

OSSProj

3팀 MAC

중간 보고

2017112288 민한결

2019112567 안상연

2019112453 최필환

I 목차

01
프로젝트 개요
프로젝트 개요

02
현재 진행 사항
백엔드
프론트엔드

03
향후 진행 계획
팀활동 페이지 UI
개발 일정

| 01 프로젝트 개요

프로젝트 개요

프로젝트명

동국대학교 LMS 내 협업 파일시스템 구축

목적

동국대학교 LMS 팀 프로젝트 파일 스토리지 시스템을 구축

➡ 팀 프로젝트 과정에서 생성된 파일의 이력 관리가 가능

기대효과



작성자 파악 ↓



수정 내용 파악 ↓

▶ 협업 능력 향상

이전 버전 복구 ↓

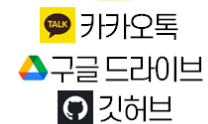
라이선스

오픈소스, 개발환경 라이선스 분석 ▶ GPL v2 라이선스

제안내용



동국대학교 E-class



파일 버전 관리 기능
팀 구성 기능
팀 공지사항 기능

| 02 현재 진행 사항

1. User

▶ 사용한 오픈소스

게시판 프로젝트(<https://github.com/hojunnnnnn/board>)

▶ 오픈소스 개선 사항

사용자 인증에 세션 방식을 사용하여 보안에서 불리, 서버 부하 가능성

➡ JWT 토큰 방식으로 변경

템플릿 엔진 사용하여 Controller의 return이 String으로 React와 통신에 부적절

➡ ResponseEntity를 return값으로 하고 DTO를 HTTP Body를 JSON 형태로 응답

▶ 구현 내용

계정 생성 요청 검증, 가입 처리

로그인 처리, 토큰 생성

회원 정보 조회

```
{
  "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIyWhuEdvjHOMC4pCnWtQ_fpcfattStJVA1nU4sgiE",
  "id": 1,
  "studentId": "2023000001",
  "password": null,
  "name": "김동일",
  "dept": "건설환경공학과"
}
```

2. Team

▶ 사용한 오픈소스

모임 관리 프로젝트(<https://github.com/lcalmsky/spring-boot-app>)

▶ 오픈소스 개선 사항

템플릿 엔진 사용하여 Controller의 return이 String으로 React와 통신에 부적절

➡ ResponseEntity를 return값으로 하고 DTO를 HTTP Body를 JSON 형태로 응답

▶ 구현 내용

팀 생성 및 생성자 팀장으로 팀 가입

팀 목록 조회

팀 세부 정보 조회

팀 삭제

```
{
  ... "teamName": "MAC1",
  ... "teamFellow": [
    ... {
      ... "studentId": "20230000001",
      ... "name": "김동일",
      ... "role": "leader"
    }
  ]
}
```

3. Invitation

▶ 사용한 오픈소스

초대 링크 생성 프로젝트(<https://github.com/heli-os/member-invitation-java-springboot>)

▶ 오픈소스 개선 사항

초대가 링크 형태로 Redis에 저장되어 TTL이 지나거나, 초대 수락 후 삭제됨

➡ 구현 난이도를 고려하여 RDBMS를 사용할 수 있도록 Invitation Table 생성

➡ isAccepted(Boolean) 속성으로 초대장 상태 관리

▶ 구현 내용

팀 ID와 사용자 리스트로 복수 팀원에게 초대 발송

토큰 사용자에게 대한 초대 조회

초대 수락과 팀 가입처리

초대 거절 및 삭제

```
1 {  
2   ... "teamId": 1,  
3   ... "leaderId": 1,  
4   ... "fellowIds": [3, 4]  
5 }
```

Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize

```
1 {  
2   "send_invitations": [  
3     {  
4       "id": 3,  
5       "teamId": 1,  
6       "leaderId": 1,  
7       "fellowId": 3,  
8       "isAccepted": null  
9     },  
10    {  
11      "id": 4,  
12      "teamId": 1,  
13      "leaderId": 1,  
14      "fellowId": 4,  
15      "isAccepted": null  
16    }  
17  ]  
18 }
```


4. AWS 환경설정

▶ 이용할 AWS 서비스

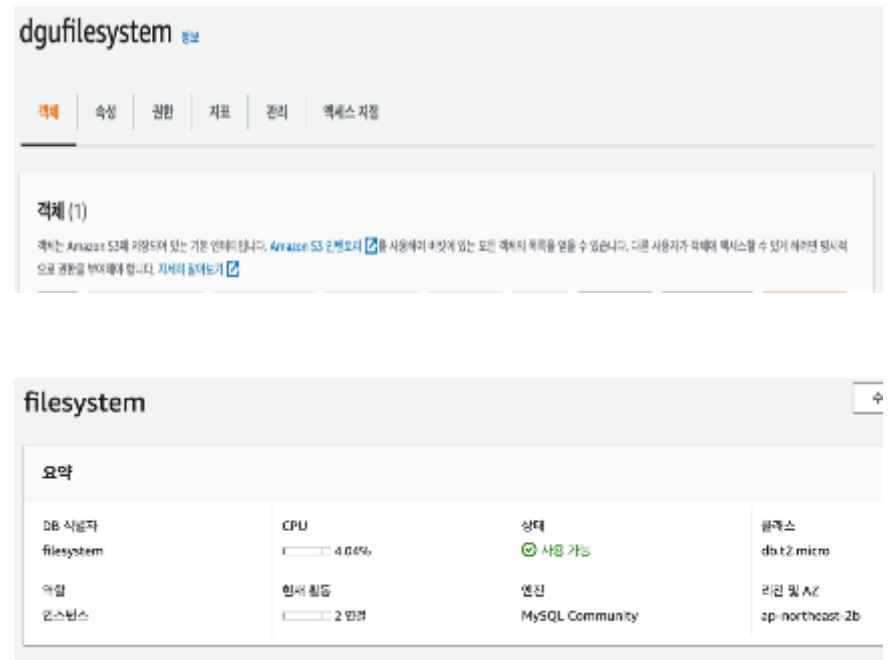
S3, RDS

▶ S3

AWS S3 버킷을 생성하여, 파일의 저장을 할 수 있고,
버전관리기능 활성화를 통해 동일한 이름.확장자에 대해 관리 가능.

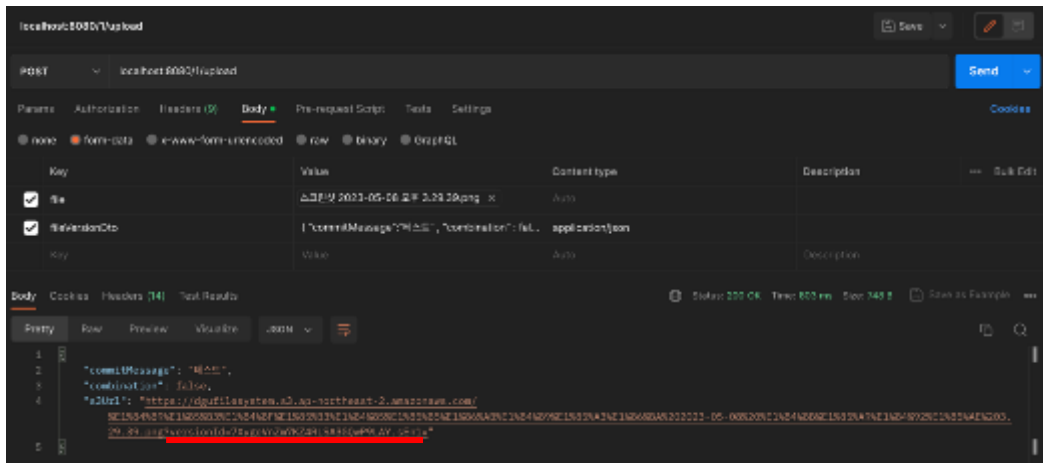
▶ RDS

생성되는 RDS의 엔드포인트를 이용하여 어플리케이션과 연동



5. File(Upload)

POSTMAN을 통한 API 테스트



Member가 파일을 업로드 하려는 teamId를 pathvariable로 전달.
해당 팀의 파일 스토리지에 올릴 File을 multipart/form-data로 전송.
커밋 메세지, 파일 합본 여부 등 DTO에 넣어서 전송.

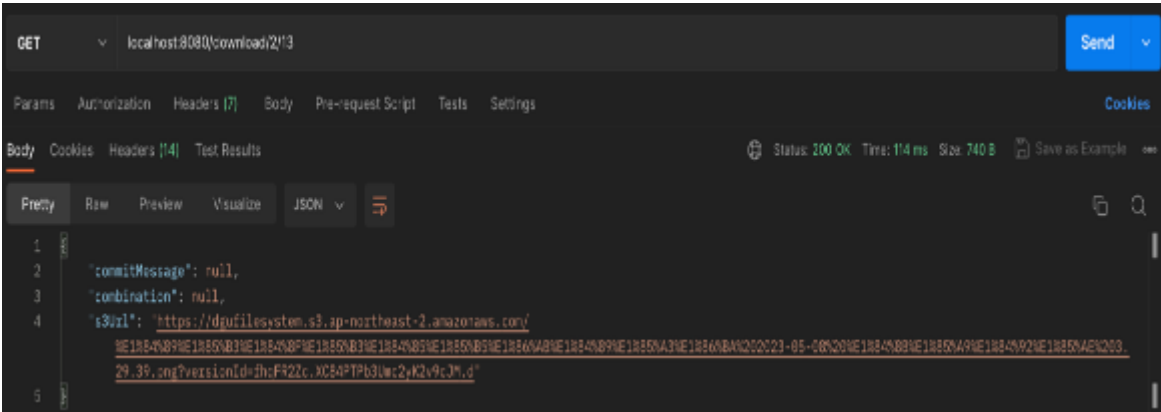
AWS S3 버킷의 저장 정보 확인

버전 (2)					다운로드	열기	삭제	작업
<input type="checkbox"/>	버전 ID	유형	마지막 수정	크기	스토리지 클래스			
<input type="checkbox"/>	D7KEzLqYsai7AIVrXdiMDN4RiPv_jxhX (현재 버전)	png	2023. 5. 10. am 9:21:13 AM KST	15.3KB	Standard			
<input type="checkbox"/>	<u>7XygeVnZm7KZ4RLSA3GQwP9LAYsEn1v</u>	png	2023. 5. 10. am 1:42:17 AM KST	15.3KB	Standard			

POSTMAN을 통해 업로드한 파일을 S3를 통해 조회할 수 있음.
S3상의 파일 버전 ID와 POSTMAN의 응답의 url 버전 ID와 동일.

6. File(Download)

POSTMAN을 통한 API 테스트



다운로드 받고자 하는 File에 대한 Id와 해당 File의 VersionId를 Pathvariable을 통해 url에 전달.
다운로드 요청시 S3 url에 대한 값만 DTO를 통해서 응답.

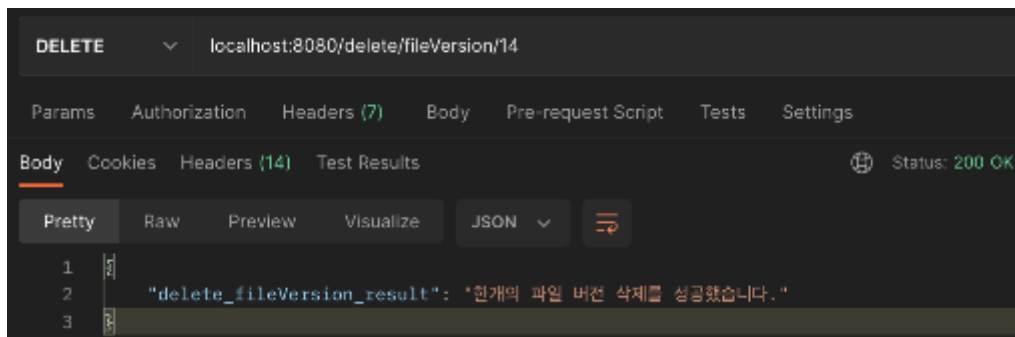
응답 Uri 조회



POSTMAN에서 받은 응답을 웹브라우저 URL에 입력시 조회되는 화면.
현재 이미지 파일의 경우 조회가 되고, 그 외의 파일은 다운로드.
향후 프론트에서 이미지파일에 대한 다운 구현 예정.


7. File(Delete)

POSTMAN을 통한 API 테스트(버전 단일 삭제)



삭제하고자 하는 File 혹은 FileVersion에 대해 url을 설정.
삭제 대상에 대해 Id를 pathvariable로 전달.
삭제 성공시 삭제 성공에 대한 공통응답을 반환.

S3 저장정보 조회

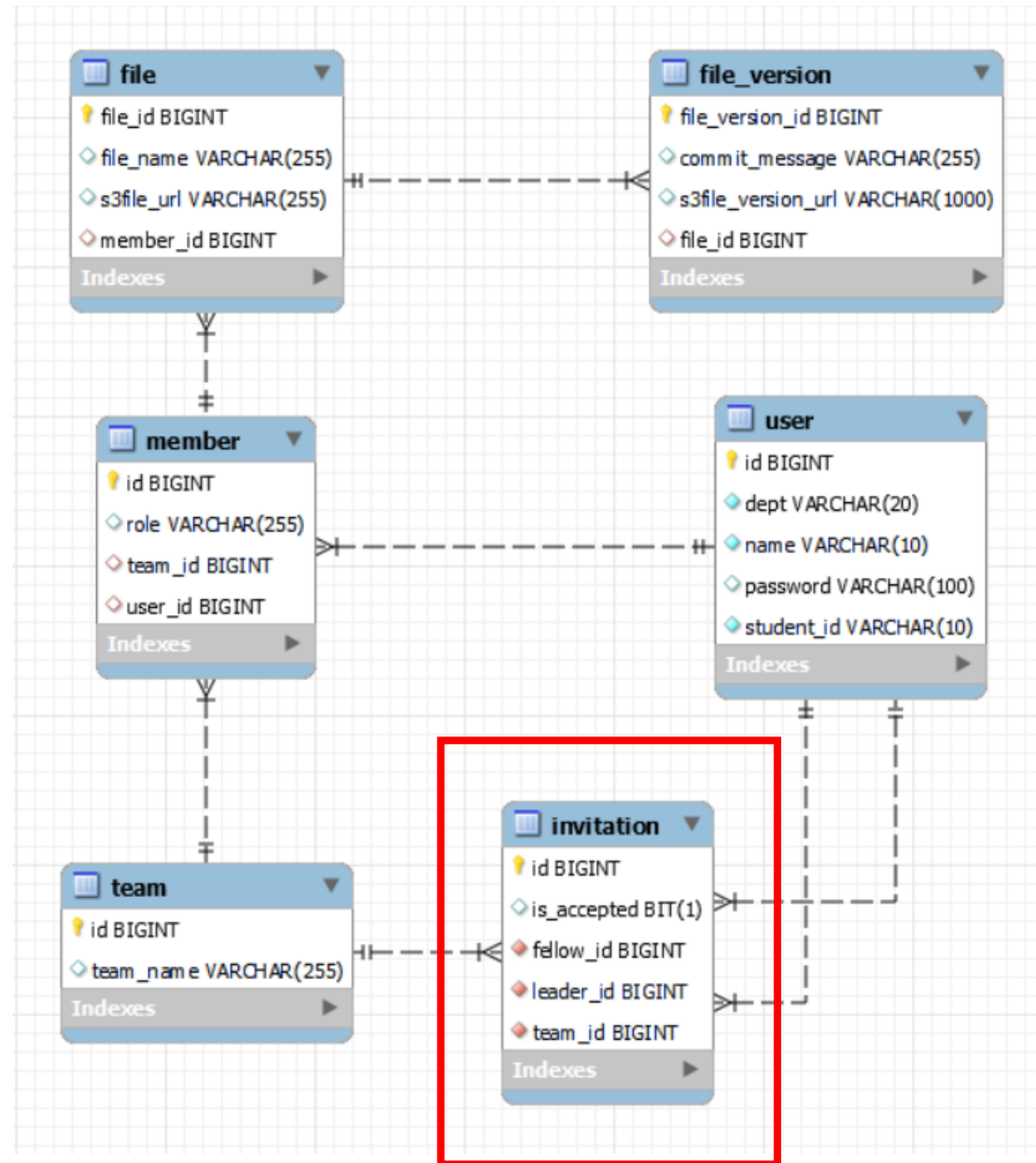


<input type="checkbox"/>	버전 ID	유형	마지막 수정	크기	스토리지 클래스
<input type="checkbox"/>	D7KEzLqYsai1AIVrXdmDN4RIPv_jxhX (현재 버전)	png	2023. 5. 10. am 9:21:13 AM KST	15.3KB	Standard
<input type="checkbox"/>	7XygeVnZm7KZ4RLSA3GQwP9LAYsEn1v	png	2023. 5. 10. am 1:42:17 AM KST	15.3KB	Standard

<input type="checkbox"/>	버전 ID	유형	마지막 수정	크기	스토리지 클래스
<input type="checkbox"/>	D7KEzLqYsai1AIVrXdmDN4RIPv_jxhX (현재 버전)	png	2023. 5. 10. am 9:21:13 AM KST	15.3KB	Standard

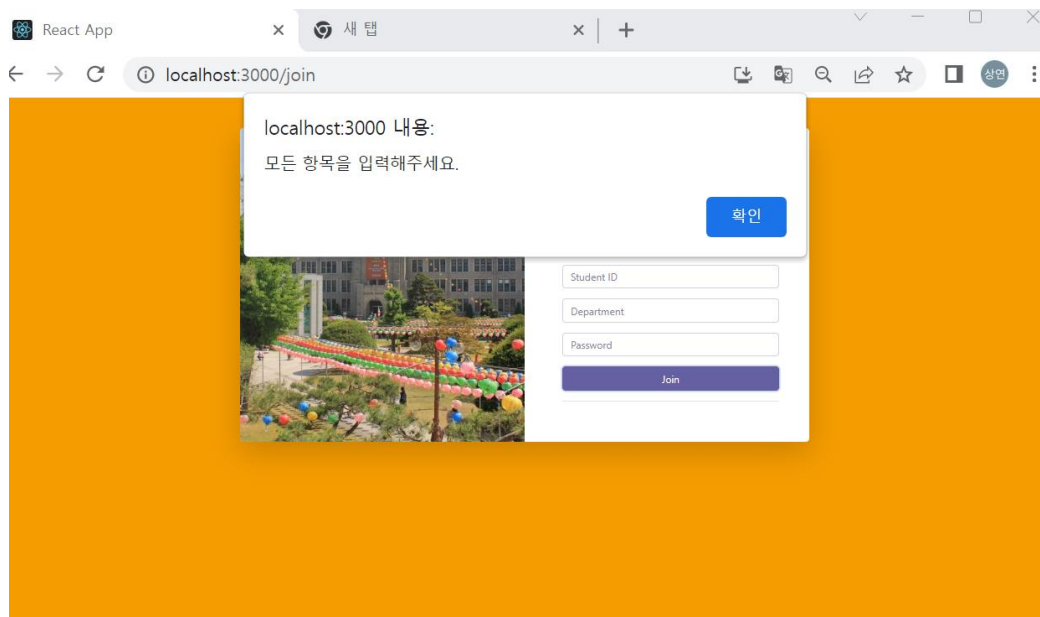
기존 S3에 저장된 FileVersion은 2개였는데, POSTMAN을 통해 FileVersionId가 14인 파일 버전을 삭제함.
따라서 위의 사진과 같이 파일 버전이 단일 삭제된 것을 확인.

백엔드 – 전체 DB 구조

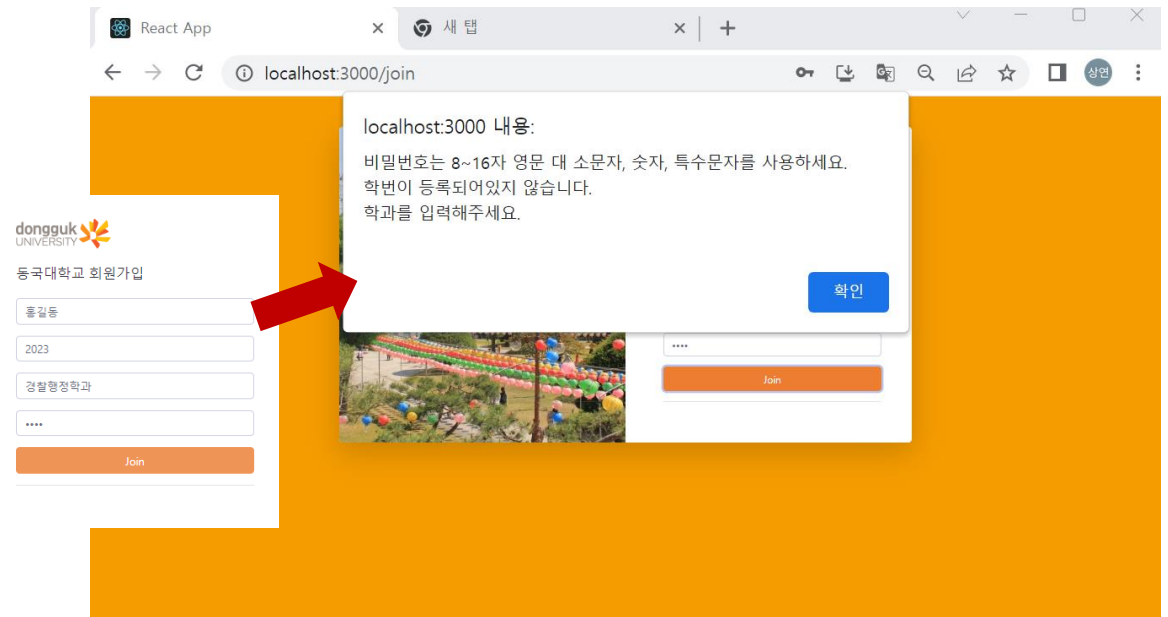


1. 회원가입 페이지 입력에 따른 응답 구현

- ▶ 회원가입 페이지 UI 구현 및 백엔드 연동
- ▶ 회원가입 페이지에서 발생한 입력을 백엔드에 전송.
- 입력이 없을 경우
: 모든 항목을 입력해야 한다는 메시지를 alert창으로 출력.

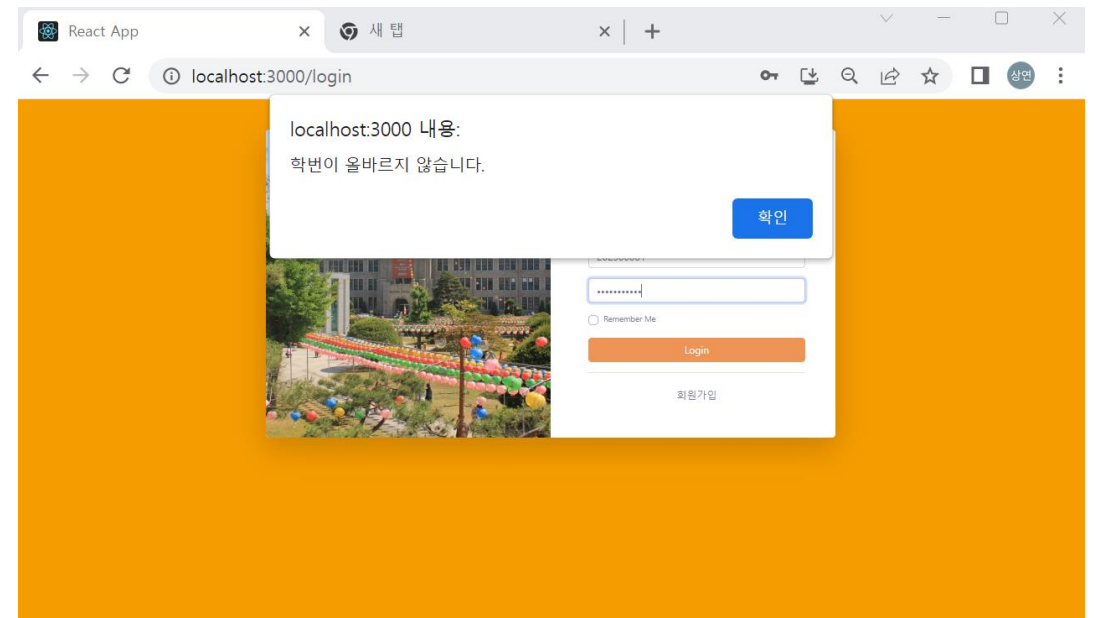
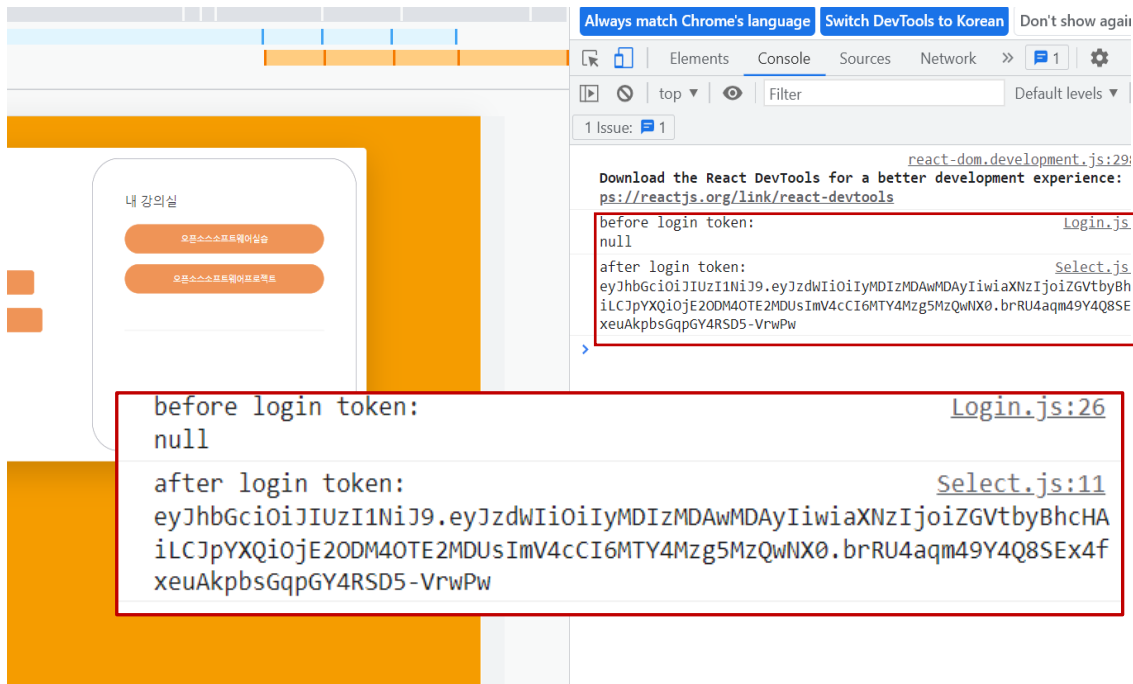


- `status = 400 (bad request)`
: `jsonData`를 불러와 error 메시지를 alert창으로 출력.



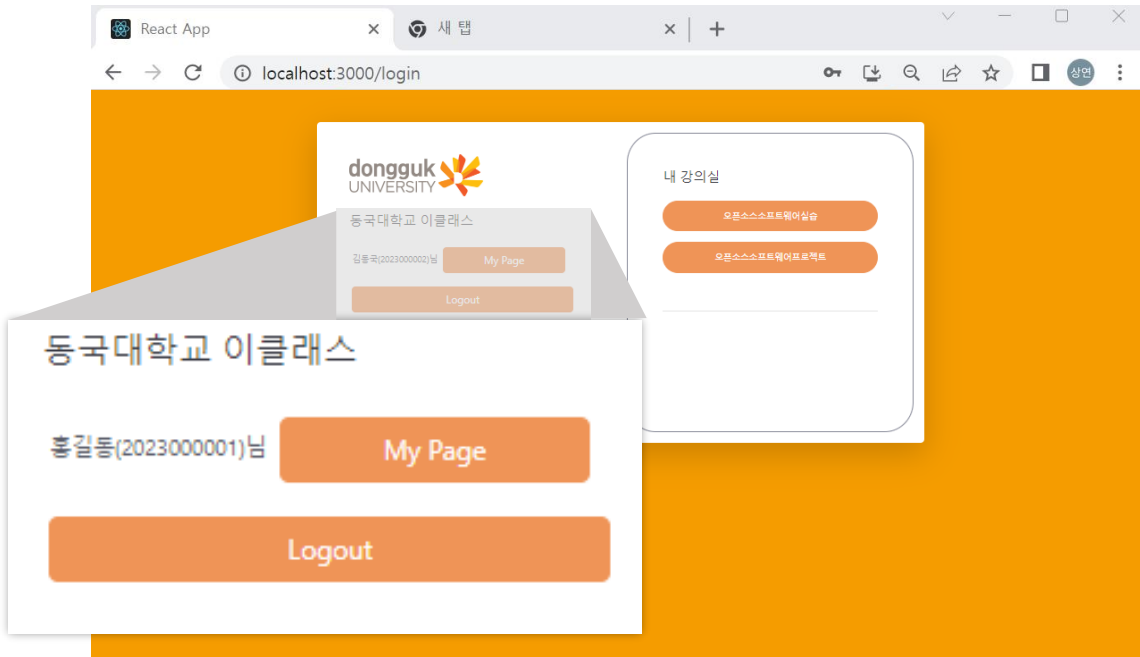
2. 로그인 페이지 입력에 따른 응답 구현

- ▶ 로그인 페이지 UI 구현 후 페이지에서 발생한 입력을 백엔드에 전송.
- ▶ 토큰 기반 인증을 위해 로그인 성공 시 토큰을 발급받고 세션 스토리지에 토큰 저장.
- ▶ bad request에 따른 경고창 출력은 회원가입 페이지와 동일.



3. 토큰을 이용해 사용자 정보 출력

- ▶ 로그인을 하면 발급된 토큰을 세션 스토리지에 저장
- ▶ 과목 선택, 팀 활동 페이지 UI 구현 후 토큰으로 해당 사용자 정보를 요청, 정보 출력
- Ex. 2023000001 홍길동 로그인



```
function Select() {  
  const [userInfo, setUserInfo] = useState({});  
  const navigate = useNavigate();  
  
  useEffect(() => {  
    const token = localStorage.getItem("token");  
    console.log(token);  
    fetch("http://localhost:8080/user", {  
      headers: {  
        Authorization: `Bearer ${token}`,  
      },  
    })  
      .then((response) => response.json())  
      .then((data) => {  
        setUserInfo(data);  
      })  
      .catch((error) => console.log(error));  
  }, []);  
  
  const handleLogout = () => {  
    localStorage.removeItem("token");  
    navigate("/login");  
  };  
}
```

```
/* <!--사용자 이름, 학번 정보--> */  
<span className="mr-2 d-none d-lg-inline text-gray-800 small border-right-0">  
  {`${userInfo.name}(${userInfo.studentId})`}님  
</span>
```


| 03 향후 진행 계획

팀활동 페이지 UI



개발 일정

분류	내용	담당	4월4주	5주	5월1주	2주	3주	4주	5월5주	6월1주	6월2주
웰컴 페이지	로그인 기능 구현	민한결									
	프론트 템플릿 수정	안상연									
파일 스토리지	AWS 설정/파일 업로드 구현	최필환									
팀 활동 페이지	팀 구성 기능 구현	민한결									
웰컴 페이지	로그인 요청 및 응답 처리	안상연									
파일 스토리지	버전별 파일 관리/다운로드	최필환									
	파일 관련 기능 구현										
팀 활동 페이지	공지사항 기능 구현	민한결									
	팀 활동 요청 및 응답 처리	안상연									
	팀 구성 요청 및 응답 처리										
	파일 업로드, 다운로드, 삭제	최필환									
	파일 목록 페이지징 처리										
	예외 처리, 배포	민한결									
	공지사항 요청 및 응답 처리										
	파일 스토리지 요청/응답	안상연									
	통합 테스트, 배포	최필환									
최종 수정 및 발표		모 두									

| Q&A

감사합니다.