

공통 프로젝트 회고 – AI Contact (커플 기반 AI 감성 서비스)

1. 프로젝트 한 줄 요약

연인을 위한 AI 감성 소통 서비스를 개발했다.

AI 아기 아바타, 커플 다이어리·네컷 만화 자동 생성, WebRTC 기반 화상 통화 기능을 구현하며 기술적 완성도와 사용자 감성 경험을 동시에 달성했다.

2. 프로젝트 개요

- 프로젝트명: AI Contact (아이컨택)
- 기간: 2025.06 ~ 2025.08 (약 8주)
- 팀 구성: 3명 (프론트엔드 3, 백엔드 3)
- 사용 기술:
 - FE: React (Vite + TypeScript), Swiper, FullCalendar
 - BE: Spring Boot 3.x (Java 21), JWT 인증, JPA, H2/MySQL
 - Infra: AWS S3(파일 저장), Vercel(프론트 배포), 자체 서버(백엔드 배포)
- 목표: 연인을 위한 감성 AI 서비스 구축

3. 주요 성과와 하이라이트

- 🧠 AI 아기 아바타 구현 → 클릭 시 표정 변화, Apple Memoji 스타일 UI 적용
- 📅 커플 다이어리 및 네컷 만화 자동 생성(GMS 기반) 기능 완성 → 추억 보관소 제공
- 📺 LiveKit(WebRTC) 기반 화상 통화 → 커플 간 원거리 소통 강화
- 🔗 커플 연결/해제 기능 + JWT 보안 → 안정적인 사용자 인증 및 관계 관리
- 💾 S3 기반 파일 저장소 구축 → 이미지·편지 안전 보관

4. 도전 과제와 해결 과정

- 도전 과제 1: RoomId 관리의 복잡성

- **문제 상황:** 초기 설계에서는 채팅을 위해 roomId를 별도로 발급하고, 커플이 채팅을 시작할 때마다 roomId를 조회해야 했다. 하지만 서비스 특성상 **커플은 한 쌍의 고유 대화방만 필요**했기 때문에, 매번 roomId를 관리하는 구조가 불필요하게 복잡했다.
 - **원인 분석:** DB 설계 단계에서 범용적인 채팅 서비스 구조를 따라간 것이 문제였다. 커플 단위 특화 서비스임에도 roomId라는 중간 키를 억지로 두면서 API 요청 흐름이 늘어나고, 버그 가능성도 커졌다.
 - **해결 방법:**
 - 채팅방 식별자를 roomId 대신 **coupleId**로 직접 매핑
 - WebSocket 구독 경로를 /topic/chat/{coupleId} 형태로 단순화
 - 백엔드 저장 로직에서도 ChatMessage 엔티티가 **roomId → coupleId**를 직접 참조하도록 구조 변경
 - **결과:** 채팅 연결 과정이 단순해져 **API 호출 수가 1회 줄었고**, 프론트엔드 로직도 직관적으로 개선되었다.
-

5. Lessons Learned

- JPA Lazy Loading 관리와 엔티티 매핑 최적화
 - 장애 상황을 빠르게 재현·분석하는 **디버깅·트러블슈팅 역량 강화**
-

6. 다음에 시도하고 싶은 것

- 📱 **모바일 PWA 출시:** 모바일 사용자 확대 및 접근성 강화
-

7. 마무리 한 줄

이번 프로젝트는 **AI와 실시간 통신 기술을 결합해 감성 중심 서비스를 구현한 경험**이었다. 앞으로는 **확장성 있는 아키텍처와 더 깊은 사용자 경험 설계**를 목표로 성장하겠다.
