

5월 4주차 회의록

팀	MAC
참석자	한결 민(상) 상연 안 최필환
장소	원흥관 3층 아리수
시간	23.05.28 13:00~15:00

팀 주제

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

주요 안건

- 진행도 파악
- 프로젝트 변경 사항
- 진행 사항 파악

1. 진행도 파악

분류	내용	담당	4월4주	5주	5월1주	2주	3주	4주	5월5주	6월1주	6월2주
웹캠 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									
웹캠 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									
	리프레시 토큰 구현, DB 암호화	민한결									
	파일 스토리지 요청/응답	안상연									
	통합 테스트, 배포	최필환									
	최종 수정 및 발표	모 두									

2. 진행 사항 파악

1. 백엔드

a. 최필환

- EC2 생성 및 탄력적 IP 주소 할당

i-0316da2010ab54f4e (dguLMS)에 대한 인스턴스 요약 정보		
less than a minute 전에 업데이트됨		
인스턴스 ID i-0316da2010ab54f4e (dguLMS)	퍼블릭 IPv4 주소 43.200.84.136 개방 주소법	프라이빗 IPv4 주소 172.31.40.93
IPv6 주소 -	인스턴스 상태 ○ 중지됨	퍼블릭 IPv4 DNS ec2-43-200-84-136.ap-northeast-2.compute.amazonaws.com 개방 주소법
호스트 이름 유형 IP 이름: ip-172-31-40-93.ap-northeast-2.compute.internal	프라이빗 IP DNS 이름(IPv4만 해당) ip-172-31-40-93.ap-northeast-2.compute.internal	탄력적 IP 주소 43.200.84.136 [퍼블릭 IP]
프라이빗 리소스 DNS 이름 응답 IPv4(A)	인스턴스 유형 t2.micro	AWS Compute Optimizer 찾기 권장 사항을 위해 AWS Compute Optimizer에 옵트인합니다. 자세히 알아보기
자동 할당된 IP 주소 -	VPC ID vpc-04a6f52e3f6ea838e	Auto Scaling 그룹 이름 -
IAM 역할 -	서브넷 ID subnet-0f220a4e0026e4531	
IMDSv2 Optional		

인스턴스의 퍼블릭 ip주소가 고정될 수 있도록 탄력적 IP주소 부여

- 응답 수정

```
@Transactional
public ResponseEntity write(String studentId, Long teamId, NoticeRequest noticeRequest){
    try{
        User user = userRepository.findById(studentId).get();
        Member member = memberRepository.findByIdByUserIdAndTeamId(user.getId(), teamId).orElse( null);
        if (member==null){
            return ResponseEntity.badRequest().body(ResponseDto.builder().error(Collections.singletonMap("write_notice", "올바른 사용자가 아닙니다.")).build());
        }
        String content = noticeRequest.getContent();
        if (content==null){
            return ResponseEntity.badRequest().body(ResponseDto.builder().error(Collections.singletonMap("write_notice", "내용을 입력하십시오.")).build());
        }
        Notice notice = noticeRequest.toEntity();
        notice.setMember(member);
        noticeRepository.save(notice);
        NoticeResponse dto = NoticeResponse.builder()
            .noticeId(notice.getNoticeId())
            .noticeContent(notice.getContent())
            .build();
        return ResponseEntity.ok().body(Collections.singletonMap("write_notice", "공지사항을 생성했습니다.));
    }catch (Exception e){
        log.error(e.getMessage());
        ResponseDto responseErrorDto = ResponseDto.builder().error(Collections.singletonMap("write_notice", "공지사항 생성에 실패했습니다.")).build();
        return ResponseEntity.badRequest().body(responseErrorDto);
    }
}
```

이전 피드백을 반영하여, 어떤 부분이 에러가 나는지에 대한 자세한 에러메세지 전달

- 파일 다운로드 API수정 및 분리

기존에는 하나의 api를 통해 파일버전과 파일을 다운받았는데, 아래와 같이 분리하였음.

```
//파일 버전 다운로드
new *
@GetMapping("/{team}/{teamId}/fileDownload/{fileId}/{fileVersionId}")
public ResponseEntity downloadFileVersion(@AuthenticationPrincipal String studentId, @PathVariable Long fileId, @PathVariable Long fileVersionId, @PathVariable Long teamId){
    return fileDownloadService.downloadFileVersion(studentId, fileId, fileVersionId, teamId);
}

//파일 다운로드
+ thisshwan2
@GetMapping("/{team}/{teamId}/fileDownload/{fileId}")
public ResponseEntity downloadFile(@AuthenticationPrincipal String studentId, @PathVariable Long fileId, @PathVariable Long teamId){
    return fileDownloadService.downloadFile(studentId, fileId, teamId);
}
}
```

- 프론트에서 다운로드시 파일명 오류 수정

<기존방식>

웹 페이지에 다운로드 버튼에 S3URL만 할당.

<발생 문제>

업로드시 파일명을 유니크한 파일명으로 업로드 하기 때문에 다운로드시 파일 명이 변경되는 문제.

<해결방안>

```
FileVersionDto fileVersionDto = FileVersionDto.builder().
    s3Url(findFile.getS3FileUrl()).
    fileName(findFile.getRealFileName()).build();

//프론트 단에서 이름을 원래 이름으로(dto에 있는이름으로 바뀌어서 다운)
return ResponseEntity.ok().body(fileVersionDto);
```

다운로드 버튼 클릭시 API를 호출 → DTO자체를 응답.(해당 DTO에는 원래 파일명을 가져와서 빌드)

b. 민한결

- 팀원 초대 수락 거절 기능 확립

```
invitation.setIsAccepted(false);
InvitationDto invitationResponseDto = InvitationDto.builder()
    .invitationId(invitation.getId())
    .leaderId(invitation.getLeader().getId())
    .fellowId(invitation.getFellow().getId())
    .teamId(invitation.getTeam().getId())
    .build();
invitationRepository.delete(invitation);
```

초대 거절의 경우 거절과 동시에 삭제하도록 수정

초대 수락의 경우 수락 후 상태만 변경하여 구별 가능하도록 기존처럼 유지

- 이미 초대된 학생을 초대 모달 학생 목록에서 필터링하도록 수정

```
users = users.stream()
    .filter(user -> !memberUsers.contains(user))
    .filter(user -> !invitedUsers.contains(user))
    .collect(Collectors.toList());
```

Team Service에 필터링 로직 추가 및 Team Controller에 필터링한 학생 목록 조회 api 추가

- noticeDto 및 Service의 필드에 notice 추가하여 프론트 작업시 가독성 향상

```
public class NoticeResponse {

    no usages
    private Long noticeId;
    no usages
    private String noticeContent;
}
```

2. 프론트엔드

a. 민한결

- 초대장 수락 거절 로직 수정 사항 프론트 반영



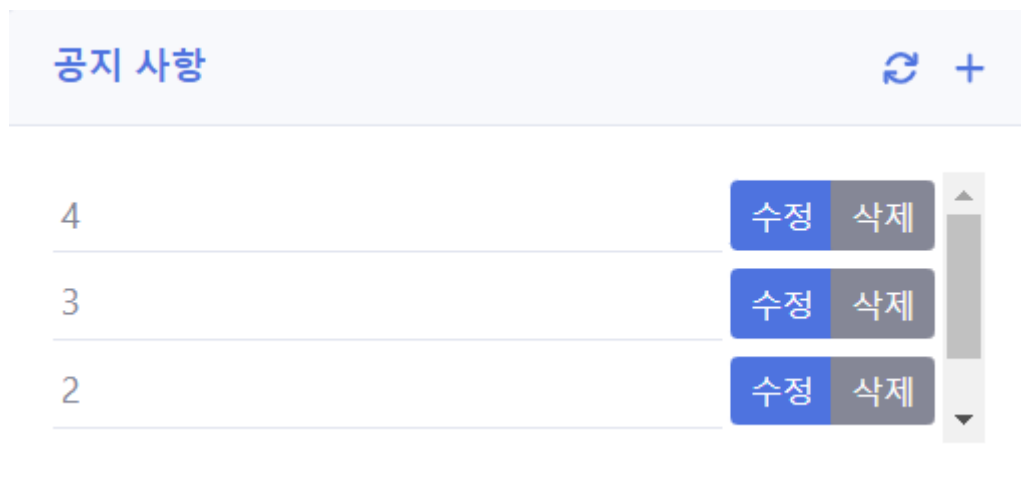
초대장 수락시 수락 버튼 비활성화 및 거절 버튼을 삭제 버튼으로 변경

초대장 개수는 수락하지 않은 초대장만 표시하도록 변경

- 이미 초대하거나 팀원인 학생 필터링한 학생 리스트 조회 api로 초대장 모달 수정

```
const fetchUserList = () => {
  const token = sessionStorage.getItem("token");
  fetch(`http://localhost:8080/team/${teamId.id}/user`, {
    headers: {
      Authorization: `Bearer ${token}`
    }
  })
}
```

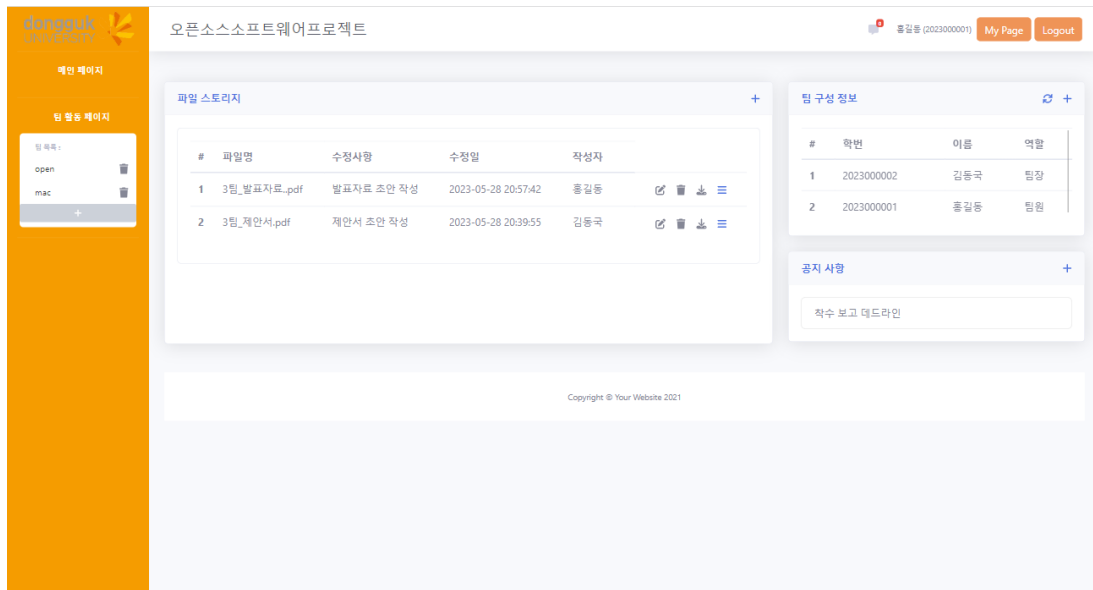
- 공지사항 컴포넌트 기능 구현



1. 안상연

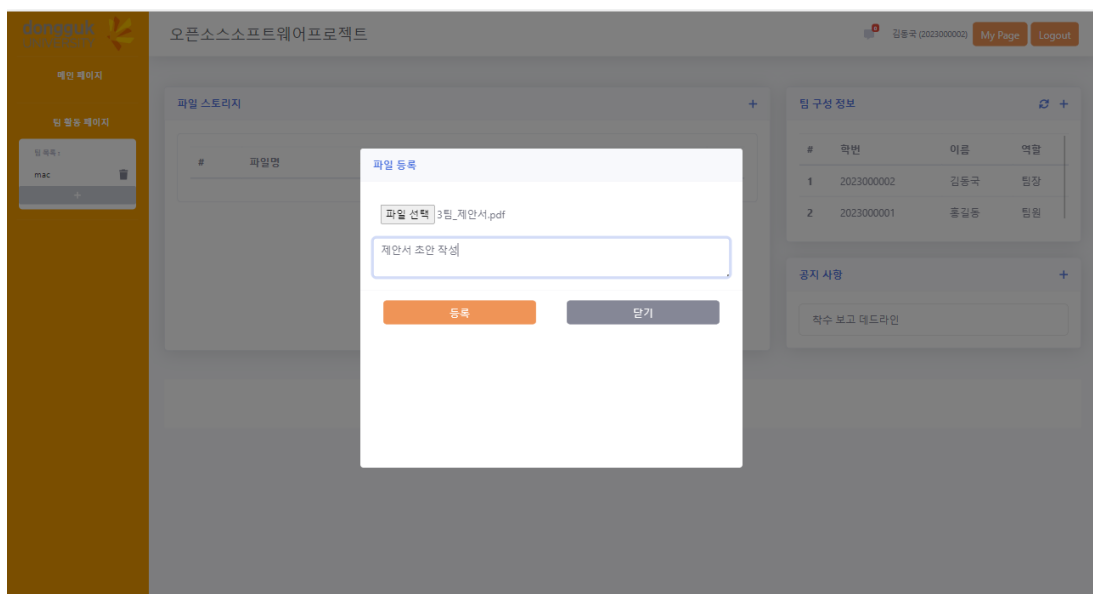
API 명세서를 바탕으로, fetch함수로 원격 API 호출, 토큰으로 사용자 정보 접근하여 구현.

- 파일 스토리지 섹션에 목록 출력



teamId를 이용해 선택한 팀에 따른 파일 목록 출력

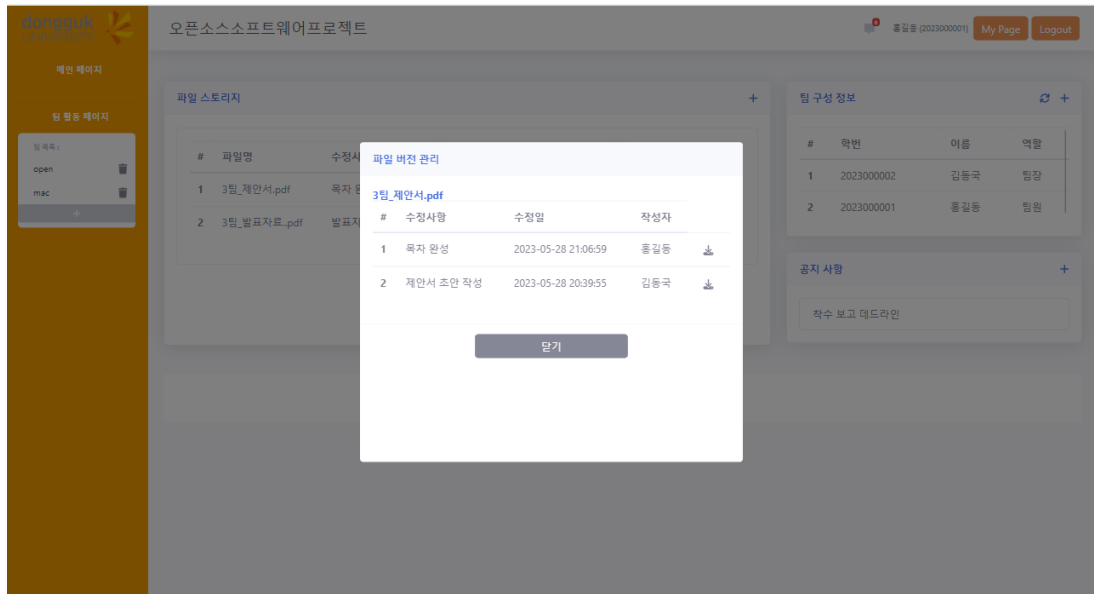
- 파일 업로드



파일 스토리지 섹션 우측상단의 플러스 버튼 클릭 시 파일 업로드 모달이 띄워지고, 파일 선택 및 수정 사항 입력 후 등록 버튼 클릭 시 추가한 파일 포함해 파일 목록 업데이트.

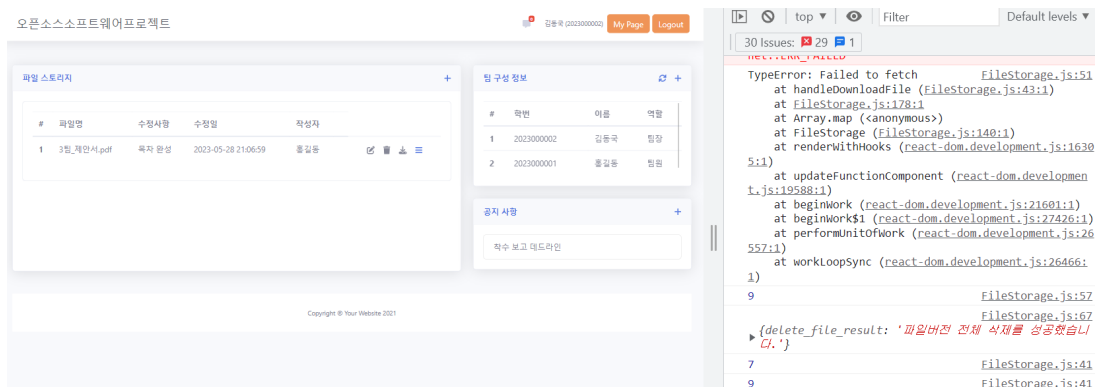
버전별 파일 업로드 모달 UI도 위와 같이 구현.

- 파일 이력 조회



버전별 파일 이력 조회 모달 구현. 해당 버전의 파일명을 상단에 보여주고, 파일 이력을 리스트 형식으로 출력. 우측 다운로드 버튼을 눌러 파일 저장 가능.

- 파일 삭제



파일 버전 삭제 구현. 삭제 후 파일 리스트 다시 업데이트.

- 로그인 없이 팀활동 페이지에 먼저 접근하는 등 발급된 토큰이 존재하지 않을 경우와 토큰 유효 시간이 만료된 경우 다시 로그인 페이지로 리디렉션되도록 구현.

3. 향후 진행 사항

a. 최필환

- 도커 활용 EC2 빌드
- 프론트와 연동 마무리

b. 민한결

- 프론트 에러 처리
- 리프레시 토큰 구현
- DB 암호화

c. 안상연

- 파일 스토리지 마무리
- 유저 친화적 인터페이스 디테일 수정
- 각 사용자 입력 에러 처리