



캡스톤 1조

박규연, 김우림, 배준형, 심재민, 안금장

AI가 생성한 퀴즈를 친구들과 함께 푸는 서비스

그건 지피티 3.5로 못해요



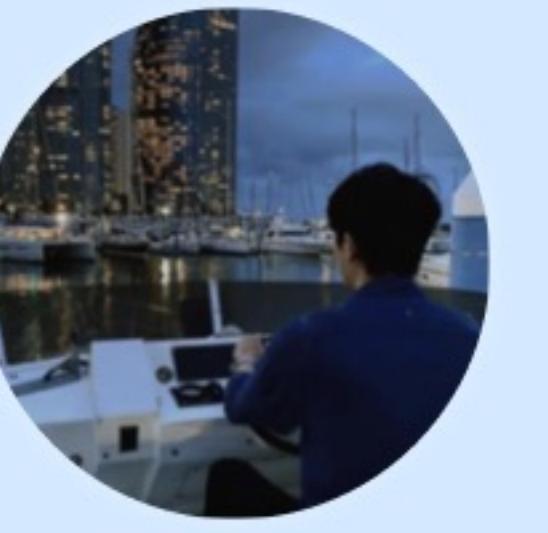
🤖 박규연

'+ 앞 - 단'+



🎨 김우림

api 언제나와요?



🎨 배준형

위퀴즈의 심장❤ 백엔드



⌚ 심재민

'- 뒷 - 단'



⌚ 안금장



WeQuiz

위퀴즈~? Yes,,,

5명의 참여자가 **WeQuiz**를 만들었습니다.

WAITING

메세지를 입력하세요



준비



## 문서 기반 퀴즈 자동 생성

사용자가 업로드한 PDF를 분석하여 문서의 내용에 기반한 퀴즈를 AI가 생성합니다.

## 친구들과의 퀴즈 대결

친구들과 함께 실시간으로 퀴즈를 풀면서 재미있게 지식을 공유하고 학습할 수 있는 기회를 제공합니다.

## 게임화된 UI

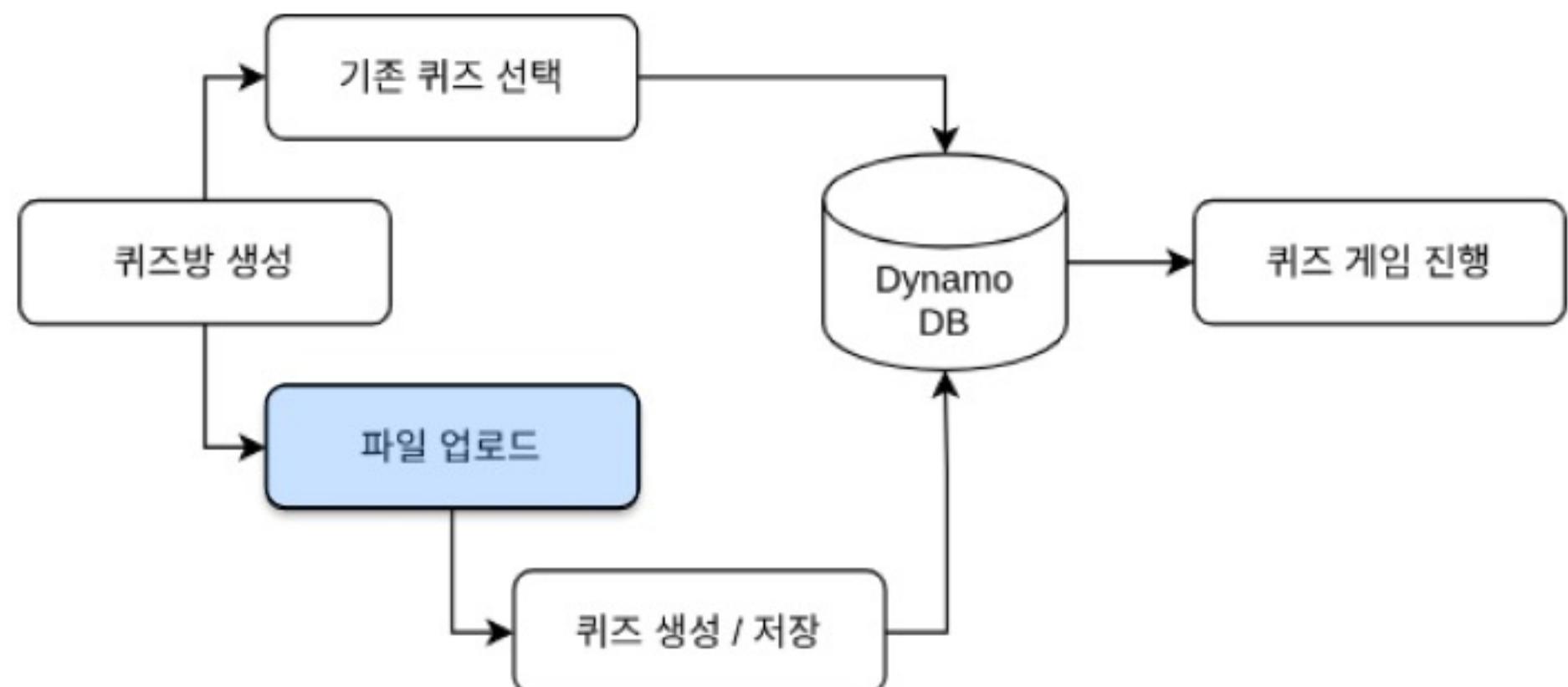
게임같고 친근한 UI를 통해 사용자가 퀴즈 생성 및 퀴즈 풀이에 참여하는 과정을 쉽고 재미있게 경험할 수 있도록 유도합니다.

## 다양한 분야에서의 활용

사용자가 업로드하는 자료에 따라 학습 뿐만 아니라 단순히 재미와 오락을 위한 퀴즈도 생성할 수 있습니다.

## 방 만들기 - 파일 업로드

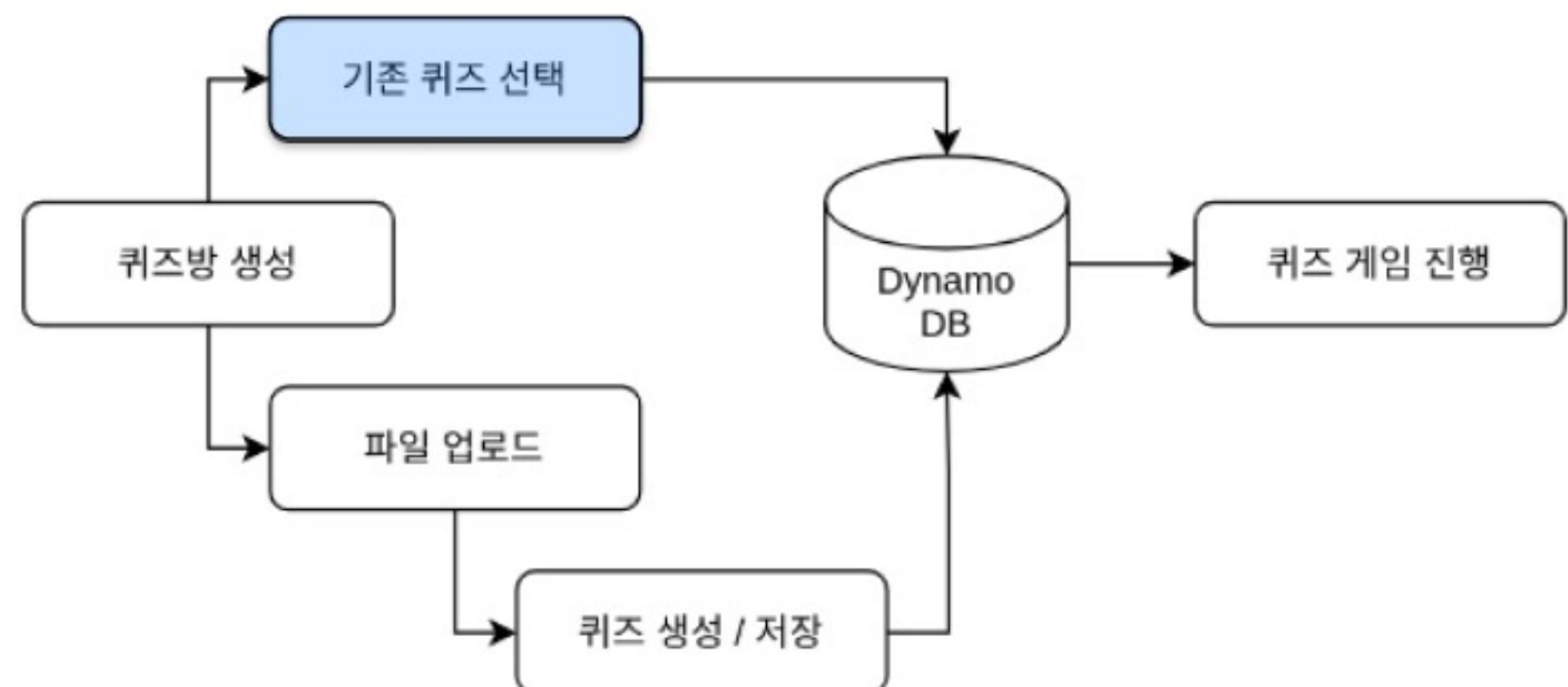
사용자가 업로드한 파일을 기반으로  
LLM이 퀴즈 생성



The screenshot shows the '방 만들기' (Room Creation) interface. At the top, there's a '설정' (Setting) section with fields for '방 제목' (Room Title) set to '무한도전 퀴즈', '설명' (Description) set to '무도 팬들만', '인원 수' (Number of Participants) set to 5, and '문제 수' (Number of Questions) with options 5개, 10개 (selected), and 15개. Below this is a '문제 만들기' (Question Creation) section with '직접 제작' (Direct Creation) selected over '리스트 선택'. A message '업로드 완료!' (Upload Complete!) is displayed next to the file '무한도전 - 나무위키.pdf'. At the bottom right is a large blue '생성하기' (Create) button.

## 방 만들기 - 기존 퀴즈 선택

자신이 기존에 만든 퀴즈 중에서 선택하여  
DynamoDB에 있는 데이터를 가져옴



**설정**

방 제목	무한도전 퀴즈
설명	무도 팬들만
인원 수	5
문제 수	<input checked="" type="radio"/> 5개 <input type="radio"/> 10개 <input type="radio"/> 15개

**문제 만들기**

직접 제작  리스트 선택

문제를 선택해주세요.

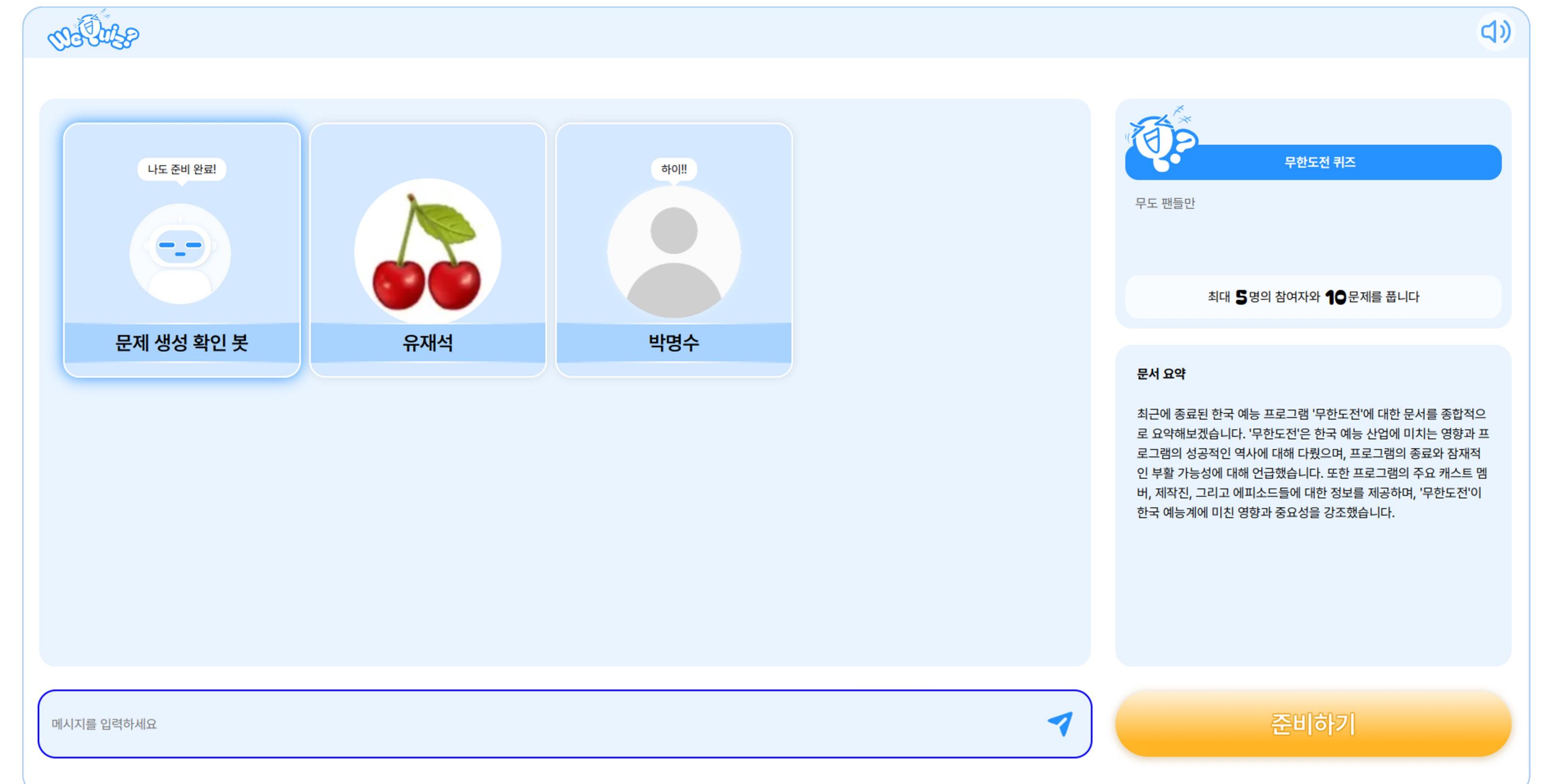
- 1번 문제 리스트
- 2번 문제 리스트
- 3번 문제 리스트
- 4번 문제 리스트
- 5번 문제 리스트
- 6번 문제 리스트
- 7번 문제 리스트

**생성하기**

This screenshot shows the 'Quiz Creation' section of the WeQuiz interface. It includes a 'Settings' tab with fields for room title ('무한도전 퀴즈'), description ('무도 팬들만'), participant count ('5'), and question count ('5개' is selected). Below the settings is a 'Question Creation' section with a 'List Selection' tab selected. It displays a dropdown menu of seven question lists. At the bottom is a large blue 'Create' button.

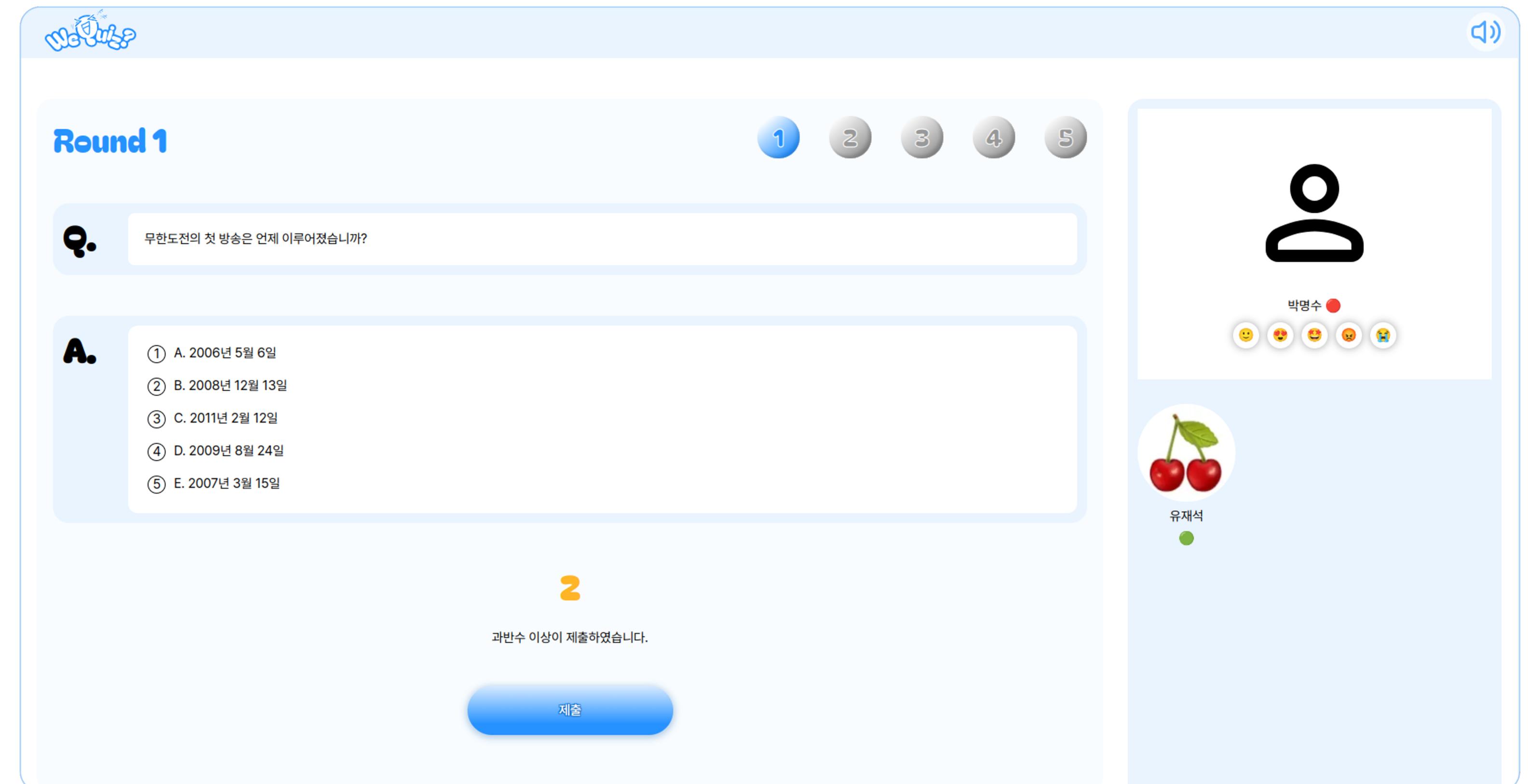
## 대기실

- 문제 생성을 기다리는 동안 채팅
- 실시간 준비 상태 반영
- AI의 문서 요약 표시
- 봇 캐릭터를 활용한 문제 생성 확인



## 퀴즈 풀이

- 5지선다, O/X, 객관식 유형 제공
- 과반수 제출 시 3초 카운트 다운 일관된 시간을 제공하기 위해 서버에 서 카운트
- 한 문제가 끝나면 모든 플레이어의 답안을 채점하여 DynamoDB에 업로드



The screenshot shows a quiz interface from the WeQuiz app. At the top, it says "Round 1" and has numbered buttons 1 through 5. Below that is a question card with a question mark icon and the text "무한도전의 첫 방송은 언제 이루어졌습니까?". To the right of the question are five numbered options: A. 2006년 5월 6일, B. 2008년 12월 13일, C. 2011년 2월 12일, D. 2009년 8월 24일, and E. 2007년 3월 15일. Below the options is a large orange number "2". At the bottom of the screen is a blue button labeled "제출". On the right side of the screen, there is a user profile section with a large black outline icon, a red cherry icon with the name "유재석", and a row of small emoji icons.

## 중간 점수

- 5문제씩 라운드 구분
- 라운드 종료 시 누적 점수와 등수 표시
- 점수 계산법은 문제 난이도와 풀이 시간을 모두 고려

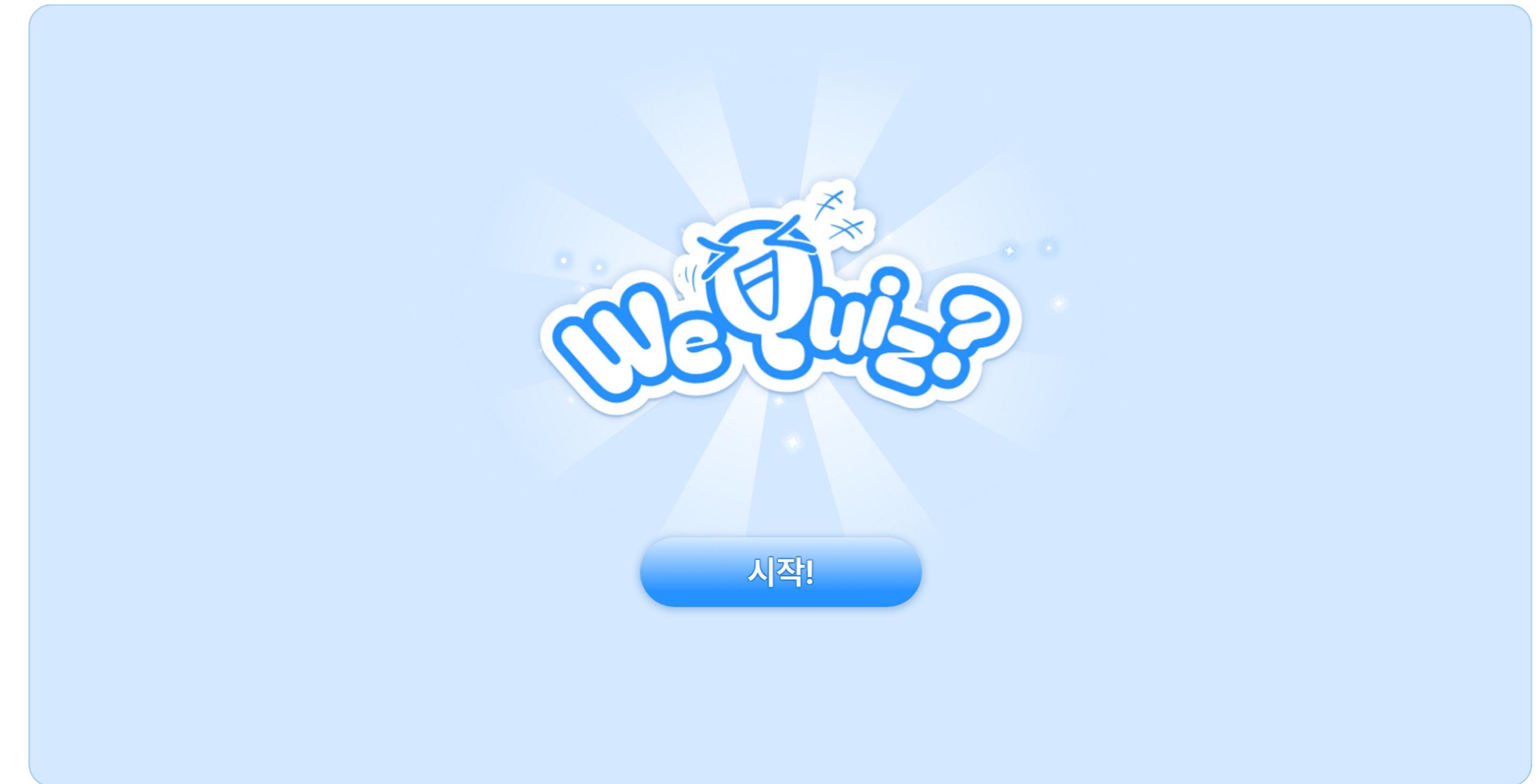


## 최종 결과

- 문제 정답과 정답률 확인
- 문제 별로 모든 플레이어의 답안 확인 가능

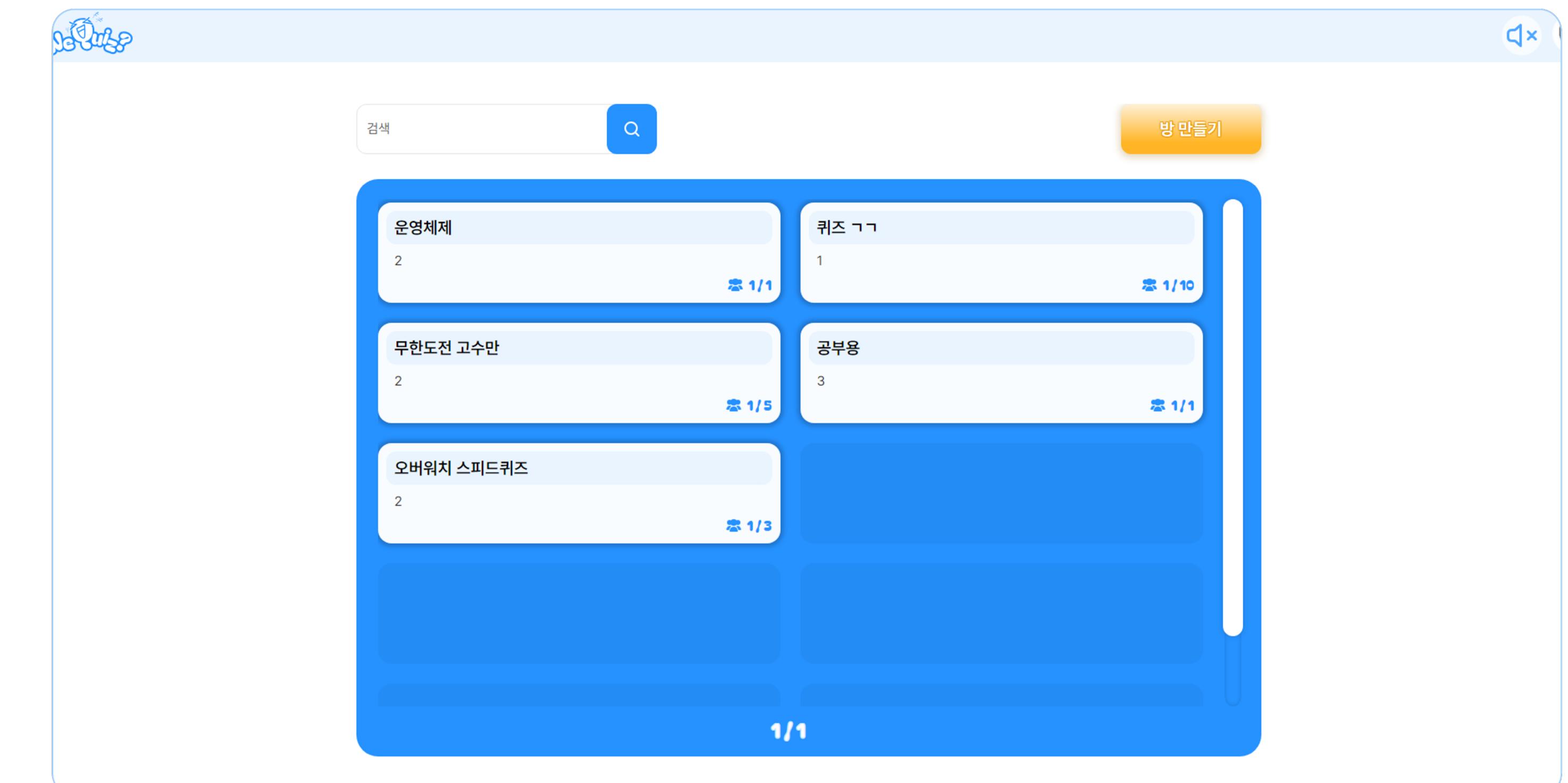


## 시작 페이지



## 로비

- 방 생성, 참여자 수 변경, 방 상태 변경  
의 이벤트가 발생하면 실시간으로 접속 가능한 방 표시



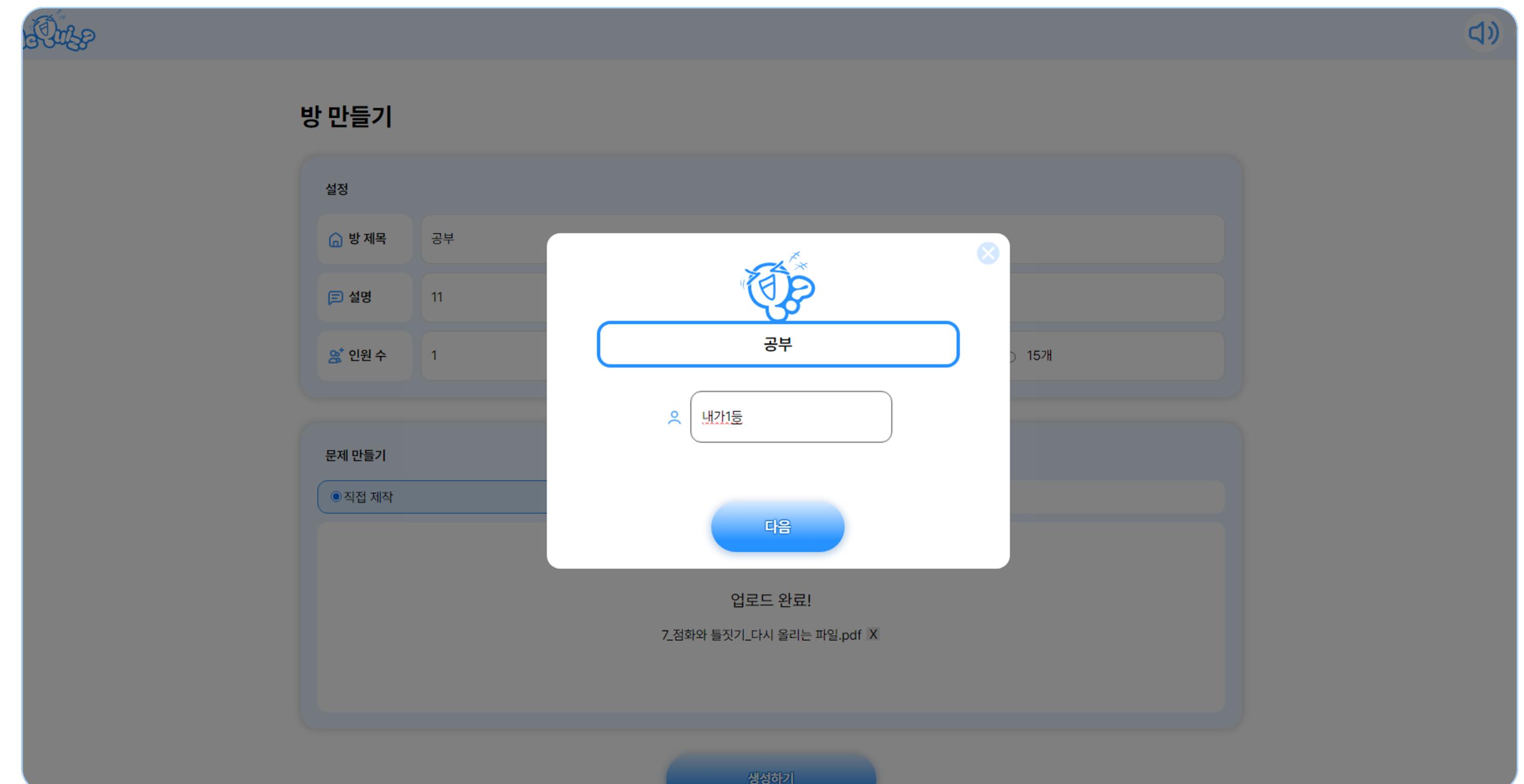
## 코드 입장

- 코드로 빠른 입장



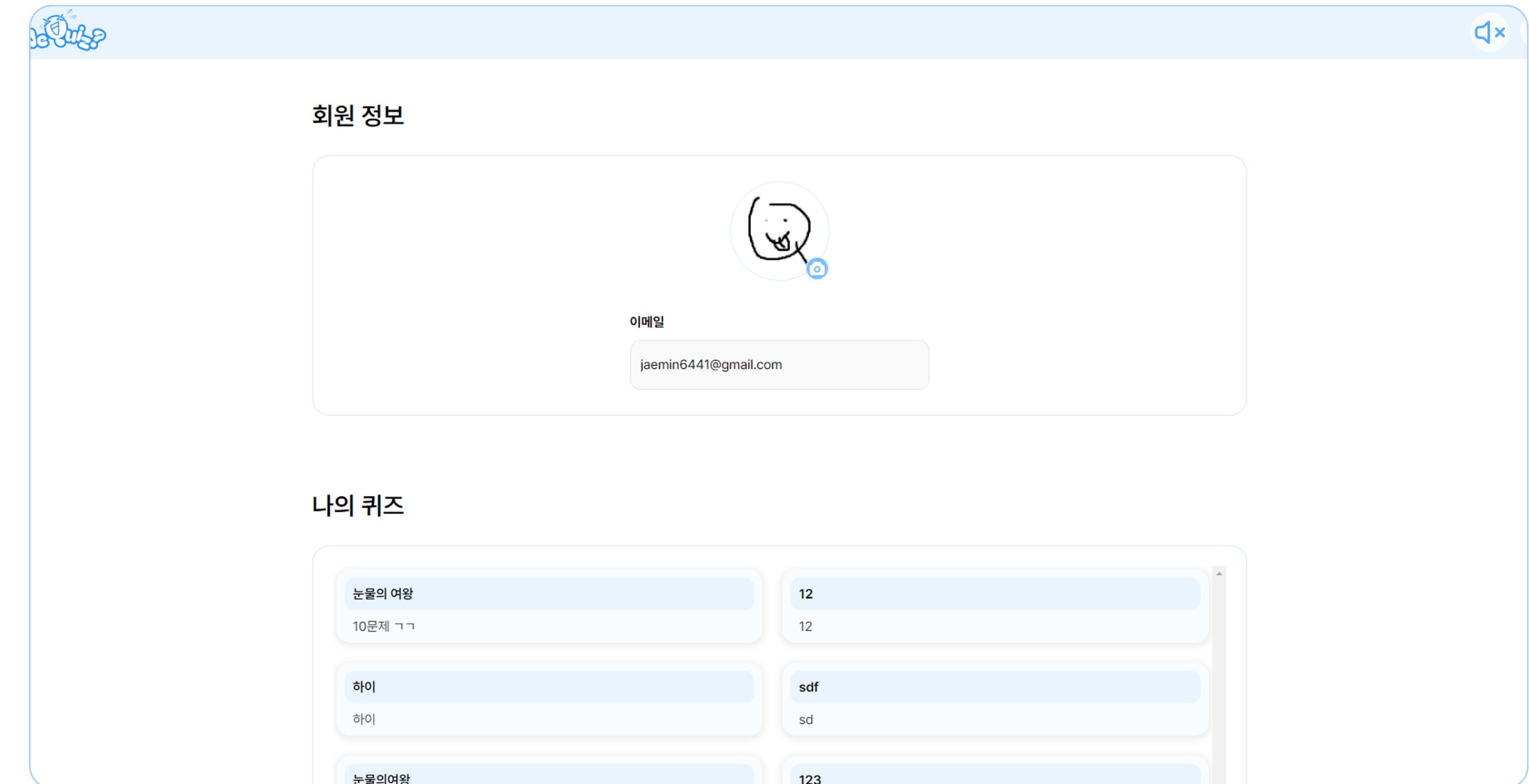
## 닉네임 설정

- 방마다 다른 닉네임 설정 가능



## 마이페이지

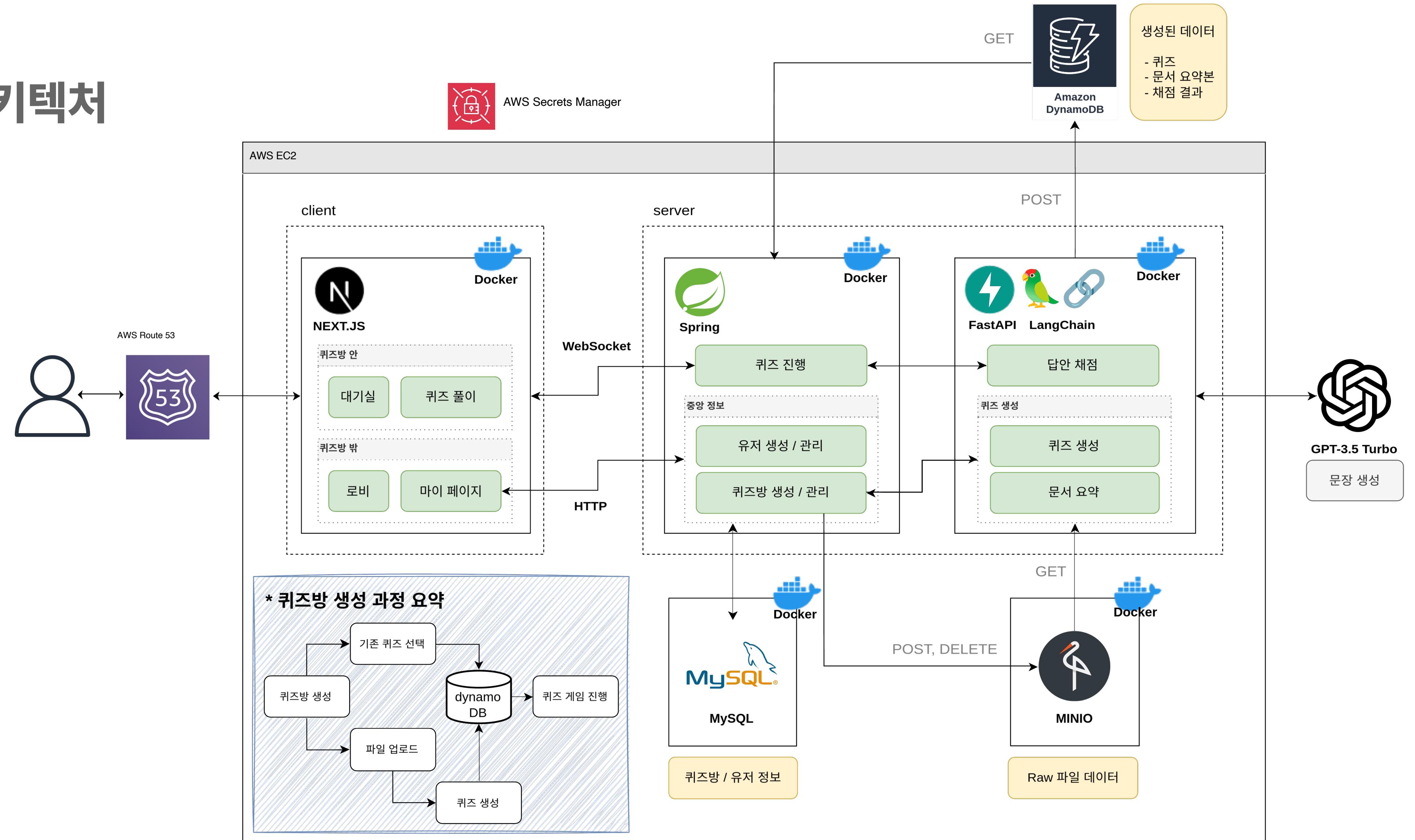
- 프로필 사진 변경
- 비밀번호 변경
- 참여했던 퀴즈 결과 보기



The mockup displays two main sections of the MyPage interface:

- 회원 정보 (Member Information):** This section shows a placeholder profile picture (a cartoon head) and an email address input field containing "jaemin6441@gmail.com".
- 나의 퀴즈 (My Quizzes):** This section lists completed quizzes with their titles, descriptions, scores, and participation counts:
  - 눈물의 여왕 (10문제 그 그) - Score: 12
  - 하이 (하이) - Score: sdf
  - 눈물의여왕 - Score: 123

## 시스템 아키텍처



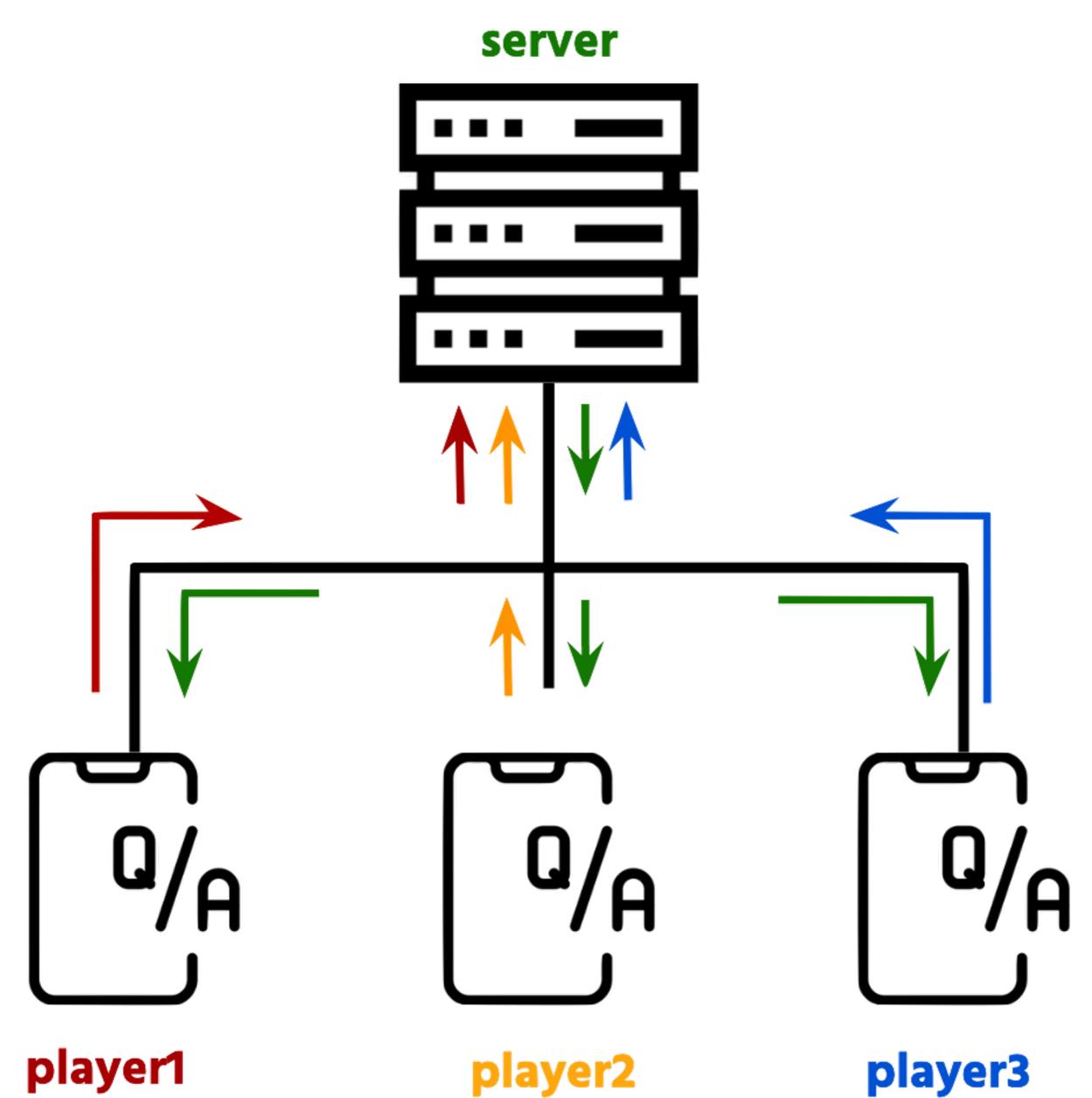
## 실시간 멀티플레이

게임 진행과 관련된 부분은 웹소켓을 사용하여 실시간 게임 구현

접속 가능한 방, 준비/제출 상태, 요약과 퀴즈,  
카운트다운, 점수 정보를 소켓으로 전달

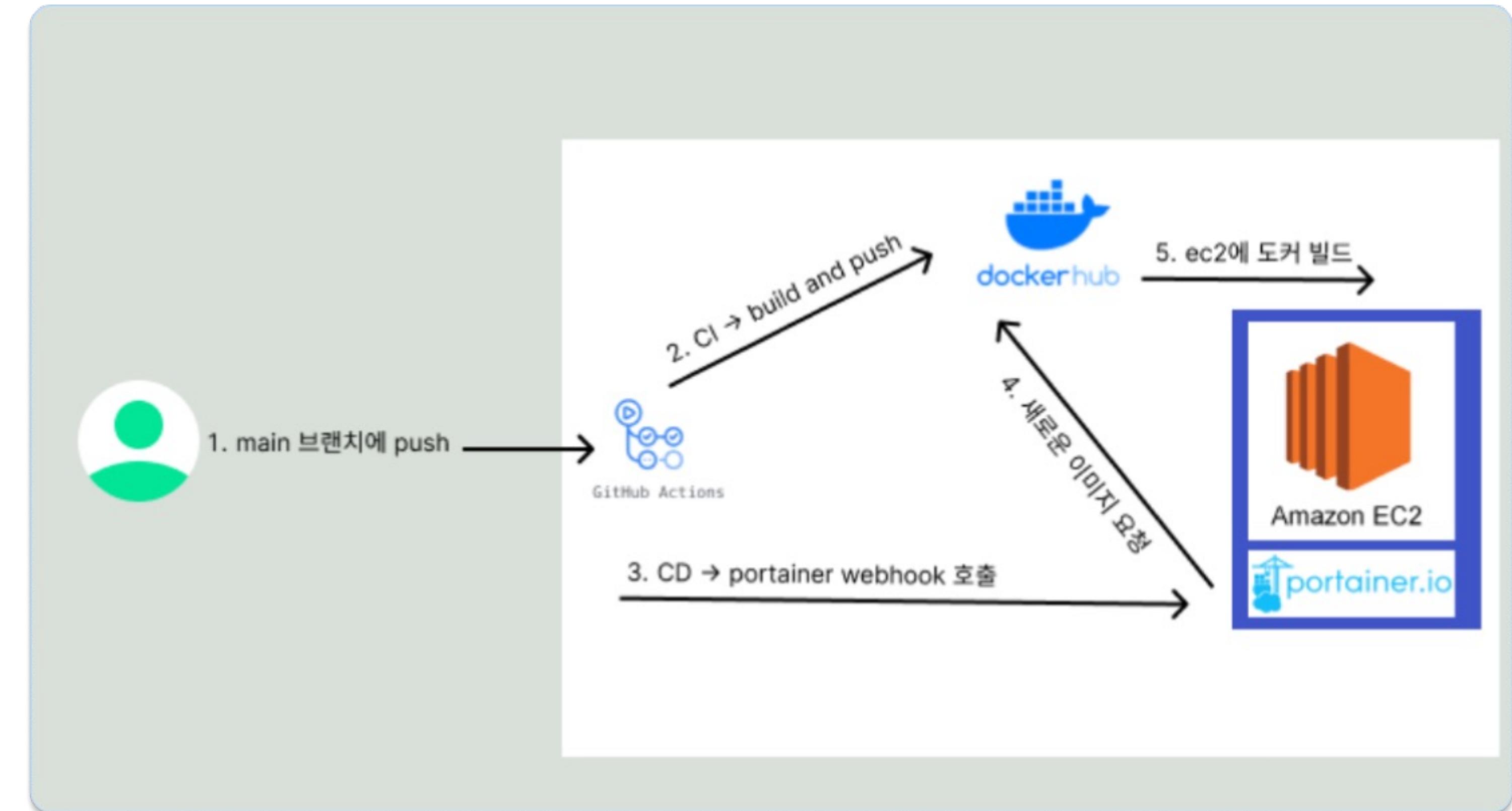
게임 진행 상태를 구분하여 소켓 재접속 시 상태에 따라 필요한 정보만 전송

연결이 일시적으로 끊기더라도 재접속 시 일관된 데이터를 제공함



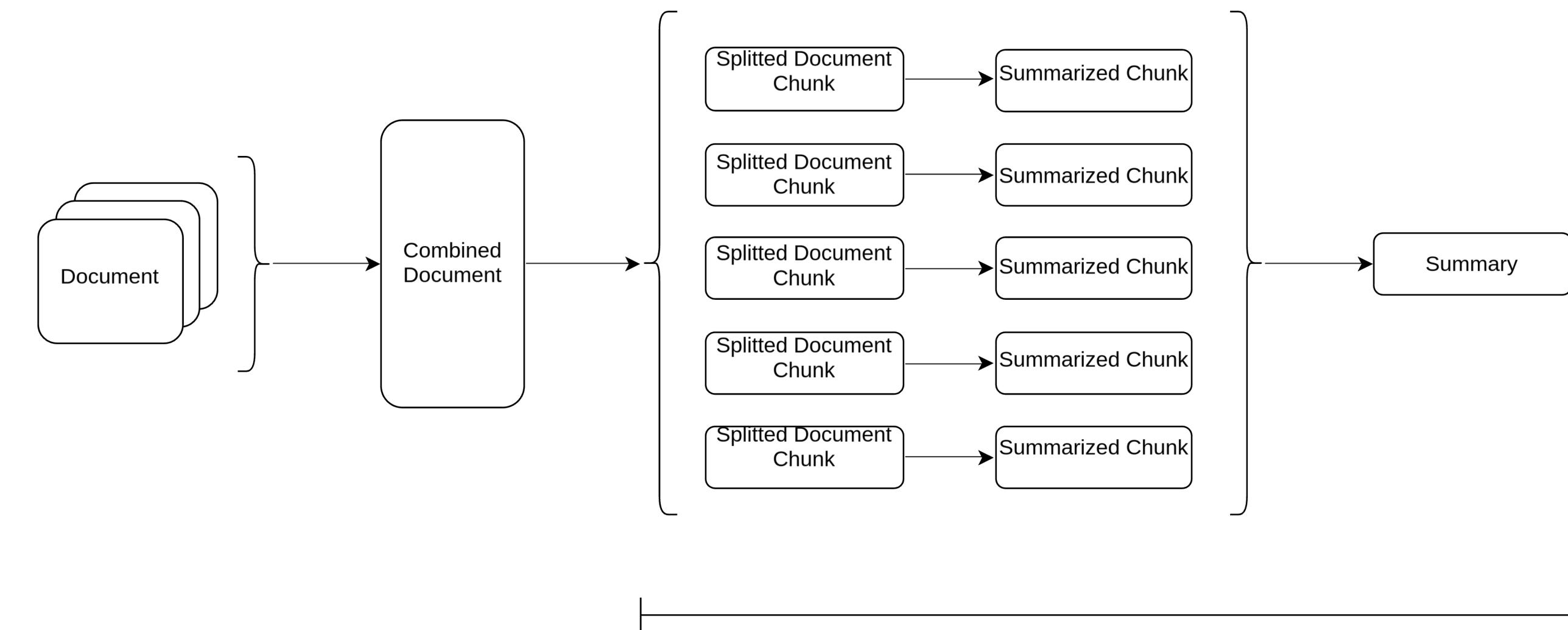
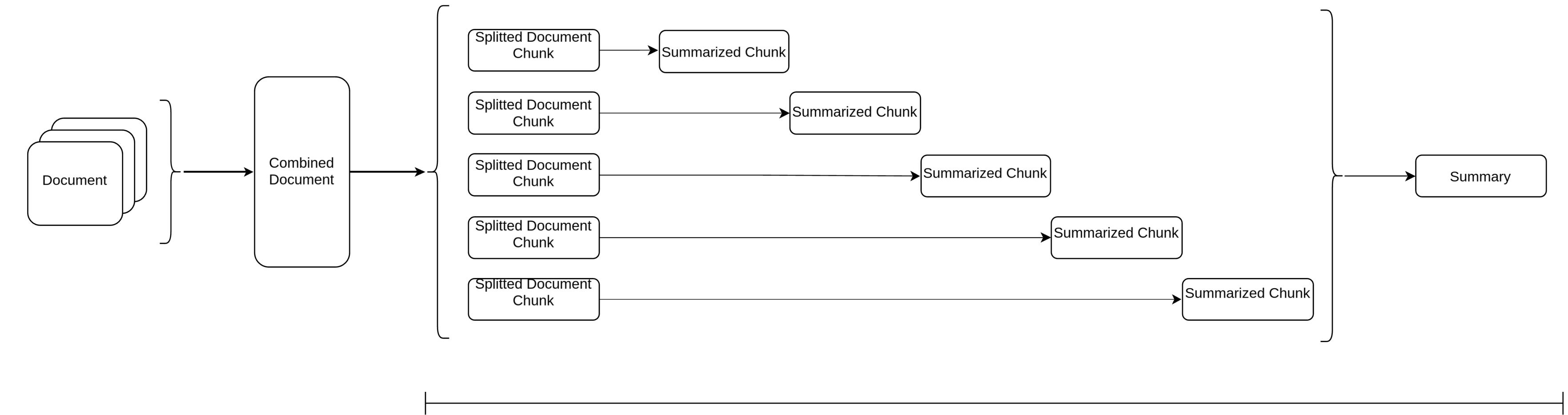
## CI/CD

- Git Actions와 Portainer를 통한 빌드 및 배포 자동화



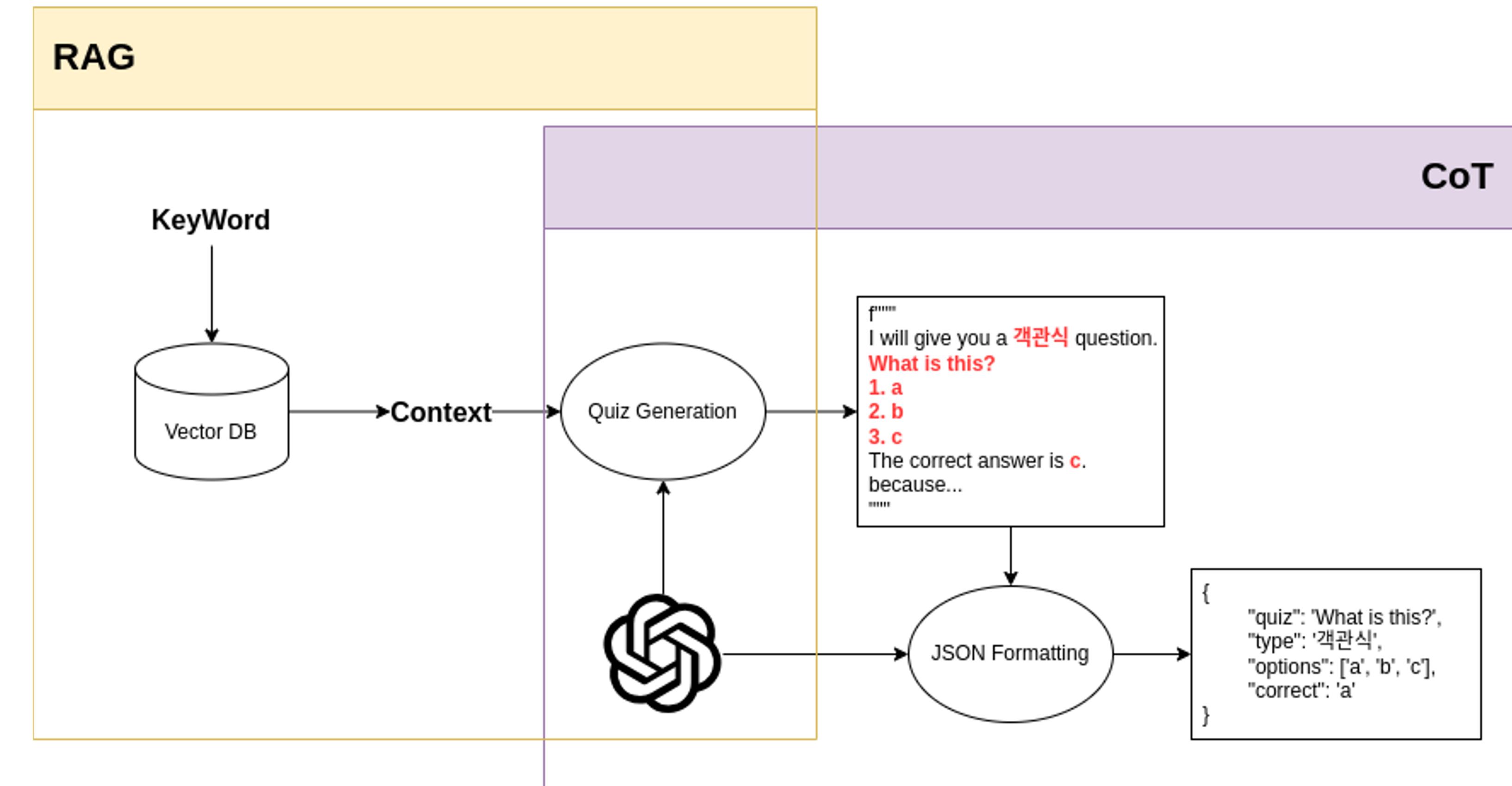
## 요약

- Map 쪼갠 문서 요약
- Reduce 합쳐서 최종 요약



## 퀴즈 생성

- RAG 키워드와 관련된 문서 조각을 검색해서 사용
- CoT 퀴즈 생성 태스크를 두 개의 sub task로 분리



## 요약과 문제 생성 방법

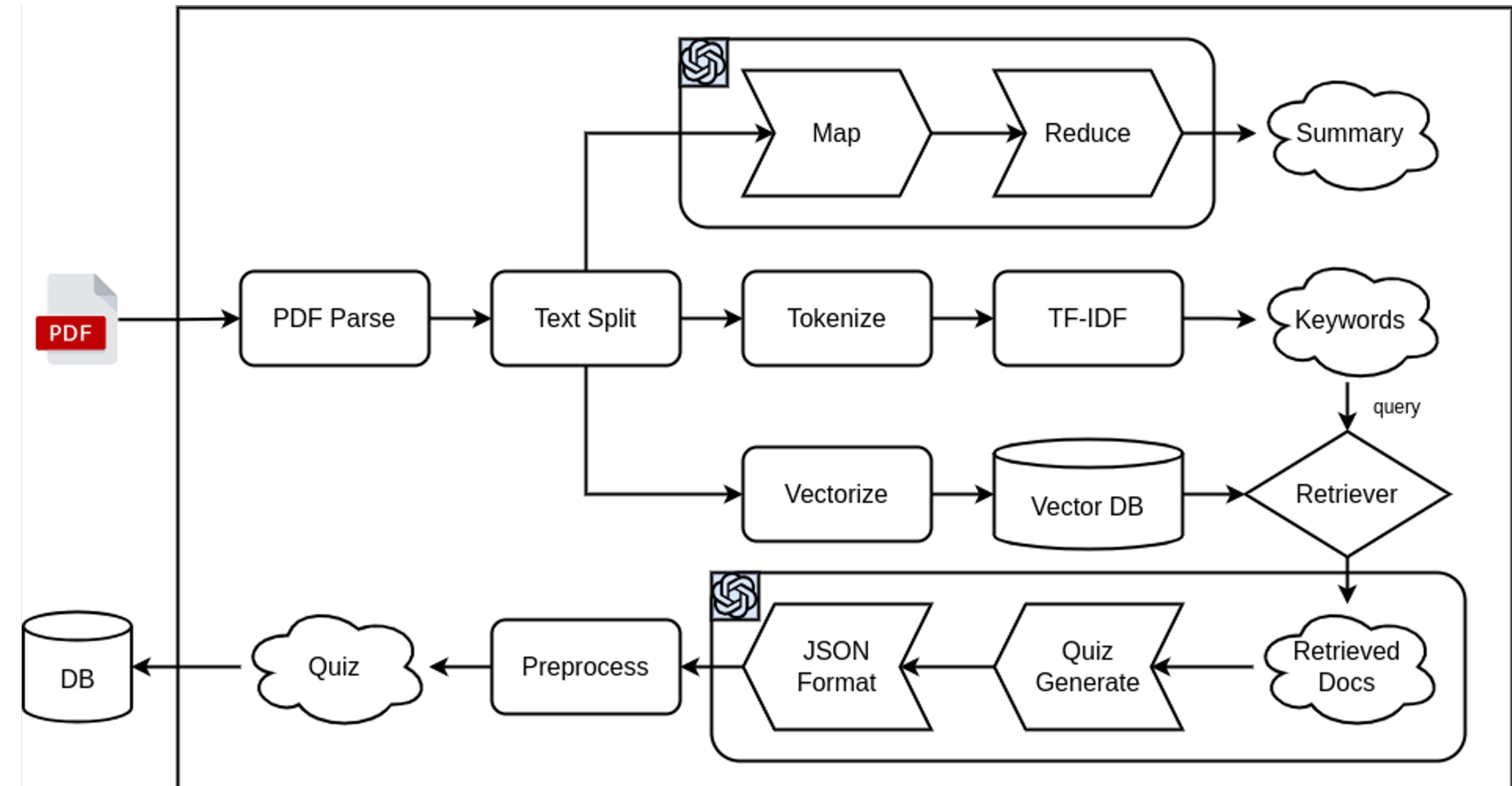
4개의 sub task 발생

1. 문서 요약

2. 키워드 추출

3. 문서 벡터화

4. 퀴즈 생성



방장도 없는데 게임 시작은 어느 타이밍에?



친구들, AI봇까지 모두 준비 완료하면!



내용이 없는 PDF를 넣으면 어떻게 되나요?



양질의 문제 제공을 위해,  
PDF에 포함된 토큰수가 너무 적으면 예외 발생!



많은 참여자가 많은 문제를 풀면  
최종성적표 나오는 시간이 오래 걸리지 않나요?



채점은 한문제마다 진행이 바로바로!  
5문제를 풀면 1라운드가 종료되고 중간 성적을 보여줘요!  
(로딩을 없애기 위해 채점할 시간을 약간 약간 벌어요...)



퀴즈 게임이 심심한데?



과반수 이상 답안 제출 시 카운트다운 시작!

퀴즈가 생성될 때까지 기다려야 하는구나...



대기실에서 친구들을 초대하고, 채팅을 나눠요.  
AI가 생성한 문서 요약을 읽으며 퀴즈 풀 준비를 해요.

갑자기 등장하는 긴 요약글이 당황스러워;;



ChatGPT가 답변을 하는 것처럼 streaming 형태로  
(당황스럽지 않게) 문서 요약 등장!

## 대기 없이 다음 문제 진행

3초 카운트다운이 끝나고 다음 문제로 넘어가기 직전에 미제출자 처리 후  
채점과 점수 계산까지 해야 하여 다음 문제 표시까지 시간 소요



카운트다운이 종료될 때 미제출자 처리만 하고 다음 문  
제로 넘어간 후 채점과 점수 계산을 하도록 변경

## 안정적인 퀴즈 Update

N개의 퀴즈를 생성하며 연속적으로 DynamoDB에 업데이트 할 때, 동시성 문제가 종종 발생함



퀴즈를 5개 생성될 때마다 업데이트 하여 부하를 줄이고,  
데이터에 버전을 추가하여 기대하는 버전에 맞을 경우 업데이트를 시도하는  
낙관적 락 방식을 적용하여 해결

## 언어를 고려한 문서 분석

영어, 한글 PDF를 모두 수용해야하는 상황에서  
언어에 따라 키워드 추출과 Text SPlit의 수준이 달라지는 문제



텍스트에 대해 먼저 언어 감지를 한 후에,  
언어에 따라 다른 토크나이저를 사용하도록 개선

### 맥락에 맞는 문제 형식 선택

문제의 형식(객관식, 단답형)을 직접 임의로 지정해주고 llm에 생성 요청을 했을 때,  
맥락이나 문항에 대한 문제 형식이 적절하지 않은 문제



문제의 형식을 제공된 context를 보고 결정하는 프롬프트 추가

문제 형식 지정의 주체를 ml 서버 -> LLM로 이전한 것

### 게임 진행 상태 구분

게임 진행 중 새로고침 등 소켓 재접속이 발생할 때  
클라이언트에서 게임 진행 상태에 대한 정보가 없어 실시간 데이터 요청 불가



게임 진행 상태를 QUIZ\_SOLVING, COUNTDOWN, RESULT  
세 가지로 구분하여 서버에서 상태에 맞는 데이터를 전송

## 게임 진행 상태 구분

Phase 1: 문제 풀이 상황

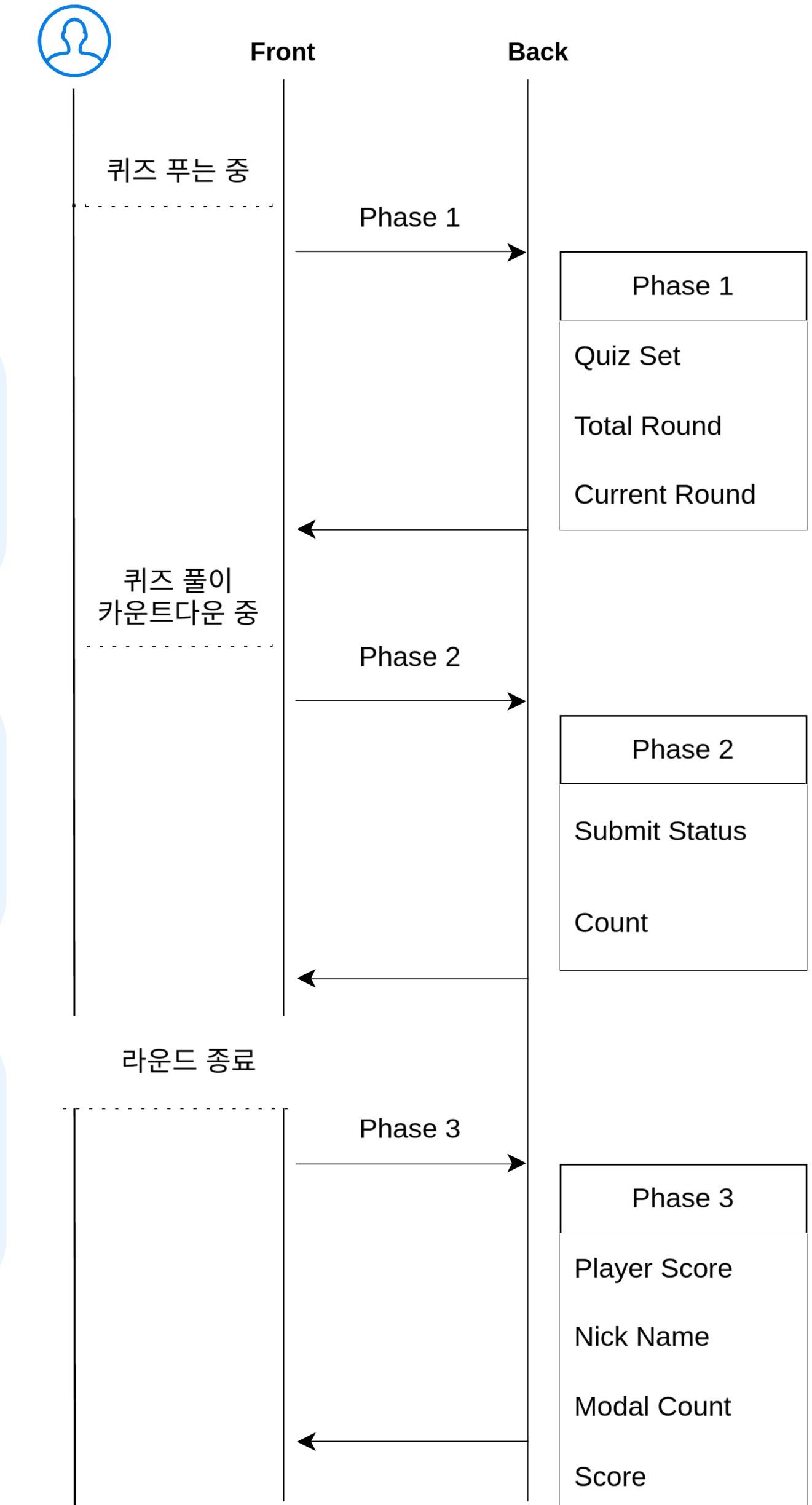
Phase 2: 카운트 다운 상황

Phase 3: 라운드 종료 상황

퀴즈 정보, 전체 라운드, 현재 라운드

Phase 1의 정보, 제출 상황, 카운트

중간 등수를 확인할 수 있는 정보,  
중간 등수 모달 카운트



## 꾸준히 유저를 유지할 방법

- 이전에 생성한 문제를 풀 수 있는 기능
- 다양한 문서로 문제를 만들기

## 백 서버와 ML 서버의 분리

- 퀴즈 생성 요청 후, 퀴즈 생성되는 동안 ML 서버와 소통하지 않고, 퀴즈를 DynamoDB에서 폴링
- ML 서버를 서비스 흐름과 분리하여 퀴즈 생성 모듈로써 활용

## 방별 퀴즈 관리

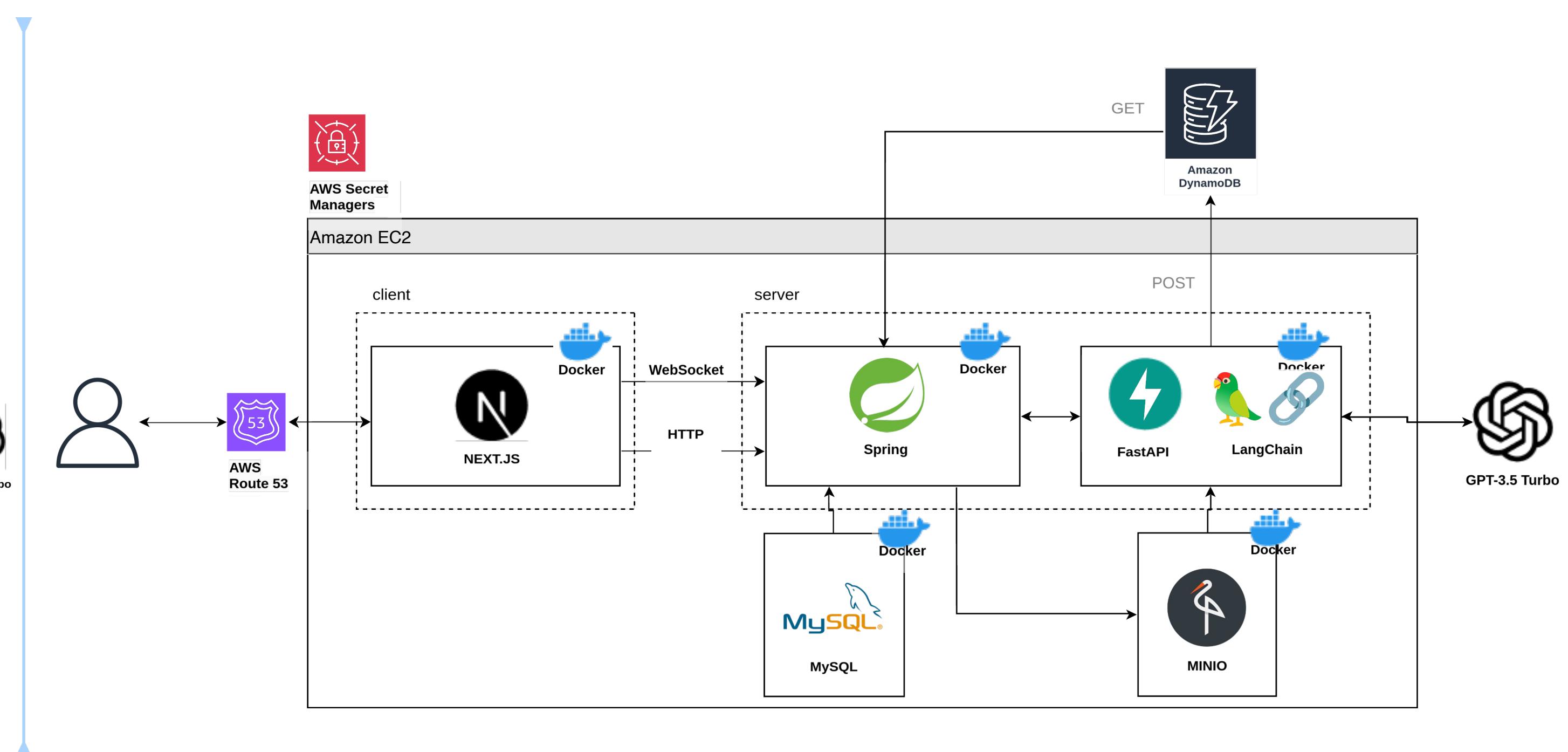
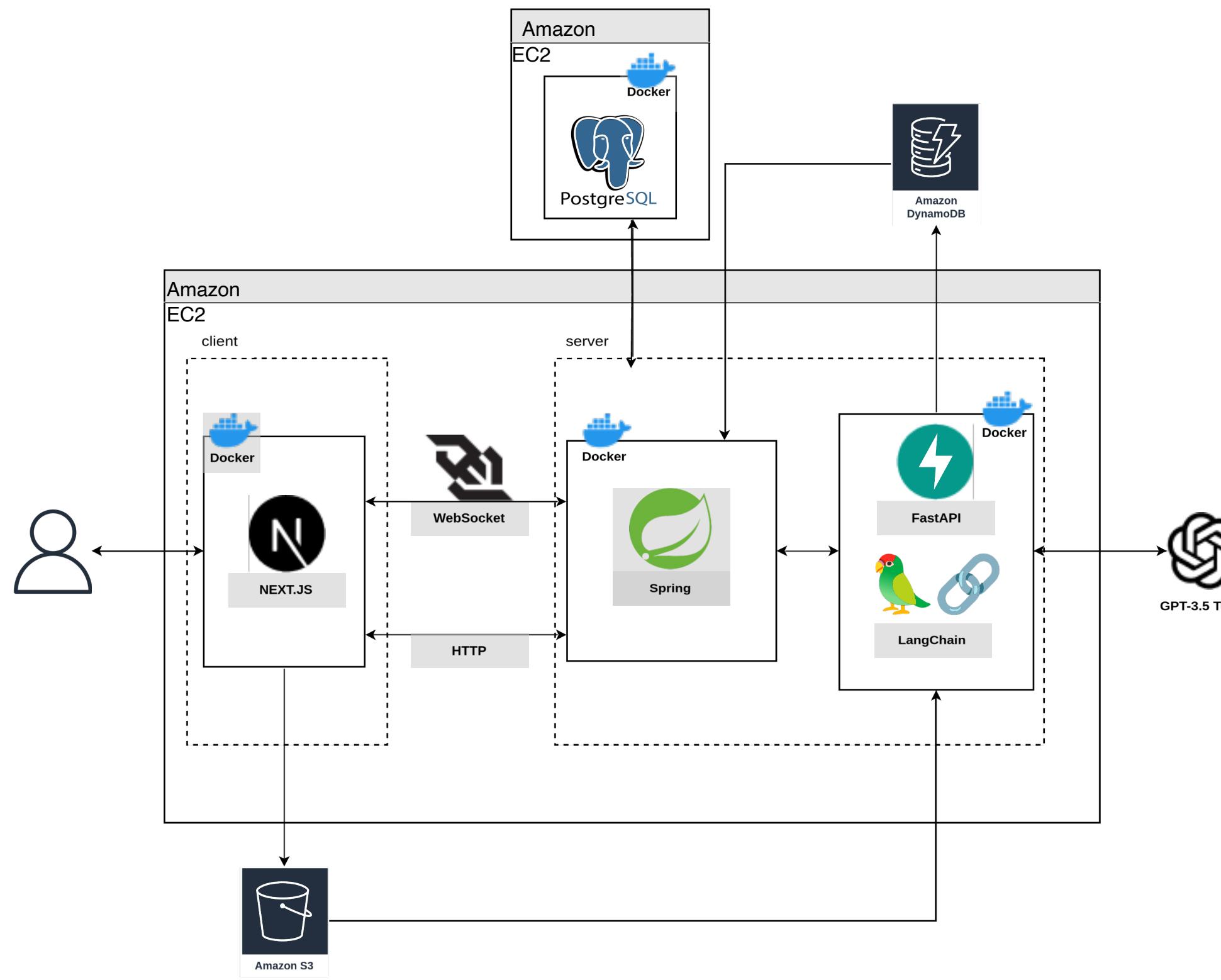
- 의존도를 높이지 않고 인메모리에서 관리
- 한 문제씩 전송하기 위해 퀴즈 큐를 만들되 메모리 관리를 위해 5문제씩 저장

## 생성 시간 단축

- 요약을 생성하는 함수와 퀴즈 생성 함수를 비동기로 구현

# 시스템 아키텍처 개선

- 네트워크 통신 비용 감소를 위해 S3 -> EC2 내에 MinIO 서버 구성
- 무거운 PostgreSQL에서 가볍고 상황에 더 적합한 MySQL로 변경
- 스토리지 접근 권한을 서버로 제한



## 향후 계획

- ▶ 베타 서비스 출시 예정
- ▶ 서버가 트래픽에 따라 동적으로 스케일링 될 수 있도록 인프라 구축
- ▶ 사용자 의견을 받을 수 있는 간단한 설문 API 추가
- ▶ 유저 유입, 행동 및 설문 모니터링 및 분석
- ▶ GPU 서버를 추가하여 이미지 내 텍스트 처리 기능 ON
- ▶ QuizSet 랭킹 페이지 생성 -> 만든이와 상관 없이 누구나 좋은 퀴즈를 재사용할 수 있도록

## 협업 툴



### Notion

회의록, 진행 상황 공유  
전체 일정 관리



### Github

Issue, PR, Projects



### Swagger

API 명세



### Discord

주간 회의, 실시간 소통  
Github 웹훅 연결

## 협업 방식

매주 새로 배운 내용이나 트러블 슈팅 사항을 팀원들 간 공유

### 내용

no.	참여자	팀원별 활동 내용	공부 내용 url	소감
1	박규연 (20203065)	LLM inference 비동기 처리	<a href="#">Tistory</a>	비동기처리로 이렇게 속도가 향상될 수 있다니!!! 다행이다...
2	심재민 (20203093)	java syncronized	<a href="#">Velog</a>	운영체제를 나름 열심히 들은 것 같은데도 스레드는 참 어려운 것 같다...
3	배준형 (20152829)	CSS 기술	<a href="#">Notion</a>	CSS와 HTML 면접 대비 겸 공부를 해보았다..
4	김우림 (20190068)	소켓으로 페이지네이션 구현하기	<a href="#">Tistory</a>	페이지네이터 컴포넌트를 직접 만들고 웹소켓을 이용한 페이지네이션으로 실시간 게임방 리스트를 불러와보았다!
5	안금장 (20181637)	Collection Interface	<a href="#">Tistory</a>	자바에서 많이 쓰이는 Collection을 직접 뜯어보며 자바의 자료구조에 대해서 공부했다!

## 협업 방식

철저한 GitHub 컨벤션과 이슈 단위의 브랜치 를 준수

[feat/#478] 🐛 fix: 결과표 스타일 수정	ryanbae94 committed 5 days ago
[Feat/#455] 🐛 Fix: 과반수 계산 방식 수정	cherry031 committed 5 days ago
[feat/#475] ✨ Feat: 문장단위 TextSplitter 제작	noooey committed 5 days ago

feat/#482



feat/#354



feat/#465



feat/#475



feat/#478



feat/#476



## 협업 방식

### 상세한 Issue , Pull request 작성 및 Merge 전 상호 코드 리뷰

[feat/#311] ✨ Feat: 기존 문제 가져오는 api #424

Merged cherry031 merged 16 commits into main from feat/#311 5 days ago

Conversation 3 Commits 16 Checks 1 Files changed 18

cherry031 commented 2 weeks ago • edited

Member ...

#### 개요

- [BE] 기존 문제 가져오는 api #311

#### 작업사항

- DynamoDB에 user\_id로 글로벌 보조 인덱스를 생성하여 user\_id를 기준으로 검색하여 본인이 만든 퀴즈 리스트를 가져옵니다.
- "/rooms/existQuiz" 로 Get 요청하면 기존 퀴즈 리스트 가져오기
- DTO: `List<String> existQuizIdList` -> api 명세 참고
- CreateRoom Post 요청 DTO에 `String existQuizDocId` 추가했습니다. 기존 퀴즈면 문서id 담아서 보내주고 아니면 null로 보내주세요
- +++ 추가)
- 기존 문제 가져올 때 그 문서에 만들어져 있는 퀴즈 개수를 가져와서 방 문제 개수로 설정했습니다. 기존 문제 선택할 때는 클라에서 문제 개수 선택 막아주세요 [@Krimwoo](#) [@ryanbae94](#)

#### 스크린샷(선택)

koomchang reviewed 2 weeks ago

chatty-be/src/main/java/com/chatty/chatty/quizroom/controller/QuizRoomController.java

```

...
...
@@ -42,6 +43,11 @@ public ResponseEntity<RoomIdResponse> create(@Valid @RequestBody RoomIdRequest
    ...
    return ResponseEntity.status(HttpStatus.CREATED).body(roomIdResponse);
}
+
@GetMapping("/create")

```

koomchang 2 weeks ago

/existing 등이 나을 것 같아요! /create 는PostMapping에 어울리는 url 같아요

Reply...

Resolve conversation

```
const protectedPatterns = [
  '/main-lobby',
  '/create-room',
  '/waiting-room/:id'
].map(path => new URLPattern({ pathname: path }));

const publicRoutes = ['/sign-in', '/sign-up'];

export function middleware(request: NextRequest) {
  const token = request.cookies.get('refreshToken');
  const currentPath = request.nextUrl.pathname;

  const isProtectedRoute = protectedPatterns.some((pattern) =>
    pattern.test(request.nextUrl),
  );

  if (!token && isProtectedRoute) {
    const url = request.nextUrl.clone();
    url.pathname = '/sign-in';
    url.searchParams.set('message', '로그인에 필요한 페이지입니다.');
    return NextResponse.redirect(url);
  }

  if (token && publicRoutes.includes(currentPath)) {
    const url = request.nextUrl.clone();
    url.pathname = '/main-lobby';
    url.searchParams.set('message', '이미 로그인된 상태입니다.');
    return NextResponse.redirect(url);
  }

  const response = NextResponse.next();
  response.headers.set('Content-Security-Policy', 'upgrade-insecure-requests');

  return response;
}
```

## 라우팅 가드 구현

- 클라이언트 사이드에서 로그인 여부 판단 시 빈 컵데기 렌더링, 이로 인한 사용자의 불편함, 불필요한 api 호출의 문제가 발생
- Next.js의 middleware를 활용하여 서버사이드에서 로그인 여부 판단 후 페이지 접근을 제한

- LLM의 input으로 넣어줄 수 있는 token 수가 제한적
- 많은 내용을 input으로 한 번에 넣으면 LLM이 맥락을 충분히 파악하지 못함
- 문서요약, 키워드추출, 벡터디비 생성 각각 목적에 따라 LLM이 input을 이해해야 하는 정도가 다름  
-> 용도에 따른 input 문서 양 최적화 필요 (비용, 시간 최적화)
- LangChain의 TextSplitter 모듈은 chunk사이즈는 맞춰주나, 문장 자체를 보존하지 않음.

## Custom TextSplitter 구현

- 텍스트를 문장 단위 분리  
(언어에 따라 다른 토크나이저 사용)
- 지정된 chunk size 이내로 문장을 모아 chunk 생성
- 지정된 overlap 만큼 chunk 간의 문장 중복 허용

## DB 구조 - DynamoDB

퀴즈 문서						
	문제번호	id	Type	NOT NULL	Default value	Comment
Y	요약	description	String	NOT NULL	Default value	Comment
	문제 개수	num_of_quiz	Integer	NOT NULL	Default value	Comment
	생성 시간	timestamp	String	NOT NULL	Default value	Comment
	만든유저id	user_id	Long	NOT NULL	Default value	Comment
	버전	version	Integer	NOT NULL	Default value	Comment
	문제들	questions	List<Quiz>	NOT NULL	Default value	Comment

Quiz						
	문제 id	id	Type	NOT NULL	Default value	Comment
	정답	correct	String	NOT NULL	Default value	Comment
	문제	question	String	NOT NULL	Default value	Comment
	문제 번호	question_number	Integer	NOT NULL	Default value	Comment
	유형	type	String	NOT NULL	Default value	Comment
	선택지	options	List<String>	NOT NULL	Default value	Comment

채점 문서						
	문서 id	id	Type	NOT NULL	Default value	Comment
Y	생성 시간	timestamp	String	NOT NULL	Default value	Comment
	채점 결과들	answers	List<Answer>	NOT NULL	Default value	Comment

Answer						
	문제 id	id	Type	NOT NULL	Default value	Comment
	정답	correct_answer	String	NOT NULL	Default value	Comment
	문제 번호	quiz_num	Integer	NOT NULL	Default value	Comment
	채점 결과	markeds	List<Mark>	NOT NULL	Default value	Comment

Mark						
	정답 여부	marking	Type	NOT NULL	Default value	Comment
	유저가쓴 답	user_answer	String	NOT NULL	Default value	Comment
	유저 id	user_id	Long	NOT NULL	Default value	Comment

## DB 구조 - MySQL

