심박수·산소포화도)
SosEvent	SOS 발동 이벤트
SafetyZone	위험구역 폴리곤
SafetyReport	안전 보고서
SmartSafetyPlan	스마트안전관리계획서
SafetyBook	안전보건 대장
ControllerLog	관제사 업무일지
PostReview	사후의견서

테이블	용도
LhApiTransmitLog	LH API 전송 이력
OtpCode	SUPER_ADMIN OTP 인증 코드

API 엔드포인트 전체 목록

인증 (/api/auth)

```

POST /login # 로그인
POST /refresh # 토큰 갱신
POST /logout # 로그아웃
GET /me # 내 정보
POST /super-admin/request-otp # OTP 발송
POST /super-admin/verify-otp # OTP 검증

```

현장 (/api/site)

```

GET / # 현장 정보 조회
PUT / # 현장 정보 수정 (SITE_MANAGER)
GET /weather # 현장 날씨 조회

```

근로자 (/api/workers)

```

GET / # 근로자 목록
POST / # 근로자 등록
GET /stats # 근로자 통계
GET /tbm-pending # TBM 미이수 목록
GET /:id # 근로자 상세
PUT /:id # 근로자 수정
DELETE /:id # 근로자 삭제

```

출역 (/api/attendance)

```

POST /checkin # 체크인
POST /checkout # 체크아웃
GET /today # 오늘 출역 현황
GET /summary # 기간별 요약
GET /lh-log # LH 전송 로그
POST /lh-transmit/retry # LH 재전송

```

CCTV (/api/cctv)

GET	/channels	# 채널 목록
POST	/channels	# 채널 등록
PUT	/channels/:id	# 채널 수정
GET	/events	# 이벤트 목록
PUT	/events/:id/resolve	# 이벤트 해제
POST	/broadcast	# 방송 (네트워크 스피커)

중장비 (/api/machinery)

GET	/	# 중장비 목록
POST	/	# 중장비 등록
PUT	/:id	# 중장비 수정
GET	/:id/locations	# 위치 이력
GET	/collision-events	# 충돌 이벤트 목록
PUT	/collision-events/:id/resolve	# 충돌 이벤트 해제

안전 (/api/safety)

GET	/sos	# SOS 목록
PUT	/sos/:id/acknowledge	# SOS 확인
PUT	/sos/:id/resolve	# SOS 해제
GET	/reports	# 안전 보고서 목록
POST	/reports	# 보고서 등록
PUT	/reports/:id	# 보고서 수정
GET	/zones	# 위험구역 목록

문서 (/api/documents)

GET	/smart-plan	# 안전관리계획서 조회
POST	/smart-plan	# 계획서 작성
PUT	/smart-plan/:id	# 계획서 수정
PUT	/smart-plan/:id/submit	# 계획서 제출
PUT	/smart-plan/:id/approve	# 계획서 승인
GET	/safety-book	# 안전보건 대장
POST	/safety-book	# 대장 등록
PUT	/safety-book/:id	# 대장 수정
DELETE	/safety-book/:id	# 대장 삭제
GET	/controller-log	# 업무일지 목록
POST	/controller-log	# 업무일지 작성
PUT	/controller-log/:id/submit	# 업무일지 제출
PUT	/controller-log/:id/approve	# 업무일지 승인
GET	/post-review	# 사후의견서 목록
POST	/post-review	# 사후의견서 등록

대시보드 (/api/dashboard)

GET /summary # 통합 현황 요약

설정 (/api/settings)

GET /users # 사용자 목록
 POST /users # 사용자 등록
 PUT /users/:id # 사용자 수정
 DELETE /users/:id # 사용자 삭제
 POST /users/:id/reset-password # 비밀번호 초기화
 GET /devices # 장비 목록
 POST /devices # 장비 등록
 PUT /devices/:id # 장비 수정
 DELETE /devices/:id # 장비 삭제
 GET /notifications # 알림 설정 조회
 PUT /notifications # 알림 설정 수정

내부 API (/internal) — x-internal-key 헤더 인증 필수

POST /cctv/events # AI 감지 이벤트 수신
 POST /worker/location # 비콘 위치 수신
 POST /machinery/location # GPS 위치 수신
 POST /machinery/collision # 협착 감지 수신
 POST /biometric # 생체데이터 수신
 POST /sos/trigger # SOS 발동 수신

헬스체크

GET /health # 서버 상태 확인

Socket.IO 실시간 이벤트

서버 → 클라이언트 (emitToSite)

이벤트	트리거	데이터
worker:location:update	비콘 위치 수신	workerId, beaconId, zone
machinery:location:update	GPS 위치 수신	machineryId, lat, lng
machinery:collision	협착 감지	machineryId, nearbyWorkerId, distance

이벤트	트리거	데이터
biometric:alert	생체이상 감지	workerId, name, 체온/심박/산소
sos:triggered	SOS 발동	workerId, name, location
zone:intrusion	위험구역 진입	workerId, workerName, zoneName, zoneType

환경변수 전체 목록

```

# — 데이터베이스 —————
DATABASE_URL=                # PostgreSQL 연결 문자열
DIRECT_URL=                  # Supabase 전용 (현장 서버 전환 시 삭제)

# — JWT 인증 —————
JWT_SECRET=                  # Access Token 서명 키
JWT_REFRESH_SECRET=         # Refresh Token 서명 키

# — 보안 —————
INTERNAL_API_KEY=           # 내부 장비 API 인증 키 (UUID)
NODE_ENV=                    # production / development

# — 이메일 OTP —————
GMAIL_USER=                  # 발송용 Gmail 주소
GMAIL_APP_PASSWORD=         # Gmail 앱 비밀번호 (16자리)

# — SMS —————
SMS_API_KEY=                 # CoolSMS API Key
SMS_API_SECRET=             # CoolSMS API Secret
SMS_SENDER=                  # 발신번호 (등록된 번호)

# — 지도 / 날씨 —————
NEXT_PUBLIC_KAKAO_MAP_KEY=   # 카카오맵 JavaScript 키
WEATHER_API_KEY=            # OpenWeatherMap API 키

# — LH 연동 —————
LH_API_URL=                  # LH 출역 API 엔드포인트

# — CCTV 스트리밍 —————
RTSP_CHANNELS=              # JSON 배열 [{id, url}, ...]
HLS_BASE_URL=               # HLS 서빙 기본 URL

# — 프론트엔드 —————
NEXT_PUBLIC_API_URL=        # 백엔드 API URL

```

운영 스크립트 (scripts/)

파일	용도
backup.sh	DB 일일 백업 (매일 새벽 2시 cron · 30일 보존)
ssl-setup.sh	Let's Encrypt SSL 인증서 발급
ssh-setup.sh	SSH 원격 접속 설정
onboard.sh	현장 서버 초기 설치 자동화

Makefile 주요 명령어

```
make migrate    # DB 마이그레이션  
make backup     # 수동 백업 실행
```

커밋 이력

- 브랜치: main

향후 고도화 계획

- 오픈 데이터셋을 활용한 AICCTV(쓰러짐, 화재, 협착 등 AI 안전감시) 개발(늘봄A-eye 대체)
- 모바일 안전신고 APP 연계