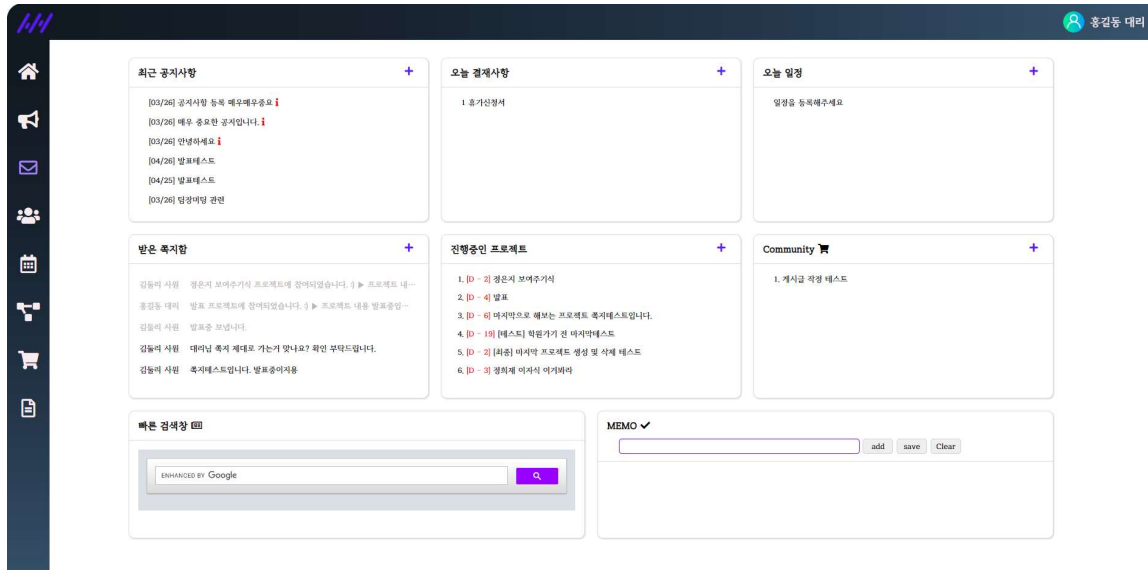




WEHUB

Final Project - Groupware

1. Main 화면 구현



주요 구현내용

1. 디자인

- 웹사이트의 느낌보다 시스템의 느낌이 나도록 디자인
- grid를 사용하여 구역을 나누었으며, ModelAndView를 사용해 구현기능 리스트를 불러와 한눈에 파악하기 쉽도록 디자인

2. 메모기능

- Ajax를 이용하여 비동기로 구현하였고, Stream을 사용해서 메모의 내용을 join하여 DB에 저장 및 업데이트가 가능하도록 구현

3. 빠른검색창

- Script를 추가하고, google developer 설정을 통해 구글검색창 구현

```
//메인 화면 띄우기
@RequestMapping(value="/main")
public ModelAndView mainPage(ModelAndView model,
    @SessionAttribute("loginMember")Member loginMember) {

    // project리스트 가져오기
    List<Project> projectList = null;

    projectList = projectService.homeProjectList(loginMember.getUser_no());

    for(int i = 0; i < projectList.size(); i++) {
        String[] arr = projectList.get(i).getParticipant().split("/ ");
        projectList.get(i).setProjectCount(arr.length);
    }

    //쪽지 리스트 가져오기
    List<Message> receiveList = null;

    Map<String, Object> map = new HashMap<String, Object>();
    map.put("msgTo", loginMember.getUser_no());

    receiveList = messageService.getHomeReceiveList(loginMember.getUser_no());

    //쪽지 아이콘 색 변하게 하는 코드
    int unreadCheck = 0;
    unreadCheck = messageService.getUnreadCheck(loginMember.getUser_no());

    // 메모리스트 가져오기
    String[] array = null;
    String content = null;
    List<String> memoContent = new ArrayList<>();
    List<Memo> memoList = memoService.getMemoList(loginMember.getUser_no());

    if(!memoList.isEmpty()) {
        content = memoList.get(0).getMemoContent();

        array = content.split("_");

        for(int i=0; i<array.length; i++) {
            memoContent.add(array[i]);
        }
    }

    //오늘 일정 Section
    DateTime dt = new DateTime();
    String today = dt.toString("yyyy M d");
    String[] arr = today.split(" ");

    List<DateData> todaySchedule = ScheduleService.todaySchedule(loginMember, arr);

    //커뮤니티 Section
    List<Community> communityList = communityService.selectMainList();

    //공지사항 Section
    List<Notice> NoticeList = boardService.selectList();
}
```

2. 공지사항 게시판 구현

게시판

공지사항

자유게시판

공지사항 검색

공지사항

No	제목	작성자	작성일자	첨부파일	댓글	조회
▲	공지사항 등록 매우매우중요	김둘리	2021. 3. 26. 오후 10:11:43		0	5
▲	매우 중요한 공지입니다.	김둘리	2021. 3. 26. 오후 9:42:43	-	0	2
▲	안녕하세요	김둘리	2021. 3. 26. 오후 2:56:27		0	34
▲	공지사항!!!	김둘리	2021. 3. 23. 오전 11:27:24		0	104
1	발표테스트	홍길동	2021. 4. 26. 오후 5:25:35	-	0	1
2	발표테스트	홍길동	2021. 4. 25. 오후 12:46:02	-	2	3
3	팀장미팅 관련	김영희	2021. 3. 26. 오후 11:20:31	-	1	14
4	댓글수 테스트	김둘리	2021. 3. 25. 오후 1:41:24	-	1	11
5	게시판상세조회테스트 + 수정테스트	홍길동	2021. 3. 25. 오전 12:28:24	-	0	4
6	게시판상세조회테스트	김둘리	2021. 3. 23. 오후 11:07:06		0	4
7	확인하세요	김둘리	2021. 3. 23. 오전 11:27:25	-	0	10
8	확인하세요	김둘리	2021. 3. 23. 오전 11:27:23		0	11
9	공지사항입니다	김둘리	2021. 3. 23. 오전 11:27:21		1	5
10	공지!!!	김둘리	2021. 3. 23. 오전 11:27:20		0	7

처음페이지 << 1 2 3 >> 마지막페이지

글쓰기

```

<select id="selectNoticeList" parameterType="map" resultMap="noticeResultMap">
  SELECT ROWNUM, A.* FROM (
    SELECT N.NOTICE_NO,
           N.NOTICE_WRITER_NO,
           M.USER_NAME,
           N.NOTICE_TITLE,
           N.NOTICE_CONTENT,
           N.NOTICE_ORIGINAL_FILENAME,
           N.NOTICE_READCOUNT,
           N.NOTICE_CREATE_DATE,
           N.NOTICE_MODIFY_DATE,
           N.STATUS,
           (SELECT COUNT(*)
            FROM COMMENTS
            WHERE COMMENT_NOTICE_NO = NOTICE_NO )AS COMMENTCOUNT
    FROM NOTICE N
    JOIN MEMBER M ON(N.NOTICE_WRITER_NO = M.USER_NO)
    WHERE (N.STATUS='Y' AND N.NOTICE_TYPE='N')
    <if test="searchList != 'null'">
      AND (N.NOTICE_TITLE LIKE '%' || #{searchText} || '%'
          OR N.NOTICE_CONTENT LIKE '%' || #{searchText} || '%')
    </if>
    ORDER BY N.NOTICE_NO DESC) A
</select>

<select id="noticeCount" parameterType="map" resultType="_int">
  SELECT COUNT(*)
  FROM NOTICE N
  WHERE [(STATUS='Y' AND NOTICE_TYPE='N')]
  <choose>
    <when test="searchList != 'null'">
      AND (N.NOTICE_TITLE LIKE '%' || #{searchText} || '%'
          OR N.NOTICE_CONTENT LIKE '%' || #{searchText} || '%')
    </when>
  </choose>
</select>

```

주요 구현내용

- 디자인
 - 필독게시글과 일반게시글을 2가지의 리스트를 가져와 구분하여 게시글의 중요도 표시
- 기능
 - 게시판 내용 및 제목 검색기능 구현
 - 쿼리 조회 시 검색 내용까지 포함하여 페이징 구현

2-1) 공지사항 게시판 상세보기 구현

게시판상세조회테스트

작성자 : 김둘리

부서 : 개발팀

2021-03-23 23:07:06

조회 : 5

첨부파일 : 제목 없는 디자인.png

게시판 상세조회테스트입니다.

확인부탁드립니다.

목록으로

Comments (1)

댓글을 입력하세요

등록

홍길동

수정 삭제

테스트 확인 완료하였습니다.

1. 게시글 구현

- 작성자명, 부서, 작성일
- 조회수를 쿠키에 저장시키는 조회수증가 로직을 구현
- 첨부파일 클릭 시 다운로드 가능
- ajax를 이용하여 비동기로 댓글 등록, 수정, 삭제 가능하게 구현

```
//조회수 로직
Cookie[] cookies = request.getCookies();
String boardHistory = "";
boolean hasRead = false;

if(cookies != null) {
    String name = null;
    String value = null;

    for(Cookie cookie : cookies) {
        name = cookie.getName();
        value = cookie.getValue();

        //boardHistory인 쿠키값을 찾기
        if("boardHistory".equals(name)) {
            boardHistory = value; //현재저장된 값 대입
            if(value.contains("|" + noticeNo + "|")) {
                //읽은 게시글
                hasRead = true;
                break;
            }
        }
    }
}

if(!hasRead) {
    Cookie cookie = new Cookie("boardHistory", boardHistory + "|" + noticeNo + "|");

    cookie.setMaxAge(-1);
    response.addCookie(cookie);
}

Notice notice = service.findNoticeByNo(noticeNo);

if(!hasRead) {
    // DB저장로직 작성하기

    int noticeReadCount = notice.getNoticeReadCount() + 1;

    Map<String, Object> map = new HashMap<String, Object>();

    map.put("noticeNo", noticeNo);
    map.put("noticeReadCount", noticeReadCount);

    notice.setNoticeReadCount(service.updateReadCount(map));
}
```

2-1) 공지사항 게시판 파일관련 메소드 코드

```
private String saveFile(MultipartFile file, HttpServletRequest request) {
    String renamePath = null;
    String originalFileName = null;
    String renameFileName = null;
    String rootPath = request.getSession().getServletContext().getRealPath("resources");
    String savePath = rootPath + "/upload/notice";

    log.info("Root Path : " + rootPath);
    log.info("Save Path : " + savePath);

    //SavePath가 실제로 존재하지 않으면 폴더를 생성하는 로직
    File folder = new File(savePath);

    if(!folder.exists()) {
        folder.mkdirs();
    }

    originalFileName = file.getOriginalFilename();
    renameFileName =
        LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyyMMdd_HHmssSSS")) +
        originalFileName.substring(originalFileName.lastIndexOf(".")); //확장자가 없는 파일 확인하러
    renamePath = savePath + "/" + renameFileName;

    try {
        //업로드한 파일 데이터를 지정한 파일에 저장한다.
        file.transferTo(new File(renamePath));
    } catch (IOException e) {
        System.out.println("파일 전송에러 : " + e.getMessage());
        e.printStackTrace();
    }

    return renameFileName;
}
```

```
@RequestMapping(value="notice/fileDown", method = { RequestMethod.GET })
public ResponseEntity<Resource> fileDown(
    @RequestParam("oriname") String oriname, @RequestParam("rename") String rename,
    @RequestHeader(name="user-agent") String header) {

    try {
        Resource resource = resourceLoader.getResource("resources/upload/notice/" + rename);
        File file = resource.getFile();
        boolean isMSIE = header.indexOf("Trident")!=-1||header.indexOf("MSIE")!=-1;
        String encodeRename = "";

        if(isMSIE) {
            encodeRename = URLEncoder.encode(oriname, "UTF-8");
            encodeRename = encodeRename.replaceAll("\\+", "%20");
        } else {
            encodeRename = new String(oriname.getBytes("UTF-8"), "ISO-8859-1");
        }

        return ResponseEntity.ok()
            .header(HttpHeaders.CONTENT_DISPOSITION, "attachment;filename=\"" + encodeRename + "\"")
            .header(HttpHeaders.CONTENT_LENGTH, String.valueOf(file.length()))
            .header(HttpHeaders.CONTENT_TYPE, MediaType.APPLICATION_OCTET_STREAM.toString())
            .body(resource);
    } catch (IOException e) {
        e.printStackTrace();

        return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).build();
    }
}
```

2-2) 공지사항 게시판 글쓰기 구현

게시판

공지사항

자유게시판

공지사항 검색

Go

☐ 중요공지 ▲

제목

작성자가

홍길동

내용

첨부파일

파일 선택

선택된 파일 없음

등록하기

목록으로

1. 게시글 작성/수정 화면 구현

- 중요공지 체크 시 필독 게시물로 등록
- 수정하기로 체크해제 가능
- MultipartFile로 파일 업로드 가능

```

@RequestMapping(value = "notice/write", method = {RequestMethod.POST})
public ModelAndView noticeWrite(
    @SessionAttribute(name="loginMember", required=false) Member loginMember,
    HttpServletRequest request, Notice notice, @RequestParam("upfile") MultipartFile upfile, ModelAndView model) {

    if(notice.getNoticeType() == null) {
        notice.setNoticeType("N");
    } else {
        notice.setNoticeType("Y");
    }

    //꼭지 아이콘 색 변하게 하는 코드
    int unreadCheck = 0;
    unreadCheck = messageService.getUnreadCheck(loginMember.getUser_no());

    int result = 0;

    if(loginMember.getUser_id().equals(notice.getUserId())) {
        notice.setNoticeWriterNo(loginMember.getUser_no());

        System.out.println(loginMember.getUser_no());

        if(upfile != null && !upfile.isEmpty()) {
            // 파일들 저장하는 로직 작성
            String renameFileName = saveFile(upfile, request);

            System.out.println(renameFileName);

            if(renameFileName != null) {
                notice.setNoticeOriginalFileName(upfile.getOriginalFilename());
                notice.setNoticeRenamedFileName(renameFileName);
            }
        }

        result = service.saveBoard(notice);

        System.out.println(result);

        if(result > 0) {
            model.addObject("msg", "게시글이 정상적으로 등록되었습니다.");
            model.addObject("location", "/notice/list");
            model.addObject("unreadCheck",unreadCheck);
        } else {
            model.addObject("msg", "게시글 등록에 실패하였습니다.");
            model.addObject("location", "/notice/list");
            model.addObject("unreadCheck",unreadCheck);
        }
    } else {
        model.addObject("msg", "잠궜된 접근입니다.");
        model.addObject("location", "/home");
    }

    model.setViewName("common/msg");

    return model;
}
    
```


3-1) 받은 쪽지함 구현

```
//쪽지 아이콘 색 변하게 하는 코드
int unreadCheck = 0;
unreadCheck = messageService.getUnreadCheck(loginMember.getUser_no());
```

2. 쪽지아이콘
- 읽지 않은 쪽지가 있을 시
아이콘의 색이 변경되도록
구현

3. 체크박스
- 전체 선택 또는 원하는 쪽지만
선택하여 삭제, 또는 보관함
이동이 가능하게 구현
- 원하는 쪽지만 선택하여
읽지않은 상태로 변경 가능

1. 검색기능
- 전체, 이름, 내용을 선택하여 검색 가능

```
function chkAll() {
  if ($("#checkAll").is(':checked')) {
    $("input[type=checkbox]").prop("checked", true);
  } else {
    $("input[type=checkbox]").prop("checked", false);
  }
}
```

```
var cnt = $("input[name='chk']:checked").length;
var arr = new Array();

$("input[name='chk']:checked").each(function() {
  arr.push($(this).attr('value'));
});
```

4. 받은 쪽지
- 읽지 않은 쪽지와 읽은 쪽지의 색을 구분하여
한눈에 확인할 수 있게 구현
- 쪽지 내용 클릭 시 받은 쪽지 모달로 간편하게
상세확인 가능
- 삭제 클릭시 삭제기능 구현(휴지통으로 이동)
- 답장 클릭 시 쪽지를 보낸 사원이 자동으로 입력됨

3-2) 보낸 쪽지함 구현



보낸 쪽지함 주요기능

- 보낸 쪽지함에서는 영구삭제만 가능하도록 구현
- 보낸날짜확인 및 수신확인 구현
- 전체삭제, 전체 보관함 이동 가능
- 검색기능 구현

```
-- 쪽지테이블
CREATE TABLE SEND_MSG (
    SEND_NO NUMBER PRIMARY KEY,
    SENDER_NO NUMBER NOT NULL,
    RECEIVER_NO NUMBER NOT NULL,
    SEND_CONTENT VARCHAR2(1000),
    SEND_DATE DATE DEFAULT SYSDATE,
    DELETE_DATE DATE DEFAULT SYSDATE,
    STATUS VARCHAR2(1) DEFAULT 'Y' CHECK (STATUS IN ('Y', 'N', 'S')),
    CONSTRAINT FK_SENDER_NO FOREIGN KEY (SENDER_NO) REFERENCES MEMBER (USER_NO) ON DELETE SET NULL,
    CONSTRAINT FK_RECEIVER_NO FOREIGN KEY (RECEIVER_NO) REFERENCES MEMBER (USER_NO) ON DELETE SET NULL
);
```

```
function detailMsg(msgNo){
    var msgNo = msgNo;
    var readCheck = $("#readCheck"+ msgNo).val();

    console.log(readCheck);
    console.log(msgNo);

    if(readCheck == 'N'){
        $.ajax({
            type: "get",
            url: "${path}/message/readCheck",
            data:{
                msgNo:msgNo
            },
            success:function(data){
                document.querySelector("#modalNo" + msgNo).classList.remove("fade");
            },
            error: function(e){
                alert("실패");
                console.log(e);
            }
        });
    }

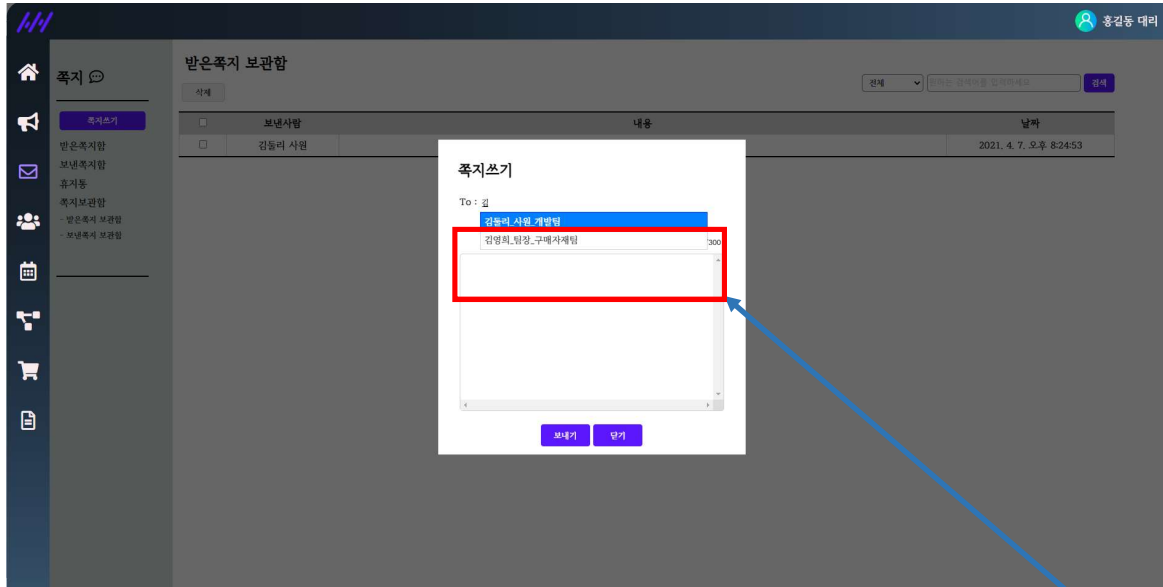
    const open = () => {
        document.querySelector("#modalNo" + msgNo).classList.remove("fade");
    }

    const close = () => {
        document.querySelector("#modalNo" + msgNo).classList.add("fade");
    }

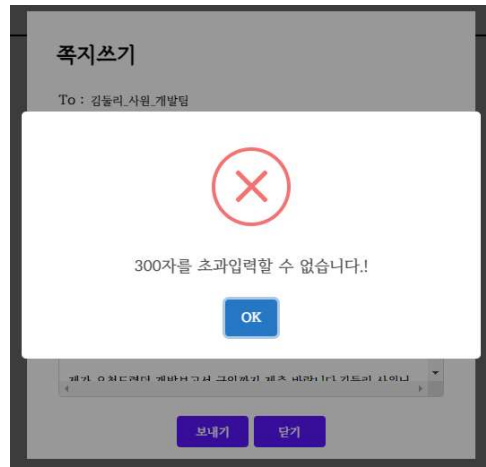
    document.getElementById('detail'+ msgNo + '').addEventListener("click", open);
    document.getElementById('exitBtn'+ msgNo + '').addEventListener("click", close);
    document.querySelector(".bg").addEventListener("click", close);

}
```


3-3) 쪽지쓰기 구현



```
$(document).ready(() => {
    $("#memSearchInput").autocomplete({
        source: function(request, response){
            console.log($("#memSearchInput").val());
            $.ajax({
                url: "${path}/search/json",
                type: "get",
                dataType: 'json',
                data: {
                    userName: $("#memSearchInput").val()
                },
                success: function(data){
                    var result = data;
                    response(result);
                },
                error: function(e){
                    alert("ajax에러발생..!")
                }
            });
        },
        focus: function(event, ui) {
            return false;
        },
        minLength: 1, // 최소 글자수
        autoFocus: true,
        delay: 300, // 검색창에 글자 써지고 나서 autocomplete 창 뜰 때 까지 딜레이 시간(ms)
    });
});
```



쪽지쓰기 주요기능

- 받은쪽지함, 보낸쪽지함, 휴지통, 보관함 등에서 쪽지쓰기 모두 가능
- 받는사람 입력 시 jQuery ui Autocomplete 라이브러리를 사용하여 입력한 글자에 해당하는 사원이 모두 나올 수 있게 구현
- 쪽지 내용은 총 300자까지 입력 가능하며, 초과될 시 300자를 초과할 수 없다는 Alert창 구현

4. 프로젝트 공유 게시판 구현



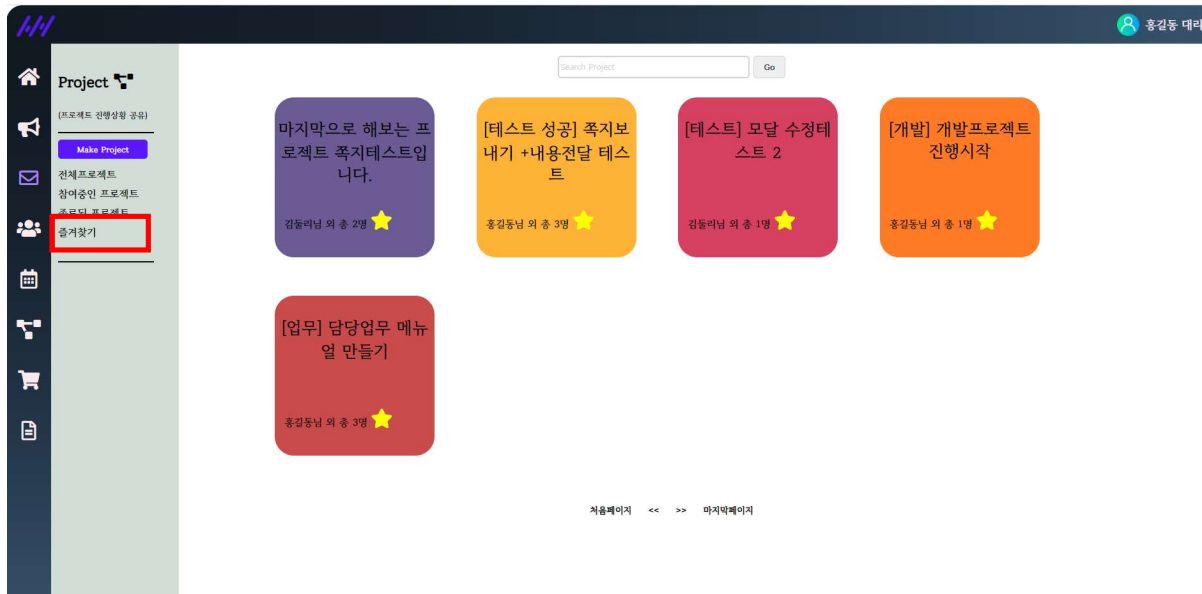
프로젝트 공유게시판 주요기능

- 디자인
 - 프로젝트의 공유 게시판이기 때문에 생동감 있도록 div로 디자인하였고 해당 div마다 색을 지정하여 프로젝트 등록 시 색 자동변환 가능
- 기능
 - Ajax로 개인별 즐겨찾기 추가 기능 구현
 - 원하는 프로젝트 등록 가능
 - Modal을 사용하여 프로젝트상세보기 화면 구현
 - 프로젝트 생성자 및 참가자, 내용, D-day 확인가능
 - 프로젝트 생성자만 프로젝트 종료 가능
 - 프로젝트 종료된 프로젝트에서 이력확인 가능
 - 본인이 프로젝트를 생성했거나, 참가자에 포함된 경우 참여중인 프로젝트에서 확인 가능

```
function closeProject(proNo){
    console.log(proNo);
    Swal.fire({
        title: '프로젝트를 정말 종료하시겠습니까?',
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',
        cancelButtonColor: '#d33',
        confirmButtonText: '네 종료하겠습니다!'
    }).then((result) => {
        if (result.isConfirmed) {
            $.ajax({
                type: "post",
                url: "${path}/project/close",
                data: {
                    proNo: proNo
                },
                success: function(data){
                    console.log(data);

                    var closeProject = document.getElementById('projectDivNo' + proNo);
                    closeProject.remove();
                    document.querySelector(".proNo" + proNo).classList.add("fade");
                },
                error: function(e){
                    Swal.fire({
                        icon: 'error',
                        text: '프로젝트 종료 실패! 관리자에게 문의하세요'
                    })
                    console.log(e);
                }
            });
        }
    })
    Swal.fire({
        icon: 'success',
        text: '프로젝트가 성공적으로 종료되었습니다.'
    })
}
```

4-1) 프로젝트 공유 게시판 즐겨찾기



```
<select id="getFavList" parameterType="map" resultMap="projectResultMap">
  SELECT P.PROJECT_NO,
  P.PROJECT_MAKE,
  P.PARTICIPANT,
  P.PROJECT_TITLE,
  P.PROJECT_CONTENT,
  P.START_DATE,
  P.DUE_DATE,
  P.STATUS,
  M.USER_NAME,
  M.RANK,
  D.DEPT_NAME,
  (SELECT COUNT(*) FROM PROJECT) AS PCOUNT,
  NVL2(B.USER_ID, 'Y', 'N') AS BOOKMARK
FROM PROJECT P
JOIN (SELECT USER_ID, PROJECT_NO FROM BOOKMARK WHERE USER_ID=#{userNo}) B
ON (P.PROJECT_NO = B.PROJECT_NO)
JOIN MEMBER M ON(P.PROJECT_MAKE = M.USER_NO)
JOIN DEPARTMENT D ON(M.DEPT_CODE = D.DEPT_CODE)
WHERE P.STATUS = 'Y'
<if test="searchText != 'null'">
  AND (P.PROJECT_CONTENT LIKE '%||#{searchText}||'%'
  OR P.PROJECT_TITLE LIKE '%||#{searchText}||'%'
  OR P.PARTICIPANT LIKE '%||#{searchText}||'%'
  OR M.USER_NAME LIKE '%||#{searchText}||'%'
</if>
ORDER BY P.PROJECT_NO DESC
</select>
```

북마크 테이블

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 BOOKMARK_NO	NUMBER	No	(null)	1	북마크번호
2 USER_ID	NUMBER	No	(null)	2	사번
3 PROJECT_NO	NUMBER	No	(null)	3	프로젝트번호

* 개인별 즐겨찾기 기능

- Ajax를 사용하여 클릭시 비동기로 즐겨찾기 별 색이 바뀔 수 있도록 구현
- 즐겨찾기 리스트 조회 시 북마크 테이블에서 해당 userNo를 조회하는 select문을 join하여 해당 사원이 즐겨찾기한 게시글만 표시할 수 있도록 구현

```
function clickFav(proNum){
  var className = $('#favBtn'+ proNum).attr('class');
  $.ajax({
    type: "post",
    url: "${path}/project/makeFav",
    data:{
      proNum:proNum
    },
    success:function(data){
      if(className == 'favY'){
        $('#favBtn'+ proNum).attr('class', 'favN');
        $('#favBtn'+ proNum).css('color', 'white');
      }else {
        $('#favBtn'+ proNum).attr('class', 'favY');
        $('#favBtn'+ proNum).css('color', 'yellow');
      }
    },
    error: function(e){
      alert("즐거찾기 실패");
      console.log(e);
    }
  });
}
```

4-2) 프로젝트 생성

프로젝트 생성

프로젝트명 :

참가자 : 추가

진행기간 : ~

일 월 화 수 목 금 토

2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

30 31

프로젝트 생성 닫기

```
function deleteMem(val){
    var deleteMember = document.getElementById(val);
    deleteMember.remove();
}

$(document).ready(function () {
    $.datepicker.setDefaults($.datepicker.regional['ko']);
    $('#startDate').datepicker({
        changeMonth: true,
        changeYear: true,
        nextText: '다음 달',
        prevText: '이전 달',
        dayNames: ['일요일', '월요일', '화요일', '수요일', '목요일', '금요일', '토요일'],
        dayNamesMin: ['일', '월', '화', '수', '목', '금', '토'],
        monthNamesShort: ['1월', '2월', '3월', '4월', '5월', '6월', '7월', '8월', '9월', '10월', '11월', '12월'],
        monthNames: ['1월', '2월', '3월', '4월', '5월', '6월', '7월', '8월', '9월', '10월', '11월', '12월'],
        dateFormat: 'yy-mm-dd',
        maxDate: 0,
        onClose: function( selectedDate ) {
            // 시작일(startDate) datepicker가 닫힐때
            // 종료일(endDate)의 선택할수있는 최소 날짜(minDate)를 선택한 시작일로 지정
            $('#endDate').datepicker( "option", "minDate", selectedDate );
        }
    });
    $('#endDate').datepicker({
        changeMonth: true,
        changeYear: true,
        nextText: '다음 달',
        prevText: '이전 달',
        dayNames: ['일요일', '월요일', '화요일', '수요일', '목요일', '금요일', '토요일'],
        dayNamesMin: ['일', '월', '화', '수', '목', '금', '토'],
        monthNamesShort: ['1월', '2월', '3월', '4월', '5월', '6월', '7월', '8월', '9월', '10월', '11월', '12월'],
        monthNames: ['1월', '2월', '3월', '4월', '5월', '6월', '7월', '8월', '9월', '10월', '11월', '12월'],
        dateFormat: 'yy-mm-dd',
        onClose: function( selectedDate ) {
            // 종료일(endDate) datepicker가 닫힐때
            // 시작일(startDate)의 선택할수있는 최대 날짜(maxDate)를 선택한 시작일로 지정
            $('#startDate').datepicker( 'setDate', 'today' );
        }
    });
});
```

1. 참가자

- 참가자는 jQuery 자동완성 기능으로 구현
- 추가버튼을 통해 참가자에 추가 가능, X버튼 클릭시 해당 참가자 삭제 가능

프로젝트 생성

프로젝트명 :

참가자 : 추가

진행기간 : ~

프로젝트 생성 닫기

3. 프로젝트 진행기간

- 프로젝트 진행기간은 jQuery datepicker를 이용하여 날짜를 선택 가능하도록 구현

4. 프로젝트 생성

- 프로젝트 생성버튼 클릭 시 해당 프로젝트의 참가자들에게 프로젝트 참여 쪽지 발송

프로젝트 생성

프로젝트명 :

참가자 : 추가

진행기간 : ~

프로젝트 생성 닫기

2. Alert

- 프로젝트명, 참가자, 진행기간, 내용 등 입력하지 않은 부분이 있을 경우 SweetAlert으로 경고 Alert창 생성
- SweetAlert을 사용하여 더 직관적이고 디자인적으로 완성된 화면 구성

받은쪽지

From : 홍길동 대리 (개발팀)
Date : 2021. 4. 10. 오후 1:48:44

[테스트 성공] 쪽지보내기 +내용전달 테스트 프로젝트에 참여되었습니다. :)

프로젝트 내용 :
프로젝트 목표 : 프로젝트 생성과 동시에 참가자들에게 쪽지가 보내지는것을 목표로 한다.

프로젝트 단계
1. split을 사용하여 참가자들의 사번을 조회해온다.
2. list에 객체추가하여
3. 마다아디노 라이브러리 가져와서 쪽지보내기

삭제 답장 닫기

4-2) 프로젝트 생성 쪽지발송 쿼리

Project

(프로젝트 현황창 공유)

Make Project

현재 프로젝트

참여중인 프로젝트

종료된 프로젝트

즐거찾기

프로젝트 생성

프로젝트명 :

참가자 : 01

도우미, 사원, 인사팀

김도미, 사원, 개발팀 / 홍길동, 대리, 개발팀 / 박도치, 차장, 개발팀 / 박은진, 과장, 개발팀 /

프로젝트 진행기간 :

(0/300)

프로젝트 생성

닫기

```
<insert id="sendProjectMsg" parameterType="java.util.List">
    INSERT INTO MESSAGE (
        MSG_NO,
        MSG_TO,
        MSG_FROM,
        MSG_CONTENT,
        CREATE_DATE,
        DELETE_DATE,
        READ_DATE,
        READ_STATUS,
        STATUS,
        SEND_STATUS
    )
    SELECT SEQ_MSG_NO.NEXTVAL AS MSG_NO
        , T.*
    FROM(
        <foreach collection="list" item="item" index="index" separator="UNION ALL">
            SELECT
                #{item.msgTo} AS MSG_TO,
                #{item.msgFrom} AS MSG_FROM,
                #{item.msgContent} AS MSG_CONTENT,
                SYSDATE AS CREATE_DATE,
                SYSDATE AS DELETE_DATE,
                SYSDATE AS READ_DATE,
                'N' AS READ_STATUS,
                'Y' AS STATUS,
                'Y' AS SEND_STATUS
            FROM DUAL
        </foreach>
    ) T
</insert>
```

받은쪽지

From : 홍길동 대리 (개발팀)

Date : 2021. 4. 10. 오후 1:48:44

[테스트 성공] 쪽지보내기 +내용전달 테스트 프로젝트에 참여되었습니다. :)

프로젝트 내용 :

프로젝트 목표 : 프로젝트 생성과 동시에 참가자들에게 쪽지가 보내지는것을 목표로 한다.

프로젝트 단계

1. split을 사용하여 참가자들의 사반을 조화해온다.
2. list에 객체추가하여
3. 마드아디노 크리모 자서천여 쪽지보내기

삭제

답장

닫기

5. 회원가입 시 사번조회 기능

Create an account

Welcome

사번입력 click!

파일 선택 선택된 파일 없음

이름 UserName

직급 Rank

부서이름 DepartmentName

이메일 E-Mail

아이디 Userid

아이디 중복 확인

패스워드 Password

패스워드 확인 PasswordCheck

주소 찾기

우편번호 Zip/Postal Code

주소 State/Province/Region/City

상세주소 Street Address

내선번호 ExtensionNumber

휴대번호 Phone

가입하기

벌써 회원이신가요?

사번을 입력해 주세요

사번을 입력해 주세요 :)

OK Cancel

사번입력 click!

파일 선택 선택된 파일 없음

주소 State/Province/Region/City

상세주소 Street Address

내선번호 ExtensionNumber

휴대번호 Phone

가

사번입력 click!

파일 선택 선택된 파일 없음

가

벌써 회원이신가요?

이미 가입된 사번입니다!!

OK

주소 State/Province/Region/City

상세주소 Street Address

내선번호 ExtensionNumber

휴대번호 Phone

가입하기

벌써 회원이신가요?

유효하지 않은 사번입니다!!

OK

주소 State/Province/Region/City

상세주소 Street Address

내선번호 ExtensionNumber

휴대번호 Phone

가입하기

벌써 회원이신가요?

사번입력버튼

- 회원가입 화면에서 인사팀에서 등록된 신입사원인지 확인하는 기능 구현
- 인명관리 탭에서 신입사원 DB에 추가된 후 추가된 사번 입력 시 올바른 사번이면 해당 사번 자동입력
- 사번 입력버튼 클릭시 사번입력 alert창 구현
- 유효하지 않은 사번 또는 이미 가입된 사번 입력시 경고 Alert

5. 회원가입 시 사번조회 기능

Create an account

Welcome

사번입력 click!

파일 선택 선택된 파일 없음

정온지

사원

W01

이메일 E-Mail

아이디 Userid

아이디 중복 확인

패스워드 Password

패스워드 확인 PasswordCheck

주소 찾기

우편번호 Zip/Postal Code

주소 State/Province/Region/City

상세주소 Street Address

내선번호 ExtensionNumber

휴대번호 Phone

가입하기

별써 회원이신가요?

```
function findNewMem(){
    const { value: userNo } = Swal.fire({
        title: '사번을 입력해 주세요',
        input: 'text',
        inputLabel: '사번을 입력해 주세요 :)',
        showCancelButton: true,
        inputValidator: (value) => {
            console.log(value);

            $.ajax({
                type: "post",
                url: "${path}/member/findNewMem",
                dataType: "json",
                data: {
                    value: value
                },
                success: function(data) {
                    if(data.member != 0 && data.member != 1){
                        $("#user_name").attr('class', 'newMemInput');
                        $("#rank").attr('class', 'newMemInput');
                        $("#dept_code").attr('class', 'newMemInput');

                        document.getElementById('newUserNo').value = data.member.new_no;
                        document.getElementById('user_name').value = data.member.new_name;
                        document.getElementById('rank').value = data.member.new_rank;
                        document.getElementById('dept_code').value = data.member.new_dept;

                    } else if(data.member == 0){
                        document.getElementById('newUserNo').value = "";
                        document.getElementById('user_name').value = "";
                        document.getElementById('rank').value = "";
                        document.getElementById('dept_code').value = "";

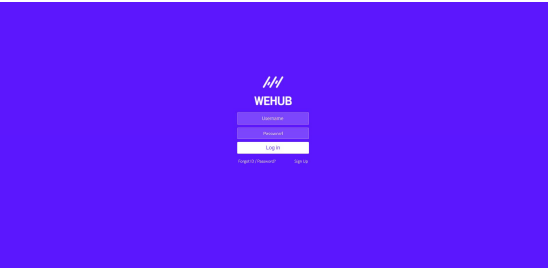
                        Swal.fire({
                            icon: 'error',
                            text: "유효하지 않은 사번입니다.!",
                        })
                    } else if(data.member == 1){
                        document.getElementById('newUserNo').value = "";
                        document.getElementById('user_name').value = "";
                        document.getElementById('rank').value = "";
                        document.getElementById('dept_code').value = "";

                        Swal.fire({
                            icon: 'error',
                            text: "이미 가입된 사번입니다.!",
                        })
                    }
                },
                error: function(e) {

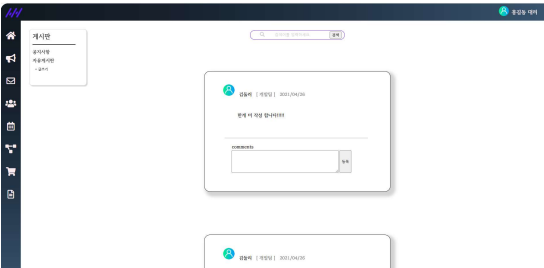
                }
            });

            if (!value) {
                return 'You need to write something!'
            }
        }
    });
};
```

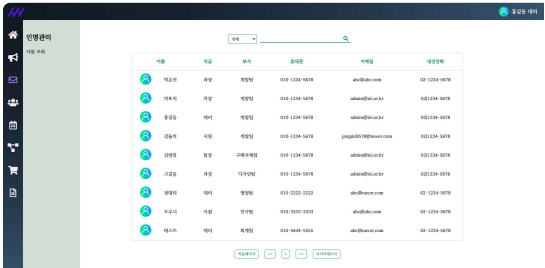
6. 추가기능 화면



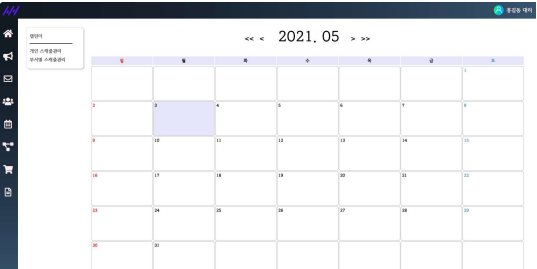
1. 로그인화면



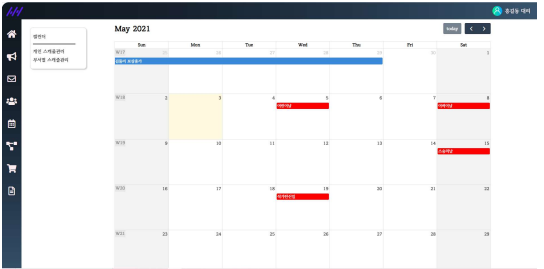
2. 자유게시판 화면



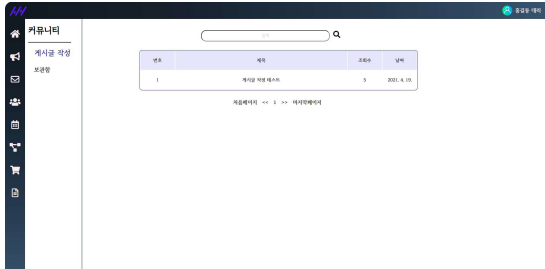
3. 인명관리 게시판



4. 개인별 일정관리 캘린더



5. 부서별 일정관리 캘린더



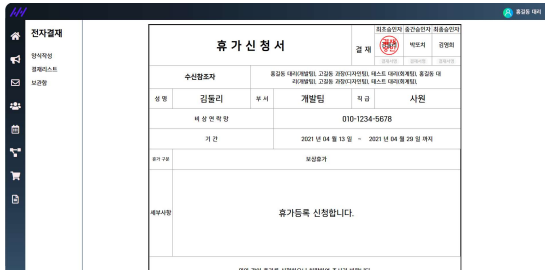
6. 중고거래 게시판 화면



7. 중고거래 게시판 글쓰기(CK editor)



8. 전자결재 결재현황 화면 구현



9. 전자결재 결재 화면 구현