



4조 미니 프로젝트 : '베질걸: 99' S.A



프로젝트 기본 설명

- 프로젝트 이름 : 베질걸: 99
- 프로젝트 설명 : TMI 퀴즈 게시판
 - 항해99 14기 인원들이 본인의 TMI를 퀴즈로 만들어서 등록하고, 다른 사람의 퀴즈를 맞춰보는 웹 애플리케이션입니다.
- 주요 기능
 - 회원 가입, 로그인
 - 퀴즈 등록
 - 객관식이나 다지선다의 형태로 퀴즈 방식을 골라 등록하도록 만들기
 - 등록할 때 답변과 정답이 아닌 객관식 답들도 같이 입력하도록 만들기
 - 회원인 사람만 로그인 후에 퀴즈를 등록할 수 있도록 만들기
 - 등록된 퀴즈 게시판 조회
 - (욕심) 로그인한 사용자에게 한해서, 로그인한 사용자가 맞춘 문제(퀴즈 게시글)에 이미 푼 문제라는 표시 남기기, 또는 보이지 않게 하기
 - 댓글 추가
 - (욕심) 대댓글 진짜 욕심
 - 문제 풀기
 - 다른 사람이 만든 게시글을 조회하면 답안을 선택, 또는 문자열을 입력해서 제출 → 정답 여부를 alert로 알려주기
 - (욕심) 한번에 풀면 추가 점수
 - (욕심) 점수 랭킹 페이지
 - 정답을 맞췄을 때 얻는 점수를 기준으로 점수가 높은 순으로 랭킹을 보여주도록 만들기 (👁️1위는 항해 인싸 타이틀 증정)
 - (욕심) 마이 페이지
 - 내가 맞춘 문제들을 볼 수 있도록 만들기..?
 - 내가 만든 문제들을 볼 수 있도록 만들기

프로젝트 참여 명단

Aa 이름	🔗 Github	🔗 Blog	@ 이메일
제목 없음			

추가 소통 공간 (디스코드) : <https://discord.gg/3QvEgAp9>

역할 분담

- 백엔드
 - 회원가입, 로그인 : 김은서, 송우근님
 - CRUD , 퀴즈 문제풀이, 답안 순서랜덤생성: 강동현님, 최하나님
- 프론트엔드
 - 로그인
 - 회원가입
 - 퀴즈 목록 조회
 - 퀴즈 상세 조회
 - 퀴즈 등록
 - 퀴즈 수정
 - 댓글 등록
 - 댓글 삭제
 -

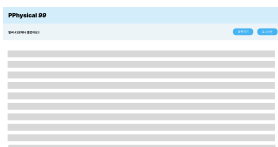
와이어프레임



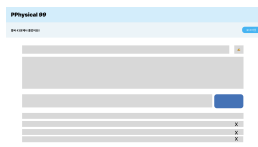
로그인



회원가입



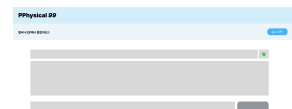
퀴즈 목록 조회



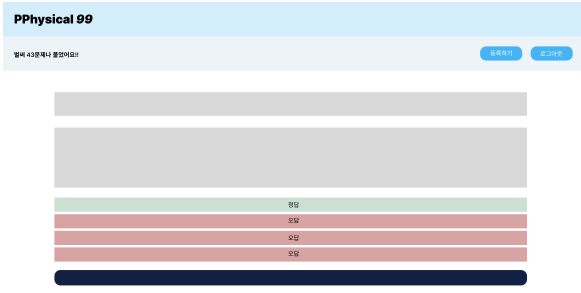
퀴즈 상세조회 (주관식)



퀴즈 수정 (객관식)



퀴즈 수정 (주관식)

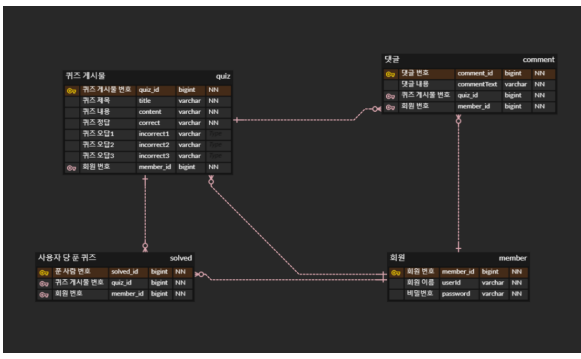


퀴즈 등록



퀴즈 상세조회 - 이미 푼 문제

ERD



API

▼ API 펼쳐보기

API

※ 상태	Aa 데 이 터 테 이 블	≡ 기 능	≡ METHOD	≡ URL	≡ token	≡ request	≡ response	≡ 담당
------	----------------------------------	-------------	-------------	----------	------------	--------------	---------------	---------

※ 상태	Aa 데 이 터 테 이 블	≡ 기 능	≡ METHOD	≡ URL	≡ token	≡ request	≡ response	≡ 담당
Done	메 인 페 이 지	퀴 즈 리 스 트	GET	/quiz	Bearer {token}	-	{ "success": true, "message": "전체 퀴즈 조회 성공!", "data": { ... {quiz data} ... } }	최하나, 강동현
Done	회 원 전 보	회 원 가 입	POST	/signup	-	{ "userId": "userId", "password": "password", }	{ "success": true, "message": "회원 가입 완료!", "data": null }	김은서, 송우근
Done	제 목 중 복 여 기	회 원 가 입 시 아 이 디 중 복 확 인	POST	/signup/valid	-	{ "userId": "userId" }	{ "success": true, "message": "사용 가능 한 아이디입니다.", "data": null } OR { "success": false, "message": "아이디 중 복", "data": null }	

※ 상태	Aa 데 이 터 테 이 블	≡ 기 능	≡ METHOD	≡ URL	≡ token	≡ request	≡ response	≡ 담당
Done	제 목 영 문	로 그 인	POST	/login	-	{ "userId": "userId", "password": "password", }	{ "success": true, "message": "로그인 성 공", "data": null } OR { "success": false, "message": "비밀번호 가 일치하지 않습니다.", "data": null } OR { // 회 원가입 하지 않은 사람 이 로그인할 때 "success": false, "message": "존재하지 않는 회원입니다.", "data": null }	김은서, 송우근
Done	퀴 즈 게 시 판	퀴 즈 게 시 글 등 록 (객 관 식 (4 지 선 다), 주 관 식, OX)	POST	/quiz/register	Bearer {token}	{ "title": "title", "content": "content", "correct": "correct", "incorrect1": "incorrect1", "incorrect2": "incorrect2", "incorrect3": "incorrect3" }	{ "success": true, "message": "퀴즈 등록 성공!", "data": quizId }	최하나, 강동현 → 객관 식, OX 모두 구현 — 식(4지선다) 은 correct, incorrect, inc 이 모두 값이 들어가야 문제는 correct, incorr 나머지는 null 값으로

☼ 상태	Aa 데 이 터 테 이 블	≡ 기 능	≡ METHOD	≡ URL	≡ token	≡ request	≡ response	≡ 담당
Done	제 백 영 이	퀴 즈 게 시 글 상 세 조 회	GET	/quiz/{quiz-id}	Bearer {token}	-	<pre>{ // 처음 문제를 조회할 때 "success": true, "message": "도전하지 않은 문제입니다.", "data": { "id": "quizId", "userId": "userId", "title": "quizTitle", "content": "quizContent", "answerList": ["correct", "", "", ""], "solved": true (or false), "commentList": [..., { "id":integer, "userId":"userId", "comment":"comment" }, ...] } } OR { // 틀렸 던 문제를 조회할 때 "success": true, "message": "히히 틀렸 음.", "data": { "id": "quizId", "userId": "userId", "title": "quizTitle", "content": "quizContent", "answerList": ["correct", "", "", ""], "solved": true (or false), "commentList": [..., { "id":integer, "userId":"userId", "comment":"comment" }, ...] } } OR { // 맞춘 문제를 조회할 때 "success": true, "message": "이미 맞춘 문제입니다.", "data": { "id": "quizId", "userId": "userId", "title": "quizTitle", "content": "quizContent", "answerList": ["correct", "", "", ""], "solved": true (or false), "commentList": [..., { "id":integer, "userId":"userId", "comment":"comment" }, ...] } }</pre>	최하나, 강동현

☼ 상태	Aa 데 이 터 테 이 블	≡ 기 능	≡ METHOD	≡ URL	≡ token	≡ request	≡ response	≡ 담당
Done	제목 없음	퀴즈 게시 글 수 정하 기	PUT	/quiz/{quiz-id}	Bearer {token}	{ "title": "quizTitle", "quizContent": "quizContent" }	{ "success": true, "message": "퀴즈를 수 정하였습니다.", "data": null } OR { // 퀴즈 게시 글이 존재하지 않을 때 "success": false, "message": "해당 퀴즈 가 존재하지 않습니다.", "data": null } OR { // 작 성하지 않은 사람이 수 정하려 할 때 "success": false, "message": "다른 회원 이 작성한 퀴즈입니다.", "data": null }	최하나, 강동현 → 답안 구성완료
Done	제목 영역	퀴즈 게시 글 삭 제하 기	DELETE	/quiz/{quiz-id}	Bearer {token}	-	{ "success": true, "message": "퀴즈가 삭 제되었습니다.", "data": null } OR { // 퀴즈 게시 글이 존재하지 않을 때 "success": false, "message": "해당 퀴즈 가 존재하지 않습니다.", "data": null } OR { // 작 성하지 않은 사람이 삭 제하려 할 때 "success": false, "message": "아이디가 같지 않습니다!", "data": null }	최하나, 강동현
Done	퀴즈 답안 출하 기	퀴즈 답안 제 출하 기	POST	/quiz/{quiz_id}/solving	Bearer {token}	{ "correct": "correct" }	{ "success": true, "message": "정답입니 다~!", "data": null } OR { "success": true, "message": "틀렸습니 다!", "data": null } OR { "success": false, "message": "해당 퀴즈 가 존재하지 않습니다.", "data": null }	최하나, 강동현
Done	댓글	댓글 작성 하 기	POST	/comment/{quiz-id}	Bearer {token}	{ "comment" : "sadasdasdasd" }	{ "success": true, "message": "댓글을 등 록했습니다.", "data": null } OR { "success": false, "message": "해 당 퀴즈가 존재하지 않 습니다.", "data": null }	

※ 상태	Aa 데 이 터 테 이 블	≡ 기 능	≡ METHOD	≡ URL	≡ token	≡ request	≡ response	≡ 담당
Not started	제목 없음	댓글 수정하기 - 보류	PUT	/comment/{comment-id}	Bearer {token}	{ "comment" : "newComment" }	아직 추가 하지 않음	
Done	제목 없음	댓글 삭제하기	DELETE	/comment/{comment-id}	Bearer {token}	-	{ "success": true, "message": "해당 댓글을 삭제했습니다.", "data": null } OR { "success": false, "message": "해당 댓글이 존재하지 않습니다.", "data": null }	
Not started	(음심) 대 댓글							
Not started	(음심) 퀴즈 좋아요	퀴즈 좋아요	PUT	/quiz/like/{id}				
Not started	(음심) 사용자 별 점수							
Not started	제목 없음							
Not started	제목 없음							
In progress	(음심) 만의 페이지	블러오기	GET	/api/user				최하나, 강동현
Not started	제목 없음							

공통 규칙

1. merge후 채팅에 기록 남기기
2. 커밋 메시지 양식 맞추기

▼ 커밋 유형 참고


```
// [커밋유형] 커밋 이름 : 커밋한 사람 이니셜  
[FEAT] 로그인 기능 구현 완료 : HKD
```

- feat : 새로운 기능에 대한 커밋
- fix : 버그 수정에 대한 커밋
- build : 빌드 관련 파일 수정에 대한 커밋
- chore : 그 외 자잘한 수정에 대한 커밋
- ci : CI관련 설정 수정에 대한 커밋
- docs : 문서 수정에 대한 커밋
- style : 코드 스타일 혹은 포맷 등에 관한 커밋
- refactor : 코드 리팩토링에 대한 커밋
- test : 테스트 코드 수정에 대한 커밋

3. 어려운 디버깅 다같이 하자!
4. 우리 미워하지말아요
5. 브랜치명 이니셜로 따서 진행하기 > SJY, LDE, KES, SUK, KDH, CHN

▼ 코딩 작성 규칙


- 변수, 함수명 작성
 - 카멜 케이스 사용

 ex) camelCase

- 함수명 작성
 - 동사+명사 형태로 작성

 ex) getUserInformation()

- 클래스명 작성
 - 파스칼 케이스 사용

 ex) CamelCase

- Flag로 사용되는 변수
 - Boolean 변수 작성: 조동사+flag 종류로 작성

ex) isNum, hasNum

- 약칭

- 약칭 사용하지 않기 (부득이하게 필요한 경우 상의하기)

```
val idx //bad
val index //good

val cnt //bad
val count //good
```

- 주식 규칙

- 한 줄은 // 사용
- 여러 줄 주석은 /* */ 사용

```
/* 한 줄 주석
   */
/* 여러 줄
   * 주석인 경우
   */
```

- magic Number

- magic Number: 코드에서 숫자, 문자열을 직접 사용하는 것을 지양
- 따로 constant 파일을 생성하여 관리
- 변수명은 대문자로 작성

```
/*constant파일
val LEFT = 'left';

//다른 파일
if (dir === LEFT) {
  //...
}
```

6. 제발제발제발!!!! 회의 중에 내용들 회의록에 적어요!!

🕒 일간 계획

- 🌄 09:00 - 09:30 아침 조례
- 🧑‍💻 09:30 - 12:30 아침 코딩
- 🍱 12:30 - 13:30 점심 시간
- 💻 13:30 - 16:00 오후 코딩 (1차)
- 🗓️ 16:00 - 16:30 정기 회의
- 🧑‍💻 16:30 - 18:30 오후 코딩 (2차)

- 🕒 18:30 - 19:30 저녁 시간
- 🧑‍💻 19:30 - 21:30 저녁 코딩
- 📝 21:00 - 22:00 저녁 종례
- 🖥️ 22:00 - 야간 자율 코딩

📅 주간 계획 및 회의록

▼ 05/05 (금)

▼ 공통

- 개발 기획
- 와이어 프레임 작성
- API 작성
- DB Schema 작성
- Github Repository 만들거
- ERD 작성
- 역할 나누기

▼ 05/06 (토)

▼ Schedule

- 🕒 09:00 - 10:00 아침 조례
- 🧑‍💻 10:00 - 12:30 아침 코딩 + 기술 매니저 순회
- 🍽️ 12:30 - 13:30 점심 시간
- 🖥️ 13:30 - 16:00 오후 코딩 (1차)
- 🗓️ 16:00 - 16:30 정기 회의
- 🧑‍💻 16:30 - 18:30 오후 코딩 (2차)
- 🕒 18:30 - 19:30 저녁 시간
- 🧑‍💻 19:30 - 20:00 저녁 코딩 (1차)
- 🧑‍💻 20:00 - 21:30 강학
- 🗓️ 21:30 - 22:00 마무리 회의
- 🖥️ 22:00 - 야간 자율 코딩

▼ 05/07 (일)

▼ 회의 정리

- 현재 진행된 사항
 - 추가
- 오늘 진행할 이슈
 - 추가
- 내일 진행될 사항
 - 추가

▼ 05/08 (월)

- 2023.05.08

▼ 05/09 (화)

- 2023.05.09

▼ 05/10 (수)

- 2023.05.10

▼ 05/11 (목)

• **오전까지**

▼ 공통

- 업무 공유
- 이슈 공유
- 기능 구현
- FE, BE 합치기

▼ 회의 정리

- 현재 진행된 사항
 - 추가
- 오늘 진행할 이슈
 - 추가
- 내일 진행될 사항
 - 추가

• **5/11 (목) 19:00 까지 : 주특기 프로젝트 완성본 제출**

▼ 제출할 것

- 완성작 시연 영상 유튜브 URL (예시 링크)
 - 💡 발표 영상 촬영 가이드 (예시 포함)
- 배포 도메인 주소
- 제목/설명
- FE(클라이언트) GitHub + BE(서버) GitHub
- 제출 링크 : <https://hanghae99-v2.spartacodingclub.kr/v2/homework> (제출 당일 활성화)

▼ 5/11 (목) 저녁 : 팀별 프로젝트 회고 발표

▼ 회고 발표란?

- 간단하게 우리의 프로젝트 시연을 하고, 우리가 이번 주에 개발을 하며 주로 배운 점 (트러블 슈팅 등)을 발표해주시면 됩니다.
- **발표자** : 팀 내의 어떤 분이 하셔도 좋습니다. 연습하는 자리니 너무 긴장하지 마세요!
- **발표 시간 구성 (발표 시간은 5분 내외 + 기술 매니저님들의 피드백 15분 내외)**
 - 1분 30초: 프로젝트 소개 + 시연
 - 회원가입, 로그인 등의 기능이 있다면, 그런 과정은 뛰어넘어주셔도 됩니다. 계정을 생성한 상태로 주요 기능만 간단하게 시연해주세요!
 - 2분 30초 : 개발을 진행하면서 어려웠던 점, 해결한 내용 (트러블슈팅) / 새로 도전한 기술이 있다면?
 - 일주일 동안 어떤 공부를 해서 적용했는지, 뭐가 어려웠고 어떻게 해결했는지, 해결하지 못한 것은 어떤 것이 있는지 편하게 정리해주세요.
 - 새로 도전한 기술이 있다면 (소셜 로그인, 무한 스크롤 등) 자랑해주세요. 물론 없어도 괜찮아요!
 - 1분 : 앞으로 시간이 더 있었다면 어떤 것을 더 할 수 있었는지
 - 지금 어떤 것을 하라는 의미에서 발표하는 내용이 아니에요. 앞으로도 많은 도전을 하시게 될 텐데, 어떤 것을 더 하면 좋았을지를 회고해보면 앞으로의 도전에 큰 참고가 될 것입니다!

		message
--	--	---------
