



# B4U

**Bookmark for You**  
이 북마크.. 뭐더라?

K-Digital Training X Multicampus

AI 활용 지능형 서비스 개발 A반

박경태

# 목차

---

01 프로젝트 배경

02 프로젝트 소개

03 결과 화면

04 느낀점

B4U

# 프로젝트 배경



폴더 형태의  
북마크

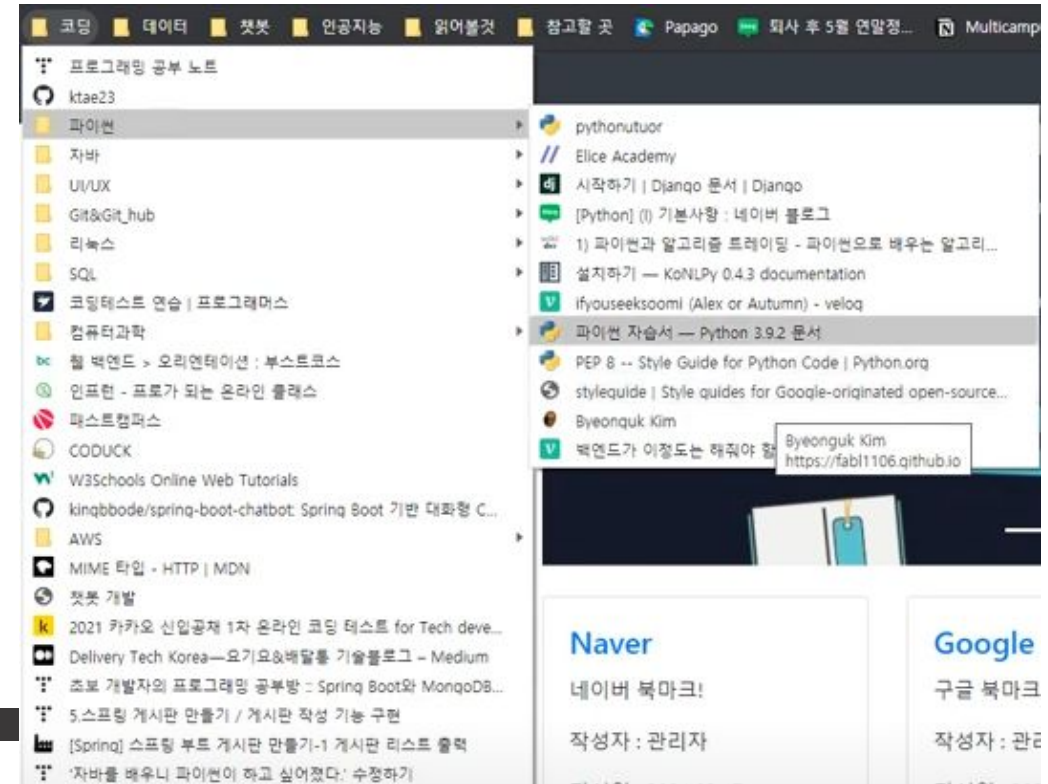
VS.

게시판 형태의  
북마크



폴더 형태의  
북마크

VS.



해당 북마크가 어떤 곳이었는지 폴더 명으로 유추는  
가능하나 실제로 들어가기 전까지는 직관적으로  
알아보기 어려움

<p>21</p> <p>세션테스트</p> <p>세션에서 id 가져와라 좋은 말 할때</p> <p>작성자 : zz2381</p> <p>작성일 : 2021.03.18</p>	<p>17</p> <p>북마크 생성 그만할래</p> <p>북마크 생성 그만하고 다른 기능 만들어야해</p> <p>작성자 : zz2381</p> <p>작성일 : 2021.03.17</p>	<p>14</p> <p>에러는 왜 날까</p> <p>되다 안되다 그러나</p> <p>작성자 : zz2381</p> <p>작성일 : 2021.03.17</p>
<p>11</p> <p>북마크</p> <p>오류 나서 생성 안되어도 번호는 오른다</p> <p>작성자 : zz2381</p> <p>작성일 : 2021.03.17</p>	<p>7</p> <p>수정 테스트</p> <p>수정 테스트 진행</p> <p>작성자 : test</p> <p>작성일 : 2021.03.17</p>	<p>6</p> <p>북마크 테스트</p> <p>구글 최고</p> <p>작성자 : zz2381</p> <p>작성일 : 2021.03.17</p>
<p>5</p> <p>될때까지</p> <p>수정 테스트</p> <p>작성자 : zz2381</p> <p>작성일 : 2021.03.17</p>	<p>2</p> <p>글번호테스트</p> <p>글번호 테스트</p> <p>작성자 : zz2381</p> <p>작성일 : 2021.03.17</p>	<p>1</p> <p>시퀀스테스트</p> <p>시퀀스 테스트</p> <p>작성자 : zz2381</p> <p>작성일 : 2021.03.17</p>

VS.

게시판 형태의  
북마크

간단한 설명과 작성일이 있기 때문에 어떤 북마크인지 직관적인 이해가 가능



# 프로젝트 소개

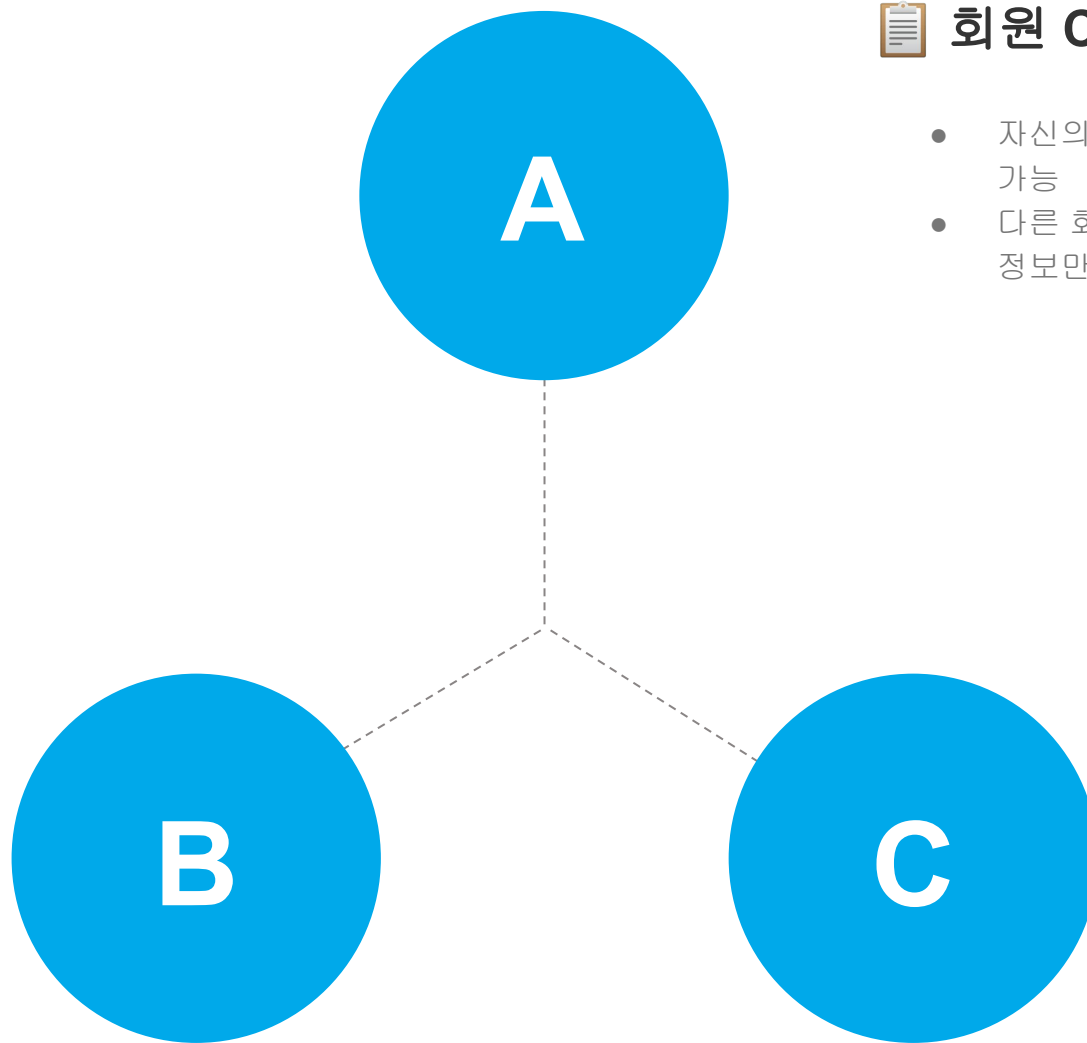
# 주요 기능

## 회원 CRUD

- 자신의 계정만 수정, 탈퇴 가능
- 다른 회원에 대해 제한된 정보만 조회 가능

## 관리자 계정 및 권한

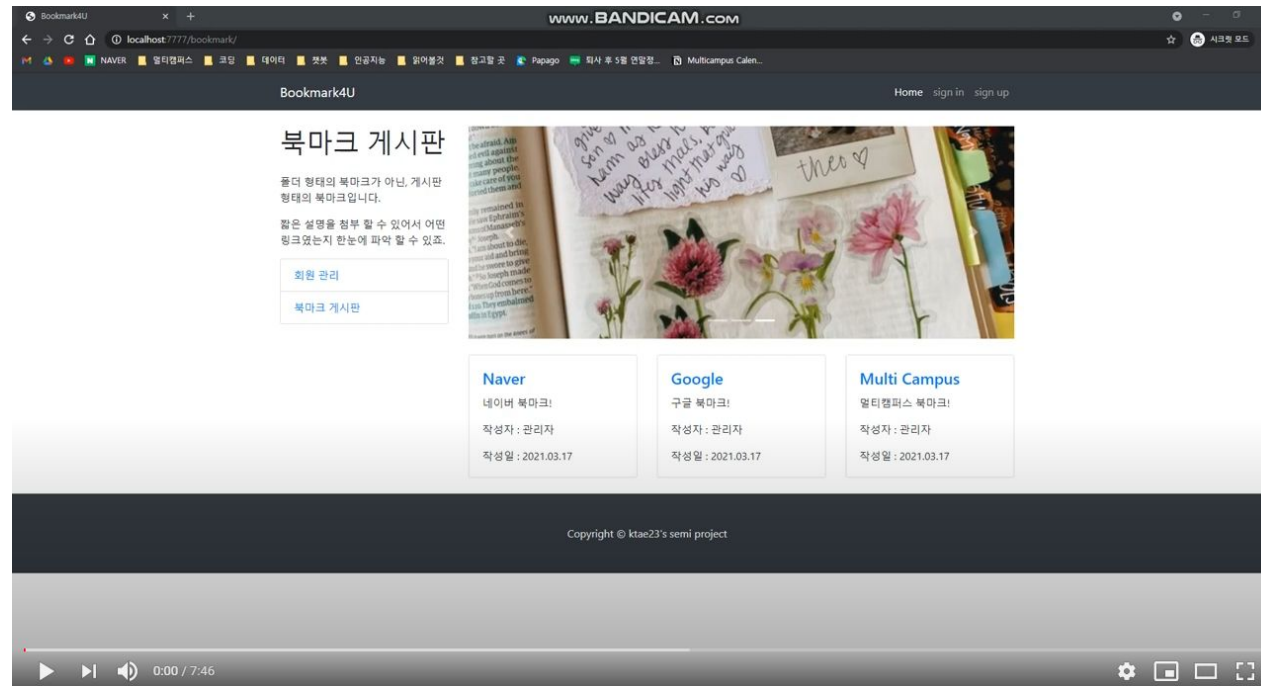
- 회원 CRUD 모든 권한
- 게시글 CRUD 모든 권한



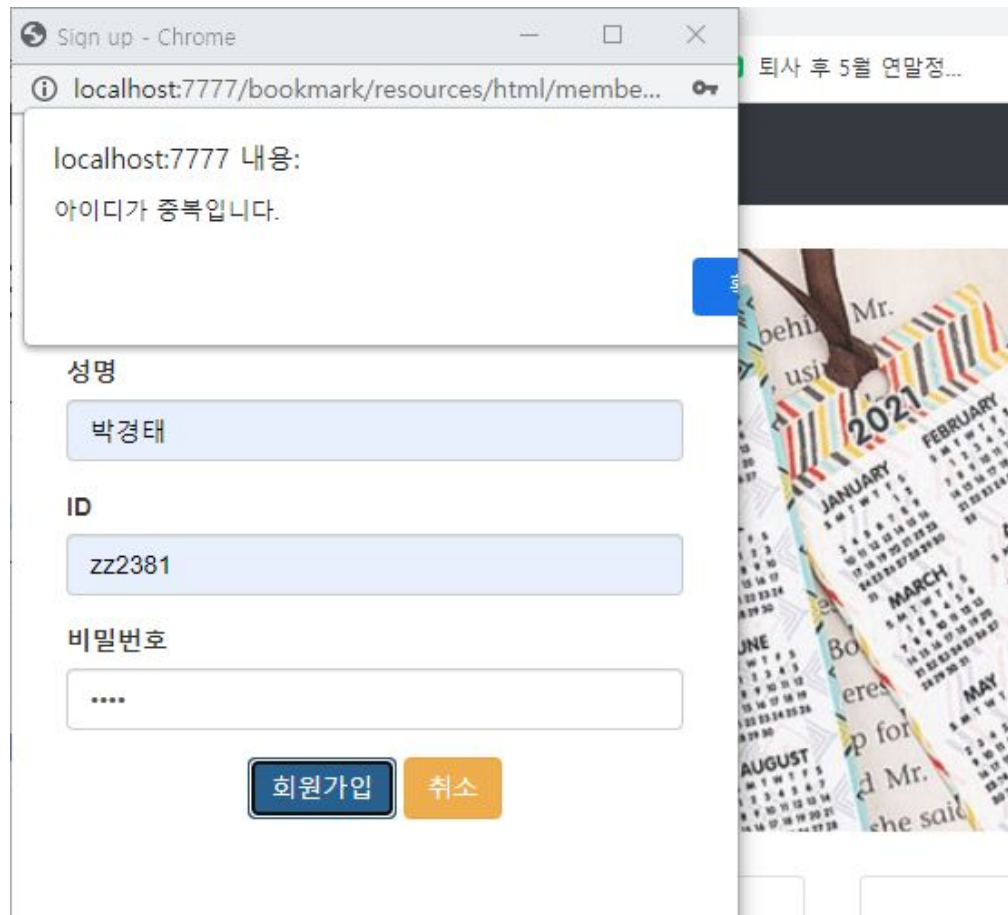
## 게시판 CRUD

- 자신의 게시물만 수정, 삭제 가능
- 회원 탈퇴 시 작성한 게시글 모두 삭제 됨





결과 시연 영상



회원 가입 진행 시 ID 중복 체크

중복 된 계정이 없을 경우 가입 됨

```
// Memeber CRUD
// Memeber Create
@RequestMapping(value = "memberInsert",
    method = {RequestMethod.POST},
    produces = "application/text; charset=utf8")
@ResponseBody
public String memberInsert(HttpServletRequest request,
    HttpServletResponse response) {

    String id=request.getParameter("id");
    String pw=request.getParameter("pw");
    String name=request.getParameter("name");

    try {
        MemberVO m=new MemberVO(id,pw,name);
        int result = memberService.idCheck(m);
        if(result != 1) {
            memberService.memberInsert(m);
            return name+"님 회원가입 되셨습니다";
        }else {
            return "아이디가 중복입니다.";
        }
    }catch(Exception e) {
        return e.getMessage();
    }
}
```

localhost:7777/bookmark/resources/html/selectM...

localhost:7777 내용:  
로그인 실패

zz2381

비밀번호  
....

로그인 취소

```
// Sign in
@RequestMapping(value = "memberlogin",
    method= {RequestMethod.POST},
    produces = "application/text; charset=utf8")
@ResponseBody
public String memberlogin(HttpServletRequest request,
    HttpServletResponse response) {
    String id=request.getParameter("id");
    String pw=request.getParameter("pw");

    JSONObject json=new JSONObject();
    try {
        String name = memberService.memberlogin(new MemberVO(id,pw));
        MemberVO member=new MemberVO(id,pw,name);

        if(name!=null) {
            HttpSession session=request.getSession();
            session.setMaxInactiveInterval(3600);
            System.out.println(session.getMaxInactiveInterval());
            session.setAttribute("member", member);
            json.put("name", name);
        }else {
            json.put("msg", "로그인 실패");
        }
    }catch(Exception e) {
        json.put("msg", e.getMessage());
    }
    return json.toString();
}
```

로그인 시 비밀번호가 다르면 실패

```

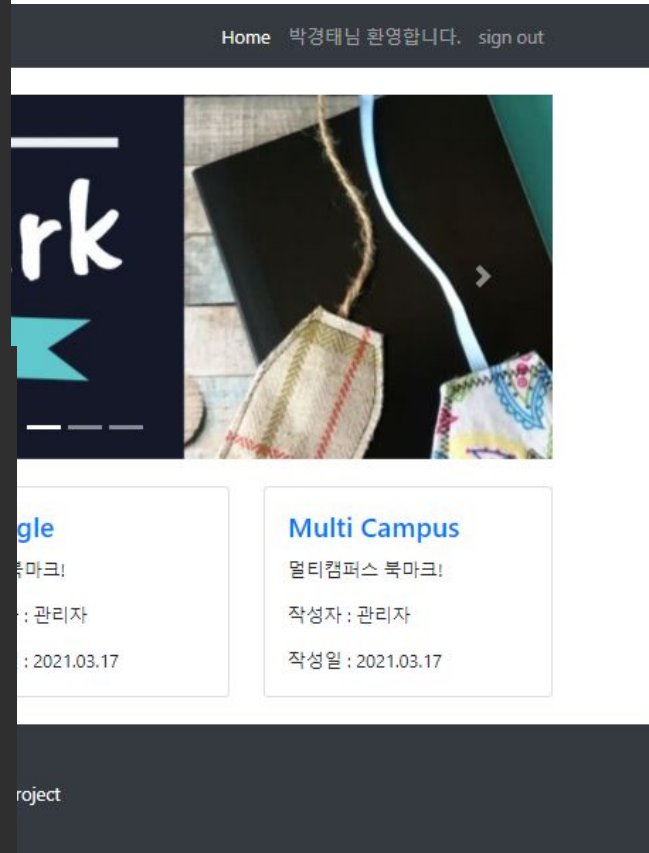
$(document).ready(function(){
    $("#loginBtn").click(function(){ // 로그인

        var id=$("#id").val();
        var pw=$("#pw").val();

        if(id == 'admin' && pw == '0000'){
            $.post("../memberlogin",
                {
                    id:id,
                    pw:pw
                },
                function(data, status){
                    var obj=JSON.parse(data);
                    if(obj.name){
                        alert(obj.name + "님 환영합니다.");
                        opener.parent.$.cookie("logged4admin",obj.name + "님 환영합니다.");
                        opener.parent.location.reload();
                        window.close();
                    }else{
                        alert(obj.msg);
                        location.reload();
                    }
                }
            );
        }else{
            $.post("../memberlogin",
                {
                    id:id,
                    pw:pw
                },
                function(data, status){
                    var obj=JSON.parse(data);
                    if(obj.name){
                        alert(obj.name+ "님 환영합니다.");
                        opener.parent.$.cookie("logged",obj.name + "님 환영합니다.");
                        opener.parent.location.reload();
                        window.close();
                    }else{
                        alert(obj.msg);
                        location.reload();
                    }
                }
            );
        }
    });
});

```

로그인 성공 시 “〇〇님  
환영합니다” 노출





## 회원 관리

회원 정보를 생성, 조회, 수정, 삭제할 수 있는 페이지입니다.

[회원 정보 수정](#)
[회원 목록](#)

[관리자](#)

id : admin

가입일 : 2021.03.18

[회원이가입테스트](#)

id : usertest

가입일 : 2021.03.17

[수정이름](#)

id : user

가입일 : 2021.03.17

[가나다](#)

id : 1234

가입일 : 2021.03.17

[리로드테스트](#)

id : reload

가입일 : 2021.03.17

[회원](#)

id : idtest

가입일 : 2021.03.17

[시간테스트](#)

id : datetest

가입일 : 2021.03.17

[회원](#)

id : 회원

가입일 : 2021.03.16

[test](#)

id : test

가입일 : 2021.03.16

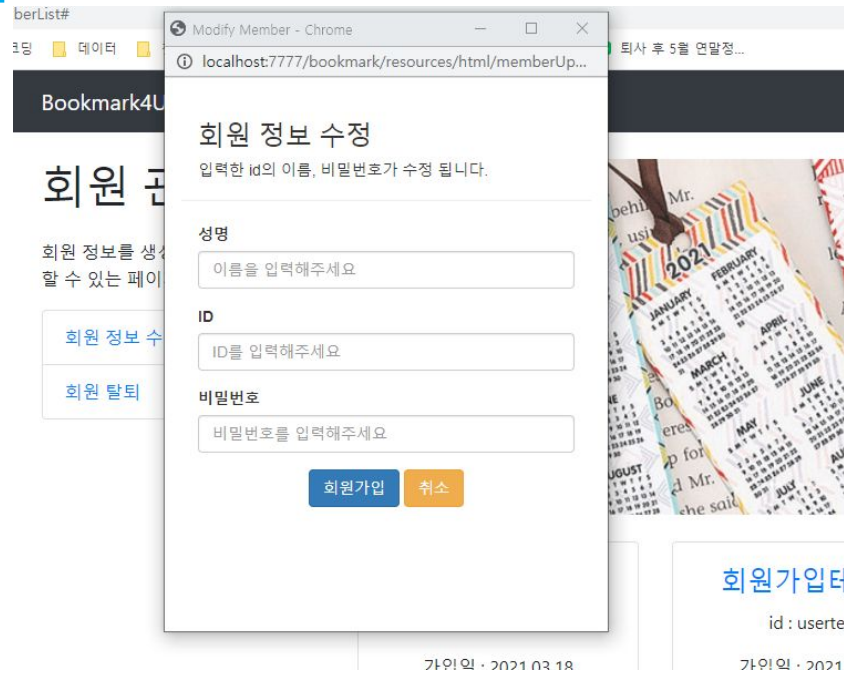
```
// Member Read
@RequestMapping(value = "/memberList", method = RequestMethod.GET)
public ModelAndView memberList( HttpServletRequest request,
    HttpServletResponse response) throws Exception {

    List<MemberVO> memberList = memberService.memberList();
    ModelAndView mav = new ModelAndView();

    HttpSession session = request.getSession(false);
    MemberVO member = (MemberVO) session.getAttribute("member");
    if(member != null) {
        mav.setViewName("memberList");
        mav.addObject("memberList", memberList);
    }else {
        mav.setViewName("index");
    }

    return mav;
}
```

회원 조회



정보 수정  
수정 시 세션에서 받아온 현재 접속자 계정과  
비밀번호를 수정하고자 하는 회원과 비교해  
일치하지 않으면 실패

```
// Member Update
@RequestMapping(value = "/memberUpdate", method= {RequestMethod.POST},
    produces = "application/text; charset=utf8")
@ResponseBody
public String memberUpdate( HttpServletRequest request,
    HttpServletResponse response) throws Exception {

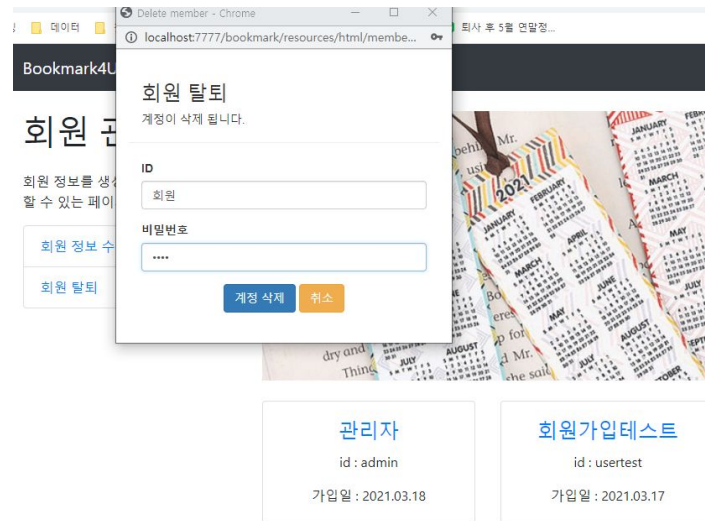
    String id=request.getParameter("id");
    String pw=request.getParameter("pw");
    String name=request.getParameter("name");

    HttpSession session=request.getSession(false);
    MemberVO member = (MemberVO) session.getAttribute("member");
    String getId = member.getId();
    String getPw = member.getPw();

    if(getPw.equals(pw) && getId.equals(id)) {
        try {
            MemberVO m = new MemberVO(id, pw, name);
            memberService.memberUpdate(m);

            return "정보가 수정 되었습니다";
        }catch(Exception e) {
            return e.getMessage();
        }
    }else if(getId.equals("admin")){
        try {
            MemberVO m = new MemberVO(id, pw, name);
            memberService.memberUpdate(m);

            return "정보가 수정 되었습니다";
        }catch(Exception e) {
            return e.getMessage();
        }
    }
    else {
        return "다른 사용자 계정을 수정 할 수 없습니다.";
    }
}
```



```
// Member Delete
@RequestMapping(value = "/memberDelete", method= {RequestMethod.POST},
    produces = "application/text; charset=utf8")
@ResponseBody
public String memberDelete( HttpServletRequest request,
    HttpServletResponse response) throws Exception {

    String id=request.getParameter("id");
    String pw=request.getParameter("pw");
    HttpSession session=request.getSession(false);
    MemberVO member = (MemberVO) session.getAttribute("member");
    String getId = member.getId();
    String getPw = member.getPw();

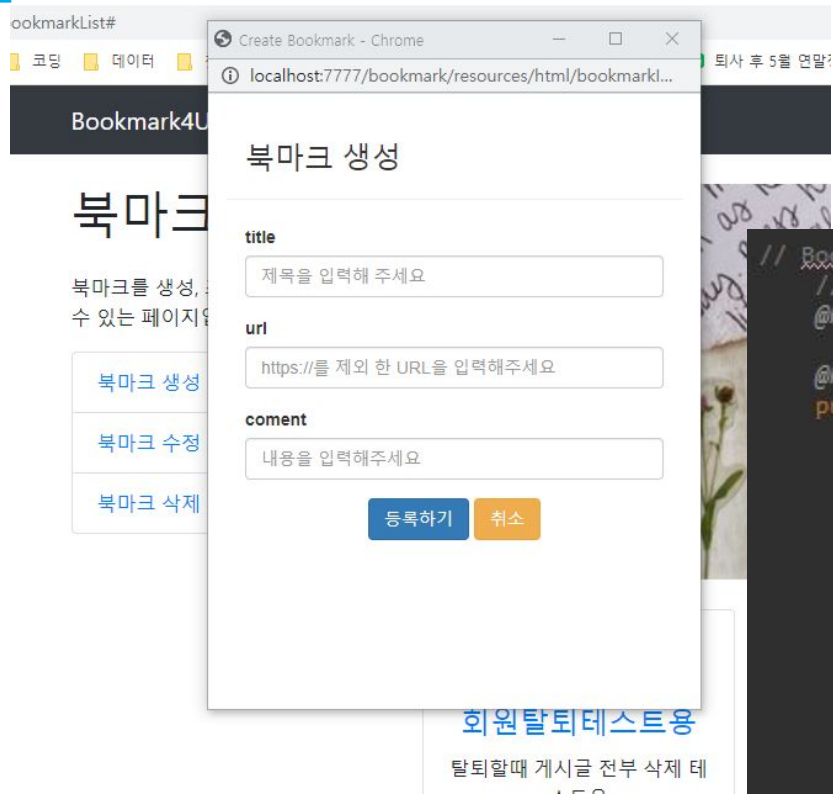
    if(getPw.equals(pw) && getId.equals(id)) {
        try {
            MemberVO m = new MemberVO(id,pw);
            memberService.memberDelete(m);

            return "탈퇴 완료";
        }catch(Exception e) {
            return e.getMessage();
        }
    }else if(getId.equals("admin")){
        try {
            MemberVO m = new MemberVO(id,pw);
            memberService.memberDelete(m);

            return "삭제 완료";
        }catch(Exception e) {
            return e.getMessage();
        }
    }else {
        return "다른 사용자 계정을 탈퇴 할 수 없습니다.";
    }
}
```

계정 삭제 시 작성한 글도 모두 삭제 됨





북마크 작성

```
// Bookmark CRUD
// Bookmark Create
@RequestMapping(value = "/bookmarkInsert",method= {RequestMethod.POST},
    produces = "application/text; charset=utf8")
@ResponseBody
public String bookmarkInsert( HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    HttpSession session=request.getSession(false);
    MemberVO member = (MemberVO) session.getAttribute("member");
    String memid=member.getId();

    String title=request.getParameter("title");
    String url=request.getParameter("url");
    String coment=request.getParameter("coment");

    try {
        Long bookmark_no = bookmarkService.getBookmark_no();
        System.out.println("홈컨트롤러 북마크 번호");
        BookmarkVO b = new BookmarkVO(title, url, coment, memid,bookmark_no);
        bookmarkService.bookmarkInsert(b);
        System.out.println("홈컨트롤러 북마크 인서트");

        return title+"이(가) 작성 되었습니다";
    }catch(Exception e) {
        return e.getMessage();
    }
}
```



## 북마크 관리

북마크를 생성, 조회, 수정, 삭제 할 수 있는 페이지입니다.

[북마크 생성](#)
[북마크 수정](#)
[북마크 삭제](#)


24

### 회원탈퇴테스트용

탈퇴할때 계시를 전부 삭제 테스트용

작성자 : 회원

작성일 : 2021.03.18

22

### 글 수정 테스트

세션 이용한 수정

작성자 : zz2361

작성일 : 2021.03.18

21

### 세션테스트

세션에서 id 가져와라 좋은 말 할때

작성자 : zz2361

작성일 : 2021.03.18

17

### 북마크 생성 그만할래

북마크 생성 그만하고 다른 기능 만들어야해

작성자 : zz2361

작성일 : 2021.03.17

16

### 코멘트칸 테스트

코멘트를 길게 쓰면 어떻게 나올지 테스트코멘트를 길게 쓰면 어떻게 나올지 테스트코멘트를 길게 쓰면 어떻게 나올지 테스트

작성자 : 회원

작성일 : 2021.03.17

14

### 에러는 왜 날까

되다 안되다 그러나

작성자 : zz2361

작성일 : 2021.03.17

11

### 북마크

오류 나서 생성 안되어도 번호는 우르다

7

### 수정 테스트

수정테스트 진행

6

### 북마크 테스트

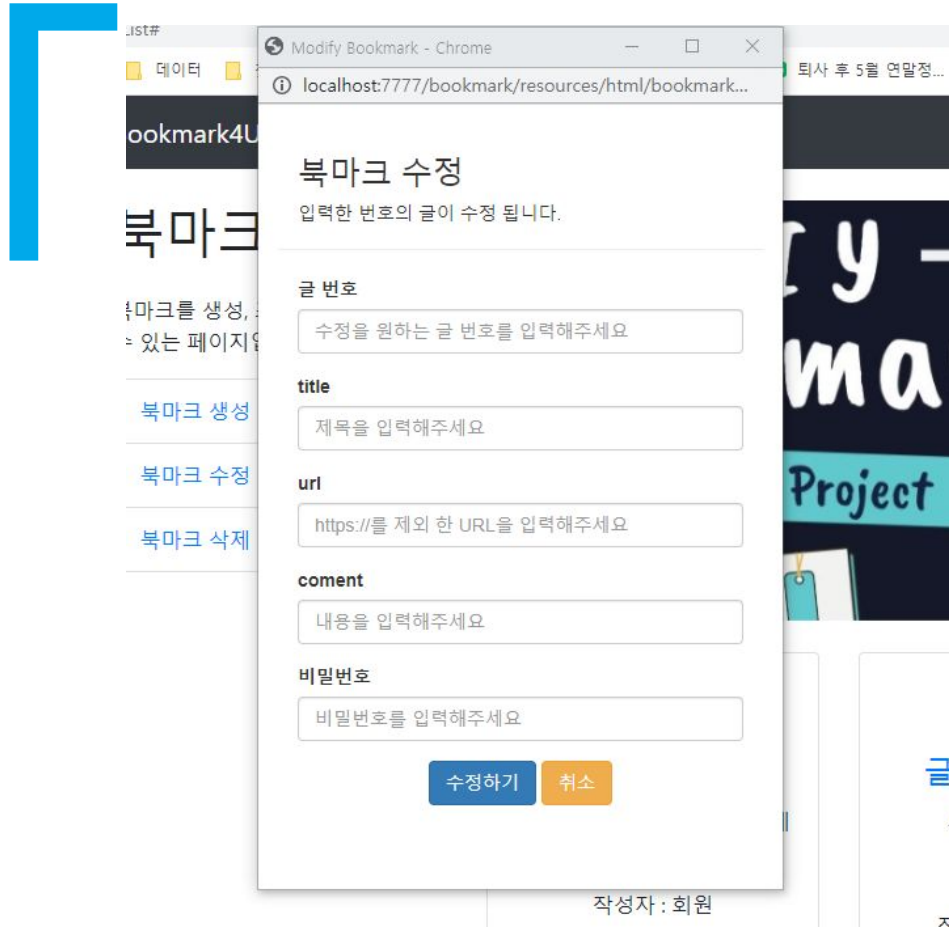
구글 최고

```
// Bookmark Read
@RequestMapping(value = "/bookmarkList", method= RequestMethod.GET)
public ModelAndView bookmarkList( HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    List<BookmarkVO> bookmarkList = bookmarkService.bookmarkList();
    ModelAndView mav = new ModelAndView();

    HttpSession session = request.getSession(false);
    MemberVO member = (MemberVO) session.getAttribute("member");
    if(member != null) {
        mav.setViewName("bookmarkList");
        mav.addObject("bookmarkList", bookmarkList);
    }else {
        mav.setViewName("index");
    }

    return mav;
}
```

북마크 조회



게시물 수정  
수정 시 게시물 번호와 일치하는 ID를 가져와  
접속중인 세션의 ID와 비밀번호 비교 후 진행

```
// Bookmark Update
@RequestMapping(value = "/bookmarkUpdate", method= {RequestMethod.POST},
    produces = "application/text; charset=utf8")
@ResponseBody
public String bookmarkUpdate( HttpServletRequest request,
    HttpServletResponse response) throws Exception {

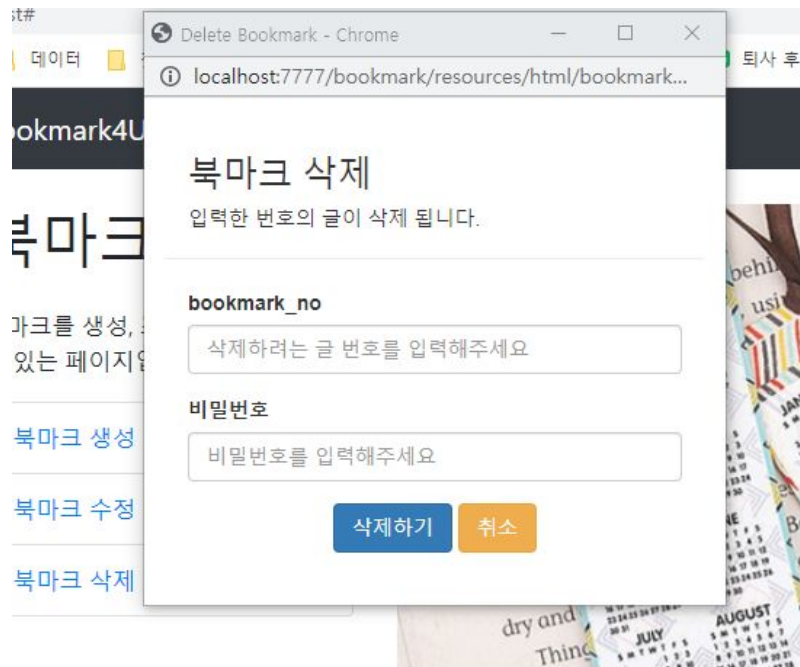
    HttpSession session=request.getSession(false);
    MemberVO member = (MemberVO) session.getAttribute("member");
    String getId=member.getId();
    String getPw=member.getPw();

    Long bookmark_no = Long.parseLong(request.getParameter("bookmark_no"));
    String title=request.getParameter("title");
    String url=request.getParameter("url");
    String coment=request.getParameter("coment");
    String pw=request.getParameter("pw");

    String id = bookmarkService.checkWriter(bookmark_no);
    if(getPw.equals(pw) && getId.equals(id)) {
        try {
            BookmarkVO b = new BookmarkVO(title, url, coment, bookmark_no);
            bookmarkService.bookmarkUpdate(b);

            return bookmark_no+"번 글이 수정 되었습니다";
        }catch(Exception e) {
            return e.getMessage();
        }
    }else if(getId.equals("admin")){
        try {
            BookmarkVO b = new BookmarkVO(title, url, coment, bookmark_no);
            bookmarkService.bookmarkUpdate(b);

            return bookmark_no+"번 글이 수정 되었습니다";
        }catch(Exception e) {
            return e.getMessage();
        }
    }else {
        return "작성자가 아닙니다";
    }
}
```



게시물 삭제  
 삭제 시 게시물 번호와 일치하는 ID를 가져와  
 접속중인 세션의 ID와 비밀번호 비교 후 진행

```
// Bookmark Delete
@RequestMapping(value = "/bookmarkDelete", method= {RequestMethod.POST},
    produces = "application/text; charset=utf8")
@ResponseBody
public String bookmarkDelete( HttpServletRequest request,
    HttpServletResponse response) throws Exception {

    HttpSession session=request.getSession(false);
    MemberVO member = (MemberVO) session.getAttribute("member");

    String getPw = member.getPw();
    String getId = member.getId();
    String pw=request.getParameter("pw");

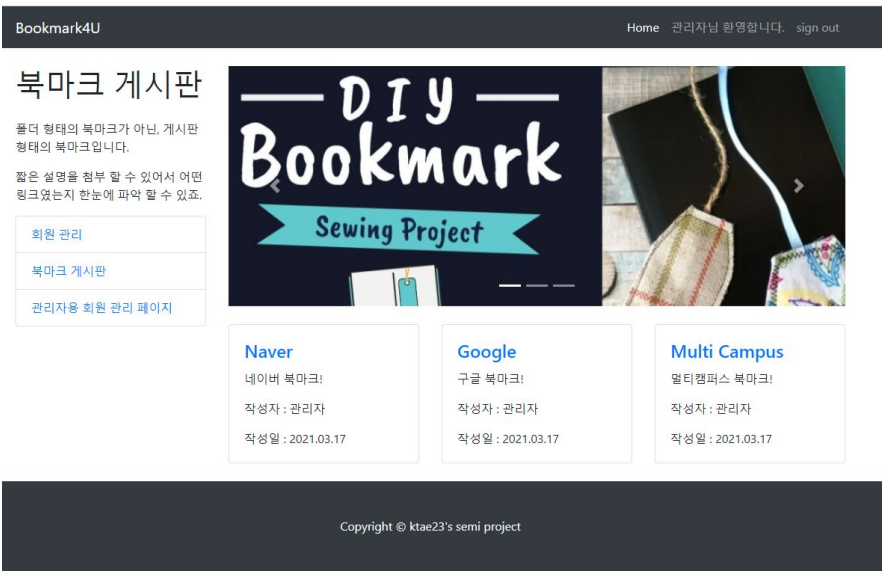
    Long bookmark_no = Long.parseLong(request.getParameter("bookmark_no"));
    String id = bookmarkService.checkWriter(bookmark_no);
    System.out.println("북마크 삭제" + bookmark_no);

    if(getPw.equals(pw) && getId.equals(id)) {
        try {
            BookmarkVO b = new BookmarkVO(bookmark_no);
            bookmarkService.bookmarkDelete(b);

            return bookmark_no+"번 글이 삭제 되었습니다";
        }catch(Exception e) {
            return e.getMessage();
        }
    }else if(getId.equals("admin")){
        try {
            BookmarkVO b = new BookmarkVO(bookmark_no);
            bookmarkService.bookmarkDelete(b);

            return bookmark_no+"번 글이 삭제 되었습니다";
        }catch(Exception e) {
            return e.getMessage();
        }
    }else {
        return "작성자가 아닙니다";
    }
}
```



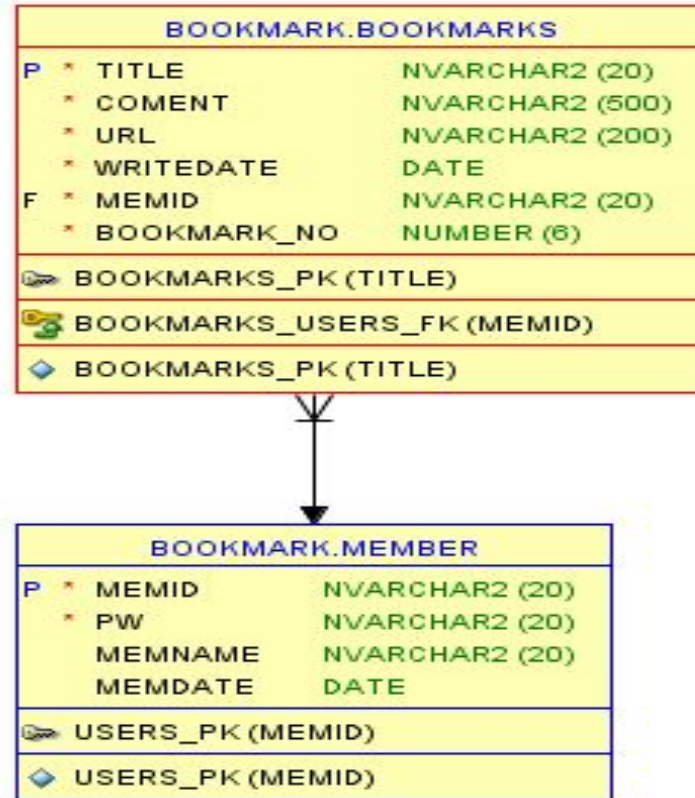


관리자로 접속 시 관리자용 페이지 메뉴 생성

관리자는 모든 회원 정보와 게시물에 대해 **CRUD** 권한이 있음








테이블 구조

# 느낀 점





확실히 이번 프로젝트를 통해서 에러 잡기, 구글링, 설정 읽기, 어떻게 요청을 주고 받는지 그 구조 등이 보였다.

정말 난 작은 점 같은 실력을 가졌고, **자료구조, 알고리즘, HTTP 통신, Web 구조, 코딩 테스트, TDD, Git flow... 등등**

앞으로 공부 해야 할 공부가 너무 너무 많고 4일간 의자 -> 침대 -> 의자 -> 침대 뿐이었지만 정말 재밌었다.

내가 무언갈 만들었고, 그게 동작한다는 것도 재밌지만

풀리지 않는 문제에 맞닥드렸을 때 이를 찾아서 해결하고 새로운 지식이 생기는 과정이 너무 재밌다.

방에 처박혀서 4일 간 집 밖에 한 발자국도 안나갔는데도 답답하지도 않고 친구들이 보자고 부르는 시간도 아까울 정도였다.

내 지식과 실력이 일천하지만 개발이란 것을 계속 하고 싶은 욕심이 난다.



감사합니다