14주차 팀 주간회의

를 팀 주제	동국대학교 대관신청 웹 시스템
≡ 팀명	ProdigyProject
◈ 유형	팀 주간 회의
■ 일시	@2023년 6월 1일
를 참석자	
**	

🤨 회의 주제

- 1. 관리자 모드, 마이페이지 구현
- 2. 프론트,백 연동(연동 과정에서의 오류 해결)

회의 내용

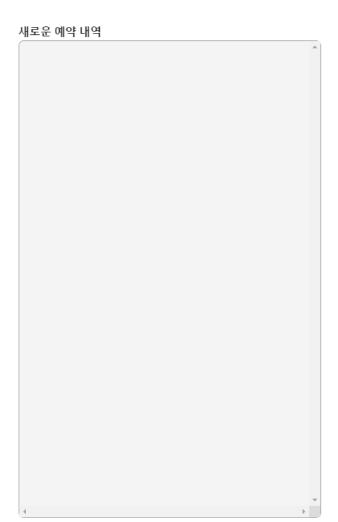
0. 프로젝트 타임라인

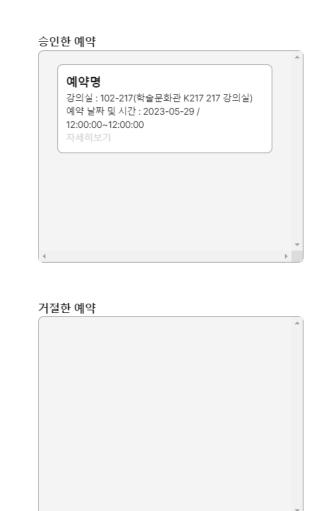
Gantt	Chart	3월~4월 2주	3주	4주	5월 1주	2주	3주	4주	6월 1주	2주	3주
Gantt Chart 일정		3 결~4 결 2 ㅜ	>⊤	4+	2월 1十	2⊤	3ㅜ 중간발표	4十	0월 1구	2⊤	3ㅜ 최종발표
작업	담당자										
오픈소스 분석	김유리, 홍범선, 홍서이										
기존 컴포넌트 수정	김유리	·									
컴포넌트 구현	김유리										
기존 페이지 수정	김유리										
새 페이지 구현	김유리										
기능 구현	김유리, 홍범선										
API 설계	홍범선										
API 구현 및 테스트	홍범선										
front - db 연동	홍범선, 홍서이										
AWS 서버 연동 및 배포	홍범선, 홍서이										
DB 설계	홍서이										
DB 구현	홍서이										
발표준비	김유리, 홍범선, 홍서이										
	김유리, 홍범선, 홍서이										
-											

1. 진행상황

김유리(Frontend)

• 관리자 모드 UI 제작 및 기능 추가





```
const [approval, setApproval] = useState([]);
const [denied, setDenied] = useState([]);
const handleApprovalClick = (reserv, index) => {
 if(window.confirm("해당 예약을 승인하시겠습니까?")) {
    reserv.approval = "T";
    postApproval(reserv.approval, reserv.id)
    .then(response => {
     if (response.status === 200) {
       setWait(wait.filter((_, i) => i !== index));
       setApproval(prev => [...prev, reserv]);
   })
    .catch(error => {
     console.error("Error:", error);
   });
};
const handleDenyClick = (reserv, index) => {
 if(window.confirm("해당 예약을 거절하시겟습니까?")) {
    reserv.approval = "F";
    postApproval(reserv.approval, reserv.id)
    .then(response => {
     if(response.status === 200) {
       setWait(wait.filter((_, i) => i !== index));
       setDenied((prev) => [...prev, reserv]);
    })
    .catch(error => {
    console.error(error)
   });
```

• 로그아웃 기능 추가

```
const handleLogout = () => {
    logout()
    .then(response => {
        if(response.status === 200) {
            router.push("/");
        }
    })
    .catch(error => {
        console.error(error);
    })
```

• 사용자가 시간 선택할 때 연속된 시간 체크 수정 및 최대 4시간 조건 추가

```
const handleTimeSelection = (time) => {
 const newSelectedTimes = [...selectedTimes];
 if (newSelectedTimes.includes(time)) {
   // 이미 선택된 시간인 경우 해제
   const index = newSelectedTimes.indexOf(time);
   newSelectedTimes.splice(index, 1);
   newSelectedTimes.push(time);
 const isSequential = newSelectedTimes
   .map(selectedTime => selectedTime.split(':').map(time => parseInt(time)))
   .sort((a, b) => {
     if (a[0] !== b[0]) {
       return a[0] - b[0];
       return a[1] - b[1];
   .every((time, index, array) => {
     if (index === 0) {
       return true;
     const prevTime = array[index - 1];
    return (time[0] === prevTime[0] && time[1] === prevTime[1] + 30) || (prevTime[1] === 30 && time[0] === prevTime[0] + 1);
 if (newSelectedTimes.length <= 8 && isSequential) {</pre>
   setSelectedTimes(newSelectedTimes);
```

• From(회원가입, 로그인, 예약) 내용 서버로 보내는 코드 추가

```
const apply = async (data, responseData, user) => {
   const requestData = {
     room_id: responseData.room,
     date: responseData.selectedDate,
     reason: data.reason,
     event_name: data.name,
     people: data.headcount,
     group_name: data.organization,
     event_content: data.outline,
     user_id: user.id,
     approval: "W",
     start_time: responseData.startTime,
     end_time: responseData.endTime,
   console.log(requestData);
   const response = await api.post(`/buildings/${requestData.date}/${responseData.buildingname}/${requestData.room_id}/reservation`, requestData);
    return response;
   console.error(error);
const schema = yup.object().shape({
reason: yup.string().required("신청사유를 입력해주세요."),
 name: yup.string().required("예약명(행사명)을 입력해주세요"),
 headcount: yup.string().required("인원수를 입력해주세요. "), organization: yup.string().required("단체명 또는 주최자명을 입력해주세요"),
 outline: yup.string().required("행사개요를 입력해주세요. ")
```

• 서버로부터 데이터 받는 코드 추가

홍범선(Backend)

- 프론트,데이터베이스 연동 완료
- 기존 페이지 연동과정에서의 오류 발생 해결 및 기능개선
 - 。 클라이언트에서 서버로 jwt전송안되는 오류 수정(클라이언트측 헤더에 지정)

```
const response = await api.get("/mypage",{
    headers:{
       cookie: context.req.headers.cookie || '',
    }
});
```

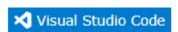
- 。 토큰인증에서 토큰의 정보로 학생,관리자 구별가능하게 추가
- 。 예약 db에 저장시 클라이언트에서 예약id보내지 않아도 설정되게 변경
- 회원가입, 예약 정보 저장 오류 수정 → 저장되는 것 확인
- 관리자모드, 마이페이지 연동
 - 。 관리자모드와 마이페이지 jwt인증하도록 구현

```
router.get('/mypage',verifyToken, myReservation);
router.get('/admin',verifyToken, getReservation);
router.post('/admin',verifyToken, approveReservation);
```

- ∘ 관리자모드 get:승인여부 정하지 않은 예약, post: 승인결과 db저장
- 。 마이페이지 get: jwt에 담겨있는 user의 id에 해당하는 예약

홍서이(Database)

- 데이터베이스 삽입 관련 코드 작성
 - 。 회원가입 유저 정보를 데이터베이스에 저장하는 코드 작성
 - 。 요일, 강의실 정보에 대하여 예약정보, 수업 정보를 불러오는 함수 작성
 - 。 예약 정보 저장시 날짜로 요일을 계산하여 저장하는 함수 추가
- 데이터베이스 수정
 - 예약 신청 내역에 대한 상태 추가(W:대기중, T:승인, F:거부)
 - 。 DB.Lecture의 중복값 제거
- README.md에 사용한 기술, 프로젝트 설명 추가



Role	Name	Skills						
Frontend	김유리							
Backend	홍범선							
DBA	홍서이	Amazon AWS Python MySQL ©X Express						

2. 앞으로의 계획

김유리(Frontend)

- 최종 발표 준비
- 피드백 및 수정

홍범선(Backend)

- 최종 발표 준비
- 피드백 및 수정
- AWS배포

홍서이(Database)

- AWS 이용해서 프로젝트 배포하기
- README.md 파일에 프로젝트 세부 내용 설명, 사진 추가하기
- 최종 발표 준비

➡ 다음 회의 내용

- 개인별 진행 상황 공유
- 최종발표
- 6/5(월) 교수님 피드백 내용에 대한 내용 논의
- Readme파일
- 시연내용