

/\* elice \*/

Enterprise

LLM 에이전트 구축

커리큘럼 설계

가전·전자 제조업 대기업 R&D 본부 임직원 대상 | 4주 과정

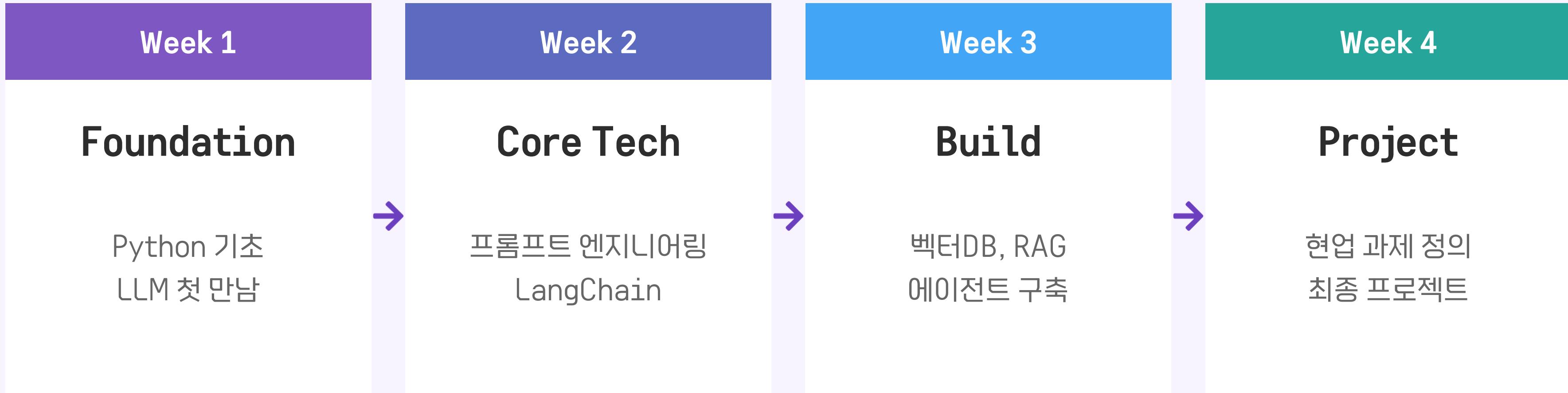
Python 기초부터 RAG 기반 에이전트 구축까지

# 교육과정 한눈에 보기

목표	대상	방식
Python 기초부터 RAG 기반 에이전트까지 전 사이클 완주	비SW 직군 다수의 R&D 임직원 (입문자 친화)	PBL 중심, 주차별 미니 프로젝트 누적 → 최종 프로젝트

기간	일정	총 시간	인원
4주 × 5일 = 20일	6시간 (점심 제외)	120시간	20~30명 / 5인 1조

# 전체 구조와 최종 산출물



## Driving Question

"우리 팀의 업무 문제를 해결하는  
LLM 에이전트를 만들 수 있을까?"

## 최종 산출물

RAG 기반 문서 질의응답 에이전트  
(출처 표시, 할루시네이션 대응 포함)

## PART 1

---

# 교육과정 설계 근거

요구사항 추적표

핵심 용어 정의

교육 목표

학습자 분석

PBL 설계

운영 설계

기술 스택

# 요구사항 추적표 (1/2)

	요구사항	반영 위치	반영 방법
R1	비SW 직군 다수 / SW 개발 경험 없음	1주차 전체	Python을 도구로 접근. 여러 대응법 포함
R2	Python 기초부터 LLM 에이전트 구축까지	1주차→4주차	Foundation→Core Tech→Build→Project 4단계
R3	프로젝트 과제 도출→알고리즘→LLM 연동→결과물	3~4주차	에이전트 워크플로우 설계로 정의
R4	교육 후 현업 데이터로 에이전트 구축 가능	4주차 + Follow-up	현업 유사 샘플 사용, 현업 전환 가이드 2세션
R5	LLM 연동 시 학습 (파인튜닝/RAG)	2주차 Day5 + 3주차	파인튜닝: 개념 1세션 / RAG: 심화 4세션
R6	벡터 DB 구성 방법	3주차 Day1~2	ChromaDB로 임베딩→저장→검색 전 과정 실습
R7	효과적인 프롬프트 구성 방법	2주차 Day1~2	시스템 프롬프트, Few-shot, CoT 패턴별 실습
R8	할루시네이션 최소화 최적화 기법	3주차 Day4	출처 기반 응답, 신뢰도 필터링, 답변 검증 전용 세션

# 요구사항 추적표 (2/2)

	요구사항	반영 위치	반영 방법
R9	간단한 LLM 에이전트 구축	3주차 Day5 + 4주차	RAG 기반 단일 에이전트 + 도구 1~2개 수준
R10	LangChain/LangGraph 등 LLM 프레임워크	2~3주차	LangChain 중심 실습, LangGraph 개념+데모
R11	AI 과제를 주도적으로 리딩할 핵심 인력	4주차	조별 수행 + 개인 담당 모듈 → 개인 산출물 제출
R12	PBL 기반 과정 기획·운영	전 과정	Driving Question + 주차별 미니 프로젝트 누적 → 최종
R13	20~30명 / 오프라인	운영 설계	5인 1조 팀 편성
R14	사전 교육 등 자유롭게 제안	운영 설계	온라인 자가학습(3h), 진단 설문, 환경 사전 세팅
R15	경영진 차원 AI 활용 과제 추진 방침	1주차 Day1 + 4주차 Day5	오픈닝에 조직 비전 연결, 경영진 보고 가능 형태로 발표

# 핵심 용어 정의

## 간단한 LLM 에이전트

RAG 기반 단일 에이전트 + 도구 1~2개  
문서를 넣으면 자연어 질의응답이 가능한 수준

비SW 입문자 4주 기준 현실적 도달선

## 알고리즘 개발

에이전트의 처리 로직 설계  
질의 분석 → 문서 검색 → 프롬프트 조립 → 응답 생성 → 검증

비SW 맥락에서 처리 흐름 설계

## 개별 과제 도출

조별 프로젝트 내에서 각 개인이 담당하는 모듈  
(프롬프트/벡터DB/체인 등)을 개인 산출물로 제출

개별 과제 도출 반영

## 현업 데이터

교육 중에는 현업과 동일 구조의 가상 샘플 사용  
현업 전환 가이드로 실제 데이터 적용 절차 제공

대기업 R&D 보안 현실 반영

# 교육 목표

## 교육 목표

"간단한 LLM 에이전트 구축 프로젝트를 위한 개별 과제를 도출하고, 알고리즘 개발 및 LLM을 연동하여 자신만의 에이전트를 구축할 수 있다."

## 달성 목표 (4주 후 수강생이 만드는 결과물)

자신의 현업 문서(제품 사양서, 기술 보고서 등)를 벡터 DB에 저장하고,  
자연어 질의에 대해 문서 기반으로 정확한 답변을 생성하며 출처를 표시하는 RAG 기반 LLM 에이전트.  
할루시네이션 대응이 적용되고, 에이전트 처리 로직(워크플로우)을 스스로 설계한 결과물.

### RAG 기반 문서 검색

벡터 DB를 활용한 문서 검색  
및 맥락 기반 응답

### 출처 표시 및 정확성

답변의 근거가 되는  
문서 출처를 명확히 표시

### 할루시네이션 대응

검증 체인을 통한  
답변 신뢰도 확보

# 학습 희망 내용 대응 전략

요구 사항	대응 전략	깊이
파인튜닝/RAG 방법	파인튜닝 vs RAG 비교 소개 → RAG 실습 중심	파인튜닝: 개념 1세션 / RAG: 심화 4세션
벡터 DB 구성 방법	ChromaDB로 임베딩→저장→검색 전 과정 실습	2세션
프롬프트 구성 방법	시스템 프롬프트, Few-shot, CoT 패턴별 실습	2세션
할루시네이션 최소화	출처 기반 응답 + 검증 체인 + "모르면 모른다" 프롬프트	전용 1세션 + 전 과정 적용

# 학습자 분석과 설계 반영

## 코딩 경험

없음 ~ 최소 수준

- Python을 도구로 쓸 수 있는 최소 문법만 다룸. 에러 대응법 별도 세션 포함

## 도메인 지식

제품/공정/품질 깊은 전문성

- 최대 강점으로 활용하여 모든 예제를 제조업 시나리오로 구성

## 학습 동기

경영진 방침(외적) + 업무 혁신(내적)

- 매 세션 "이 기술이 내 업무를 어떻게 바꾸는가" 연결

## 디지털 리터러시

엑셀, 사내 시스템 수준

- Jupyter 우선 사용. 환경 사전 세팅으로 초기 장벽 제거

## 좌절 지점

에러 메시지 해독 불가 → 패닉

- 에러 읽는 법 + AI에게 에러 질문하는 법 세션 포함

# 수준별 대응 전략

사전 진단 설문 결과로 3단계 분류 → 실습별 기본/심화 2단계 과제 제공

A그룹

빠른 학습자

20%

특성

Python 경험 有 또는

코딩 유경험자

대응 전략

**심화 과제(Stretch Goal)**

조원 코드 리뷰어 역할

B그룹

표준 학습자

60%

특성

엑셀 수준,

코딩 경험 無

대응 전략

**기본 커리큘럼**

**그대로 수행**

C그룹

추가 지원 필요

20%

특성

디지털 도구 사용

최소 수준

대응 전략

**조내 버디 매칭**

**핵심 코드 스니펫 제공**

## 실습별 2단계 과제 구조

예시: 3주차 Day3 — RAG QA 체인 구축

기본

제품 사양서 1종 → ChromaDB 저장 → 단일 질문 QA 체인 완성

전원

심화

문서 3종 혼합 저장 → 출처별 필터링 + 답변 신뢰도 점수 표시

**A그룹 + 희망자**

# 핵심 설계 원칙

## 1 도구 중심 접근

Python을 프로그래밍 언어로 가르치지 않음.  
LLM 에이전트에 필요한 최소 문법만 다룸.

## 2 제조업 맥락 전면 적용

모든 예제를 제품 사양서 검색, 품질 보고서 분석,  
설비 매뉴얼 QA 등으로 구성.

## 3 매 세션 Quick Win

매 세션 "오늘 만든 것"이 눈에 보이는 산출물 생성.

## 4 PBL 누적 구조

Driving Question 중심,  
주차별 미니 프로젝트가 최종으로 누적.

# PBL 설계와 누적 원리

## DRIVING QUESTION

"우리 팀의 업무 문제를 해결하는 LLM 에이전트를 만들 수 있을까?"

1

AI와 대화하려면 무엇이 필요한가?

Python + API 호출

🏆 나의 첫 AI 비서

2

AI에게 정확한 일을 시키려면?

프롬프트 + LangChain 체인 + 도구

🏆 똑똑한 QA 봇

3

AI가 우리 문서를 읽고 정확히 답하게 하려면?

벡터DB + RAG + 할루시네이션 대응

🏆 기술문서 검색 에이전트

4

우리 팀 문제를 해결하는 나만의 에이전트?

현업 과제 적용

🏆 최종 프로젝트: ① ② ③ 통합

누적 원리: ①의 API 호출 → ②에서 LangChain 체인 → ③에서 벡터DB+RAG 확장 → 최종 현업 문서 적용

# 교육 운영 설계

기간	일일 시간	총 교육시간	인원 / 조 편성
4주 × 5일 = 20일	6시간 (점심 제외)	120시간	20~30명 / 5인 1조

## 일일 시간표

09:30-10:20	오전 1	강의 / 복습	50분
10:30-11:20	오전 2	강의 / 데모	50분
11:30-12:30	오전 3	데모 / 가이드 실습	60분
12:30-13:30	점심	식사 및 휴식	60분
13:30-14:20	오후 1	실습	50분
14:30-15:20	오후 2	실습	50분
15:30-16:30	오후 3	회고 / 미니퀴즈 / Q&A	60분

# 사전 교육 (교육 시작 1주 전)

온라인 자가학습 약 3시간

내용	소요	목적
Python/Anaconda 설치 + Jupyter 실행 확인	30분	Day 1 환경 문제 방지
Python 기초 맛보기 (변수, 출력, 리스트)	1시간	"체험" 수준 사전 노출
ChatGPT 직접 사용해보기 + 소감 3줄 작성	30분	AI에 대한 기대감 형성
"LLM이란?" 10분 영상 시청	30분	기본 개념 사전 노출
교육용 환경 테스트 코드 실행	30분	환경 정상 작동 확인

# 평가 체계

 진단	교육 전	사전 설문 (코딩 경험, 기대 학습 내용)	수준 파악, 조 편성 참고
 형성	매일	미니 퀴즈 3문항 + 실습 산출물 확인	학습 이탈 조기 감지
 중간	2주차 금	미니 프로젝트 ② 조별 시연	피드백 + 방향 조정
 총괄	4주차 금	최종 프로젝트 발표 + 동료 평가 + 개인 산출물 제출	교육 목표 달성 확인

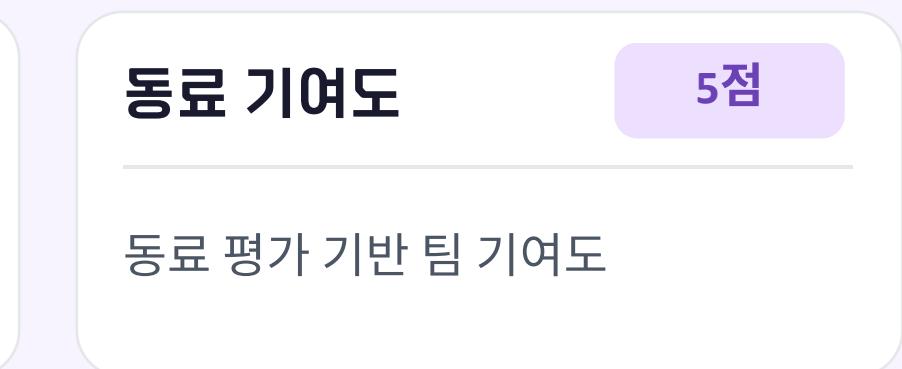
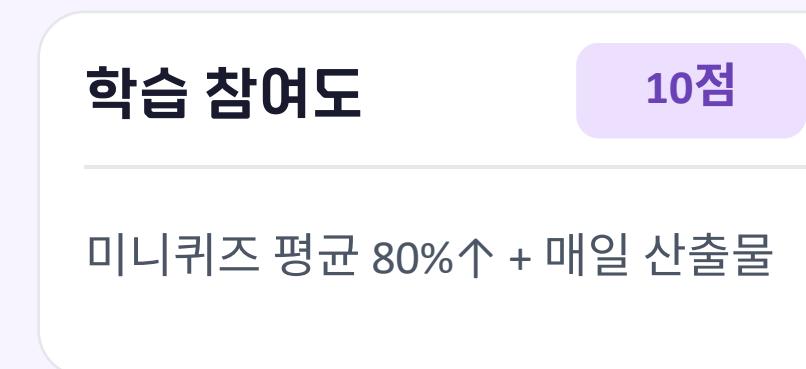
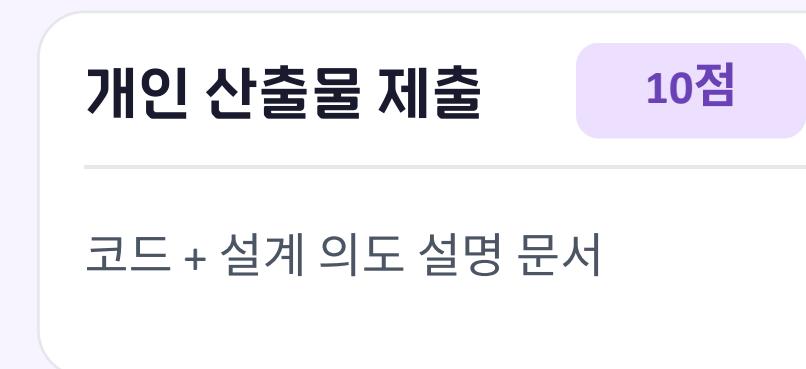
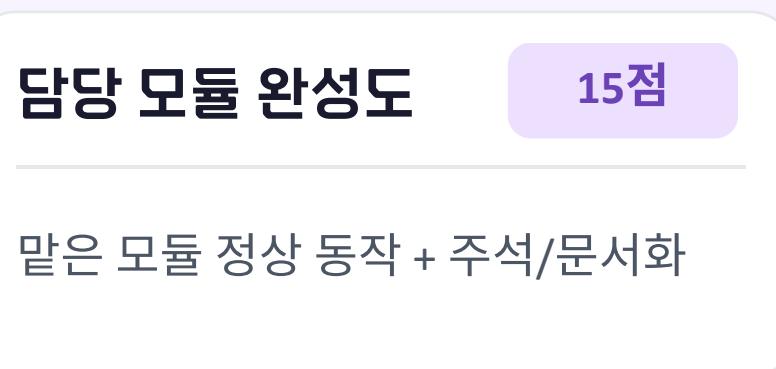
# 최종 프로젝트 평가 루브릭

총점 100점 조별 60 + 개인 40

## 조별 평가 (60점)

평가 항목	배점	상 (100%)	중 (70%)	하 (40%)
RAG 파이프라인 동작	15	전 과정 정상 작동, 다양한 질문에 적절 답변	기본 동작하나 일부 오류	구성했으나 미작동
출처 표시 정확성	10	근거 문서·페이지 정확 표시	출처 있으나 일부 부정확	미구현
할루시네이션 대응	10	검증 체인 + 신뢰도 필터 적용	1~2가지 기법 적용	미적용
현업 적용성	15	현업 시나리오 기반, 구체적 전환 계획	제조업 맥락 반영, 전환 미흡	일반 예제 수준
발표 (시연+설명)	10	시연 매끄럽고 기술 근거 명확	시연 되나 설명 부족	시연 실패 또는 불충분

## 개인 평가 (40점)



 **Excellent**  
90~100점

 **Good**  
70~89점

 **Needs Improvement**  
50~69점

 **Incomplete**  
~49점

# 기술 스택

## Python 3.11+

비전공자 학습 곡선 가장 낮음

R: "Python 기초부터"

## Jupyter Notebook

셀 단위 실행, 비전공자 친화적

## OpenAI API (GPT-4o-mini)

안정적, 문서 풍부, 교육용 비용 효율적

R: "LLM을 연동하여"

## LangChain 0.3+

업계 표준 LLM 프레임워크

R: "LangChain/LangGraph 등"

## LangGraph

에이전트 워크플로우 설계

R: "LangChain/LangGraph 등"

## ChromaDB

로컬 벡터 DB, 설치 간단, 외부 전송 없음

R: "벡터 DB 구성 방법"

# 엔터프라이즈 보안 고려

교육 전

교육용 전용 API 키 발급 (사전 크레딧 충전), 개인 키 사용 금지

교육 중

현업 실제 데이터 반입 금지 → 현업과 동일 구조의 가상 샘플 데이터만 사용

1주차 Day4

API 키 보안 기본 수칙 소개 (환경변수 관리, 코드에 하드코딩 금지)

3주차

ChromaDB는 로컬 실행 → 데이터 외부 전송 없음 확인

4주차 Day3

현업 전환 가이드: Azure OpenAI 전환, 사내 데이터 마스킹, IT 보안팀 협의 체크리스트

## PART 2

---

# 4주 커리큘럼

1주차: Foundation

2주차: Core Tech

3주차: Build

4주차: Project

## WEEK 1

# 1주차: Foundation (Python 기초 & LLM 첫 만남)

1  
DAY

오전 교육 전체 로드맵 시연 · AI 제조업 사례 · Jupyter 환경 확인 · 변수/리스트/딕셔너리

오후 데이터 타입 · 제품 정보 딕셔너리 실습 · 회고 & 미니퀴즈

🏆 산출물: 제품 정보 저장/출력 코드

2  
DAY

오전 조건문 (if/else) - 불량 판정 · 반복문 (for) · 품질 검사 데이터 불량품 찾기

오후 함수 - 반복 작업 줄이기 · 제품 스펙 비교 함수 · 미니퀴즈

🏆 산출물: 품질 검사 자동 판정 스크립트

3  
DAY

오전 파일 읽기/쓰기 (텍스트, CSV) · JSON 데이터 다루기 · CSV 불량률 계산

오후 ★ 에러 메시지 읽는 법 (5종) · 보충 심화 실습 · 미니퀴즈

🏆 산출물: CSV 불량률 계산 + 제품 카탈로그 프로그램

4  
DAY

오전 LLM이란? - 동작 원리 · API 개념 · API 키 발급 & 보안 수칙

오후 첫 API 호출 (temperature/max\_tokens) · 시스템 메시지 역할 부여 · 아이디어 브레인스토밍

🏆 산출물: 가전제품 질문 답변 간단한 챗봇

5  
DAY

오전 1주차 종합 복습 · 미니 프로젝트 가이드

오후 "나의 첫 AI 비서" 구현 · 조별 시연 · 주간 회고

🏆 산출물: 나의 첫 AI 비서 (미니 프로젝트 ①)

# 2주차: Core Tech (프롬프트 엔지니어링 & LangChain)

1  
DAY

오전 1주차 복습 퀴즈 · 프롬프트 엔지니어링 기초 · 기본 패턴 (역할, 맥락, 형식, 제약)

오후 같은 질문 + 다른 프롬프트 결과 비교 · 품질 검사 보고서 요약 프롬프트 설계

🏆 산출물: **프롬프트 패턴별 결과 비교표**

2  
DAY

오전 Few-shot (예시 학습) · Chain-of-Thought (단계별 사고) · 구조화된 출력(JSON) 유도

오후 불량 유형 분류 프롬프트 설계 · 프롬프트 반복 개선 · 미니퀴즈

🏆 산출물: **불량 분류 자동화 프롬프트 (입력→JSON)**

3  
DAY

오전 LangChain 개념 · 핵심: Model, Prompt Template, Parser, Chain · 첫 번째 체인 데모

오후 제품 사양 요약 체인 · 체인 연결 (요약→번역 Sequential Chain) · 미니퀴즈

🏆 산출물: **제품 사양서 요약+번역 LangChain 체인**

4  
DAY

오전 대화 메모리 - AI가 이전 대화 기억 · 도구(Tool) - 외부 도구 사용 · 시연

오후 대화 메모리 제품 상담 챗봇 · 도구 연결 실습 (계산기, 날짜) · 미니퀴즈

🏆 산출물: **대화 기억 + 도구 사용 상담 챗봇**

5  
DAY

오전 파인튜닝 vs RAG 비교: 개념, 장단점, 적용 시나리오

오후 "똑똑한 QA 봇" 구현 · 중간 시연 발표 · 주간 회고

🏆 산출물: **똑똑한 QA 봇 (미니 프로젝트 ②)**

# 3주차: Build (벡터 DB, RAG, 에이전트 구축)

1  
DAY

오전 1~2주차 통합 복습 · 임베딩 이해 (텍스트→숫자 좌표) · 벡터 DB 개념

오후 텍스트 임베딩 생성 · ChromaDB 설치 + 문서 저장 + 유사도 검색 체험

🏆 산출물: 제품 설명 3건 → ChromaDB 저장 → 검색 코드

2  
DAY

오전 문서 로딩 (PDF, TXT, CSV) · 청킹 개념 (긴 문서 나누기) · chunk\_size/overlap 데모

오후 제품 사양서 PDF → 청킹 → 임베딩 → ChromaDB 저장 + 유사도 검색

🏆 산출물: PDF → 벡터 DB → 검색 파이프라인

3  
DAY

오전 RAG 아키텍처 설계 (질의→검색→프롬프트→응답) · LangChain RAG 체인

오후 제품 사양서 QA 체인 구축 · 출처 표시 기능 추가 · 미니퀴즈

🏆 산출물: RAG 기반 문서 QA 체인 (출처 포함)

4  
DAY

오전 할루시네이션: 왜 발생? 제조업 위험성 · 대응 4기법 · 비교 데모

오후 RAG에 할루시네이션 대응 적용 · 파라미터 실험 (chunk\_size/k값) · 미니퀴즈

🏆 산출물: 할루시네이션 대응 적용 고도화 RAG

5  
DAY

오전 LangGraph 개념: 워크플로우 그래프 · ReAct 에이전트 시연

오후 "기술문서 검색 에이전트" 구현 · 조별 시연 · 주간 회고

🏆 산출물: 기술문서 검색 에이전트 (미니 프로젝트 ③)

# 4주차: Project (나만의 에이전트 구축 & 발표)

1  
DAY

**오전** 1~3주차 기술 총복습 · AI 과제 도출 방법론 · 조별 브레인스토밍

**오후** 기획서 작성 · 개인 담당 모듈 정의 · 1:1 멘토링

🏆 산출물: **프로젝트 기획서 + 개인 담당 모듈 명세**

2  
DAY

**오전** 데이터 준비 · 문서 로딩 & 청킹 · ChromaDB 구축

**오후** RAG 파이프라인 구현 · 기본 질의응답 테스트 · 코드 리뷰

🏆 산출물: **✓ 벡터 DB 적재 + 기본 RAG 동작 코드**

3  
DAY

**오전** 품질 개선 (출처 표시, 신뢰도 필터) · 에러 처리 & 예외 대응

**오후** ★ 현업 전환 가이드: Azure OpenAI 전환 · 보안 체크리스트

🏆 산출물: **✓ 고도화 에이전트 + 현업 전환 계획**

4  
DAY

**오전** 통합 테스트 · 버그 수정 · 최종 점검

**오후** 발표 자료 준비 · 개인 산출물 문서화 · 리허설

🏆 산출물: **최종 테스트 완료 코드 + 발표 자료**

5  
DAY

**오전** 최종 발표 (조별 10분: 시연5 + 설명3 + Q&A 2)

**오후** 동료 평가 · 개인 산출물 제출 · 수료식 · 후속 안내

🏆 산출물: **🎉 최종 프로젝트 완성 + 수료**

# 4주차 개인 역할 분담 예시 (5인 조)

역할	담당 업무	개인 산출물
프로젝트 리더	전체 기획·조율, 발표	기획서
데이터 담당	문서 수집·전처리·청킹	데이터 파이프라인 코드
벡터DB 담당	임베딩·ChromaDB·검색 최적화	벡터DB 코드
프롬프트/체인 담당	프롬프트 설계·LangChain 체인	프롬프트 설계서 + 체인 코드
검증/문서 담당	할루시네이션 대응·테스트·문서화	테스트 보고서

→ 조별 프로젝트 + 개인 담당 모듈 명시로 "AI 과제를 주도적으로 리딩할 수 있는 핵심 인력" 양성 (R11)

# 주차별 학습 목표 & 산출물 요약

<b>WEEK 1</b>	<b>학습 목표</b> Python으로 데이터를 다루고, LLM API를 호출하여 응답을 받을 수 있다	<b>산출물</b> <b>나의 첫 AI 비서</b>	요구사항 R1, R2
<b>WEEK 2</b>	<b>학습 목표</b> 프롬프트를 설계하고, LangChain으로 체인을 구성할 수 있다	<b>산출물</b> <b>똑똑한 QA 봇</b>	요구사항 R7, R10
<b>WEEK 3</b>	<b>학습 목표</b> 벡터 DB를 구축하고, RAG 기반 에이전트를 구현할 수 있다	<b>산출물</b> <b>기술문서 검색 에이전트</b>	요구사항 R5, R6, R8, R9
<b>WEEK 4</b>	<b>학습 목표</b> 현업 과제를 정의하고, 나만의 에이전트를 구축·발표할 수 있다	<b>산출물</b> <b>최종: ①②③ 통합</b>	요구사항 R3, R4, R11

감사합니다