



사용 기술 스택 아키텍처



상태

프로젝트

프론트엔드 기술 스택 (Vue 기반)

1. Vue 3.5.18

- **선택 이유:**
 - **성능 최적화:** Vue 3.5는 리액티브 시스템 리팩토링, 메모리 사용량 최대 56% 감소, 배열 처리 최대 10배 속도 향상 등 성능 개선 포함
 - **안정성 확보:** v3.5는 breaking change 없이 내부 개선만 이루어진 Minor 버전이며, LTS 단계로 유지보수 지원 중
- **기대 효과:** 빠른 초기 렌더, 적은 번들 크기, 안정적인 구조.

2. TypeScript (최신 5.x 버전)

- **선택 이유:**
 - Vue 자체가 TypeScript로 작성되었고, 공식적으로 **TypeScript**를 완벽 지원
 - 정적 타입 검사로 코드 안정성 강화와 IDE 자동 완성 지원으로 협업 효율 극대화.
- **기대 효과:** 코드 품질 향상, 버그 비용 감소, 타입 기반 리팩토링 수월화.

3. 상태관리 – Pinia 2.3.1

- **선택 이유:**
 - Vuex 후속 공식 상태 관리 라이브러리이며 **모듈화 설계 + TS 자동 타입 추론** 지원
 - Mutations 없이 actions 기반 구조로 비전공자도 이해하기 쉬움.
- **기대 효과:** 상태 관리 명확, 유지보수 용이, 번들 크기 작고 디버깅 용이.

4. 빌드 및 PWA 지원 – Vite + vite-plugin-pwa

- **선택 이유:**
 - **Vite**는 Vue 제작자가 만든 빠른 개발 서버로 HMR과 번들 최적화에 강점
 - **vite-plugin-pwa** 연동 시 **오프라인 지원, 설치, 푸시 알림 등 PWA 요소 구현** 쉽게 가능

- **기대 효과:** 웹앱 형태의 설치 가능 UI 제공, 오프라인 기반 주차 안내 메시지 지원 등.
-

5. 라우팅 – Vue Router 4

- **선택 이유:**
 - Vue 3 지원과 Composition API 기반 라우팅으로 구조적 설계 용이
 - 동적 라우트, 중첩 라우트, 라우트 보호 기능 등 다양한 기능 내장.
 - **기대 효과:** 사용자 웹앱, 관리자 대시보드 간 명확하고 효율적인 화면 흐름 구성 가능.
-

백엔드

1. Django

- **선택 이유:** Python 기반의 풀스택 웹 프레임워크
- **기대 효과:** ORM, 어드민, 템플릿 등 내장 기능으로 개발 생산성 극대화

2. Django REST Framework

- **선택 이유:** RESTful API 설계를 위한 확장성 높은 라이브러리
- **기대 효과:** 직관적 직렬화·뷰·권한 관리로 API 개발 가속

3. rest_framework_simplejwt (Simple JWT)

- **선택 이유:** JWT 기반 토큰 인증이 필요해서
- **기대 효과:** 클라이언트 인증·인가를 안전하게 처리

4. dj-rest-auth

- **선택 이유:** 회원가입·로그인·로그아웃 API를 빠르게 구성하고자
- **기대 효과:** 인증 흐름 구현 시간 절감

5. django-allauth

- **선택 이유:** 구글 등 소셜 로그인 지원을 위해
- **기대 효과:** OAuth2 인증·연동을 표준화된 방식으로 제공

6. Django Channels

- **선택 이유:** WebSocket을 통한 실시간 스트리밍/푸시 알림 구현 필요
- **기대 효과:** 비동기 이벤트 기반 통신으로 실시간 반응성 확보

7. MySQL

- 선택 이유: 안정적인 관계형 데이터베이스
- 기대 효과: 대용량 데이터 관리 및 트랜잭션 보장

8. python-decouple

- 선택 이유: 환경 변수(비밀키·DB 설정) 관리를 분리하기 위해
- 기대 효과: 보안 강화 및 설정 이식성 향상

9. django-cors-headers

- 선택 이유: 외부 도메인에서 API 호출 허용 필요
- 기대 효과: CORS 설정을 간단히 관리

10. pywebpush

- 선택 이유: 브라우저 푸시 알림 전송 지원을 위해
- 기대 효과: 사용자에게 실시간 푸시 알림 제공

아두이노

1. ESP8266WiFi

- 선택이유: ESP8266 칩의 내장 WiFi 기능 활용
- 기대효과: 무선 네트워크 연결로 원격 제어 및 데이터 통신 가능

2. WebSocketsClient

- 선택이유: Django 서버와 실시간 양방향 통신 필요
- 기대효과: 번호판 인식 결과를 실시간으로 수신하여 즉시 반응

3. U8g2lib

- 선택이유: OLED 디스플레이에 한글 텍스트 출력 지원
- 기대효과: 번호판 정보를 시각적으로 확인 가능

4. Servo

- 선택이유: 차량 출입 게이트 제어를 위한 서보모터 구동
- 기대효과: 자동화된 게이트 개폐로 무인 출입 시스템 구현

5. Adafruit_VL53L0X

- 선택이유: 정확한 거리 측정으로 차량 진입/이탈 감지

- 기대효과: 차량이 완전히 통과한 후 자동으로 게이트 닫기

6. Wire

- 선택이유: I2C 통신으로 OLED, 거리센서 등 다중 디바이스 연결
 - 기대효과: 단순한 배선으로 여러 센서 통합 관리
-

라즈베리파이(입출입 카메라)

1. websocket-client

- 선택이유: Python에서 WebSocket 연결 및 메시지 송수신을 위해
- 기대효과: 실시간 양방향 통신으로 지연 없이 이미지·텍스트 전송

2. requests.Session

- 선택이유: HTTP 요청 시 커넥션 풀링으로 세션 재사용
- 기대효과: 네트워크 오버헤드 감소 및 안정적 통신

3. ultralytics YOLO

- 선택이유: 번호판 영역을 빠르게 검출 가능한 객체 탐지 모델
- 기대효과: 실시간 프레임당 정확한 번호판 위치 인식

4. EasyOCR

- 선택이유: 한국어 지원 OCR 엔진
- 기대효과: 번호판 문자를 효율적으로 추출

5. Picamera2 + libcamera

- 선택이유: Raspberry Pi 카메라 하드웨어 직관적 제어
- 기대효과: 고/저해상도 캡처를 유연하게 설정·운영

6. OpenCV

- 선택이유: 이미지 색상 변환·크롭·리사이즈 등 전처리
- 기대효과: OCR 전 최적화된 영상 데이터 생성

7. Pillow (PIL)

- 선택이유: 이미지에 검출 박스 및 텍스트 어노테이션
- 기대효과: 결과 영상 위에 시각적 피드백 제공

8. JSON + Base64

- 선택이유: 구조화된 메시지 포맷 및 바이너리 이미지 인코딩
- 기대효과: WebSocket 전송 호환성 및 간편 파싱 보장

9. 자동 재연결 로직

- 선택이유: 네트워크 불안정 시 통신 지속성 확보
- 기대효과: 연결 끊김 시 자동 복구로 무중단 서비스 제공