붙임 ① (표준서식, 훈련생 작성용)

IT 프로젝트 개발 계획서(안)

(「자체 수집 데이터 기반 멀티모달 AI와 LLM을 활용한 PC 가격예측·추천 시스템」)

2026.01.20~2026.02.10

훈련과정명 (소속)	클라우드 데브옵스 프론트엔드&백엔드 자바(JAVA) 풀스택 (중앙정보기술인재개발원)
팀 명	브리또
팀장 성명	서재홍
팀원 성명	박윤호
팀원 성명	하성호
팀원 성명	이주희
지도교사	정지웅

1. 개 요

□ 프로젝트명

자체 수집 데이터 기반 멀티모달 AI와 LLM을 활용한 PC 가격예측·추천 시스템

□ 개발 배경 및 필요성

- 중고 PC 및 조립 PC 시장에서 소비자가 적정 가격을 판단하기 어려운 문제가 존재함
- 다나와 등 가격 비교 사이트의 데이터를 활용하여 AI 기반 가격 분석을 제공하고자 함
- 상품 이미지와 스펙을 동시에 분석하는 멀티모달 AI를 통해 보다 정확한 가격 예측이 가능함
- 벡터 데이터베이스(Weaviate)를 활용한 이미지 유사도 검색으로 비슷한 상품 비교 기능을 제공함
- 사용 목적(사무, 게임, 개발, 디자인, AI)에 따른 CPU/GPU 벤치마크 기반 PC 추천기능이 필요함
- 로컬 LLM(Ollama)을 활용하여 추천 사유를 자연어로 자동 생성, 사용자 이해도를 높이고자함

□ 개발목표

- 3개의 독립적 서브시스템으로 구성된 통합 PC 분석 플랫폼 구축
- 시스템 1: EfficientNet 앙상블 기반 AI 가격 예측 시스템
- 시스템 2: CLIP + Weaviate 기반 이미지 유사도 검색 시스템
- 시스템 3: 벤치마크 기반 용도별 PC 추천 + Ollama AI 설명 시스템
- Spring Boot + Flask 이중 백엔드 구조와 Vue.js 프론트엔드로 통합 웹 서비스 제공

□ 시스템 구성 개요

본 프로젝트는 3개의 독립적인 서브시스템으로 구성되며, 각 시스템은 별도의 AI 모델과 데이터 파이프라인을 가진다.

구분	시스템1	시스템2	시스템3
명칭	AI 가격예측	이미지 유사도 검색	용도별 PC 추천
핵심 AI	EfficientNet 4-모델 앙상블	CLIP (VIT-B/32)	Ollama LLM (exaone3.5)
데이터 소스	다나와 실시간 크롤링	Weaviate 벡터 db	Passmark 벤치마크
백엔드	Flask (5000)	Flask (5000)	Spring Boot(8083)
DB	price_prediction	Weaviate(8081)	danawa_new_db
입력	다나와 상품 URL	상품이미지	사용 목적 선택
출력	적정가격 + 매수판단	유사 상품 Top-N	추천 PC 10선 + AI 사유

□ 기존 서비스(플랫폼)에 관한 고찰

- 기존 가격비교 사이트는 단순 가격 나열에 그치며 AI 기반 적정가 분석 기능이 부재함
- 이미지 기반 유사 상품 검색 기능을 제공하는 PC 가격비교 서비스는 현재 없음
- 사용 목적에 따라 CPU/GPU 성능 가중치를 다르게 적용하여 맞춤 추천하는 서비스가 부족함
- 본프로젝트는 스펙 + 이미지를 결합한 멀티모달 AI 분석과 벤치마크 기반 맞춤 추천으로 차별화된 서비스를 구현함

□ 산출물

산출물	설명	담당
ERD	데이터베이스 구조 설계서	전체
시스템 아키텍처	서비스 구조 및 기술 스택 문서	서재홍
멘토링 일지		박윤호
발표 PPT	최종 발표 자료	서재홍

2. 서브시스템별 상세 설계

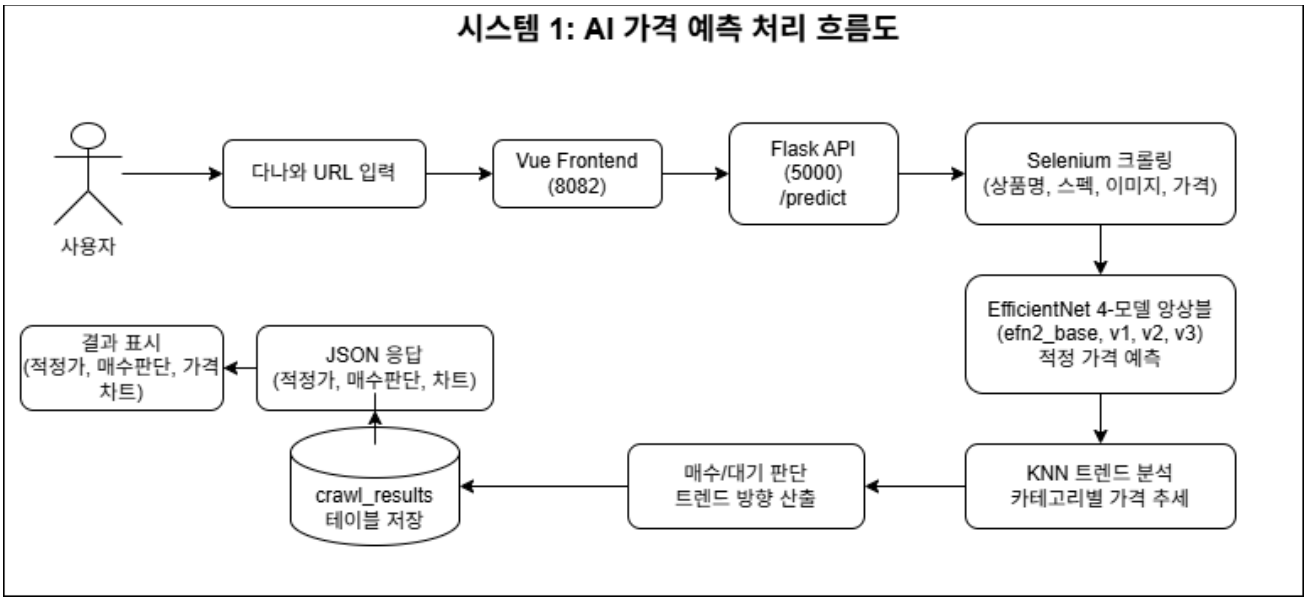
□ 시스템1 : AI 가격 예측 시스템

개요

다나와 상품 URL을 입력하면 실시간 크롤링으로 상품 정보를 수집하고, EfficinetNet 4-모델 앙상블로 적정 가격을 예측하여 매수/대기 판단을 제공하는 시스템이다.

주요기능

기능	상세설명
다나와 크롤링	Selenium 기반 다나와 상품 페이지 실시간 크롤링, 상품명/스펙/이미지/가격 자동 수집
AI 가격 예측	EfficientNet 4-모델 앙상블(efn2_base, v1, v2, v3)으로 이미지 + 스펙 기반 적정 가격 예측
가격 트렌드 분석	KNN 기반 카테고리별 가격 추세 분석, 시계열 가격 이력 차트 제공
매수 판단	예측가 대비 현재가 비교로 매수/대기 판단 및 추천 유형 제공
크롤링 결과 저장	에측 결과를 crawl_results 테이블에 저장하여 이력 관리



기술 스택

구성요소	기술	역할
백엔드	Flask (Python 3.x)	AI 예측 API 서버 (:5000)
AI 모델	EfficientNet (PyTorch)	4-모델 앙상블 가격 예측
크롤링	Selenium WebDriver	다나와 실시간 크롤링
트렌드 분석	scikit-learn KNN	가격 추세
DB	PostgreSQL(price_prediction)	예측 결과 저장

DB 테이블 (price_prediction)

테이블	주요 컬럼	설명
products	id, product_code, name, specifications(JSONB), manufacturer, category_id	상품 마스터
price_history	id, product_id(FK), price, crawled_date	가격 이력

product_benchmarks	id, product_id(FK), cpu_korean_name, cpu_score, gpu_score	상품별 벤치마크 매핑
cpu_mapping_rules	id, Korean_name, English_name, cpu_mark, source	CPU 한글명 Passmark 매핑
gpu_mapping_rules	id, gpu_name, score, price_usd, gpu_id	GPU명 점수 매핑
crawl_results	id, product_code, url, name, price, predicted_price, rec_type, trend_direction	크롤링 + 예측 결과

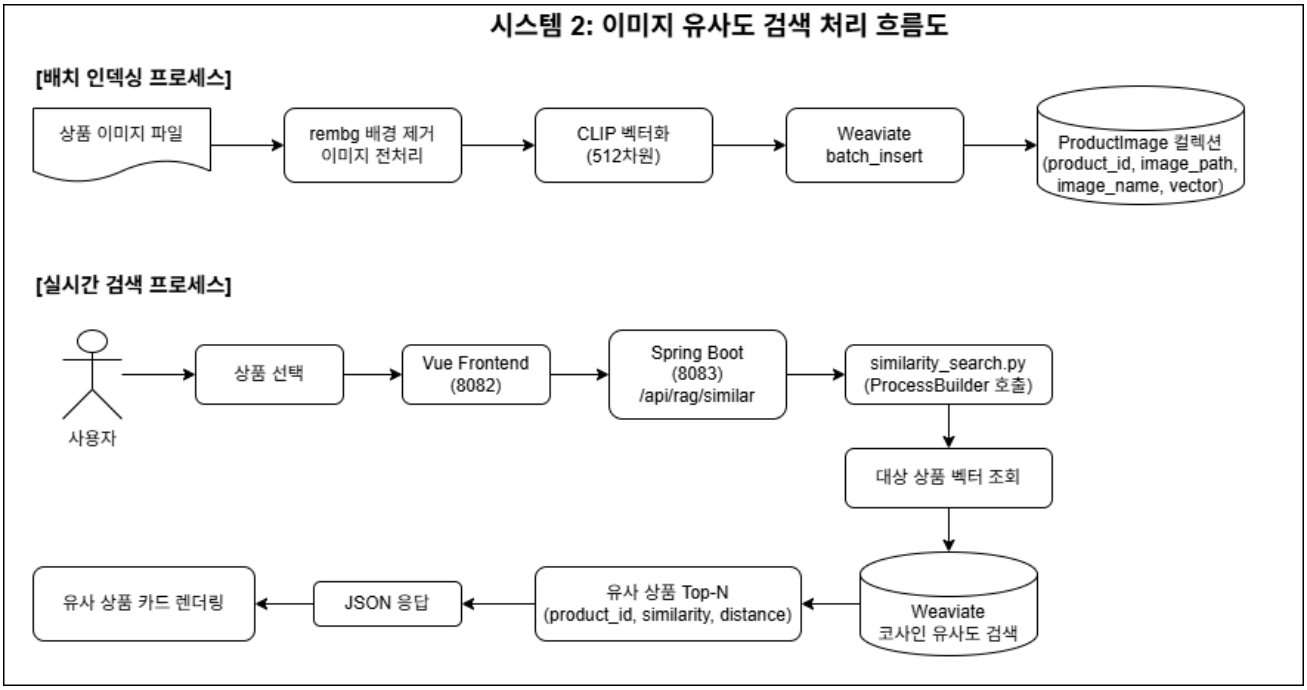
□ 시스템 2 : 이미지 유사도 검색 시스템

개요

CLIP 모델로 상품 이미지를 512차원 벡터로 변환하고, Weaviate 벡터 DB에서 코사인 유사도 기반으로 외관이 유사한 상품을 검색하는 시스템이다.

주요 기능

기능	상세 설명
이미지 전처리	rembg(onnxruntiiome-gpu) 기반 배경 제거로 상품 이미지 정제
이미지 벡터화	CLIP(openai/clip-vit-base-patch32)으로 상품 이미지를 512차원 벡터로 변환
벡터 인덱싱	변환한 벡터를 Weaviate ProductImage 컬렉션에 배치 인덱싱
유사도 검색	쿼리 이미지의 벡터와 코사인 유사도 기반 Top-N 유사 상품 검색
필터링 검색	product_id 기반 필터를 적용한 조건부 유사도 검색



기술 스택

구성요소	기술	역할
이미지 전처리	rembg + onnxruntime-gpu	상품 이미지 배경 제거
AI 모델	CLIP(open ai/clip-vit-base-patch32)	이미지 -> 512차원 벡터 변환
벡터 DB	Weaviate 1.28.4 (Docker)	벡터 저장 및 유사도 검색 (:8081)
인덱싱	batch_indexing.py	상품 이미지 일괄 벡터화 및 저장
검색	similarity_search.py	코사인 유사도 기반 검색
연동	Spring Boot ProcessBuilder	Python 스크립트 호출

Weaviate 컬렉션 스키마

컬렉션	필드	타입	설명
ProductImage	product_id	TEXT	상품코드 (예: 207861.jpg)
	image_path	TEXT	이미지 파일 경로
	image_name	TEXT	이미지 파일명
	vector	FLOAT[512]	CLIP 임베딩 벡터

□ 시스템3 : 용도별 PC 추천 시스템

개요

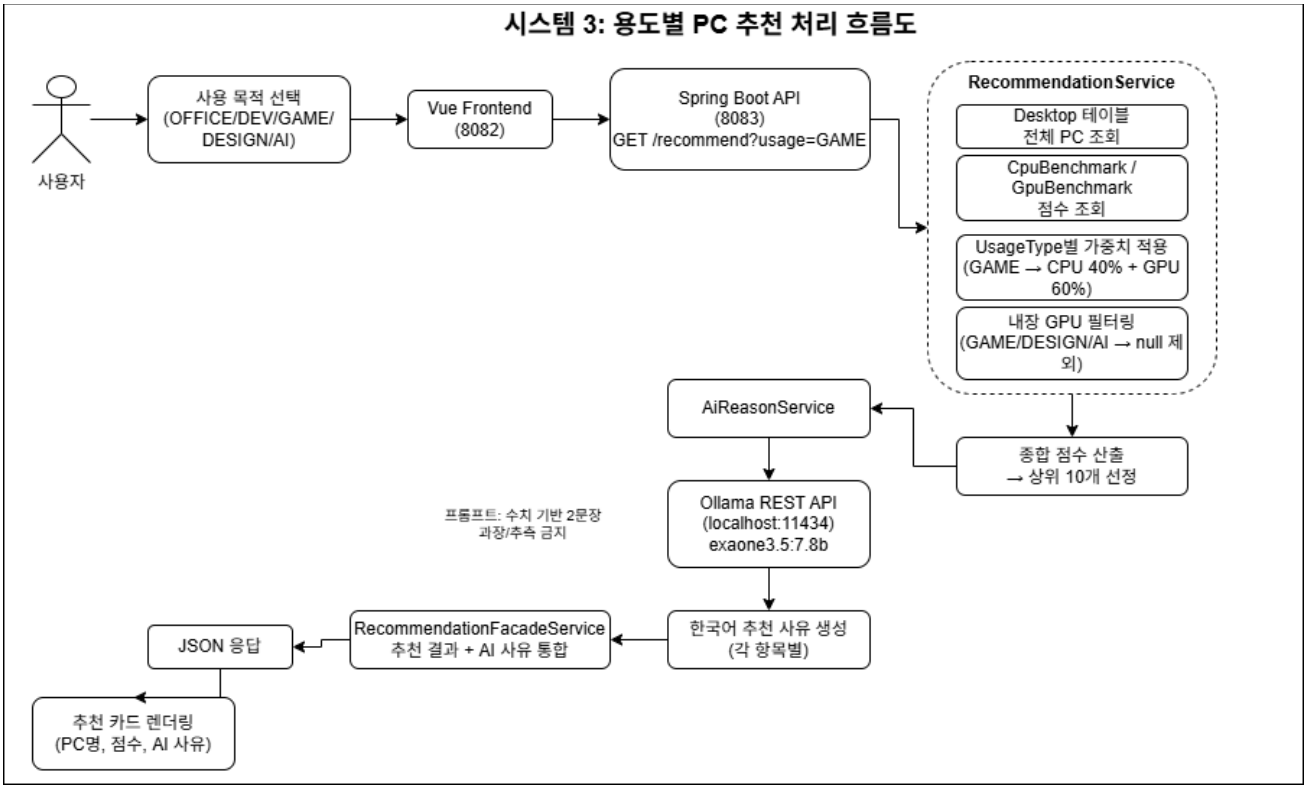
사용 목적(사무/개발/게임/디자인/AI)을 선택하면 Passmark CPU.GPU 벤치마크 점수에 가중치를 적용하여 최적의 PC 10선을 추천하고, Ollama 로컬 LLM이 각 추천 항목에 대해 수치 기반 추천 사유를 자동 생성하는 시스템이다.

주요 기능

기능	상세설명
용도별 가중치 적용	사용 목적에 따라 CPU/GPU 벤치마크 비중을 자동 적용하여 종합 점수 산출
내장 GPU 필터링	게임/디자인/AI 용도는 GPU(gpu_id가 null)인 PC를 제외
상위 10선 추천	종합 점수 시준 상위 10개 데스크탑 PC 선정
AI 추천 사유 생성	Ollama(exaone3.5:7.8b) LLM이 벤치마크 수치 기반으로 한국어 추천 사유 2문장 자동 생성

사용 목적별 CPU/GPU 가중치

사용 목적	CPU 비중	GPU 비중	내장 GPU 허용	설명
OFFICE (사무)	70%	30%	O	문서 작성, 웹 브라우징 등 CPU 중심 작업
DEV (개발)	60%	40%	O	컴파일, 빌드 등 CPU 비중 높은 작업
GAME (게이밍)	40%	60%	X	3D 렌더링, 게임 등 GPU 의존도 높은 작업
DESIGN (디자인)	20%	80%	X	영상 편집, 그래픽 작업 등 GPU 집약 작업
AI (딥러닝)	20%	80%	X	모델 학습, 추론 등 GPU 연산 집약 작업



기술 스택

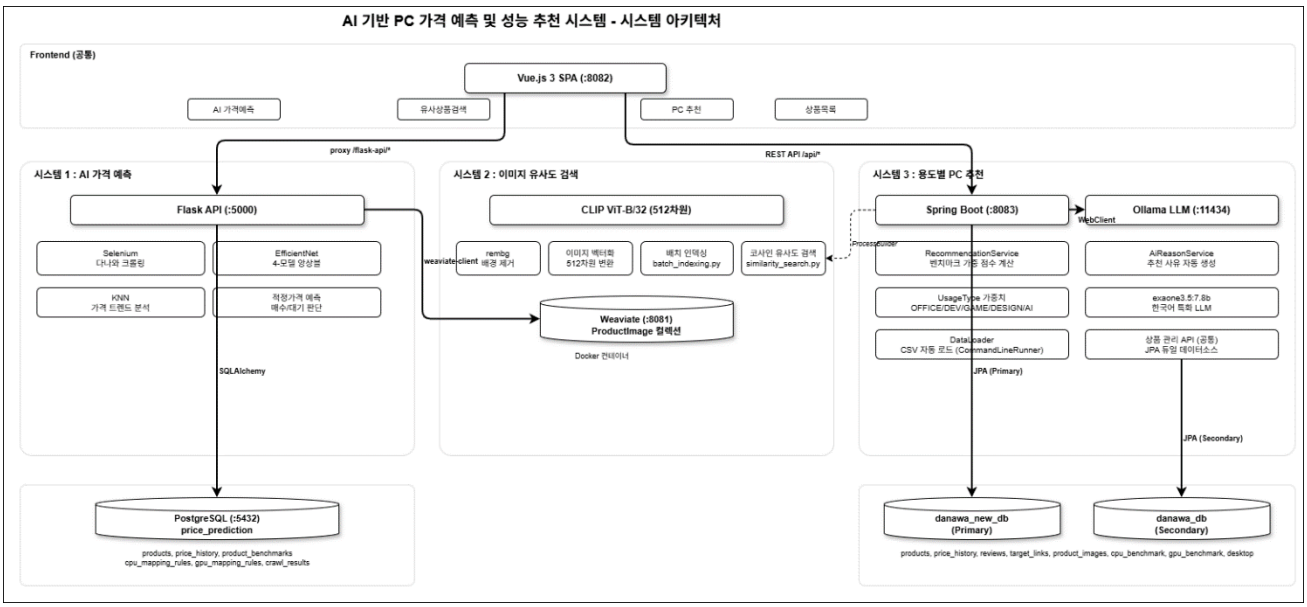
구성요소	기술	역할
백엔드	Spring Boot 3.1.5 (WebFlux)	추천 API 서버 (:8083)
추천 로직	RecommendationService	가중 점수 계산, 상위 10선 선정
AI 사유 생성	Ollama (exaone3.5:7.8b)	한국어 추천 사유 자동 생성 (:11434)
API 호출	Spring WebFlux WebClient	Ollama REST API 비동기 호출
DB	PostgreSQL (danawa_new_db)	벤치마크/데스크탑 데이터
데이터 로드	Data Loader (CommandLineRunner)	앱 시작 시 CSV 자동 로드

DB 테이블 (danawa_new_db)

테이블	주요 컬럼	설명
Cpu_benchmark	cpu_id(PK), score	Passmark CPU 벤치마크 점수 (700건+)
Gpu_benchmark	gpu_id(PK), score	Passmark GPU 벤치마크 점수 (200건+)
Desktop	id, product_code, name, specifications, cpu_id(FK), gpu_id(FK)	데스크탑 PC 정보 (3800건+)

3. 공통 인프라 및 시스템 구조

□ 전체시스템 구조도



□ 서비스 포트 구성

서비스	포트	담당 시스템	데이터 베이스
Vue.js Frontend (Dev)	8082	공통 (UI)	
Spring Boot API	8083	시스템 3 + 상품관리	Danwa_new_db, danawa_db
Flask AI API	5000	시스템 1,2	Price_prediction

Weaviate	8081	시스템 2	내장 스토리지 (Docker Volume)
Ollama LLM	11434	시스템 3	
PostgreSQL	5432	공통	3개 DB 운영

□ 데이터베이스 구성

본프로젝트는 3개의 PostgreSQL 데이터베이스와 1개의 벡터 DB를 운영한다.

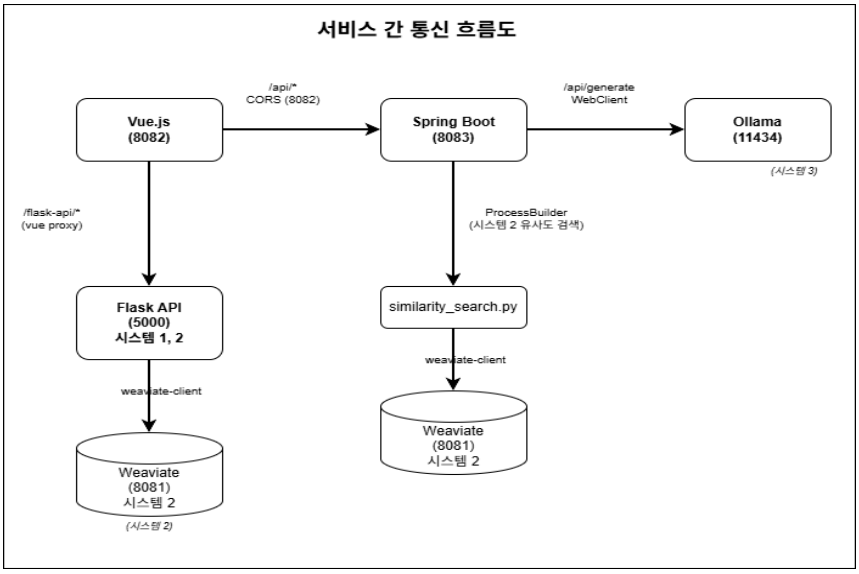
DB	용도	접근 서비스	주요 테이블
Danawa_new_db	신규 상품 데이터 + 벤치마크/데스크톱	Spring Boot (Primary)	Products, price_history, reviews, target_links, product_images, cpu_benchmark, gpu benchmark, desktop
Danawa_db	기존 상품 데이터 (레거시)	Spring Boot (Secondary)	Products, price_history, reviews, target_links
Price_prediction	AI 예측 결과 + 매핑 규칙	FLASK	Products, price_history, product_benchmarks, cput_mapping_rules, gpu_mapping_rules, crawl_results
Weavia	이미지 벡	Flask /	ProductImage (CLIP 512차원)

te	터 저장	Spring Boot	

□ 공통 기능 : 상품 데이터 관리

기능	상세 설명
상품 목록 조회	Spring Boot JPA 기반 상품 목록 조회, 카테고리별 필터링, 페이징 처리
듀얼 데이터소스	Danawa_db(레거시), danawa_new_db(신규)를 EntityManagerFactory 분리로 동시 운영
상품 상세 정보	상품 스펙, 가격 이력, 카테고리별 스펙 필터링 통합 제공
학습 데이터 셋 관리	수집된 10,800+ 상품 데이터 조회, 필터링, 벤치마크 점수 기반 정렬
벤치마크 자동 로드	CommandLineRunner 기반 DataLoader가 앱 시작 시 CSV에서 벤치마크 데이터 자동 로드

□ 서비스 간 통신 흐름도



4. 프로젝트 개발 일정 (WBS)

□ WBS 개요

단계	기간	주요 활동
1. 계획/분석	1주차	목표 정의, 요구분석, 계획서 작성
2. 설계/구현	2주차	시스템 1,2 구현 (AI 예측 + 유사도 검색)
3. 추천 시스템	3주차	시스템 3 구현 (PC 추천 + AI 설명)
4. 통합/배포	4주차	3개 시스템 통합, 테스트, 발표

□ 상세 WBS

작 업 명	담당자	개발 일정			비고
		1 W	2 W	3 W	
1. 계획					
- 목표정의	전체				2 일 소요
- 계획서 작성	전체				
- 요구사항 분석	전체				
- 계획서 작성	전체				
2. 설계					
- 시스템 아키텍처	전체				4일 소요
- ERD 설계	전체				
- 화면 설계	전체				
3. 시스템 1					
- 다나와 크롤링 모듈	서재홍/이주희				7일 소요
- EfficientNet 모델 학습	서재홍				
- Flask AI API 구현	서재홍				

- KNN 트렌드 분석 모듈	서재홍				
-Price_prediction DB 구축	서재홍				
4. 시스템 2					
- CLIP 벡터화 파이프라인	박윤호/하성호				3일 소요
- Weaviate 인덱싱/검색	박윤호/하성호				
- Docker Compose 구성	하성호				
- 이미지 전처리 (rembg)	하성호				
5. 시스템 3					
- 벤치마크 데이터 수집					3 일 소요
- 추천 로직 구현					
- Ollama LLM 연동					
- 추천 UI 개발					
6. 공통 : 상품 관리					
- Spring Boot API	이주희				2 일 소요
- Vue.js Frontend	박윤호				
- PostgreSQL 데이터 소스	서재홍				
7. 통합 및 테스트					
- 3 개 시스템 통합	서재홍				4 일 소요
- 단위/통합 테스트	전체				
- 오류 수정	전체				
8. 배포 및 발표					
- Docker 컨테이너화	하성호				2일 소요
- 최종 발표 자료 작성	서재홍				

5. 조직 구성 및 업무 분장

□ 팀원 구성

이름	역할	담당업무
서재홍	AI 엔지니어/모델학습	시스템 1
박윤호	백엔드/프론트엔드	시스템 2
이주희	AI 엔지니어/백엔드	시스템 3
하성호	백엔드/DB설계	시스템 2

□ 서브 시스템별 담당자

서브시스템	담당 업무	담당자
시스템 1 : AI 가격 예측		
Flask AI API	가격 예측 REST API, 크롤링 연동	서재홍
EfficientNet 학습	4-모델 앙상블 학습, 하이퍼파라미터 튜닝	서재홍
KNN 트렌드	가격 추세 분석 모듈	서재홍
다나와 크롤링	Selenium 기반 상품 데이터 수집	서재홍/이주희
시스템 2 : 유사도 검색		
CLIP 벡터화	이미지 -> 512차원 벡터 변환 파이프 라인	박윤호/하성호
Weaviate 연동	벡터DB 인덱싱, 유사도 검색 구현	박윤호/하성호
이미지 전처리	Rembg 배경 제거 파이프 라인	하성호
시스템 3 : PC 추천		
추천 로직	벤치마크 가중 점수 계산, 추천 컨트롤러	이주희

Ollama LLM 연동	추천 사유 자동 생성, 프롬프트 설계	이주희
벤치마크 데이터	Passmark CPU/GPU 벤치마크 데이터 수집 및 정체	이주희
공통		
Spring Boot API	상품 관리, 듀얼 데이터 소스, JPA	이주희
Vue.js Frontend	전체 UI (상품목록, AI 예측, 유사 검색, PC 추천)	전체
PostgreSQL DB	ERD 설계, 3개 DB 스키마 관리	전체
Docker 배포	Weaviate 컨테이너화, docker-compose 구성	하성호

6. 개발 환경

□ 개발 도구

구분	도구	용도
IDE	Intellij IDEA / VS Code	코드 작성 및 디버깅
버전관리	Git, GitHub	소스 코드 버전 관리
빌드도구 (Java)	Gradle 8.8	Spring Boot 빌드 및 의존성 관리
빌드도구 (.JS)	npm	Vue.js 빌드 및 의존성 관리
빌드도구 (Python)	pip	Python 패키지 관리
API 테스트	curl/브라우저	REST API 테스트
DB 관리	PgAdmin / psql	PostgreSQL 조회/관리
컨테이너	Docker Desktop	Weaviate 컨테이너 실행

GPU 가속	NVIDIA CUDA	EfficientNet/CLIP 모델 추론 가속
LLM 런타임	Ollama	로컬 LLM 모델 실행

□ 기술 스택

구분	기술	버전	비고
Language (Backend)	Java	17	시스템 3, 공통
Language (AI)	Python	3.x	시스템 1,2
Language (Frontend)	JavaScript	ES6+	공통
Framework (Backend)	Spring Boot	3.1.5	시스템 3, 공통
Framework (AI)	Flask	Latest	시스템 1,2
Framework (Frontend)	Vue.js	3.x	공통
ORM	Spring Data JPA/Hibernate	6.2.13	시스템 3, 공통
Database	PostgreSQL	18	3개 DB 운영
Vector DB	Weaviate	1.28.4	시스템 2
LLM	Ollama (exaone3.5:7.8b)	Latest	시스템 3
AI 모델 (가격)	EfficientNet	Pytorch	시스템 1
AI 모델 (이미지)	CLIP	Openai/clip-vit-base-patch32	시스템 2
이미지 전처리	Rembg	Onnxruntime-gpu	시스템 2
크롤링	Selenium	Latest	시스템 1
차트	Chart.js	Latest	공통
Container	Docker	Latest	시스템 2 (Weaviate)

GPU	NVIDIA RTX 4060 Ti	CUDA	시스템 1,2

7. 기대 효과 및 활용 분야

□ 시스템별 기대효과

시스템	기대효과
시스템 1 : AI 가격 예측	텍스트 + 이미지 멀티 모달 분석으로 소비자가 PC 구매 시 적정 가격 판단 가능, KNN 트렌드 분석으로 매수 타이밍 제공
시스템 2 : 유사도 검색	CLIP 벡터 기반 외관 유사 상품 자동 추천으로 비교 구매 지원, 벡터 DB 활용으로 대규모 상품 실시간 검색 가능
시스템 3 : PC 추천	사용 목적별 CPU.GPU 가중치 적용으로 맞춤 PC 추천, LLM이 수치 기반 추천 사유를 자동 생성하여 사용자 이해도 향상

□ 기술적 기대효과

- 멀티모달 AI 활용 : 텍스트와 이미지를 동시에 분석하여 단일 모달 대비 높은 예측 정확도 달성
- 확장 가능한 아키텍처 : 3개 독립 서브시스템 구조로 각 시스템을 개별적으로 확장.수정 가능
- 로컬 AI 운영 : 외부 API 의존 없이 Ollama 로컬 LLM으로 개인정보 보호 및 비용 절감

□ 활용분야

- 전자제품 가격비교 : PC, 노트북, 모니터등 전자제품 적정가 분석

- PC 구매 가이드 : 사용 목적에 맞는 최적 PC사양 추천
- 중고거래 플랫폼 : 중고 PC 거래 시 적정 시세 산출 참고 자료
- 이커머스 : 온라인 쇼핑몰 상품 추천 및 가격 분석 서비스
- 데이터 분석 : 카테고리별 가격 트렌드 분석 및 시장 동향 파악

8. 위험 요소 및 대응 방안

위험 요소	관련 시스템	영향도	대응 방안
AI 모델 예측 정확도 저하	시스템 1	높음	4-모델 앙상블로 단일 모델 대비 안정성 확보, 지속적 데이터 수집 및 재학습
다나와 크롤링 차단	시스템 1	높음	요청 간격 조절, User-Agent 관리, 크롤링 실패 시 캐시 데이터 활용
Ollama LLM 응답 지연	시스템 3	중간	추천 10건 순차 생성으로 지연 가능, 비동기 처리 또는 캐싱으로 대응
Weaviate 벡터DB 장애	시스템 2	중간	Docker Volume 영구 저장소 사용, 인덱싱 배치 스크립트로 재구축 가능
GPU 메모리 부족	시스템 1,2	중간	배치 크기 조절, CPU 풀백 지원, onnxruntime 최적화
벤치마크 데이터 갱신	시스템 3	낮음	앱 시작 시 CSV 자동 로드(DataLoader), 주기적 CSV 업데이트로 최신 유지
포트 충돌	공통	낮음	서비스별 고정 포트 할당, 환경설정 파일로 관리