

TREND

5TH PROJECT

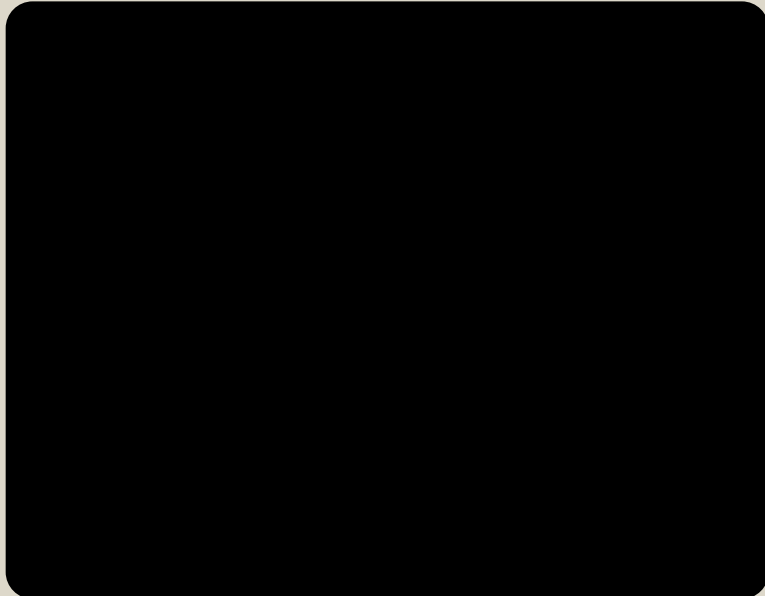
LENS

OUR TEAM

5TH PROJECT

trendlens

CHUNG GUYUN



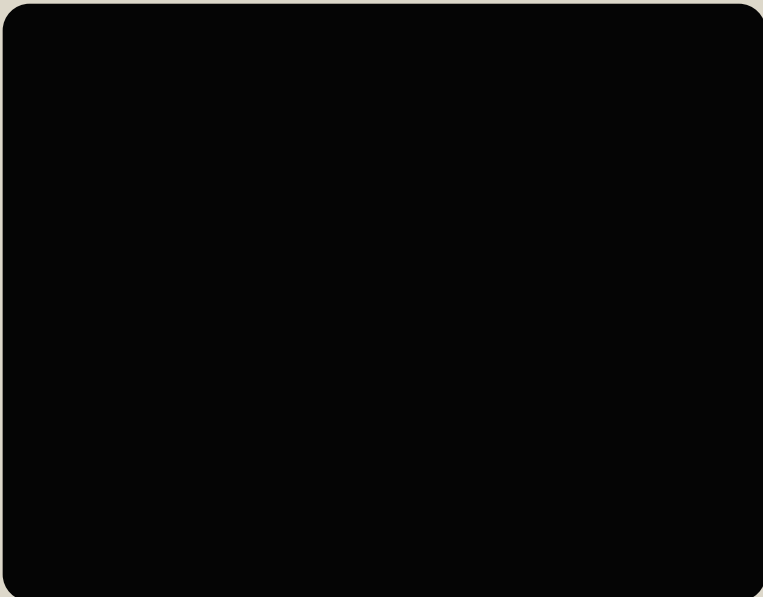
DEVOPS & KUBERNETES
& CI/CD 구현

PARK BOYOUNG



CRAWLING + NAVER API + AI분석

PARK JINHYUK



외부 결제 시스템(TOSS PAYMENTS)
연동을 이용한 구독 기반 서비스

HONG SUNMIN



SPRING SECURITY
JWT 발급 및 재발급
로직 구현

JEON JAEMAN



SAM3D-BODY 를 이용한 신체측정
GEMINI API 이용하여 체형에 맞는
최신 패션 트렌드 추천

AGENDA

1

프로젝트개요

2

기술 스택

3

프로젝트를
시작하며..

4

프로젝트
DATA BASE 구조

5

담당 도메인

6

아키텍처

7

트러블 슈팅

8

프로젝트를
마치며..

TrendLens

TRENDLENS는

다양한 패션 E커머스의 상품·룩 데이터를 수집해

‘지금 뜨는 트렌드’를 빠르게 따라갈 수 있게 도와주는 서비스입니다.

멤버십 가입 회원은 이미지 기반 체형 분석을 통해

개인 체형 특성에 맞춘 트렌드 추천을 받아,

유행을 무작정 따라가는 게 아니라

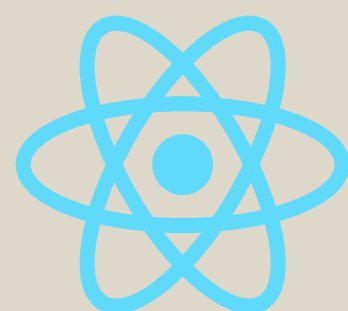
‘나에게 어울리는 유행’을 선택할 수 있습니다.

DEVELOPMENT ENVIRONMENT & TECHNICAL SKILLS

5TH PROJECT

trendlens

FRONT



REACT



TYPE SCRIPT



TAILWIND CSS



SHADCN UI

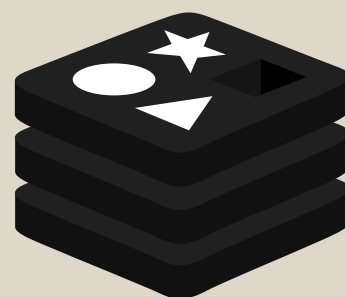
BACK & DB



SPRING
BOOT

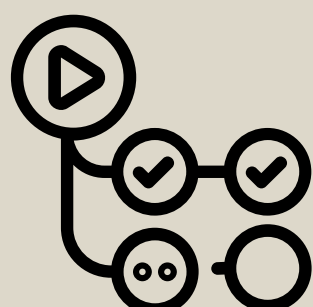
ORACLE

ORACLE



REDIS

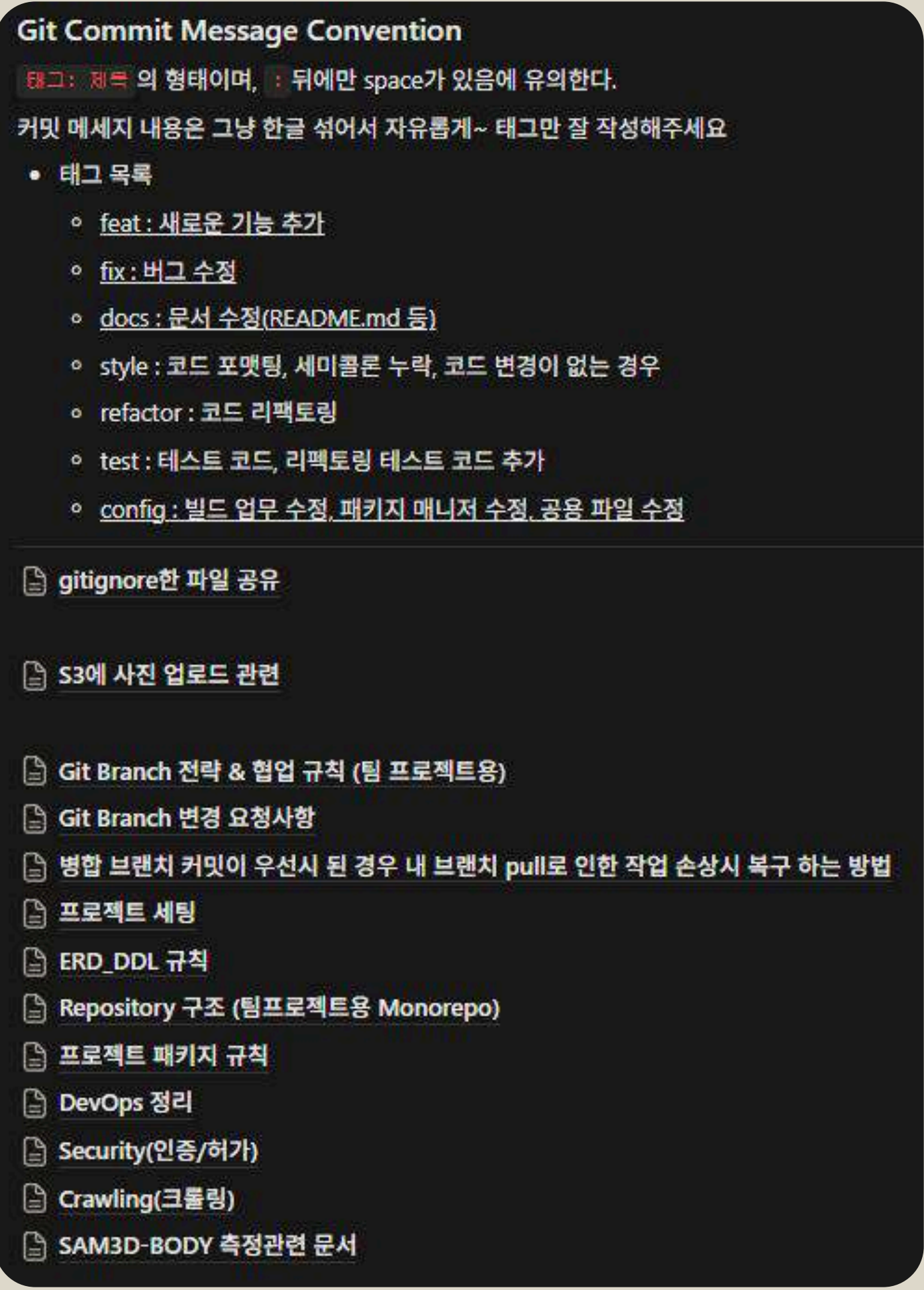
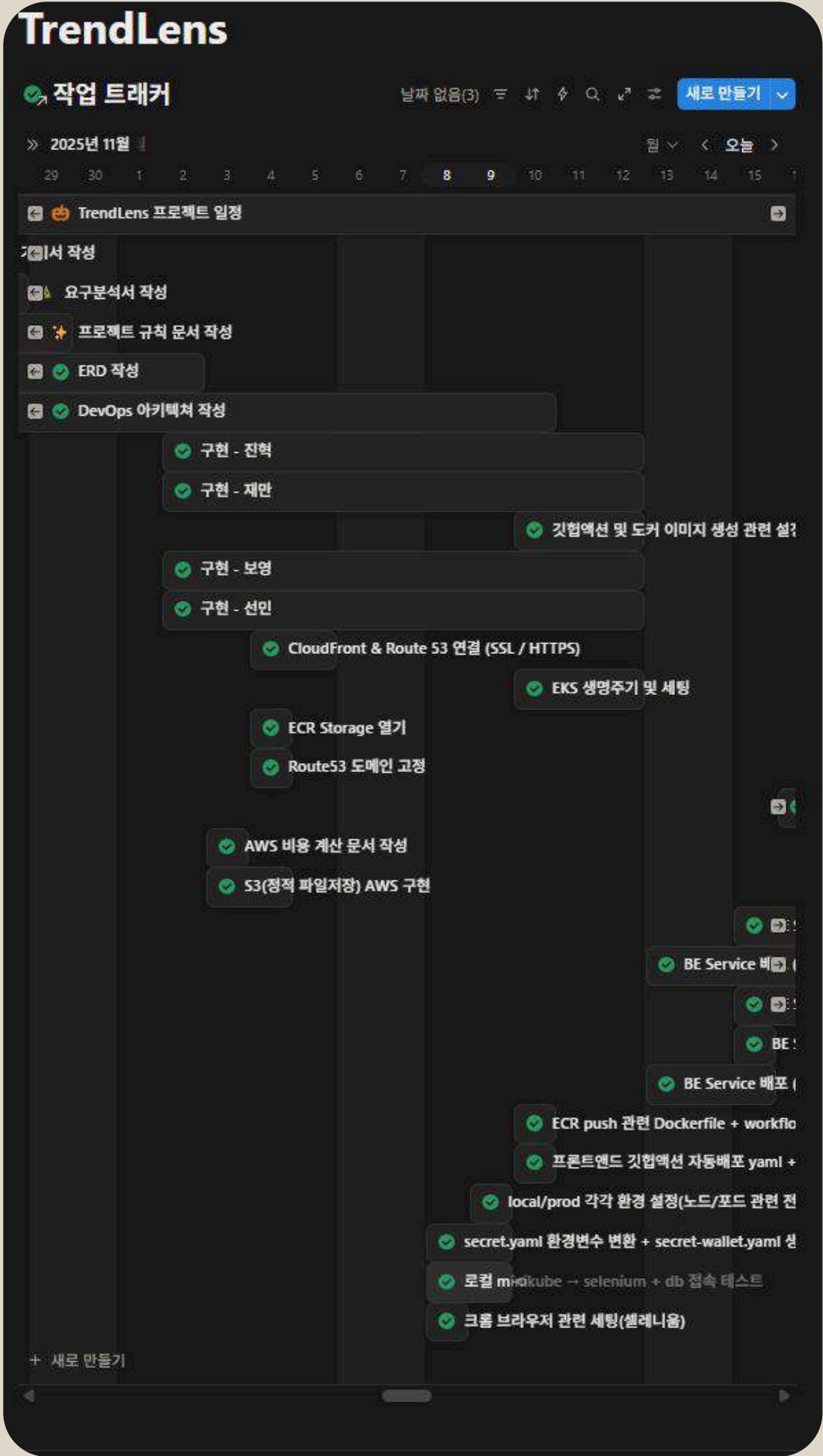
DEV OPS



Github Actions

aws





이전 프로젝트에서는 협업 규칙 없이 진행하면서

커밋 메시지/브랜치 흐름이 제각각이라 히스토리 추적이 어렵고,

담당하지 않은 도메인의 정보가 문서로 남지 않아 중복되는 코드와 시간적

큰 비용이 발생했습니다.

TrendLens에서는 프로젝트 시작 단계에서 Git Commit Convention,

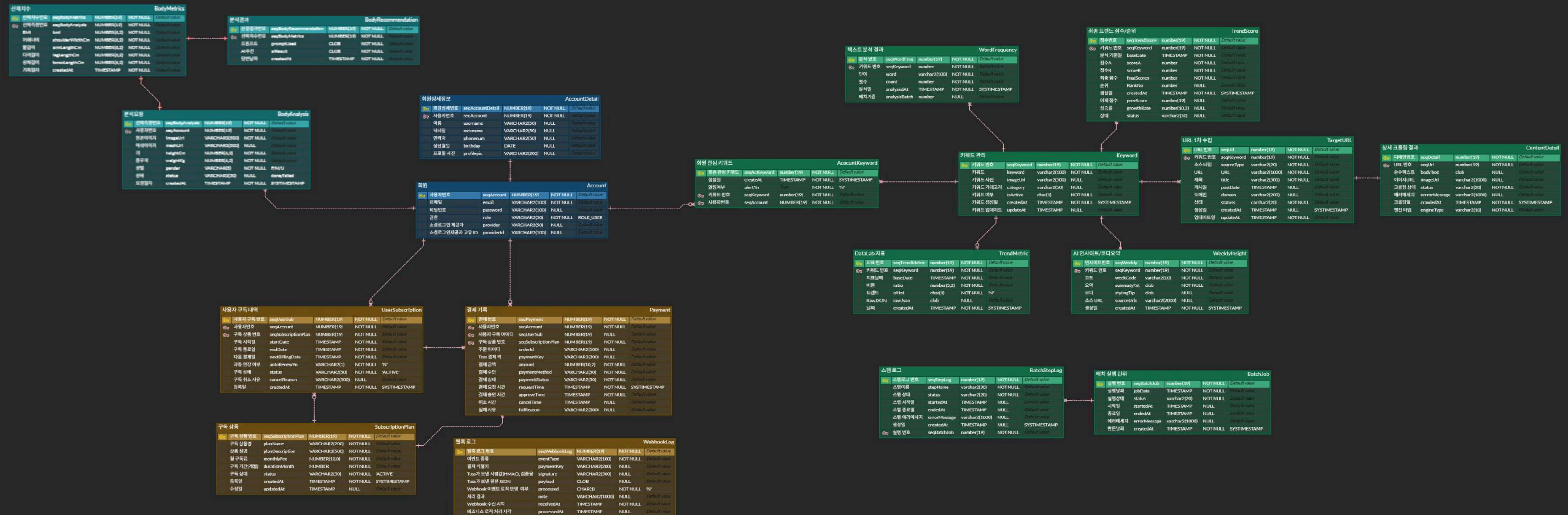
브랜치 전략, 패키지/레포 구조, ERD/DDL 규칙, DevOps·Security 가이드,

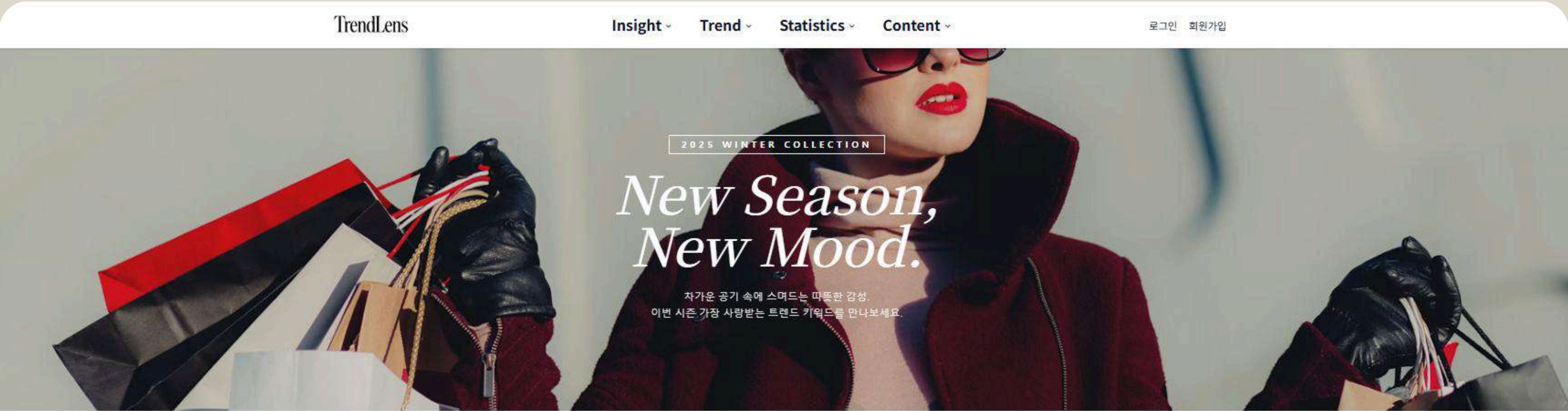
S3 업로드/크롤링/SAM3D 문서를 노션으로 정리하여 공유하였습니다.

작업 트래커(캘린더+타임라인)로 일정·담당·진척도를 가시화해 관리했습니다.

그 결과, 충돌/중복작업을 줄이고 갑작스러운 일정 변경과 도메인 기능 구현 시

시간에 대한 비용 소모 없이 처리 할 수 있었습니다.





Weekly Ranking

2025. 12. 16. UPDATED

KEYWORD SEARCH



INSIGHT BRIEF

Gemini AI가 분석한 금주 트렌드 요약

HOT ISSUE

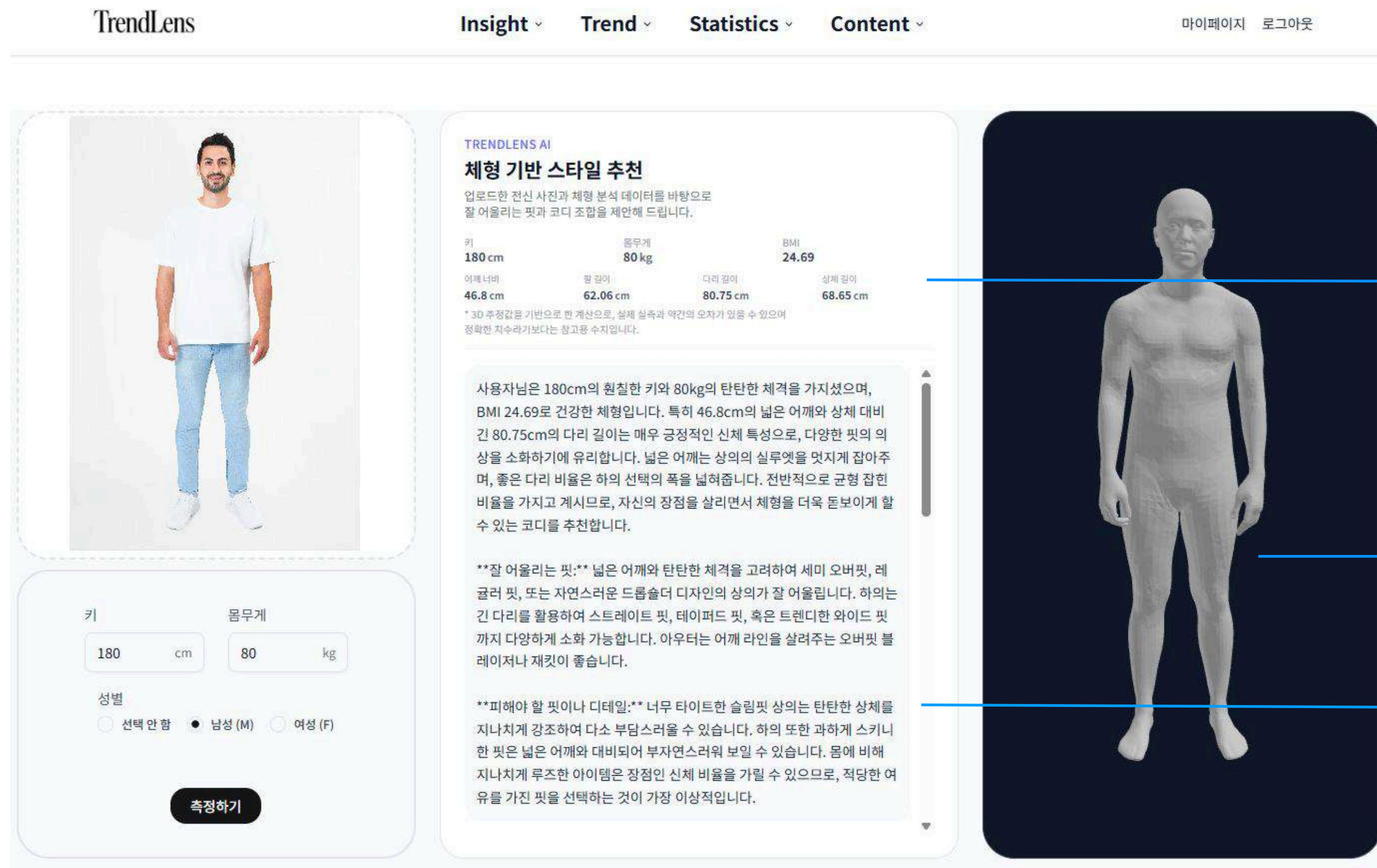
이번 주 가장 뜨거운 키워드,
"247시리즈 TORAY 프리마로프트 패디드 맥
시핏 코트 BLACK"의 급상승 이유

RISEING

아우터 카테고리에서
새롭게 주목받는 브이투 70103 VEGAN LEATHER
MUSTANG_BLACK 스타일링 팁

"전반적으로 기온 하락에 따른 보온성 아이템의 검색량
이 증가하고 있습니다. 특히 무패용 패딩이 인기입니다."

BODY ANALYZE



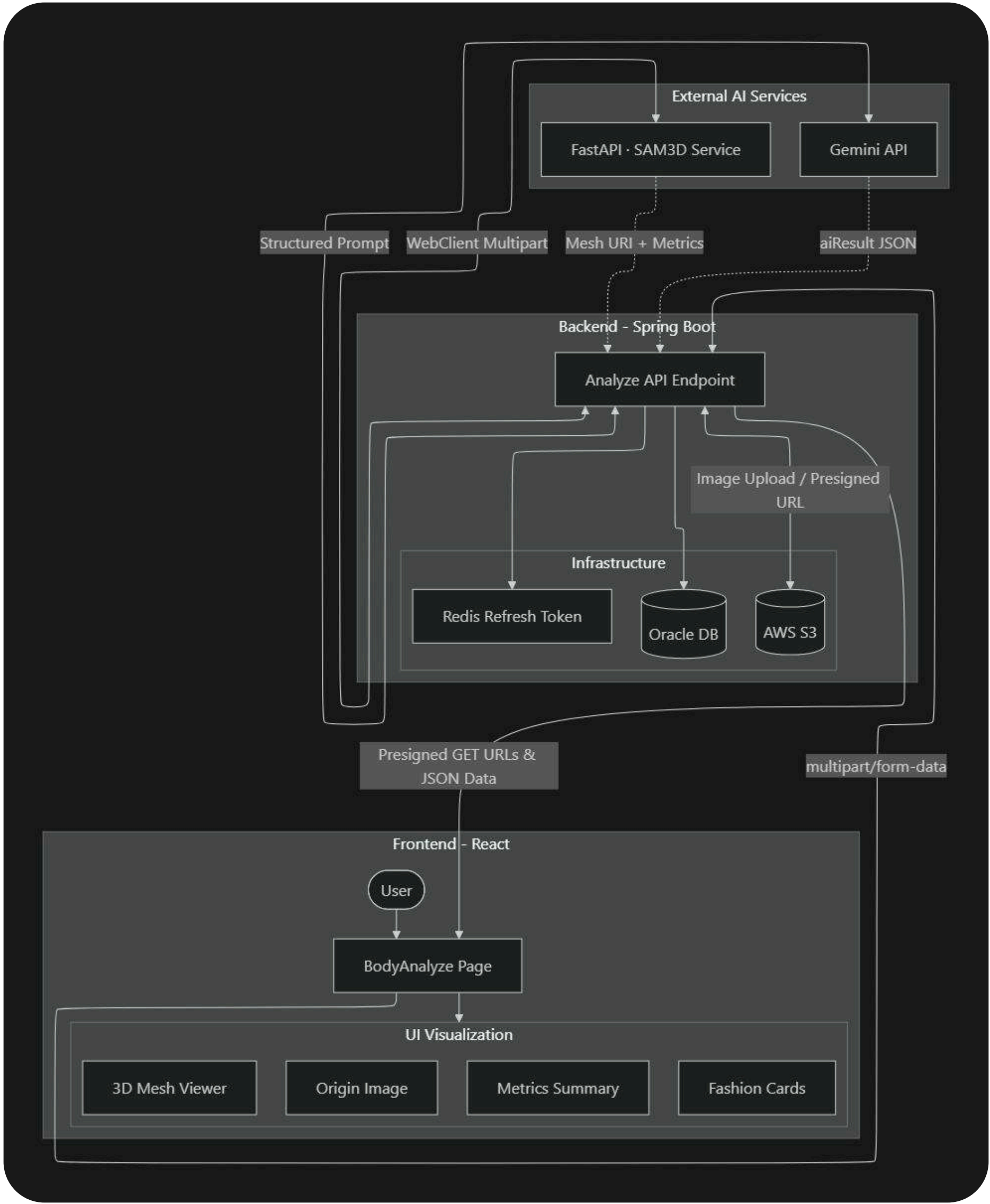
문제

- 트렌드 정보는 많지만 내 체형에 맞는지는 판단이 어렵다.
- 체형 분석 결과가 숫자로 끝나면 신뢰가 약하고,
추천 결과가 텍스트로만 끝나면 설득력이 떨어진다는 판단.

해결

- 3D MESH(OBJ) 를 생성해서 시각적으로 보여줌
- 사용자 입력 키(CM) 로 MESH를 실측 스케일로 정규화
- 치수 지표를 추천 프롬프트에 직접 연결해
“체형 기반 코디”를 구조화(JSON)로 제공.

전신 사진 + 신체정보(키/몸무게/성별)를 입력하면 SAM 3D BODY로 3D MESH를 생성하고
치수 지표를 산출한 뒤, GEMINI 기반 프롬프트로 체형 맞춤 최신 트렌드 코디를 추천하는 웹서비스.

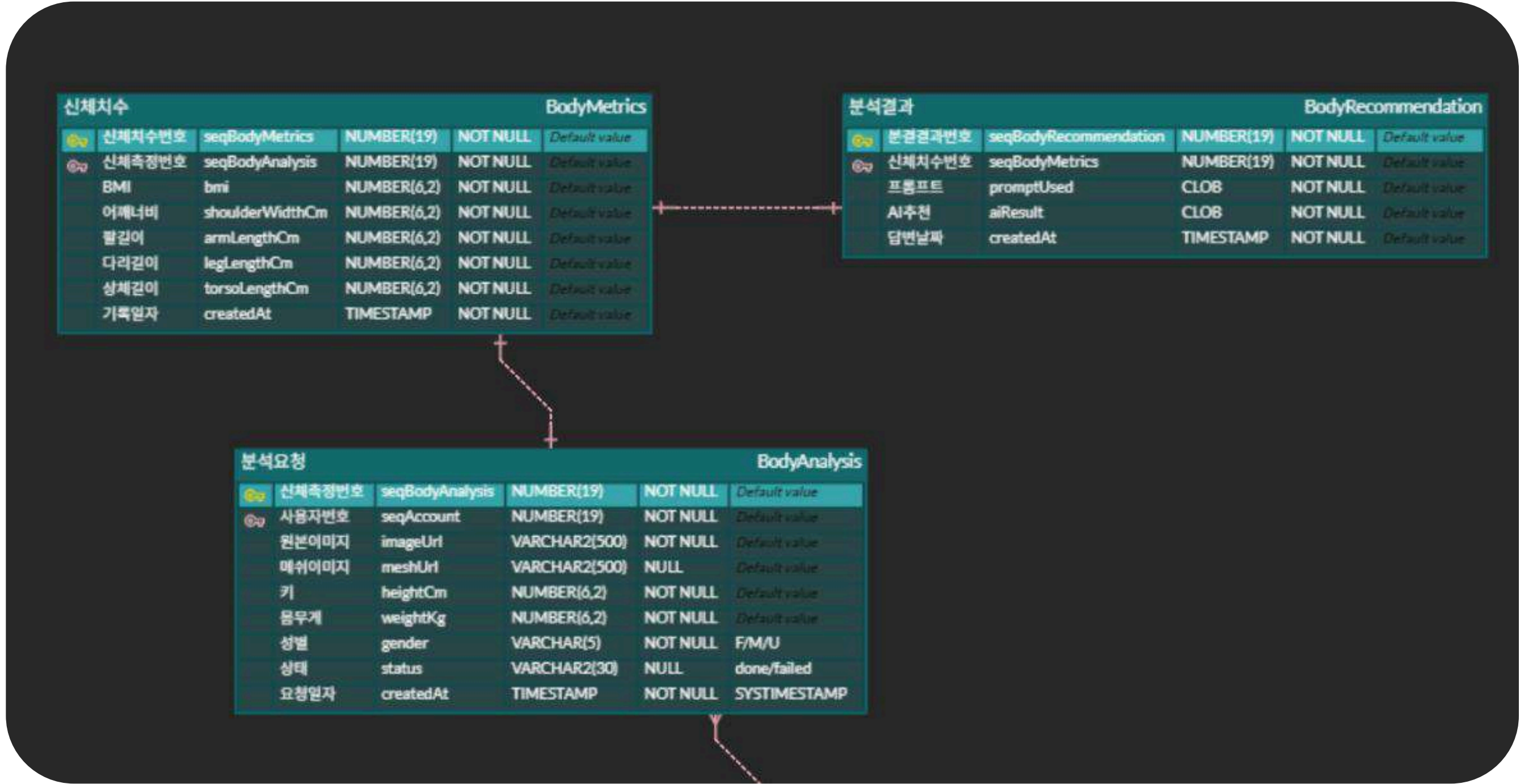


Backend: AI Analysis & Orchestration Layer

| Category | Filename | Role & Description | Key Tech |
|---------------|------------------------------|---|-------------|
| Core | BodyAnalyzeController.java | API Gateway 프론트엔드의 요청을 수신하고 입력 데이터를 검증하는 진입점입니다. | Spring Boot |
| Orchestration | BodyAnalyzeService.java | Pipeline Manager 3D 생성 → S3 업로드 → Gemini 추천으로 이어지는 전체 로직의 순서와 에러를 제어합니다. | Java Core |
| AI Client | Sam3dBodyClient.java | GPU Communication WebClient를 사용해 RunPod(FastAPI)과 비동기 통신하며 3D 메쉬 생성을 요청합니다. | WebClient |
| AI Client | FashionRecommendClient.java | LLM Integration Google Gemini Pro Vision API를 호출하여 시각 정보 기반의 의상 추천을 수행합니다. | Gemini API |
| AI Logic | FashionPromptFactory.java | Prompt Eng LLM이 비즈니스 로직에 맞는 JSON 포맷을 출력하도록 구조화된 프롬프트를 생성합니다. | JSON |
| Infra | BodyImageStorageService.java | S3 Storage 대용량 3D 파일 처리를 위한 AWS S3 Pre-signed URL 발급 및 생명주기를 관리합니다. | AWS S3 |

Frontend: Interactive 3D UI

| Category | Filename | Role & Description | Key Tech |
|----------|-------------------------|--|----------------|
| Page | bodyanalyze.tsx | Main Container 업로드 → 분석 대기 → 결과 확인까지의 전체 사용자 여정(Journey)을 통합 관리합니다. | React |
| 3D Vis | MeshViewerBox.tsx | WebGL Viewer 생성된 .obj 파일을 웹에서 three.js로 실시간 렌더링하며 마우스 인터랙션을 제공하는 3D 뷰어입니다. | Three.js (R3F) |
| Input | OriginImageBox.tsx | Data Interface 사용자 신체 스펙 입력 및 업로드한 원본 이미지를 미리 보여 주는 UI 컴포넌트입니다. | Tailwind |
| Result | FashionRecommendBox.tsx | Visual Output Gemini가 분석한 추천 스타일과 팀을 가독성 좋은 카드 뉴스 형태로 시각화합니다. | UI/UX |



테이블 구성

BODYANALYSIS:
원본/메쉬 URL + 키/몸무게/성별 입력값 저장,
STATUS(DONE/FAILED)로 처리 추적, CREATEDAT으로 히스토리 관리

BODYMETRICS:
BMI·어깨·팔/다리·몸통 등 측정치 수치 저장
(요청과 연결되어 결과 추적 가능)

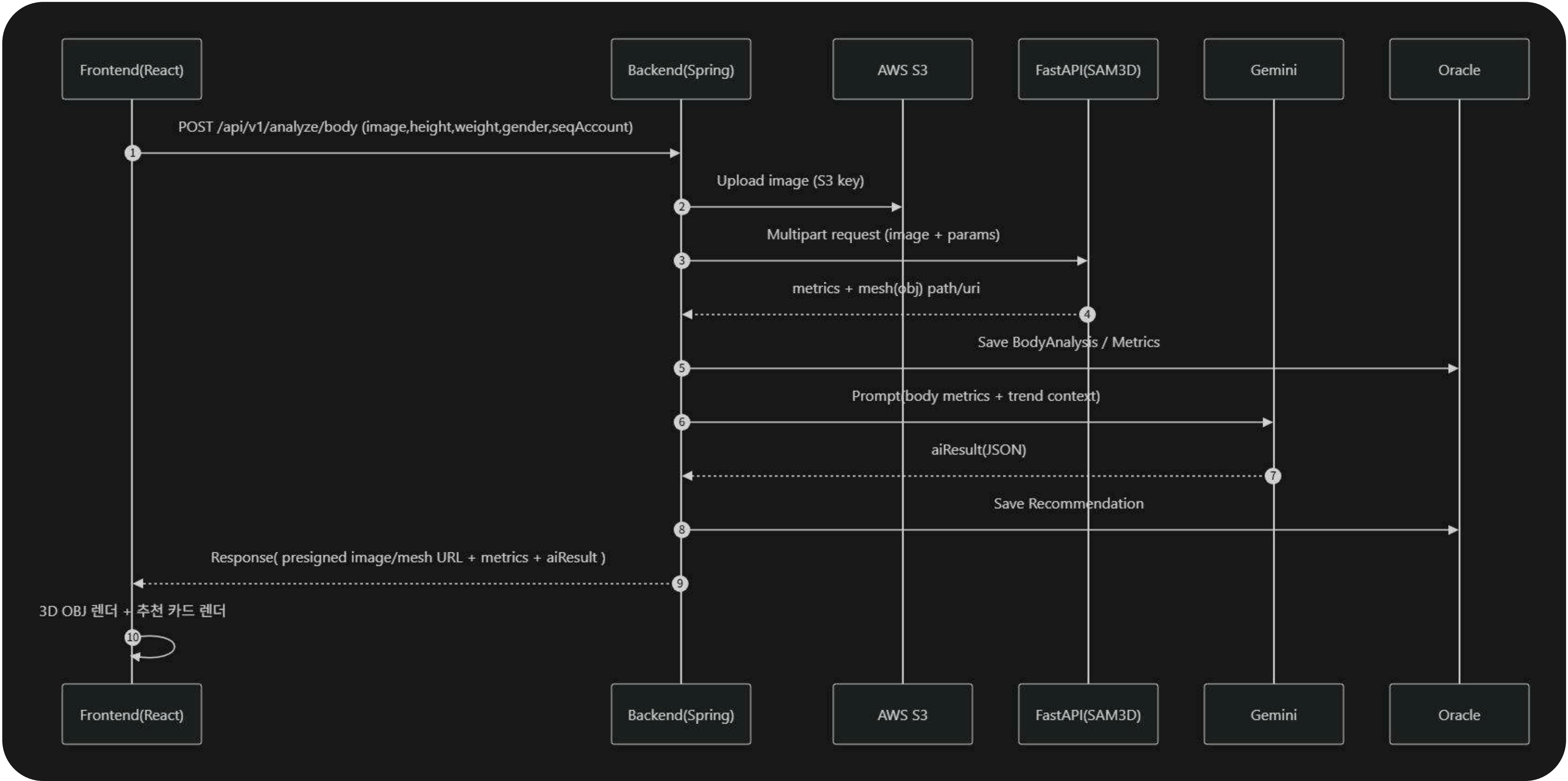
BODYRECOMMENDATION:
GEMINI PROMPTUSED + AIRESULT(CLOB/JSON) 저장
→ 추천 결과 재현/버전관리

관계도

BODYANALYSIS → BODYMETRICS → BODYRECOMMENDATION
단계별로 분리 저장해 부분 실패 추적과 재추천/재처리가 쉬움

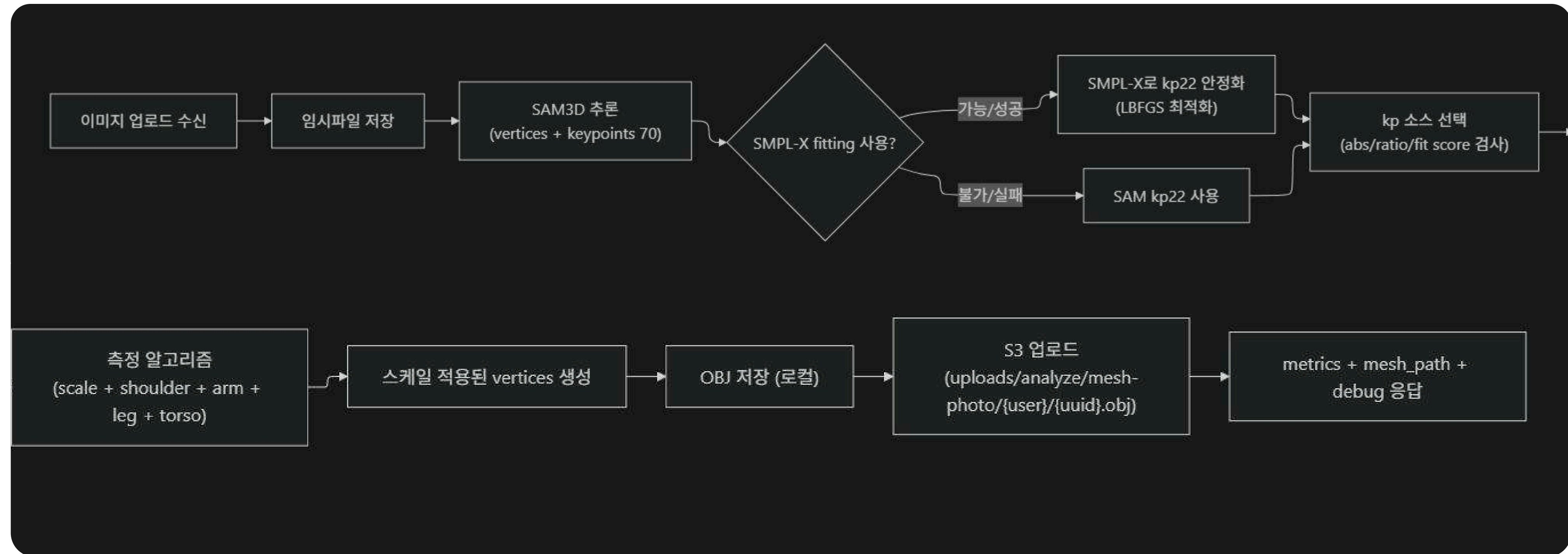
설계 의도

- 지표 확장에 유리한 정규화
- PROMPT/응답 저장으로 재현성 확보
- STATUS/시간 기반으로 운영 디버깅
- 향후 추천 품질 분석·A/B·프롬프트 개선에 활용



분석 요청 FLOW

FastAPI 파이프라인



SMPL-X를 조건부로 붙인 이유

- SAM3D kp는 포즈/가림/품질에 따라 흔들릴 수 있어 끝점 기반 측정에 민감
- SMPL-X는 인체적으로 plausible한 형태로 kp22를 안정화하는 데 유리
- 다만 모든 케이스에 강제하지 않고, 통과 조건(abs/ratio/fit) 만족 시만 채택
- 실패 시 SAM kp로 fallback하여 서비스 안정성 유지

팔/다리 선택 전략(긴 쪽 선택)

- 사진 기반 추정은 가림/굽힘으로 짧게 측정되는 편향이 빈번
- 좌/우를 각각 평가(clip/ratio/segment/outlier) 후 유효한 값만 남김
- 유효 범위 안에서 긴 쪽을 선택해 짧아짐 편향을 완화

TrendLens

Insight

Trend

Statistics

Content

로그인

회원가입

전신 사진을 업로드하세요
(클릭해서 파일 선택)

키

몸무게

cm

kg

성별

선택 안 함

남성 (M)

여성 (F)

측정하기

TRENDAI

체형 기반 스타일 추천

업로드한 전신 사진과 체형 분석 데이터를 바탕으로
잘 어울리는 핏과 코디 조합을 제안해 드립니다.

왼쪽에서 전신 사진과 기본 정보를 입력한 뒤
“체형 측정하기” 버튼을 눌러주세요.
분석이 완료되면 이 영역에 스타일 요약과 코디 추천이 표시됩니다.

측정 대시보드

TrendLens

Insight

Trend

Statistics

Content

마이페이지

로그아웃

키

180

cm

몸무게

80

kg

성별

선택 안 함

남성 (M)

여성 (F)

측정하기

TRENDLENS AI

체형 기반 스타일 추천

업로드한 전신 사진과 체형 분석 데이터를 바탕으로 잘 어울리는 핏과 코디 조합을 제안해 드립니다.

키

180 cm

몸무게

80 kg

BMI

24.69

어깨 너비

46.8 cm

팔 길이

62.06 cm

다리 길이

80.75 cm

상체 길이

68.65 cm

* 3D 추정값을 기반으로 한 계산으로, 실제 실측과 약간의 오차가 있을 수 있으며 정확한 치수라기보다는 참고용 수치입니다.

사용자님은 180cm의 훤칠한 키와 80kg의 탄탄한 체격을 가지셨으며, BMI 24.69로 건강한 체형입니다. 특히 46.8cm의 넓은 어깨와 상체 대비 긴 80.75cm의 다리 길이는 매우 긍정적인 신체 특성으로, 다양한 핏의 의상을 소화하기에 유리합니다. 넓은 어깨는 상의의 실루엣을 멋지게 잡아주며, 좋은 다리 비율은 하의 선택의 폭을 넓혀줍니다. 전반적으로 균형 잡힌 비율을 가지고 계시므로, 자신의 장점을 살리면서 체형을 더욱 돋보이게 할 수 있는 코디를 추천합니다.

잘 어울리는 핏: 넓은 어깨와 탄탄한 체격을 고려하여 세미 오버핏, 레귤러 핏, 또는 자연스러운 드롭숄더 디자인의 상의가 잘 어울립니다. 하의는 긴 다리를 활용하여 스트레이트 핏, 테이퍼드 핏, 혹은 트렌디한 와이드 핏 까지 다양하게 소화 가능합니다. 아우터는 어깨 라인을 살려주는 오버핏 블레이저나 재킷이 좋습니다.

피해야 할 핏이나 디테일: 너무 타이트한 슬림핏 상의는 탄탄한 상체를 지나치게 강조하여 다소 부담스러울 수 있습니다. 하의 또한 과하게 스키니한 핏은 넓은 어깨와 대비되어 부자연스러워 보일 수 있습니다. 몸에 비해 지나치게 루즈한 아이템은 장점인 신체 비율을 가릴 수 있으므로, 적당한 여유를 가진 핏을 선택하는 것이 가장 이상적입니다.

남성 측정 ver

TrendLens

Insight
Trend
Statistics
Content

마이페이지
로그아웃

키
160
cm
몸무게
60
kg

성별

☐ 선택 안 함
☐ 남성 (M)
☒ 여성 (F)

측정하기

TRENDLENS AI

체형 기반 스타일 추천

업로드한 전신 사진과 체형 분석 데이터를 바탕으로 잘 어울리는 핏과 코디 조합을 제안해 드립니다.

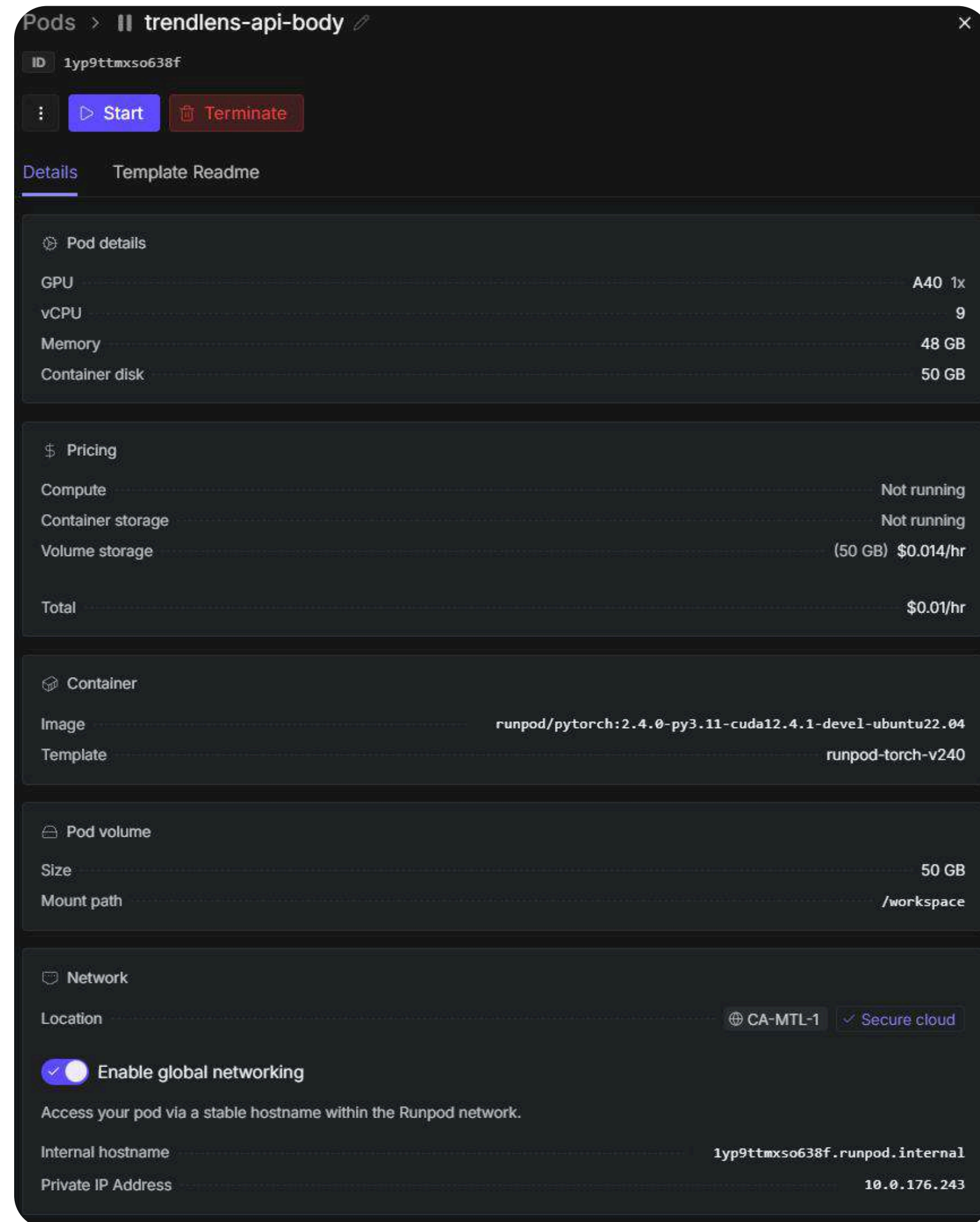
| | | | |
|---------|---------|----------|----------|
| 키 | 몸무게 | BMI | |
| 160 cm | 60 kg | 23.44 | |
| 어깨 너비 | 팔 길이 | 다리 길이 | 상체 길이 |
| 36.8 cm | 44.8 cm | 74.63 cm | 58.17 cm |

* 3D 추정값을 기반으로 한 계산으로, 실제 실측과 약간의 오차가 있을 수 있으며 정확한 치수라기보다는 참고용 수치입니다.

신체 정보에 따르면, 키(160cm), 몸무게(60kg), BMI(23.44)는 표준 체형에 속하며 균형 잡힌 실루엣을 가지고 계십니다. 어깨너비(36.8cm)는 평균적이지만, 팔 길이(44.8cm)가 다소 짧은 편이므로 소매 길이에 유의하여 상의를 선택하는 것이 중요합니다. 다리 길이(74.63cm)와 상체 길이(58.17cm)를 고려할 때, 하이웨스트 하의를 통해 다리를 시각적으로 길어 보이게 연출하는 것이 좋습니다.

체형에 잘 어울리는 핏으로는 허리선을 강조하는 하이웨스트 하의와 크롭 기장의 상의/아우터가 잘 어울려 다리 길이를 시각적으로 늘려줄 수 있습니다. 상의와 아우터는 레귤러 또는 살짝 오버핏을 선택하여 편안함과 트렌디함을 동시에 잡을 수 있으며, 하의는 스트레이트 또는 와이드 핏으로 체형 커버와 동시에 스타일리시함을 더할 수 있습니다. 어깨가 너무 좁아 보이지 않도록 약간 드롭되거나 오버핏의 상의/아우터도 좋습니다.

피해야 할 핏이나 디테일은 다음과 같습니다. 팔 길이가 긴 상의는 피하거나 소매를 걷어 연출하는 것이 좋습니다. 또한 다리가 짧아 보일 수 있는 로우라이즈 하의나 키를 작아 보이게 할 수 있는 너무 길고 부해 보이는 기장의 상의/아우터는 주의하는 것이 좋습니다. 전반적으로 자신의 장점을 부각하고 단점을 자연스럽게 보완하는 코디를 추천합니다.



AWS 프리티어 한계로 GPU 서버를 못 빌림 → RunPod로 분리 구축

팀 프로젝트를 AWS 무료 크레딧 기반으로 배포하려 하였으나

프리티어 환경에서는 GPU 인스턴스를 사실상 확보할 수 없었습니다.

BodyAnalyze는 SAM3D-Body로 3D mesh를 생성하는 구간에서 병목이 발생하는데,
이 작업은 GPU가 없으면 응답 시간이 급격히 길어져 데모수준이지만 실제로 서비스 가능한
수준까지 응답시간을 올리고 싶었습니다.

그래서 추론 파이프라인만 별도 GPU 서비스로 분리했고,

RunPod에서 A40 GPU 인스턴스를 대여해 **FastAPI 서버**를 구축했습니다.

SAM3D-Body 모델은 Hugging Face에서 리소스를 로딩하도록 구성하고
Spring Boot(WebClient)에서 FastAPI로 멀티파트 요청을 보내는 구조로 연결해
AWS 프리티어 배포 제약과 성능 요구를 동시에 해결했습니다.



```
1 #!/bin/bash
2 set -e
3
4 echo "SAM 3D Body API 시작..."
5
6 cd /workspace
7
8 if [ ! -d "venv" ]; then
9     echo "가상환경이 없습니다"
10    exit 1
11 fi
12
13 source venv/bin/activate
14
15 cd sam-3d-body
16
17 if [ ! -f "app.py" ]; then
18     echo "app.py 파일이 없습니다"
19     exit 1
20 fi
21
22 echo "GPU 정보:"
23 nvidia-smi --query-gpu=name,memory.total,memory.free --format=csv
24
25 echo ""
26 echo "API 서버 시작 (http://0.0.0.0:8000)"
27 echo ""
28
29 python app.py
```

For detailed documentation and guides, please visit:
<https://docs.runpod.io/> and <https://blog.runpod.io/>

root@69348cd69f1e:/workspace# ./run_api.sh
SAM 3D Body API 시작...
GPU 정보:
name,memory.total [MiB], memory.free [MiB]
NVIDIA A40, 46068 MiB, 45489 MiB
API 서버 시작 (http://0.0.0.0:8000)

INFO: Started server process [488]
INFO: Waiting for application startup.

RunPod 파드 재시작 시 환경이 초기화되는 문제 → venv 기반으로 재현 가능한 런타임 고정

RunPod 환경을 쓰면서 예상 못한 문제가 하나 있었습니다.

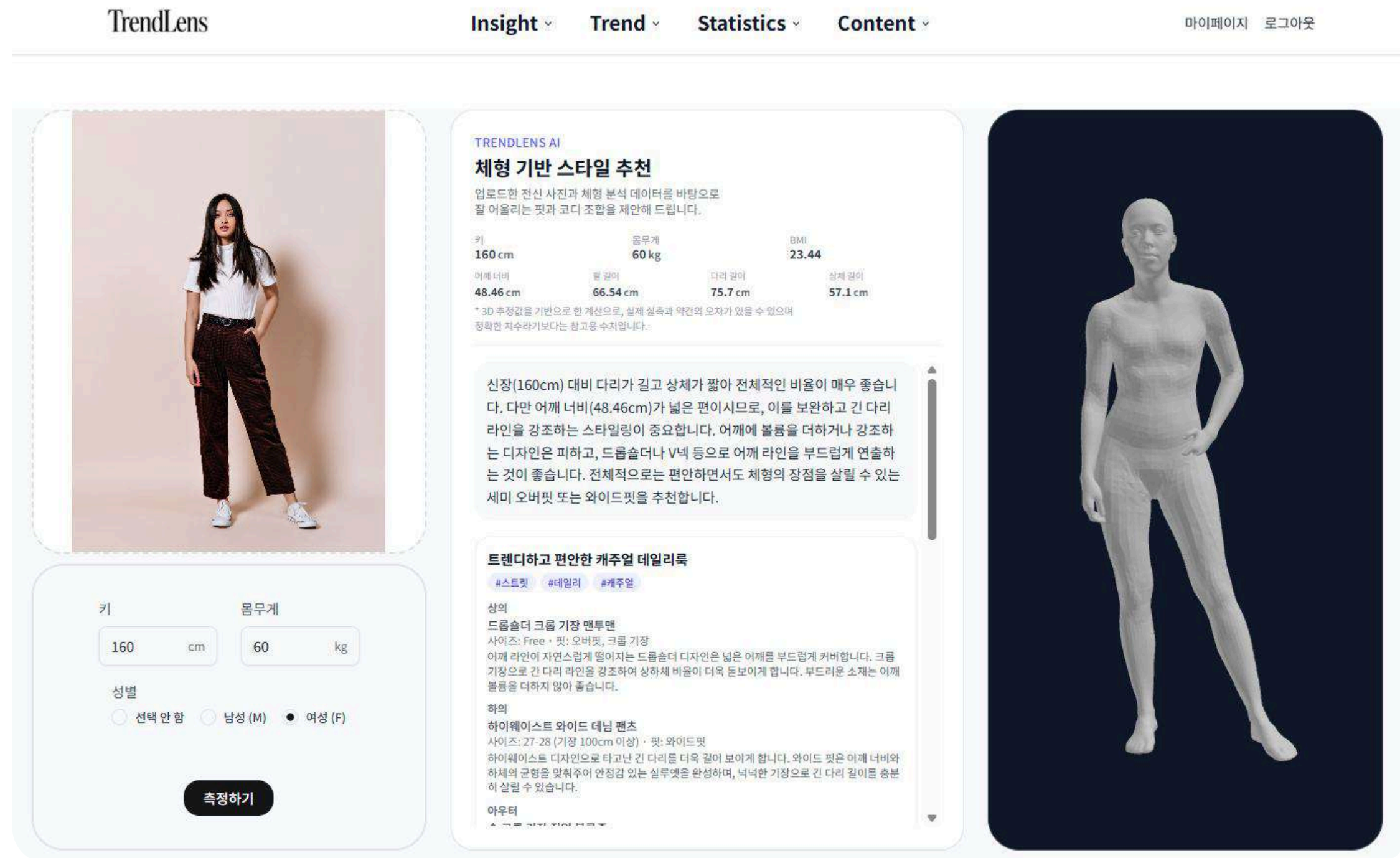
파드를 꺾다 켜면 workspace를 제외한 영역이 초기화되면서, 설치했던 파이썬 디펜던시와 설정이 통째로 초기화 되었습니다.

처음에는 “다시 설치하면 되지”라고 접근했는데, torch/관련 라이브러리 설치 시간이 매우 길었고, 매번 수동 셋업을 반복하는 건 운영 관점에서 리스크가 크다고 생각했습니다.

해결 전략은 **의존성 설치를 workspace로 고정하는** 것이었습니다.

- workspace 내부에 venv 가상환경을 생성
- 필요한 패키지 버전을 고정해 설치
- 재시작 후에도 run_api.sh만 하면 동일 환경이 즉시 복구되도록 구성 및 bash 파일 생성

결과적으로 서버 재기동/스케일링 같은 운영 시나리오에서도 환경이 흔들리지 않게 되었고, 추론 서버를 개발단계에서 **한 번 만들어두고 오래 쓰는 형태로** 안정화할 수 있었습니다.



짹다리 & 주머니 손 사진

```
✓ SMPLX fit done | choose={'used': False, 'reason': 'smpl_bad_abs_or_ratio_or_fit', 'sam_h_m': 1.1423929929733276, 'sam_sh_m': 0.18667182326316833, 'smpl_h_m': 0.8248018622398376, 'smpl_sh_m': 0.23625554144382477, 'ratio_h': 0.7219948540590313, 'ratio_sh': 1.2656197240370537, 'fit_rmse': 0.20347656309604645, 'fit_mse': 0.04140270873904228, 'abs_ok': True, 'ratio_ok': True, 'fit_ok': False, 'smpl_h_ok': True, 'smpl_sh_ok': True, 'scale_ok': True}
INFO: 100.64.1.1:52288 - "POST /analyze-body HTTP/1.1" 200 OK
```

SMPL-X 피팅 적용 여부 확인 로그

메쉬 기반 측정만으로는 포즈/가림에 민감 → SMPL-X를 조건부로 도입

초기 버전에서는 SAM3D가 생성한 MESH와 KEYPOINT를 그대로 사용해 신체 치수 (어깨/팔/다리/상체)를 계산했습니다.

실제 사용자 입력을 테스트해보니 특정 포즈나 촬영 조건에서 측정값이 허용범위 오차를 벗어났습니다.

- 팔이 몸에 붙거나 살짝 굽혀진 포즈(A포즈에서도 각도/가림이 발생)
- 손목/손끝 같은 끝점이 가려져 KEYPOINT가 튜는 경우
- 짹다리/골반 기울어짐처럼 체형이 비대칭으로 잡히는 경우

이 상태에서 MESH만 믿으면 측정값이 튜는 상황을 컨트롤하기 힘들었습니다.

그래서 구글링 시 인체 구조적으로 관절 배치를 제공하는 SMPL-X를 알게되었고, KP(KEYPOINT)22 안정화 목적으로 FITTING을 도입했습니다.

다만 SMPL-X를 무조건 사용하는 것이 아니라

- FITTING이 성공하더라도 ABS 범위/비율/FIT RMSE 기준을 통과할 때만 채택
- 기준을 만족하지 못하면 기존 SAM KP로 FALLBACK

결론적으로 SMPL-X는 불안정한 케이스에서 관절을 안정화하는 보조 장치로 설계했고,

이 방식으로 포즈/가림 케이스에서 측정의 일관성이 좋아졌습니다.

또한 실패 시에도 서비스가 멈추지 않도록 안정성을 확보했습니다.

프로젝트를 시작할 때 가장 먼저 했던 건 무엇을 만들지를 정하기 전에, 팀원들이 각자 구현해보고 싶은 도메인과 역할을 솔직하게 공유하는 일이었습니다.

역할분담을 위한 대화가 아니라, 서로가 몰입할 수 있는 주제를 찾는 과정이었던 것 같습니다. 진솔하게 이야기했던 대화를 바탕으로 공통 관심사를 정리해보니 자연스럽게 패션이라는 키워드로 합쳐지게 되었고, TRENDLENS라는 패션 매거진 및 AI를 이용한 콘텐츠를 제공하는 웹서비스쪽으로 방향성이 빠르게 잡혔습니다.

기획 단계에서 팀원 중 한분이 크롤링을 담당하겠다고 하셨는데 크롤링 시 수집되는 트렌드 데이터를 그냥 메인페이지나 클라이언트에게 노출하는 것이 아니라, 수집된 데이터를 기반으로 LLM을 직접 만들어 학습시키고, 그 결과로 최신 트렌드와 사용자 체형을 동시에 반영하여 패션 아이템을 추천하는 서비스를 만드는 흐름을 구상했습니다. 그러나 실제로는 로컬 PC 환경, 제한된 개발 기간, 그리고 운영 안정성 같은 현실적인 조건을 고려했을 시에 직접 학습시킨 LLM을 제공할 수 없었기에 최종적으로는 자체 학습 모델 대신 GEMINI API를 사용해 패션 아이템 추천을 구현했습니다.

아쉬움이 남는 선택이긴 했지만, 반대로 보면 '지금 가능한 자원 안에서 서비스를 완성시키는 판단'이기도 했던 것 같습니다. 생각해보면 기술적으로도 기억에 남는 것들이 많습니다. 팀이 원하는 사용자 경험을 구현하기 위해 REACT를 도입했는데, 라우팅, 상태 관리, API 연동, 빌드/배포까지 하나씩 처음부터 학습하며 개발해야했습니다. 결국 그런 우여곡절 덕분에 SPA가 주는 장점이 프로젝트에서 어떻게 힘을 발휘하는지 체감했습니다.

특히 BODYANALYZE처럼 입력, 분석, 결과 렌더링이 빠르게 이어져야 하는 흐름에서, 화면 전환 없이 부드럽게 상태를 유지하고 사용자 경험을 설계할 수 있다는 점이 크게 와닿았습니다. 무엇보다 좋았던 건 팀 분위기였습니다. 문제가 생기면 "누가 잘못했냐"가 아니라 "어떻게 같이 해결하냐"로 움직이는 팀이었던 것 같습니다.

디버깅과 원인 분석을 같이 하고, 해결 과정을 공유하면서 프로젝트 속도가 오히려 빨라지는 경험을 했으며 혼자였으면 시간 비용이 많이 발생했을 이슈들도 팀원들과 함께 고민하며 해결했던 것들이 좋은기억으로 남아있어 앞으로도 협업 시에 이런 좋은 기억을 꺼내어 진취적으로 비단 내가 담당한 부분의 문제가 아니더라도 해결하려 해볼 것 같습니다.

개인으로는 이번 프로젝트에서 PYTHON을 처음 다뤄본 것이 인상 깊게 남습니다. FASTAPI로 추론 서버를 구성하고, 모델 결과를 처리하면서 공부하기 위해 배우는 언어가 아니라 문제를 해결하려고 배우는 언어로 PYTHON을 경험했던 것 같습니다. 그래서 다음에 본격적으로 학습할 언어로 PYTHON을 고려하고 있으며 물론 늘 저답게 무언가 만들어보면서 공부할 생각입니다.

이 프로젝트를 팀원들과 진행하면서 얻어가는 것 중에 가장 크다고 느끼는건 하나의 웹 서비스를 만든 경험을 넘어서, AI 시대에 내가 어떤 방향으로 성장하고 싶은지를 더 선명하게 그릴 수 있었던 것 같습니다. 앞으로는 AI를 써보는 것에서 끝내지 않고, 데이터와 모델과 제품 경험을 연결해서 실제로 가치가 되는 서비스를 더 많이 만들어보고 싶다는 생각을 했습니다.