



EDUCRAFT

LLM 기반 학습 보조 프로그램 - 9팀

중간발표

FRONT-END

USER
INTERFACE



회원가입

Sign Up

로그인

Login

회원가입과
로그인 창 UI



모든 북마크

EduCraft

자료 업로드

문제 생성

문제 풀이

학습 자료 제목

제목을 입력하십시오

학습 자료 내용

내용을 입력하십시오

파일 업로드

파일 선택 선택된 파일 없음

자료 올리기

자료 업로드창 UI

문제 생성

문제 유형 선택

객관식



문제 내용

문제 내용을 입력하시오

옵션 1

옵션 2

옵션 3

옵션 4

문제 생성

문제 생성창 UI



모든 북마크

EduCraft

자료 업로드

문제 생성

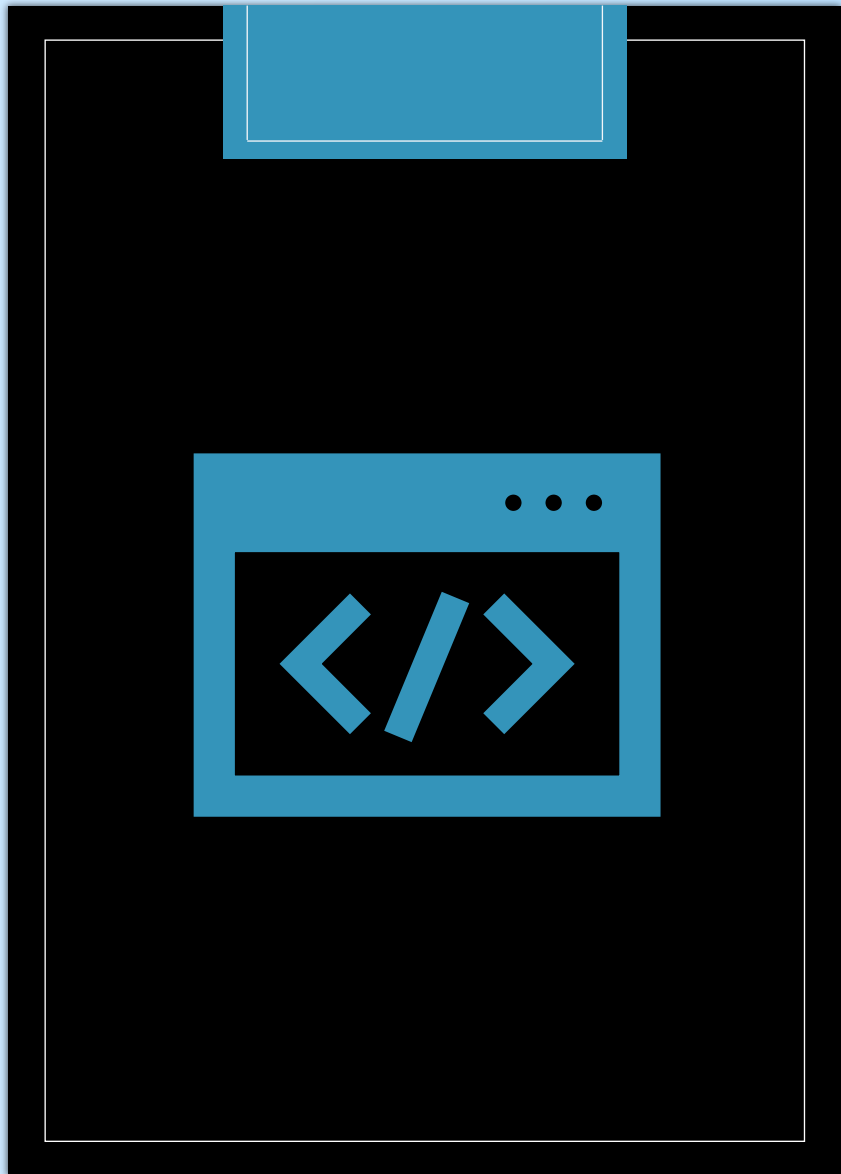
문제 풀이

문제 예시

정답은?

정답

문제풀이창 UI



BACK-END

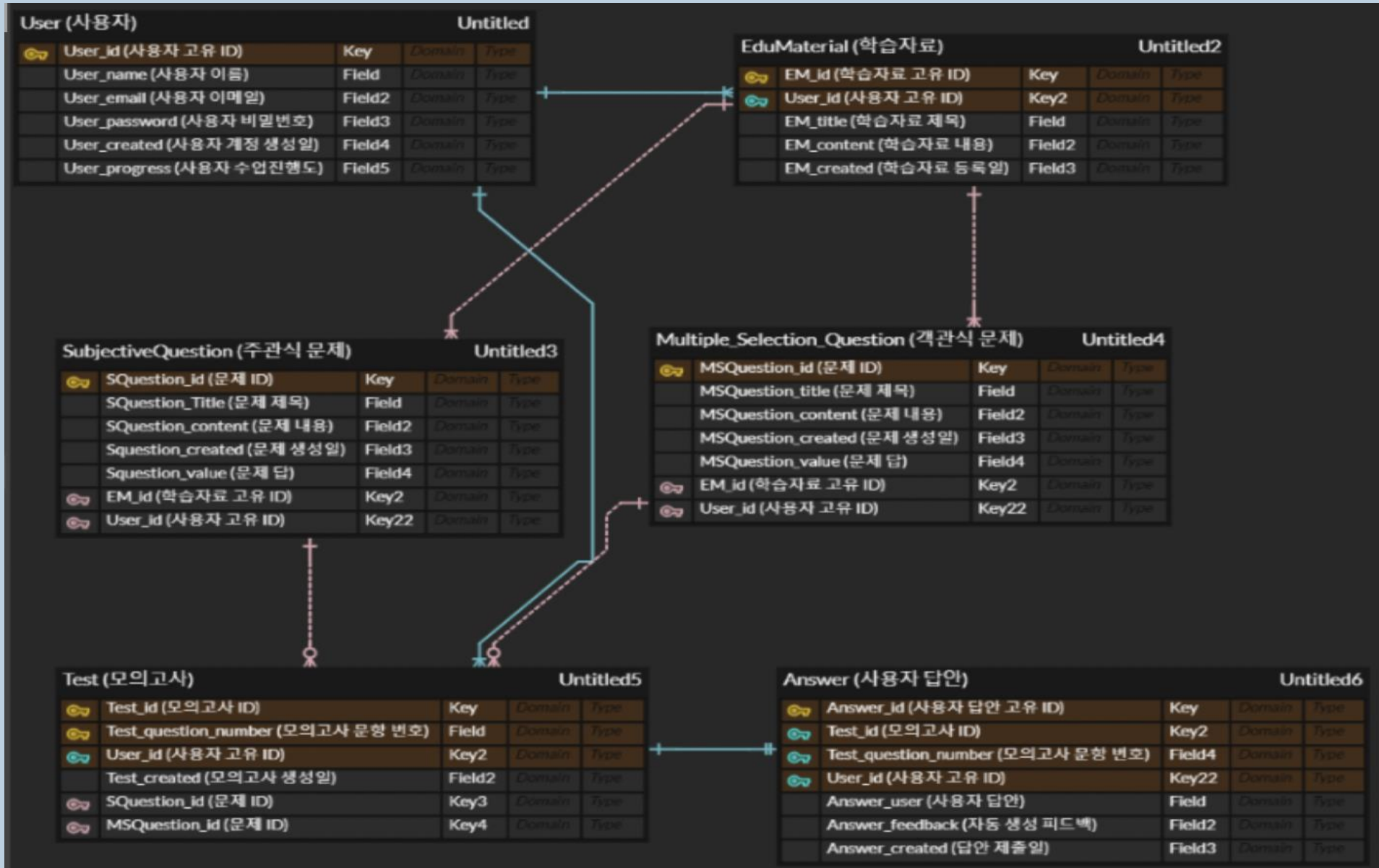
DATABASE / API

사용한 SQL 문 일부

```
1 CREATE TABLE User (  
2     User_id INT AUTO_INCREMENT PRIMARY KEY,  
3     User_name VARCHAR(50) NOT NULL,  
4     User_password VARCHAR(255) NOT NULL,  
5     User_created TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
6     User_progress INT DEFAULT 0  
7 );  
8 CREATE TABLE EduMaterial (  
9     EM_id INT AUTO_INCREMENT PRIMARY KEY,  
10    User_id INT,  
11    EM_title VARCHAR(255) NOT NULL,  
12    EM_content TEXT NOT NULL,  
13    EM_created DATETIME DEFAULT CURRENT_TIMESTAMP,  
14    FOREIGN KEY (User_id) REFERENCES User(User_id)  
15 );  
16 CREATE TABLE SubjectiveQuestion (  
17    SQuestion_id INT AUTO_INCREMENT PRIMARY KEY,  
18    SQuestion_title VARCHAR(255) NOT NULL,  
19    SQuestion_content TEXT NOT NULL,  
20    SQuestion_created DATETIME DEFAULT CURRENT_TIMESTAMP,  
21    SQuestion_value TEXT NOT NULL,  
22    EM_id INT,  
23    User_id INT,  
24    FOREIGN KEY (EM_id) REFERENCES EduMaterial(EM_id),  
25    FOREIGN KEY (User_id) REFERENCES User(User_id)  
26 );  
27 CREATE TABLE Multiple_Selection_Question (  
28    MSQuestion_id INT AUTO_INCREMENT PRIMARY KEY,  
29    MSQuestion_title VARCHAR(255) NOT NULL,  
30    MSQuestion_content TEXT NOT NULL,  
31    MSQuestion_created DATETIME DEFAULT CURRENT_TIMESTAMP,  
32    MSQuestion_value VARCHAR(255) NOT NULL,  
33    EM_id INT,  
34    User_id INT,  
35    FOREIGN KEY (EM_id) REFERENCES EduMaterial(EM_id),  
36    FOREIGN KEY (User_id) REFERENCES User(User_id)  
37 );
```

MYSQL 을
이용한 DB 구현

DATABASE 설계도



작성한 JavaScript 코드 일부

```
29 // 학습 자료 업로드 API
30 app.post('/EduMaterials/upload', (req, res) => {
31   const { EM_id, User_id, EM_title, EM_content } = req.body;
32
33   // 필수 필드 검증
34   if (!EM_id || !User_id || !EM_title || !EM_content) {
35     return res.status(400).json({
36       status: 'error',
37       error: '잘못된 요청입니다.',
38       message: '필수 필드(자료 업로드칸)가 누락되었습니다.'
39     });
40   }
41
42   // 지원하지 않는 미디어 타입 검증
43   const validExtensions = ['.txt', '.hwp', '.mp3']; // 지원하는 파일 확장자
44   const fileExtension = EM_content.split('.').pop(); // 파일 확장자 추출
45
46   if (!validExtensions.includes(`${fileExtension}`)) {
47     return res.status(415).json({
48       status: 'error',
49       error: '지원하지 않는 미디어 타입입니다'
50     });
51   }
52
53   // 데이터베이스에 자료 업로드 (여기서는 예시로 console.log 사용)
54   // 실제로는 데이터베이스에 INSERT 쿼리를 실행해야 합니다.
55   const query = 'INSERT INTO EduMaterial (EM_id, User_id, EM_title, EM_content, EM_created) VALUES (?, ?, ?, ?, NOW())';
56
57   db.query(query, [EM_id, User_id, EM_title, EM_content], (err, result) => {
58     if (err) {
59       console.error('DB 오류: ', err); // 에러 메시지 출력
60       return res.status(500).json({
61         status: 'error',
62         error: '서버 오류입니다.',
63         message: '자료 업로드 중 오류가 발생했습니다.'
```

POSTMAN 을
이용한 API
구현 / 테스트

로그인 요청 테스트

The screenshot displays a REST client interface with a POST request to `http://localhost:3030/users/login`. The request body is a JSON object containing `"User_id": "4321"` and `"User_password": "8765"`. The response is a 200 OK status with a JSON body containing `"message": "로그인 성공"`.

Request:

```
POST http://localhost:3030/users/login
```

Body:

```
{
  "User_id": "4321",
  "User_password": "8765"
}
```

Response:

```
200 OK - 5 ms - 265 B
{
  "message": "로그인 성공"
}
```

회원가입 요청 테스트

POST http://localhost:3030/users/signup

Save Send

Params Authorization Headers (9) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "User_id": "4321",
3   "User_name": "jyp",
4   "User_password": "5678"
5 }
6
7 }
```

Body Cookies Headers (7) Test Results 201 Created - 12 ms - 282 B Save Response

Pretty Raw Preview Visualize

```
{"message": "User registered successfully"}
```

PUT http://localhost:3030/u: + No environment

HTTP http://localhost:3030/users/signup?User_id=4321&User_password=5678 Save

PUT http://localhost:3030/users/signup?User_id=4321&User_password=5678 Send

Params Authorization Headers (9) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> User_id	4321		
<input checked="" type="checkbox"/> User_password	5678		
Key	Value	Description	

```
1  
2 "update_User_password": "8765",  
3 "update_User_email": "pjy@naver.com"  
4
```

회원정보수정 요청 테스트

MySQL Workbench

hansung2 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 x SQL File 10* member

Limit to 1000 rows

SCHEMAS

Filter objects

memo_db

edu_db

Tables fetching...

Views fetching...

Stored Procedures fetching...

Functions fetching...

sakila

sys

world

Query 1

```
7 User_name VARCHAR(50) NOT NULL,  
8 User_password VARCHAR(255) NOT NULL,  
9 User_created TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
10 User_progress INT DEFAULT 0  
11 );  
12  
13 SELECT * FROM User;
```

Result Grid

User_id	User_name	User_password	User_created	User_progress	User_email
1234	jyp	5678	2024-10-29 00:00:00	0	NULL

Schema: edu_db

Body

DELETE http://localhost:3030/users

Params Authorization Headers (9) Body Scripts

none form-data x-www-form-urlencoded raw

```
1 {  
2   "User_id": "4321",  
3   "User_password": "8765"  
4 }  
5
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize

{"message": "회원정보가 삭제되었습니다."}

회원 탈퇴 요청 테스트

작성한 Python 코드 일부

```
1 import openai
2 import os
3
4 openai.api_key = os.getenv("OPENAI_API_KEY")
5
6 try:
7     response = openai.ChatCompletion.create(
8         model="gpt-4",
9         messages=[
10             {"role": "user", "content": "한성대학교에 대해 객관식 문제 5개를 만들어줘"}
11         ]
12     )
13     print(response.choices[0].message['content'])
14 except openai.error.OpenAPIError as e:
15     print(f"An error occurred: {e}")
16
```

DB / 서버와 연계하여 문제 저장, 생성 가능

OPENAI
API를 이용하여
질문하는 템플릿

C:\Users\vmffo\Desktop>python3 test.py

1. 한성대학교는 몇 년에 설립되었나요?
 - a) 1932년
 - b) 1941년
 - c) 1920년
 - d) 1938년
2. 한성대학교의 상징인 '예념지식'은 무엇을 의미하나요?
 - a) 예의를 갖추고 지식을 습득하라
 - b) 예술과 지식을 합하라
 - c) 예지력으로 지식을 얻으라
 - d) 예능과 지식을 겸비하라
3. 한성대학교는 몇 개의 학부가 있나요?
 - a) 20개
 - b) 17개
 - c) 15개
 - d) 23개
4. 다음 중 한성대학교가 위치하지 않은 곳은 어디인가요?
 - a) 서울
 - b) 진주
 - c) 평택
 - d) 진해
5. 한성대학교의 심볼 동물은 무엇인가요?
 - a) 호랑이
 - b) 토끼
 - c) 사자
 - d) 코끼리

터미널에서의 테스트 결과

11-12주차

프론트엔드

문제풀이 결과 및 피드백 표시 UI 개발
문제 생성 및 피드백 요청 API 연동
계정 정보 수정 UI 개발
문제 난이도 및 유형 선택 UI 개발

백엔드

학습 자료를 기반으로 문제를 자동 생성하는 LLM 기능 구현
문제 풀이 후 즉각 피드백을 제공하는 로직 개발
문제 생성 및 피드백 제공 API 구현
사용자 정보 저장 및 업데이트 기능 API 개발

남은 일정 계획

13주차 ~ 최종발표 전

프론트엔드

사용자 정보 관리 API 연동
UI/UX 개선과 디버깅
다양한 브라우저와 기기에서의 UI 테스트
코드 최적화 및 배포 준비

백엔드

JWT 기반 인증 시스템 구현
사용자 맞춤 문제 설정 저장 로직 구현
대규모 요청 처리 성능 테스트 및 최적화
프론트엔드와 백엔드 간의 API 통합 테스트

남은 일정 계획



감사합니다