

```
public Portfolio getPortfolio()  
{ Kim Kwang Hoon }
```

김광훈 포트폴리오 010. 5136. 3964

Mail. kim772882@naver.com

Web. 2020.11월 10일 오픈예정

Git. github.com/kim-kwanghoon?tab=repositories



public void index()



01. 팀프로젝트

그룹웨어 프로젝트

- 개발 배경 및 목적
- 개발 환경 및 서비스 개요
- 작업기간
- 서비스 개요
- 프로젝트 ERD
- 기능 구현

02. 개인프로젝트

개인소개 사이트

- 개발 배경 및 목적
- 개발 환경 및 서비스 개요
- 서비스 개요 / 프로젝트 ERD
- 기능 구현



```
public void myPart()
```

```
{ 그룹웨어 전자결재 }
```

```
    팀프로젝트
```

String 개발 배경 및 목적;

많은 회사들이 이용하고 있는 그룹웨어 솔루션으로,
사내 업무 및 자원관리를 효율적으로 하기 위한 웹
애플리케이션



포폴과 이미지는 무관합니다

String 개발 환경 및 서비스 개요;

FrontEnd



JavaScript



jQuery



HTML5



CSS3



Ajax

WAS



Apache Tomcat
9.0

Database



MySQL 8.0

BackEnd



Spring 4.3



MyBatis



Java 1.8



JSP

String 작업기간;

작업기간	1 week	2 week	3 week	4 week	5 week
프로젝트 기획	<div></div>				
UI 설계 및 구현	<div></div>				
데이터 베이스 설계		<div></div>			
기능 구현			<div></div>		
백엔드 프로그래밍			<div></div>		
테스트 및 디버깅				<div></div>	

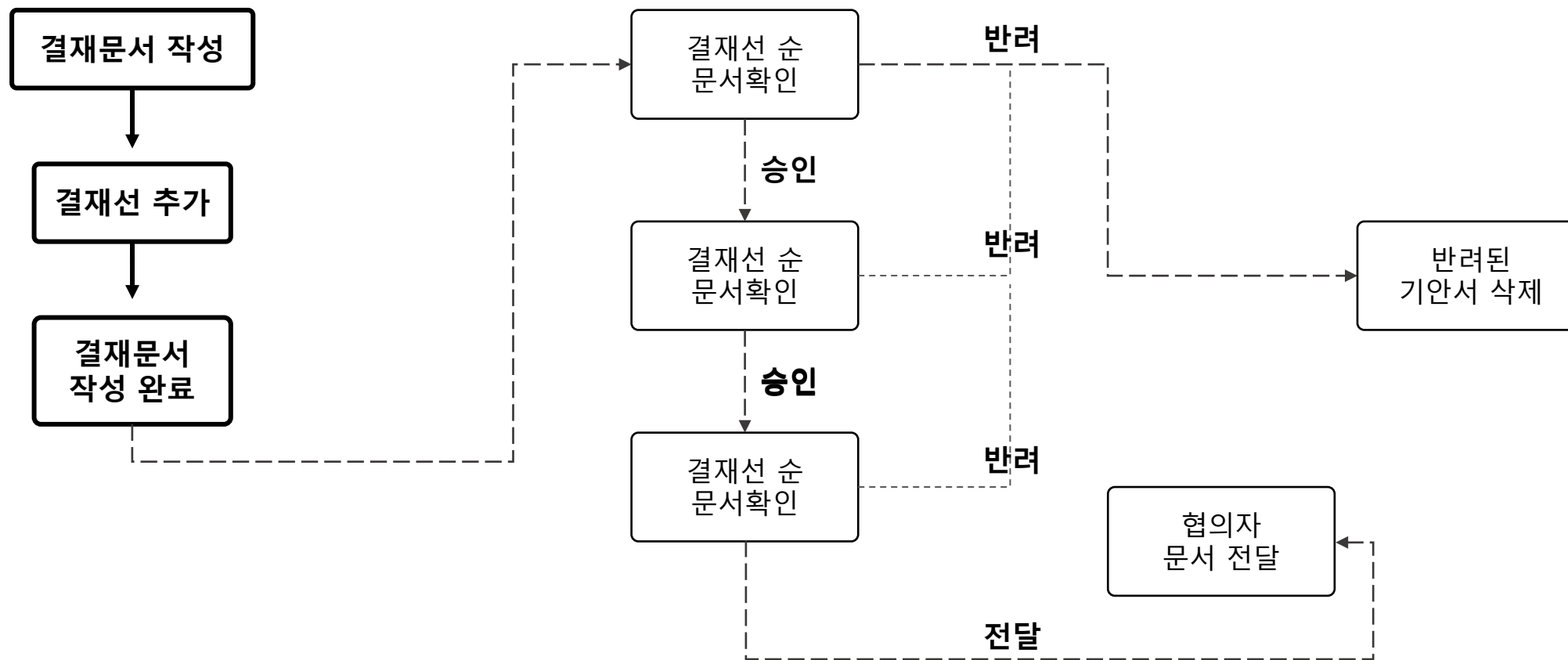
String 서비스 개요;

기능리스트

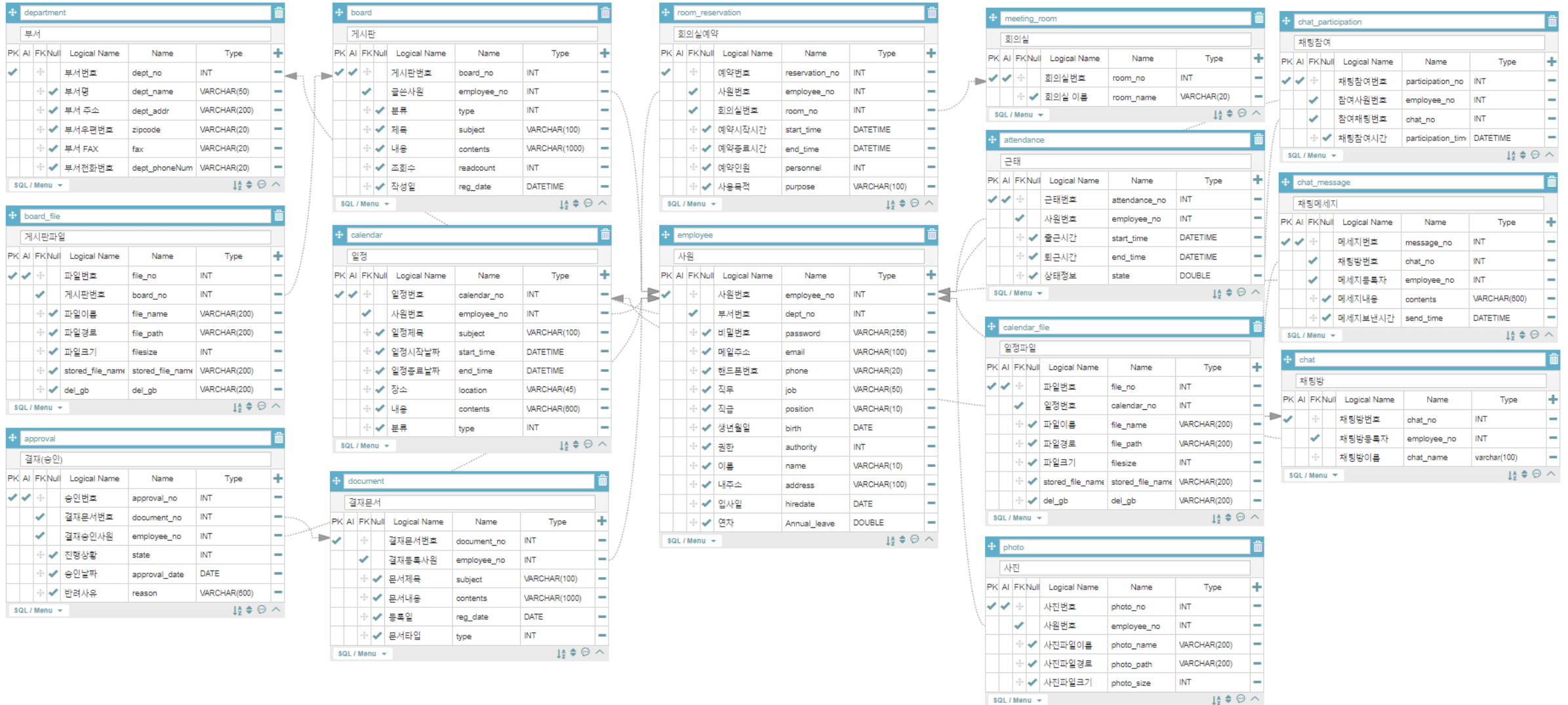


String 서비스 개요;

전자결재 프로세스



String 프로젝트 전체 ERD;



String 기능 구현 전 방법에 대한 고민;

1. 구현되어 있지 않은 기능 연동을 어떻게 하지?
 - 가상의 설정값으로 작업하고 연동될 기능이 완성된 후 연동 작업
2. 사용자에게 편리한 문서 작성을 위한 방법은?
 - 네이버 스마트 에디터 API를 사용하여 문서작성의 다양한 기능을 한번에 가져오도록 작업
3. 결재가 완료된 문서 보는 방법
 - 결재테이블 state(상태)의 합산 결과를 통해 승인여부 확인
4. 결재 진행 상태에 따른 표기를 어떻게 해야 할까?
 - DB에서 숫자로 가져온 상태 값을 JSP 부분에서 자바스크립트로 한글로 변환
5. 결재가 포함된 직원 순서로 결재 문서를 보여줄 방법은
 - 결재선에 자신이 포함된 문서를 찾고 문서 속 승인번호가 자신보다 적은 사람의 상태 값을 판단하여 가져옴



[0] 전자결재 문서 작성하기

내가 작성한 기안서			
문서번호	제목	작성일	타입
71	휴가신청서 20200930 - 20201001	2020. 9. 29	1

View "domtList"

전자결재작성

결재문서 리스트				
NO	시스템	서식함	양식명	담당자
5	문서결재	공통	휴가신청서	-
4	문서결재	공통	업무일지	-

View "domtpage/1.do"

휴가신청서

결

제

기안

김광훈

1

문서번호	작성일	작성부서	작성자
자동채번(77)	2020-10-14	영업팀	김광훈

제목을 입력하세요

내용

2

SmartEditor2 API 적용

1. 문서 작성시 기본정보 자동 입력

```

<h2>휴가신청서</h2>
<table id="domtinfoTable">
  <tr>
    <td>문서번호</td>
    <td>작성일</td>
    <td>작성부서</td>
    <td>작성자</td>
  </tr>
  <tr>
    <td>자동채번(${document_no})</td>
    <td id="now_date"></td>
    <td>${dept_name}</td>
    <td>${name}</td>
  </tr>
</table>

```

```

<script type="text/javascript">
  $(document).ready(
    function() {
      var date = new Date();
      var yyyy = date.getFullYear();
      var mm = new String(date.getMonth() + 1);
      var dd = new String(date.getDate());
      var nowdate = yyyy + "-" + (("00" + mm.toString()).slice(-2))
        + "-" + (("00" + dd.toString()).slice(-2));
      $("#now_date").html(nowdate);
      $(".now_date2").attr("value", nowdate);
    }
  );
</script>

```

2. Naver 스마트 에디터 API

```

<textarea class="textar" name="contents" id="naversm" class="naversm"></textarea>
<script>
  var oEditors = [];
  $(function() {
    nhn.husky.EZCreator.createInIFrame({
      oAppRef: oEditors,
      elPlaceHolder: "naversm",
      sSkinURI : "<%=request.getContextPath()%>/resources/smartereditor/SmartEditor2Skin.html",
      htParams : {
        bUseToolbar : true, // 툴바 사용 여부
        bUseVerticalResizer : true, // 입력창 크기 조절바 사용 여부
        bUseModeChanger : true, // 모드 탭(Editor | HTML | TEXT) 사용 여부
        fOnBeforeUnload : function() {
        }
      },
      fOnAppLoad : function() {
        oEditors.getByName("naversm").exec("PASTE_HTML",
          [ contents ]);
      },
      fCreator : "createSEditor2"
    });
    $("#save").click(function() {
      oEditors.getByName("naversm").exec("UPDATE_CONTENTS_FIELD", []);
    });
  });
</script>

```

// 프로젝트 경로 찾기

[1] 전자결재 문서 결재선 추가하기

휴가신청서

2

View "appLineAddForm"

문서번호: 자동채번(77) | 작성일: 2020-10-14 | 작성부서: 영업팀

내용: [내용 입력 영역]

1

결재, 참조선 추가하기

::부서선택::
영업팀
디자인
개발팀
경영관리팀

::사원선택::
김영상
김아름
한소희
손익민
윤예지

취소 | 결재선 | 결재요청

이름 선택 시 추가

결	기안	결	기안	영업팀	영업팀
제	김광훈	제	김광훈	김영상	김아름

1. 결재선 추가하기

```
<script type="text/javascript">
var windowObj;
function openChildWindow() {
    windowObj = window.open("../emplist_selectAdd.do", "",
        "width=500,height=400,left=700");
}
</script>
```

// 1.jsp에서 자식창
(appLineAddForm.jsp) 오픈

```
<script type="text/javascript">
function sendToParent() {
    var empno = $("#emp option:checked").val();
    $(opener.document).find("#app_line_add_input").append(
        '<input name="employee_no1" type="hidden" value="'+empno+'"/>');

    var deptname = $("#dept option:checked").text();
    var name = $("#emp option:checked").text();
    $(opener.document).find("#first").append("<td>" + deptname + "</td>");
    $(opener.document).find("#second").append("<td>" + name + "</td>");
    window.close();
}
</script>
```

// appLineAddForm.jsp(자식창) 에서
선택 시 부모창(1.jsp)에 결재선 테이블 자동 추가

2. 결재선DB에 다중 insert

```
@RequestMapping(value = "/insert_App_And_Domt.do", method = RequestMethod.POST)
public String insert_App_And_Domt(int[] employee_no1, DomtDto domtdto, DomtApprovalDto domtapp) {
    System.out.println(domtdto.getDocument_no());
    domtdto.setEmployee_no(employee_no1[0]);
    domtService.insert_app_domt(domtdto);
    domtService.insert_app_domt_app(domtapp, employee_no1);
    domtService.insert_app_domt_app_up(domtdto);
    return "redirect:/appdomt.do";
}
```

```
public void insert_app_domt_app(DomtApprovalDto domtapp, int[] employee_no1){
    int a = 0;
    for(int i=0; i < employee_no1.length; i++) {
        domtapp.setEmployee_no(employee_no1[i]);
        a += domtdao.insert_app_domt_app(domtapp);
    }
}
```

[1] 전자결재 문서 결재선 추가하기

The screenshot shows a web application interface for document approval. A modal window titled "결재,참조선 추가하기" (Approval, Reference Line Addition) is open, displaying two columns of department and employee selection options. The modal is numbered 1 and 2.

1. 컨트롤러 Ajax JSON파싱

```
@RequestMapping(value = "/appLineAddForm_select_one.do", method = RequestMethod.POST)
public void deptList(HttpServletResponse resp) {
    List<EmployeeDto> list = domtService.appDomt_domt_employee_add_listselect();
    Gson json = new Gson();
    resp.setContentType("text/html;charset=utf-8");
    PrintWriter out;
    try {
        out = resp.getWriter();
        out.print(json.toJson(list));
    } catch (IOException e) {
        e.printStackTrace();
    }
    // 데이터 값을 json객체에 담아 전달
}
```

2. Ajax로 데이터 요청한 값 가져오기

```
<h4>결재,참조선 추가하기</h4>
<select id="dept" onchange="selectDept()" size="10">
    <option value="":: 부서선택::</option>
</select>
<select id="emp" onchange="sendToParent()" size="10">
    <option value="":: 직원선택::</option>
</select>
<div id="empInfo"></div>

function selectDept() {
    var dept_no = $("#dept").val();
    if (dept_no == "") {
        $("#emp option").each(function() {
            $("#emp option:eq(1)").remove();
        });
        return;
    }

    var params = "dept_no=" + dept_no;
    $.ajax({
        type : "post",
        url : "appLineAddForm_select_two.do",
        data : params,
        dataType : "json"
    }).done(
        function(args) {
            $("#emp option").each(function() {
                $("#emp option:eq(0)").remove();
            });
            $("#emp").append("<option value='':: 직원선택::</option>");

            for (var idx = 0; idx < args.length; idx++) {
                $("#emp").append(
                    "<option value='"+args[idx].employee_no+"'>"
                    + args[idx].name + "</option>");
            }
        }).fail(function(e) {
            alert(e.responseText);
        });
    }
}
```

[2] 작성된 전자결재 문서 페이지 리스트

1

2 내가 작성한 기안서

문서번호	제목	작성일	타입
76	Test page	2020. 9. 29	1
75	지원요청	2020. 9. 29	1
74	납품확인서 요청	2020. 9. 29	1
73	휴가신청서 20200615	2020. 9. 29	1
72	휴가신청서 20200511	2020. 9. 29	1

1 2 3 4 5 >

전자결재작성

3 진행중인 기안서

문서번호	제목	작성일	타입
71	휴가신청서 20200930 - 20201001	2020. 9. 29	1

1

```
@RequestMapping(value = "/appdomt.do", method = RequestMethod.GET)
public String appDomt_domtList(@RequestParam(value = "p", defaultValue = "1") int pageNum,
    @RequestParam(value = "per", defaultValue = "5") int per,
    Model model, HttpSession request) {
    String employee_no = (String) request.getAttribute("mc_employee_no");

    domtListModel myDomt = domtService.list(pageNum, per, employee_no);
    model.addAttribute("domt_list3", myDomt);
    int number = myDomt.getCount() - (pageNum - 1) * per;
    model.addAttribute("number", number);

    domtListModel ingDomt = domtService.list2(pageNum, per, employee_no);
    model.addAttribute("domt_list4", ingDomt);
    int number2 = ingDomt.getCount() - (pageNum - 1) * per;
    model.addAttribute("number2", number2);
    return "domtList";
}
```

2. 내가 작성한 기안서 SQL

```
<select id="appDomt_domt_imok_list"
    resultType="mc.domt.model.DomtDto">
    select * from document d where (select count(*) from
    approval a where d.document_no = a.document_no and state = 1)
    = (select
    count(*) from approval a where d.document_no = a.document_no) and
    d.employee_no = #{employee_no}
</select>
```

// appLineAddForm.jsp(자식창) 에서
선택 시 부모창에 결재선 자동 추가

3. 진행중인 기안서 SQL

```
<select id="list2" resultType="mc.domt.model.DomtDto" parameterType="map">
    select * from document d, approval a
    where a.employee_no = #{employee_no}
    and a.document_no = d.document_no
    and
    (select state from approval a1
    where a1.document_no = a.document_no
    and a1.approval_no = a.approval_no - 1) = 1
    and state = 0 order by d.document_no DESC limit #{start},#{per} ;
</select>
```

// appLineAddForm.jsp(자식창) 에서
선택 시 부모창에 결재선 자동 추가

[3] 작성된 전자결재 문서 페이지 상세 조회

- 자신에게 올라온 결재 문서 (결재선)

결재	영업팀	영업팀	영업팀
	손익민 승인	김광훈 -	김영상 -

휴가신청서

문서번호	작성일	작성부서	작성자
자동채번(71)	2020-09-29	영업팀	손익민

휴가신청서 20200930 - 20201001

휴가신청서 20200930 - 20201001

사유 : 집을 이사합니다

사원번호 : 1번과 2번을 결재선에 올립니다

승인

반려

[목록으로](#)

결재	영업팀	영업팀	영업팀
	손익민 승인	김광훈 승인	김영상 -

결재	영업팀	영업팀	영업팀
	손익민 승인	김광훈 승인	김영상 반려

```
@RequestMapping(value = "/imokok.do", method = RequestMethod.GET)
public String imok(HttpSession request, DomtApprovalDto domt_app_no) {
    String employee_no = (String) request.getAttribute("mc_employee_no");
    domt_app_no.setEmployee_no(Integer.parseInt(employee_no));
    domtService.imok(domt_app_no);
    return "redirect:/appdomt.do";
}
// state 승인 update문 실행
```

```
@RequestMapping(value = "/imnono.do", method = RequestMethod.GET)
public String imno(HttpSession request, DomtApprovalDto domt_app_no) {
    String employee_no = (String) request.getAttribute("mc_employee_no");
    domt_app_no.setEmployee_no(Integer.parseInt(employee_no));
    domtService.imno(domt_app_no);
    return "redirect:/appdomt.do";
}
// state 반려 update 문 실행
```

```
<script>
$(document).ready(function() {
    var state = "";
    $('.no_han').each(function() {
        state = $(this).text();
        if (state == 0) {
            $(this).text('-');
        } else if (state == 1) {
            $(this).text('승인');
        } else if (state == 2) {
            $(this).text('반려');
        }
    });
});
</script>
```

// state 의 값을 가져와 스크립트로 상태 한글처리

-결재문서 결재선 테이블 예시

approval_no	document_no	employee_no	state	approval_date
105	68	1	1	2020-08-26
106	68	2	2	2020-08-26
107	68	3	0	2020-08-26
NULL	NULL	NULL	NULL	NULL

[4] 승인완료 페이지

승인완료

문서번호	제목	작성일	타입
72	휴가신청서 20200511	2020. 9. 29	1
73	휴가신청서 20200615	2020. 9. 29	1
74	납품확인서 요청	2020. 9. 29	1
75	지원요청	2020. 9. 29	1
76	Test_page	2020. 9. 29	1
79	20201014 ~ 20201018 휴가신청 요청	2020. 10. 14	1

[목록으로](#)

휴가신청서

문서번호	작성일	작성부서	작성자
자동재번(79)	2020-10-14	영업팀	김광훈

20201014 ~ 20201018 휴가신청 요청

20201014 ~ 20201018 휴가신청 요청 합니다

[목록으로](#)

// 결재선 테이블의 모든 승인 상태가 되면
승인완료 리스트업

결	영업팀	영업팀	영업팀
재	김광훈	김영상	김아름
	승인	승인	승인

View "domtView.do"

```
@RequestMapping(value = "/domtView.do", method = RequestMethod.GET)
public String domtViewPage(Model model2, int document_no) {
    Domt_P_ApprovalDto dpadto1 = domtService.domtViewPage_domt(document_no);
    model2.addAttribute("document_no", dpadto1.getDocument_no());
    model2.addAttribute("employee_no", dpadto1.getEmployee_no());
    model2.addAttribute("subject", dpadto1.getSubject());
    model2.addAttribute("contents", dpadto1.getContents());
    model2.addAttribute("reg_date", dpadto1.getReg_date());
    model2.addAttribute("type", dpadto1.getType());
    model2.addAttribute("dept_name", dpadto1.getDept_name());
    model2.addAttribute("name", dpadto1.getName());
    List<Domt_P_ApprovalDto> list = domtService.domtViewPage_appline(document_no);
    model2.addAttribute("domt_appline", list);
    return "/domtView";
}
```

```
<select id="domtViewPage_domt"
    resultType="mc.domt.model.Domt_P_ApprovalDto" parameterType="int">
    SELECT document_no,employee_no,subject,contents,reg_date,dept_name,name FROM
    document natural join employee natural join department where
    document_no = #{document_no};
</select>

<select id="domtViewPage_appline"
    resultType="mc.domt.model.Domt_P_ApprovalDto" parameterType="int">
    select
    employee_no,name,dept_no,dept_name,state from approval natural join
    employee natural join department where document_no =#{document_no}
    group by employee_no;
</select>
```

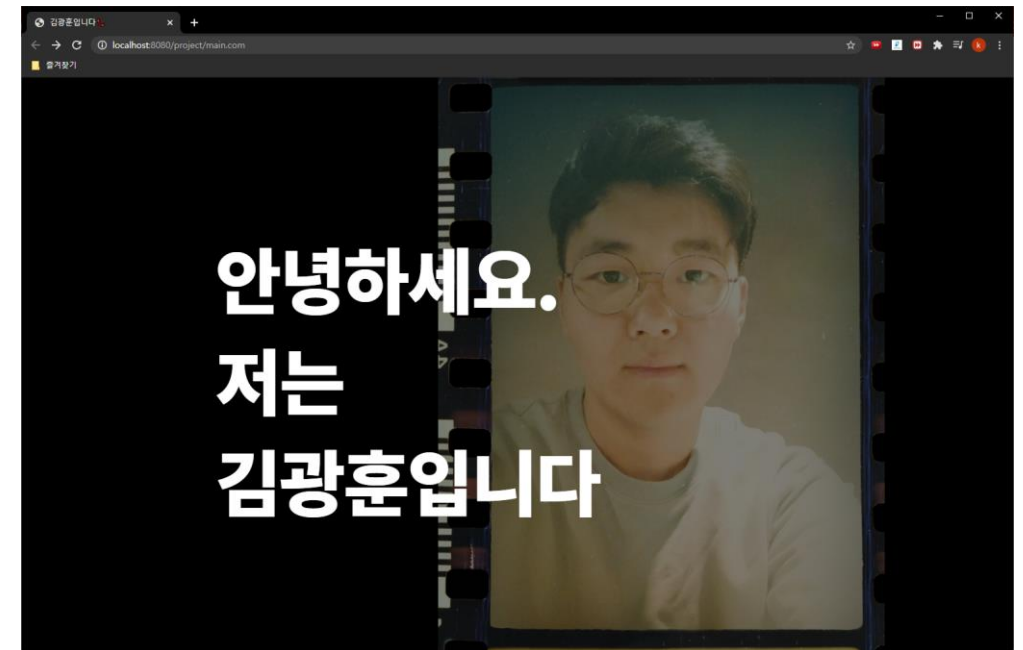

public void myPart()

{ **개인소개 사이트** }

개인프로젝트

String 개발 배경 및 목적;

웹으로 포트폴리오를 보여주기 위한 사이트
직접작성한 코드와 이미지를 업로드 하기 위한 웹
애플리케이션



String 개발 환경 및 서비스 개요;

FrontEnd



JavaScript



jQuery



HTML5



CSS3



Ajax

WAS



Apache Tomcat
9.0

Database



MySQL 8.0

BackEnd



Spring 4.3



MyBatis



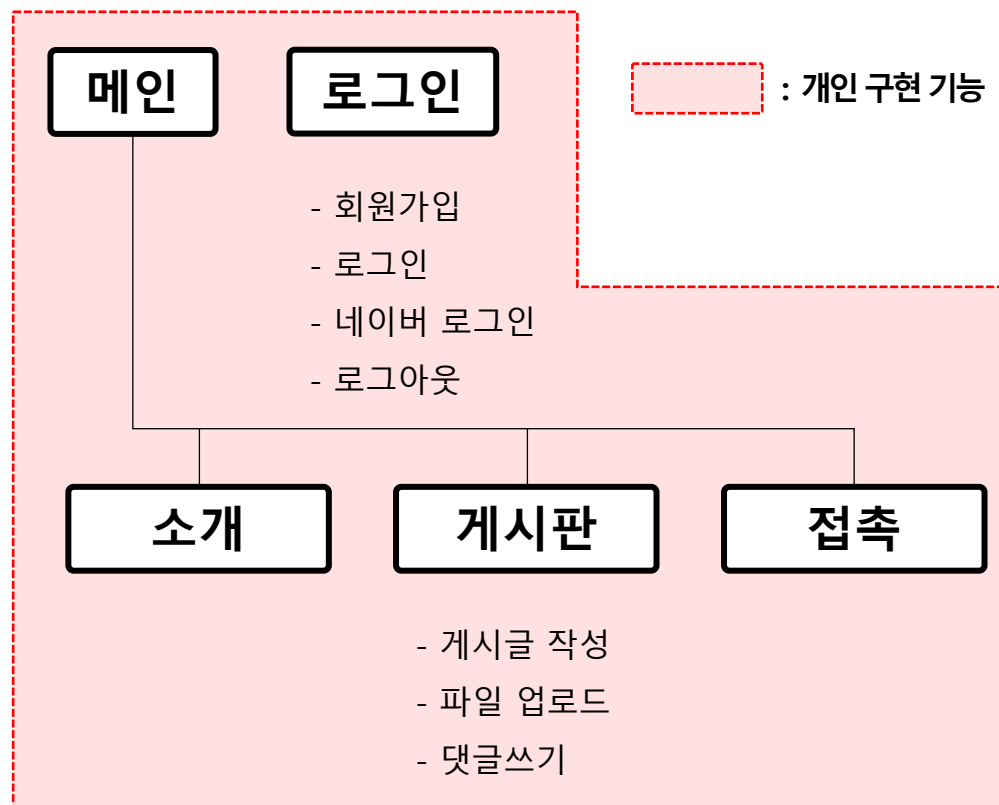
Java 1.8



JSP

String 서비스 개요;

기능리스트 / 프로젝트 ERD



khjoin						
로그인정보						
PK	AI	FK	UQNull	Logical Name	Name	Type
		+	✓	메일	mail	VARCHAR(50)
		+	✓	비번	pass	VARCHAR(50)
		+	✓	이름	name	VARCHAR(50)
		+		핸드폰	phone	INT

kh_board						
게시판						
PK	AI	FK	UQNull	Logical Name	Name	Type
✓	✓	✓		게시판번호	bno	INT
		+	✓	제목	title	VARCHAR(100)
		+	✓	내용	content	VARCHAR(100)
		+	✓	작성자	writer	VARCHAR(100)
		+	✓	작성일	regDate	DATETIME
		+	✓	페이지뷰	viewCnt	INT

kh_file						
게시판파일						
PK	AI	FK	UQNull	Logical Name	Name	Type
✓	✓	+	✓	파일번호	fileNo	INT
		+	✓	게시판번호	bno	INT
		+	✓	파일원본명	fileOr	VARCHAR(100)
		+	✓	파일변경명	fileCh	VARCHAR(100)
		+	✓	파일주소	fileSi	INT

kh_board_comment						
게시판 댓글						
PK	AI	FK	UQNull	Logical Name	Name	Type
		+	✓	게시판번호	bno	INT
✓	✓	+	✓	댓글번호	commentNo	INT
		+	✓	댓글내용	commentText	VARCHAR(5000)
		+	✓	댓글아이디	commentId	VARCHAR(100)
		+	✓	댓글비밀번호	commentPw	VARCHAR(100)
		+	✓	댓글날짜	commentDa	DATETIME

[0] 페이지 이동

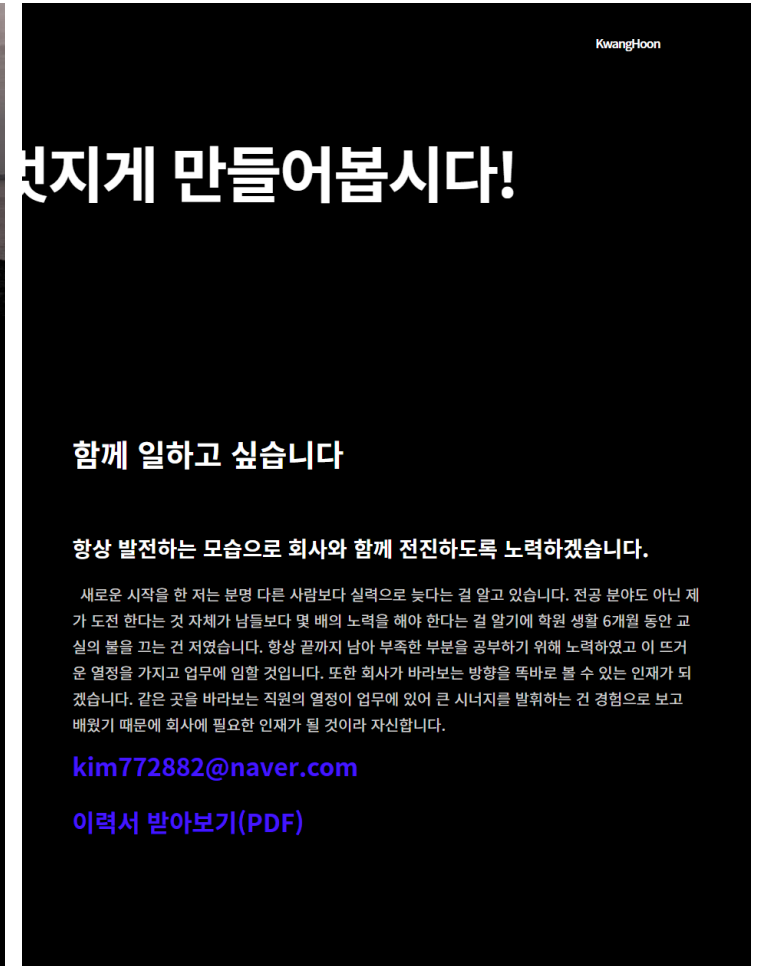
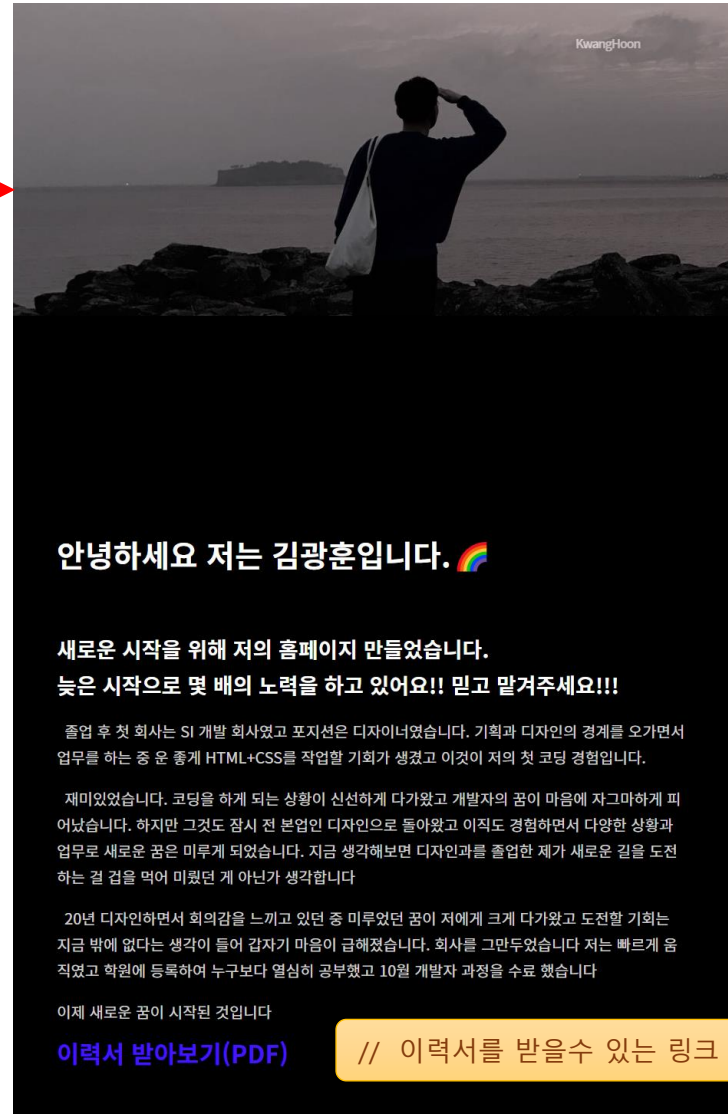
View "about.com"

안녕하세요 화면

김광훈입니다 화면

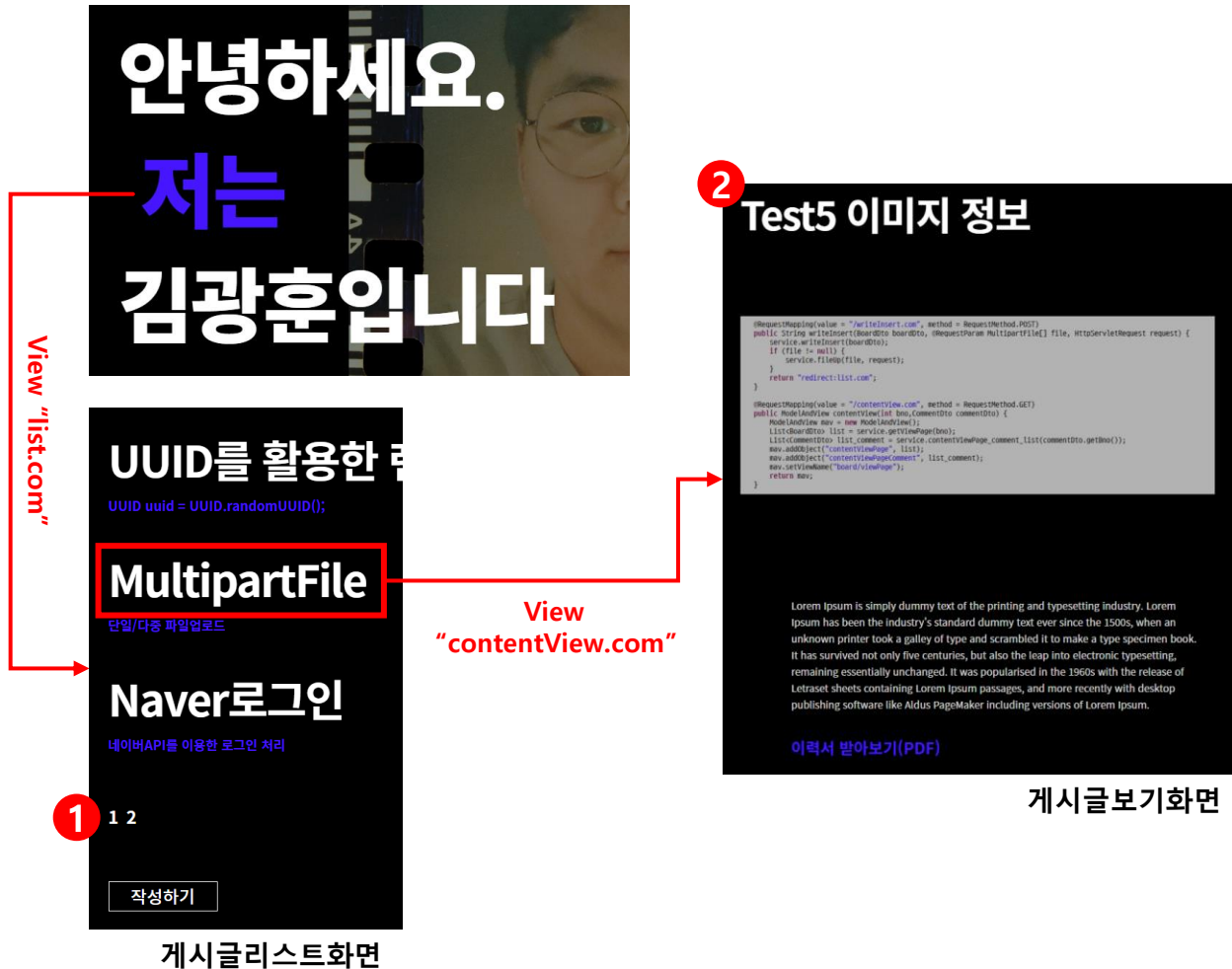


View "contact.com"



// 이력서를 받을수 있는 링크

[1] 작성된 게시물 보기



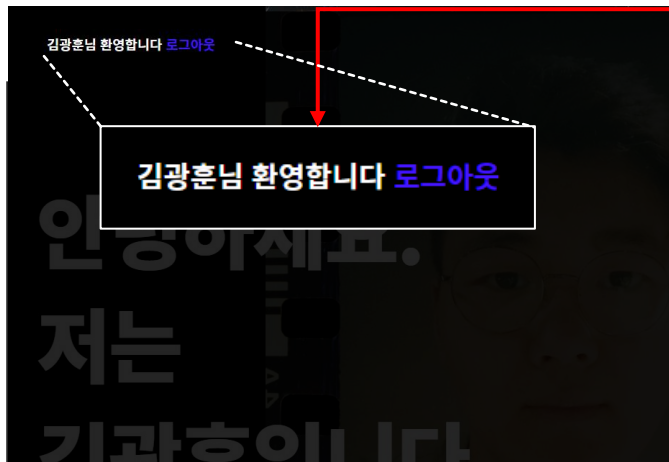
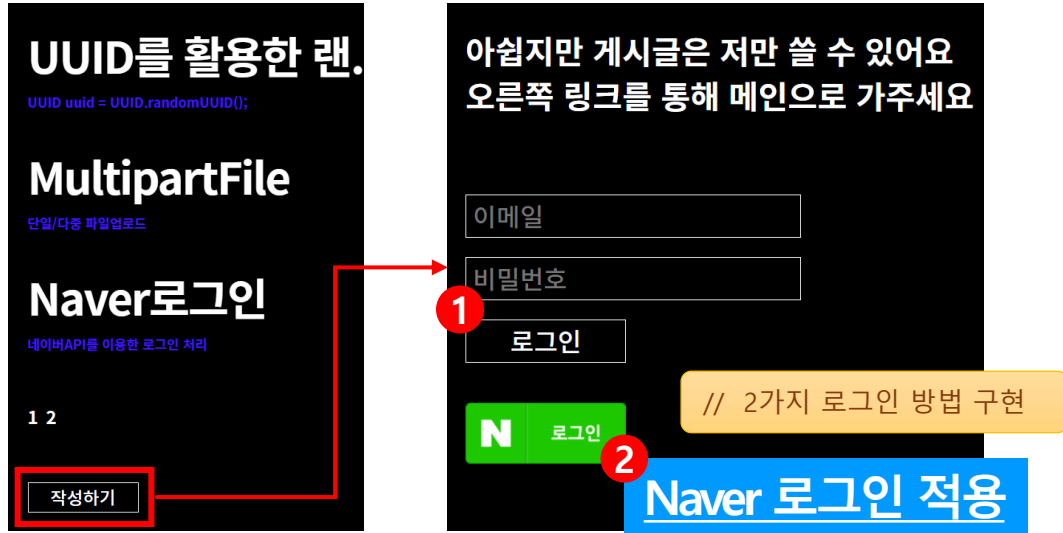
1. 게시물 페이징 처리

```
@RequestMapping(value = "/list.com", method = RequestMethod.GET)
public String getList2(
    @RequestParam(value = "p", defaultValue = "1") int pageNum,
    @RequestParam(value = "per", defaultValue = "10") int per,
    Model m) {
    BoardListModel list = service.list(pageNum, per);
    m.addAttribute("list", list);
    int number = list.getCount() - (pageNum - 1) * per;
    m.addAttribute("number", number);
    return "board/list";
}
```

2. 게시물 보기

```
@RequestMapping(value = "/contentView.com", method = RequestMethod.GET)
public ModelAndView contentView(int bno, CommentDto commentDto) {
    ModelAndView mav = new ModelAndView();
    List<BoardDto> list = service.getViewPage(bno);
    List<CommentDto> list_comment = service.contentViewPage_comment_list(commentDto.getBno());
    mav.addObject("contentViewPage", list);
    mav.addObject("contentViewPageComment", list_comment);
    mav.setViewName("board/viewPage");
    return mav;
}
```

[2] 회원가입/로그인/네이버아이디로그인



1. 회원가입을 통한 로그인

```
@RequestMapping(value = "/mb_login.com", method = RequestMethod.POST)
public ModelAndView mblogin(String mail, String pass, HttpSession session) {
    ModelAndView mav = new ModelAndView("main");
    Map<String, String> loginMap = service.mblogin(mail);
    if (loginMap == null) {
        mav.addObject("check", 0);
        mav.setViewName("login");
    } else if (!loginMap.get("pass").equals(pass)) {
        mav.addObject("check", 1);
        mav.setViewName("login");
    } else {
        session.setAttribute("mail", mail);
        mav.setViewName("main");
    }
    return mav;
}
```

2. Naver아이디를 통한 로그인

```
@RequestMapping(value = "/login.com", method = { RequestMethod.GET, RequestMethod.POST })
public String login(Model model, HttpSession session) {
    String naverAuthUrl = naverLoginBO.getAuthorizationUrl(session);
    model.addAttribute("url", naverAuthUrl);
    return "login";
}

@RequestMapping(value = "/callback.com", method = { RequestMethod.GET, RequestMethod.POST })
public String callback(Model model, @RequestParam String code, @RequestParam String state, HttpSession session)
    throws IOException, ParseException {
    OAuth2AccessToken oauthToken;
    oauthToken = naverLoginBO.getAccessToken(session, code, state);
    apiResult = naverLoginBO.getUserProfile(oauthToken);
    JSONParser parser = new JSONParser();
    Object obj = parser.parse(apiResult);
    JSONObject jsonObj = (JSONObject) obj;
    JSONObject response_obj = (JSONObject) jsonObj.get("response");
    String nickname = (String) response_obj.get("nickname");
    String id = (String) response_obj.get("id");
    String name = (String) response_obj.get("name");
    System.out.println(nickname);
    session.setAttribute("sessionnickname", nickname);
    session.setAttribute("sessionId", id);
    session.setAttribute("sessionname", name);
    model.addAttribute("result", apiResult);
    return "main";
}
```

// 네이버에 등록된 이름을 가져옴

[3] 게시물 작성하기

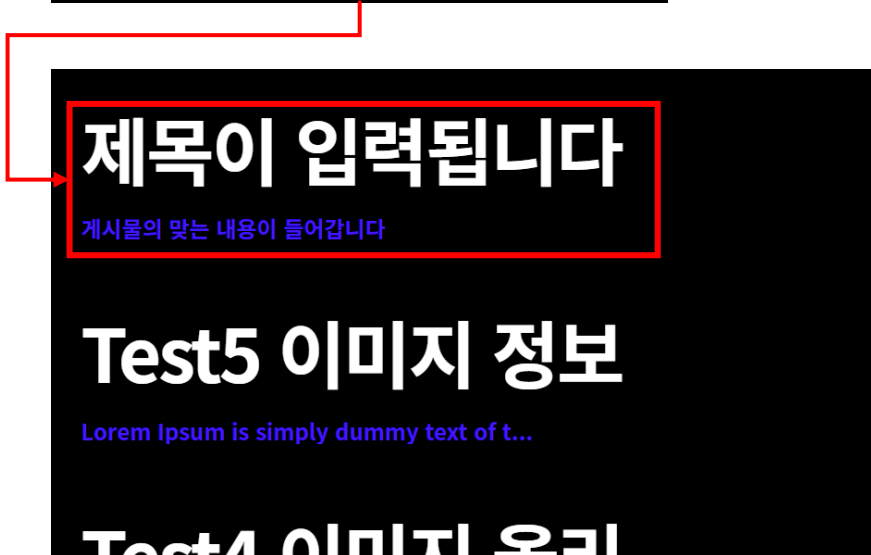
1 kim772882@naver.com //작성자 자동입력

제목이 입력됩니다

게시물의 맞는 내용이 들어갑니다

2 파일 선택 Imgtest.PNG

업로드



2. 파일업로드

```
@RequestMapping(value = "/writeInsert.com", method = RequestMethod.POST)
public String writeInsert(BoardDto boardDto, @RequestParam MultipartFile[] file,
    HttpServletRequest request) {
    service.writeInsert(boardDto);
    if (file != null) {
        service.fileUp(file, request);
    }
    return "redirect:list.com";
}

-> public void fileUp(MultipartFile[] file, HttpServletRequest request) {
    for (MultipartFile fileList : file) {

        String or = fileList.getOriginalFilename();
        UUID uuid = UUID.randomUUID();
        String ch = uuid + "-" + fileList.getOriginalFilename();
        int si = (int) fileList.getSize();

        String SAVE_PATH = request.getSession().getServletContext().getRealPath("/resources/");
        System.out.println(SAVE_PATH);
        String saveFileName = SAVE_PATH + ch;

        try {
            fileList.transferTo(new File(saveFileName));
        } catch (IllegalStateException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        BoardFileDto boardFileDto = new BoardFileDto();
        boardFileDto.setBno(maxBno());
        boardFileDto.setFileOr(or);
        boardFileDto.setFileCh(ch);
        boardFileDto.setFileSi(si);

        bdDao.fileInforInsert(boardFileDto);
    }
}
```

//Null값 체크 후 파일 업로드 실행

//같은 이름 저장도 가능 하기 위해
파일명을 랜덤+오리지널 으로 변경
하여 저장

[4] 댓글작성하기

Letrasheet sheets containing Lorem Ipsum passages, and more recent publishing software like Aldus PageMaker including versions of Lorem Ipsum

Queen
Pressure pushing down on me 압박감이 나를 짓누르고

TestName
Lorem Ipsum is simply dummy text of the printing

이력서 받아보기(PDF)

1 CodingKing!

잘보고 갑니다 좀 더 열심히 해봅시다 화이팅!!!!

등록

publishing software like Aldus PageMaker including versions of Lorem Ipsum

Queen
Pressure pushing down on me 압박감이 나를 짓누르고

TestName
Lorem Ipsum is simply dummy text of the printing

CodingKing!
잘보고 갑니다 좀 더 열심히 해봅시다 화이팅!!!!

이력서 받아보기(PDF)

닉네임

내용을 작성해주세요

등록

1. Ajax로 댓글작성

```
<script type="text/javascript">
function commentAdd() {
    if (confirm('댓글을 작성하시겠습니까?')) {
        var CommentDto = $('#commentFrom').serialize();
        $.ajax({
            url : 'contentViewPage_comment.com',
            type : 'POST',
            data : CommentDto,
            dataType : "json"
        }).done(
            function(args) {
                var commentText = args.commentText
                var commentId = args.commentId
                var commentDa = args.commentDa

                $("#commentTable").append(
                    "<div style='border-left: 3px solid #585858; padding-left: 20px;'"
                    + "<strong style='color:#643dfd;'>" + commentId
                    + "</strong>&nbsp;&nbsp;&nbsp;"
                    + "<br>" + commentText + "<br></div>");
            });
        } else {
            alert("취소되었습니다");
        }
    }
}
</script>
```

→ @RequestMapping(value = "/contentViewPage_comment.com", method = RequestMethod.POST)
@ResponseBody
public String contentViewPage_comment_insert_select(CommentDto commentDto) {
 service.contentViewPage_comment_insert(commentDto);
 int commentNo = commentDto.getCommentNo();
 CommentDto list_comment = service.contentViewPage_comment_add(commentNo);
 Gson json = new Gson();
 return json.toJson(list_comment);
}

//입력된 값 즉시 반영

{ **Thank You!** }

Return;

김광훈 포트폴리오 010. 5136. 3964

Mail. kim772882@naver.com