AIO 1기 · FINAL PROJECT · WEEK 1

최종 프로젝트 킥오프

팀을 만들고, 주제를 고르고, 함께 일하는 방식을 정하는 한 주

11/10 (화) – 11/13 (금)

STEP 1 · 프로젝트 기획 및 설계

THIS WEEK

이번 주 흐름 — 4일 안에 기획을 끝냅니다

11/10 화 · 오늘

팀 만들기

- 팀 빌딩 · 역할 탐색
- 팀장 선정
- 팀 헌장 · 그라운드 룰
- 협업 환경 세팅

11/11 수

주제 정하기

- 5개 주제 검토
- 선택 기준으로 점수화
- 문제 정의 · 타깃 사용자
- 핵심 시나리오

11/12 목

설계하기

- 요구사항 명세
- 아키텍처 · 기술 스택
- 에이전트 구성 초안
- WBS 초안

11/13 금 · 마감

제출하기

- 기획서 (요구사항 명세 포함)
- WBS (화면 캡처)
- 대시보드 제출 상태 갱신

THE BIG PICTURE · 9 WEEKS

이번 주 기획이 9주의 설계도입니다

STEP 1 · 1주차
기획 · 설계
지금 여기

STEP 2 · 2주차
API · Function Calling
FastAPI · DB · 인증

STEP 3 · 3–4주차
단일 에이전트
LangGraph · 메모리 · 자기 성찰

STEP 4 · 5–6주차
멀티 에이전트
협업 · Auto-Healing · 보안

STEP 5 · 7–8주차
대화형 앱 · 배포
스트리밍 UI · AWS

12/8 (화) 중간 발표

1/4 (월) 최종 발표 · 소스코드 · 시연영상

주제와 요구사항이 흔들리면 이후 산출물이 전부 흔들립니다. 이번 주에 범위를 현실적으로 잡는 것이 가장 중요합니다.

TEAM BUILDING · 권장 역할

팀 역할 나누기

PM · 기획

일정 · 산출물 · 대시보드 관리, 기획서 총괄

전 단계

백엔드

FastAPI · Supabase · Redis · JWT 인증

STEP 2 중심

AI · 에이전트

LangGraph · 프롬프트 · RAG · 메모리

STEP 3–4 중심

프론트 · UX

Streamlit · 스트리밍 UI · 사고 과정 대시보드

STEP 2 · 5

인프라 · DevOps

GitHub · Docker · AWS · CI/CD · 모니터링

STEP 4–5 중심

QA · 보안 · 문서

테스트 결과서 · 보안 가드레일 · 기술 문서

STEP 3–5

한 사람이 여러 역할을 겸하고, 역할마다 주 담당 + 보조 담당을 두세요. 모두가 코드에 기여해야 합니다 — 최종 발표에서 개인별 기여도를 평가합니다.

TEAM LEADER

팀장은 신중하게 정합니다

팀장이 하는 일

- 운영 매니저와의 주 소통 창구 — 일정, 산출물 제출, 지원 요청
- 멘토와 멘토링 일정 협의 → 디스코드 팀 채널에 공유
- 팀 일정과 대시보드 현황 챙기기

정하는 방법

- 가위바위보 금지 — 책임감과 의지가 있는 사람
- 정해지면 대시보드 '팀원 및 역할'에 '팀장' 표시
- 최종 1등 상금 100만 원은 팀장에게 우선 지급

팀장 = 모든 일을 혼자 하는 사람이 아니라, 일이 굴러가게 만드는 사람

TEAM CHARTER

팀 헌장 — 한 장으로 합의하기

01
팀명 · 미션
우리가 해결할 문제를 한 문장으로

02
목표
1/4에 시연할 모습과 개인별 성장 목표

03
역할과 책임
팀장 · 주 담당 · 보조 담당

04
그라운드 룰
소통 · 행동 · 약속의 최소 지침

05
소통 방식
채널 · 응답 시간 · 공유 위치

06
회의 리듬
데일리 스탠드업 · 주간 회고

07
의사결정
합의 → 안 되면 투표 → 팀장 결정

08
갈등 · 이탈 대응
먼저 팀 안에서, 안 되면 매니저에게

GitHub README 또는 팀 드라이브에 저장하고, 전원이 동의 표시를 남깁니다. 프로젝트 중간에 회고하며 고쳐도 됩니다.

GROUND RULES · 예시

그라운드 룰 — 거창한 목표 말고 실천할 약속

어떻게 소통할까

- 팀 대화는 디스코드에서
- 수업 시간 중 메시지는 2시간 안에 응답
- 30분 넘게 막히면 바로 공유
- 피드백은 사람이 아니라 코드·문서에

어떻게 행동할까

- 매일 시작 10분 스탠드업
- main 브랜치 직접 push 금지
- PR은 1명 이상 리뷰 후 머지
- 회의 결과는 그날 문서로 남김

무엇을 지킬까

- 마감 하루 전 팀 내부 초안 공유
- 지각·불참은 미리 공지
- 일일 진행 기록은 당일 작성
- 맡은 일을 못 하면 빨리 말하기

위 내용은 예시입니다. 팀이 직접 토론해서 5~10개로 정하고, 지키지 못했을 때 어떻게 할지도 함께 정하세요.

SETUP · 오늘 끝내기

협업 환경 세팅

디스코드 팀 채널

주 소통과 자료 공유 — 매니저도 함께 봅니다. 오픈채팅·개인 메신저는 최소화, 이력서 등 개인정보는 멘토와 1:1로.

GitHub

팀 저장소 · Projects 보드(WBS) · 브랜치 전략(main / develop / feature) · README에 팀 헌장

대시보드 팀 시트

팀 정보 · 팀원 및 역할 · 일일 진행 기록 · 산출물 제출 상태 — 막히면 '0팀(작성예시)' 참고

팀 드라이브

기획서 · 발표 자료 공동 작성, 회의록 보관

오늘 입력: 통합 대시보드 팀 구성 표 — 주제(미정이면 비워 두기) · 기술 스택 · 팀장 · 팀원 · GitHub 주소

TOPICS · 제시 주제 중 택1

5개 프로젝트 주제

#	주제	도메인	핵심 사용자
1	멀티 에이전트 기반 장애 예측·자가 진단·자율 복구 시스템	인프라 관리	운영자 · SRE
2	지능형 업무 협업 및 자율 수행 멀티 에이전트	통합 비서 · 일정·자산	직장인 · 팀
3	멀티 에이전트 기반 지능형 고객 응대 플랫폼	금융 상담 · 자산 분석	금융 고객 · 상담원
4	AI 에이전트 기반 웹 정보 수집 및 심층 리서치 보고서 생성	시장 조사 · 경쟁사 분석	기획 · 마케팅
5	AI 협업 에이전트 기반 SW 전주기 개발 자동화 플랫폼	코드 생성 · 테스트	개발팀

모든 주제가 같은 기술 흐름을 거칩니다 — FastAPI + Function Calling → LangGraph 단일 에이전트 → 멀티 에이전트 + Auto-Healing → 스트리밍 UI + AWS 배포

TOPICS · 에이전트 구성 예시

주제별로 그려 보는 멀티 에이전트

#	에이전트 협업 흐름 (예시)	Auto-Healing 초점 (가이드)
1	모니터링 → 진단 → 복구 실행 → 보고	장애 자동 진단 · 복구
2	요청 분석 → 일정 관리 → 자산 관리 → 조율	일정 충돌 자동 조정
3	의도 분류 → 상담 → 자산 분석 → 응답 검수	상담 오류 자동 교정
4	조사 계획 → 웹 수집 → 분석 → 보고서 작성	데이터 누락 자동 보완
5	요구 분석 → 코드 생성 → 테스트 → 리뷰·수정	코드 오류 자동 수정

주제를 고를 때 “우리 팀은 어떤 장애·오류를 일부러 만들고, 그것이 자동으로 복구되는 장면을 보여줄 수 있는가?”를 꼭 물어보세요.

HOW TO CHOOSE

주제 선택 스코어카드

기준	확인 질문	점수 1–5
관심 · 동기	9주 동안 팀 전원이 몰입할 수 있는가?	
데이터 · API	무료·모의 데이터와 외부 API를 바로 구할 수 있는가?	
Auto-Healing	오류를 일부러 만들고 자동 복구를 시연할 수 있는가?	
시연 가능성	3~5분 안에 핵심 흐름을 끝까지 보여줄 수 있는가?	
범위	우리 인원으로 8주 안에 E2E 완성이 가능한가?	
포트폴리오	면접에서 문제와 해법을 설명하기 좋은가?	

① 각자 점수 → ② 합산 → ③ 상위 2개 토론 → ④ 결정

동점이면 데이터·API와 범위 점수가 높은 쪽

PROBLEM DEFINITION

문제 정의 캔버스

누가
타깃 사용자는 누구이고 어떤 상황에 있는가

무엇이 문제인가
사용자가 겪는 불편과 그 비용(시간·돈·위험)

지금은 어떻게
현재 해결 방식과 그 한계

우리 솔루션
에이전트들이 무엇을 대신 해 주는가

왜 멀티 에이전트인가
역할 분리와 자동 복구가 왜 필요한가

성공 지표
무엇을 재면 성공인지 (처리 시간, 복구율 등)

한 문장 테스트: “[누가]는 [문제] 때문에 [비용]을 치르고 있다. 우리는 [솔루션]으로 [지표]를 개선한다.”

DELIVERABLE 1 · 11/13 (금)

기획서 권장 목차

01 **프로젝트 개요** — 배경 · 목표 · 팀 소개

02 **문제 정의** — 타깃 사용자 · 현재 방식

03 **핵심 시나리오** — 사용자 흐름

04 **요구사항 명세** — 기능 · 비기능

05 **서비스 아키텍처 · 기술 스택**

06 **에이전트 구성 · 협업 흐름** (초안)

07 **일정 · 역할 분담** — WBS 요약

08 **리스크와 대응** — 기술 · 일정 · 데이터

요구사항 명세(04)는 기획서 안에 반드시 포함됩니다. 발표 평가의 '기획' 항목도 이 문서를 바탕으로 봅니다.

REQUIREMENTS

요구사항 명세 쓰는 법

ID FR(기능) · NFR(비기능)

우선순위 Must · Should · Could

수용 기준 완료를 확인하는 방법

ID	요구사항	우선	수용 기준
FR-01	서버 지표 이상 시 진단 에이전트를 자동 호출한다	Must	이상 이벤트 → 진단 시작 로그 확인
FR-02	진단 결과에 따라 재시작·롤백 등 복구를 실행한다	Must	수동 개입 없이 복구 완료 기록
FR-03	에이전트 사고 과정을 대시보드에 실시간 표시한다	Should	단계별 스트리밍 표시 확인
NFR-01	프롬프트 인젝션 입력을 차단한다	Must	공격 테스트 케이스 차단 결과 기록
NFR-02	JWT로 인증되지 않은 요청을 거부한다	Must	미인증 요청 시 401 응답

예시: 1번 주제(장애 자동 복구) 기준 · 팀 주제에 맞게 다시 작성하세요

ARCHITECTURE · 초안

아키텍처와 기술 스택 그리기

사용자 인터페이스	Streamlit 프로토타입 · 스트리밍 대화 UI · 사고 과정 대시보드
API 서버	FastAPI(비동기) · Pydantic 검증 · JWT 인증
에이전트 엔진	LangGraph · OpenAI Function Calling · MCP · 자기 성찰 · 보안 가드레일
데이터 · 메모리	Supabase · Redis 세션 · Vector DB 장기 기억
운영 · 배포	Docker · AWS · LangSmith 모니터링 · Auto-Healing 파이프라인

과정에서 배운 기술 범위 안에서 고르세요. 커리큘럼 밖 기술이 꼭 필요하면 강사·매니저와 먼저 상의합니다.

DELIVERABLE 2 · 11/13 (금)

WBS 만드는 법

원칙

- STEP → 작업 묶음 → 1~3일 단위 작업으로 쪼개기
- 대시보드 산출물 마감일에서 거꾸로 계산
- 마감 하루 전 팀 내부 리뷰 일정 포함
- 12/8 중간 발표 준비 기간 따로 잡기
- 작업마다 담당자 1명 · 버퍼 남기기

예시 행

ID	작업	담당	기간
1.1	팀 헌장 · 그라운드 룰	전원	11/10
1.2	요구사항 명세	PM	11/11–12
2.1	API 엔드포인트 설계	백엔드	11/16–18
2.2	ERD · 인덱스 설계	백엔드	11/17–19
3.1	상태 전이 그래프 설계	AI	11/23–26

제출 형태: GitHub Projects 화면 또는 시트 파일을 글자가 잘 보이게 캡처해서 제출

TODAY · 진행 순서 예시

오늘 워크숍은 이렇게 진행합니다

20분	**자기소개** — 강점 · 해 보고 싶은 역할 · 관심 주제
20분	**팀장 선정 · 역할 분담** — 주 담당 + 보조 담당
40분	**팀 헌장 · 그라운드 룰** 작성과 합의
30분	**협업 환경 세팅** — 디스코드 · GitHub · 대시보드 · 드라이브
40분	**주제 탐색** — 5개 주제 검토 · 스코어카드 1차
20분	**팀별 공유 · Q&A** — 팀명 · 팀장 · 후보 주제 발표

CHECKLIST · D-3

금요일까지 끝낼 것

오늘 (화)

- ✓ 팀장 선정 · 대시보드에 표시
- ✓ 팀 헌장 · 그라운드 룰 합의
- ✓ 디스코드 · GitHub · 드라이브 개설
- ✓ 첫 일일 진행 기록 작성

금요일 (11/13)까지

- ✓ 주제 확정 · 대시보드 입력
- ✓ 기획서 (요구사항 명세 포함) 제출
- ✓ WBS 캡처 제출
- ✓ 첫 멘토링 일정 디스코드 공유

좋은 팀이
좋은 프로젝트를 만듭니다

오늘은 코드보다 사람과 약속에 집중하는 날입니다.

첫 마감 · 11/13 (금) 기획서 + WBS

질문은 디스코드 팀 채널로

Q & A