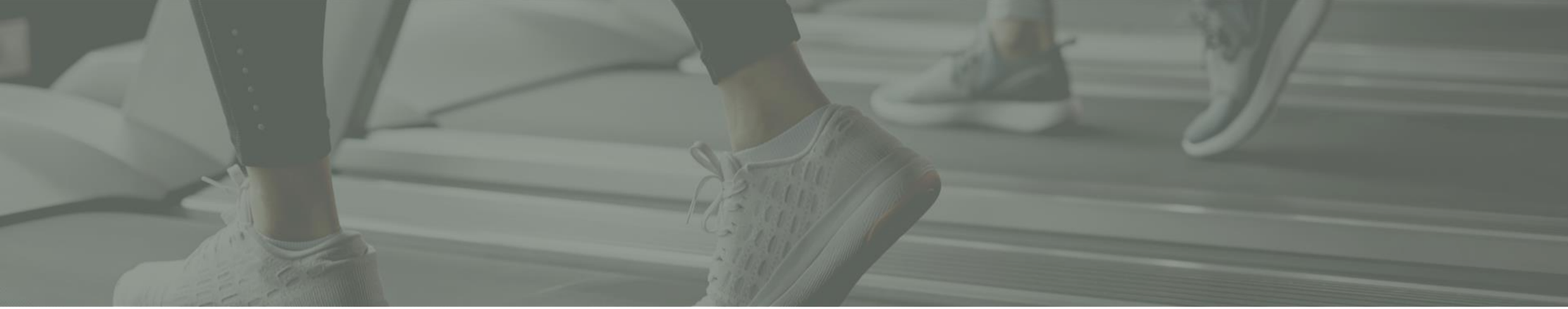
MS AI SCHOOL
3 기 시리얼

TRAIN WITH ME!

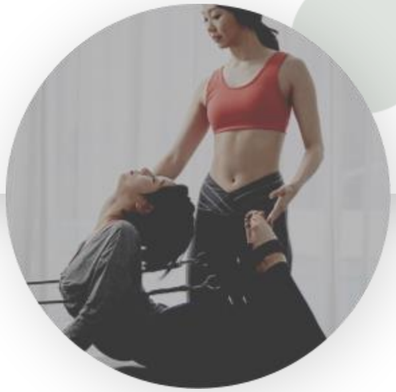
AI FITNESS TRAINER



CONTENTS

AI FITNESS TRAINER 란?

시리얼 조의 AI Fitness Trainer는 매일 아침 먹는 시리얼처럼 운동할 때
매일 사용할 수 있는 운동 자세 교정 서비스 입니다.



msai-cereal/ai_fitness_trainer_v2: Web-Based Exercise Posture Evaluation and AI Voice Feedback System (github.com)

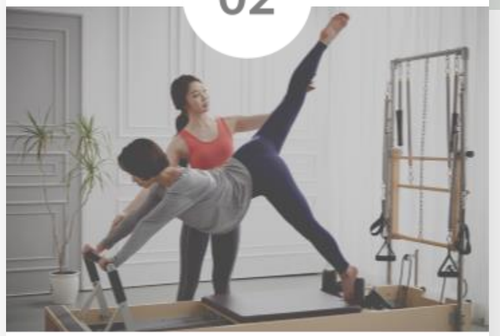
AI FITNESS TRAINER 주요 서비스

01

6가지 운동 분류

푸쉬업, 풀업,
사이드 레터럴 레이즈, 바벨 스쿼트,
버피 테스트, 크로스 런지
총 6가지의 운동이
준비되어 있습니다.

02



실시간 웹캠

실시간 웹캠으로
운동을 바로바로 실시하고
표시된 키포인트로
자세를 확인할 수 있습니다.

03

운동별 피드백 출력

현재 실행 중인 운동 표시와 함께
카운트, 운동 시간, 운동 별 피드백을
실시간 텍스트 및 음성으로
제공합니다.

04

운동 현황 파악

바 차트, 원형 차트, 레이다 도표를
사용하여 각 운동의
현황을 파악할 수 있으며,
모든 운동의 달성률을
비교할 수 있습니다.

시리얼 팀원들

요셉 태하 현상 정빈 아리 은교

프로젝트 진행 일정

[11.15-11.21]

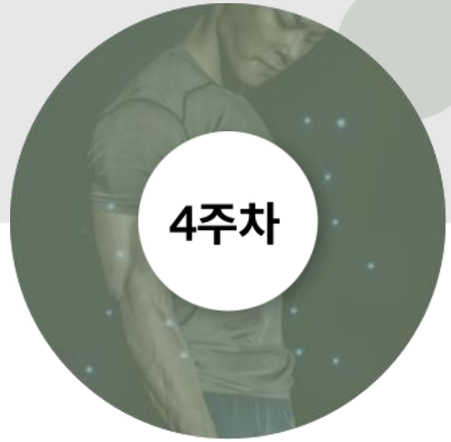
- 객체인식 프로젝트 회고
- AI 프로젝트를 통한 프로젝트 디벨롭 목표 및 방향성 설정

[11.22-11.28]

- YOLO 경량화
- API 개발, 웹소켓 연결
- UI/UX 디자인 및 와이어 프레임

[11.29-12.05]

- 파트별 기능 개발
- 모든 파트 기능 통합



[12.05-12.12]

- 최종 개선 및 마무리

개발 환경

Key Point 검출 학습

웹 애플리케이션 개발

데이터

데이터의 출처

- AI Hub 피트니스 자세 이미지(슬릭코퍼레이션)
- MPII Human Pose Dataset

데이터 전처리 방법

6가지 운동 선택
키포인트 정보 정규화 및 json -> txt 변환
각도에 따른 이미지 분류
crop, padding, resize 시도
누락된 키포인트 annotation 일부 추가
일부 나쁜 데이터 제거 (keypoint 휴먼 에러, 학습 방해하는 포즈)

YOLO 경량화 및 성능 개선 시도

∵ 학습한 m모델은 웹 서비스 배포하기에는 무거움

점수 향상 x

1. 주변 환경 이미지 덜 들어가게 crop+padding+resize -> 큰 차이x
2. 다른 프레임워크 환경으로 변경 -> 큰 차이x
3. flaot16, in8, 양자화 -> 사용 못할 만큼 성능 저하
4. s, n 모델 -> 특정 운동이 사용하기에는 아쉬운 성능
5. 특정 운동만 추가 학습 -> 점수 하락
   -> 데이터 문제 의심(기본적으로 운동 당 1.5만 장 정도 학습 데이터)

YOLO 경량화 및 성능 개선 시도

점수 향상 o

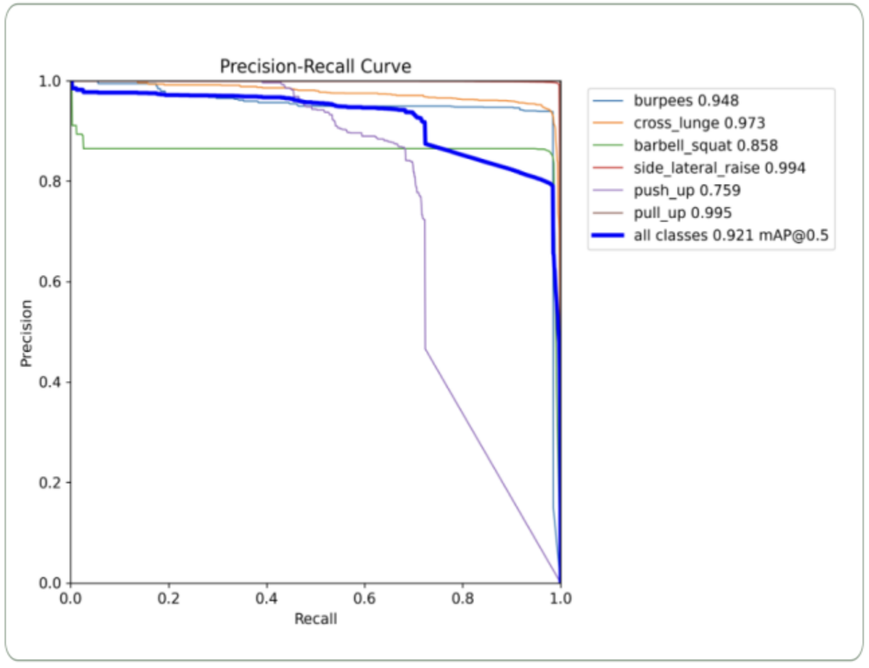
1. MPII 데이터로 사전 학습 -> 실제 성능은 기존과 큰 차이 x
2. n 모델 100epoch(기존 학습은 20epoch로만 진행) -> 실제 성능은 기존과 큰 차이 x
3. 운동 별 모델 학습 -> 특정 운동 제외 괜찮은 성능 -> 단일 모델 대비 사용이 불편함
4. 질 나쁜 데이터 정리 후 학습
   (극단적인 데이터 셋 비율 차이에도 가장 좋은 점수를 가짐)
   -> n: 놓치는 키 포인트로 인해 테스트 결과가 좋았다가 나빴다가 함
   -> s: 사용할 수 있는 성능

모델 평가 지표 및 방법

mAP(Mean Average Precision)
: Object Detection 모델 성능 지표.
AP: Precision-Recall 곡선 아래 면적을 의미

여러 클래스에 대한 AP 계산 -> 평균 값 사용
각 클래스에 대해 정확성과 검출률 고려

mAP50: IOU thresholds가 0.5일 때 AP 값
mAP50-95: IOU thresholds가 0.5에서 0.95사이(step=0.05)일 때 AP 값의 평균



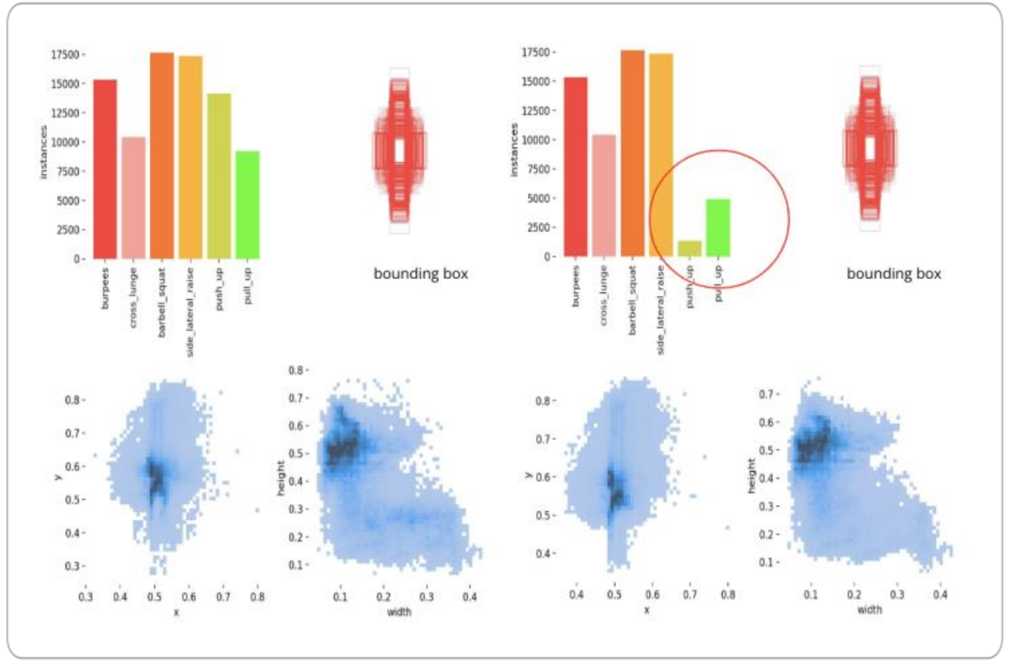
비디오 및 웹캠 테스트
미리 촬영한 운동 영상으로 테스트해서 성능을 확인
성능이 괜찮아 보이면, 영상으로 카운팅 및 자세 교정 함수를 수정하고, 웹캠 테스트에서 추가로 수정함

데이터 비율

데이터 정리 전 데이터 정리 후

class 비율

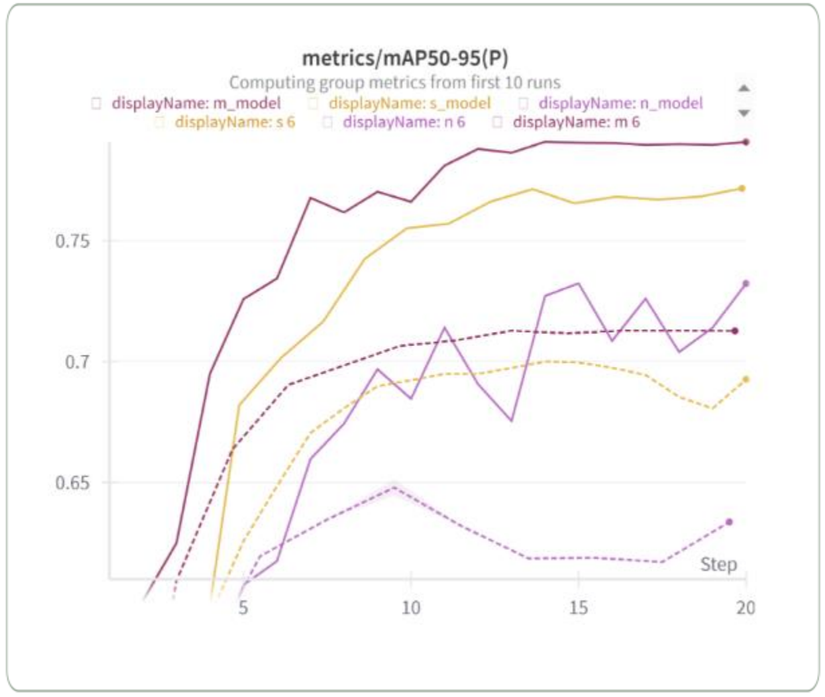
Bbox 정보



모델 성능 비교

x_model(—) : 푸쉬업 풀업 데이터 정제 후

x 6(- -): 데이터 정제 전
모델 성능 : m > s > n
모델 속도 : n > s > m



모델 성능 비교

```python
# 푸쉬업
def count_push_up(pts, flag):
    Shoulder_L, Shoulder_R = pts[5], pts[6]
    Elbow_L, Elbow_R = pts[7], pts[8]
    Wrist_L, Wrist_R = pts[9], pts[10]
    Hip_L, Hip_R = pts[11], pts[12]

    # 어깨-팔꿈치-손목 각도
    Arm_L = cal_angle(Shoulder_L, Elbow_L, Wrist_L)
    Arm_R = cal_angle(Shoulder_R, Elbow_R, Wrist_R)

    # 엎드렸을 때(손이 무릎 밑으로 갔을 때)
    if Elbow_L[1] > Hip_L[1] and Elbow_R[1] > Hip_R[1]:
        if flag and (Arm_L + Arm_R) > 300:
            return 1, False

        # 팔을 구부렸으면, 충분히 구부렸는지 확인
        # 어깨-팔꿈치-손목의 각도 확인
        elif not flag and Arm_L < 120 or Arm_R < 120:
            flag = True
        # 팔꿈치와 어깨의 y좌표 비교
        elif not flag and abs(Shoulder_L[1] - Elbow_L[1]) < 0.012 or abs(Shoulder_R[1] - Elbow_R[1]) < 0.012:
            flag = True

    return 0, flag
```

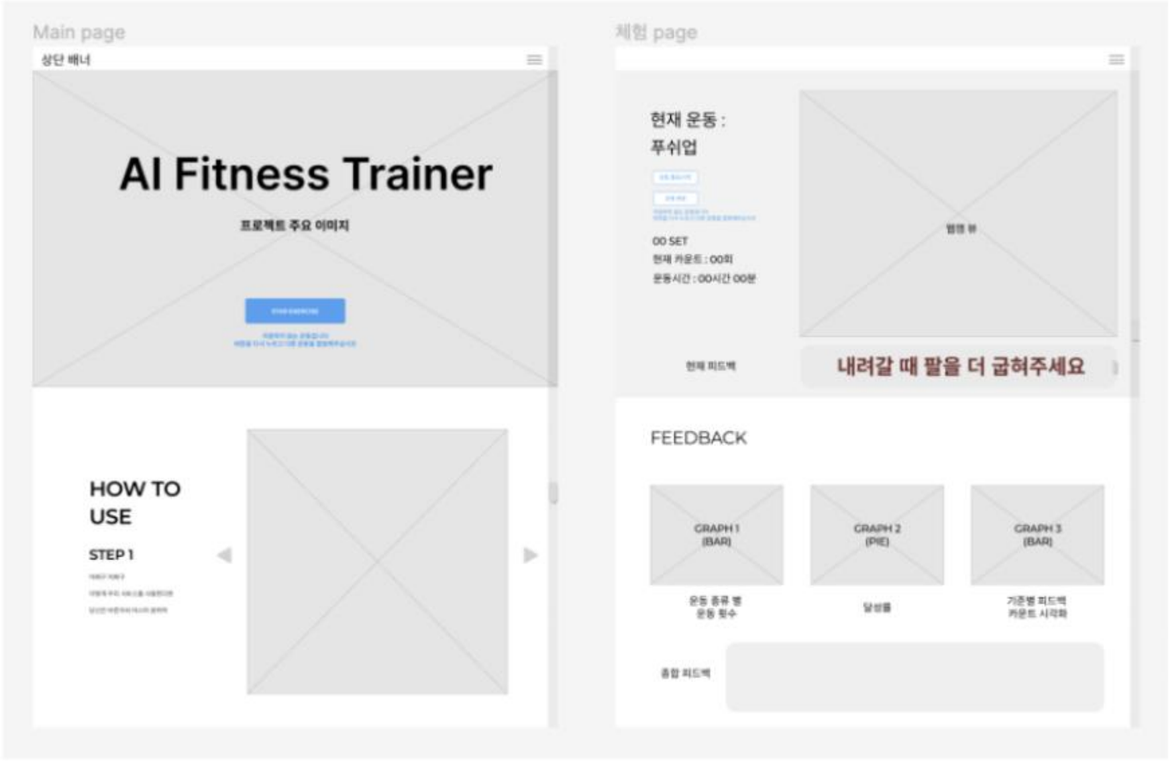
카운팅 함수

키포인트 정보로 길이, 길이비, 각도, 위치 관계 등을 비교해서 운동별 작성

-> 자세평가 함수도 비슷하게 구현

모델 성능 비교

FRONTEND



1. UI/UX 디자인

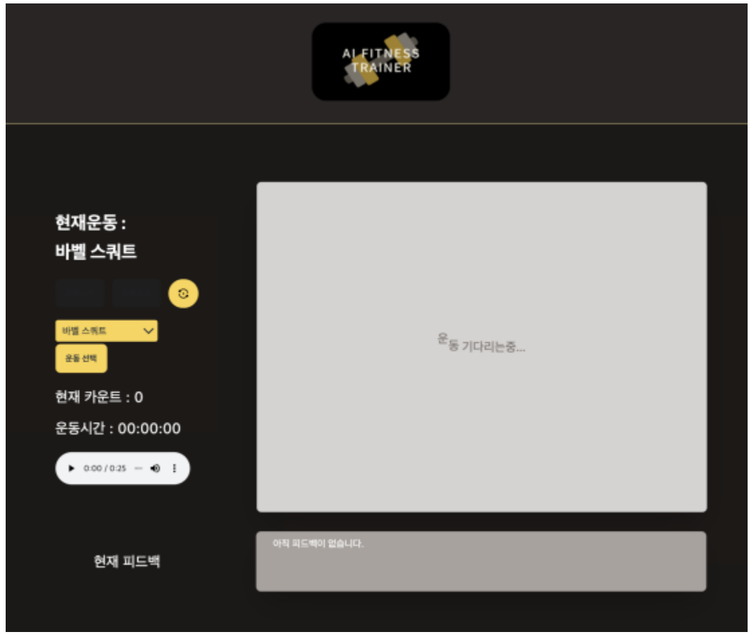
2. 와이어프레임 제작

-> 운동 자세 교정 프로그램에 필요한 기능 정의 및 와이어프레임 작성

FRONTEND

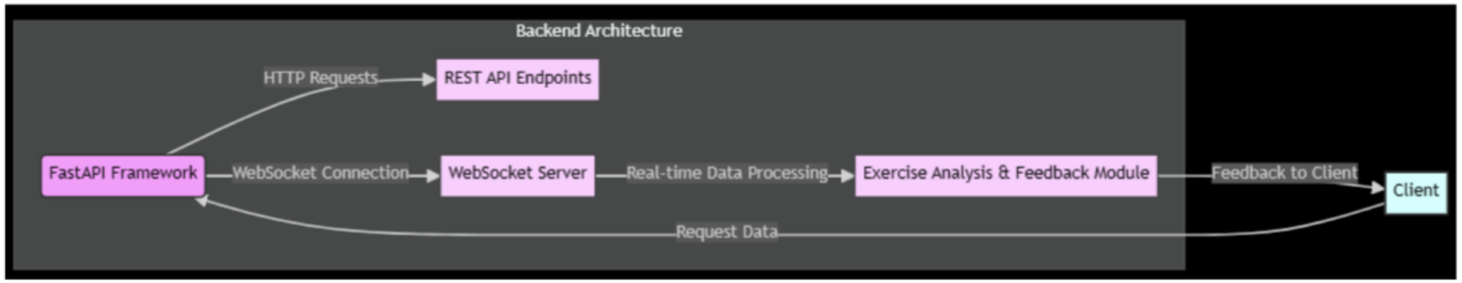
3. React 활용하여 기능 구현

1) 운동 선택 / 변경 버튼
2) 현재 운동 횟수 카운팅
3) 피드백
- 텍스트 피드백
- 차트 피드백
4) Charts.js 활용하여 차트 구현



BACKEND

FastAPI Pydantic

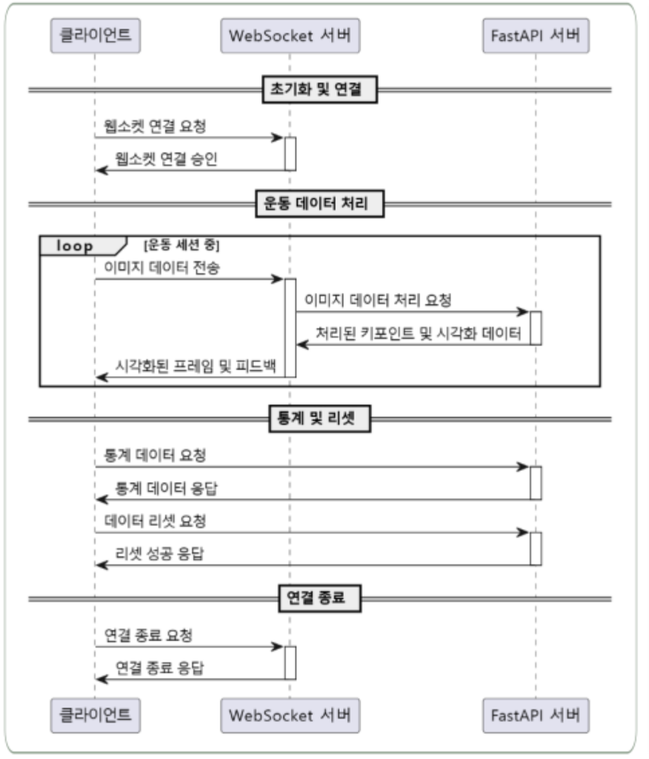


주요 요소 : REST API
WebSocket Server
Fine-tuning한 YOLOv8 모듈
운동 분석 및 피드백 모듈

BACKEND

실시간 데이터의 처리와 최적화

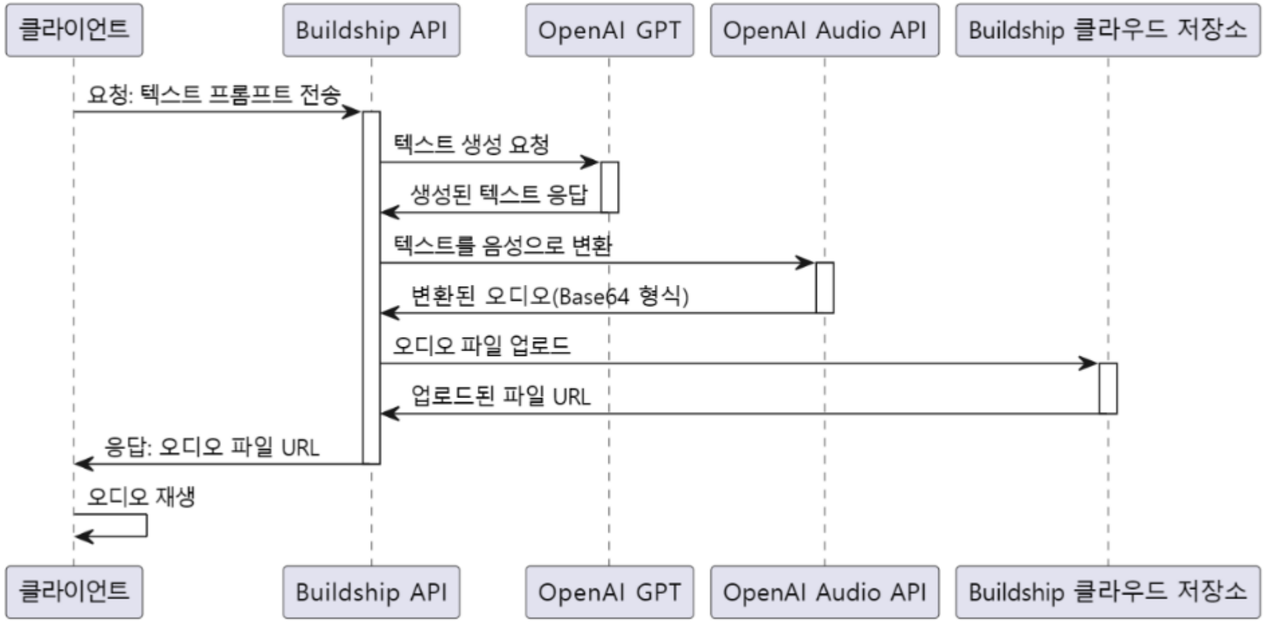
- 실시간 이미지 데이터 처리의 복잡성
  예) 데이터 인코딩/디코딩, 프레임 분석
- 성능 최적화를 위한 전략
  예)비동기 처리, FPS 속도 제한, 모델 경량화, etc.



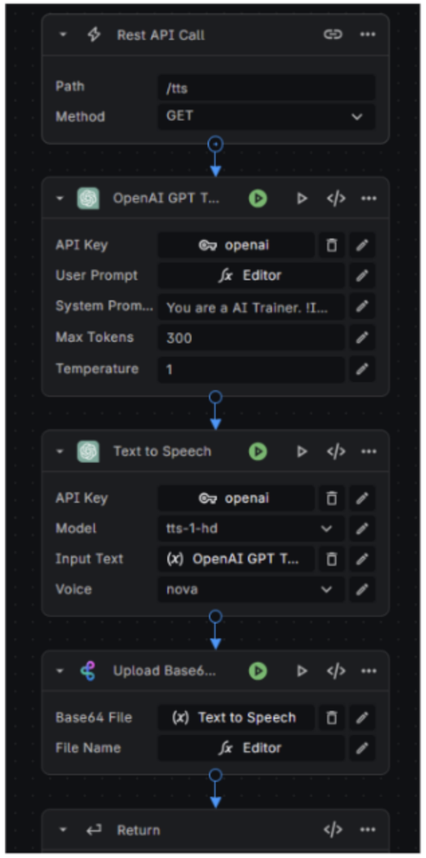
(클라이언트-서버 상호작용 및 데이터 플로우)

BACKEND

low-code backend builder 및 third-party API 활용: 개발 기간 단축



(운동 분석 및 피드백 시스템 플로우)



BACKEND

대략 음성 AI 피드백을 받기까지 평균 10초 정도 소요

Text Generation ~4.8s | Audio Generation ~4.6s | Uploading file ~0.1s

프로젝트 시연

깃허브 주소: msai-cereal/ai_fitness_trainer_v2: Web-Based Exercise Posture Evaluation and AI Voice Feedback System (github.com)

도전 과제

1. 기술 복잡성과 러닝 커브
- 복합 기술 활용의 어려움 (FastAPI, Pydantic, WebSocket, Buildship 등)
-. 기술적 이해와 적용에 대한 높은 러닝 커브

2. 실시간 데이터 처리의 도전
- 웹캠 데이터의 실시간 처리와 성능 문제
- Base64 인코딩/디코딩, 버퍼 관리의 복잡성

3. 포즈 인식 모델의 적용
- YOLO 모델 기반 포즈 인식의 정확도 및 성능 최적화 과제
- 실시간 시각화 구현의 기술적 도전

4. 운동 분석 로직의 복잡성
- 다양한 운동 유형에 대한 정밀한 분석과 계산

5. 전역 데이터 관리
- 실시간 세션 및 전역 데이터의 효율적 관리
- 데이터 동기화와 일관성 유지의 중요성

6. 사용자 경험(UX)과 인터페이스 설계
- 실시간 피드백 제공을 위한 사용자 인터페이스의 중요성
- 사용자 친화적인 상호작용 및 데이터 표시 방식

본 프로젝트는 기술적, 설계적, 운영적 측면에서 귀중한 학습 기회 제공
향후 비슷한 프로젝트 진행 시 참고할 수 있는 경험과 인사이트 얻음


곰곰...
QNA

MS AI SCHOOL
3 기 시리얼

THANK YOU!