



국민대학교  
소프트웨어융합대학  
다학제간캡스톤디자인

계획서		
프로젝트 명	Learning Mate	
팀 명	Learning Mate	
Confidential Restricted	Version 1.1	2025-MAR-27

# 다학제간 캡스톤 디자인 프로젝트

프로젝트 명	<i>Learning Mate</i>
팀 명	<i>Learning Mate</i>
문서 제목	LEARNINGMATE-수행계획서

Version	1.1
Date	2025-03-27

팀원	한승민 (조장)
	정의현
	정진우
	차민수

**CONFIDENTIALITY/SECURITY WARNING**

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 인공지능학부 개설 교과목 다학제간캡스톤디자인 수강 학생 중 프로젝트 “Learning Mate” 를 수행하는 팀 “Learning Mate”의 팀원들의 자산입니다. 국민대학교 소프트웨어융합대학 소속 팀 “Learning Mate”의 팀원들의 서면 허락 없이 사용되거나, 재가공 될 수 없습니다.

**문서 정보 / 수정 내역**

Filename	LearningMate-수행계획서.doc
원안작성자	한승민, 정의현, 정진우, 차민수
수정작업자	한승민, 정의현, 정진우, 차민수

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2025-03-14	한승민	1.0	최초 작성	
2025-03-25	정의현	1.1	피드백 반영	개요 및 개발 목표, 개발 내용 부분 수정

# 목 차

- 1 개요
  - 1.1 프로젝트 개요
  - 1.2 추진 배경 및 필요성
- 2 개발 목표 및 내용
  - 2.1 목표
    - 2.1.1 세부 목표 및 결과물
  - 2.2 연구/개발 내용
    - 2.2.1 세부 목표 및
  - 2.3 개발 결과
    - 2.3.1 시스템 기능 요구사항
    - 2.3.2 시스템 품질 요구사항
    - 2.3.4 결과물 목록 및 상세 사양
    - 2.3.3 시스템 기능 및 구조
  - 2.4 기대효과 및 활용방안
- 3 배경 기술
  - 3.1 기술적 요구사항
  - 3.2 현실적 제한 요소 및 그 해결 방안
    - 3.2.1 하드웨어
    - 3.2.2 소프트웨어
    - 3.2.3 기타
- 4 프로젝트 팀 구성 및 역할 분담
- 5 프로젝트 비용
- 6 개발 일정 및 자원 관리
  - 6.1 개발 일정
  - 6.2 일정별 주요 산출물
  - 6.3 인력자원 투입계획
  - 6.4 비 인적자원 투입계획
- 7 참고 문헌

# 1 개요

## 1.1 프로젝트 개요

본 프로젝트는 오프라인에서 온라인으로 넘어가는 교육 환경 속에서 증가하는 공학 및 수식 기반 질의에 대응하기 위해 효율적인 온라인 학습 경험을 지원하는 종합 솔루션을 개발하는 것을 목표로 한다. 이를 위해 공학 수식과 기호들을 효율적으로 전처리하고 강의 음성 데이터를 LLM이 이해할 수 있도록 가공하여 Retrieval-Augmented Generation(RAG) 기법을 적용하고자 한다.

프로젝트의 주요 개발 범위는 크게 네 가지로 구분된다.

**첫째**, hwp로 된 강의 자료에 포함된 복잡한 표나 이미지를 정보나 순서의 손실이 없도록 JSON 포맷으로 변환하는 전처리 모듈을 구현한다.

**둘째**, 강의 영상에서 강의자의 모든 발화를 time stamp와 함께 json파일로 저장한다.

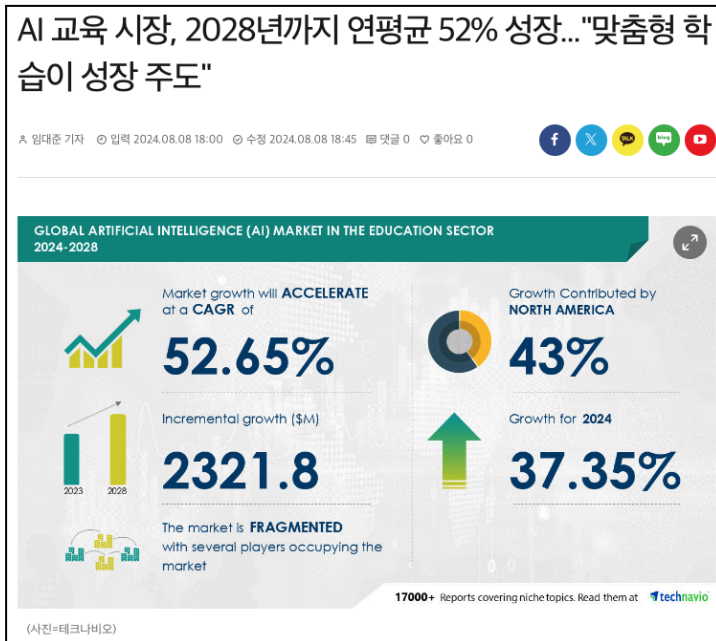
**셋째**, 강의에서 중요한 내용이 요약된 샘플 영상을 자동으로 생성하는 파이프라인을 구축한다.

**넷째**, 강의 추천 시스템을 비롯한 강의 내용에 대한 질의응답 시스템을 구축한다.

## 1.2 추진 배경 및 필요성

### 1.2.1. 기술의 시장 현황

최근 온라인 교육 시장은 전 세계적으로 급격히 성장하고 있으며, 비대면 환경에 대한 수요가 증가함에 따라 콘텐츠 수가 폭발적으로 늘어나는 추세다. 이에 따라 다음과 같은 변화가 두드러지게 나타난다.



**이러닝(e-Learning)의 대중화** : 다양한 주제와 학습 수준을 포괄하는 이러닝 플랫폼이 우후죽순으로 등장하고 있다.

**AI 기반 학습 보조 기술의 부상** : 자동 질의응답, 맞춤형 추천 등 개인화 기능을 갖춘 플랫폼의 시장 점유율이 높아지고 있다.

**빅데이터 활용** : 대형 이러닝 사업자는 방대한 학습 이력을 축적하여, 이를 분석·활용하는 방향으로 사업을 확장하고 있다.

### 1.2.2. 이러닝 플랫폼의 챗봇 시스템 기술 발전 현황

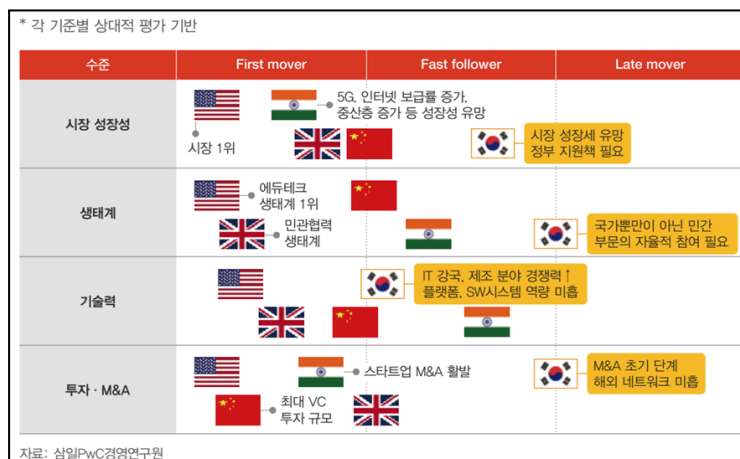
e-learning 플랫폼은 코로나 시기를 보내면서 몇 년간 급속한 발전을 이루며 교육의 패러다임을 변화시키고 있다. 오프라인이 대세였던 교육은 점차 온라인으로 넘어가면서 일타강사들을 추가로 배출하기 시작하였고 거리와 시간의 제약없이 누구나 원하는 강사의 강의를 들을 수 있도록 변화하고 있다. 게다가 듀오링고(Duolingo)나 스픽(speak)과 같은 기존 교육 애플리케이션은 생성형 AI를 적극 도입하여 개인 맞춤형 티칭으로 확장을 진행하고 있다. MATHia나 아이런(ilearn)과 같은 수학 튜터링 시스템들은 수식을 이해하고 이에 따른 답변을 내기 위해 시스템을 발전시키고 있다.

챗봇 시스템 또한 ChatGPT-3.5부터 최근에는 4.5 version까지 등장하였고, Grok, Claude 등 여러 LLM이 발전하면서 이미지나 영상과 같은 데이터를 받아서 처리할 수 있는 Multi-modal LLM이 등장하였다. 여기에 웹 검색이나 DB를 이용하여 성능을 높이는 RAG 시스템, 상황에

따라 LLM이 도구를 선택해서 사용하게 되는 Agent 시스템 등 시스템적으로도 발전하였다.

### 1.2.3. 현재 개발된 관련 시스템 현황 및 개선 필요성

현재 국내외 대부분의 온라인 교육 플랫폼은 단순한 챗봇, 텍스트 중심의 질의응답 시스템을 제공하고 있고, 공학적인 식이 포함된 내용이나 화학식 등에 대한 응답은 상당히 부족한 상태이다. 또한 자동화된 강의 자료 생성이나 영상 요약 기술은 부분적으로만 지원되고 있다.



(표-1 - 주요 국가별 에듀테크 산업 수준)

(표-1)에 따르면 AI를 도입한 에듀테크의 전망은 갈수록 올라가고 있으나, 한국의 에듀테크 산업은 아직 초기 단계인 것으로 나타났다. 삼일PwC 경영연구원이 5개 국가별로 에듀테크 산업수준을 평가한 결과, 한국은 시장성장성과 생태계, 투자 부문에서 미국, 영국, 중국, 인도보다 크게 뒤처졌으며 기술력은 중간 수준에 머물렀다.

기존 이러닝 플랫폼의 챗봇 시스템은 수강생의 질문에 대해 교수나 상담원이 직접 답변해야 해서 응답까지의 소요시간이 굉장히 길다. 기업체에서 제공해준 통계 자료로부터 확인할 수 있던 것은 학생들의 질문에 대한 답변까지 걸리는 소요 시간이 최소 하루 이상이라는 것이었다. 이러한 문제는 답변을 자동화하는 것을 통해 해결할 수 있는데, 수학과 과학 과목과 같은 과목에서의 Q&A 자동화 도입이 효과적이지 않은 가장 큰 이유는-수식이나 도형 같은 복잡한 자료를 처리하지 못 한다는 것에 있다.

## 2 개발 목표 및 내용

### 2.1 목표

본 프로젝트는 학생들의 질문에 보다 신뢰성 있는 답변을 제공하고자 강의 자료를 바탕으로 복잡한 공학식 및 기호 등을 텍스트로 전처리한 것을 데이터베이스로 구축하고 이를 기반으로 RAG 시스템을 도입한다. 또한, 강의 주제와 긴밀하게 연관되거나 중요한 강의 내용에 대한 요약 영상을 자동으로 생성하는 파이프라인을 구현함으로써 수강생들의 강의 핵심 내용 파악을 돕는다. 이를 위해 한글(HWP) 및 강의 영상 데이터를 JSON으로 변환하는 데이터 전처리 솔루션을 개발하며 수식, 표, 공학 기호 등을 유지하는 LaTeX 변환, 수식이나 공학적 내용이 들어간 강의 음성 텍스트 변환(STT), 타임스탬프 기반 문장 구조화 기능을 포함한다.

#### 2.1.1 세부 목표 및 결과물

##### 1. 데이터 전처리 모듈

- 한글 파일(HWP) → JSON 변환, 영상 텍스트 추출(ASR) 및 타임스탬핑
- 최종 산출물: 전처리 소프트웨어, 변환된 JSON 샘플

##### 2. RAG 기반 질의응답 모듈

- 벡터 DB 구축, 문서 임베딩, ChatGPT API 연동
- 최종 산출물: Q&A REST API, 질의응답 UI 프로토타입

##### 3. 영상 요약 모듈 요약

- 알고리즘 적용, 편집된 하이라이트 영상 생성
- 최종 산출물: 요약 영상 출력

##### 4. 강의 소개 자동 생성 모듈

- 다수 강의 정보를 일괄 처리하여 소개 페이지 자동 생성
- 최종 산출물: 자동 생성된 강의 소개 자료(웹 템플릿 또는 JSON 형식)

### 2.2 연구/개발 내용

#### 1. HWP TO JSON 코드의 방법

- HWP 문서를 JSON으로 어떻게 변환할 것인지 연구 필요
- 기존 방식 검토 및 실험  
HWP → JSON 직접 변환의 경우

직접 한글 파일을 열고 변환 과정에서 실행 시간이 길어지는 단점이 있음  
Windows 기반의 OS에서만 동작이 가능

HWP → PDF → JSON 변환의 경우

JSON 변환을 위한 다양한 자료와 시도가 존재하여 먼저 검토

그러나 문서의 순서 저장이 어려운 문제가 있음

특히, 이미지나 표가 있는 경우 텍스트 중간에 표나 이미지가 삽입되는 치명적인 문제 발생

HWP → Markdown/HTML → JSON 변환의 경우

HTML과 MD의 특성으로 Table이나 Image의 추출이 용이

문서의 순서를 지키거나 Metadata를 추출하는 데에도 이점이 있음

- HWP를 여러 형식으로 변환한 후, 문서 형태가 원본과 얼마나 유사한지 비교하는 실험 필요
- HWP에서 텍스트나 수식은 직접 추출하고 형태의 유지가 중요한 Table, Tag를 이용해서 얻을 수 있는 이미지의 경우에는 HTML을 통해서 JSON으로 변환하도록 계획을 수립

## 2. Table을 어떻게 Dictionary로 변환할 방법

- Table의 행, 열 중 선택하여 기준으로 정리할 방법을 고안

## 3. Image를 OCR 모델을 이용하여 처리하는 방법

- ocr 모델을 사용했을때 최대한 텍스트가 잘 추출 되도록 방법을 고안,
- Graph와 같은 형태에서는 어떤 방식으로 처리를 할 것인지 고안

## 4. 한글 파일에서 수식 추출 방법

- 추출된 수식을 LaTeX으로 변환할 방법 고안
- 수식 추출 시 발생하는 시간 지연 문제 해결방안 고안

## 5. 수식, 이미지, 표 추출 시 위치 정보 손실을 없애는 방법

- 수식, 이미지, 표를 따로 추출하면 원본 파일에서의 텍스트와의 위치정보를 알 수 없기 때문에 원할한 rag 시스템을 위해서는 텍스트와의 위치정보를 유지할 방안 고안

## 6. STT 모델의 한계를 극복하는 방법

- 강의 영상에서 STT를 수행할 경우, 수학식이나 화학식과 같은 특수한 기호를 사용할 경우 정확히 변환하지 못하는 문제를 해결할 방안 고안

## 7. Time stamp 제공 방법

- 강의 영상에서 추출된 텍스트가 시간 정보와 함께 저장되도록 고안



## 8. 강의 샘플 영상 생성

- 강의 영상에서 중요 문장 추출 후 요약 영상 생성

## 2.3 개발결과

### 2.3.1 시스템 기능 요구사항

#### 1. 지식기반 데이터 전처리

- 원본이 되는 한컴오피스 한글 표준문서(버전13 이상, hwp)를 RAG에 적용할 수 있는 json 포맷으로 변환
- 원본에 포함된 주요 서식(표, 수학적, 공학식 등)에 대한 정보 유실이 없어야 함 (LaTeX 형식으로 변환)
- 이미지, 차트 등에 대한 정보는 문맥의 앞, 뒤 내용이나 캡션 정보를 활용하여 텍스트로 변환하여 json에 표시되어야 함
- 파일 변환은 하나의 파일, 특정 폴더 전체 등 선택할 수 있어야 함
- 변환된 json 파일명은 원본의 파일명을 따라야 함

#### 2. 영상 데이터 전처리

- 영상의 음성을 json 형식으로 변환
- 강의명, 강좌(챕터)명, 강사명, 강의내용을 json에 포함해야 함
- 문장 단위로 재생 시간에 대한 time stamp 정보를 포함해야 함
- 파일 변환은 하나의 파일, 특정 폴더 전체 등 선택할 수 있어야 함
- 변환된 json 파일명은 원본의 파일명을 따라야 함

#### 3. 타임스탬프 기반 샘플 영상 추출

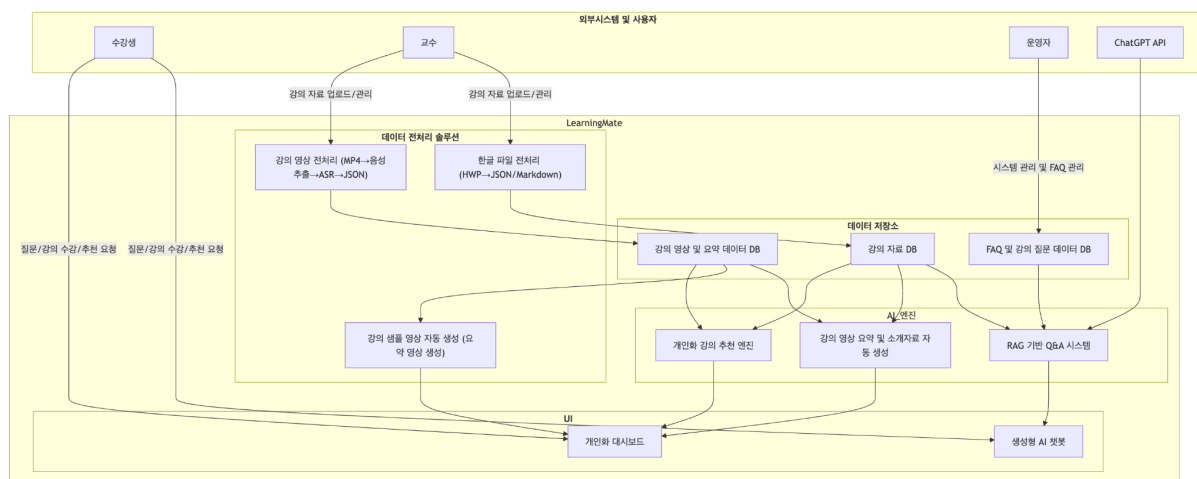
- 원본 강의영상에 대한 요약 영상 생성
- 요약 영상의 길이는 1~10분 사이로 지정할 수 있으며, 요약 내용에 따라 지정된 시간 대비 30초 정도의 편차를 허용
- 음성출력과 영상 싱크 일치
- HD, FHD, 2K, 4K 해상도를 선택하여 생성이 가능
- 영상의 형식은 mp4, WebM, mov, avi에서 선택하여 저장

#### 4. 질문 자동 응답 시스템 구축

- 강의 내용과 연동하여 질문의 답변을 자동 생성
- 이미지, 공학식 등 또한 수월하게 인식 및 분석
- 강의 자료 및 FAQ 학습을 통한 정확한 답변 생성
- API 키를 입력한 이후 RAG 시스템을 사용할 수 있어야 한다.
- 사용자는 강의 영상을 업로드하여 질문을 생성할 수 있어야 한다.

- 보안
  - 사용자의 API 키는 암호되어 저장되지 않고 세션에서만 유지되어야 한다.
  - 유저의 로그인은 JWT Token을 이용하여 Stateless 하게 유지해야 한다.
  - 모든 데이터 전송은 HTTPS를 통해 암호화되어야 한다.
- 성능
  - 질문에 대한 응답은 10초 이내에 처리되어야 한다.
  - 동시 요청 1000건을 처리할 수 있도록 API 인증 시스템을 설계해야 한다.
- 확장성
  - 데이터 전처리의 과정은 비동기적으로 처리되어야 한다
  - 대용량 강의 영상이 업로드될 경우, 비동기 처리를 통해 서버 부하를 최소화해야 한다.
- 사용성
  - API 키가 유효하지 않은 경우, 명확한 오류 메시지와 해결 방법을 제공해야 한다.
  - 질문이 유효하지 않은 경우 재요청 등의 후처리가 가능하도록 해야 한다.
- 유지보수성
  - 코드는 모듈화된 구조로 작성되어야 한다.
  - 모든 코드는 PEP-8에 맞게 작성되어야 한다.

## System Architecture



### 2.3.4 결과물 목록 및 상세 사양

대분류	소분류	기능
데이터 전처리	process_hwp_docs.py	hwp 오토메이션 API 사용
	clipboard.py	hwp에서 html 추출
	table_parser.py	Parsing Table
	image_ocr.py	Image를 Text로 변환 및 문맥데이터 추가
	data_cleaner.py	텍스트 데이터 전처리 및 메타데이터 추출
	latex_parser.py	수식 추출
	json_formatter.py	json 형태로 저장
	성능 개선	비동기 처리 및 로직 개선
영상 기반 데이터 처리	학습 데이터 다운 및 전처리	라이브러리 형태에 맞추어서
	fine-tuning 코드 작성	Wav2vec2
	fine-tuning 후 결과 확인	Wav2vec2
	STT 모델 비교	Wav2vec2 vs Whisper
	모델 선정	Whisper
	추가 데이터 다운	Aihub - 한국어 강의 음성
	데이터셋 서버 업로드	Aihub - 한국어 강의 음성
	k-12 데이터 전처리 2	사투리 보정 및 특수 기호, 제거
	k-12 데이터 전처리 3	숫자와 알파벳만 포함된 파일 추출
	whisper fine-tuning 코드	
	k-12 데이터 전처리 4	파일 간 '외래어 표기법' 일관성 맞추기
	k-12 데이터 전처리 5	문장 부호'를 제외한 특수 기호 제거
	k-12 데이터 전처리 6	학습 데이터 분리
	kr_univ 데이터 전처리 2	사투리 보정 및 특수 기호, 제거
	kr_univ 데이터 전처리 3	숫자와 알파벳만 포함된 파일 추출

모델 학습 및 추론	inference_on.py	inference 진행
	main.py	Train 및 test 진행
	model.py	Whisper model, Wav2vec2 model load 및 processor load
	data.py	학습데이터 및 추론데이터 load
Backend	app.py	app 주요 진입점
	model/	Database model
	views/	여러 뷰포인트
	└ rag.py	rag 관련 route 정의
	└ auth.py	인증 관련 route 정의
	config.py	기본적인 환경변수 및 설정
	uploads/	업로드된 파일 관리 폴더

## 2.4 기대효과 및 활용방안

### - 기대효과

학생들은 강의 내용을 빠르게 검색하고 질문하여 필요한 정보를 즉각적으로 얻을 수 있어 학습자의 학습 효율이 극대화될 것이다. 텍스트뿐만 아니라 이미지, 공학식, 수학식을 포함한 질의응답이 가능해 전공 과목 학습이 더욱 효과적으로 지원된다. 또한, 요약 영상 기능을 활용하면 중요한 강의 부분을 선택적으로 시청할 수 있기 때문에 시간 절약 측면에서 큰 효과가 있을 것이다.

### - 활용방안

대학교와 기업에서 RAG 시스템을 활용하면 강의나 교육 자료에 대한 자동 응답이 가능해져 학습 효율이 높아질 것이다. 특히 수학·공학처럼 복잡한 서식도 지원해 STEM 분야에서도 유용하다. 또한 결과물을 JSON 형태로 제공하기 때문에 이를 통한 AI 학습 분석과 맞춤형 추천 시스템으로 확장할 가능성도 크다.

### 3 배경 기술

#### 기술적 요구사항

##### 1.1 개발 환경

- 운영체제
  - macOS (개발용)
  - Linux (배포용, Docker 컨테이너 기반)
  - Windows(개발용)
- 개발 언어 및 프레임워크
  - Python 3.10+
  - FastAPI 또는 Flask (웹 서버)
  - SQLAlchemy (ORM)
- 데이터베이스:
  - SQLite (개발 및 배포)
  - 필요 시 PostgreSQL 전환 가능
- 서버 운영 및 배포:
  - Nginx 또는 Gunicorn을 사용하여 WSGI/ASGI 서버 배포
  - docker를 이용한 환경 통일
- 코드 스타일 및 정적 분석:
  - PEP-8 스타일 준수
  - Black (코드 포맷터)
  - Flake8 (정적 코드 분석)
  - isort (import 정렬)
- 기타 패키지 및 도구:
  - PyJWT (인증)
  - bcrypt 또는 argon2-cffi (비밀번호 해싱)

- Ollama (LLM 로드 및 RAG 구축)
- Whisper (강의 음성 인식)
- FFmpeg (오디오 처리)

## 1.2 프로젝트 결과물 확인 환경

- 운영체제:
  - Linux 기반 서버 또는 macOS (사전에 Windows에서 변환된 HWP Pickling 데이터 필요)
  - Windows 환경 (HWP 10 버전 이상 설치 필수)
- 필수 소프트웨어 및 라이브러리:
  - Docker & Docker Compose (컨테이너 실행)
  - Nginx (Reverse Proxy, 필요 시)
  - Gunicorn (WSGI 서버, 필요 시)
  - Python 3.10+
  - SQLite (내장 DB, 별도 설치 불필요)
  - HWP (10버전 이상, Windows 환경 필수)
  - Pickle 데이터 (Linux 환경에서 필요)
- 서버 구성 방법:
  - nginx를 Reverse Proxy로 구성하여 Gunicorn(FastAPI/Flask)와 연결
  - 볼륨 마운트를 통해 SQLite DB 및 강의 데이터를 관리

## 현실적 제한 요소 및 그 해결 방안

### 3.2.1 하드웨어

- STT 작업에 필요한 Transformer 기반 모델은 오디오 데이터의 큰 용량과 긴 시퀀스를 처리하기 위해 높은 연산량과 메모리 사용량이 요구되므로 Whisper 모델 중 파라미터가 상대적으로 적은 Small 모델을 사용하고 원본 데이터 중 관련성이 높은 데이터만 선별하여 학습함으로써 불필요한 연산을 줄이면서 성능을 확보하고자 한다.

### 3.2.2 소프트웨어

- 한글(HWP) 파일 처리를 위해서는 ActiveX 및 COM 기술로 인해 Windows 환경이 필수적이어서 한글 파일 처리를 Windows에서 진행한 뒤 후속 작업을 서버에서 수행하는 방식으로 운영한다.
- STT의 오차는 불가피하지만 대학 강의 영상 데이터셋을 활용한 Fine-Tuning과 LLM API 또는

로베이스 후처리 기법으로 허용된 오차 범위 내에서 오차를 최대한으로 줄여 학술적 표현의 인식 정확도를 향상시키고자 한다.

## 4 프로젝트 팀 구성 및 역할 분담

이름	역할
한승민	- 데이터 전처리, DB 구축, VCS, U.I., 백엔드
정의현	- 지식기반 데이터 전처리, 서버 구축, 백엔드
정진우	- Transformer 모델 학습, 배포, DB 구축 및 연동, 백엔드
차민수	- 지식기반 데이터 전처리, U.I., 서버, 백엔드

## 5 프로젝트 비용

항목	예산치 (MD)
hwp 변환 타입 설정	3
Image 추출 및 OCR을 이용하여 Text 변환	6
HTML으로 부터 표 데이터 추출 및 Json으로 변환	3
수식 추출 및 LaTeX 변환	6
Text 데이터 전처리	3
STT Transformer 모델 추론 환경 설정	3
STT Transformer 최적 모델 실험	6
STT Transformer 모델 Fine-Tuning 환경 설정	3
STT Transformer 모델 Fine-Tuning 실험	6
Database 설계 및 구축	6
뷰포인트 생성 및 로직 구현	6
Auth 로직 추가	3
서버 배포 및 운영	6
요약 Transformer 모델 추론 환경 설정	3
요약 Transformer 모델 최적 모델 실험	6

요약 Transformer 모델 Fine-Tuning 환경 설정	3
요약 Transformer 모델 Fine-Tuning 실험	6
요약 데이터를 이용해 핵심 문장 판별	6
핵심 문장을 기준으로 요약영상 생성	6
합	90

## 6 개발 일정 및 자원 관리

### 개발 일정

개발 일정	2025년			
	2월	3월	4월	5월
1. 프로젝트 기획				
2. 관련 지식 학습 및 연구				
3. 지식기반 데이터 전처리				
4. 영상 데이터 전처리				
5. 샘플 영상 생성				
6. RAG 시스템 구현				
7. 서버 구축				
8. 사용자 인터페이스 (UI) 디자인				
9. 중간 점검				
10. 최종 점검				
11. 프로젝트 종료				

### 일정별 주요 산출물

마일스톤	개요	시작일/종료일
------	----	---------



프로젝트 기획	개발 환경 완성 (GPU 서버 할당, Python Environment 구축, 기본 응용 작성 및 테스트 완료) <b>산출물 :</b> 1. 과업 지시서 2. 프로젝트 계획서	2/15~2/26
관련지식 학습 및 연구	관련 지식 파악 및 팀 지식 공유 <b>산출물 :</b> 1. 관련 지식 학습 및 연구 결과 2. 지식 공유	2/27~3/5
지식 기반 데이터 전처리	hwp to json 구현 완료 <b>산출물 :</b> 1. hwp to json 프로그램 (Gradio를 통한 UI 구축) 2. WBS 3. 주차별 보고서	3/6~4/2
영상 데이터 전처리	영상 데이터 Json으로 변환 완료 <b>산출물 :</b> 1. 변환 프로그램 2. 연구 보고서	3/6~4/2
샘플 영상 생성	샘플 영상 생성 프로그램 작성 <b>산출물 :</b> 1. 샘플 영상 생성 프로그램 2. WBS 3. 주차별 보고서	4/2~4/28
중간 점검	데이터 전처리 시스템 구현 완료 <b>산출물:</b> - 중간 보고서	4/21~4/28
RAG 시스템 구현	주어진 데이터와 LLM 연결 <b>산출물 :</b> 1. Source Code 2. README (사용하는 기술 관련)	4/15~5/23

RAG 서버 구축	<p>서버 구축 및 배포</p> <p><b>산출물 :</b></p> <ol style="list-style-type: none"> <li>1. Github repository</li> <li>2. docker image</li> </ol>	4/15~5/23
사용자 인터페이스 구축	<p>User Interface 디자인 및 구축</p> <p><b>산출물 :</b></p> <ol style="list-style-type: none"> <li>1. Figma에 나온 결과물</li> <li>2. 실제 Front-end로 구현</li> </ol>	4/15~5/23
최종 점검	<p>최종 보고</p> <p><b>산출물:</b></p> <ul style="list-style-type: none"> <li>- 최종보고서</li> <li>- 최종 발표</li> </ul>	5/23~5/30

## 인력자원 투입계획

이름	개발항목	시작일	종료일	총개발일(MD)
한승민	영상 데이터 전처리	2025-02-09	2025-04-20	12
차민수, 정의현	hwp 파일 전처리	2025-02-09	2025-04-20	12
한승민, 정진우	영상 데이터 전처리	2025-02-09	2025-04-20	12
한승민, 정진우	샘플 영상 생성 프로그램 구현	2025-04-02	2025-04-28	10
정의현, 차민수	RAG 시스템 구현	2025-04-15	2025-05-23	15
한승민, 정의현	사용자 User Interface 디자인	2025-04-15	2025-05-23	10
정의현, 정진우	서버 구축	2025-04-15	2025-05-23	15
팀원 전부	최종보고서 작성 및 발표 준비	2025-05-23	2025-05-30	2

## 6.4 비 인적자원 투입계획

항목	Provider	시작일	종료일	Required Options
DLPC	국민대학교	25-02-17	25-05-30	
GPU 서버	르몽	25-02-17	25-05-30	cd 등의 cmd 권한
강의 자료	르몽	25-02-18	25-05-30	다양한 데이터 확보

## 7 참고 문헌

번호	제목	출처
1	STT와 Text 요약을 NLP를 이용하여 만든 크롬 확장 프로그램	<a href="#">Chrome Extension for Speech-to-Text Conversion and Text Summarization Using NLP   IEEE Conference Publication   IEEE Xplore</a>
2	한글 파일 변환 툴 1. 한컴 통합문서뷰어: 한컴에서 지원하는 툴 (유료) 2. pyhwp: python 모듈로 hwp 파일을 html, txt 등으로 변환이 가능하다	<a href="https://www.nhnccloud.com/kr/marketplace/content-delivery/hancom-document-viewer?lang=ko">https://www.nhnccloud.com/kr/marketplace/content-delivery/hancom-document-viewer?lang=ko</a> <a href="https://pyhwp.readthedocs.io/en/latest/converters.html#hwp5txt-text-conversion">https://pyhwp.readthedocs.io/en/latest/converters.html#hwp5txt-text-conversion</a>
3	OCR 모델 오픈AI Azure AI Document Intelligence Loader	<a href="#">langchain_community.document_loaders.doc_intelligence.AzureAIDocumentIntelligenceLoader — 🦜🔗 LangChain 0.2.17</a>
4	RAGChain의 한글 Loader	<a href="https://nomadamas.gitbook.io/ragchain-docs/ragchain-structure/file-loader/hwp-loader">https://nomadamas.gitbook.io/ragchain-docs/ragchain-structure/file-loader/hwp-loader</a>
5	pdf 문서의 표와 이미지 전처리 하는 방식 (저희는 hwp를 json으로 전처리 해야하긴 하지만 영감을 줄 수 있을 것 같음)	<a href="https://youtu.be/O3qFWRObAXw?si=8r8D3wJqsn-2sRWw">https://youtu.be/O3qFWRObAXw?si=8r8D3wJqsn-2sRWw</a> 예제 코드: <a href="https://github.com/IHAGI-c/pdf-RAG-with-langchain/tree/c22b33ce73011b53a0b8eb111d4f0c8a0c29e87d/src">https://github.com/IHAGI-c/pdf-RAG-with-langchain/tree/c22b33ce73011b53a0b8eb111d4f0c8a0c29e87d/src</a>
6	Amazon Bedrock으로 Multi Modal 문서에 대해 RAG 적용하기	<a href="https://aws.amazon.com/ko/blogs/tech/amazon-bedrock-multi-modal-document-rag/">https://aws.amazon.com/ko/blogs/tech/amazon-bedrock-multi-modal-document-rag/</a>
7	Multimodal RAG: Chat with PDFs(Image & Tables)	<a href="https://youtu.be/uLrReyH5cu0?si=iA7sLZ9jfpCWoSex">https://youtu.be/uLrReyH5cu0?si=iA7sLZ9jfpCWoSex</a>
8	수식이 포함된 이미지를 LaTeX 코드 또는 자연어 텍스트로 변환하는 방법 - MER(수식 인식) + NLP(자연어 처리) 조합 → 이미지 → LaTeX 변환 후, LaTeX을 텍스트	<a href="https://www.mdpi.com/2076-3417/14/3/1140">https://www.mdpi.com/2076-3417/14/3/1140</a>

9	NHN cloud - OCR 모델 성능 향상 시행착오 # <i>text detection</i> # <i>CNN</i> # <i>Transformer</i>	<a href="https://www.youtube.com/watch?v=FvXzYnFj7Sg&amp;t=408s">https://www.youtube.com/watch?v=FvXzYnFj7Sg&amp;t=408s</a>
10	단순 참고) 수학 수식 표현 문법 - Markdown	<a href="https://rpruim.github.io/s341/S19/from-class/MathinRm.d.html">https://rpruim.github.io/s341/S19/from-class/MathinRm.d.html</a>