

프로젝트 기술서

프로젝트 명	Triple A Achieve Anytime, Anywhere		
포트폴리오 사이트주소	https://github.com/kimchifokum/finalproject	개발기간	21.11.08~12.06
주요기술	1. Spring Boot, Mybatis 사용 2. Java, MVC2 활용 3. Jquery 또는 자바스크립트를 활용한 유효성 검사 4. Ajax 활용 DB접근 5. Oracle 11g 활용 6. HTML,CSS,JS를 활용한 View구현	투입인원	5(본인 포함)
제작 목적	>자기주도 학업 사이트 코로나 시국으로 인해 학교나 학원 활동이 비 대면으로 활동하기 때문에 학업에 큰 어려움을 겪는 것을 알게 되었고 온라인을 통해 로드 맵, 크루 활동 등을 통하여 더욱더 학생들이 학업에 재미와 흥미를 주기 위해 구현하였습니다. 1. 코로나 시국이라 온라인으로 만 수업을 들어야하기 때문에 정보 부족 2. 로드 맵을 통해 나만의 스케줄러 생성으로 시간관리 용이 3. 크루 활동을 통해 타인과 교류가 원활 하고 하나의 목표의식을 가질 수 있다.		
본인의 역할			

구성 및 흐름



로그인 및 회원관리

회원, 관리자(Admin) 각각 접근 가능하도록 별도의 DB Table로 구현
(또한 접속일 역시 DB에 기록)

Session을 통해 로그인 했을 시와 안했을 시 메뉴가 변하게 지정하였다.

로그인

당신의 꿈을 이루여주는 사이트

아이디

admin

비밀번호

비밀번호

☒ 아이디 저장

로그인

아이디와 비밀번호를 잊어버리셨나요?

아이디 찾기 비밀번호 찾기

이미 가입한 회원이신가요?

회원가입

로그인

회원가입

민지민지 님

마이페이지

로그아웃

```
// 로그인 시키기(ajax)
@ResponseBody
@PostMapping("/login")
public int login(UserInfoVo userInfoVo, Model model, HttpServletRequest request) {
    System.out.println("userInfoVo 아이디: " + userInfoVo.getU_id());
    System.out.println("userInfoVo 비밀번호: " + userInfoVo.getU_pw());
    HttpSession session = request.getSession();
    UserInfoVo userVo = userInfoService.userLogin(userInfoVo);
    int loginCheck = 0;
    if (userVo != null) {
        loginCheck = 1;

        // 최종접속일
        int result = userInfoService.finalDate(userInfoVo);

        session.setAttribute("session_u_no", userVo.getU_no());
        session.setAttribute("session_u_detail_no", userVo.getU_detail_no());
        session.setAttribute("session_u_id", userVo.getU_id());
        session.setAttribute("session_u_name", userVo.getU_name());
        session.setAttribute("session_u_nickName", userVo.getU_nickName());
        session.setAttribute("session_u_profile", userVo.getU_profile());
        session.setAttribute("session_u_email", userVo.getU_email());
        session.setAttribute("session_u_tel", userVo.getU_tel());
        session.setAttribute("session_u_signDate", userVo.getU_signDate());
        session.setAttribute("session_u_modDate", userVo.getU_modDate());
        session.setAttribute("session_u_finalDate", userVo.getU_finalDate());
        session.setAttribute("session_u_intro", userVo.getU_intro());

    }
    return loginCheck;
}
```

기본적으로 Ajax를 통해 구현하였다.

```
$.ajax({
    url: "./login",
    type: "post",
    data: {
        "u_id": $("#u_id").val(),
        "u_pw": $("#u_pw").val()
    },
    success: function(data){
        console.log(data);
        if(data==1){
            alert("로그인이 성공적으로 완료되었습니다.");
            location.href = "../main";
        }else{
            alert("ID 또는 PASSWORD가 틀렸습니다. 다시 입력하시길 바랍니다.");
            location.href = "./login";
        }
    },
    error: function(){

    }

});
});
});
```

아이디 저장 기능(쿠키를 통해 체크박스 체크 시 아이디가 저장된다.)

```

$(document).ready(function(){
    // 저장된 쿠키값을 가져와서 ID 칸에 넣어준다. 없으면 공백으로 들어감.
    var userInputId = getCookie("userInputId");
    $("#input[id='u_id']").val(userInputId);

    if($("#input[id='u_id']").val() != ""){
        // 그 전에 ID를 저장해서 처음 페이지 로딩 시, 입력 칸에 저장된 ID가 표시된 상태라면,
        $("#saveBtn").attr("checked", true); // ID 저장하기를 체크 상태로 두기.
    }

    $("#saveBtn").change(function(){ // 체크박스에 변화가 있다면,
        if($("#saveBtn").is(":checked")){ // ID 저장하기 체크했을 때,
            var userInputId = $("#input[id='u_id']").val();
            setCookie("userInputId", userInputId, 7); // 7일 동안 쿠키 보관
        }else{ // ID 저장하기 체크 해제 시,
            deleteCookie("userInputId");
        }
    });

    // ID 저장하기를 체크한 상태에서 ID를 입력하는 경우, 이럴 때도 쿠키 저장.
    $("#input[id='u_id']").keyup(function(){ // ID 입력 칸에 ID를 입력할 때,
        if($("#saveBtn").is(":checked")){ // ID 저장하기를 체크한 상태라면,
            var userInputId = $("#input[id='u_id']").val();
            setCookie("userInputId", userInputId, 7); // 7일 동안 쿠키 보관
        }
    });

    //쿠키를 위한 함수
    function setCookie(cookieName, value, exdays){
        var exdate = new Date();
        exdate.setDate(exdate.getDate() + exdays);
        var cookieValue = escape(value) + ((exdays==null)?"":";expires="+exdate.toGMTString());
        document.cookie = cookieName + "=" + cookieValue;
    }

    function deleteCookie(cookieName){
        var expireDate = new Date();
        expireDate.setDate(expireDate.getDate() - 1);
        document.cookie = cookieName + "=" + "; expires=" + expireDate.toGMTString();
    }
}

```

체크박스가 체크된 상태이면 session.invaldate 이후에도 이전에 입력 되었던 아이디가 input에 담긴다.
체크 해제시에는 담기지 않는다.

비밀번호

☒ 아이디 저장

로그인

아이쿠!! 아이디 또는 비밀번호가 틀렸습니다.

메인 페이지

게시판 정보, 크루 정보 및 카테고리 정보를 화면에 구현

자유 게시판

더보기 +

정보 게시판

더보기 +

[입시] 안녕하세요. 이번에 안양 대에 진학하려는 19살 학생입니다.

[대외활동] 탄소중립 대국민 아이디어 공모전

[고민] 님들 내가 고민이 있어서 그런데 한번 오셔서 고민 상담 부탁드립니다

[공부법] 사회문화 공부

[일상] 로고 ㅋㅋㅋㅋ

[공부법] 수능 생활과 윤리

[일상] 팔 vs 슈크림

[공부법] 내가 생각하는 영어공부법

크루

더보기 +

 <p>2학기 중간고사 대비 영어공부</p> <p>2021-12-19 ~ 2021-12-24</p>	 <p>재수 준비반</p> <p>2021-12-15 ~ 2022-11-01</p>	 <p>디자이너 크루</p> <p>2021-12-23 ~ 2021-12-31</p>	 <p>2021-12-03 오늘 발표 도와 주실분??</p> <p>2021-12-15 ~ 2021-12-16</p>
---	---	---	--

```
<c:forEach items="${ map.freeBoard }" var="board">
  <c:choose>
    <c:when test="${ board.b_cate_no == 1 }"><c:set var="cate" value="일상" /></c:when>
    <c:when test="${ board.b_cate_no == 2 }"><c:set var="cate" value="고민" /></c:when>
    <c:when test="${ board.b_cate_no == 3 }"><c:set var="cate" value="입시" /></c:when>
    <c:when test="${ board.b_cate_no == 4 }"><c:set var="cate" value="진로" /></c:when>
  </c:choose>
  <div class="left-under">
    <span class="title">[${ cate }]<strong class="content"><a href="/board/view/1?b_post_no=${ board.b_post_no }">${ board.b_post_no }</a>
  </div>
</c:forEach>

</div>
<div class="bottomroad-right">

  <div class="bottomroad-right-top">
    <div class="bottomroad-right-left">
      <h1>정보 게시판</h1>
    </div>
    <div class="bottomroad-right-right">
      <a href="/board/list/3">더보기 + </a>
    </div>
  </div>

  <c:forEach items="${ map.infoBoard }" var="board">
    <c:choose>
      <c:when test="${ board.b_cate_no == 7 }"><c:set var="cate" value="대외활동" /></c:when>
      <c:when test="${ board.b_cate_no == 11 }"><c:set var="cate" value="공부법" /></c:when>
    </c:choose>
    <div class="left-under">
      <span class="title">[${ cate }]<strong class="content"><a href="/board/view/1?b_post_no=${ board.b_post_no }">${ board.b_post_no }</a>
    </div>
  </c:forEach>
```

회원가입

회원이 가진 정보들을 관리
(회원가입 시 데이터를 Insert하며, 아이디나 비밀번호 분실 시 Select 하여 검색)

※ 정보 전달 과정



회원가입

<div>View</div>	<div><일부생략></div> <div><pre><div class="sign-up-htm"> <div class="textarea"> <div class="group"> <label>이름<small id="nameS"></small></label> <div> <input type="text" class="u_name" id="u_name" name="u_name" p: </div> </div> <div class="group"> <label>아이디<button type="button" class="sameBtn" onclick="sam <div> <input type="text" class="u_id" id="u_id" name="u_id" pla </div> </div> <div class="group"> <label>비밀번호<small id="pwS"></small></label> <div> <input type="password" class="u_pw" id="u_pw" name="u_pw" </div> </div> <div class="group"> <label>비밀번호확인 <small class="inputPw" id="inputPw"></small> </label> </div> </div> </div></pre></div> <div><일부생략></div> <div>정규표현식 적용, 아이디 및 닉네임 중복 체크 기능 구현(ajax구현)</div> <div><pre>\$.ajax({ url:"./idChk", type:"post", dataType:"json", data : { "u_id": \$("#u_id").val() }, success: function(data){ if(data==1){ alert("중복된 아이디입니다."); }else{ alert("사용가능한 아이디입니다."); } }, error:function(){ } });</pre></div>
-----------------	---

Controller

<일부 생략>

```
//회원가입 (이메일)폼
@RequestMapping("/signUpEmail")
public String signUpEmail() {
    return "user/signUpEmail";
}

//회원가입폼
@RequestMapping("/signUp")
public String signUp() {
    return "user/signUp";
}

//회원가입
@RequestMapping("/sign")
public String signUp(UserInfoVo userInfoVo,
    @RequestParam MultipartFile file,
    Model model) {
    String newFileName="";
    if( file.getSize()!=0) {
        String orginFileName = file.getOriginalFilename();
        long time = System.currentTimeMillis();
        newFileName = String.format("%d_%s", time,orginFileName);
        String fileSaveUrl="C:/team1FileSave/";
        File f = new File(fileSaveUrl+newFileName);
        //파일 업로드
        try {
            file.transferTo(f);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    //파일이름
    userInfoVo.setU_profile(newFileName);
    int result = userService.userWrite(userInfoVo);
    model.addAttribute("userInfoVo",userInfoVo);
    System.out.println("result:" + result);
    return "user/signUpCompl";
}
```

Service

<일부생략>

```
//회원로그인
public UserInfoVo userLogin(UserInfoVo userInfoVo) {
    UserInfoVo userVo = userInfoMapper.userLogin(userInfoVo);
    return userVo;
}

//회원가입
public int userWrite(UserInfoVo userInfoVo) {
    int result = userInfoMapper.userWirte(userInfoVo);
    return result;
}

//회원가입완료
public UserInfoVo userInfoList(UserInfoVo userInfoVo) {
    UserInfoVo userVo = userInfoMapper.selectUser(userInfoVo);
    return userVo ;
}

//이메일 인증
public String sendEmail(String u_email) throws Exception {
    //임시비밀번호 생성
    String pwCode="";
    pwCode = getCreateKey();
    System.out.println("pwCode:"+pwCode);
    //이메일전송 관련 2가지 형태 1. text만 가능
    //SimpleMailMessage message2 = new SimpleMailMessage();
    // html코드 전송가능
    MimeMessage mail = javaMailSender.createMimeMessage();
    MimeMessageHelper message = new MimeMessageHelper(mail,true,"utf-8");

    //보내는 주소
    message.setTo(u_email);
    //보내는 사람주소
    message.setFrom("sungho969891@gmail.com");
    //메일 제목
    message.setSubject("회원가입 이메일인증 확인번호 안내");
    //메일 전송 내용
    String emailText = mailContent(u_email, pwCode);
}
```

Dao

```

@Data
@NoArgsConstructor
@AllArgsConstructor
public class UserInfoVo {

    private int u_no;
    private int u_detail_no;
    private String u_name;
    private String u_id;
    private String u_pw;
    private String u_tel;
    private String u_email;
    private String u_nickName;
    private String u_profile;
    private Timestamp u_signDate;
    private Timestamp u_finalDate;
    private Timestamp u_modDate;
    private String u_intro;

}

```

Mapper
Xml

```

<!-- 생략 -->
<!-- 마이페이지 -->
<select id="mypage" resultType="com.site.team1.vo.UserInfoVo">
    select * from user_info where u_id=#{u_id}
</select>

<!-- 로그인확인 -->
<select id="userLogin" resultType="com.site.team1.vo.UserInfoVo">
    select * from user_info where u_id=#{u_id} and u_pw=#{u_pw} and u_detail_no <![CDATA[<>]]> 9
</select>

<!-- 회원가입 -->
<insert id="userWirte">
    insert into user_info values (
        user_info_seq.nextval,1,#{u_name},#{u_id},
        #{u_pw},#{u_tel},#{u_email},#{u_nickName},#{u_profile},sysdate, sysdate,sysdate,#{u_intro})
</insert>

<!-- 회원가입 완료 -->
<select id="selectUser" resultType="com.site.team1.vo.UserInfoVo">
    select u_id,u_name,u_nickname,u_email from user_info where u_id=#{u_id}
</select>

<!-- 아이디 찾기 -->
<select id="selectUserId" resultType="com.site.team1.vo.UserInfoVo">
    select * from user_info where u_name=#{u_name} and u_email=#{u_email} and u_tel=#{u_tel}
</select>

<!-- 비밀번호 찾기 -->
<select id="pwFindCom" resultType="com.site.team1.vo.UserInfoVo">
    select * from user_info where u_id=#{u_id}
</select>

<!-- 비밀번호 변경 -->
<update id="pwCompl">
    update user_info set u_pw=#{u_pw} where u_id=#{u_id}
</update>

<!-- 아이디 중복 체크 -->

```

회원정보 찾기 (아이디 및 비밀번호)

아이디 찾기

```
// 아이디 찾기 폼
@GetMapping("/idFind")
public String idFind() {
    return "user/idFind";
}

//아이디 찾기
@PostMapping("/idC")
public String idCom(UserInfoVo userInfoVo, Model model) {
    System.out.println(userInfoVo.getU_email());

    UserInfoVo userVo = userService.selectId(userInfoVo);
    model.addAttribute("userVo", userVo);

    return "user/idCom";
}

//아이디 중복 체크
@ResponseBody
@RequestMapping("/idChk")
public int idChk(UserInfoVo userInfoVo) {
    int result = userService.idChk(userInfoVo);
    System.out.println("아이디 중복확인"+result);
    return result;
}

<!-- 아이디 찾기 -->
<select id="selectUserId" resultType="com.site.team1.vo.UserInfoVo">
    select * from user_info where u_name=#{u_name} and u_email=#{u_email} and u_tel=#{u_tel}
</select>
```

비밀번호 찾기

비밀번호는 잊어버렸을 경우 보안을 위해 새로 발급하도록 되어 있다.(따라서 select가 아닌 update)

```
<!-- 비밀번호 변경 -->
<update id="pwCompl">
    update user_info set u_pw=#{u_pw} where u_id=#{u_id}
</update>

//비밀번호찾기폼
@RequestMapping("/pwFind")
public String pwFind() {
    return "user/pwFind";
}

//비밀번호를 찾기
@PostMapping("/pwC")
public String pwCom(@RequestParam String u_id, Model model) {
    System.out.println("넣어온 값1 : "+u_id);
    UserInfoVo userInfoVo= userService.pwFindCom(u_id);
    System.out.println(userInfoVo.getU_id());
    model.addAttribute("userInfoVo", userInfoVo);

    return "user/pwCom";
}

//비밀번호변경
@RequestMapping("/pwCompl")
public String pwCom(UserInfoVo userInfoVo, Model model) {
    int result = userService.pwCompl(userInfoVo);
    model.addAttribute("userInfoVo", userInfoVo);
    return "user/pwCompl";
}
```


마이 페이지

메인 화면(내 정보, 크루, 로드 맵, 게시판)

로그인 시 Session을 통해 이용 가능한 서비스이다.

회원정보 일부를 화면에 보여준다.


마이페이지 | 홈

민지민지님,
환영합니다.

내 정보 관리

크루를 통해
구성원의 일부가 되어보세요.

크루 관리



내 프로필

회원명
이민지

닉네임
민지민지

이메일
minji1234@naver.com

로드맵 관리

로드맵 관리


이야기를 나누세요
50만명의 커뮤니티

게시물 관리

마이 페이지 내 정보

내 프로필 변경, 회원정보 수정, 회원탈퇴 기능이 있다.

이민지 님의 페이지



민지민지


프로필변경


회원명 이민지


아이디 minji1234

휴대폰 01052124026

이메일 minji1234@naver.com

 이민지님의 회원 가입일
2021-12-01 19:27
우리가 만남 첫 기념일

 이민지님의 최종 접속일
2021-12-07 18:17
반가워 오늘도 파이팅!!

 이민지님의 회원 수정일
2021-12-01 19:27
너 변했어...그래도 너가 좋아

소개글

안녕하세요. 전 민지라고 해요. 친하게 지내요^^ 그리고 저 님친있으니까, 짝점거리지 마세요^^

회원정보수정

회원영구탈퇴

대략적인 마이 페이지 코드

(session에 미리 저장 되어 있는 회원정보)

```
//마이페이지 홈
@GetMapping("/mypage")
public String mypage(Model model,HttpServletRequest request) {
    HttpSession session = request.getSession();
    UserInfoVo userInfoVo = new UserInfoVo();
    userInfoVo.setU_id((String)session.getAttribute("session_u_id"));
    userInfoVo.setU_profile((String)session.getAttribute("session_u_profile"));
    userInfoVo.setU_name((String)session.getAttribute("session_u_name"));
    userInfoVo.setU_nickname((String)session.getAttribute("session_u_nickname"));
    userInfoVo.setU_email((String)session.getAttribute("session_u_email"));
    userInfoVo.setU_tel((String)session.getAttribute("session_u_tel"));
    userInfoVo.setU_signDate((Timestamp)session.getAttribute("session_u_signDate"));
    userInfoVo.setU_modDate((Timestamp)session.getAttribute("session_u_modDate"));
    userInfoVo.setU_finalDate((Timestamp)session.getAttribute("session_u_finalDate"));
    userInfoVo.setU_intro((String)session.getAttribute("session_u_intro"));
    System.out.println("userInfoVo id값:"+userInfoVo.getU_id());
    return "mypage/mypage";
}

//마이페이지
@RequestMapping("/my_info")
public String my_info(Model model,HttpServletRequest request) {
    HttpSession session = request.getSession();
    UserInfoVo userInfoVo = new UserInfoVo();
    userInfoVo.setU_id((String)session.getAttribute("session_u_id"));
    userInfoVo.setU_profile((String)session.getAttribute("session_u_profile"));
    userInfoVo.setU_name((String)session.getAttribute("session_u_name"));
    userInfoVo.setU_nickname((String)session.getAttribute("session_u_nickname"));
    userInfoVo.setU_email((String)session.getAttribute("session_u_email"));
    userInfoVo.setU_tel((String)session.getAttribute("session_u_tel"));
    ...
}
```

그러한 session을 통해 회원정보를 view로 노출

```
<div class="nickname">
    <h3>${session_u_nickname}</h3>
</div>
<div class="Btn">
    <button type="button" onclick="profileCh()">프로필변경</button>
</div>
</div>
</div>

<div class="middle-right">
    <div class="middle-right-1">
        <ul>
            <li><span class="title">회원명</span><span class="content">${session_u_name}</span></li>
            <li><span class="title">아이디</span><span class="content">${session_u_id}</span></li>
            <li><span class="title">휴대폰</span><span class="content">${session_u_tel}</span></li>
            <li><span class="title">이메일</span><span class="content">${session_u_email}</span></li>
        </ul>
    </div>
    <div class="middle-right-2">
        <div class="date">
            
            <h3>${session_u_name}<small>님의 회원 가입일</small></h3>
            <p><fmt:formatDate value="${session_u_signDate}" pattern="yyyy-MM-dd HH:mm"/></p>
            <p class="hi">우리가 만남 첫 기념일</p>
        </div>
        <div class="date2">
            
            <h3>${session_u_name}<small>님의 최종 접속일</small></h3>
            <p><fmt:formatDate value="${session_u_finalDate}" pattern="yyyy-MM-dd HH:mm"/></p>
            <p class="hi">반가워 오늘도 파이팅!!</p>
        </div>
    </div>
</div>
```

회원 정보 수정

회원정보는 ID와 이름을 제외한 모든 정보를 Update를 통해 수정한다.

사용자 이름	이민지
아이디	minji1234
비밀번호	<input type="text"/>
비밀번호 확인	<input type="text"/>
이메일	<input type="text"/>
휴대전화	<input type="text"/>

수정하기

수정 취소

```
<!-- 회원정보수정 -->
<update id="userModifyCom">
  update user_info set u_id=#{u_id} , u_pw=#{u_pw} , u_email=#{u_email}, u_tel=#{u_tel},u_moddate=sysdate where u_no=#{u_no}
</update>
```

수정이 완료되면 보안을 위해 회원정보 수정 완료라는 alert과 함께 session.invalidate 한 후에 session을 날리고 재 로그인 한다.(다시 메인 화면으로 이동)

```
<head>
  <meta charset="UTF-8">
  <title>회원정보변경</title>
</head>
<body>
<%
  session.invalidate();
%>
  <script type="text/javascript">
    alert("회원정보가 변경되었습니다.");
    alert("다시 로그인 해주시길 바랍니다.");
    location.href="/user/login";
  </script>
</body>
</html>
```

회원 탈퇴

회원 유형(u_detail_no)을 데이터 상에서 숫자로 구현하였다.

1은 일반회원, 9는 탈퇴 회원 0은 관리자

회원이 탈퇴를 하게 될 경우, 회원 유형(u_detail_no)이 1에서 9로 변하게 된다.

U_NO	U_DETAIL_NO	U_NAME
128	1	이민지
129	1	김지훈
130	0	admin
113	1	김말자
159	1	최준현
180	9	김은수

탈퇴 회원의 데이터 보관 및 삭제 여부는 관리자만이 판단 할 수 있다.

따라서 회원탈퇴를 누른다 해도 데이터가 바로 삭제 되지 않으며, 관리자페이지에서 승인 및 관리한다.

쿼리 또한 delete가 아닌 update로 회원 탈퇴 시 ,회원 유형(u_detail_no)을 1에서 9로 변환해준다.

```
<!-- 회원탈퇴 -->
<update id="userDelete">
    update user_info set u_detail_no = 9 where u_id=#{u_id}
</update>
```

회원 탈퇴를 하게 되면 alert과 함께 session.invalidate를 해주어야 한다.

```
<html>
<head>
    <meta charset="UTF-8">
    <title>회원탈퇴</title>
</head>
<body>
<%
    session.invalidate();

%>
<script type="text/javascript">
    alert("회원탈퇴가 성공적으로 이루어졌습니다.");
    location.href="/main";
</script>
</body>
</html>
```

탈퇴한 회원의 데이터 정보는 관리자 페이지에서 확인 할 수 있으며, 회원탈퇴는 관리자가 직접한다.

192.168.0.88:8181 내용:

선택회원을 탈퇴처리 하시겠습니까?

회원탈퇴 | CREW 현황 | 로드맵 현황 |

WITH TO US

회원ID | 검색어 입력 | 조회

10개씩 보기

<input type="checkbox"/>	번호	아이디	회원명	이메일	닉네임	회원가입일	최종접속일
<input type="checkbox"/>	1	admin	admin	admin1234@naver.com	관리자	2021-12-01 19:27:41.0	2021-12-08 14:00:21.0
<input checked="" type="checkbox"/>	2	zzzzzz	김말자	malza@naver.com	말자최고	2021-12-01 19:15:58.0	2021-12-06 13:48:48.0
<input type="checkbox"/>	3	sungho96982	김은수	kmhjn33@naver.com	안양의꽃사슴	2021-12-06 14:51:48.0	2021-12-06 14:52:51.0
<input checked="" type="checkbox"/>	4	ddd	김장훈	ddd@naver.com	진지한 고초아채	2021-11-19 10:46:45.0	2021-11-19 10:46:45.0
<input checked="" type="checkbox"/>	5	jihun1234	김지훈	jihune@naver.com	트럼프	2021-12-01 19:27:40.0	2021-12-06 13:48:24.0
<input type="checkbox"/>	6	aaaa	김춘자	test@naver.com	aaa	2021-11-19 10:40:59.0	2021-12-08 13:55:18.0
<div>상세정보 신고함 0 건 게시물 9 건 참여 CREW 8 건 진행 로드맵 21건 <input type="text" value="쪽지보내기"/> <input type="button" value="회원탈퇴"/> <input type="button" value="CLOSE"/></div>							
<input checked="" type="checkbox"/>	7	iii	나키오 카와구치	iii@naver.com	활동적인 악파만족	2021-11-19 10:46:53.0	2021-12-03 10:43:38.0
<input checked="" type="checkbox"/>	8	fff	옆집할매	fff@naver.com	새로운 자리권이	2021-11-19 10:46:48.0	2021-12-03 10:12:26.0
<input checked="" type="checkbox"/>	9	eee	원할머니	eee@naver.com	용감한 주름치마	2021-11-19 10:46:47.0	2021-12-03 10:12:02.0
<input checked="" type="checkbox"/>	10	ccc	이말자	ccc@naver.com	밝은 아마츄어	2021-11-19 10:46:44.0	2021-12-07 20:31:30.0

프로필 변경

프로필은 회원정보 변경과는 별개로 구성하였다.

프로필사진이나 닉네임 소개 글은 회원정보 변경과는 반대로 자주변경하기 때문에 따로 구분 해놓는다.

기본적으로 변경한 내용은 Dao를 통해 받아오지만 session에 마이 페이지가 session으로 노출하였기 때문에 controller에서 session으로 세팅해준다. 변경하지 않은 회원정보는 그대로 세팅 되어 있다.

```
//프로필 변경 완료
@RequestMapping("/modiProfile")
public String modify_profile(UserInfoVo userInfoVo, @RequestParam MultipartFile file, Model model, HttpServletRequest request) {
    HttpSession session = request.getSession();
    String newFileName = "";
    if (file.getSize() != 0) {
        String orginFileName = file.getOriginalFilename();
        long time = System.currentTimeMillis();
        newFileName = String.format("%d_%s", time, orginFileName);
        String fileSaveUrl = "C:/team1FileSave/";
        File f = new File(fileSaveUrl + newFileName);
        //파일 업로드
        try {
            file.transferTo(f);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    //파일을
    userInfoVo.setU_profile(newFileName);
    int result = userService.profileModify(userInfoVo);
    session.setAttribute("session_u_nickName", userInfoVo.getU_nickName());
    session.setAttribute("session_u_profile", userInfoVo.getU_profile());
    session.setAttribute("session_u_intro", userInfoVo.getU_intro());
    userInfoVo.setU_id((String)session.getAttribute("session_u_id"));
    userInfoVo.setU_email((String)session.getAttribute("session_u_email"));
    userInfoVo.setU_tel((String)session.getAttribute("session_u_tel"));
    userInfoVo.setU_name((String)session.getAttribute("session_u_name"));
    userInfoVo.setU_signDate((Timestamp)session.getAttribute("session_u_signDate"));
    userInfoVo.setU_modDate((Timestamp)session.getAttribute("session_u_modDate"));
    userInfoVo.setU_finalDate((Timestamp)session.getAttribute("session_u_finalDate"));
    System.out.println("controller:" + result);
    return "redirect:./my_info";
}
```

새로운 프로필사진

파일 선택 선택된 파일 없음

새로운 닉네임

새로운 소개글

변경 완료