



5월 3주차 회의록

팀	MAC
참석자	 최필환(한)  한결 민(상) 상연 안
장소	원흥관 3층 아리수
시간	23.05.21 12:00~14:00

팀 주제

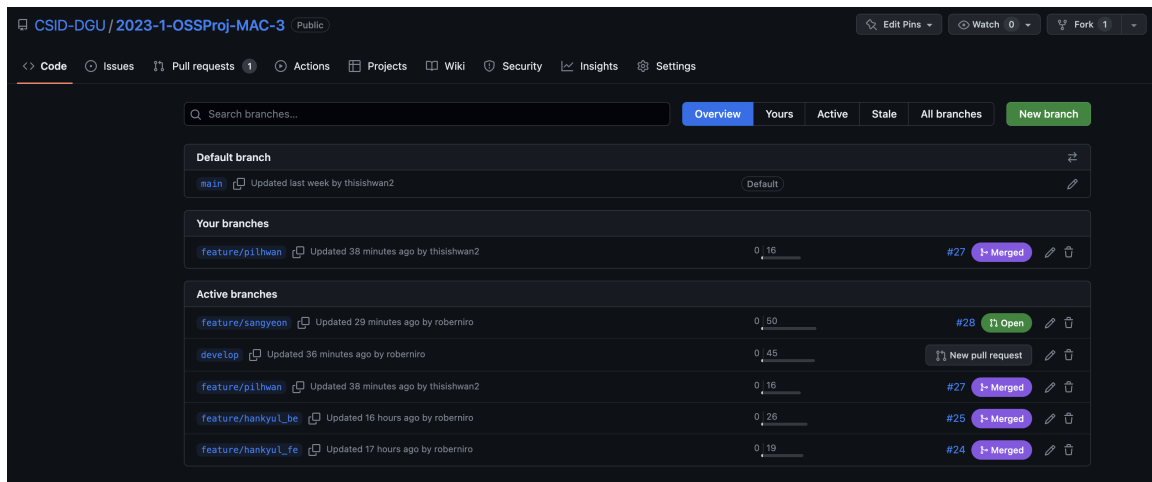
동국대학교 E-class 내 파일 스토리지 기능 구현 → 협업 능력 향상

주요 안건

- 진행도 파악
- 프로젝트 변경 사항
- 진행 사항 파악

1. 프로젝트 변경사항

1. 브랜치 확장



1. 진행도 파악 및 역할 재조정

분류	내용	담당	4월4주	5주	5월1주	2주	3주	4주	5월5주	6월1주	6월2주
웹컴 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									
웹컴 페이지	로그인 요청 및 응답 처리	안상연									
파일 스토리지	파일 관련 기능 구현	최필환									
팀 활동 페이지	공지사항 기능 구현	민한결									
	팀 구성 요청 및 응답 처리	안상연									
	파일 목록 페이지 처리	최필환									
	공지사항 요청 및 응답 처리	민한결									
	파일 스토리지 요청/응답	안상연									
	통합 테스트, 배포	최필환									
최종 수정 및 발표		모 두									

2. 진행 사항 파악

1. 백엔드

a. 최필환

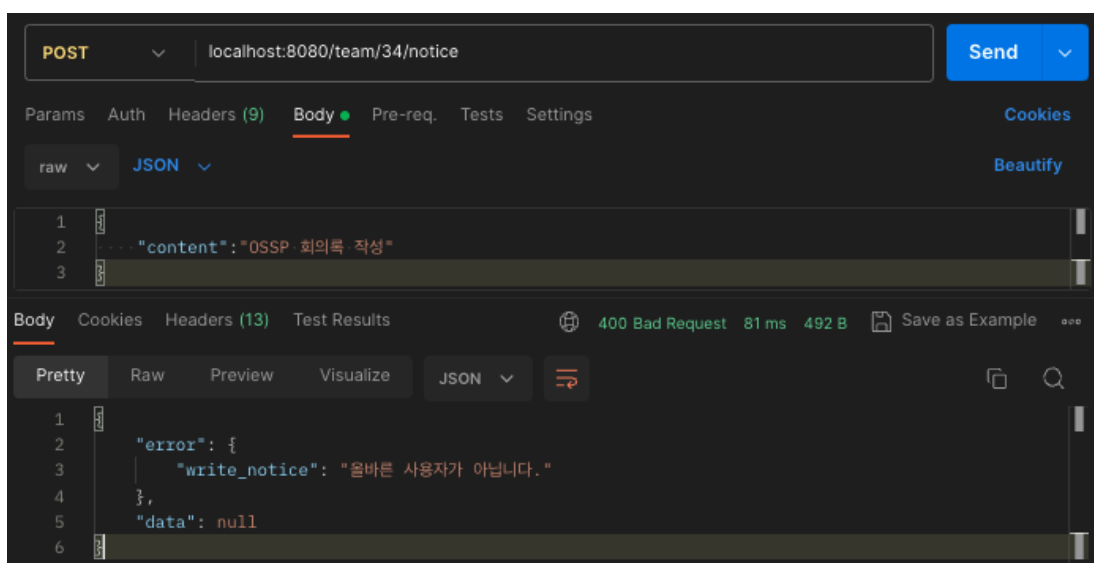
- 공지사항 api 명세서

domain	Aa feature	method	path	request	response(200)	response(400)
notice	공지사항 조회 <input type="checkbox"/> 일기	get	/team/{teamId}/notice		{ "get_notices": [{ "id": 1, "content": "OSSP 회의록 작성" }, { "id": 2, "content": "OSSP 보고서 작성" }, { "id": 3, "content": "OSSP 발표자료 작성" }] }	{ "error": { "get_notice": "공지사항 조회에 실패했습니다." }, "data": null }
notice	공지사항 등록	post	/team/{teamId}/notice	{ "content": "OSSP 회의록 작성" }	{ "id": 1, "content": "OSSP 회의록 작성" }	{ "error": { "write_notice": "공지사항 생성에 실패했습니다." }, "data": null }
notice	공지사항 수정	put	/team/{teamId}/notice/{noticeId}	{ "content": "OSSP 과제" }	{ "id": 1, "content": "OSSP 과제" }	{ "error": { "update_notice": "공지사항 수정에 실패했습니다." }, "data": null }
notice	공지사항 삭제	delete	/team/{teamId}/notice/{noticeId}		{ "delete_notice": "공지사항 삭제에 성공했습니다." }	{ "error": { "delete_notice": "공지사항 삭제에 실패했습니다." }, "data": null }

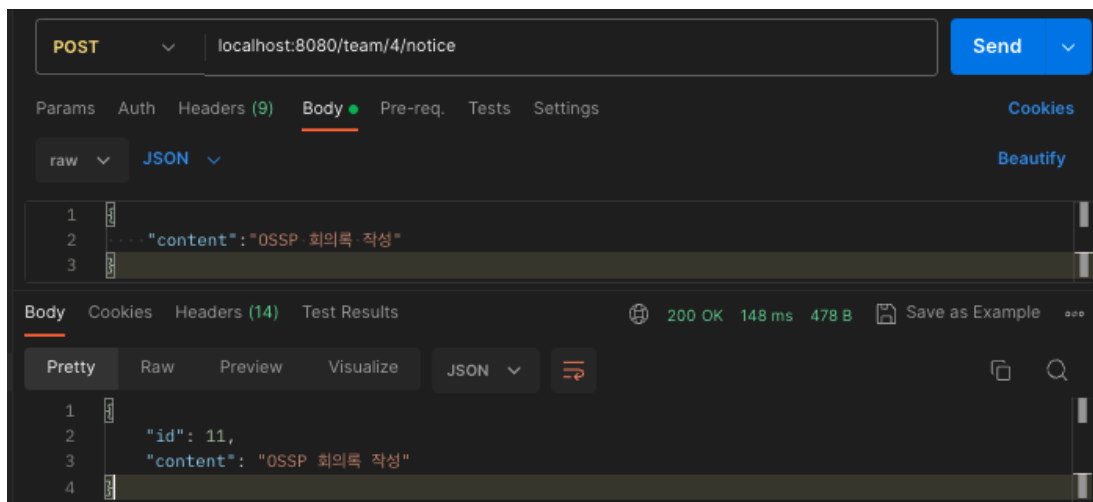
각 팀에 맞는 member인지에 대한 검증을 통해 기본적인 공지사항에 대한 CRUD와 요청 응답을 구현.

- Member 검증

만약 올바르지 않은 사용자가 접근했을 경우 공통적으로 아래와 같이 올바른 사용자가 아니란 것을 응답해준다.

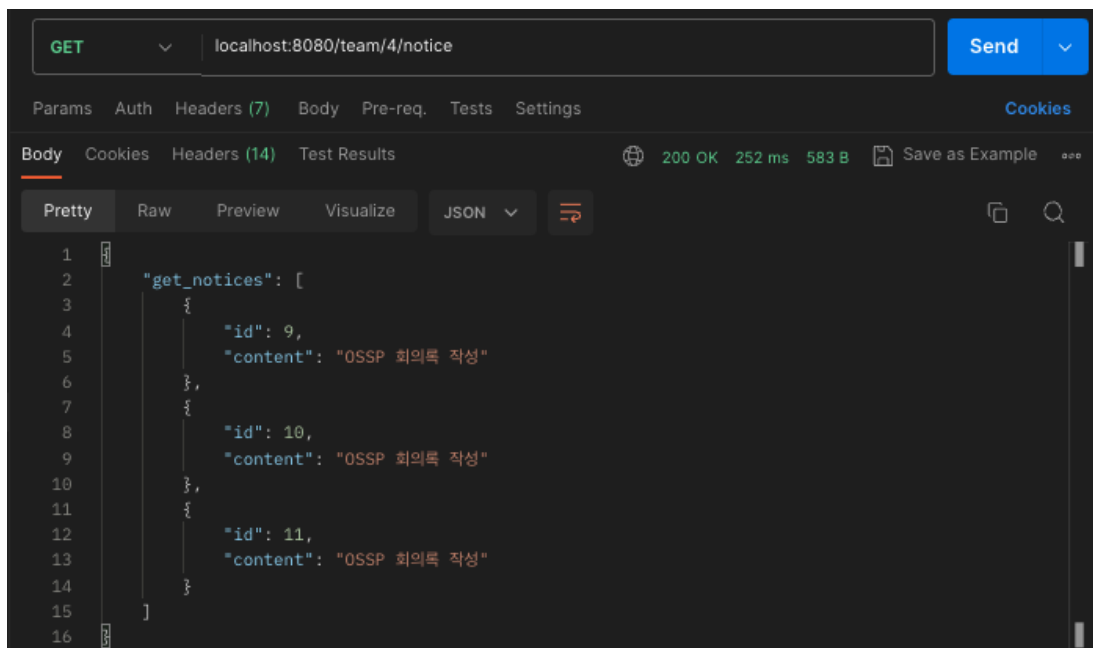


- 공지사항 등록



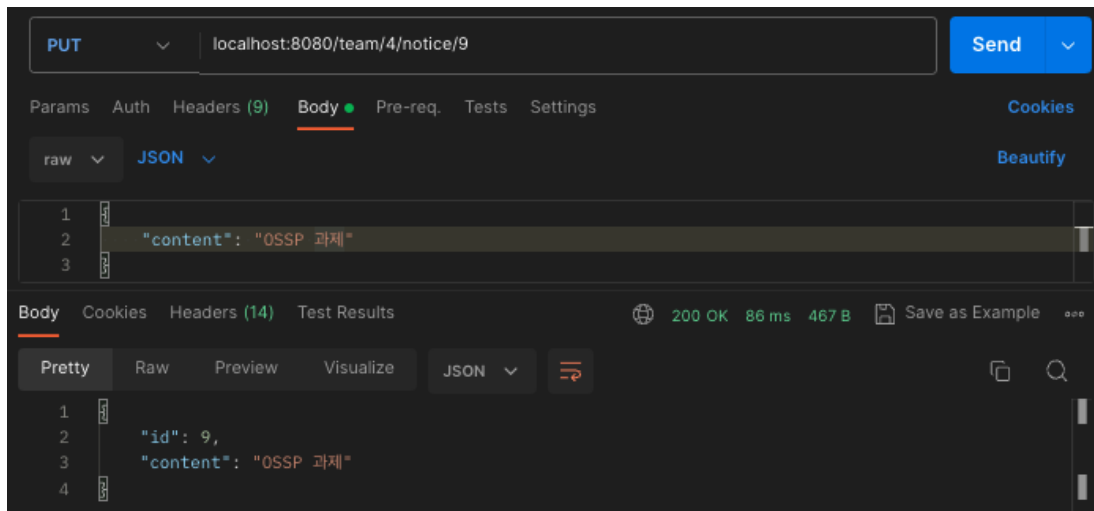
공지할 문구를 요청으로 보내면, 응답을 반환한다.

- 공지사항 조회

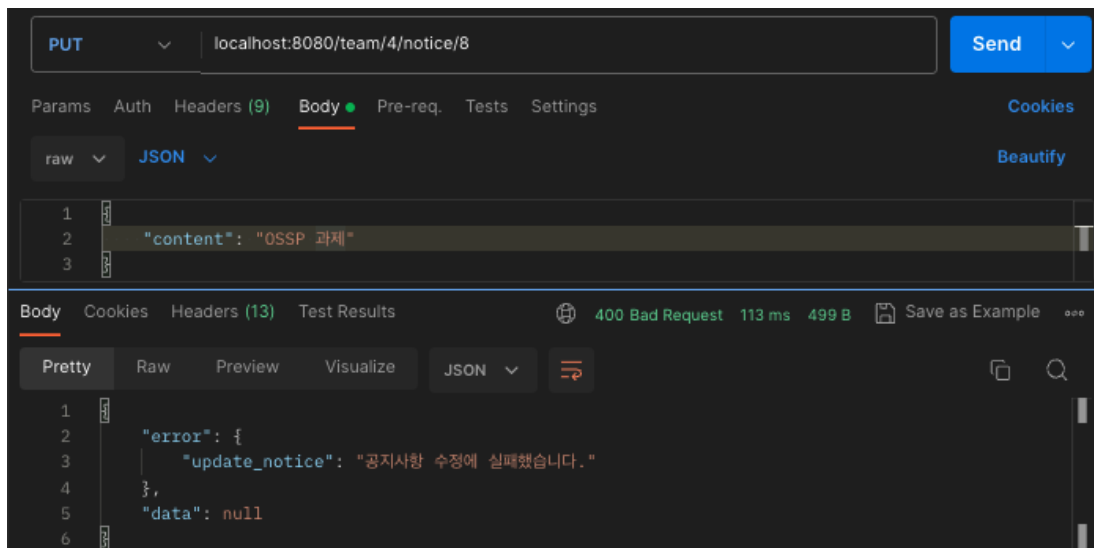


특별한 요청없이 공지사항을 GET으로 조회하면 json으로 쌓여진 리스트를 반환해 준다.

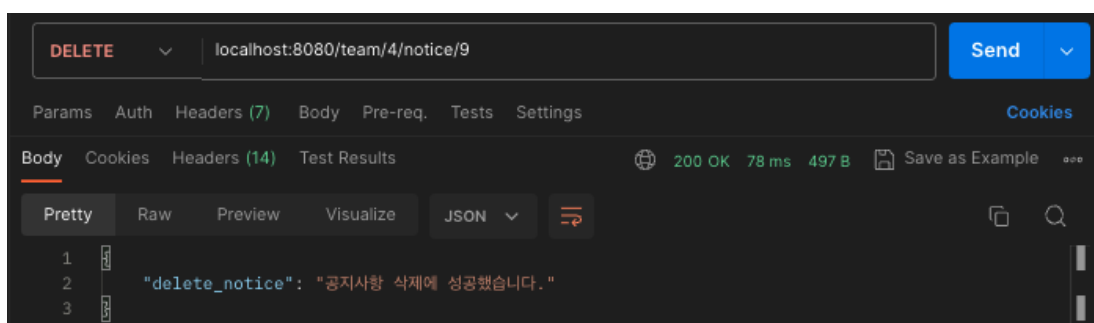
- 공지사항 수정



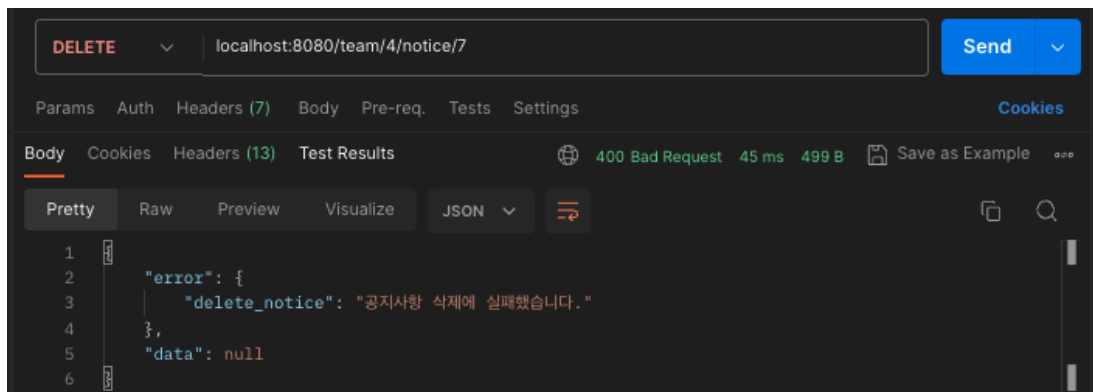
수정할 content를 요청으로 전달하면 수정된 응답을 보여준다. 만약 없는 파일을 수정하려 하거나 잘못된 수정을 한다면 아래와 같이 실패를 반환한다.



- 공지사항 삭제



원하는 공지사항에 대한 pathvariable을 url로 전달하여서 공지사항을 삭제한다.
만약 올바르지 않은 삭제 접근시 아래와 같이 삭제 실패 응답을 준다.



b. 민한결

- API 명세서 작성

domain	As feature	method	path	request	response(200)	response(400)	query string
user	회원가입 <small>(대 일기)</small>	post	/user/signup	{ "studentId": "2023000001", "password": "abcd1234!", "name": "김동욱", "dept": "건설환경학과" }	{ "token": null, "id": 1, "studentId": "2023000001", "password": null, "name": "김동욱", "dept": "건설환경학과" }	{ "error": { "valid_password": "비밀번호는 8~16자 영문 대 소문자 숫자 특수문자를 사용하세요.", "valid_validStudentId": "학번이 등록되어있지 않습니다.", "valid_deptValid": "학과를 입력해주세요요.", "data": null } }	
user	로그인	post	/user/signin	{ "studentId": "2023000001", "password": "abcd1234!" }	{ "token": " eyJHSUoCOWIzOTNIUjRyZWZlVW9kYmQzMDAwMDAwNiJkaXNhcjovZDZhbmBncHAcAklClGYKCDGSEGOQGNkdSpqdmM4cCtSM7MTVMUMAMDOCOXOMvWSVPfO8r4mN3KaLBmxWdE9yOmJlW8lpbzUCeM.", "id": 1, "studentId": "2023000001", "password": null, "name": "김동욱", "dept": "건설환경학과" }	{ "error": { "valid_login": "학번이 올바르지 않습니다." }, "data": null }	
user	로그인 학생 정보 조회	get	/user		{ "token": null, "id": 1, "studentId": "2023000001", "password": null, "name": "김동욱", "dept": "건설환경학과" }		
user	다른 학생 리스트 조회	get	/user/all		{ "token_list": [{ "token": null, "id": 2, "studentId": "2023000002", "password": null, "name": "김단국", "dept": "산업시스템공학과" }, { "token": null, "id": 3, "studentId": "2023000003", "password": null, "name": "김민국", "dept": "멀티미디어공학과" }] }		
team	팀 생성	post	/team	{ "teamName": "MAC" }	{ "teamId": "1", "teamName": "Fellow", "teamFollow": null }	{ "error": { "create_team": "팀 이름을 입력해주세요요.", "data": null } }	
team	가입한 팀 목록 조회	get	/team		{ "get_teams": [{ "teamName": "MAC", "teamID": "1" }, { "teamName": "MS", "teamID": "2" }]	{ "error": { "get_teams": "팀 정보가 존재하지 않습니다." }, "data": null }	
team	가별 팀 조회	get	/team/teamId		{ "teamName": "MAC", "teamFollow": [{ "studentId": "2023000001", "name": "김동욱", "role": "Leader" }, { "studentId": "2023000002", "name": "김단국", "role": "Member" }] }	{ "error": { "get_team": "팀 정보가 존재하지 않습니다." }, "data": null }	
team	팀 삭제	delete	/team/teamId		{ "delete_team": "팀 삭제에 성공했습니다." }	{ "error": { "delete_team": "해당 팀이 존재하지 않습니다." }, "data": null }	
invitation	팀원 초대	post		{ "teamId": 1, "leaderId": 1, "fellowIds": [2,3] }	{ "send_invitations": [{ "invitationId": 1, "teamId": 1, "leaderId": 1, "fellowId": 2, "isAccepted": null }, { "invitationId": 2, "teamId": 1, "leaderId": 1, "fellowId": 3, "isAccepted": null }] }	{ "error": { "send_invitation": "팀장이 아닙니다." }, "data": null }	
invitation	초대 조회	get	/team/invitation		{ "get_invitations": [{ "invitationId": 1, "teamId": 1, "leaderId": 1, "fellowId": 2, "isAccepted": null }]	{ "error": { "get_invitations": "초대한 팀이 존재하지 않습니다." }, "data": null }	
invitation	초대 수락	put	/team/invitation/{invitationId}/accept		{ "invitationId": 1, "teamId": 1, "leaderId": 1, "fellowId": 2, "isAccepted": true true }	{ "error": { "accept_invitation": "초대한 사용자가 아닙니다." }, "data": null }	
invitation	초대 거절	put	/team/invitation/{invitationId}/reject		{ "invitationId": 1, "teamId": 1, "leaderId": 1, "fellowId": 2, "isAccepted": false }	{ "error": { "reject_invitation": "이미 수락된 초대입니다." }, "data": null }	
invitation	초대 삭제	delete	/team/invitation/{invitationId}		{ "delete_invitation": "초대가 삭제되었습니다." }	{ "error": { "delete_invitation": "초대한 사용자가 아닙니다." }, "data": null }	

- 에러 처리 통일

프론트엔드 작업 편의성 위해 에러 리스폰스 바디 형태 통일

The screenshot shows a REST client interface with tabs for Body, Cookies, Headers (13), and Test Results. The Body tab is selected, and the response is displayed in a 'Pretty' JSON format. The JSON structure is as follows:

```

1  {
2    "error": {
3      "get_invitations": "초대된 팀이 존재하지 않습니다."
4    },
5    "data": null
6  }

```

The screenshot shows a REST client interface with tabs for Body, Cookies, Headers (13), and Test Results. The Body tab is selected, and the response is displayed in a 'Pretty' JSON format. The JSON structure is as follows:

```

1  {
2    "error": {
3      "get_team": "팀 정보가 존재하지 않습니다."
4    },
5    "data": null
6  }

```

- Invitation Dto 및 Invitation Service 수정

초대장 조회시 발송자와 팀명 확인할 수 있도록 id만 전달 → 이름도 전달로 수정

Pretty

Raw

Preview

Visualize

JSON

```

1  {
2    "get_invitations": [
3      {
4        "invitationId": 9,
5        "teamId": 5,
6        "teamName": "MAC5",
7        "leaderId": 1,
8        "leaderName": "김건국",
9        "fellowId": 2,
10       "isAccepted": false
11     },
12     {
13       "invitationId": 11,
14       "teamId": 6,
15       "teamName": "MAC6",
16       "leaderId": 1,
17       "leaderName": "김건국",
18       "fellowId": 2,
19       "isAccepted": false
20     }
21   ]
22 }
```

2. 프론트엔드

a. 민한결

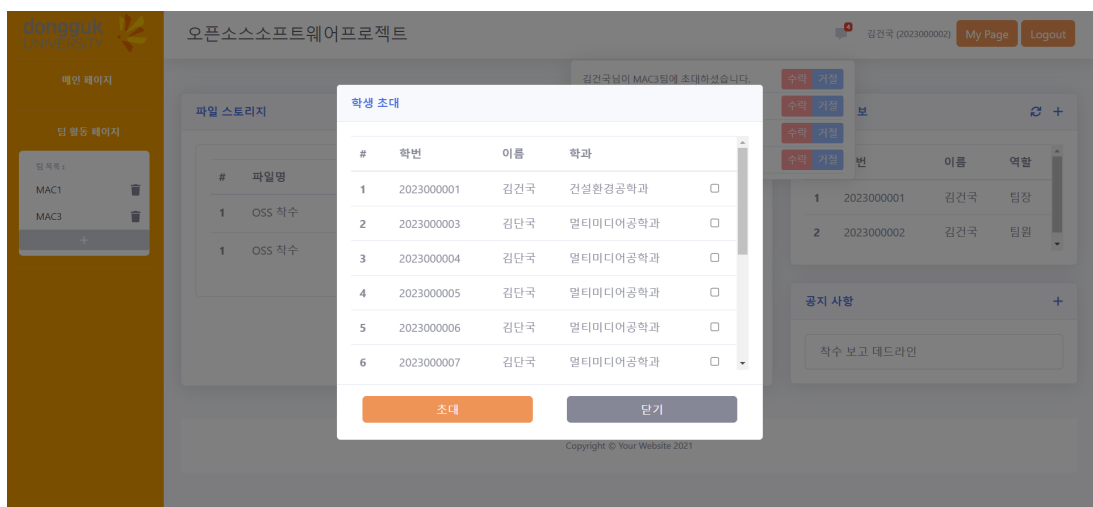
- 팀원 조회 구현

팀원 학번, 이름, 역할 렌더링 및 새로고침 버튼 추가



- 팀원 초대 구현

팀 정보 영역 + 버튼 클릭시 모달로 학생 조회 및 체크한 학생에게 초대장 발송 구현



- 초대장 카운터 구현

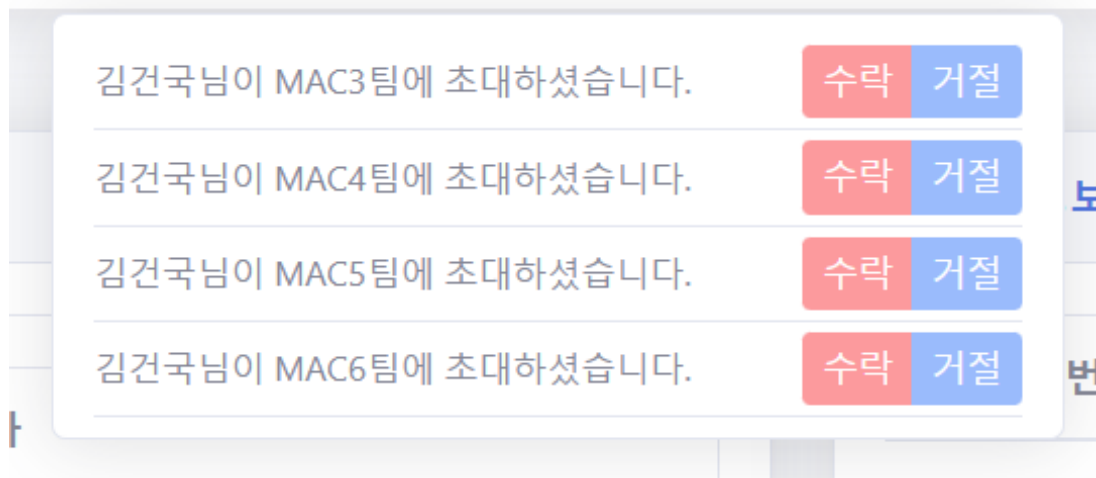
초대장 조회 API fetch시 useState 혹은 redux로 초대장수 관리하여 아이콘에 초대장 개수 표시

```
const fetchInvitation = () => {
  const token = sessionStorage.getItem("token");
  fetch("http://localhost:8080/team/invitation", {
    headers: {
      Authorization: `Bearer ${token}`
    }
  })
    .then((response) => response.json())
    .then((data) => {
      if (data && Array.isArray(data.get_invitations)) {
        setAlertCount(data.get_invitations.length);
        setInvitationList(data.get_invitations);
      }
    })
    .catch((error) => console.log(error));
};
```

```
<div>
  <FontAwesomeIcon icon={faMessage} />
  {/* Counter - Alerts */}
  <span className="badge badge-danger badge-counter">
    {alertCount > 0 ? alertCount : 0}
  </span>
</div>
```

- 초대장 조회 구현

네비게이션 바에서 메시지 버튼 클릭시 드롭다운 형식으로 초대장 조회



- 초대장 수락 구현

수락 버튼 클릭시 초대장 수락 api에 put요청하여 상태 변경

```
▼{invitationId: 5, teamId: 3, teamName: null, leaderId: 1, leaderName: null,
  fellowId: 2
  invitationId: 5
  isAccepted: true
  leaderId: 1
  leaderName: null
  teamId: 3
  teamName: null}
```

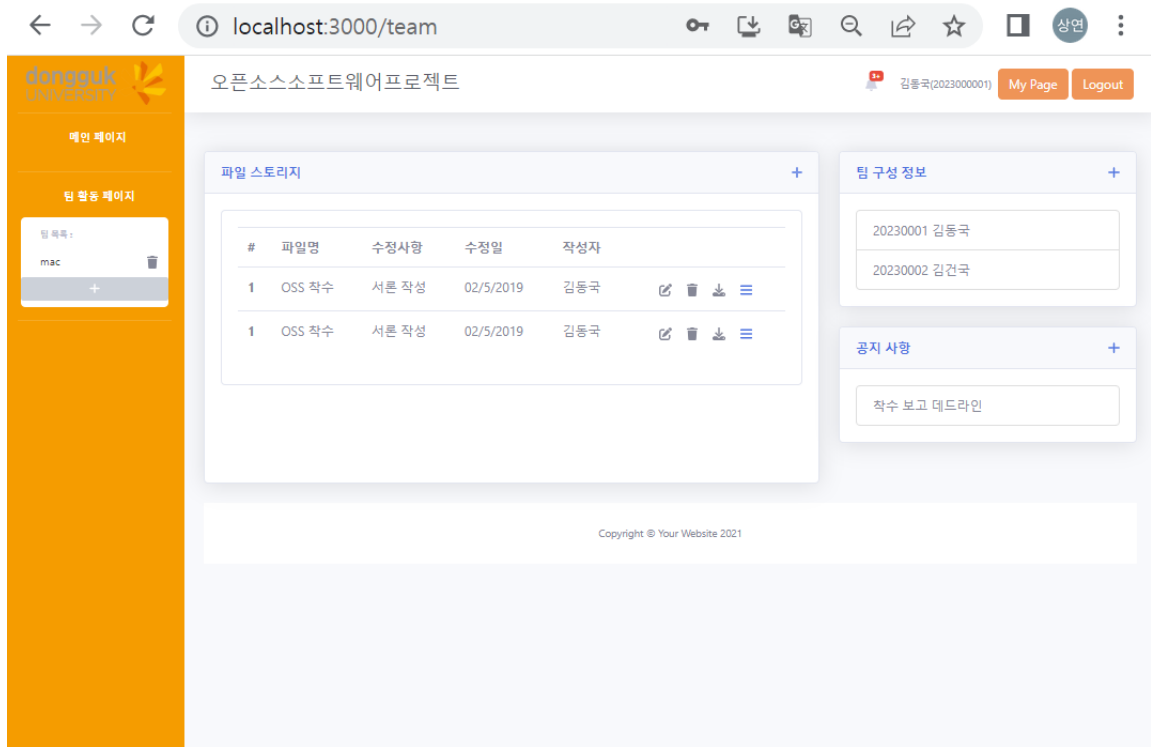
- 초대장 거절 구현

거절 버튼 클릭시 초대장 거절 api에 put요청하여 상태 변경

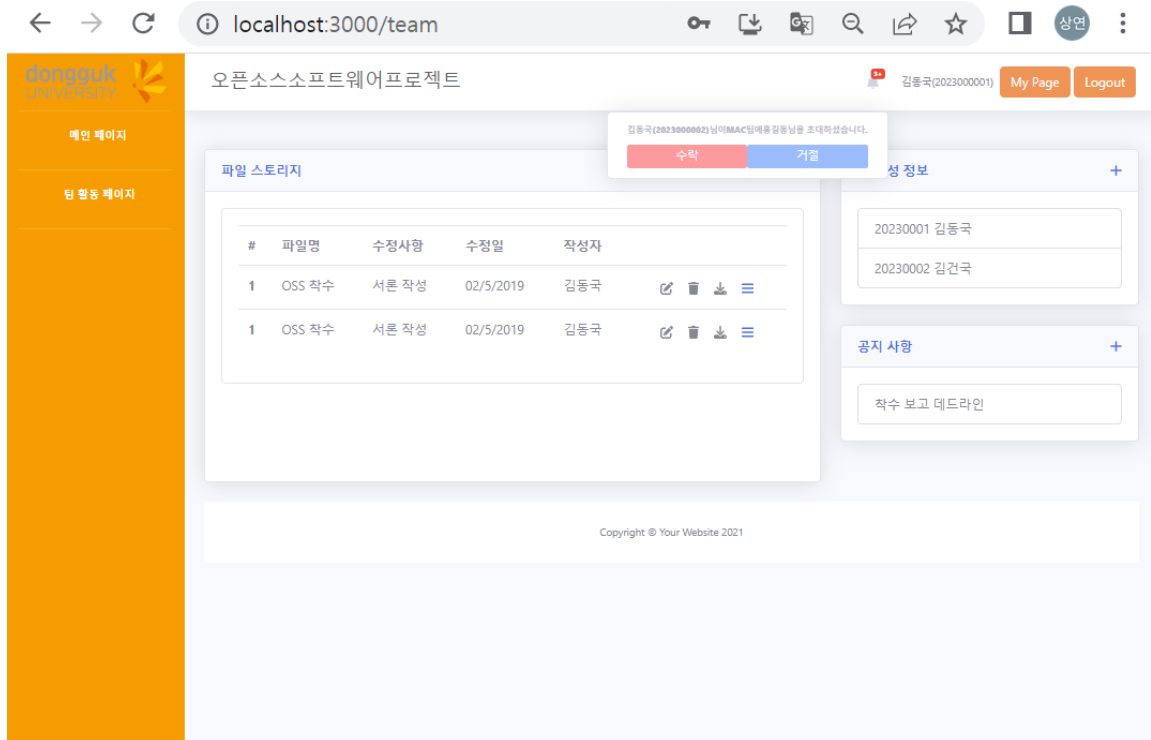
```
▼{invitationId: 7, teamId: 4, teamName: null, leaderId: 1, leaderName: null,
  fellowId: 2
  invitationId: 7
  isAccepted: null
  leaderId: 1
  leaderName: null
  teamId: 4
  teamName: null}
```

b. 안상연

- 팀 활동 페이지 UI 수정

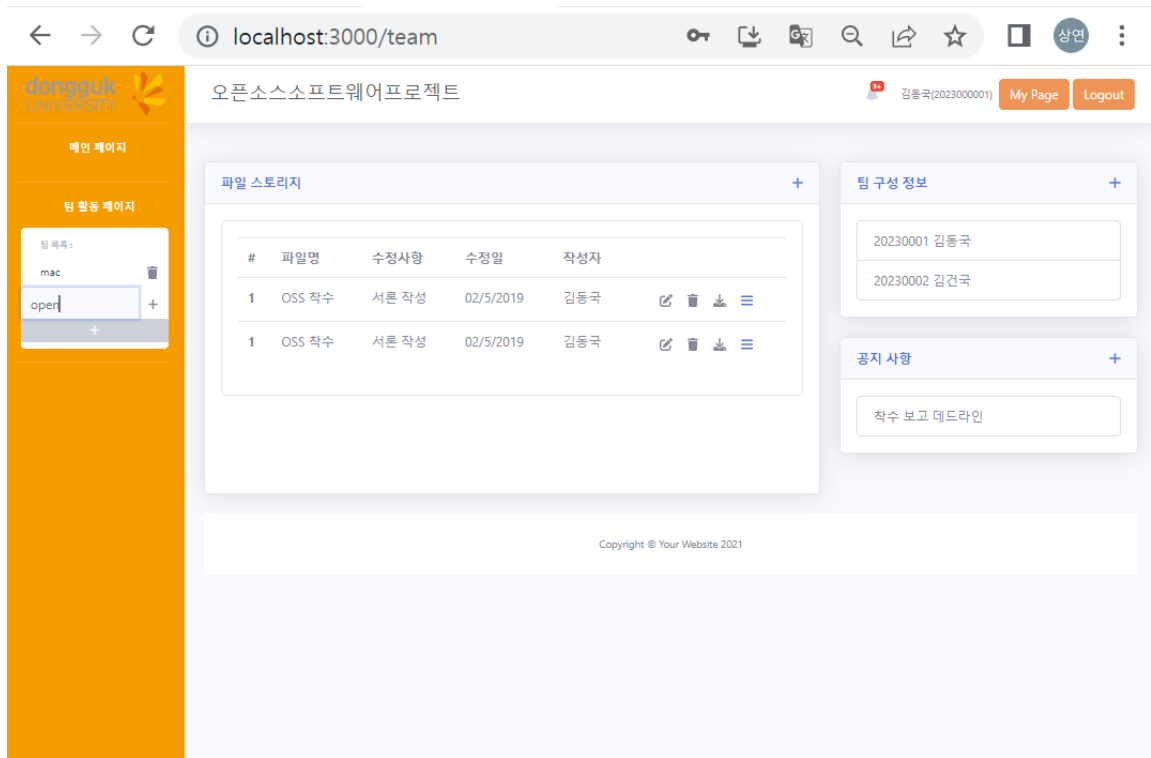


팀 활동 페이지 메뉴에 드롭다운 메뉴 추가. 사용자가 속해 있는 팀 목록과 팀 삭제 버튼, 팀 생성 버튼 구현.

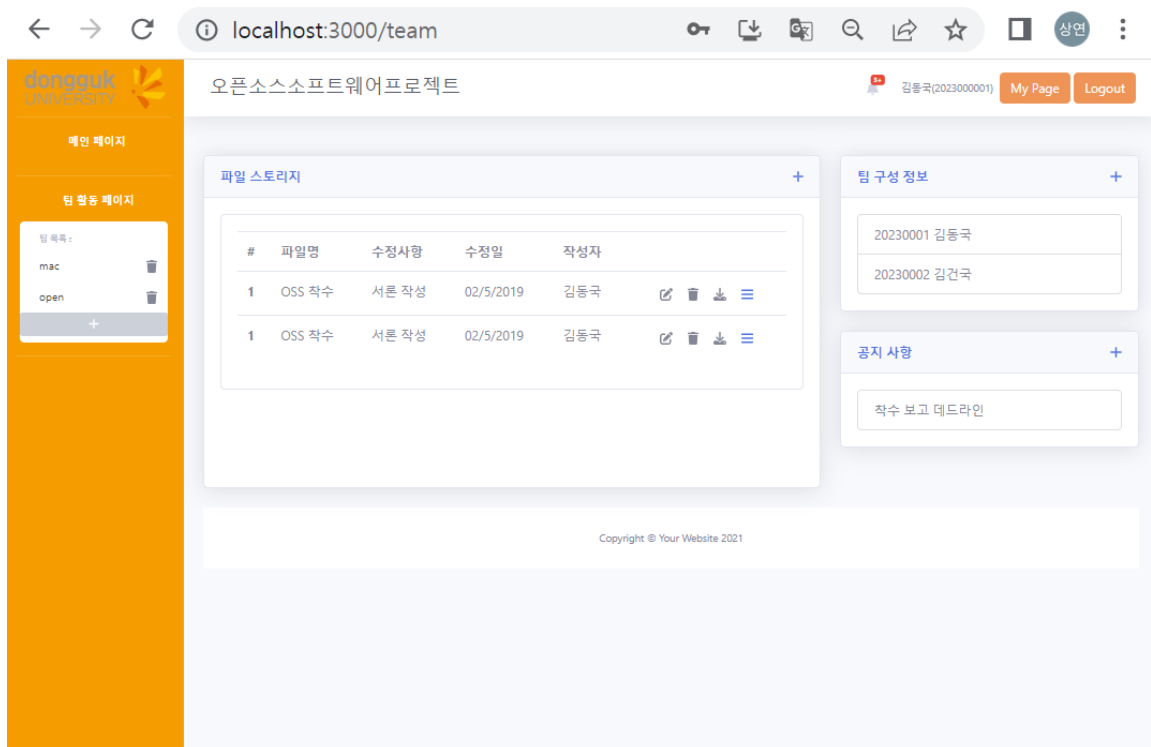


사용자 정보 옆에 벨 모양 아이콘 추가. 아이콘 클릭 시 초대자, 초대된 팀 이름, 수락, 거절 버튼 구현.

- 팀 생성 기능 구현

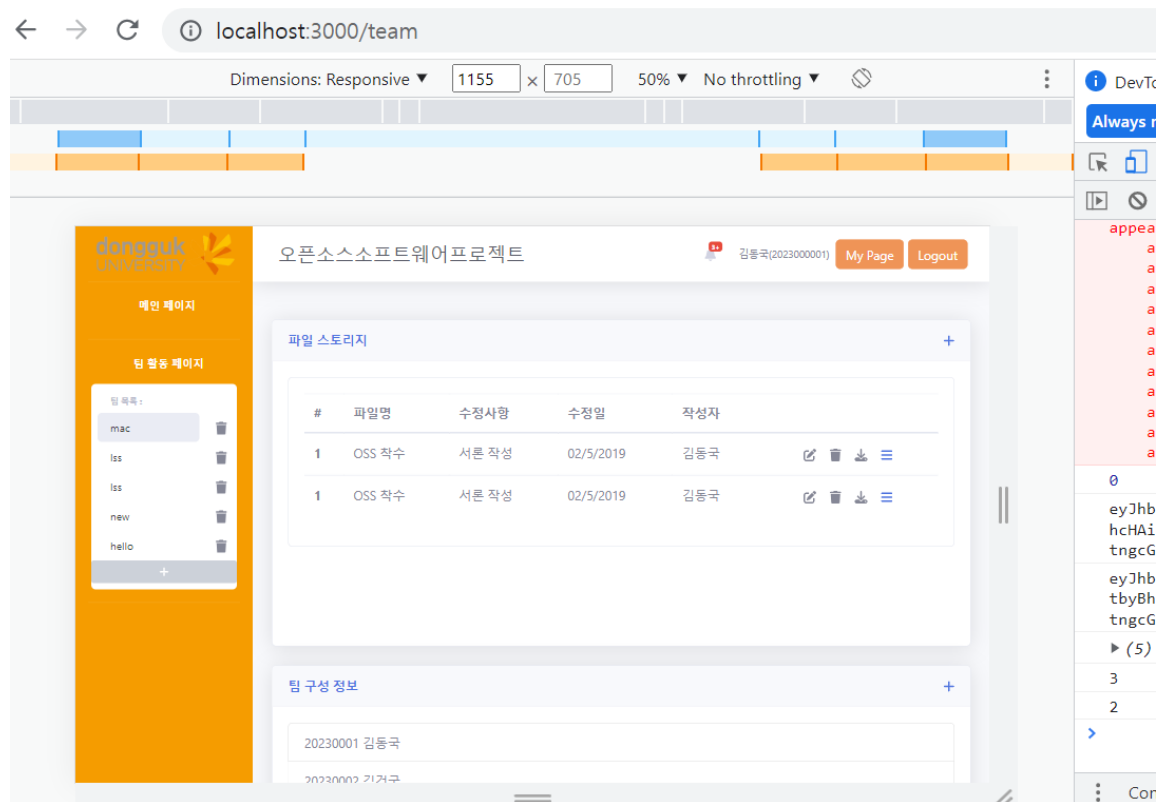


플러스 모양의 팀 생성 버튼 클릭 시, 버튼 위로 새로운 팀명을 입력할 수 있는 입력창과 생성 버튼 구현.



팀명 입력 후 입력창 오른쪽의 생성 버튼 클릭(혹은 엔터키) 시 위와 같이 팀 생성.

- 팀 목록 출력 구현

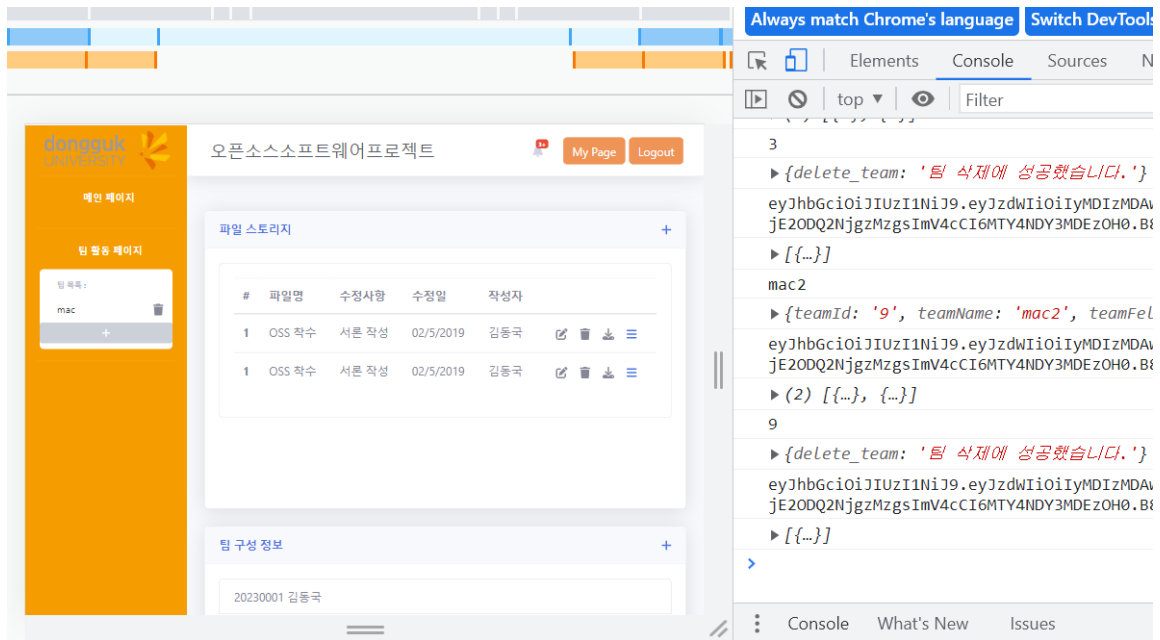


사용자가 속해 있는 팀이 하나 이상일 경우, 생성 순서대로 드롭다운 메뉴에 출력.

팀 버튼을 클릭함에 따라 오른쪽 메인 화면의 세 섹션 정보가 달라질 예정. (팀 아이디를 이용해 구현)

위 그림은 클릭한 팀(ex. mac)의 팀 아이디(ex. 2)를 콘솔에 출력해본 결과.

- 팀 삭제 구현



팀 이름 옆에 삭제 버튼을 클릭 시, mysql과 UI에서 모두 삭제.

3. 향후 진행 사항

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	파일 관련 기능 구현	최필환									
팀 활동 페이지	개별팀 조회 및 초대장 요청 및 응답 처리	민한결									
	팀 목록 조회 요청 및 응답 처리	안상연									
	공지사항 기능 구현, 파일 페이지 처리	최필환									
	공지사항 요청 및 응답 처리	민한결									
	파일 스토리지 요청/응답	안상연									
	통합 테스트, 배포	최필환									
최종 수정 및 발표		모 두									

a. 최필환

- AWS EC2 생성
- 파일 페이지 처리
- 배포

b. 민한결

- 초대장 발송시 이미 초대한 학생 필터링 백엔드 보강

- 팀장이 발송한 초대장 조회 기능 백엔드 보강
- 초대장 기능 프론트 보완
- 공지사항 기능 프론트 구현

c. 안상연

- 파일 조회 프론트 구현
- 파일 등록 프론트 구현
- 파일 삭제 프론트 구현
- 파일 이력 조회 프론트 구현