AIO 1기 · FINAL PROJECT

최종 프로젝트 가이드

수강생 안내 — 일정, 산출물, 평가 기준, 팀 운영 방법

2026.11.10 – 2027.01.04

수업 38일 · 5단계 · 중간/최종 발표

OVERVIEW

한눈에 보는 최종 프로젝트

38일
수업일 · 11/10(화) ~ 1/4(월), 총 9주

5단계
기획 → 엔진 → 단일 에이전트 → 멀티 에이전트 → 배포

16건
팀별 산출물 · 평가 12 · 발표 2 · 제출 2

12/8
중간 발표 (화) · 5주차

1/4
최종 발표 (월) · 소스코드·시연영상 제출

20시간
팀당 멘토링 상한 (온·오프라인 합산)

TEAMS & TOPICS

팀별 프로젝트 주제

팀	프로젝트 주제	Auto-Healing 초점 (STEP 4)
1팀	멀티 에이전트 기반 장애 예측·자가 진단·자율 복구 시스템	장애 자동 진단·복구
2팀	지능형 업무 협업 및 자율 수행 멀티 에이전트 (일정·자산 관리)	일정 충돌 자동 조정
3팀	멀티 에이전트 기반 지능형 고객 응대 플랫폼 (금융 상담)	상담 오류 자동 교정
4팀	AI 에이전트 기반 웹 정보 수집 및 심층 리서치 보고서 생성	데이터 누락 자동 보완
5팀	AI 협업 에이전트 기반 SW 전주기 개발 자동화 플랫폼	코드 오류 자동 수정

STEP 1에서 제시 주제 중 하나를 고르고, 정해지면 대시보드 팀 구성 표에 주제와 세부 주제를 입력하세요.

ROADMAP · 9 WEEKS

5단계 로드맵

STEP 1

기획·설계

1주차
11/10 – 11/13

요구사항 · 아키텍처 · WBS

STEP 2

API · Function Calling 엔진

2주차
11/16 – 11/20

FastAPI · Supabase · Redis

STEP 3

단일 에이전트 추론

3–4주차
11/23 – 12/4

LangGraph · 메모리 · 자기 성찰

STEP 4

멀티 에이전트 · Auto-Healing

5–6주차
12/7 – 12/18

협업 엔진 · MCP · 보안 가드레일

STEP 5

대화형 앱 · AWS 배포

7–8주차
12/21 – 12/31

스트리밍 UI · Docker · 통합 테스트

12/8 (화) 중간 발표 — STEP 4 진행 중

1/4 (월) 최종 발표 · 소스코드 · 시연영상

STEP 1 · 1주차 · 11/10 – 11/13

프로젝트 기획 및 설계

목표 · 전 팀 공통

팀 빌딩과 주제 구체화, 요구사항 정의, 서비스 구조와 기술 스택 설계

세부 수행

- 팀 구성 및 주제 선정 (제시 주제 중 택1)
- 요구사항 정의 및 기능 명세 작성
- 서비스 아키텍처 · 기술 스택 설계
- WBS 수립 및 일정 계획

산출물 · 마감 11/13 (금)

기획서 (요구사항 명세 포함)

문서 제출 · 평가 대상

WBS

GitHub 프로젝트 화면 또는 시트를 글자가 잘 보이게 캡처해 제출

노션에 STEP 1 전용 평가표는 없음 — 중간 발표 '기획' 항목에서 다시 평가됩니다.

STEP 2 · 2주차 · 11/16 – 11/20

웹 서비스 기초 · Function Calling 엔진

세부 수행

- FastAPI 비동기 API 서버 설계
- Pydantic 유효성 검증 · RESTful API 설계
- Supabase 저장 · CRUD 구조
- Redis 세션 · JWT 인증·권한 구조
- OpenAI Function Calling · Tool Use 외부 API 호출
- Streamlit 프로토타입 · 인터페이스
- Git/GitHub 버전 관리 · 협업 환경

산출물 · 평가 포인트

API 설계 문서

- 엔드포인트 규칙의 명확성·일관성
- 데이터 모델과 입력 검증 규칙
- Function Calling 호출 흐름의 타당성

DB 스키마 설계서

- 정규화된 ERD와 테이블 관계
- Supabase·Redis 세션, JWT 인증 구조
- 인덱스 전략과 쿼리 최적화

STEP 3 · 3–4주차 · 11/23 – 12/4

LangGraph 단일 에이전트 추론·오케스트레이션

세부 수행

- CoT/ReAct 프롬프트 · Structured Output 제어
- Function Calling 다중 도구 선택·호출
- 임베딩 · Vector DB 설계·구축
- 단기(Session)·장기 기억 하이브리드 메모리
- 상태 전이 그래프 · 대화 흐름 영속성
- 자기 성찰(Self-reflection) 로직
- LangSmith 디버깅·평가·모니터링

산출물 · 평가 포인트

상태 전이 그래프 설계서 · 11/27

- 상태 노드·전이 조건의 완전성
- CoT/ReAct와 그래프 흐름의 정합성
- 모듈화로 확장 대응

자기 성찰 테스트 결과서 · 12/4

- 오류 탐지 시 보정·재시도 루프
- 사고 궤적과 최종 답의 인과관계
- 반복 종료 조건 · 토큰 비용 통제

STEP 4 · 5–6주차 · 12/7 – 12/18

멀티 에이전트 협업 엔진 · Auto-Healing

세부 수행

- 역할 기반 에이전트 분리 · 업무 인계 구조
- 중앙 제어형 · 분산 협업 작업 분배
- 상태 기반 데이터 흐름 · 컨텍스트 전달 (MCP 등)
- AIPP · n8n 노코드 멀티 에이전트 워크플로우
- 컨테이너 패키징 · 서버리스 배포
- 장애 유형별 Auto-Healing 복구 파이프라인
- 프롬프트 인젝션 방어 · 권한 제어 · 응답 필터링

산출물 · 평가 포인트

협업 아키텍처 설계서 · 12/11
역할 분리와 인계 논리 · MCP 컨텍스트 전달 · 메시지 실패·데드락 예외 처리

AI 보안 가드레일 설계 · 12/18
인젝션 방어(입력 필터·샌드박스) · 최소 권한 원칙 · 유해 응답 차단

Auto-Healing 결과서 · 12/18
네트워크·API·LLM 장애별 자동 복구 시연 · 복구 성공률·평균 복구 시간 제시

STEP 5 · 7–8주차 · 12/21 – 12/31

대화형 AI 앱 최적화 · 실서비스 배포

세부 수행

- 시나리오·서비스 기획, 에이전트 협업 흐름도
- 외부 API 브릿지 · 노드 간 오케스트레이션 최적화
- LangGraph 협업 엔진 완성 · 백엔드 연동
- 스트리밍 UI · 에이전트 사고 과정 대시보드
- 사용자 피드백·로그 기반 UX 개선
- AWS/Docker 컨테이너 배포
- 통합 테스트 · 최종 시연회

산출물 · 평가 포인트

대화형 AI 웹 앱 · 12/24 · 배포 URL
스트리밍 응답·UI 완성도 · 사고 과정 가시화 · 무중단·오토스케일링

통합 테스트 결과서 · 12/31*
주요 시나리오 E2E 결과 기록 · 엣지 케이스·결함 재검증

배포 아키텍처 설계도 · 12/31*
AWS 서비스·컨테이너 구성도 · 부하·비용·운영 대비 배포 방식 적정성

* 확정 마감일은 대시보드 일정을 확인하세요

DEADLINES

산출물 마감 일정

마감	주차	산출물	형태
11/13 (금)	1	기획서(요구사항 명세 포함) · WBS	문서
11/20 (금)	2	API 설계 문서 · DB 스키마 설계서	문서
11/27 (금)	3	단일 에이전트 상태 전이 그래프 설계서	문서
12/4 (금)	4	자기 성찰 로직 테스트 결과서	문서
12/8 (화)	5	중간 발표 PT 자료	발표 평가
12/11 (금)	5	멀티 에이전트 협업 아키텍처 설계서	문서
12/18 (금)	6	AI 보안 가드레일 설계 문서 · Auto-Healing 구현 결과서	문서
12/24 (목)	7	대화형 AI 웹 애플리케이션	배포 URL
12/31 (목)*	8	통합 테스트 결과서 · 배포 아키텍처 설계도	문서
1/4 (월)	9	최종 발표 PT · 소스코드 · 시연영상(3–5분)	발표·GitHub·MP4

* 8주차 마감 기준 표기 — 확정 마감일은 대시보드 일정을 확인하세요

PRESENTATION CRITERIA

중간·최종 발표 평가 기준

영역	중간 발표 · 12/8	최종 발표 · 1/4 (추가로 보는 것)
기획	문제 정의·솔루션의 논리, 현실적인 범위·일정, 요구사항이 명세에 반영됐는가	로드맵·마일스톤 구체성, 사용자 피드백을 수집해 반영한 사례
구현	명세 대비 구현 범위, Git 브랜치·협업 체계, 핵심 기능 1개 이상 안정 시연	E2E 통합 완성도, 엣지 케이스 오류 처리, 중단 없는 데모
기술	스택 적합성·최신성, 계획 대비 성취도, 확장을 고려한 아키텍처	프로덕션급 확장성, CI/CD·모니터링·장애 대응, 기술 문서화
발표역량	기승전결 구성·시간 관리, 기술 Q&A 대응력	한계 인식과 대안 제시, 개인별 기여도 명시

발표 자료를 만들 때 이 표를 체크리스트로 쓰세요. 최종 발표에서는 개인별 기여도를 꼭 밝혀야 합니다.

MENTORING

멘토링 활용법

우리 팀이 할 일

- 팀장이 멘토와 일정을 잡고 디스코드 팀 채널에 공유
- 질문과 자료는 멘토링 전에 디스코드에 미리 올리기
- 멘토 피드백을 반영한 내용은 대시보드 일일 진행 기록에 남기기
- 시작 시각에 맞춰 모여 증빙 사진 촬영에 협조

멘토는 정답 대신 실무 경험으로 방향을 제시하고, 과정에서 배운 기술 범위 안에서 지도합니다. 커리큘럼 밖 기술이 필요하면 강사·매니저와 먼저 상의하세요.

팀당 20시간

온·오프라인 합산 상한, 오프라인 권장 · 팀 단위와 1:1 모두 가능

무엇을 물어볼까

기획서 작성 · 기술 가이드 · 일정 관리 · 산출물 피드백 · 이력서·포트폴리오·GitHub

개인정보는 1:1로

이력서 등은 디스코드에 올리지 말고 멘토와 이메일 등으로 주고받기

TEAM OPERATIONS

소통과 팀 운영 규칙

소통 채널

디스코드 채널이 기본 — 매니저도 함께 확인. 오픈채팅·개인 메신저는 최소화. 이력서 등 개인정보는 디스코드 대신 멘토와 1:1(이메일).

팀장 선정

운영 매니저와의 주 소통 창구(일정·제출·지원 요청). 가위바위보 말고 책임감 있는 사람으로. 정해지면 팀 시트 역할을 '팀장'으로.

그라운드 룰

시작 전에 팀이 합의하는 최소 행동 지침. 거창한 목표 대신 어떻게 소통하고 행동하며 무엇을 지킬지 실천 위주로.

리워드

최종 발표 1등 팀 상금 100만 원. 심사위원 평가로 선정하며 상금은 팀장에게 우선 지급.

DASHBOARD

프로젝트 대시보드 구조

통합 대시보드 · 전체 현황

- 프로젝트 기간 · 중간·최종 발표일
- 다음 마감과 전체 제출률
- 팀 구성 — 주제 · 기술 스택 · 팀장·팀원 · 멘토 · GitHub
- 단계별 산출물 일정 (■ 작업 · ★ 마감)

우리 팀 시트 · 팀이 직접 작성

- 팀 정보 · 팀원 및 역할 (팀장 표시)
- 진행 현황 — 진행 일차 · 이번 주 · 제출 · 보완 요청
- 산출물 제출 상태
- 일일 진행 기록
- 강사·매니저 피드백 확인

막히면 ‘0팀(작성예시)’ 시트를 참고하세요

팀 작성 칸 · 우리가 입력 | 회색 · 자동 계산 | 파랑 · 강사·매니저 의견 | 노랑 · 운영진 입력

DASHBOARD RULES

대시보드 사용 규칙

01 **우리 팀 시트의 팀 작성 칸만 입력** — 회색(자동 계산)과 파란 칸(강사·매니저 의견)은 수정하지 않습니다.

02 **행을 넣거나 지우지 않기** — 산출물 표는 통합 대시보드와 행 순서로 연결되어 있습니다.

03 **마감일은 대시보드가 기준** — 일정이 바뀌면 팀 시트 마감일도 자동으로 바뀝니다.

04 **파란 칸 피드백 확인** — 반영한 내용은 일일 진행 기록에 남깁니다.

WEEKLY ROUTINE · 추천

우리 팀 주간 루틴

월요일

이번 주 계획

대시보드에서 이번 주 마감 산출물을 확인하고 역할을 나누기

매일

일일 기록

한 일 · 막힌 점 · 멘토 피드백 반영 내용을 팀 시트에 기록

마감일

제출

산출물을 제출하고 팀 시트의 제출 상태를 갱신

금요일

돌아보기

파란 칸 피드백과 보완 요청을 확인하고 다음 주에 반영

강사·매니저는 대시보드에서 제출 수, 보완 요청, 일일 기록 작성, 피드백 확인 여부를 팀별로 봅니다.

REMEMBER

꼭 기억할 것

배운 기술 범위 안에서

커리큘럼 밖 기술은 교육 규정상 권장되지 않습니다 — 꼭 필요하면 강사·매니저와 먼저 상의

필수 산출물이 먼저

UI/UX 설계서, RAG 검색 품질 평가 결과서 같은 추가 산출물은 운영 매니저와 협의 후 진행

기여는 기록으로 남기기

최종 발표에서 개인별 기여도를 밝혀야 합니다 — 일일 기록과 GitHub 커밋이 근거가 됩니다

FIRST DEADLINE · D-5

첫 마감 11/13 (금)

기획서(요구사항 명세 포함) · WBS

- ✓ 11/10 팀 빌딩 · 팀장 선정
- ✓ 팀별 그라운드 룰 합의
- ✓ 제시 주제 중 택1 · 대시보드 팀 구성 입력
- ✓ 디스코드 팀 채널 · 멘토 일정 협의
- ✓ GitHub 저장소 · 프로젝트 보드 개설
- ✓ 첫날부터 일일 진행 기록 작성

MIDTERM · 12/8 (화) · 5주차

중간 발표 준비

이때까지 갖출 것

- STEP 1~3 산출물 제출 완료 (6건)
- FastAPI 백엔드 + Function Calling 엔진 동작
- LangGraph 단일 에이전트와 자기 성찰 루프
- STEP 4 멀티 에이전트 설계 방향

발표에서 보여줄 것

핵심 기능 1개 이상 안정 시연
실제 서비스 상황을 반영한 시나리오로

문제 정의 → 솔루션 → 범위·일정
요구사항이 명세에 어떻게 반영됐는지

아키텍처와 GitHub 협업 체계
확장 방향 · 브랜치 전략 · 계획 대비 진척

FINAL · 1/4 (월) · 9주차

최종 발표와 제출

최종 발표 PT

중단 없는 데모, 한계와 대안, 개인별 기여도

소스코드

GitHub 저장소에 올리고 저장소 주소 제출

시연영상

MP4 · 핵심 기능 위주 3~5분 내외

그 전에 끝나 있어야 할 것 (STEP 5)

12/24 대화형 AI 웹 앱 배포 URL · 12/31* 통합 테스트 결과서 · 배포 아키텍처 설계도

최종 발표 1등 팀 상금 100만 원 — 심사위원 평가로 선정

9주 동안,
함께 끝까지 만들어 봅시다

질문·소통

디스코드 팀 채널 · 일정·제출·지원 요청은 팀장이 매니저에게

가이드

노션 ‘AIO 1기 최종 프로젝트 가이드’ — 단계별 목표와 평가 기준 원문

대시보드

일정·제출 상태·피드백 확인과 일일 진행 기록

Q & A