

Fetpal 프로젝트 WBS (Work Breakdown Structure)

프로젝트 기간: 2025.09.19 ~ 2025.11.20 (약 9주)

핵심 목표: AI 기반 반려동물 케어 플랫폼 구축 + RAG 시스템 + 라이프스타일 채팅 완성



프로젝트 개요

프로젝트 범위

- Phase 1: 프로젝트 기획 및 설계 ✓
- Phase 2: 백엔드 개발 (Supabase) ✓
- Phase 3: AI 서버 개발 (YOLO 모델 훈련 및 API 구축) ✓
- Phase 4: 프론트엔드 연동 및 테스트 ✓
- Phase 5: RAG 시스템 구축 ✓
- Phase 6: 라이프스타일 채팅 리팩토링 ✓
- Phase 7: 배포 및 최종 점검 🚀

주요 목표

- ✓ 3종 YOLO 모델 개발 (Skin, Health, Eyes)
- ✓ FastAPI 기반 AI 서버 구축
- ✓ 프론트엔드 연동

4. RAG 지식 베이스 시스템 구축
 5. 라이프스타일 실시간 채팅 구현
 6. 통합 해시태그 시스템 확장 (4개 영역)
 7. 반응형 디자인 적용 (400px~1280px+)
 8. 발표 자료 및 시연 준비
-



Phase별 작업 계획 및 완료 현황

Phase 1: 프로젝트 기획 및 설계 (2025.09.19 ~ 2025.09.25)

1.1. 요구사항 분석 및 명세화

- **기간:** 2025.09.19 ~ 2025.09.20 (2일)
- **상태:** 완료 (100%)
- **세부 작업:**
 - 주제 선정: 반려동물 AI 건강 케어 플랫폼 기획
 - 주제 확정: Fetalpal - 반려동물 통합 케어 플랫폼 컨셉 수립
 - 문제정의 및 솔루션 도출: 사용자 페인포인트 분석 및 솔루션 정의
 - 핵심 기능 정의 (AI 진단, 커뮤니티, 플래너 등)
 - 페르소나 및 사용자 시나리오 정의
 - 기술 스택 선정 및 타당성 검토

1.2. 프로젝트 산출물 작성

- **기간:** 2025.09.21 ~ 2025.09.25 (5일)
- **상태:** 완료 (100%)

- 세부 작업:

- 기능 요구사항 작성: AI 건강진단 / 플래너 / 커뮤니티 등 핵심 기능 정의
- 비기능 요구사항 작성: 성능 / 보안 / 확장성 요구사항 정의
- 프로젝트 기획서 작성
- 시스템 아키텍처 설계 (MSA 구조 / 기술 스택 선정)
- 시스템 흐름도 작성 (주요 기능별 프로세스 플로우 설계)
- 데이터베이스 ERD 설계 (37개 테이블 설계)
- WBS 작성

1.3. UI/UX 디자인

- 기간: 2025.09.22 ~ 2025.09.25 (4일)
- 상태: 완료 (100%)
- 세부 작업:
 - 와이어프레임 및 프로토타입 제작
 - UI 디자인 시스템 및 컴포넌트 설계
 - Next.js 14 App Router 기반 설계

1.4. 개발 환경 및 일정 계획

- 기간: 2025.09.19 ~ 2025.09.25 (7일)
- 상태: 완료 (100%)
- 세부 작업:
 - Git 레포지토리 및 브랜치 전략 수립
 - 개발, 스테이징, 프로덕션 환경 계획
 - 전체 프로젝트 마일스톤 및 일정 수립
 - Monorepo + Hook Composition 패턴 도입

Phase 2: 백엔드 개발 (Supabase)

(2025.09.19 ~ 2025.09.25)

2.1. Supabase 프로젝트 설정 ✓

- **기간:** 2025.09.19 ~ 2025.09.20 (2일)
- **상태:** 완료 (100%)
- **세부 작업:**
 - 프로젝트 생성 및 기본 설정
 - 데이터베이스 초기화

2.2. 데이터베이스 스키마 구현 ✓

- **기간:** 2025.09.21 ~ 2025.09.25 (5일)
- **상태:** 완료 (100%)
- **세부 작업:**
 - 사용자/인증 관련 테이블 생성 (profiles, profileCompletion)
 - 반려동물 관련 테이블 생성 (palProfiles, palHealthRecords, vaccinations)
 - 커뮤니티 관련 테이블 생성 (communityPosts, postTags, postLikes, events)
 - 플래너 관련 테이블 생성 (plannerEvents, plannerExpenses)
 - 파일 메타데이터 테이블 생성
 - CamelCase 마이그레이션 적용

2.3. 데이터베이스 로직 구현 ✓

- **기간:** 2025.09.22 ~ 2025.09.25 (4일)
- **상태:** 완료 (100%)
- **세부 작업:**
 - PostgreSQL 함수 및 트리거 작성 (프로필 자동 생성, updatedAt 등)

- RLS (Row Level Security) 정책 수립 및 적용 (60개+ 정책)

2.4. 스토리지 설정

- 기간: 2025.09.24 ~ 2025.09.25 (2일)
- 상태: 완료 (100%)
- 세부 작업:
 - profileImage 등 버킷 생성
 - 폴더별 접근 제어 정책 수립 및 적용

Phase 3: AI 서버 개발 (2025.09.19 ~ 2025.10.23)

Week 1: 데이터 파이프라인 구축 (2025.09.19 ~ 2025.09.25)

3.1. 데이터셋 다운로드

- 기간: 2025.09.19 ~ 2025.09.25 (7일)
- 상태: 완료 (100%)
- 담당: LYSS
- 세부 작업:
 - Skin 데이터 수집: 반려동물 피부질환 데이터 **232,253개** 이미지 수집
 - Eyes 데이터 수집: 반려동물 안구질환 데이터 **217,547개** 이미지 수집
 - Health 데이터 수집: 반려동물 건강정보 데이터 **218,747개** 이미지 수집
 - 총 **668,547개** 이미지 (AI-Hub 공공 데이터셋)

Week 2: 데이터 전처리 및 EDA (2025.09.26 ~ 2025.10.02)

3.2. 데이터 파이프라인 구축

- **기간:** 2025.09.26 ~ 2025.09.30 (5일)
- **상태:** 완료 (100%)
- **세부 작업:**
 - JSON → YOLO 포맷 변환 자동화
 - 변환 스크립트 개발: JSON 라벨 → YOLO txt 포맷 변환
 - 중복/오류 제거 자동화: 데이터 품질 관리 시스템 구축
 - 데이터 정제율: **96.8%** (중복/오류 제거)

3.3. 데이터 전처리 및 EDA

- **기간:** 2025.09.26 ~ 2025.10.02 (7일)
- **상태:** 완료 (100%)
- **세부 작업:**
 - **Skin:** 232,253개 이미지 정제 (A7 무증상 3,480개 제거)
 - **Health:** 218,747개 이미지 클래스 매핑
 - **Eyes:** 217,547개 이미지 전처리
 - 클래스 불균형 분석: 데이터 분포 시각화 및 증강 전략 수립
 - **훈련 데이터:** 401,870개 (약 60%)
 - **검증 데이터:** 70,881개 (약 11%)

Week 3-4: YOLO 모델 개발 (2025.10.03 ~ 2025.10.16)



3.4. Health 모델 훈련

- **기간:** 2025.10.03 ~ 2025.10.09 (7일)

- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
- 모델 훈련: YOLOv8m / **28 epochs** / **21.5시간 GPU 학습**
- 성능 평가: **mAP50 88.2%**, mAP50-95 77.1%
- 클래스별 성능:
 - full_body: 98.8%
 - head: 86.8%
 - nose: 78.9%
- 모델 최적화 및 저장: best.pt 모델 저장 (52MB)

3.5. Eyes 모델 훈련

- **기간:** 2025.10.03 ~ 2025.10.09 (7일)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
- 모델 훈련: YOLOv8m / **28 epochs** / **72시간 GPU 학습**
- 성능 평가: **mAP50 25.35%**, mAP50-95 21.48%
- 30개 클래스 성능 분석: 질환별 mAP 측정 및 위험도 분류
- 모델 최적화 및 저장: best.pt 모델 저장 (50MB)

3.6. Skin 모델 훈련

- **기간:** 2025.10.10 ~ 2025.10.16 (7일)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
- 1차 훈련 실패 분석: 하이퍼파라미터 자동 최적화 문제 파악
- 하이퍼파라미터 수동 튜닝: 배치 크기/학습률 등 수동 조정

- 2차 훈련 성공: YOLOv8m / **28 epochs** / **38시간 GPU 학습**
- 성능 평가: **mAP50 18.3%**, mAP50-95 8.3% (1차 대비 2배 향상)
- 모델 최적화 및 저장: best.pt 모델 저장 (52MB)

Week 5: AI 서버 구축 및 API 개발 (2025.10.17 ~ 2025.10.23)

3.7. FastAPI 서버 구축

- **기간:** 2025.10.17 ~ 2025.10.20 (4일)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - 서버 환경 구축: FastAPI / Uvicorn / Python 3.10 설정
 - API 엔드포인트 개발:
 - `/detect-skin` (피부질환)
 - `/detect-health` (건강 체크)
 - `/detect-eyes` (안구질환)
 - `/detect` (자동 모델 선택)
 - `/health` (서버 상태 확인)

3.8. 멀티모델 관리 시스템

- **기간:** 2025.10.20 ~ 2025.10.23 (4일)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - 모델 로딩 최적화: GPU 메모리 효율적 활용 (3.2GB/8GB)
 - Lazy Loading 패턴 적용 (요청 시 모델 동적 로드)
 - 모델 캐싱 시스템으로 재로딩 시간 단축

- 에러 핸들링 구현: 모델 로드 실패 / 타임아웃 등 예외 처리

3.9. LLM(RAG) 연동 준비

- **기간:** 2025.10.17 ~ 2025.10.23 (7일)
- **상태:** 완료 (100%)
- **세부 작업:**
 - LLM Provider (OpenAI/Gemini) API 연동 준비
 - RAG 파이프라인 설계 (검색 → 질문 생성 → 답변)

Phase 4: 프론트엔드 연동 및 테스트 (2025.10.24 ~ 2025.10.31)

Week 6: 프론트엔드 연동 (2025.10.24 ~ 2025.10.31) ✓

4.1. AI 어시스턴트 페이지 개발

- **기간:** 2025.10.24 ~ 2025.10.28 (5일)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - 이미지 업로드 컴포넌트: 파일 선택 / 카메라 촬영 / 드래그앤파울
 - 진단 유형 선택 UI: 피부 / 눈 / 전체 건강 진단 선택
 - 결과 시각화: 바운딩 박스 렌더링 + 신뢰도 + 심각도 표시

4.2. YOLO 모델 통합 및 개선

- **기간:** 2025.10.25 ~ 2025.10.28 (4일)
- **상태:** 완료 (100%)
- **담당:** LYSS

- 세부 작업:
- 2025-10-25: Eyes/Health 모델 감지 이슈 분석 및 해결
 - Health 모델 confidence threshold 0.1 → 0.4 (오감지 방지)
 - 우선순위 기반 모델 선택 로직 구현 (Eyes > Skin > Health)
 - 3개 모델 신뢰도 UI 표시 개선
- 2025-10-26: PetHospital 시스템 완전 구축 (신규!)
 - 카카오 API 통합 (3km 반경 내 동물병원 검색)
 - 24시 응급실 자동 감지 시스템
 - 병원 카드 UI + 연락처 복사 Toast
- 2025-10-28: AI 챗봇 프롬프트 강화 + 해시태그 시스템 개선
 - 감지된 모든 질병명 해시태그 자동 생성
 - 해시태그 분리 포맷팅 (#비듬 #각질 #상피성잔고리)
 - Markdown 렌더러 해시태그 자동 링크 기능
 - 콘솔 로그 90% 축소 (디버깅 최적화)

4.3. API 연동 및 테스트

- 기간: 2025.10.28 ~ 2025.10.31 (4일)
- 상태: 완료 (100%)
- 담당: LYSS
- 세부 작업:
 - CORS 문제 해결: allow_origins 설정 / 크로스 도메인 요청 허용
 - 에러 핸들링: 로딩 상태 / 실패 메시지 / 재시도 로직
 - 페이지 로딩 속도 85% 개선 (3-5초 → 0.1-0.5초)
 - 평균 응답 시간: 3~5초 (이미지 분석 포함)

4.4. 라이프스타일 채팅 시스템 구축 시작

- 기간: 2025.10.24 ~ 2025.10.31 (8일)

- 상태: 완료 (100%)
 - 세부 작업:
 - lifestylechatrooms, lifestylechatmessages 테이블 생성
 - 글로벌 로비 + 해시태그 채팅방 UI 구현
 - 실시간 Supabase Realtime 구독 구현
-

Phase 5: RAG 시스템 구축 및 시스템 개선 (2025.10.31 ~ 2025.11.07)

Week 7: RAG 시스템 구축 및 전체 시스템 개선 (2025.10.31 ~ 2025.11.07)

5.1. RAG 시스템 최적화 및 시연 준비

- 기간: 2025.10.31 (1일)
- 상태: 완료 (100%)
- 담당: LYSS
- 세부 작업:
 - Health 모델 UI 완전 제거 (시연 영상용)
 - 무의미한 해시태그 완전 제거 (#무, #nose 등 필터링)
 - "더 자세히 알아보기" 답변 길이 증가 (maxOutputTokens 4096 → 8192)
 - 커뮤니티 게시글 → 지식 베이스 자동 수집 (트리거 기반)
 - Vector Similarity Search 구현 (코사인 유사도)
 - quality_score + is_verified 품질 관리 체계 구축
 - 출처 추적 시스템 (source: community, faq, youtube, manual)

5.2. Vector DB 구축 (이전 작업 기반)

- **기간:** 2025.10.14 ~ 2025.10.31 (통합 작업)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - **pgvector 확장** PostgreSQL 설치
 - **pet_knowledge_base 테이블** 구축
 - **HuggingFace Embedding API** 연동
 - Embedding 모델: sentence-transformers/all-MiniLM-L6-v2 (384차원)

5.3. LLM 답변 생성 파이프라인

- **기간:** 2025.10.14 ~ 2025.10.31 (통합 작업)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - LLM 답변 생성 파이프라인 구축
 - OpenAI GPT-4 / Google Gemini 연동
 - RAG 검색 함수 개발 (`search_pet_knowledge`)
 - 시연 영상 촬영 및 강사님/매니저님께 전송 완료

5.4. 설정 페이지 알림 시스템 대개편

- **기간:** 2025.11.02 (1일)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - **알림 설정 구조 재정리** (4개 → 명확한 구분)
 - 푸시 알림 (총괄 마스터 토글)

- 플래너 알림 (일정 + 건강 통합)
- 커뮤니티 알림 (댓글, 좋아요)
- 라이프스타일 알림 (채팅방 활동) - 신규 추가!
- DB 스키마 업데이트: lifestylenotifications 퀘리 추가
- 분석 데이터 수집 투명성 강화 (펼쳐보기 기능 추가)
- 프로필 공개 토글 삭제 (불필요한 기능 제거)

5.5. 반응형 디자인 추가

- 기간: 2025.11.06 (1일)
- 상태: 완료 (100%)
- 담당: LYSS
- 세부 작업:
 - 커뮤니티 페이지 모바일 반응형 추가 (768px, 480px)
 - AI Assistant 좌측 패널 모바일 반응형 개선
 - Dashboard 우측 패널 기능 카드 모바일 반응형 강화
 - CSS 우선순위 강화 (!important 패턴)

5.6. 커뮤니티 실시간 댓글 시스템

- 기간: 2025.11.07 (1일)
- 상태: 완료 (100%)
- 담당: LYSS
- 세부 작업:
 - 실시간 댓글 시스템 구축
 - 모달 내 즉시 반영
 - 커뮤니티 페이지 동기화
 - 양방향 동기화 구현

Phase 6: 라이프스타일 채팅 리팩토링

(2025.11.08 ~ 2025.11.10)

Week 8: 라이프스타일 채팅 완성 (2025.11.08 ~ 2025.11.10) 

6.1. 반응형 디자인 완전 구현

- **기간:** 2025.11.09 (1일)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - **Lifestyle CSS 전체 반응형 적용** (400px 모바일 대응)
 - **8개 CSS 파일 반응형 완성:**
 - realtimechat.module.css (메인 채팅 레이아웃)
 - bookmarktabs.module.css (책갈피 탭)
 - hashtagrooms.module.css (해시태그 방 목록)
 - chatroom.module.css (채팅룸 - 기존 반응형 강화)
 - participantsdropdown.module.css (참여자 드롭다운)
 - moremenu.module.css (더보기 메뉴)
 - lifestyle.module.css (페이지 레이아웃)
 - _components/lifestyle.module.css (공통 컴포넌트)
 - **4단계 브레이크포인트 적용:**
 - 1280px: 10-15% 축소
 - 900px: 20-25% 축소
 - 768px: 30-40% 축소
 - 400px: 최대 축소 (왼쪽/오른쪽 잘림 완전 해결)

6.2. UI/UX 개선 및 CSS 리팩토링

- 기간: 2025.11.10 (1일)
- 상태: 완료 (100%)
- 담당: LYSS
- 세부 작업:
- 좌우 패널 높이 완벽 매칭: `align-items: stretch` + `min-height: 0`
- Sticky 기능 복원: `position: sticky, top: 0`
- 툴팁 가시성 완전 해결: `z-index: 10000` + `overflow: visible`
- Z-Index 레이어링 시스템 확립: 0 → 10 → 100 → 199999 → 200000
- Flexbox 높이 통제 마법 공식: flex container + `align-items: stretch`
- Flex 스크롤 작동: `min-height: 0` 패턴 적용
- MessagesContainer 스크롤 완벽 구현

6.3. 해시태그 시스템 확장 (이전 작업 완료)

- 기간: 2025.10.08 (완료됨)
- 상태: 완료 (100%)
- 담당: LYSS
- 세부 작업:
- `lifestylechatmessagehashtags` 테이블 추가
- DB 트리거 기반 자동 추출 (`#태그` 정규식 파싱)
- 통합 해시태그 시스템 구현 (4개 영역):
 - communityPostHashtags
 - plannerEventHashtags
 - plannerExpenseHashtags
 - `lifestylechatmessagehashtags` ★ (신규)
- 인기 해시태그 조회 함수 (`get_popular_hashtags_in_room`)
- 실시간 usageCount 카운팅

6.4. 회원탈퇴 로직 개선 (이전 작업 완료)

- **기간:** 2025.11.02 ~ 2025.11.03 (완료됨)
 - **상태:** 완료 (100%)
 - **담당:** LYSS
 - **세부 작업:**
 - 라이프스타일 채팅 데이터 삭제 로직 추가
 - RAG 지식 베이스 삭제 로직 추가
 - 관리자용 삭제 함수 생성 (`admin_delete_user_completely`)
 - 테이블명 대소문자 정확히 구분 (CamelCase vs 소문자)
-

Phase 7: 배포 및 최종 점검 (2025.11.03 ~ 2025.11.20)

Week 8-9: 배포 및 발표 준비 (2025.11.03 ~ 2025.11.20)



7.1. Vercel 배포 및 모바일 반응형 개선

- **기간:** 2025.11.03 ~ 2025.11.20 (18일)
- **실제 작업:** 2025.11.06 (1일 완료)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - **프론트엔드 Vercel 배포 성공** (<https://fetpal.vercel.app>)
 - **모바일 반응형 대규모 개선** (9개 버그 수정):
 - 커뮤니티 페이지 모바일 반응형 추가

- AI Assistant 좌측 패널 이미지 영역 개선
 - Dashboard 우측 패널 카드 모바일 강화
 - Login/Signup 백그라운드 영상 표시 수정
 - usePathname SSR 에러 해결 (가장 까다로웠음)
 - Vercel 빌드 설정 추가 (모노레포 구조)
 - Vercel 무료 플랜 크론 제한 해결
 - 백그라운드 비디오 CSS 우선순위 강화
- QR 코드 생성 및 발표 준비
 - 도메인 및 HTTPS 설정 완료
 - 발표 후 모든 배포 언디플로이 (학원 요청)

7.2. 모델 성능 평가 (이전 작업 완료)

- **기간:** 2025.10.16 ~ 2025.10.23 (완료됨)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - 모델정의서 작성: 3개 모델 스펙 / 클래스 정의 / 기술 스택
 - 성능평가 결과서 작성: mAP 분석 / 실전 테스트 / 개선 방안
 - 검증 데이터 기반 성능 측정

7.3. 프로젝트 문서 최신화

- **기간:** 2025.11.04 ~ 2025.11.14 (11일)
- **상태:** 완료 (100%)
- **담당:** LYSS
- **세부 작업:**
 - **WBS 최신화:** 주차별 작업 현황 표 작성 / 데일리 리포트 기반 날짜 조정
 - ERD / 시스템 흐름도 최신화: RAG / YOLO 시스템 반영
 - 시스템 아키텍처 최신화: MSA 구조 / 기술 스택 업데이트

- 요구사항 정의서 최신화
- 성과 및 개선사항 문서 작성
- **데일리 리포트:** 2025-10-25 ~ 2025-11-20 (총 8개 작성, 마지막: 11-10)
 - 2025-10-25: Eyes/Health 모델 이슈 해결
 - 2025-10-26: PetHospital 구축 완료
 - 2025-10-28: AI 챗봇 프롬프트 강화
 - 2025-10-31: RAG 시스템 최적화 (시연 영상용)
 - 2025-11-02: 설정 페이지 알림 시스템 개편
 - 2025-11-06: Vercel 배포 및 반응형 개선
 - 2025-11-09: Lifestyle 전체 반응형 완성
 - 2025-11-10: Lifestyle CSS 완전 리팩토링

7.4. 발표 자료 제작

- **기간:** 2025.11.08 ~ 2025.11.20 (13일)
 - **상태:** 진행중 (60%)
 - **담당:** LYSS
 - **세부 작업:**
 - Canva PPT 제작: 15-20분 발표 슬라이드 (15장 내외) [진행중 60%]
 - **시연 영상 촬영 완료:** 실제 사용 시나리오 기반 데모 영상 [완료 100%]
 - 2025-10-31: 강사님/매니저님께 시연 영상 전송 완료
 - **발표 리허설:** Q&A 시나리오 준비 / 타이밍 조율 [예정 0%]
-



4차 프로젝트 핵심 성과

실제 작업 기간 요약 (데일리 리포트 기준) ★

Phase	WBS 기간	실제 작업 일	주요 작업
Phase 4 (프론트 연동)	10.24-10.30 (7일)	10.25- 10.31 (7일)	YOLO 통합, PetHospital 구축, 해시태그 시스템
Phase 5 (RAG 구축)	10.31-11.06 (7일)	10.31- 11.07 (8일)	RAG 최적화, 설정 페이지, 반응형 디자인
Phase 6 (Lifestyle)	11.07-11.14 (8일)	11.09- 11.10 (2일)	전체 반응형 적용, CSS 리팩토링
Phase 7 (배포)	11.03-11.20 (18일)	11.06 (1일 배포)	Vercel 배포 및 모바일 최적화

주요 발견사항:

- Phase 6가 실제로는 2일만에 완성 (11.09-11.10)
- Phase 7 배포는 11.06 하루만에 완료 (반응형 개선 포함)
- Phase 4-5에 숨겨진 핵심 작업들 많음 (PetHospital, 설정 페이지 등)

데이터 수집 및 전처리

항목	수치	비고
총 수집 이미지	668,547개	AI-Hub 공공 데이터셋
훈련 데이터	401,870개	약 60% 비율
검증 데이터	70,881개	약 11% 비율
총 클래스 수	39개	Skin(6) + Health(3) + Eyes(30)
데이터 정제율	96.8%	중복/오류 제거

YOLO 모델 성능

모델명	용도	클래스 수	훈련 시간	mAP50	mAP50-95
Health	전신 건강 체크	3개	21.5시간	88.2% ⭐	77.1%
Eyes	안구질환 감지	30개	72.0시간	25.35%	21.48%
Skin	피부질환 감지	6개	38.0시간	18.3%	8.3%
합계	-	39개	131.5시간	-	-

AI 서버 API

항목	내용
프레임워크	FastAPI (Python 3.10)
엔드포인트 수	5개 (detect, detect-skin, detect-health, detect-eyes, health)
평균 응답 시간	3~5초 (이미지 분석 포함)
GPU 사용	NVIDIA GeForce RTX 4060 (8GB)
동시 처리	멀티모델 자동 전환 지원

RAG 시스템 ★★★

항목	내용
Vector DB	PostgreSQL + pgvector 확장
Embedding 모델	sentence-transformers/all-MiniLM-L6-v2 (384차원)
Embedding API	HuggingFace Inference API
검색 방식	코사인 유사도 기반 Vector Similarity Search
지식 수집	커뮤니티 게시글 자동 수집 (DB 트리거)
LLM	OpenAI GPT-4 / Google Gemini
품질 관리	quality_score (0.00-1.00) + is_verified
출처 추적	source (community, faq, youtube, manual)

라이프스타일 채팅 시스템 ★★☆

항목	내용
채팅방 유형	2가지 (global: 글로벌 로비, hashtag: 주제별 채팅)
실시간 기능	Supabase Realtime (WebSocket)
참여자 추적	접속 상태, 마지막 활동 시간
이미지 공유	Supabase Storage 연동 (lifestyle-chat-images 버킷)
해시태그 추출	DB 트리거 자동 추출 (#태그 정규식)
인기 태그	실시간 Top 5 업데이트 (사용 횟수 + 최신순)
반응형 디자인	4개 breakpoint (1280px, 900px, 768px, 400px)

통합 해시태그 시스템 확장 ★

항목	내용
연결 영역	4개 (커뮤니티, 플래너, 라이프스타일 채팅, 지출)
중앙 테이블	globalHashtags
연결 테이블	communityPostHashtags, plannerEventHashtags, plannerExpenseHashtags, lifestylechatmessagehashtags ★
자동 추출	DB 트리거 기반 정규식 파싱
통계 추적	주간/월간 사용 통계, 인기도 계산 (usageCount)

커뮤니티 시스템 개선 ★★

항목	내용
실시간 댓글	모달 내 즉시 반영 + 페이지 동기화 (2025-11-07)
좋아요 카운팅	트리거 기반 자동 집계 (성능 최적화)
작성자 프로필	사용자/반려동물 구분 표시 (authorType)
해시태그 통합	커뮤니티 + 플래너 + 라이프스타일 연동

반응형 디자인 적용 ★★☆

페이지	브레이크포인트	작업 내용	상태
Lifestyle	400px, 768px, 900px, 1280px	4단계 반응형 적용 (8개 CSS 파일)	<input checked="" type="checkbox"/> 완 료
Community	768px, 480px	모바일 반응형 추가	<input checked="" type="checkbox"/> 완 료
AI Assistant	768px, 480px	좌측 패널 모바일 개선	<input checked="" type="checkbox"/> 완 료
Dashboard	768px, 480px	우측 패널 카드 모바일 강화	<input checked="" type="checkbox"/> 완 료

Hook Composition 리팩토링 성과 ★★

파일명	Before	After	감소율
FeedDetailModal	528줄	231줄	56%
useCommunityPosts	386줄	128줄	67%
EventListSection	303줄	117줄	61%
useRealtimeComments	310줄	238줄	23%
FeedCard	561줄	160줄	71%
usePlanner	676줄	165줄	75%

데이터베이스

측정 항목	수치	비고
총 테이블 수	40개 +	라이프스타일 채팅 4개 + RAG 1개 추가
총 SQL 파일	74개	실행 중 66개 + 아카이브 8개
총 함수	30개 +	RAG 검색 함수 + 라이프스타일 해시태그 함수 포함
총 트리거	22개 +	채팅 해시태그 트리거 + RAG 트리거 추가
총 RLS 정책	60개 +	라이프스타일 채팅 + RAG 정책 포함



4차 프로젝트 주요 도전과제 및 해결

1 Skin 모델 성능 저하 문제

문제

- 1차 훈련: mAP50 9.0% (목표 대비 50% 미달)
- 하이퍼파라미터 자동 최적화가 오히려 학습 불안정 초래

해결

- 하이퍼파라미터 수동 튜닝 (lr, momentum, weight_decay 조정)
- 28 에포크 재훈련 → mAP50 18.3% 달성 (2배 향상)

2 멀티모델 관리 복잡도

문제

- 3종 모델 동시 로딩 시 메모리 부족 (8GB VRAM)

해결

- Lazy Loading 패턴 적용 (요청 시 모델 동적 로드)
- 모델 캐싱 시스템으로 재로딩 시간 단축

3 RAG 시스템 구축

문제

- 커뮤니티 경험담을 AI 답변에 활용하고 싶지만 LLM 환각 현상 발생
- 검증된 지식만 제공해야 하는 수의학적 책임

해결

- pgvector Extension + HuggingFace 임베딩으로 벡터 검색 구현
- 커뮤니티 게시글 작성 시 자동 임베딩 트리거 생성
- quality_score + is_verified로 품질 관리 체계 구축

4 라이프스타일 채팅 UI/UX 문제

문제

- 좌우 패널 높이 불일치
- Sticky 기능 작동 불량
- 툴팁이 다른 요소에 가려짐
- 804x824 화면에서 BookmarkTabs 툴팁이 보이지 않음

해결

- 좌우 패널 `min-height: 80vh` 통일
- Sticky 속성 재적용 (`position: sticky, top: 100px`)
- 툴팁 `z-index: 10000` 설정 + `overflow: visible` 처리
- **Flexbox 높이 통제:** `align-items: stretch` + `min-height: 0`
- **Sticky + Tooltip 공존:** `overflow: visible` + Z-Index 레이어링 ($0 \rightarrow 10 \rightarrow 100$
 $\rightarrow 199999 \rightarrow 200000$)

5 반응형 디자인 누락

문제

- ParticipantsDropdown, MoreMenu 컴포넌트에 1280px, 900px, 480px breakpoint 누락
- lifestyle.module.css에 일부 breakpoint 누락
- 400px 모바일에서 오른쪽 잘림

해결

- 8개 CSS 파일에 4개 breakpoint 완전 추가
- 점진적 크기 축소 (1280px: 10-15%, 900px: 20-25%, 768px: 30-40%, 400px: 최대 축소)
- 일관된 패턴 적용

6 라이프스타일 해시태그 시스템 누락

문제

- 커뮤니티, 플래너는 해시태그 시스템이 있지만 라이프스타일 채팅에는 없음

해결

- `lifestylechatmessagehashtags` 테이블 신규 추가
- DB 트리거 작성: 채팅 메시지 작성 시 자동 해시태그 추출
- `globalHashtags` 테이블과 자동 연동
- 인기 해시태그 조회 함수 (`get_popular_hashtags_in_room`)

7 회원탈퇴 로직 개선

문제

- 라이프스타일 채팅 데이터 삭제 누락
- RAG 지식 베이스 삭제 누락
- Supabase Dashboard에서 삭제 시 Foreign Key 에러

해결

- `useAppActions.js`에 채팅 데이터 삭제 로직 추가
- RAG 지식 베이스 삭제 로직 추가
- 관리자용 삭제 함수 생성 (`admin_delete_user_completely`)
- 테이블명 대소문자 정확히 구분 (CamelCase vs 소문자)

8 실시간 댓글 시스템

문제

- 댓글 작성 후 모달에서만 반영되고 커뮤니티 페이지는 새로고침 필요
- 좋아요 카운트가 실시간으로 업데이트 안 됨

해결

- Supabase Realtime 구독으로 postcomments 테이블 실시간 감지
- 트리거 기반 likesCount 자동 집계 (성능 최적화)
- 모달 ↔ 피드 카드 양방향 동기화 구현

4차 프로젝트 산출물

기술 문서

1. YOLO 모델 결과 보고서 (3종)
 - Skin_YOLO_RESULT.md
 - health_YOLO_RESULT.md
 - eyes_YOLO_RESULT.md
2. YOLO 시스템 가이드: yolo_README.md
3. API 문서: FastAPI Swagger UI
4. 모델정의서: 하이퍼파라미터 상세 설명 (3종)
5. 성능 평가 결과서: 클래스별 성능 분석
6. RAG 시스템 기획서: Vector Search 구축 가이드
7. 라이프스타일 채팅 리팩토링 문서: UI/UX 개선 사항

발표 자료

1. 성능 지표 그래프 (7종)
 - 학습곡선, F1점수, 정밀도, 재현율, PR 곡선, 혼동행렬
2. 훈련 데이터 샘플 (3장)

3. 검증 결과 비교 (정답 vs 예측)
 4. 데이터 분포도
 5. ERD 다이어그램 (40개+ 테이블)
 6. 시스템 아키텍처 다이어그램
 7. 시스템 흐름도
 8. 최종 발표 자료 (251114 작성 중 60%)
 9. 시연 영상 (예정 0%)
-



주요 기술 성과

1. Hook Composition 아키텍처 (2025-09-25)

- 4개 전문 Hook: usePetAPI, usePetFormState, usePetValidation, usePetRegistration
- 독립 시스템: 복잡한 백신 시스템 의존성 우회
- UI 보존: 기존 디자인 100% 유지하면서 최신 Hook 패턴 적용

2. 백신 시스템 구축 (2025-09)

- 6개 백신 타입 지원 (DHPPL, 켄넬코프, 광견병 등)
- 생년월일 기반 자동 일정 생성
- 플래너-백신 실시간 양방향 동기화
- 긴급도별 색상 분류 (지연/긴급/곧예정/정상)

3. 글로벌 해시태그 시스템 (2025-09~11)

- 4개 영역 자동 #{} 추출 (커뮤니티, 플래너, 라이프스타일 채팅, 지출)
- 실시간 usageCount 카운팅
- 크로스 플랫폼 연동 (커뮤니티 ↔ 플래너 ↔ 라이프스타일)

4. RAG 시스템 구축 (2025-10-14~11-06) ★★★

- **pgvector 통합**: PostgreSQL Vector 검색 기능 추가
- **HuggingFace 임베딩**: sentence-transformers/all-MiniLM-L6-v2
- **커뮤니티 자동 수집**: 게시글 작성 시 자동 지식 베이스 추가
- **Vector Similarity Search**: 코사인 유사도 기반 검색

5. 라이프스타일 실시간 채팅 (2025-10-07~11-10)



- 글로벌 로비 + 해시태그 채팅방
- 자동 해시태그 추출: 트리거 기반 #태그 자동 감지
- 실시간 인기 태그: Top 5 업데이트
- 반응형 디자인: 4단계 브레이크포인트

6. 반응형 디자인 (2025-11-06~11-10) ★★★

- 4단계 브레이크포인트: 400px, 768px, 900px, 1280px
- Flexbox 높이 통제 패턴: align-items: stretch + min-height: 0
- Sticky + Tooltip 공존 패턴: overflow: visible + Z-Index 레이어링
- 8개 CSS 파일 반응형 적용

🎓 기술적 학습 성과

배운 핵심 개념

1. Flexbox 높이 통제의 마법 공식

- align-items: stretch + min-height: 0 조합으로 자식 높이 동일화
- flex item의 최소 크기 동작 원리 깊은 이해

2. Sticky + Tooltip 공존 패턴

- `overflow: visible` + 높은 z-index로 두 기능 동시 지원
- 레이어링 시스템 설계 ($0 \rightarrow 10 \rightarrow 100 \rightarrow 199999 \rightarrow 200000$)

3. Flex 스크롤 작동

- `min-height: 0`의 숨겨진 의미 이해
- flex container/item 관계 명확히 파악

4. 4단계 반응형 시스템

- 체계적 축소 전략 설계 ($10\% \rightarrow 20\% \rightarrow 30\% \rightarrow 50\%$)
- 모든 요소에 일관된 패턴 적용

5. RAG (Retrieval-Augmented Generation)

- Vector DB 개념 및 코사인 유사도 검색
- LLM 환각 현상 최소화 기법



향후 개선 방향

모델 성능 개선

- [] Skin 모델 추가 훈련 (50+ epochs)
- [] 데이터 증강 기법 확대 (Rotation, Flip, Color Jitter)
- [] 양상블 모델 적용 (YOLOv8 + YOLOv11)

서비스 확장

- [] AI 서버 클라우드 배포 (AWS EC2 / Railway)
- [] LLM(RAG) 답변 품질 향상
- [] 라이프스타일 채팅 기능 확장 (음성 메시지, 화상 통화)

사용자 경험 개선

- [] PWA 모바일 최적화

- [] 결과 PDF 리포트 다운로드
 - [] 과거 분석 기록 조회 기능
 - [] 실시간 알림 시스템 강화
-



최종 평가

목표 달성을

목표	계획	실적	달성을
YOLO 모델 개발	3종	3종 완성	✓ 100%
AI 서버 구축	FastAPI	로컬 완성	✓ 100%
프론트엔드 연동	1개 페이지	완전 연동	✓ 100%
RAG 시스템 구축	pgvector	완성	✓ 100%
라이프스타일 채팅	기본 기능	완전 리팩토링	✓ 120%
통합 해시태그	3개 영역	4개 영역	✓ 133%
반응형 디자인	기본	4단계 완전 적용	✓ 133%
성능 평가 문서	3종	7종 완성	✓ 233%
발표 자료	PPT + 영상	준비중	🚧 60%

핵심 성과

- ✓ 668,547개 이미지 대용량 데이터 처리 경험
- ✓ 131시간 GPU 훈련 완주 (3종 모델)
- ✓ Health 모델 88.2% 높은 정확도 달성

- **멀티모델 API** 실시간 추론 시스템 구축
- **RAG 시스템 완성** pgvector + HuggingFace + LLM
- **라이프스타일 채팅 완전 리팩토링** UI/UX 대폭 개선
- **통합 해시태그 시스템 확장** 4개 영역 연결
- **반응형 디자인 완전 적용** 4단계 브레이크포인트
- **완전한 문서화 기술** 문서 15개+ 작성
- **Vercel 배포** (<https://fetpal.vercel.app>)

프로젝트 완성도

전체 완성도: 94%  (프로덕션 준비 완료)

- AI 건강진단: **100%** (3개 모델 완성)
- RAG 시스템: **100%** (지식 베이스 구축)
- 라이프스타일 채팅: **100%** (완전 리팩토링)
- 커뮤니티: **100%** (실시간 댓글 완성)
- 플래너: **100%** (백신 자동화 완성)
- 프론트엔드 배포: **100%** (Vercel)
- AI 서버 배포: **80%** (로컬만)
- 발표 준비: **60%** (PPT 작성 중)

작성일: 2025-11-14

작성자: LYSS with Claude

문서 버전: v2.0 (4차 스프린트 최종 통합)

이전 문서: [01_프로젝트_기획서.md](#)

다음 문서: [03_시스템_흐름도.md](#)