



# WFF WE FIND FLOWER

-  
우리 집에 왜 왔니?  
우리는 꽃을 찾으러 왔지!



CONTENTS

01 프로젝트 개요

02 기획의도

03 프로젝트 진행 순서

04 개발환경&적용기술

05 ERD

06 형상관리

07 Gantt Chart

08 기능명세서

09 주요 코드

10 코드 리뷰

11 후기



# Contents





... We Find Flower ...

# WFF : WE FIND FLOWERS!

## 프로젝트 개요



- **프로젝트 명 :** WFF
- **개발 목적 :** 온라인 상품 판매 쇼핑몰 반응형 웹 개발
- **프로젝트 소개 :**  
온라인 꽃 판매 쇼핑몰  
꽃말, 기념일, 이벤트 등 다양한 컨텐츠를  
바탕으로  
꽃 연관 상품을 제공하는 사이트입니다.
- **주 타겟층 :** 20-30대 남녀
- **주요 기능 :** 회원가입, 로그인, 게시판



# 기획의도

코로나 19 발생 이후 실내 공간에서의 활동 빈도가 높아짐에 따라  
화훼 관련 온라인 관심도가 증가했으며 화훼 소비 트렌드도 변화하고 있습니다.  
WFF는 트렌드를 반영하여 특별하고 감성적인 서비스를 제공합니다.

## 당신의 마음을 꽃으로 대신 전해드립니다.

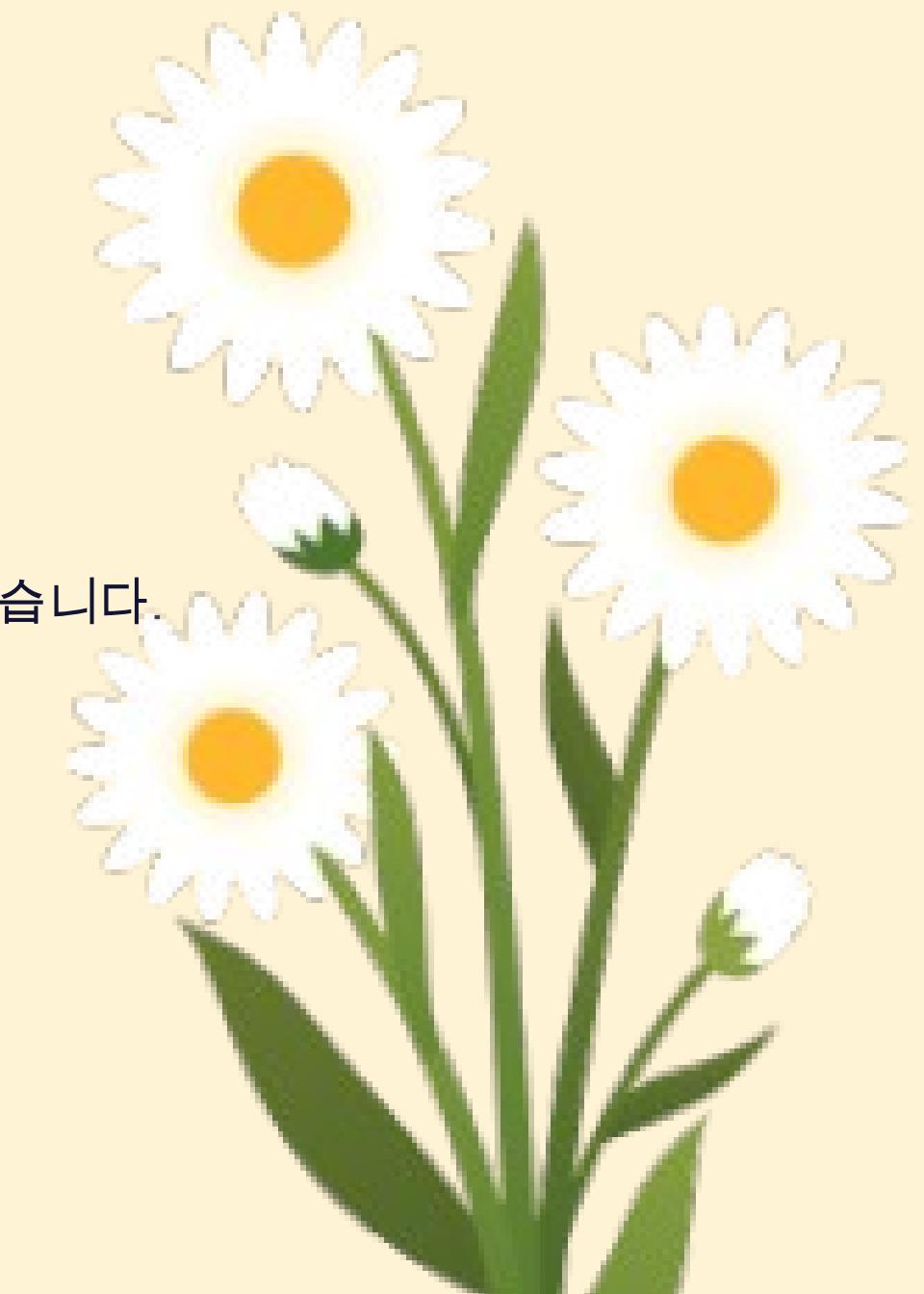
WFF 는 꽃의 의미를 담아 사랑하는 사람에게 마음을 전달할 수 있는 꽃을 선물해보세요.

## 예쁜꽃만 파는게 아니에요!

나와 우리집을 이롭게 하는 꽃,  
아픈 꽃을 받아서, 우리가 그걸 케어해서 다시 되돌려주는 시스템 (케어플라워서비스) 제공하고 있습니다.



## 왜 우리집 꽃을 찾을까요?



# 프로젝트 진행 순서

... We Find Flower ...



# 개발환경&적용기술

## FRAME WORK



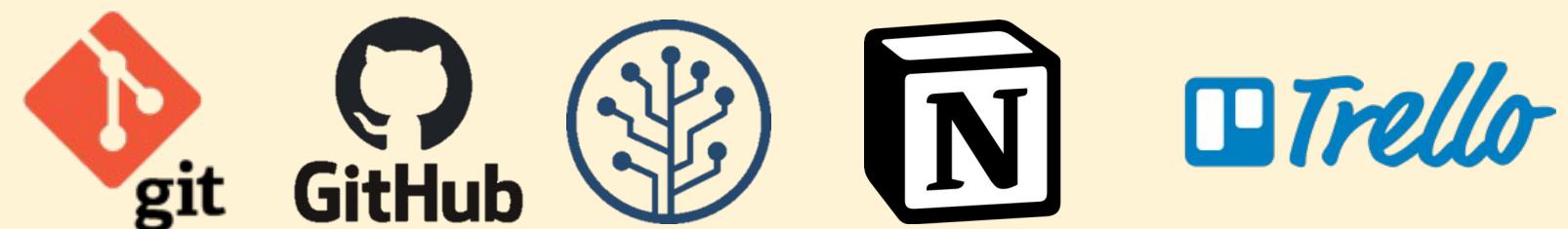
## SERVICE



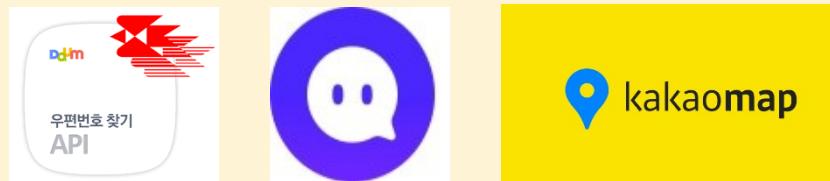
## DEVELOPER TOOL



## 형상관리 / 협업관리



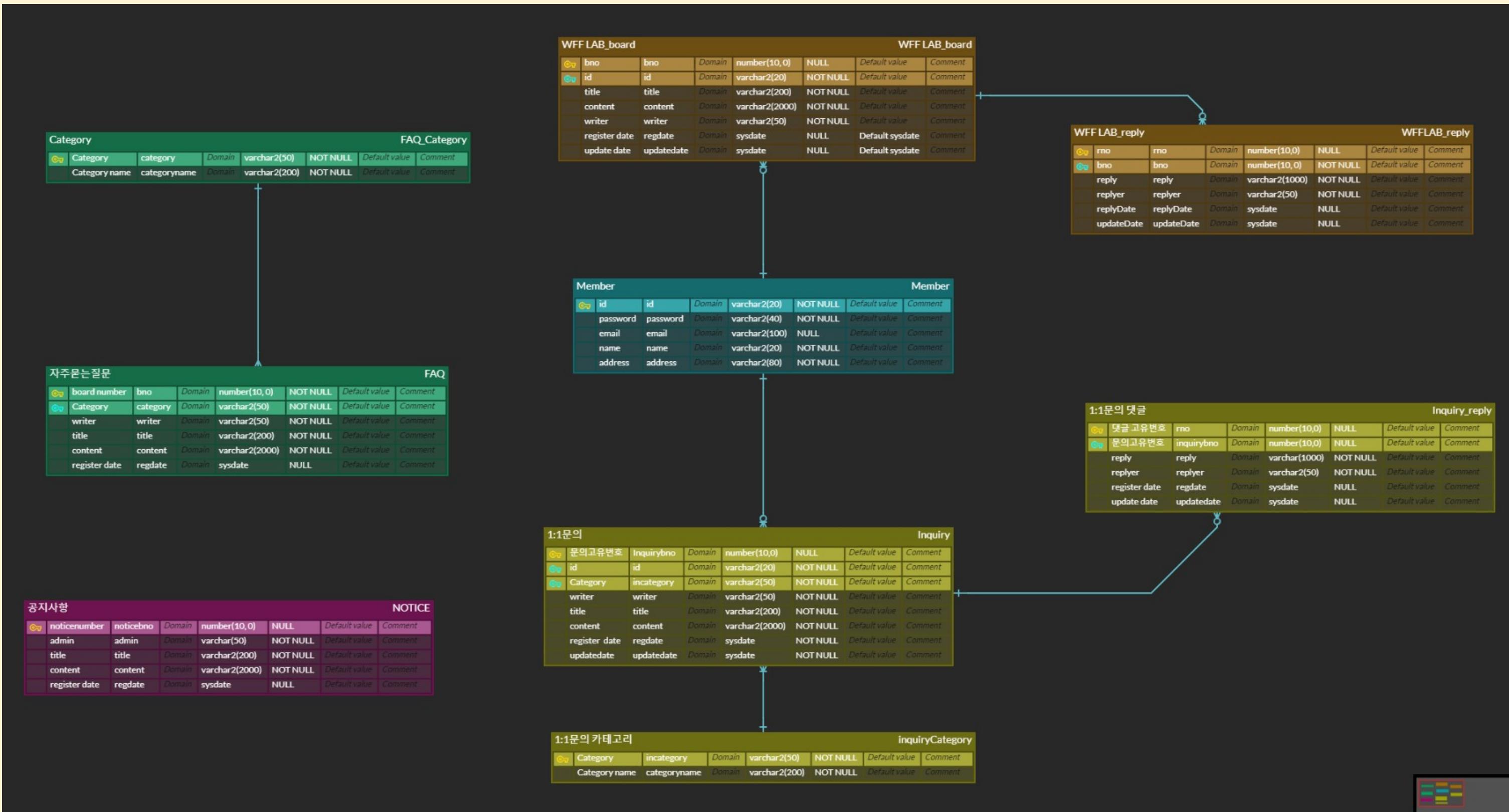
## API



## DATA BASE

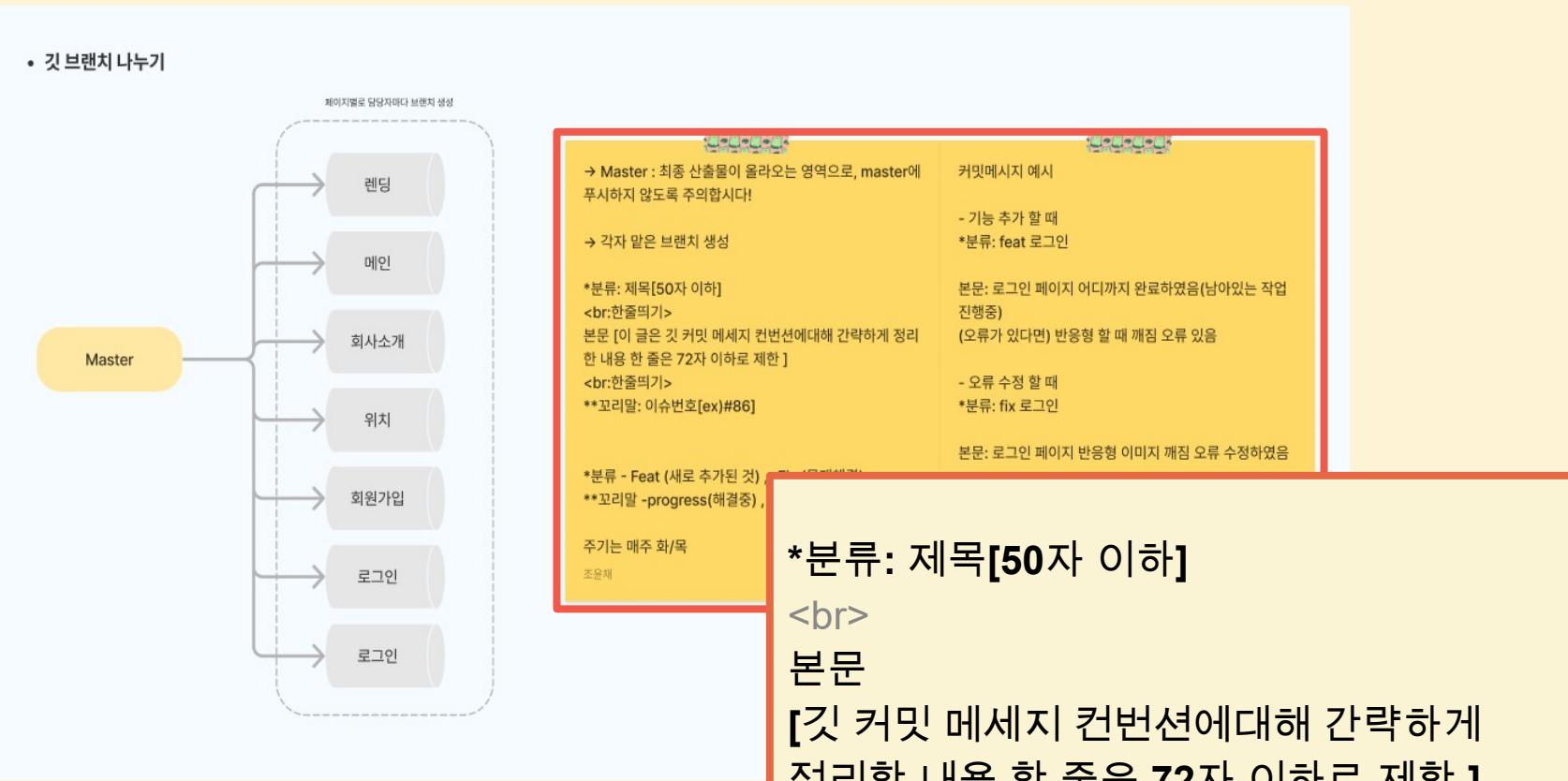


# ERD 설계



# 형상관리

••• We Find Flower •••



## 브랜치 계획 및 컨벤션 규칙

### \*분류: 제목[50자 이하]

<br>

본문

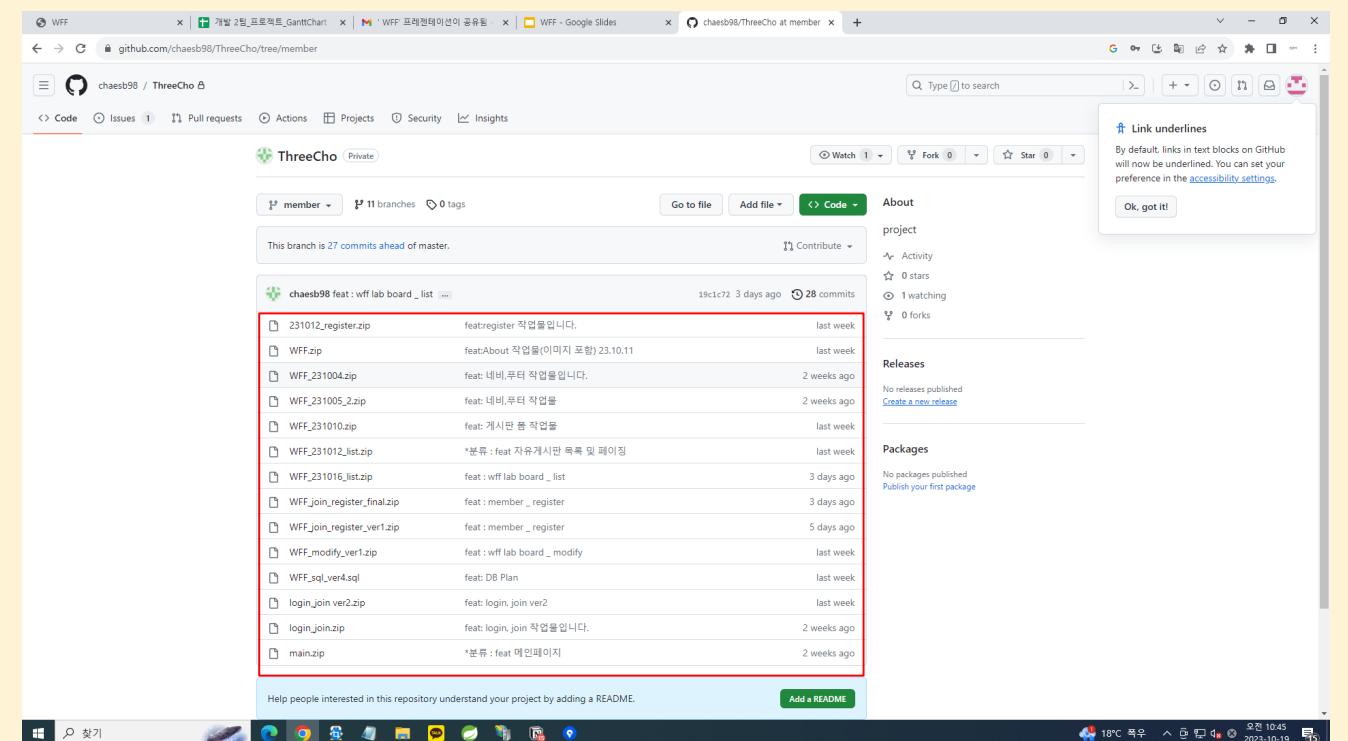
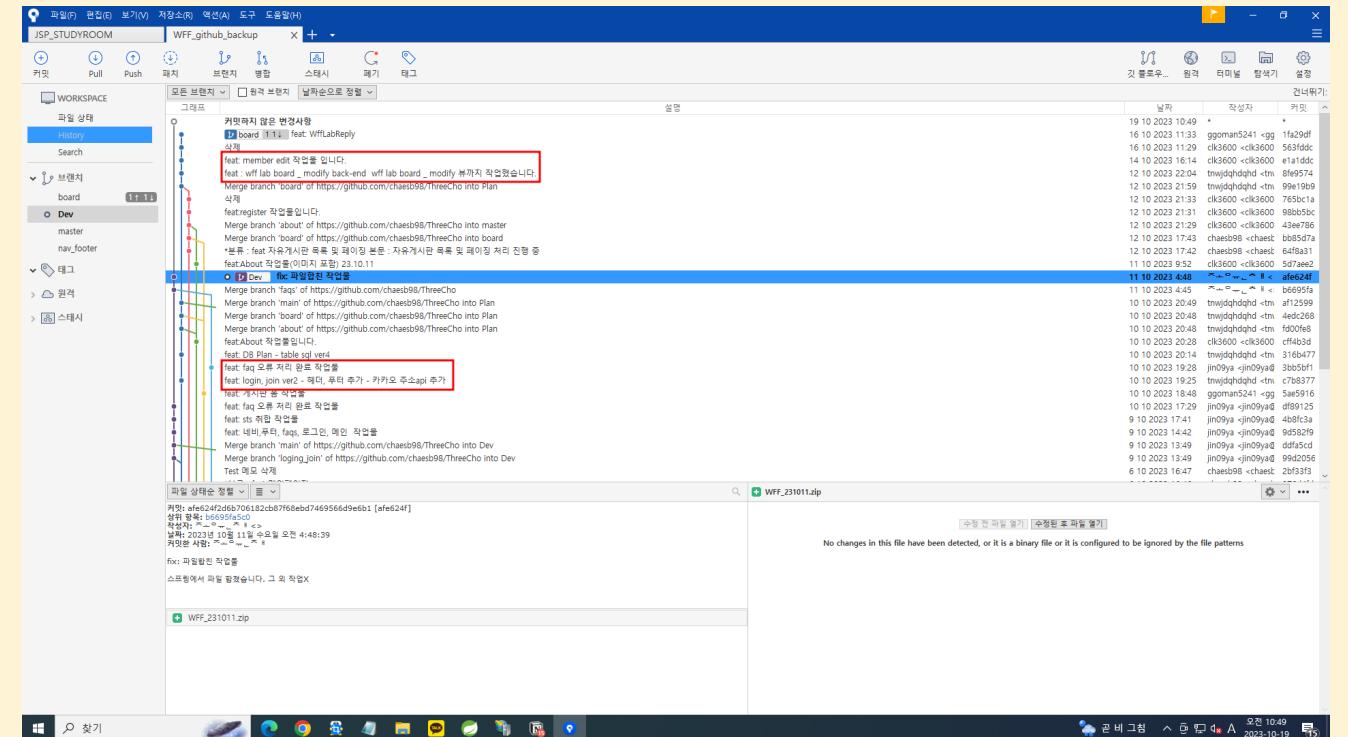
[깃 커밋 메세지 컨번션에 대해 간략하게 정리한 내용 한 줄은 72자 이하로 제한]

<br>

\*\*꼬리말: 이슈번호[ex)#86]

\*분류 - Feat (새로 추가된 것) , -Fix (문제해결)

\*\*꼬리말 -progress(해결중) , -resolve(해결완료)

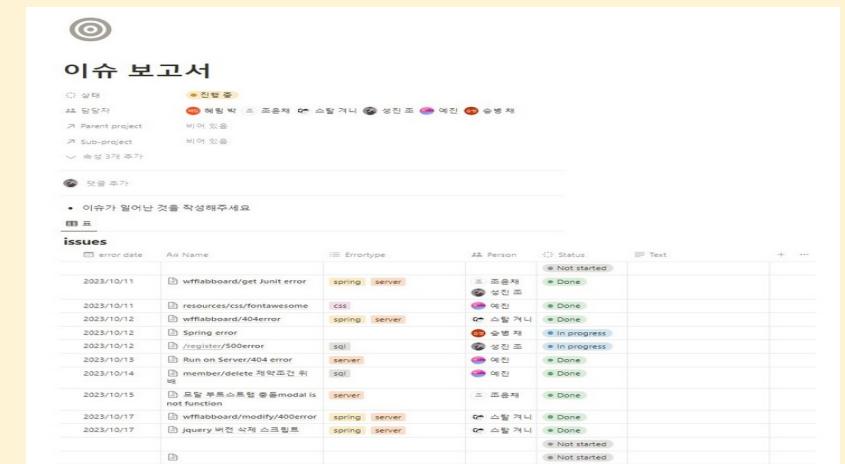
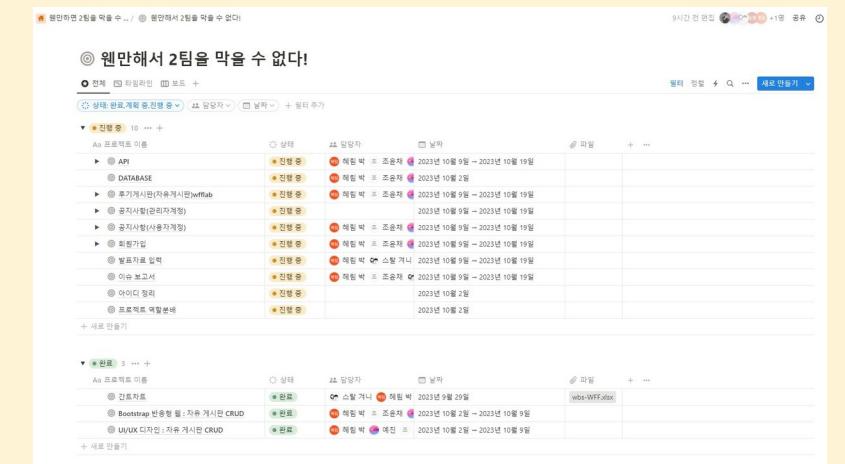


## 07 Gantt Chart

# 협업 관리

## • Notion

- 노션을 이용하여  
프로젝트 역할분배  
아이디 정리,  
이슈 보고서,  
발표자료입력,  
DATABASE 등  
진행상황 공유



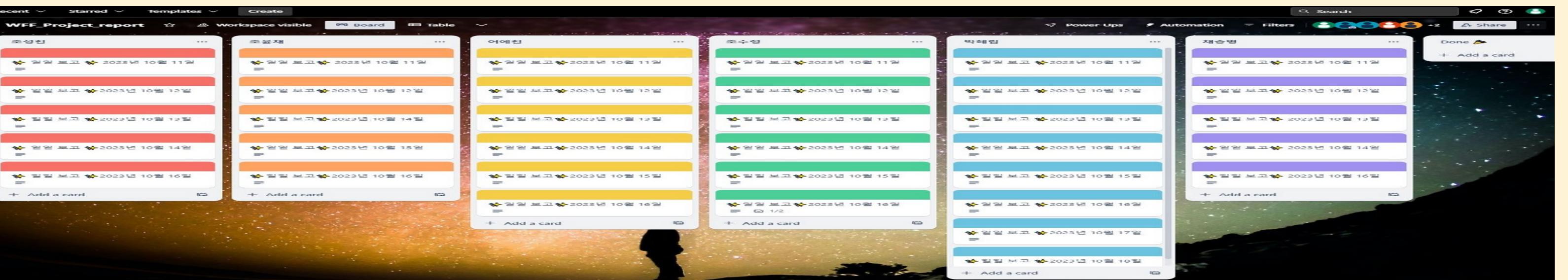
# • Figma

- # 피그마를 이용하여 주제선정을 위한 만다라트 계획표 사용 및 템플릿 선정, 아이디어 회의 공유



## ● Trello

- 트렐로를  
이용하여 일일  
진행상황 공유



WFF

# Gantt Chart

세부적인 업무 분장으로 세부화된  
일감을 분배하여 팀원간 스케줄 조정

시간분배, 단기적 목표를 통한  
능동적인 프로젝트 참여

••• We Find Flower •••

※ 개발 2팀 프로젝트명 = WFF(조윤채, 조성진, 어예진, 조수정, 박혜림, 채승병)							완료여부 표시			프로젝트진행일								
							착수검토	진행중	진행완료	10월			1주차					
담당자	구분	수행사항		시작일	종료일	진행기간	완료여부	10일	11일	12일	13일	14일	15일	16일	17일	18일	19일	
공통	UI/UX 컨셉 기획(Figma)	포트폴리오 : 주제/제목/부제목/기획의도 선정		2023.10.10	2023.10.10	1	진행완료											
	고객 요구 분석(Figma)	주제선정, 요구사항 취합, 요구사항 정의		2023.10.10	2023.10.10	1	진행완료											
	설계(기획, 디자인, ERD)	컨셉도출, 브랜드 기획, UI디자인, UI 설계, ERD 설계		2023.10.10	2023.10.12	3	진행완료											
	테스트	JUnit Test(유닛 테스트)		2023.10.12	2023.10.18	7	진행완료											
	코드리뷰(Code Review)	기술 및 기능 개발 코드리뷰(테스트, 오류수정, 최종테스트)		2023.10.12	2023.10.19	8	진행완료											
	형상관리(Git/Github-소스스트리)	브랜치분리, 지속적인 커밋-머지를 통해 형상관리		2023.10.12	2023.10.18	7	진행완료											
	협업관리(Trello, 노션)	업무 기록을 통한 프로젝트 진척도 공유 및 작업관리, 이슈 트래킹 카드를 통한 문제 해결 및 협업		2023.10.12	2023.10.18	7	진행완료											
조윤채	일정관리(간트차트, Gantt Chart)	세부적인 업무 분장으로 세부화된 일감을 분배하여 팀원간 스케줄 조정		2023.10.10	2023.10.19	10	진행완료											
	Front-End UI/UX 디자인: nav/footer (네비,푸터)			2023.10.11	2023.10.11	1	진행완료											
	Front-End 템플릿 소스 코드: nav/footer (네비,푸터)			2023.10.11	2023.10.13	3	진행완료											
	Front-End 템플릿 소스 코드: 게시판 register, list, get, modify			2023.10.11	2023.10.13	3	진행완료											
	Front-End jQuery/JSP 자유 게시판 조회			2023.10.11	2023.10.13	3	진행완료											
	Front-End jQuery/JSP 1차 완성된 템플릿 includes로 취합			2023.10.13	2023.10.14	2	진행완료											
	Back-End SpringMVC Mybatis 자유 게시판 조회			2023.10.13	2023.10.15	3	진행완료											
	Back-End OracleDB 설계 및 구축 : 자유 게시판 조회			2023.10.13	2023.10.15	3	진행완료											
	Back-End SpringMVC 자유 게시판 댓글 기능 구현			2023.10.16	2023.10.18	3	진행완료											
	Back-End OracleDB 설계 및 구축 : 자유 게시판 댓글 기능 구현			2023.10.16	2023.10.18	3	진행완료											
	Back-End OracleDB 설계 및 구축 : 자유 게시판 댓글 폐이징 구현			2023.10.16	2023.10.18	3	진행완료											
	Back-End SpringMVC Mybatis 회원 탈퇴			2023.10.17	2023.10.18	2	진행완료											
	Back-End OracleDB 설계 및 구축 : 회원 탈퇴			2023.10.17	2023.10.18	2	진행완료											
조성진	Front-End jQuery/JSP 자유 게시판 등록			2023.10.11	2023.10.11	1	진행완료											
	Back-End OracleDB DB Modeling			2023.10.11	2023.10.11	1	진행완료											
	Back-End SpringMVC Mybatis 자유 게시판 게시글 등록			2023.10.11	2023.10.12	2	진행완료											
	Back-End OracleDB 설계 및 구축 : 로그인/로그아웃			2023.10.14	2023.10.18	5	진행완료											
	Front-End 반응형 웹 : 찾아오시는 길 Daum 카카오 지도 API 연동			2023.10.11	2023.10.11	1	진행완료											
	Front-End 회사 찾아오시는 길 Google Maps Platform 지도 API 연동			2023.10.11	2023.10.11	1	진행완료											
	Back-End OracleDB 설계 및 구축 : 자유 게시판 게시글 등록			2023.10.11	2023.10.12	2	진행완료											
	Back-End SpringMVC Mybatis 회원 정보 수정			2023.10.13	2023.10.17	5	진행완료											
	Back-End OracleDB 설계 및 구축 : 회원 정보 수정			2023.10.13	2023.10.17	5	진행완료											
	Front-End Javascript 반응형 웹 : 찾봇 API 연동 고객 실시간 문의 웹 구현			2023.10.18	2023.10.18	1	진행완료											
어예진	Front-End UI/UX 디자인: faq (자주찾는 질문)			2023.10.10	2023.10.10	1	진행완료											
	Front-End Bootstrap 반응형 웹 : faq (자주찾는 질문)			2023.10.11	2023.10.11	1	진행완료											
	Back-End SpringMVC 자유 게시판 취합 1차(get, register, modify, delete)			2023.10.12	2023.10.12	1	진행완료											
	Front-End jQuery/JSP 게시글 삭제			2023.10.12	2023.10.13	2	진행완료											
	Front-End SpringMVC Mybatis 게시글 삭제			2023.10.12	2023.10.13	2	진행완료											
	Back-End OracleDB 설계 및 구축 : 게시글 삭제			2023.10.12	2023.10.13	2	진행완료											
	Front-End UI/UX 디자인 : 전체 취합 후 에러 보고 및 수정			2023.10.12	2023.10.14	3	착수검토											
	Back-End SpringMVC 전체 취합 후 기능 구현 및 수정			2023.10.13	2023.10.18	6	착수검토											
조수정	Back-End SpringMVC Mybatis 비밀번호 찾기			2023.10.14	2023.10.16	3	진행완료											
	Back-End OracleDB 설계 및 구축 : 비밀번호 찾기			2023.10.14	2023.10.16	3	진행완료											
	Front-End Javascript 반응형 웹 : 찾봇 API 연동 고객 실시간 문의 웹 구현			2023.10.18	2023.10.18	1	진행완료											
	Front-End UI/UX 디자인: 메인			2023.10.10	2023.10.10	1	진행완료											
	Front-End UI/UX 디자인: 로그인, 회원가입			2023.10.10	2023.10.10	1	진행완료											
	Front-End 반응형 웹 : 회원가입 카카오 주소 API 연동			2023.10.11	2023.10.12	2	진행완료											
	Back-End OracleDB 설계 및 구축			2023.10.12	2023.10.13	2	진행완료											
박혜림	Back-End SpringMVC Mybatis 자유 게시판 수정			2023.10.12	2023.10.13	2	진행완료											
	Front-End jQuery/JSP 자유 게시판 구축			2023.10.11	2023.10.12	2	진행완료											
	Front-End React 반응형 웹 : 찾봇 API 연동 고객 실시간 문의 웹 구현			2023.10.15	2023.10.15	1	진행완료											
	Back-End SpringM																	

# 기능명세서

## 기능명세서

Table +			
Aa 구분	: 주 기능	☰ 상세 기능	☰ 설명
1. 회원가입	아이디 가입	아이디, 비밀번호, 이름, 이메일, 주소입력	주소 API 사용
	회원수정	비밀번호, 이메일, 주소 정보 수정	아이디, 닉네임 권한없음. 주소 API사용
	회원삭제	아이디, 비밀번호가 일치시 회원 정보 삭제	
2. 로그인	로그인	아이디, 비밀번호 입력시 로그인	메인페이지 리턴, 네비 login → logout으로 변경 적용
	아이디찾기	이름,이메일 입력시 DB 저장된 아이디 출력	일치시 : 찾으시는 아이디는 [ ]입니다. 불일치시 : 해당 정보로 가입된 아이디 존재하지 않습니다.
	비밀번호찾기	아이디,이메일 입력시 DB저장된 비밀번호 출력	찾으시는 비밀번호는 [ ]입니다. 로그인 후 비밀번호를 변경해주세요.
3. 게시판	로그아웃	로그인 후, 변경된 nav의 logout 클릭	로그인 페이지 리턴, 네비 logout → login 으로 변경 적용 게시글 작성 권한 없음 (회원)
	조회하기	게시글의 목록 화면	로그인 하지않으면 500서버에러 출력됨
	등록하기	게시글 등록 및 DB연동	로그인시 등록
4 댓글	수정하기	게시글 수정 및 DB연동	로그인시 등록한 글 상세 조회에서 수정
	삭제하기	게시글 삭제 및 DB연동	로그인시 등록한 글 상세 조회에서 수정에서 삭제
	검색처리	키워드에 매칭된 게시글 출력	검색조건 : 제목, 내용 , 작성자, 제목or내용, 제목or작성자, 제목or내용or작성자
5. 페이징	페이징처리	각 페이지당 10개의 게시글 출력	한 단위의 페이지창 1~10개단위, 1페이지당 10개의 게시글 출력
	조회하기	댓글 내용 확인 및 DB연동	게시판 글 목록화면에서 댓글 카운팅 처리 등록된 글 상세 조회 하단에 등록된 댓글 내용확인
	등록하기	댓글 등록 및 DB연동 (버튼 활성화 → 댓글등록, Close)	글 상세 조회 New Reply 클릭하여 댓글 등록
6. 파일	수정하기	댓글 내용 수정 및 DB연동 (버튼 활성화→ 댓글수정, 댓글삭제, Close)	글 상세 조회 하단에 댓글 클릭시 모달에서 수정
	삭제하기	댓글 내용 수정 및 DB연동 (버튼 활성화→ 댓글수정, 댓글삭제, Close)	글 상세 조회 하단에 댓글 클릭시 모달에서 삭제
	페이징처리	각 페이지당 10개의 게시글 출력	한 단위의 페이지창 1~10개단위, 1페이지당 10개의 게시글 출력

# 주요 코드

## FRONTEND

MediaQuery, Bootstrap 등 활용 반응형 웹 구현

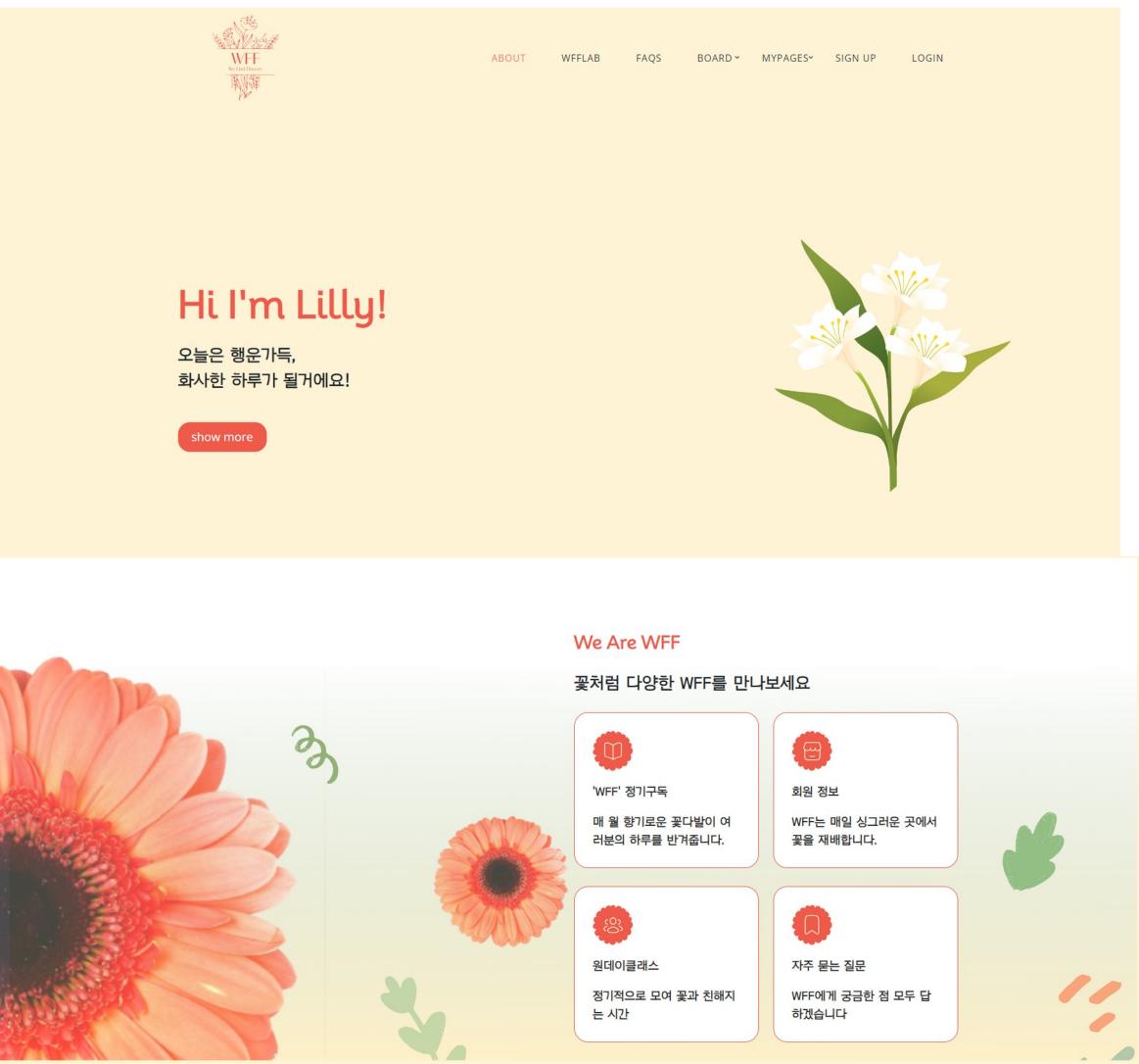
### ● 기능 안내

Bootstrap을 이용하여 UI를 제작하였음  
템플릿을 활용해 디자인을 하였음

### ● 코드 내용

Bootstrap을 이용하여 반응형, url 연결까지 완료하였음

### ● 구현이미지



### ● 주요코드

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
5
6 <%@include file="../includes/header.jsp" %>
7
8   <!--index CSS-->
9   <link rel="stylesheet" href="/resources/pages/main.css">
10
11
12 </head>
13
14 <%@ include file="../includes/nav.jsp"%>
15
16 <!--=====index=====-->
17 <div class="main-banner" id="top">
18   <div class="container">
19     <div class="row">
20       <div class="col-lg-7">
21         <div class="caption header-text">
22           <div class="line-dec">
23             
24           </div>
25           <h4><em class="eng_stress orange">Hi I'm Lilly!</em></h4>
26           <b class="kor">오늘은 행운가득, <br> 화사한 하루가 될거예요!</b> <br><br><br>
27           <div class="more scroll-to-section"><a href="#about.html">show more</a></div>
28
29         </div>
30       </div>
31     </div>

```

```

@media (max-width: 992px) {
  #top .line-dec img{
    display: none;
  }
  #top .caption{
    position: relative;
  }
  #top .more{
    position: absolute;
    bottom: 20px;
    right: 50%;
    transform: translateX(50%);
  }
}

.pre-header .Left-info ul li {
  padding-right: 10px;
  margin-right: 10px;
}
.pre-header .Left-info ul li a {
  font-size: 13px;
}
.main-banner .caption {
  padding: 30px !important;
  text-align: center;
}
.main-banner p {
  padding-right: 0%;
}
.main-banner .line-dec {
  margin: 30px auto 20px auto;
}

```

# 주요 코드

FRONTEND

MediaQuery, Bootstrap 등 활용 반응형 웹 구현

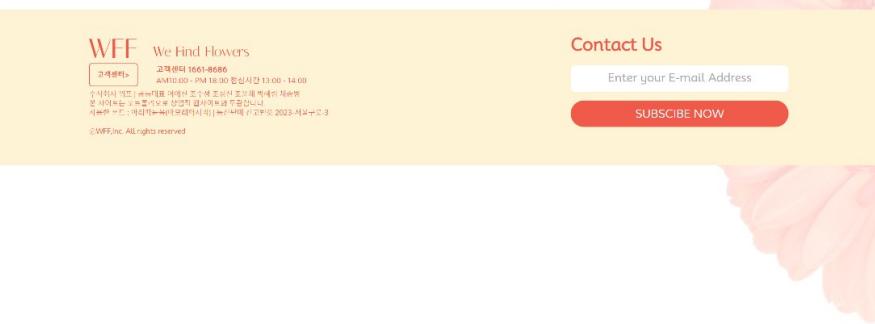
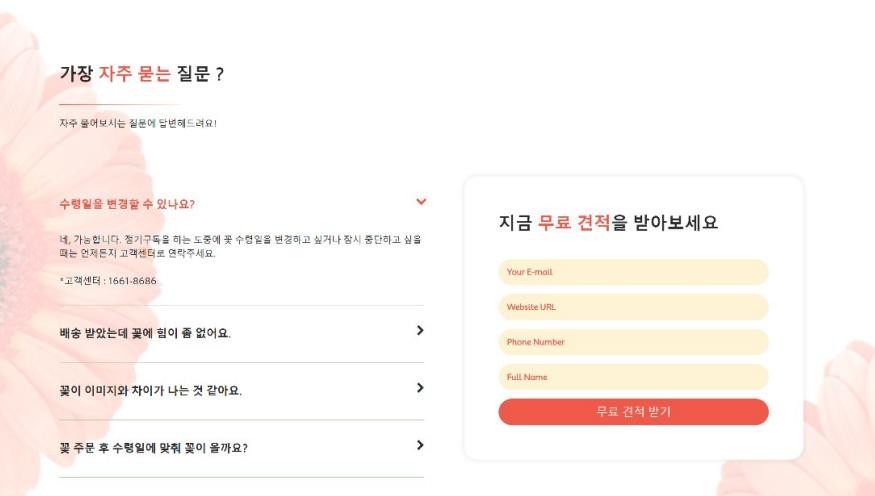
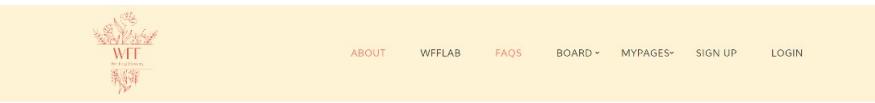
## ● 기능 안내

Bootstrap을 이용하여 UI를 제작하였음  
템플릿을 활용해 디자인을 하였음

## ● 코드 내용

Bootstrap을 이용하여 반응형, url 연결까지 완료하였음

## ● 구현이미지



## ● 주요코드

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2  pageEncoding="UTF-8"%>
3 <%@ taglib prefix="" uri="http://java.sun.com/jsp/jstl/core" %>
4 <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
5
6 <%@include file="../includes/header.jsp" %>
7
8   <!--faq CSS-->
9   <link rel="stylesheet" href="/resources/pages/faq.css">
10
11 </head>
12
13 <%@ include file="../includes/nav.jsp" %>
14
15 <!--=====faq=====-->
16 <div class="page-heading">
17   <div class="container">
18     <div class="row">
19       <div class="col-lg-7 align-self-center">
20         <div class="caption header-text">
21           <h2>WFF : WE FIND FLOWERS!</h2>
22           <div class="line-dec"></div>
23           <h4>자주 찾는 <em>질문</em> ?</h4>
24         </div>
25       </div>
26       <div class="col-lg-5">
27         
28     @media (max-width: 767px) {
29       .pre-header {
30         display: none;
31     }
32     .header-area {
33       top: 0px;
34     }
35     .main-banner {
36       margin-top: 100px;
37     }
38     .header-area .main-nav .nav li:nth-child(4) {
39       padding-right: 0px !important;
40     }
41     .header-area .main-nav .nav li {
42       background: #f7f7f7;
43     }
44     .header-area .main-nav .nav li.has-sub ul.sub-menu {
45       display: none;
46     }
47     .main-banner {
48       padding: 185px 0px;
49     }
50     .main-banner h4 {
51       font-size: 36px;
52       line-height: 44px;
53     }
54     .main-banner span {
55       margin: 0px 3px 0px 3px;
56     }
57     .main-banner .main-button a,
58     .main-banner .second-button a {
59       padding: 12px 20px;
60     }
61   }
62   .projects .item .down-content {
63     padding: 0px 0px 10px 10px;
64   }
65 
```

# 주요 코드 FRONTEND

# MediaQuery, Bootstrap 등 활용 반응형 웹 구현

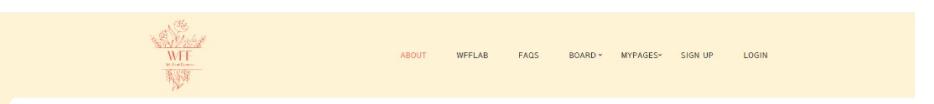
## • 기능 안내

Bootstrap을 이용하여 UI를 제작하였음  
템플릿을 활용해 디자인을 하였음

## • 코드 내용

Bootstrap을 이용하여 반응형, url 연결까지 완료 하였음

구현이미지



## • 주요코드

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>

<%@include file="../includes/header.jsp" %>

    <!--about CSS-->
    <link rel="stylesheet" href="/resources/pages/about.css">

</head>

<%@ include file="../includes/nav.jsp"%>

<=====about===== - >
<h2 class="gho">About WFF</h2>
<div class="page-heading">
<div class="container">
<div class="row">
<div class="col-lg-7 align-self-center">
<div class="caption header-text">
<p class="roc">
<h2 class="f_text3">WFF : WE FIND FLOWERS</h2><br><br>
<h3><b><span class="f_color">WFF</span>와 함께 꽃 찾으러 같이 가볼까요??</b></h3><br><br>
<span class="f_color">● </span>WFF는 꽃을 찾는 사람들을 입니다!<br><br>
<span class="f_color">● </span>꽃선물 골라하신가요? 이제는 WFF에서 꽃의 의미로 나의 마음을 전달해 보아요 !<br><br>
<span class="f_color">● </span>이제는 꽃도 반려의 시대가 왔다! 반려식물 반려 꽃의 시대 !<br></p>
<span class="f_color">● </span>WFFE 꽃을 받은 순간부터 자연으로 돌려보내는 순간까지 함께합니다.<br><br>
```

```
</div>
</div>
<div class="col-lg-5 align-self-center" >

    @media (max-width: 767px) {
        .pre-header {
            display: none;
        }
        .header-area {
            top: 0px;
        }
        .main-banner {
            margin-top: 100px;
        }
        .header-area .main-nav .nav li:nth-child(4) {
            padding-right: 0px !important;
        }
        .header-area .main-nav .nav li {
            background: #f7f7f7;
        }
        .header-area .main-nav .nav li.has-sub ul.sub-menu {
            display: none;
        }
        .main-banner {
            padding: 185px 0px;
        }
        .main-banner h4 {
            font-size: 36px;
            line-height: 44px;
        }
        .main-banner span {
            margin: 0px 3px 0px 3px;
        }
        .main-banner .main-button a,
        .main-banner .second-button a {
            padding: 12px 20px;
        }
        .projects .item .down-content {
            padding: 25px 15px 15px 15px;
        }
    }
}
```



# 주요 코드

FRONTEND

javascript, jQuery 라이브러리 적용

## ● 기능 안내

javascript, jQuery 사이트 전반에 활용했습니다.  
게시판의 CRUD에도 활용하여 html 태그에 담긴 값들을  
자바스크립트 통해서 컨트롤러에 넘겨  
DB에 반영하게끔 했음

## ● 구현이미지

번호	제목	작성자	작성일	수정일
292	테스트 제목	user00	2023-10-18	2023-10-18
291	테스트 제목	user00	2023-10-18	2023-10-18
290	테스트 제목	user00	2023-10-18	
289	테스트 제목	user00	2023-10-18	
288	테스트 제목	user00	2023-10-18	
287	테스트 제목	user00	2023-10-18	
286	테스트 제목	user00	2023-10-18	
285	테스트 제목	user00	2023-10-18	
284	테스트 제목	user00	2023-10-18	
283	테스트 제목	user00	2023-10-18	

<검색 조건 선택>  검색(Search)

## ● DB 목록

## ● 주요코드

```

// Page315 - list.jsp 게시판 조회를 위한 이벤트 처리 추가
$(".move").on("click", function(e) {
    e.preventDefault();
    actionForm.append("<input type='hidden' name='bno' value='"+$(this).attr("href")+"'>");
    actionForm.attr("action", "/wfflabboard/get");
    actionForm.submit();
});

// Page342 자바스크립트 소스 코딩 시작
var searchForm = $("#searchForm");

$("#searchForm button").on("click", function(e) {
    if(!searchForm.find("option:selected").val()){
        alert("검색 종류를 선택해 주시기 바랍니다!");
        return false;
    }

    if(!searchForm.find("input[name='keyword']").val()){
        alert("키워드를 입력해 주시기 바랍니다!");
        return false;
    }

    // 여러 종류들 중에서 검색 버튼을 클릭하면
    // 기본 검색 결과 페이지는 1페이지가 되도록 수정 처리합니다.
    searchForm.find("input[name='pageNum']").val("1");
    // 브라우저에서 검색 버튼을 클릭하면 form 태그의 전송은 막고,
    // 페이지의 번호는 1이 되도록 처리 합니다.
    // 화면에서 키워드가 없다면 검색을 하지 않도록 제어합니다.
    e.preventDefault();
    searchForm.submit();
});

// Page342 자바스크립트 소스 코딩 끝

// 등록하기 페이지로 이동
$("#regBtn").on("click", function() {
    self.location = "/wfflabboard/register";
});

```

# 주요 코드

FRONTEND

Kakao Open API 등 연동 적용

## ● 기능 안내

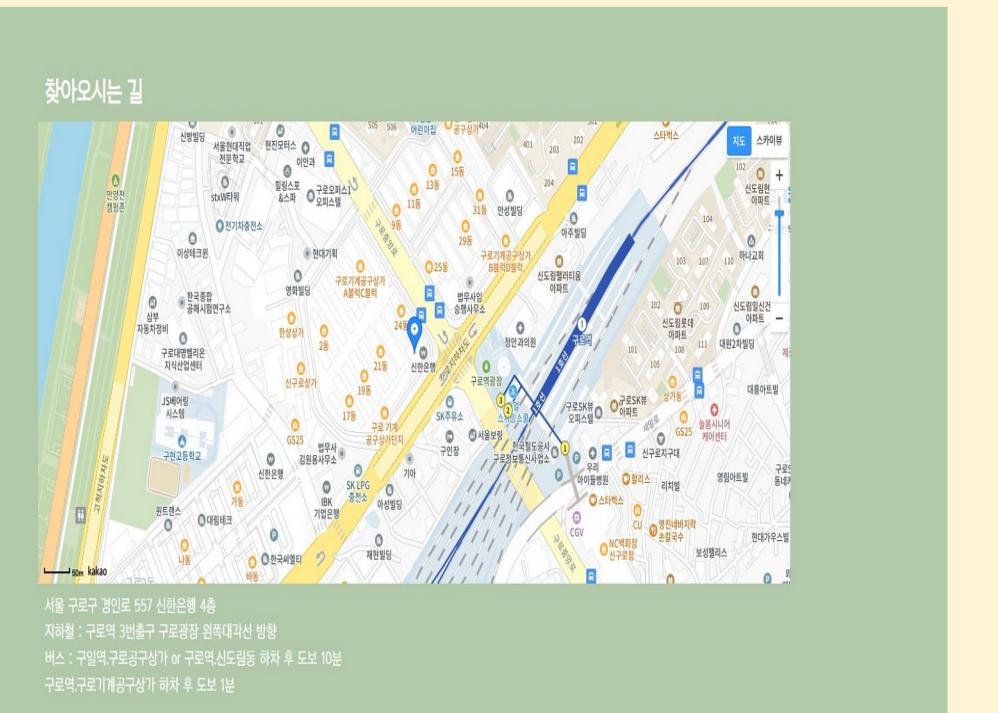
Map  
Address

## ● 코드 내용

Map -  
Kakao API를 연동하여 화면에  
지도를 띄워 마커와 함께 찾아오는길 맵 구현

Address -  
다음 주소 API를 연동하여 회원가입, 회원정보 수정시  
주소찾기 버튼을 클릭하여 주소검색 기능 구현

## ● 구현이미지



## ● 주요코드

```

<!-- https://apis.map.kakao.com/web/sample/addMapControl/ -->
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=23bfff20fd36f0395446770ee5a12951a"></script>
<script>
var mapContainer = document.getElementById("map"), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(37.5030698, 126.8789856), // 지도의 중심좌표
        level: 3 // 지도의 확대 레벨
    };
var map = new kakao.maps.Map(mapContainer, mapOption); // 지도를 생성합니다

// 일반 지도와 스카이뷰로 지도 타입을 전환할 수 있는 지도타입 컨트롤을 생성합니다
var mapTypeControl = new kakao.maps.MapTypeControl();
map.addControl(mapTypeControl, kakao.maps.ControlPosition.TOPRIGHT);

// 지도 확대 축소를 제어할 수 있는 줌 컨트롤을 생성합니다
var zoomControl = new kakao.maps.ZoomControl();
map.addControl(zoomControl, kakao.maps.ControlPosition.RIGHT);

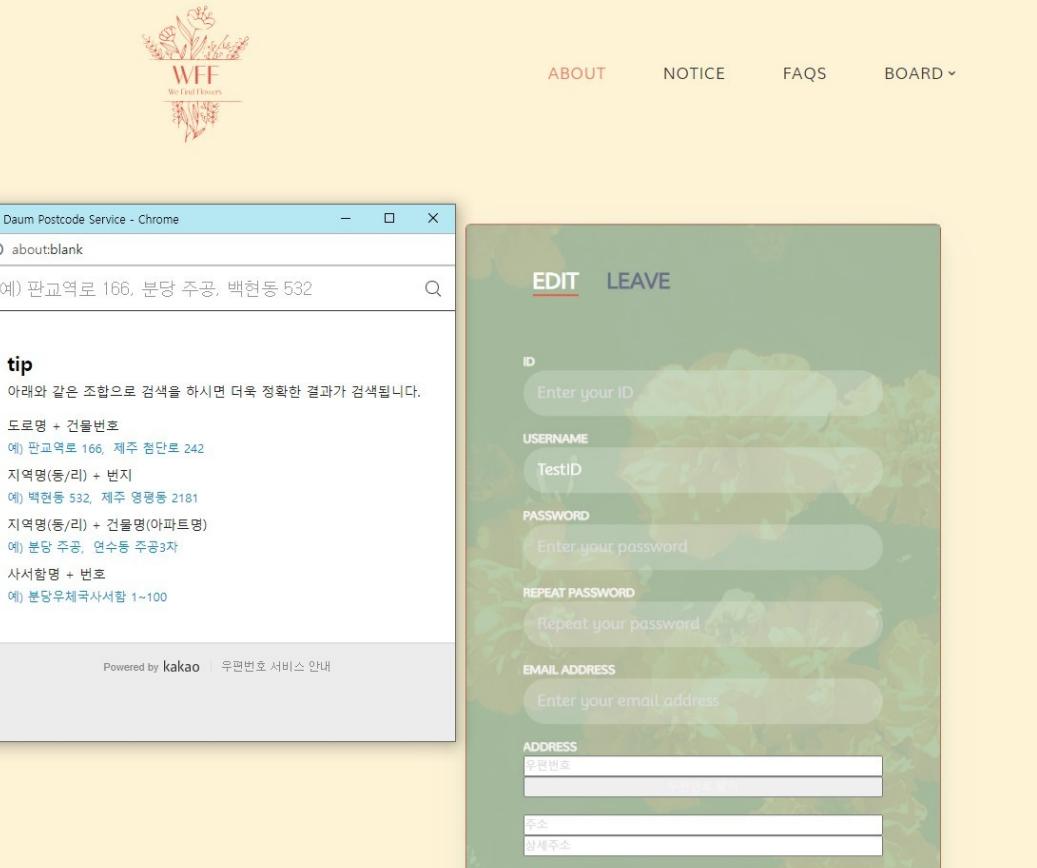
//Kakao Map 마커 생성하기(카카오 개발자센터 Kakao Maps API Sample 참고)
// https://apis.map.kakao.com/web/sample/basicMarker/
// 마커가 표시될 위치입니다
var markerPosition = new kakao.maps.LatLng(37.5030698, 126.8789856);

// 마커를 생성합니다
var marker = new kakao.maps.Marker({
    position: markerPosition
});

//window 객체 활용 map 안에 마커 봉양 표시 : resize 처리(주가포팅)
window.addEventListener("resize", function () {
    map.setCenter(new kakao.maps.LatLng(37.5030698, 126.8789856))
});

// 마커가 지도 위에 표시되도록 설정합니다
marker.setMap(map);
// 아래 코드는 지도 위의 마커를 제거하는 코드입니다
// marker.setMap(null);
</script>

```



```

<script src="//ti.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
<script>
function sample6_execDaumPostcode() {
    new daum.Postcode({
        oncomplete: function(data) {
            // 팝업에서 검색결과 항목을 클릭했을때 실행할 코드를 작성하는 부분.

            // 각 주소의 노출 규칙에 따라 주소를 조합한다.
            // 내려오는 변수가 값이 없는 경우엔 공백('')값을 가지므로, 이를 참고하여 분기 한다.
            var addr = ''; // 주소 변수
            var extraAddr = ''; // 참고항목 변수

            // 사용자가 선택한 주소 타입에 따라 해당 주소 값을 가져온다.
            if (data.userSelectedType === 'R') { // 사용자가 도로명 주소를 선택했을 경우
                addr = data.roadAddress;
            } else { // 사용자가 지번 주소를 선택했을 경우(J)
                addr = data.jibunAddress;
            }

            // 사용자가 선택한 주소가 도로명 타입일 때 참고항목을 조합한다.
            if(data.userSelectedType === 'R'){
                // 법정동명이 있을 경우 추가한다. (법정리는 제외)
                // 법정동의 경우 마지막 문자가 "동/로/가"로 끝난다.
                if(data.bname !== '' && /([동|로|가])$/g.test(data.bname)){
                    extraAddr += data.bname;
                }
                // 건물명이 있고, 공동주택일 경우 추가한다.
                if(data.buildingName !== '' && data.apartment === 'Y'){
                    extraAddr += (extraAddr !== '' ? ', ' + data.buildingName : data.buildingName);
                }
                // 표시할 참고항목이 있을 경우, 괄호까지 추가한 최종 문자열을 만든다.
                if(extraAddr !== ''){
                    extraAddr = '(' + extraAddr + ')';
                }
                // 조합된 참고항목을 해당 필드에 넣는다.
                document.getElementById("sample6_extraAddress").value = extraAddr;
            } else {
                document.getElementById("sample6_extraAddress").value = '';
            }

            // 우편번호와 주소 정보를 해당 필드에 넣는다.
            document.getElementById('sample6_postcode').value = data.zonecode;
            document.getElementById("sample6_address").value = addr;
            // 커서를 상세주소 필드로 이동한다.
            document.getElementById("sample6_dong").focus();
        }
    });
}
</script>

```

# 주요 코드

## BACKEND

Mybatis 활용 OracleDB 연동 CRUD 개발

### ● 기능 안내

게시판 게시물 리스트 및 페이지/ 게시물 조회/ 게시물 등록, 수정, 삭제/ 댓글 등록 및 댓글 수 표현

### ● 코드 내용

mapper.java, mapper.xml, service, serviceImpl을 구현 후 controller를 이용하여 게시판이 동작되도록 설계하였다.

### ● 구현이미지

번호	제목	작성자	작성일	수정일
51	제목 [0]	user	2023-10-13	2023-10-13
50	1번 글의 제목을 수정합니다! [0]	user	2023-10-13	2023-10-14
49	제목 [0]	user	2023-10-13	2023-10-13
48	제목 [0]	user	2023-10-13	2023-10-13
47	제목 [0]	user	2023-10-13	2023-10-13
46	제목 [0]	user	2023-10-13	2023-10-13
45	제목 [0]	user	2023-10-13	2023-10-13
44	제목 [0]	user	2023-10-13	2023-10-13
43	제목 [0]	user	2023-10-13	2023-10-13
42	제목 [0]	user	2023-10-13	2023-10-13

검색 조건 선택 <input> 검색(Search) 1 2 3 4 5 6

localhost/wfflabboard/get?pageNum=1&amount=10&type=&keyword=&bno=86

read

제목  
제목 찾고자용

내용  
내용 찾기

작성자

수정 글 삭제

localhost/wfflabboard/modify?bno=96&pageNum=1&amount=10&type=&keyword=

Modify

글 번호  
96

Title  
제목 찾고자용

Text area  
내용 찾기

Writer  
작성자 찾기

RegDate  
2023/10/14

Update Date  
2023/10/19

게시글 수정  
게시글 삭제

### ● 주요코드

```
// 게시글 등록
@GetMapping("/register")
public void register() { //void -> string to change!
    //Request method 'GET' not supported error!
}

@PostMapping("/register")
public String register(WffLabVO wfflabboard, RedirectAttributes rttr) {
    Log.info("register(등록) : " + wfflabboard);

    service.register(wfflabboard);
    rttr.addFlashAttribute("result", wfflabboard.getBno());
    return "redirect:/wfflabboard/list";
}

// 게시글 페이징 처리 list
@GetMapping("/list")
public void list(Criteria cri, Model model) {
    Log.info("list : "+cri);
    model.addAttribute("list", service.getList(cri));
    int total = service.getTotal(cri);

    Log.info("total"+total);

    model.addAttribute("pageMaker", new PageDTO(cri, total));
}

// 게시글 수정
@PostMapping("/modify")
public String modify(WffLabVO wfflabboard, @ModelAttribute("cri") Criteria cri, RedirectAttributes rttr) {
    Log.info("modify: 수정 처리" + wfflabboard);

    if (service.modify(wfflabboard)) {
        rttr.addFlashAttribute("result", "success");
    }

    rttr.addAttribute("pageNum", cri.getPageNum());
    rttr.addAttribute("amount", cri.getAmount());
    rttr.addAttribute("type", cri.getType());
    rttr.addAttribute("keyword", cri.getKeyword());

    return "redirect:/wfflabboard/list";
}

// 게시글 삭제
@PostMapping("/remove")
public String remove(@RequestParam("bno") Long bno, @ModelAttribute("cri") Criteria cri, RedirectAttributes rttr) {
    Log.info("remove: 삭제 처리" + bno);

    if(service.remove(bno)) {
        rttr.addFlashAttribute("result", "success");
    }

    rttr.addAttribute("pageNum", cri.getPageNum());
    rttr.addAttribute("amount", cri.getAmount());
    rttr.addAttribute("type", cri.getType());
    rttr.addAttribute("keyword", cri.getKeyword());

    return "redirect:/wfflabboard/list";
}
```

# 주요 코드

## BACKEND

댓글과 댓글 수에 대한 처리

### ● 기능 안내

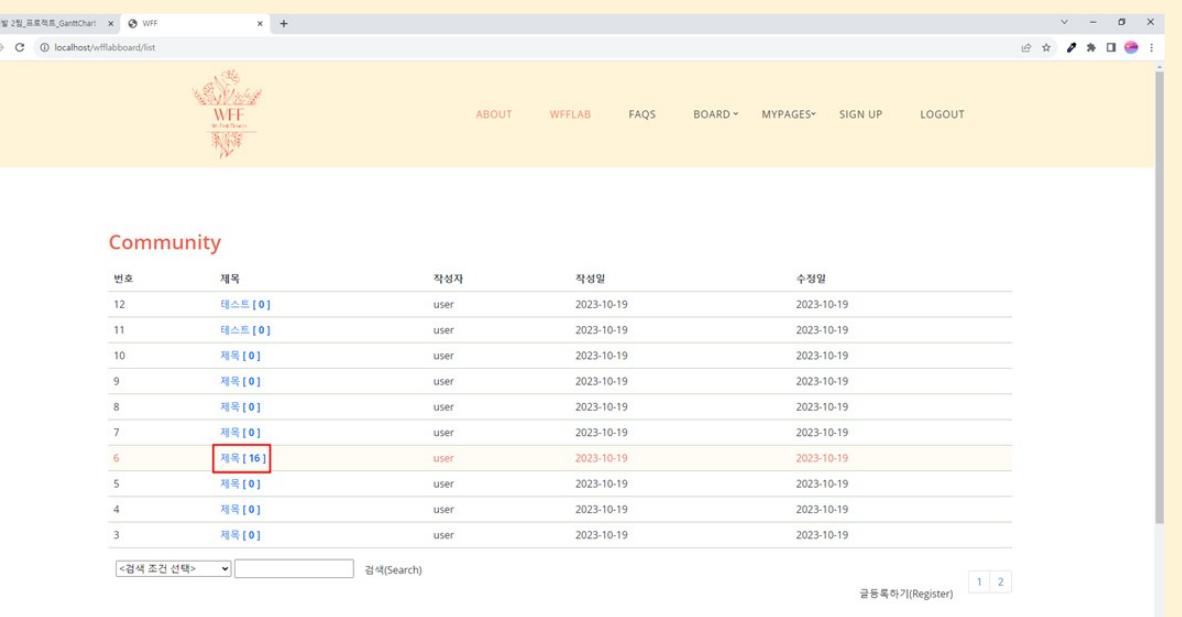
게시글에 댓글이 등록 또는 삭제되면서 변동되는 댓글 수를  
게시글의 목록 화면에서 출력될 수 있도록 구현

### ● 코드 내용

mapper.java, mapper.xml, service, serviceImpl을  
구현 후

controller를 이용하여 게시판이 동작되도록 설계하였다.

### ● 구현 이미지



### ● 주요 코드

#### WffLabVO 클래스

```
private int replyCnt; // 게시글의 댓글 숫자(댓글 수) 필드 선언
```

#### WffLabMapper 인터페이스

```
// 댓글
public void updateReplyCnt(@Param("bno") Long bno, @Param("amount") int amount);
```

#### WffLabMapper.xml

```
<update id="updateReplyCnt">
    update wfflab_board set replycnt = replycnt + #{amount} WHERE bno = #{bno}
</update>
<!-- getListWithPaging 처리 -->
<select id="getListWithPaging" resultType="org.zerock.domain.WffLabVO">
    <![CDATA[
        SELECT BNO, TITLE, CONTENT, WRITER, REGDATE, UPDATEDATE, replycnt
        FROM (
            SELECT
                /*+ INDEX_DESC(WFFLAB_BOARD PK_WFFLAB_BOARD) */
                ROWNUM rn, bno, title, content, WRITER, REGDATE, UPDATEDATE, replycnt
            from WFFLAB_BOARD
            WHERE
        )>
    <include refid="criteria"></include>
    <![CDATA[
        ROWNUM <= #{pageNum} * #{amount}
    ) )
    WHERE RN > (#{pageNum} - 1) * #{amount}
    ]]>
</select>
```

#### WffLabReplyServiceImpl 클래스

```
@Setter(onMethod_ = @Autowired)
private WffLabReplyMapper mapper;

@Setter(onMethod_ = @Autowired)
private WffLabMapper boardmapper;

// 댓글 등록
@Transactional
@Override
public int register(WffLabReplyVO vo) {
    log.info("register 댓글 등록처리 성공!" + vo);
    boardmapper.updateReplyCnt(vo.getBno(), 1); // 댓글 1개를 늘려라 (추가)
    return mapper.insert(vo);
}
```

```
// 댓글 삭제
@Transactional
@Override
public int remove(Long rno) {
    log.info("remove 댓글 삭제 처리 : " + rno);
    // Page485 다음 2줄 소스 추가 코딩해 줍니다.
    WffLabReplyVO vo = mapper.read(rno);
    boardmapper.updateReplyCnt(vo.getBno(), -1);

    return mapper.delete(rno);
}
```

#### list.jsp

```
<c:forEach items="${list}" var="wfflabboard">
    <tr>
        <td><c:out value="${wfflabboard.bno}"></c:out></td>
        <td><a class="move" href="

```

# 주요 코드

## BACKEND

### HikariCP 연동 DBCP 구현 개발

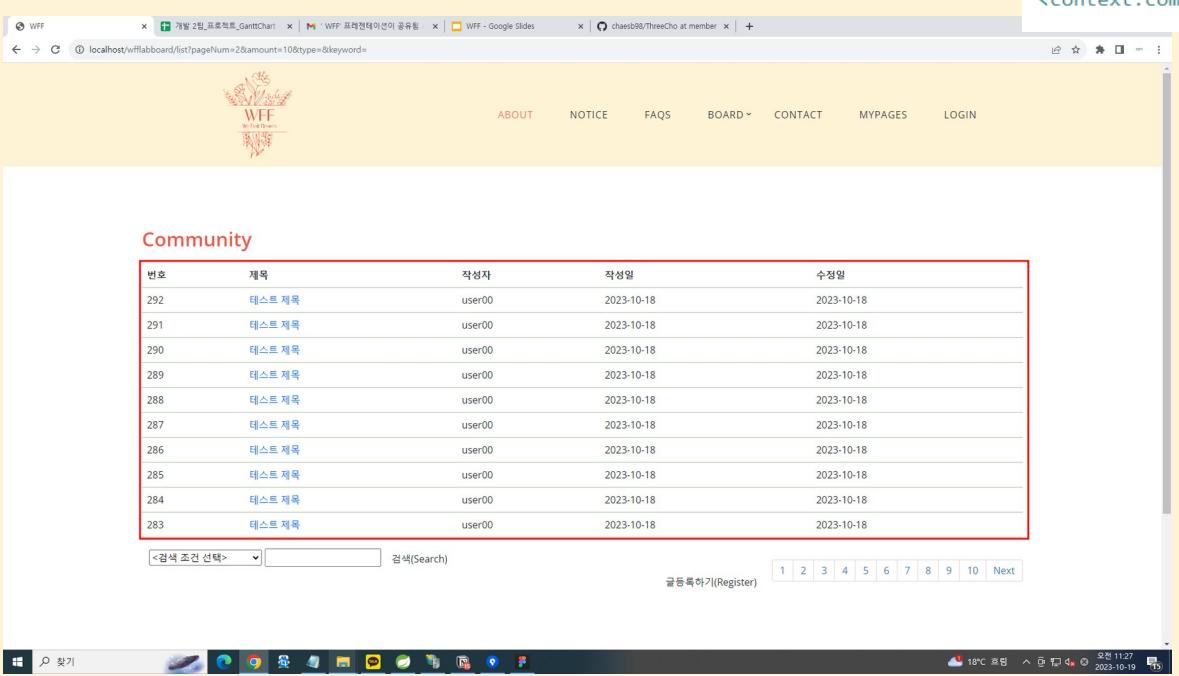
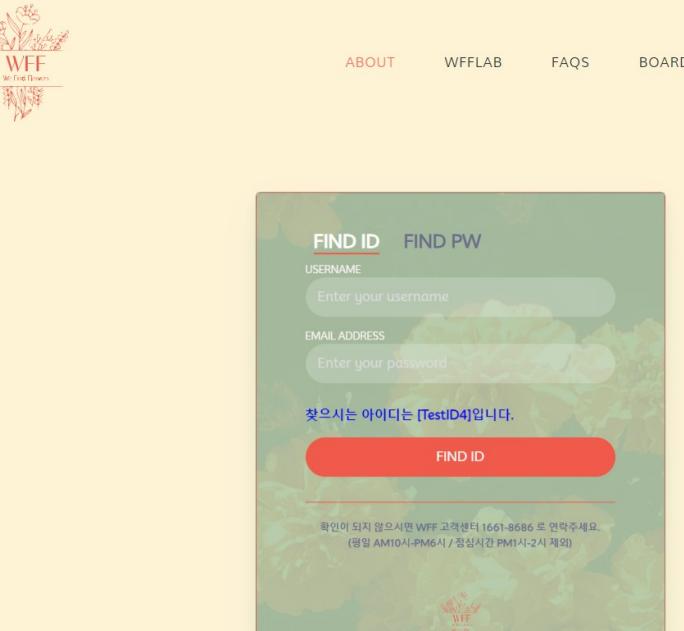
#### ● 기능 안내

HikariCP 연동을 이용하여 데이터베이스 연동 후 member, 게시판 기능을 구현하였다.

#### ● 코드 내용

HikariCP을 이용하여 데이터베이스 연결 후 게시판, member에 대한 기능 구현

#### ● 구현이미지



#### ● 주요코드

```

<bean id="hikariConfig" class="com.zaxxer.hikari.HikariConfig">
<!-- 히카리와 디비를 연결해줌 -->
<!-- p.102 log4jdbc.log4j2.properties 설정을 위해 아래 구문은 주석 처리합니다! -->
<!--
<property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/></property>
<property name="jdbcUrl" value="jdbc:oracle:thin:@localhost:1521:orcl"/></property>
-->

<property name="driverClassName" value="net.sf.log4jdb.sql.jdbcapi.DriverSpy"/></property>
<property name="jdbcUrl" value="jdbc:log4jdbc:oracle:thin:@localhost:1521:orcl"/></property>
<property name="username" value="admin"/></property>
<property name="password" value="admin"/></property>

</bean>

<!-- HikariCP configuration -->
<!-- 히카리에 있는 설정 사항들은 데이터소스들을 블여준 것이고 그 데이터 소스들을 헬에 블여주기 위해서 ref hikariConfig를 넣었다? -->

<bean id="dataSource" class="com.zaxxer.hikari.HikariDataSource" destroy-method="close">
<constructor-arg ref="hikariConfig" />
</bean>

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
<property name="dataSource" ref="dataSource"/></property>
</bean>

<mybatis-spring:scan base-package="org.zeroock.mapper"/>
<context:component-scan base-package="org.zeroock.service"></context:component-scan>

```

# 주요 코드

## BACKEND

Maven 빌드 도구 활용 개발

### ● 기능 안내

lombok를 이용하여 어노테이션 기능을 활용하였고  
ajax를 이용해 댓글 기능을 구현하였다.

### ● 코드 내용

ajax를 이용하여 댓글 기능 구현,  
pom.xml에 lombok를 추가하여 lombok 기능 사용을  
가능하게 했다.

### ● 구현이미지



### ● 주요코드

```

var wfflabreply = (function() {
  function add(reply, callback, error) {
    console.log("add안식");

    $.ajax({
      type : 'post',
      url : '/wfflabreplies/new',
      data : JSON.stringify(reply),
      contentType : "application/json; charset=utf-8",
      success : function(result, status, xhr) {
        if (callback) {
          console.log(reply);
          callback(result);
        }
      },
      error : function(xhr, status, er) {
        if (error) {
          error(er);
        }
      }
    });
  }

  package org.zerock.domain;

  import lombok.Data;

  @Data
  public class MemberVO {

    private String id; // 멤버 아이디 필드 선언
    private String password; // 멤버 비밀번호 필드 선언
    private String email; // 멤버 이메일 필드 선언
    private String name; // 멤버 이름 필드 선언
    private String address1; // 멤버 주소 필드 선언
    private String address2;
    private String address3;

  }
}

<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.2</version>
  <scope>provided</scope>
</dependency>

```

# 코드 리뷰

## BACKEND

회원가입, 회원정보 수정, 계정 삭제

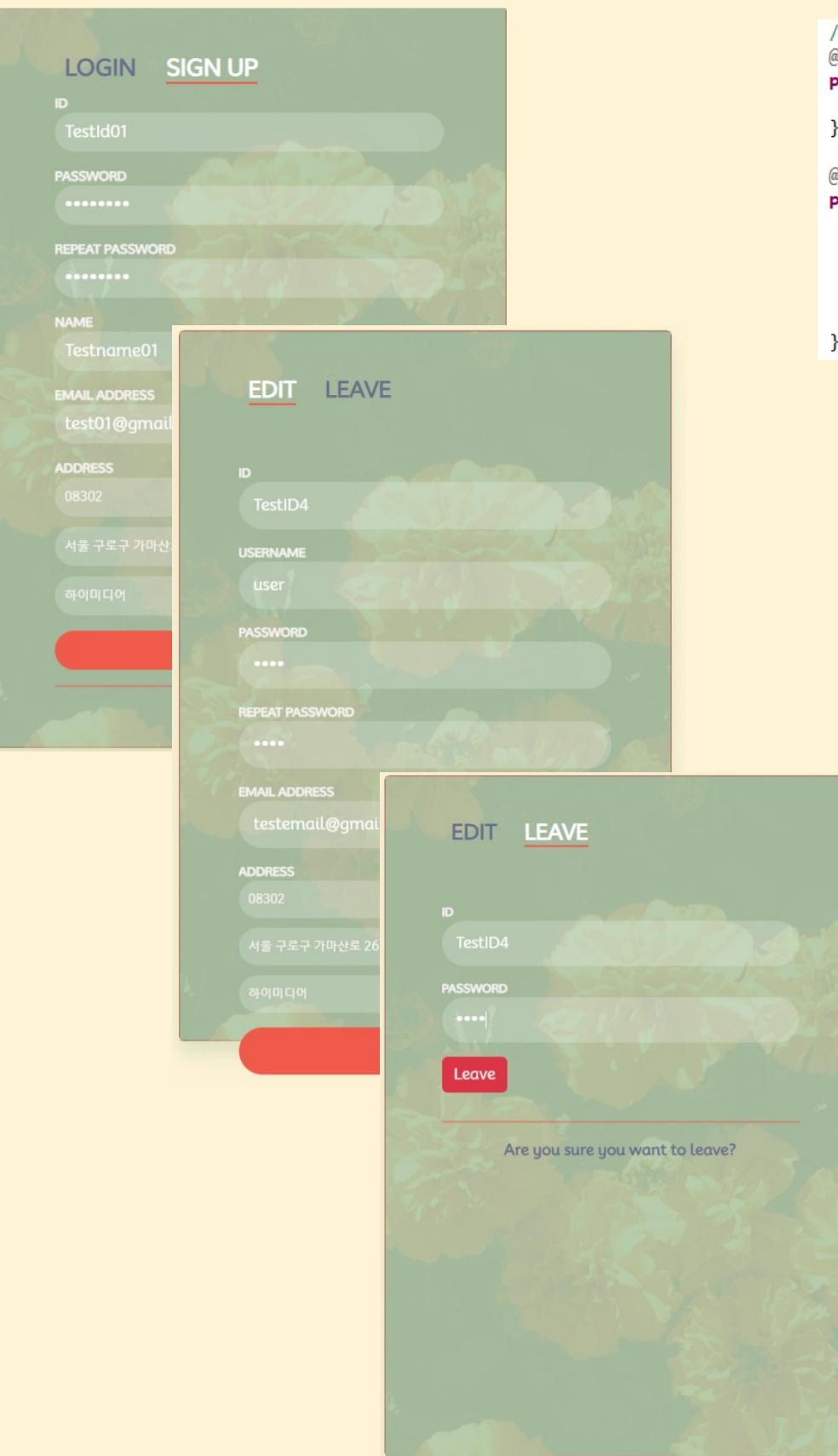
### ● 기능 안내

정보를 입력하여 회원가입처리를 하고  
회원정보 수정, 계정 삭제는 로그인 후  
동작이 되도록 구현.

### ● 코드 내용

mapper, service, serviceImpl을 이용하여 값을 받아온 후  
controller에서 처리하여 동작이 되도록 구현하였다.

### ● 구현이미지



### ● 주요코드

```
// 회원 가입
@GetMapping("/join")
public void register() {
}

@PostMapping("/join")
public String register(MemberVO member, RedirectAttributes rttr) {
    log.info("register(가입): " + member);
    memberservice.register(member);
    rttr.addFlashAttribute("result", member.getId());
    // redirect : 갱신해서 보내줌
    return "redirect:/pages/main";
}

// 회원 정보 수정
@GetMapping("/edit")
public void edit() {
}

@PostMapping("/edit")
public String modify(MemberVO member, RedirectAttributes rttr, HttpServletRequest request) {
    memberservice.modify(member);
    rttr.addFlashAttribute("result", member.getId());
    return "redirect:/pages/main";
}

// 회원 계정 삭제(탈퇴)
@GetMapping(value="/leave")
public void leave(){
}

@PostMapping(value="/leave")
public String remove(@RequestParam("id") String id, @RequestParam("password") String password, RedirectAttributes rttr) {
    log.info("remove: 삭제 처리 " + id+ ", "+password);
    // BoardController의 remove() 메서드는 삭제 후 페이지의 이동이 필요하므로
    // RedirectAttributes를 파라미터로 사용하고, 'redirect'를 이용해서
    // 삭제 처리 후에 다시 목록 페이지로 이동 처리 합니다.
    if(memberservice.deleteMember(id, password)) {
        rttr.addFlashAttribute("result", "success");
    } else {
        rttr.addFlashAttribute("result", "false");
    }
    return "redirect:/member/login";
}
```

# 코드 리뷰

## BACKEND

로그인, 로그아웃, 아이디 찾기, 비밀번호 찾기

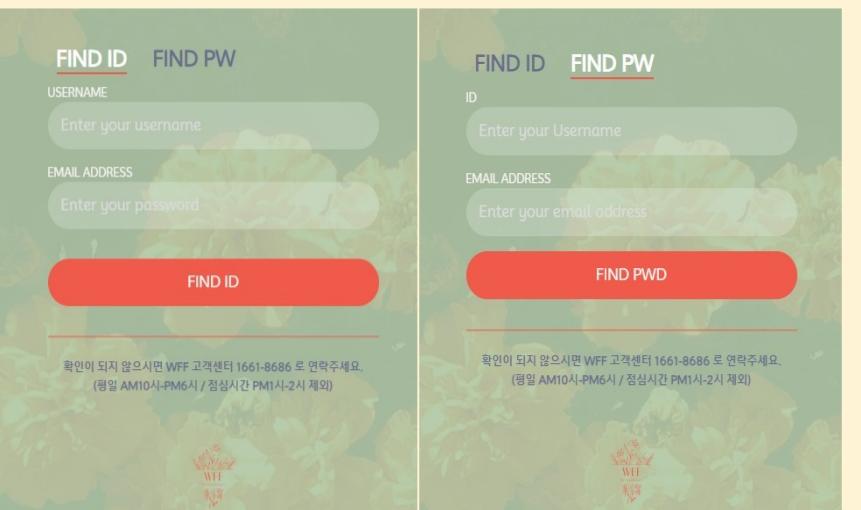
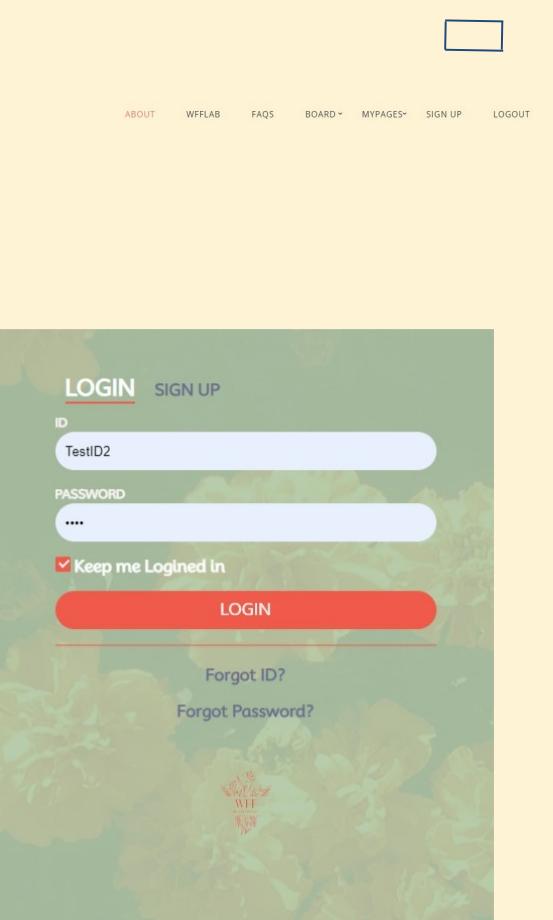
### ● 기능 안내

로그인 기능과 로그아웃 기능 구현,  
아이디 찾기, 비밀번호 찾기 구현

### ● 코드 내용

session을 이용한 로그인과 로그아웃 기능을 구현하였고,  
프론트에서 입력받은 값과 DB의 정보가 일치할 경우 아이디,  
비밀번호찾기가 가능하도록 기능을 구현하였다.

### ● 구현이미지



### ● 주요코드

```
// 로그인
@GetMapping("/login")
public void login() {
}

@PostMapping("/login")
public String login(@RequestParam("id")String id, @RequestParam("password") String password, HttpServletRequest req) throws Exception {
    MemberVO login = memberservice.login(id, password);

    if(login != null) {
        HttpSession session = req.getSession(); // 잘못되어도 session이 생긴다. 위치를 if문 안으로 이동
        session.setAttribute("member", login);
        log.info(login);
    }
    return "redirect:/pages/main";
}

// 로그아웃
@GetMapping("/logout")
public String logout(HttpSession session) throws Exception{
    session.invalidate();
    return "redirect:/member/login";
}

//회원 아이디 찾기
@GetMapping("/findId")
public void findId() {
}

@PostMapping("/findId")
public void findId(@RequestParam("name") String name, @RequestParam("email") String email, Model model) {
    log.info("Post find Id....");
    log.info("name input : "+name+" email input : "+email);

    List<MemberVO> userIdList = memberservice.findId(name, email);
    if(userIdList.size()==0 || userIdList == null) {
        model.addAttribute("fail",0);
    }else {
        log.info("user Id = "+userIdList);

        Set<String> regIdSet = new HashSet<String>();
        for(MemberVO memberList : userIdList) {
            log.info("userId : "+memberList.getId());
            for(int i=0;i<userIdList.size();i++) {
                regIdSet.add(memberList.getId());
            }
        }
        log.info("regIdSetset : "+regIdSet);
        model.addAttribute("success",1);
        model.addAttribute("userid", regIdSet);
    }
}

//회원 비밀번호 찾기
@GetMapping("/findPwd")
public void findPwd() {
}

@PostMapping("/findPwd")
public void findPwd(@RequestParam("id") String id, @RequestParam("email") String email, Model model) {
    log.info("Post Find Pwd....");
    log.info("id input : "+id+" email input : "+email);

    List<MemberVO> userPwdList = memberservice.findPwd(id, email);
    if(userPwdList.size()==0 || userPwdList == null) {
        model.addAttribute("fail",0);
    }else {
        log.info("user Pwd = "+userPwdList);

        Set<String> regPwdSet = new HashSet<String>();
        for(MemberVO memberList : userPwdList) {
            log.info("findPwd : "+memberList.getPassword());
            for(int i=0;i<userPwdList.size();i++) {
                regPwdSet.add(memberList.getPassword());
            }
        }
        log.info("regPwdSetset : "+regPwdSet);
        model.addAttribute("success",1);
        model.addAttribute("userPwd", regPwdSet);
    }
}
```

# Epilogue



••• We Find Flower •••

JSP와 Spring MVC로 진행한 프로젝트는 처음이라 새로운 경험이었지만, 많은 것을 배우고 성장할 수 있는 기회였습니다. 프로젝트를 통해 Java 기반의 웹 개발 환경에 대한 이해를 높일 수 있었으며, 특히 JSP와 Spring MVC 프레임워크를 사용하여 동적이고 효율적인 웹 어플리케이션을 구축하는 방법을 습득했습니다.

프로젝트를 통해 얻은 이러한 경험들은 팀원들과의 협업, 새로운 기술 습득, 문제 해결 능력 등 다양한 역량을 강화하는 데 기여했습니다. 앞으로의 프로젝트에서는 이러한 경험을 바탕으로 더욱 전문적이고 효과적인 웹 개발을 위해 노력하고 성장해 나가고자 할 것입니다.

look  
over  
here!

We Find Flower!



# Thank You

