



# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	WeQuiz
팀 명	WeQuiz
문서 제목	중간보고서

Version	1.0
Date	2024-03-27

팀원	박규연 (팀장)
	김우림
	배준형
	심재민
	안금장
지도교수	허대영 교수

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “WeQuiz”를 수행하는 팀 “WeQuiz”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “WeQuiz”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	중간보고서-WeQuiz.doc
원안작성자	김우림, 박규연, 배준형, 심재민, 안금장
수정작업자	

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-03-27	팀원 전원	1.0	최초 작성	초안 작성

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

목 차

1	프로젝트 목표	4
2	수행 내용 및 중간결과	5
2.1	계획서 상의 연구내용	5
2.2	수행내용	5
3	수정된 연구내용 및 추진 방향	6
3.1	수정사항	6
4	향후 추진계획	7
4.1	향후 계획의 세부 내용	7

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

## 1 프로젝트 목표


AI가 생성한 퀴즈를 친구들과 함께 푸는 서비스, <WeQuiz>를 개발합니다.

프로젝트의 주요 목표는 사용자가 자신이 업로드한 **PDF** 자료를 기반으로 자동화된 퀴즈를 구성하고, 이를 친구들과 함께 풀면서 지식을 공유하고 협력하는 경험을 제공하는 것입니다. 사용자 친화적인 게임 형식의 **UI**를 통해 사용자가 손쉽게 퀴즈를 생성하고 참여할 수 있도록 유도하며, 학습 및 정보 공유 과정을 더욱 재미있고 효과적으로 만드는 것이 핵심입니다.

1. 자료 기반 퀴즈 생성 기능: 사용자가 업로드한 **PDF**를 분석하여 관련된 퀴즈를 자동으로 생성하는 기능을 개발합니다. 이를 통해 사용자는 복잡한 과정 없이 퀴즈를 만들고, 그에 대한 답안도 효율적으로 평가받을 수 있습니다.
2. 친구와의 퀴즈 대결: 유저는 친구들과 함께 퀴즈를 풀면서 지식을 공유하고 학습할 수 있는 기회를 갖습니다. 이 과정에서 사용자 간의 경쟁과 협력이 동시에 이루어져, 더욱 재미있는 경험을 제공합니다.
3. 게임화된 UI: 게임에 가까운 **UI**를 통해 사용자가 퀴즈 생성 및 참여 과정을 쉽고 재미있게 경험할 수 있도록 유도합니다. 이는 사용자 간의 상호작용과 참여도를 높이고, 퀴즈 풀이를 하나의 즐거운 경험으로 만듭니다.
4. 다양한 분야에서의 활용: 사용자가 업로드하는 자료에 따라 학습 뿐만 아니라 단순히 재미와 오락을 위한 퀴즈도 생성할 수 있습니다.

프로젝트를 통해 얻어가고자 하는 경험적 목표는 다음과 같습니다.

1. 효율적인 협업: 코드 버전관리 뿐만 아니라 **Issue, Pull request, Project, Wiki** 등 Github의 다양한 기능을 효과적으로 활용하여 팀원 간의 원활한 협업을 경험합니다.
2. 원활한 협업: 팀원들 간에 개방적이고 명확하게 소통합니다. 매주 진행상황을 공유하는 회의를 진행하고, **integration test**를 진행하며 팀원 모두가 개발 진행상황을 파악할 수 있도록 합니다.
3. 다양한 협업툴 활용: 효율적인 협업을 위해 **Github** 뿐만 아니라 **Notion, Discord, Google Drive** 등의 도구를 선정하고, 구체적인 활용 방안을 정의하고, 적재적소에 활용합니다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

4. 지속적 통합 및 배포(CI/CD): CI/CD 파이프라인을 구축하여 테스트, 통합 및 배포 프로세스를 자동화합니다.
5. 병렬적인 개발: Front-end, Back-end, ML 파트를 나누고 업무 순서에 의존성이 없도록 업무를 분담하여 효율적으로 개발합니다.
6. MVP 우선 개발: 서비스 필수 기능 개발에 우선순위를 두어 개발합니다.
7. 애자일한 개발: 신속하게 피드백을 반영하여 사용자의 요구사항에 빠르게 대응합니다. 이 과정을 반복하며 서비스를 점진적으로 보완해 나갑니다.


 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

2 수행 내용 및 중간결과

2.1 계획서 상의 연구내용


핵심 서비스

PDF 자료를 퀴즈로 변환	유저가 pdf 파일을 입력하면 내용을 파싱 및 벡터화 하여 문서 내용을 기반으로 LLM이 퀴즈를 생성합니다.
퀴즈 방 생성	퀴즈 생성이 완료되면, 사용자들이 해당 퀴즈를 풀어볼 수 있는 방을 생성합니다.
실시간 퀴즈 풀기	생성된 방에서 게임이 시작되면, 사용자는 실시간으로 퀴즈를 풀어볼 수 있습니다. 퀴즈는 한 문제 씩 진행되고 모든 사용자는 같은 문제를 푼다. 과반수 이상이 답안을 제출하면 짧은 카운트다운 후 다음 문제로 넘어갑니다.
정답 공유	퀴즈가 모두 끝난 후, LLM을 통해 사용자가 작성한 답을 채점합니다. 또한 사용자는 서로가 작성한 답을 공유할 수 있으며, 이에 대한 토론을 나눌 수 있습니다.

 <div>국민대학교 소프트웨어학부 캡스톤 디자인 I</div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

기술 스택

프론트엔드	TypeScript / NEXT.JS / Vanilla-extract / Zustand / Web-Socket
백엔드	JAVA / Spring-boot / PostgreSQL / Redis / Web-Socket
ML	Python / FastAPI / LangChain / S3 / DynamoDB
배포환경 및 개발환경	EC2 / Ubuntu / Docker

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

## 2.2 수행내용

### 프로젝트 관리

Notion, Discord, Github, Figma 등 다양한 툴을 활용하여 협업 및 프로젝트를 관리합니다. 협업툴 활용 방안을 정의하여 용도별로 사용해야하는 툴을 명확히 구분합니다.

Github Projects에 issue를 연동하여 우선 순위에 따라 개발합니다. 현재 각자 개발하고 있는 것, 리뷰할 것, 개발해야 할 것을 팀원 모두가 투명하게 공유함으로써 팀원 간 업무 현황을 파악할 수 있도록 합니다. 이를 통해 효율적인 분업을 할 수 있습니다. 또한 issue 단위로 branch를 파서 issue 단위 개발을 진행합니다.

전체 진행 상황

규연

준형

우림

규장

재민

New view

Filter by keyword or by field

Todo 19

This item hasn't been started

wequiz #60

[BE] Security config 추가 및 jwt filter 를 통한 인가 작업

상

wequiz #70

[BE] 이메일 인증 구현

중상

wequiz #136

[ML] 파일 메타정보 저장 API

wequiz #137

[ML] Vector index 저장

wequiz #138

[ML] numOfQuiz 만큼 문제 생성

wequiz #139

[ML] type에 따른 문제 생성

wequiz #140

[ML] 문서 내 주요 키워드 추출

+ Add item

In Progress 5

This is actively being worked on

wequiz #89

[FE] 메인 로비 구현

wequiz #95

[ML] 문서 요약 Chain 구현

wequiz #156

[ML] 채점 API 생성

wequiz #118

[BE] 채점 문서 요청

상

wequiz #134

[BE] 플레이어 입퇴장

상

+ Add item

In Review 1

Pull Request 올린 상태

wequiz #135

[ML] API key, S3 Access key 시크릿메니저로 이전

+ Add item

Done 45

This has been completed

wequiz #106

[BE] 백엔드 및 DB Docking

상

wequiz #141

[ML] Docker test

wequiz #111

[FE] Docker Compose 에 frontend 추가

상

wequiz #103

[FE] 도커 알아버리기

상

wequiz #86

[FE] 플더 구조 및 내용 재구성

wequiz #92

[FE] 방만들기 페이지 구현

wequiz #54

react 카카오톡 소셜로그인 구현

+ Add item

(Github) Project - 이슈 관리

캡스톤 디자인 I

Page 8 of 19

중간보고서

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
feat/#158	18 hours ago		0	8	#168
feat/#135	3 days ago		5	4	#167
feat/#133	3 days ago		6	1	#166
feat/#128	4 days ago		17	5	#160
feat/#131	last week		45	0	

View more branches >

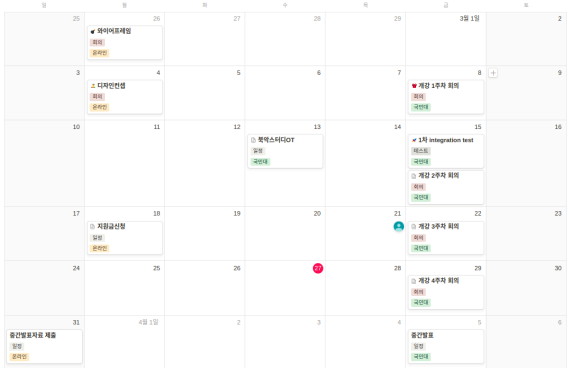
(Github) Branch

Notion을 이용하여 프로젝트의 전반적인 일정을 관리합니다. 캘린더로 일정을 체크하며 기능 개발 일정을 협의합니다. 캘린더에는 학사 일정, 회의 일정, 기능 개발 일정 등 캡스톤 프로젝트에 영향을 미치는 모든 일정을 기록합니다. 프로젝트 진행이나 회의 진행 과정에서 발생하는 중요한 내용들은 문서로 정리하여 문서함에 아카이빙 합니다. 문서별로 태그를 달고, 자주 찾는 문서는 즐겨찾기로 구분할 수 있도록 합니다. 매주 진행하는 회의 중에 발생하는 모든 내용을 회의록에 기록합니다. 워크스페이스를 통해 팀원 별로 각자 개발 진행 내용이나 메모를 할 수 있습니다.

일정표

캘린더 보기

2024년 3월



문서함

즐거찾기

전체문서

유지통

중간발표

시나리오

캡스톤 지원금

협업툴 활용 방안

예외 처리 명세서

Figma

API 명세서 1

env 파일

GitHub 컨벤션

aws

새로 만들기

정리

아이디어

정리

가이드

가이드

정리

가이드

공유

가이드

공유

아카이빙

(Notion) 캘린더 - 일정관리

(Notion) 문서 정리

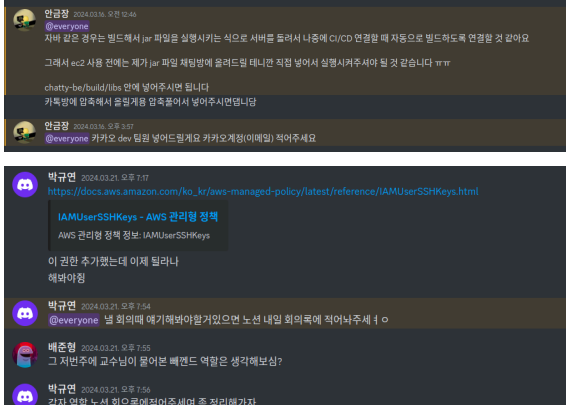
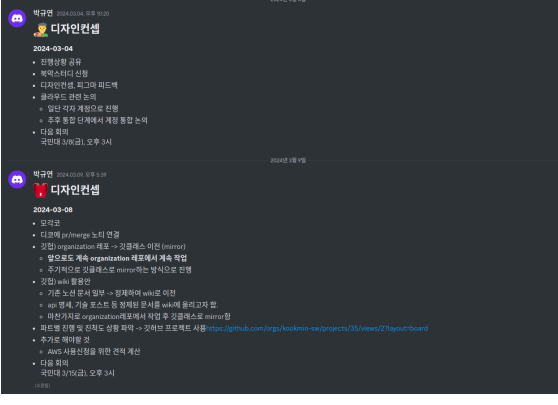
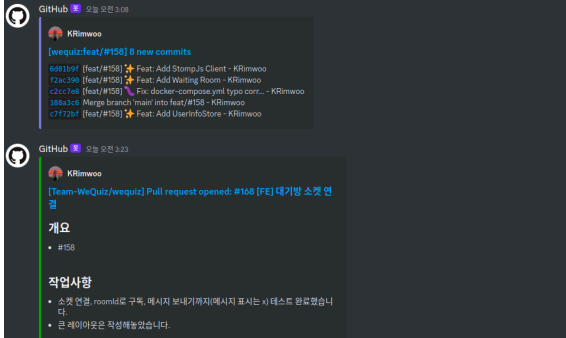
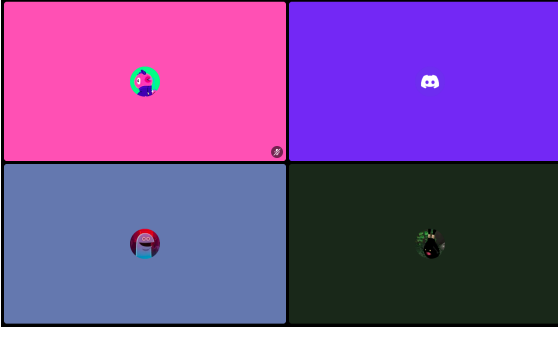
 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

<div> <div>회의록</div> <div> <div>회의록</div> <div> <div>날짜</div> <div>Ag 제목</div> <div>+</div> <div>...</div> </div> </div> <div> <div>2024년 3월 29일</div> <div>개강 4주차 회의</div> </div> <div> <div>2024년 3월 22일</div> <div>개강 3주차 회의</div> </div> <div> <div>2024년 3월 15일</div> <div>개강 2주차 회의</div> </div> <div> <div>2024년 3월 8일</div> <div>개강 1주차 회의</div> </div> <div> <div>2024년 3월 4일</div> <div>디자인컨셉</div> </div> <div> <div>2024년 2월 26일</div> <div>와이어프레임</div> </div> <div> <div>2024년 2월 19일</div> <div>기획감열</div> </div> <div> <div>2024년 2월 17일</div> <div>ERD 설계 회의</div> <div>2</div> </div> <div> <div>2024년 2월 15일</div> <div>MVP확정</div> <div>열기</div> </div> <div> <div>2024년 2월 4일</div> <div>진행상황 공유</div> </div> <div> <div>2024년 1월 22일</div> <div>예산회의</div> </div> <div> <div>2024년 1월 18일</div> <div>기획, 와이어프레임</div> </div> <div> <div>2024년 1월 15일</div> <div>피그마 가입</div> </div> <div> <div>2024년 1월 8일</div> <div>스택검토, 서비스flow, 깃헙컨텐츠</div> </div> <div> <div>2023년 12월 27일</div> <div>캡스톤 초안 회의</div> </div> <div> <div>2023년 12월 21일</div> <div>팀원구인</div> </div> <div> <div>+</div> <div>새로 만들기</div> </div> </div>	<div> <div>워크스페이스</div> <div> <div>88 갤러리 보기</div> <div> <div> <div> <div>그라디오 대응 구성</div> <div>langchain-기존 모듈 뒷부분 호스팅 구현</div> <div>모달 선정</div> <div>pdf 파싱 위기, 파싱하기</div> <div>pdf 파서 선정</div> <div>임베딩 모듈 연동, 임베딩 벡터로 변환하기</div> </div> <div>ML 박규연</div> </div> <div> <div>재민님은 보시요!</div> <div>백엔드 프레임워크 비교</div> <div>이용가적</div> </div> <div>BE 안금창</div> </div> <div> <div> <div>업로드도 프론트그레스 만들어보기.</div> <div>리다이렉트 변경</div> </div> <div>S3 업로드</div> <div>보안검토</div> <div> <div>일단 presigned url 사용 -&gt; 허락받지 않은 사용자가 s3 파일에 접근할 수 없도록</div> <div>성능 고도화</div> <div>FE 박준형</div> </div> </div> <div> <div>DBMS 비교</div> <div>postgresql 설치, wequiz db 생성</div> <div>BE 심재민</div> </div> <div> <div>시작 전 스택, 라이브러리 검토</div> <div>FE 김우림</div> </div> <div> <div>+</div> <div>새로 만들기</div> </div> </div> </div>
<div>(Notion) 주차별 회의록</div>	<div>(Notion) 워크스페이스 - 팀원 별 작업 내용 메모</div>

Figma를 활용해 디자인에 대한 토론을 나누고, 팀원들의 의견이 반영된 디자인을 명확히 확인할 수 있습니다.

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

Discord를 통해 팀원들과 온라인으로 소통합니다. 채팅을 통해 업무 관련 내용을 공유하고, Notion 회의록을 아주 중요한 내용만 간단하게 정리하여 Discord에 한 번 더 공유합니다. 또한, Github에서 발생하는 issue, pull request, merge 등의 사항들에 대한 알림을 Discord에 연결하여 팀원들이 실시간으로 새로운 이슈 및 개발 사항을 확인할 수 있도록 합니다. 음성채널을 활용하여 온라인으로 회의를 진행합니다. 동시에 화면공유를 하면서 원격으로도 팀원들 간에 진행상황을 빠르게 파악할 수 있습니다.

	
(Discord) 채팅방 - 업무 관련 내용 공유	(Discord) 회의록 - 노션 회의록을 간단하게 정리
	
(Discord) Github 웹훅 연결	(Discord) 온라인 회의


 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

## 개발 현황

- 3/15 1차 integration test
  - 로컬 integration
  - 문제 생성 파이프라인
    - front) 파일 입력하고, 방 정보 입력 -> s3에 파일 업로드
    - back) 파일 및 방 정보 ML 서버에 전달
    - ml) s3에서 파일 읽어와 파싱 및 벡터화, 벡터 기반 객관식 문제 1개 생성하여 반환
    - back) ml에서 문제 받아와서 front에 전달
- 3/22 2차 integration test
  - docker compose test
  - 문제 생성 파이프라인
    - front) 파일 입력하고, 방 정보 입력 -> s3에 파일 업로드
    - back) 파일 및 방 정보 ML 서버에 전달
    - ml) s3에서 파일 읽어와 파싱 및 벡터화, 벡터 기반 객관식 문제 1개 생성하여 반환
    - back) ml에서 문제 받아와서 front에 전달
- 3/29 3차 integration test
  - docker compose test
  - 방 생성 파이프라인
    - front) 파일 및 방 정보 입력 -> s3에 파일 업로드
    - back) 파일 및 방 정보 ML 서버에 전달
    - ml) s3에서 파일 읽어와 파싱 및 벡터화, 벡터 기반 객관식 문제 1개 생성하여 nosql에 적재 후 nosql 문서 id를 백 서버에 반환
    - back) ml에서 문제 문서id 받아와서 nosql에서 문제 받아오고 front에 전달
    - front) 문제 풀기 페이지에서 문제 보여주기

### 1. 프론트엔드 기능 구현

- ESLint, Prettier 적용
- 페이지 전환 애니메이션 구현
- 버튼 및 모달 공통 컴포넌트 구현
- S3 SDK를 활용한 멀티 파트 업로드 및 파일 처리기능 구현

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

- 챗봇 리스트 페이지 및 퀴즈 방 생성 기능 구현
- **Stomp.js** 웹 소켓 클라이언트를 이용한 채팅 기능 구현
- 로그인 / 회원가입 기능 구현
- 소셜 로그인 기능 구현
- 로그인한 사용자의 인증정보 관리 구현
- 사용자 인증정보 유무에 따른 페이지 가드 구현
- 컴포넌트 테스트 라이브러리 **Story book** 도입

## 2. 백엔드 기능 구현

- 회원가입 및 로그인 기능 구현
- 소셜 로그인 구현 (카카오 / 구글)
- 유저 및 플레이어 정보 관리 기능 구현
- 커스텀 예외 처리 구현
- 퀴즈 방 생성 및 관리 기능 구현
- 웹소켓을 활용한 채팅 기능 구현

## 3. ML 기능 구현

- **OpenAI API** 호출 비용 체크, 로깅 기능 구현
- **S3**에서 **PDF**파일을 읽고 파싱하는 기능 구현
- 파싱한 **PDF**를 벡터화하는 기능 구현
- 벡터인덱스를 기반하여 객관식 문제를 생성하는 **LLM chain** 구현

## 4. 배포 및 설정 관리

- **CORS** 설정 및 **Docker, Swagger** 설정
- **AWS** 시크릿매니저를 통한 **API 키** 관리
- **PDF** 파싱 **API** 및 **ML** 서버 도커 테스트 기능 구현

## 5. 기타 기능 및 설정 관리

- **git convention** 및 **branch rule** 적용
- **readme** 문서 작성
- **PR** 및 **Issue** 템플릿 설정

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

## 개발 이슈

ML-

### 1. pdf 파일 파싱 방식

- 현재는 langchain의 document loader에서 s3fileloader를 사용하고 있음. s3fileloader 내부적으로 기본적인 pdf 파싱 기능을 제공하고 있다고 판단되는데, pypdf같은 전문 pdf파서(약 10종류)와 성능면에서 차이가 존재하는지 확인할 필요가 있음.

### 2. 파싱한 파일을 split하는 방식에 따른 성능 차이

- 파일에 대한 요약을 생성하는 기능에서 split한 문서마다 요약하고 그 요약본들을 합쳐 하나의 요약본을 만드는 매퍼듀스 방식을 채택하고 있는데, split 방식이 최종 요약본의 질을 좌우함.
- split한 문서마다 키워드를 뽑아내는 작업에서도 split의 방식이 영향을 미칠 것으로 예상됨.

### 3. 객관식/주관식/단답형 문제 형식 반영

- 문제 형식에 따라 문제 생성 프롬프트와 체인을 각각 생성해야함.
- 문제 형식에 따라 채점 방식을 달리 해야함.
  - 주관식은 gpt로 채점
  - 단답형은 룰베이스로 채점하는 것이 적절해 보이나 gpt가 답안을 정확하게 단어로 내뱉는다는 보장이 없으므로 방식을 고민 중에 있음
  - 객관식은 gpt의 답안이 정확하게 숫자가 아니라 문장인 경우(ex. 답은 3번입니다. 왜냐하면...)를 고려하여 답안 내에 숫자를 추출한 후, 룰베이스로 채점을 진행할 예정. 이 부분도 (ii)와 비슷한 고민 중.

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

#### 4. gpt가 벡터 기반으로 문제를 생성하는 방식

- a. retriever를 memory로 넣어주는 방식, retriever 기반의 chain 모듈을 활용하는 방식, 그 외 다양한 방식의 모듈들이 있음. 각 모듈들의 차이와 명확한 용도를 알고 해당 TASK에서 가장 적절한 방안을 찾아야 함.

#### 5. 여러 문제를 생성하는 방식

- a. 하나의 문서에서 여러개의 문제를 생성하다 보면 비슷한 문제가 생기는 이슈가 발생할 수 있음. 이를 해결할 방안 중 한 가지로 문서에서 주요 키워드를 뽑아내고, 그 키워드 단위로 문제를 생성하는 방안을 고민하고 있음. 다만 유효한 키워드를 추출해 낼 수 있을 것인가? 하는 것도 새로운 문제점임.
- b. 백서버에서 보낸 api콜 하나로 계속해서 문제를 생성하고 리턴해야 함. 이 부분은 yeild로 처리할 예정이나, ml서버에서도 문제들을 잘 쌓아뒀다가 nosql에 한번에 업로드 해야하는데, 내부적으로 모든 문제가 다 생성될 때까지 캐시해놓는게 적절한 방법인지 고민해볼 필요가 있음.

BE-

#### 1. 문제 생성과 반환 방식 변경

이전 기획은 각자 제한 시간 안에 모든 문제를 다 푸는 방식이었으나, 먼저 퀴즈를 다 푼 사람은 모두가 다 풀 때까지 기다려야 하는 문제가 있어 퀴즈 진행 방식을 변경함. 변경한 기획은 문제를 하나씩 보여주고 과반수 이상이 풀면 다음 문제로 넘어가는 방식.

기획 변경에 따라 ML 서버에서 문제를 생성하고 백엔드 서버에 반환하는 방식도 변경이 필요해짐.

##### a. 이전 방식

- i. 클라이언트에서 방 생성 요청이 들어올 때 백엔드 서버에서 ML 서버에 파일 정보와 함께 문제 생성 요청을 보냄
- ii. ML 서버에서 모든 문제를 생성하여 DynamoDB에 저장하고 문서ID를 반환
- iii. 백엔드 서버에서 퀴즈방 DB에 방마다 문서ID를 함께 저장하고 클라이언트에 방 번호를 반환하면 방이 생성됨

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

- iv. 클라이언트에서 문제 정보 GET 요청이 들어오면 백엔드 서버에서 DynamoDB에 접근하여 문서 ID로 문제 정보를 가져오는 방식
- b. 기획이 변경되면서 기존 방식으로 진행했을 때의 문제를 인식함. 한 문제마다 클라이언트에서 GET 요청을 보내야 하고 백엔드 서버에서는 GET 요청이 들어올 때마다 DynamoDB에서 데이터를 읽어와야 하는데, 하나의 게임이 진행되는 동안 GET 요청을 여러 번 하는 것과 DynamoDB에 여러 번 접근하는 것 모두 비효율적이라고 생각하여 다른 방식을 고민하게 됨. 지금 생각하고 있는 방식은 다음과 같음.
  - i. 클라이언트에서 방 생성 요청이 들어올 때 백엔드 서버에서 ML 서버에 파일 정보와 함께 문제 생성 요청을 보내는 API를 비동기적으로 호출함.
  - ii. 백엔드 서버에서 클라이언트에 방 번호를 반환하고 방이 생성됨.
  - iii. 방 생성과 비동기적으로 ML 서버에서 한 문제씩 생성해 백엔드 서버에 반환
  - iv. 백엔드에서는 반환받은 문제를 Redis 등을 사용하여 큐와 비슷한 방식으로 서버에 저장
  - v. 클라이언트에서 한 문제씩 GET 요청이 들어올 때 큐에서 하나씩 반환

### 3 수정된 연구내용 및 추진 방향

#### 3.1 수정사항

변경된 내용

- 기획 내용 변경
- 사용자가 업로드한 학습 자료를 기반으로 질문에 대한 답변과 맞춤형 문제 생성 및 학습 환경을 제공하는 학습용 AI 챗봇이라는 주제의 이전 기획에서 현 기획으로 변경

변경 이유



 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

- 해당 플랫폼은 국내외로 이미 경쟁사가 있고, 구현된 기능들이 저희가 개발하고자 했던 기능과 많이 겹쳤습니다. 경쟁사가 주고자 하는 가치 이상의 것을 저희 팀이 제공하기 어렵다고 판단하였습니다.
- 문제 제작의 경우 과목에 따라 구현에 한계가 존재하였습니다. 언어와 관련된 과목의 경우 이미 존재하는 모델을 사용하여 나쁘지 않은 수준의 문제를 ML을 통해 제작할 수 있지만, 수학과 같은 과목은 모델을 직접 구현하거나 수정해야 합니다. 이는 저희가 감당할 수 있는 수준보다 많은 기술력과 리소스가 투입된다고 판단하였습니다.
- 학습을 좀 더 편하고 재미있게 할 수 있을까?라는 고민에서 친구와 함께 퀴즈를 풀자는 아이디어가 나왔습니다. 시중에 있는 서비스인 ‘열풍타’나 ‘투두메이트’는 간단한 기능을 제공하지만 동료와 함께 할 수 있다는 점에서 많은 인기를 끌고 있습니다.

#### 추진 방향


- 클라이언트에서 s3로 pdf를 업로드 하는 기능, 업로드된 파일을 바탕으로 llm을 통해 문제를 제출하는 기능은 유지하였습니다.
- 새로운 기획과 컨셉에 맞는 디자인을 새로 제작하고, 이를 기반으로 추가 기능들을 구현하고 있습니다.
- 서비스 MVP를 빠르게 개발 및 배포하여 유저들의 피드백을 받고 반영하며 점진적으로 서비스를 보완해 나갈 예정입니다.

## 4 향후 추진계획

### 4.1 향후 계획의 세부 내용

#### MVP 완성

핵심 기능만이 구현된 MVP를 최대한 빠르게 완성하여 실제 배포를 진행합니다. 이후 사용자 피드백을 받아 추가적으로 필요하다고 생각되는 기능을 구현하고, 기존 코드의 허점을 파악하여 리팩토링을 진행할 예정입니다.

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

## 서비스 배포

EC2를 활용해 1차적인 배포를 진행할 예정입니다. 또한 EC2 이외의 설정들을 어떻게 관리하고 유지보수성을 높일지 고민해야 하며, 도커와 git action을 통한 배포 자동화 방법을 고려하고 있습니다.

## 테스트 작성

현재 프론트엔드는 STORYBOOK을 도입하여 컴포넌트 단위의 테스트를 진행하였습니다. 이후 프론트엔드, 백엔드, ML 파트 모두 테스트 라이브러리를 도입하여 테스트 기반의 개발을 진행할 예정입니다.

## 파트 별 세부 목표

### Front-end

- 실시간 처리 최적화  
WeQuiz에 사용되는 핵심 도메인 중 하나인 '웹소켓을 활용한 실시간 처리'에 대한 최적화를 진행할 예정입니다. 데이터 전송 효율성을 위해 데이터 압축, 데이터 포맷 최적화를 고려할 예정입니다.
- 무한스크롤 최적화  
현재 만들어진 방 목록에 무한스크롤을 적용할 예정에 있습니다. 무한스크롤은 프론트엔드 분야에서 가장 많이 사용되는 기술 중 하나인 만큼, 단순 구현에서 끝나지 않고 데이터가 쌓였을 때 렌더링 속도 등을 측정하여 기능을 최적화할 예정에 있습니다.

### Back-end

- 예외처리  
개발한 기능들의 모든 예외 상황에 대한 처리가 필요합니다.

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	WeQuiz	
	팀 명	WeQuiz	
	Confidential Restricted	Version 1.0	2024-MAR-27

- 고객 이메일 인증  
회원가입 시 입력한 이메일을 인증하여 추후 서비스 관련 메일 등을 전송하거나 유효한 계정인지 확인할 예정입니다.
- 실시간 처리 최적화  
**Redis**를 활용해 실시간 정보를 빠르게 읽어올 수 있게 할 예정입니다.
- 코드 리팩토링  
리팩토링을 통해 스프링과 자바의 핵심인 객체지향을 원리를 활용하여 코드의 재사용성을 높이고, 유지보수성을 높일 예정입니다.
- 테스트 코드 작성  
테스트 코드 작성을 통해 직접 작성한 코드에 대한 신뢰도를 높이며 기능개발 및 코드 리팩토링할 때 용이하게 할 예정입니다.

## ML

- 문제 생성 및 채점 기능 개발
  - **CoT** 방식의 프롬프트 엔지니어링 및 실험으로 **LLM** 아웃풋을 주도하는 것이 해당 기능 구현에서 가장 핵심적이고 도전적인 부분입니다.
- 파일 메타 데이터 적재 기능 개발
  - **raw**파일 저장 대신 메타 정보를 저장함으로써 정보 저장에 드는 리소스를 최소화합니다.
  - 맵리듀스 방식을 적용하여 파일 내용을 효율적으로 요약하는 **Chain**을 개발합니다.
- **AI Back-end** 서버 개발 및 속도 최적화
  - 문제 생성, 채점, 메타 데이터 적재 등의 **API**를 개발합니다.
  - 문서 파싱, **LLM inference** 등에서 발생하는 속도 지연 문제를 해결합니다.