

## 8주차보고서

팀: 9 팀

기간: 10/21 (Mon) ~ 10/27 (Sun)

### 1. 한 일

- MySQL을 이용하여 DB를 구현하였습니다.

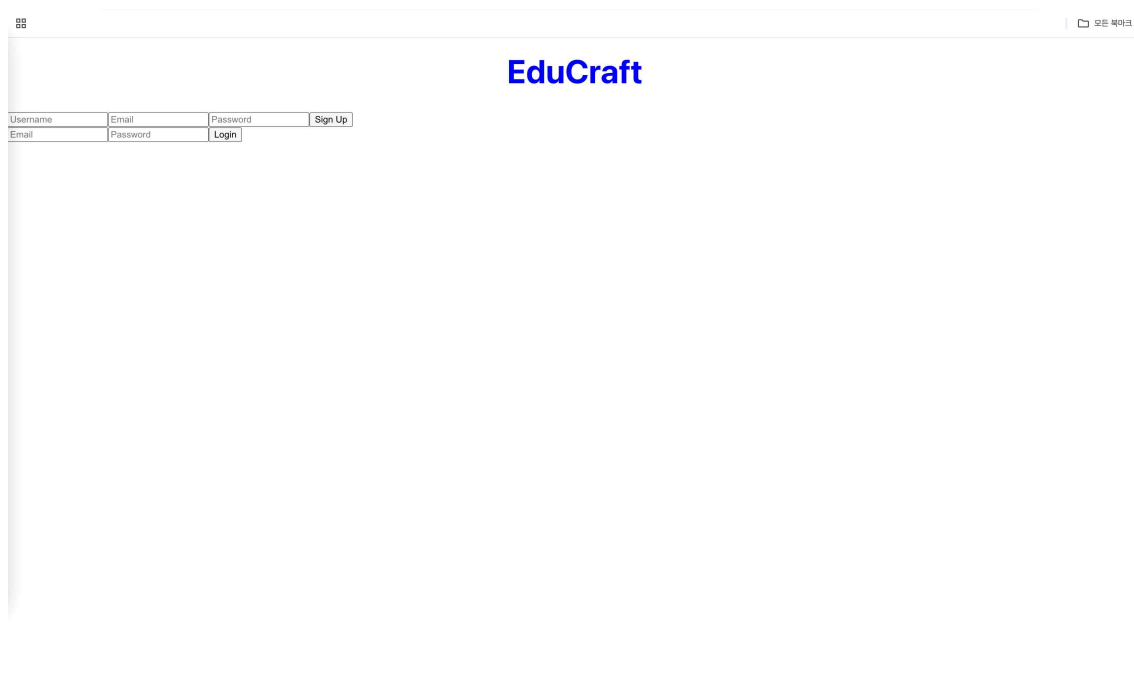
```
1 CREATE TABLE User (  
2     User_id INT AUTO_INCREMENT PRIMARY KEY,  
3     User_name VARCHAR(50) NOT NULL,  
4     User_password VARCHAR(255) NOT NULL,  
5     User_created TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
6     User_progress INT DEFAULT 0  
7 );  
8 CREATE TABLE EduMaterial (  
9     EM_id 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 );
```

- 사용한 SQL문 중 일부입니다.

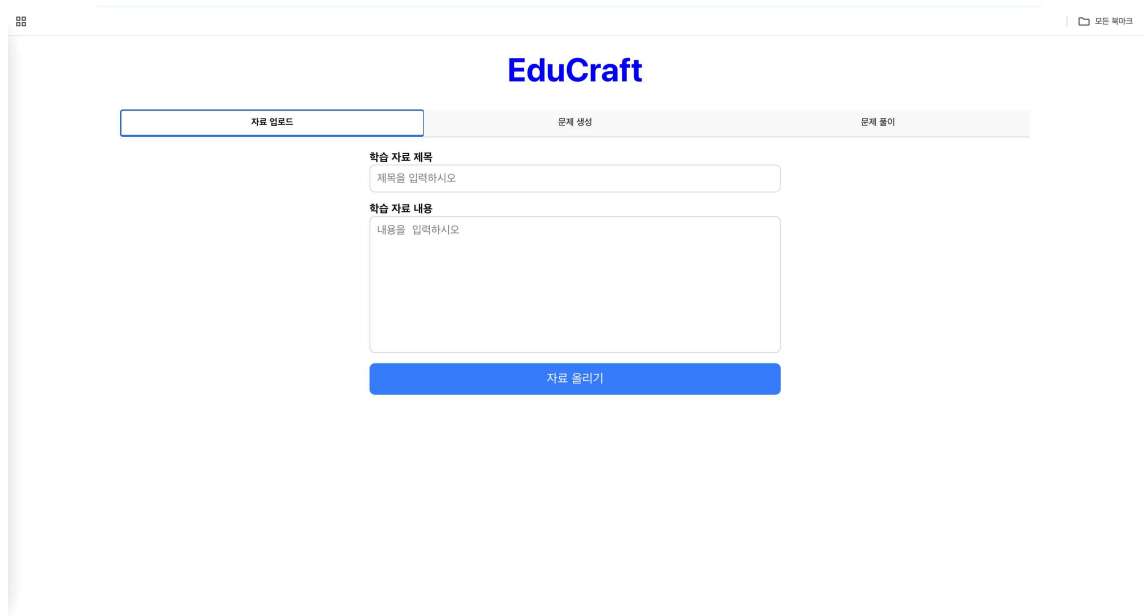
- API를 설계하고 구현하였습니다.

```
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.'
64       });
65     }
66   });
67 });
```

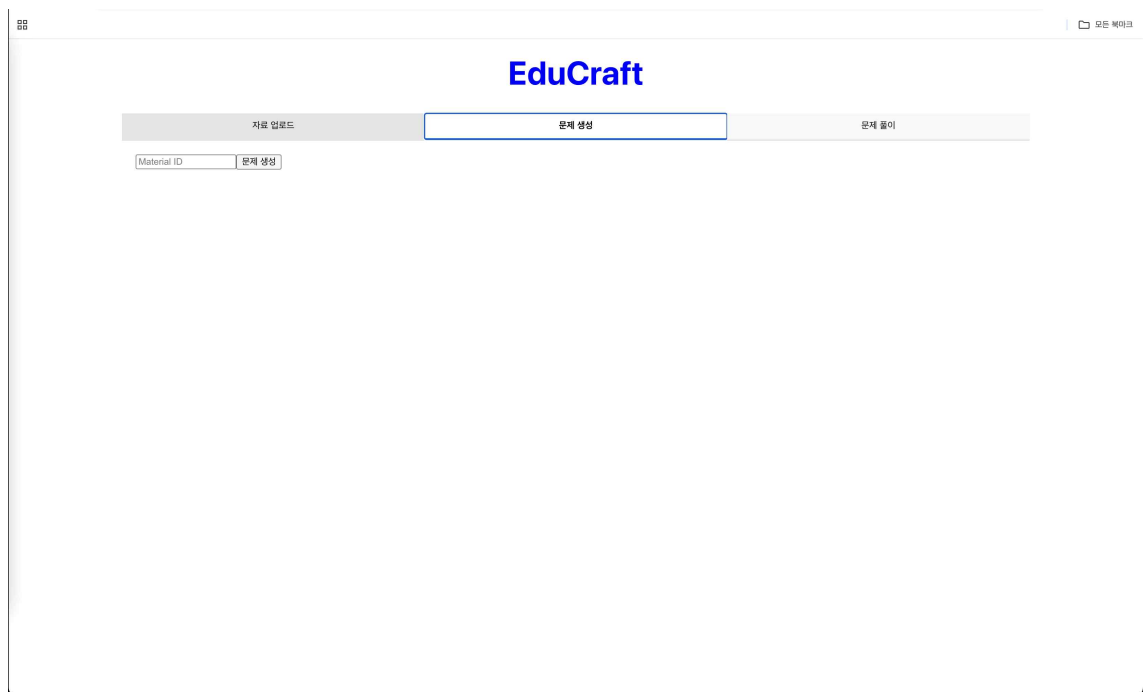
- server.js에 구현하였습니다.



- 로그인 화면 UI를 구현하였습니다. 아직 css는 미완성입니다.



- 자료 업로드 화면 UI를 구현하였습니다.



- 문제 생성 화면 UI를 구현하였습니다.



- 문제 풀이 화면 UI를 구현하였습니다.

## 2. 할 일

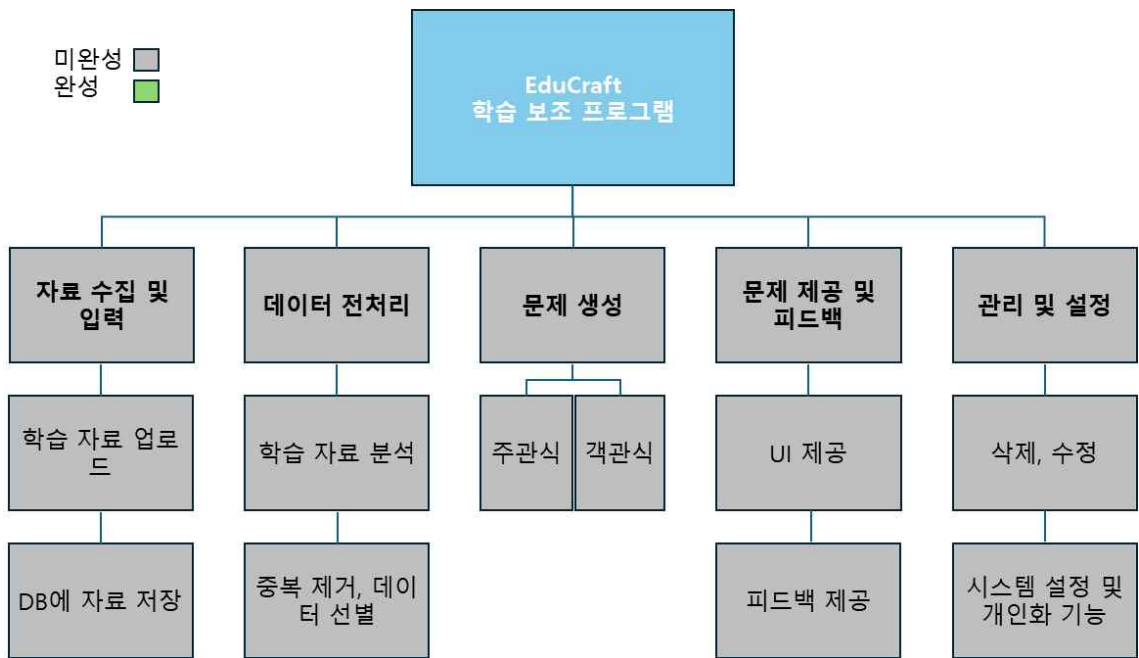
### 프론트엔드 (2명)

- 전처리 과정 시각화: 학습 자료가 전처리되는 과정을 사용자에게 시각적으로 보여줄 UI 구성
- 상태 표시: 전처리 상태 및 완료 여부를 프론트에서 보여주는 기능 추가
- 문제 생성 화면 구현: 객관식/주관식 문제 유형을 선택할 수 있는 인터페이스 개발
- 문제 풀기 화면 구현: 사용자가 문제를 푼 후, 결과와 피드백을 보여주는 UI 개발
- API 연동: 문제 생성 및 피드백 요청 API 연동 (POST /questions/generate, POST /answers)

### 백엔드 (2명)

- 전처리 알고리즘 개발: 입력된 자료를 문제 생성에 적합하게 전처리하는 알고리즘 구현
- 텍스트 분석: 불필요한 데이터 제거 및 텍스트 구조화
- LLM 연동: LLM을 통한 문제 생성 기능 구현 (학습 자료 기반 문제 생성)
- 피드백 시스템: 사용자가 문제를 풀었을 때 즉각 피드백을 제공하는 로직 개발
- API 구현: 문제 생성 및 피드백 제공 API 구현

3. 진행 상황



#### 4. 문제점

- Postman을 사용하여 API를 점검해본 결과 DB와 서버가 연동이 잘 안 되는 문제가 있었음. DB에 문제가 있는 것 같아 문제가 무엇인지 점검해볼 필요가 있음.

- (다음 주 예상 문제점)

- 프론트엔드:

상태 표시와 전처리 과정 시각화를 어떻게 명확히 할 것인지

문제 유형 선택 및 결과 피드백 UI 구성을 어떻게 직관적으로 할 것인지

API 연동할 때 예외 처리나 오류 메시지 처리는 어떻게 할 것인지

- 백엔드:

전처리 알고리즘을 어떻게 설계할 것인지

LLM 연동 및 응답 시간 문제는 어떻게 할 것인지

피드백 시스템 설계를 어떤 식으로 정확하게 할 것인지

API 구현을 어떻게 안정적으로 할 것인지