

천재교육 풀스택 4기

OKKY

2조 / 조현기, 류림, 전주연

MARKETING PLAN PROPOSAL

목차

1. 프로젝트 개요
2. 프로젝트 팀 구성 및 역할
3. 프로젝트 수행 절차 및 방법
4. 프로젝트 수행 경과
5. 자체 평가 의견

01 프로젝트 개요

01
프로젝트 주제
및 선정 배경,
기획 의도

02
프로젝트
내용

03
활용 장비 및
재료

04
프로젝트 구조

05
활용방안 및
기대 효과

01

프로젝트 개요

01-01 프로젝트 주제 및 선정 배경, 기획 의도

프로젝트 주제 및 선정 배경,
기획 의도

- 개발자들간 의사소통이 용이한 사이트 구현
- 개발자들의 정보 공유가 활발한 사이트 구현
- 개발자들이 구인구직 하는 사이트 구현



OKKY 사이트 벤치 마킹

The screenshot shows the OKKY homepage. At the top, there's a navigation bar with 'OKKY' logo, 'Q&A', '자식', '커뮤니티', '이벤트', 'JOBS', '공지사항', and '로그인'. Below the navigation, there's a 'Top Writers' section listing users like '고불이별조립' (910), 'kunu' (269), '국민상고대' (244), 'ineemong77' (152), and 'n.n.n.1' (150). To the right, there's a banner for '비전공자도 4개월이면 나만의 앱을 스토어에 출시할 수 있어요' with icons for Android and iOS. Below the banner, there are two main sections: 'Q&A' and '커뮤니티'. The 'Q&A' section has a post by '리버씨' asking about Java and JSP. The '커뮤니티' section has a post by 'mochi' asking about UI design. On the right side of the page, there's an 'ISCC 인증 컨설팅' sidebar with text about eBrain.

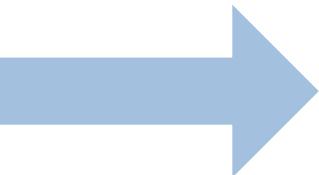
01

프로젝트 개요

01-02 프로젝트 내용

프로젝트 내용

- 프로젝트 구현 내용
- 훈련 내용과의 연관성



- 게시판 구조의 커뮤니티 사이트 구현
- HTML5, CSS3, JavaScript를 통해 UI 디자인 및 컨트롤
- Java를 기반으로 한 JSP와 DB 연동
- MVC모델 활용

01 프로젝트 개요

01-03 개발환경

개발 환경

DATA
BASE



LIBRARY



IDE/
Tools



- Eclipse 2023-12
- Apache Tomcat 10.1
 - Java 17
- Maria DB 10.11
- HeidiSQL 12.6
- Sequel Pro 1.1.2

WAS

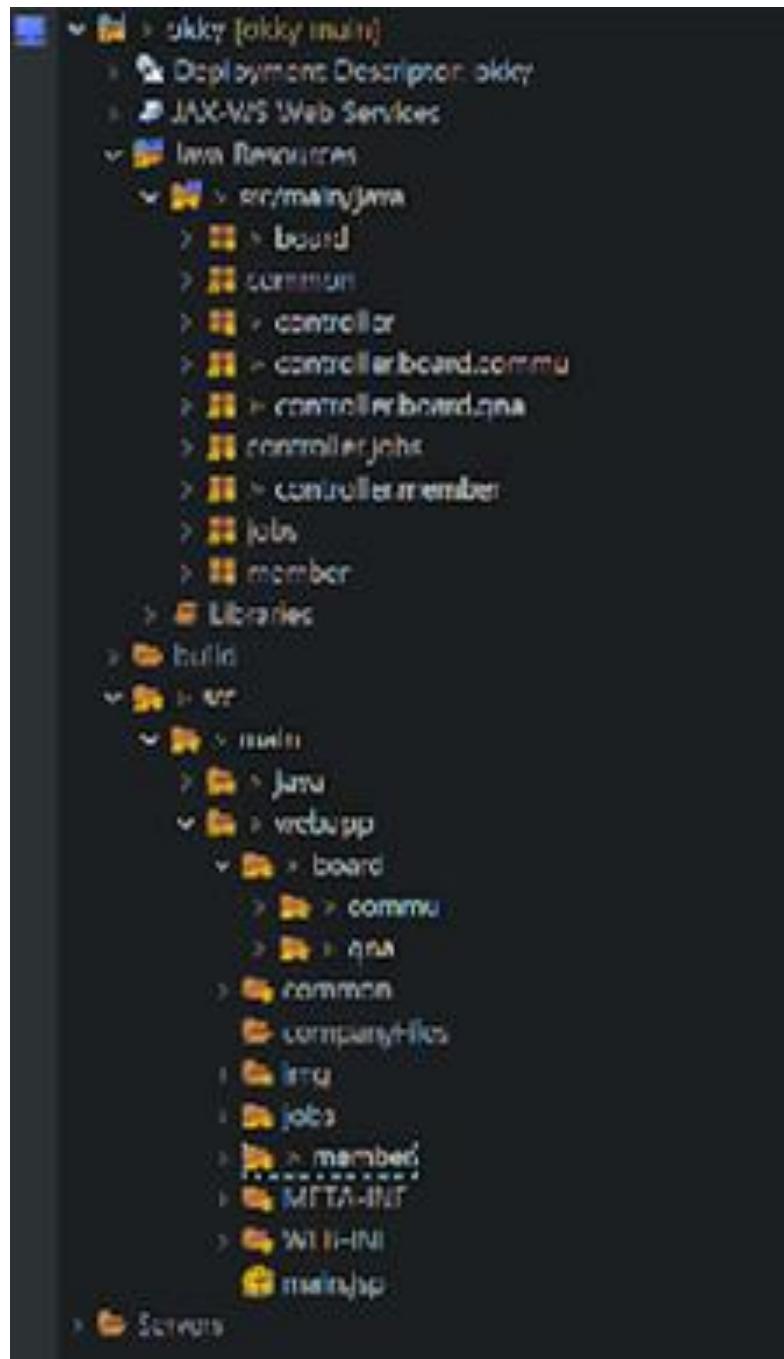


개발
언어



01 프로젝트 개요

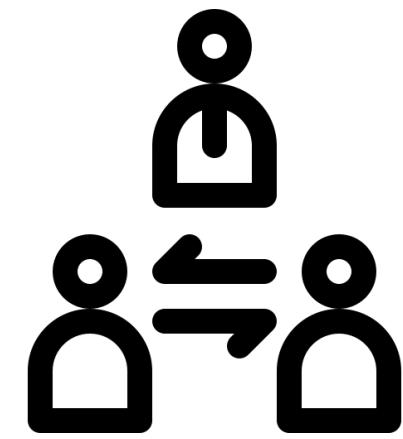
01-04 프로젝트 구조



- 각 메뉴 별 Controller 구분
- 각 메뉴 별 JSP 파일 구분

01 프로젝트 개요

01-05 활용방안 및 기대 효과



구인구직의 원활화

개발자와 기업 간 쉬운 커넥션을 통해
원활한 구인구직 활동을 기대할 수 있음



쉬운 의사소통 및 정보공유

사이트를 통해 개발자 간의 의사소통이 가능해지고
관련 정보를 쉽게 공유할 수 있음

02 프로젝트 팀 구성 및 역할

훈련생	역할	담당 업무
조현기	팀장	DB 테이블 설계 회원가입 / 로그인 페이지 디자인 및 기능 구현 구인구직 페이지 디자인 및 기능 구현 게시판 댓글 기능 구현
류림	팀원	DB 테이블 설계 QnA게시판, 커뮤니티 게시판 디자인 및 기능 구현 메인 페이지 디자인 및 기능 구현
전주연	팀원	DB 테이블 설계 마이페이지 디자인 및 기능 구현 헤더 디자인 및 기능 구현

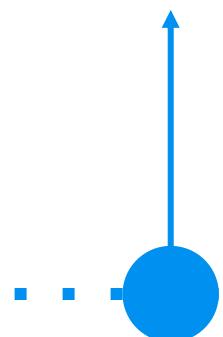
03 프로젝트 수행 절차 및 방법

3/19 (화)
사전 작업



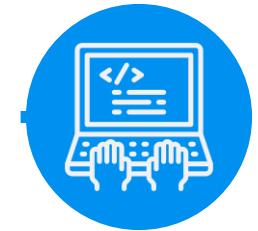
DB 테이블 설계
GitHub 연동
- 기획서 분석 작업
- 담당 페이지 역할 분담

3/20 (수)
디자인 구축



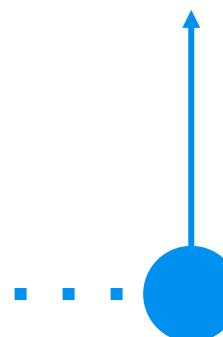
담당 페이지
디자인 구축

3/21(목) ~ 3/23(토)



기능 구현
각 담당 페이지
기능 구현
- 팀원 별 중간 점검

3/24(일)
기능 테스트



기능 테스트 후
수정 사항 반영

3/25(월)
배포 작업

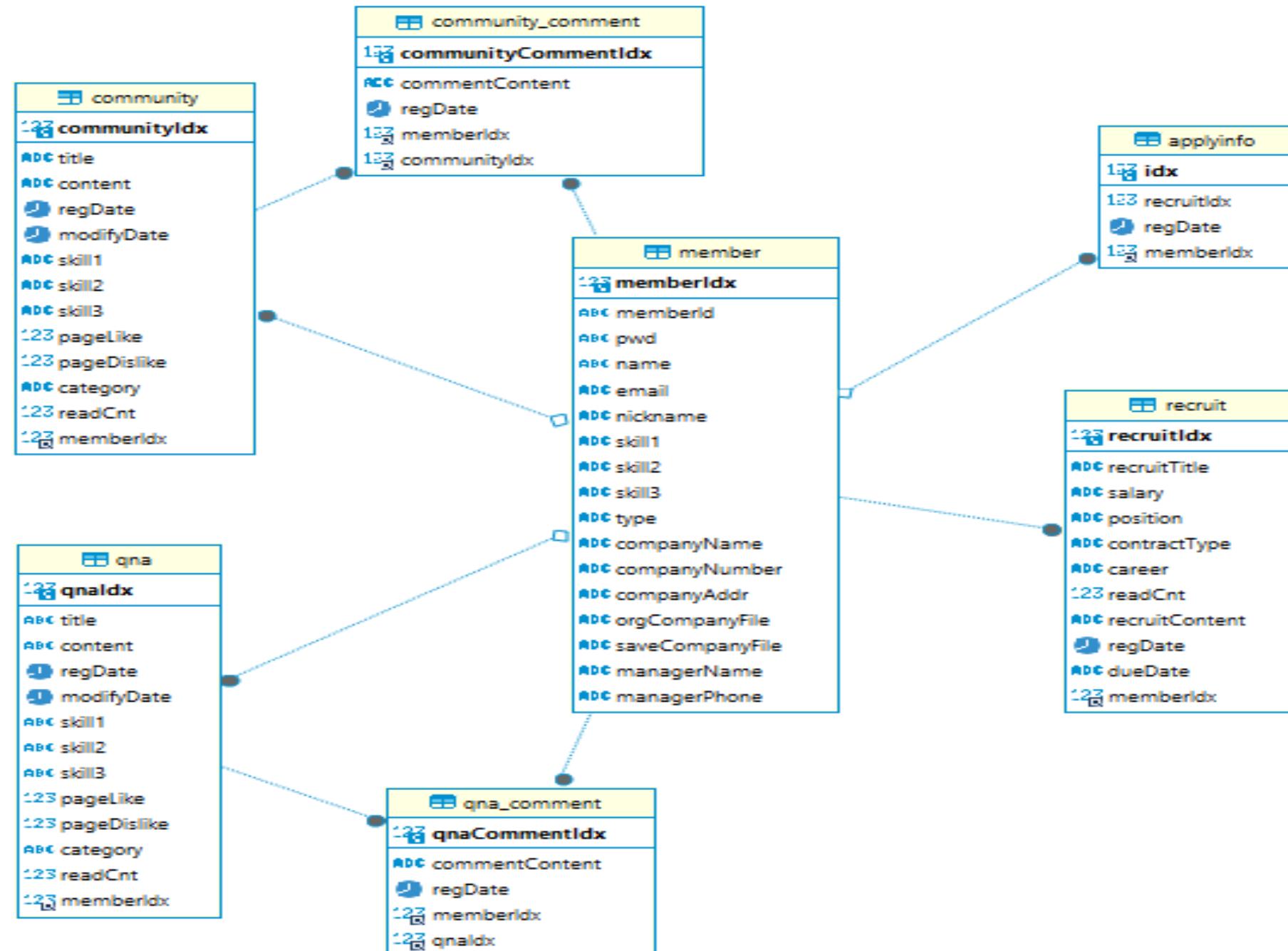


최종 작업물
배포 작업

04

프로젝트 수행 경과

데이터베이스 테이블 관계도



04

프로젝트 수행 경과

주요 컨텐츠

일반 회원 -회원가입

담당 - 조현기

기능 및 적용 기술

- 각 입력 조건에 맞는 유효성 검사를 진행하기 위한 함수 작성
- 조건에 따른 type을 지정하여 유효성 검사 함수 작성
- 조건에 충족하면 초록색, 아니면 빨간색 테두리로 표시하고, 모든 조건 충족 시 회원가입 버튼 활성화
- 버튼 클릭 시 아이디 중복 확인 진행

구현 이미지

회원가입

아이디

비밀번호

이메일

이름

닉네임

일반 회원 기업 회원

회원가입

회원가입

아이디

비밀번호

이메일

이름

닉네임

일반 회원 기업 회원

회원가입

회원가입

아이디 중복 오류
아이디 4~15자 이내

비밀번호
영문+숫자+특수문자 포함 6~20자 이내

이메일
이메일

이름
이름 2~30자 이내

닉네임
닉네임 2~20자 이내

일반 회원 기업 회원

회원가입

주요 코드

```
const engChars = "abcdefghijklmnopqrstuvwxyz";
const EngChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
const numChars = "0123456789";
const specialChars = " `~!@#$%^&*()-=_,.<,>/?;':\"\\`{}[] ";
const emailCheck = /^[A-Za-z0-9_\.\-]+@[A-Za-z0-9\-\-]+\.[A-za-z0-9\-\-]+/;
const korCheck = /[^-0-9]/g;

function combi_check(inputValue, chars) {
    let cnt = 0;
    for(let i=0; i<inputValue.length; i++) {
        if(chars.indexOf(inputValue.charAt(i)) != -1) {
            cnt += 1;
        }
    }

    return cnt;
}
```

```
function validate_check(inputValue, min, max, type) {
    if(type == 1) { // 영문 + 숫자 + 특수문자
        if(inputValue.value.length > min && inputValue.value.length <= max) {
            // 영문 + 숫자 + 특수문자
            if((combi_check(inputValue.value, engChars) > 0 || combi_check(inputValue.value, EngChars) > 0)
                && combi_check(inputValue.value, numChars) > 0
                && combi_check(inputValue.value, specialChars) > 0) {
                inputValue.style.border = "3px solid lightgreen";
            } else {
                inputValue.style.border = "1px solid red";
            }
        } else if(inputValue.value.length == 0) {
            inputValue.style.border = "";
        } else {
            inputValue.style.border = "1px solid red";
        }
    } else if(type == 2) { // 특수문자 x, 영문 + 숫자
        if(combi_check(inputValue.value, specialChars) > 0) {
            inputValue.value = inputValue.value.substring(0, inputValue.value.length - 1);
        } else {
            // 4~15자 이내 > input 배경색 변경
            if(inputValue.value.length >= min && inputValue.value.length <= max) {
                // 영문 + 숫자
                if((combi_check(inputValue.value, engChars) > 0 || combi_check(inputValue.value, EngChars) > 0)
                    && combi_check(inputValue.value, numChars) > 0) {
                    inputValue.style.border = "3px solid lightgreen";
                } else {
                    inputValue.style.border = "1px solid red";
                }
            } else if(inputValue.value.length == 0) {
                inputValue.style.border = "";
            } else {
                inputValue.style.border = "1px solid red";
            }
        }
    }
}
```

04

프로젝트 수행 경과

주요 컨텐츠

기업 회원 -회원가입

담당 - 조현기

기능 및 적용 기술

- 기업 회원 선택 시 기업에 대한 추가 정보 입력하는 칸 생성
- 파일 등록 기능 추가
- 각 입력 조건에 따른 type을 지정하여 유효성 검사를 진행하기 위한 함수 작성
- 조건에 충족하면 초록색, 아니면 빨간색 테두리로 표시하고, 모든 조건 충족 시 회원가입 버튼 활성화
- 버튼 클릭 시 아이디 중복 확인 진행

구현 이미지

회원가입

아이디

비밀번호

이메일

이름

닉네임

일반 회원 기업 회원

일반 회원 기업 회원

회사명

회사 주소

사업자등록번호

사업자 등록증 파일

담당자 이름

담당자 연락처

주요 코드

```
public int getMemberCount(String memberId) {  
    int result = 0;  
    String sql = "SELECT COUNT(memberId) FROM member WHERE memberId=?";  
  
    try {  
        psmt = conn.prepareStatement(sql);  
        psmt.setString(1, memberId);  
  
        rs = psmt.executeQuery();  
        if(rs.next()) {  
            result = rs.getInt(1);  
        }  
    } catch(Exception e) {  
        e.printStackTrace();  
        System.out.println("회원 조회 시 예외 발생");  
    }  
  
    return result;  
}  
  
if(type.equals("1")) {  
    result = dao.registMember(dto);  
} else {  
    // 파일 업로드 처리  
    String saveDir = req.getServletContext().getRealPath("companyFiles");  
    String orgFileName = "";  
  
    try {  
        orgFileName = FileUtil.uploadFile(req, saveDir);  
    } catch(Exception e) {}  
    e.printStackTrace();  
    req.setAttribute("uploadFileError", "회원가입 중 파일 업로드 에러 발생");  
    req.getRequestDispatcher("./regist.jsp").forward(req, resp);  
}  
  
if(orgFileName != "") {  
    String saveFileName = FileUtil.renameFile(saveDir, orgFileName);  
  
    dto.setOrgCompanyFile(orgFileName);  
    dto.setSaveCompanyFile(saveFileName);  
}  
result = dao.registMember(dto);  
dao.close();  
  
if(result > 0) {  
    resp.sendRedirect("./login.jsp");  
} else {  
    req.setAttribute("registErrMsg", "회원가입 실패");  
    req.getRequestDispatcher("./regist.jsp").forward(req, resp);  
}
```

04

프로젝트 수행 경과

주요 컨텐츠

로그인

담당 - 조현기

기능 및 적용 기술

- 로그인 버튼 클릭 시 회원정보를 조회
- 정보 조회 실패 시 로그인 오류 메세지 출력
- 정보 미입력 시 해당 입력칸 아래에 메세지 출력

로그인

아이디

아이디 4~15자 이내

비밀번호

영문+숫자+특수문자 포함 6~20자 이내

로그인

아직 회원이 아니신가요? [회원가입](#)

구현 이미지

로그인

아이디

아이디 4~15자 이내

비밀번호

영문+숫자+특수문자 포함 6~20자 이내

로그인

아직 회원이 아니신가요? [회원가입](#)

로그인

아이디

아이디 4~15자 이내

비밀번호

영문+숫자+특수문자 포함 6~20자 이내

비밀번호를 입력하세요.

로그인

아직 회원이 아니신가요? [회원가입](#)

주요 코드

```
// 로그인 - 회원 조회
public MemberDTO getMemberInfo(String id, String pwd) {
    MemberDTO dto = new MemberDTO();
    String sql = "SELECT * FROM member WHERE memberId=?";
    try {
        psmt = conn.prepareStatement(sql);
        psmt.setString(1, id);
        rs = psmt.executeQuery();
        if(rs.next()) {
            if(rs.getString("pwd").equals(pwd)) {
                dto.setMemberIdx(rs.getInt("memberIdx"));
                dto.setMemberId(rs.getString("memberId"));
                dto.setPwd(rs.getString("pwd"));
                dto.setEmail(rs.getString("email"));
                dto.setName(rs.getString("name"));
                dto.setNickname(rs.getString("nickname"));
                dto.setType(rs.getString("type"));
                dto.setCompanyName(rs.getString("companyName"));
                dto.setCompanyAddr(rs.getString("companyAddr"));
                dto.setCompanyNumber(rs.getString("companyNumber"));
                dto.setManagerName(rs.getString("managerName"));
                dto.setManagerPhone(rs.getString("managerPhone"));
                dto.setSkill1(rs.getString("skill1"));
                dto.setSkill2(rs.getString("skill2"));
                dto.setSkill3(rs.getString("skill3"));
                dto.setOrgCompanyFile(rs.getString("orgCompanyFile"));
            }
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("로그인 시 예외 발생");
        }
        return dto;
    }
```

```
<script>
document.querySelector("#frmLogin").addEventListener("submit", function(e) {
    e.preventDefault();

    const memberId = document.querySelector("#memberId");
    const pwd = document.querySelector("#pwd");

    // 아이디, 비밀번호 미입력 시 오류 메세지
    const warnMemberId = document.querySelector("#warn-memberId");
    const warnPwd = document.querySelector("#warn-pwd");

    if(memberId.value.trim().length == 0 && pwd.value.length == 0) {
        warnMemberId.style.display = "";
        warnMemberId.innerText = "아이디를 입력하세요.";
        warnPwd.style.display = "";
        warnPwd.innerText = "비밀번호를 입력하세요.";

        memberId.focus();
        e.preventDefault();
        return false;
    } else if(memberId.value.trim().length == 0 && pwd.value.length != 0) { // 아이디 미입력
        warnMemberId.style.display = "";
        warnMemberId.innerText = "아이디를 입력하세요.";
        warnPwd.style.display = "none";

        return false;
    } else if(memberId.value.trim() != 0 && pwd.value.trim().length == 0) { // 비밀번호 미입력
        warnMemberId.style.display = "none";
        warnPwd.style.display = "";
        warnPwd.innerText = "비밀번호를 입력하세요.";

        return false;
    } else {
        const frm = document.querySelector("#frmLogin");
        frm.submit();
    }
}, false);
```

04

프로젝트 수행 경과

주요 컨텐츠

구인구직 -메인화면(1)

담당 - 조현기

기능 및 적용 기술

- 가입된 일반회원 수, 기업회원 수, 하루동안 등록된 구직 게시글 수를 표시 (SQL문 코드 캡쳐)
- 계약직과 정규직으로 나누어 접근할 수 있는 버튼
- 로그인한 회원의 태입에 따라 일반회원은 이력서 등록, 기업회원은 공고 등록으로 이동할 수 있는 버튼
- 미로그인 시 공고 목록에 접근이 불가능하여 로그인 페이지로 이동

구현 이미지



주요 코드

```
// 일반 회원 count
public int getMemberCount() {
    int result = 0;

    String sql = "SELECT COUNT(memberIdx) FROM member WHERE TYPE='1'";
    try {
        psmt = conn.prepareStatement(sql);

        rs = psmt.executeQuery();
        rs.next();

        result = rs.getInt(1);
    } catch(Exception e) {
        e.printStackTrace();
        System.out.println("하루 게시된 글고 것수 확인 시 예외 발생");
    }
}

return result;
}

// 기업 회원 count
public int getCompanyCount() {
    int result = 0;

    String sql = "SELECT COUNT(memberIdx) FROM member WHERE TYPE='2'";
    try {
        psmt = conn.prepareStatement(sql);

        rs = psmt.executeQuery();
        rs.next();

        result = rs.getInt(1);
    } catch(Exception e) {
        e.printStackTrace();
        System.out.println("하루 게시된 글고 것수 확인 시 예외 발생");
    }
}

return result;
}

// 하루 게시된 글고 것수
public int getDayRecruitCount() {
    int result = 0;

    String sql = "SELECT COUNT(recruitIdx) FROM recruit WHERE regDate = DATE(NOW())";
    try {
        psmt = conn.prepareStatement(sql);

        rs = psmt.executeQuery();
        rs.next();

        result = rs.getInt(1);
    } catch(Exception e) {
        e.printStackTrace();
        System.out.println("하루 게시된 글고 것수 확인 시 예외 발생");
    }
}

return result;
}
```

04

프로젝트 수행 경과

주요 컨텐츠

구인구직
-메인화면(2)

담당 - 조현기

기능 및 적용 기술

- 가입된 일반회원 수, 기업회원 수, 하루동안 등록된 구직 게시글 수를 표시
- 계약직과 정규직으로 나누어 접근할 수 있는 버튼
- 로그인한 회원의 정보를 통해서 일반회원은 이력서 등록, 기업회원은 공고 등록으로 이동 할 수 있는 버튼

주요 코드

JSP

```
<a href="/okky/jobs/jobList.do?contractType=contract&sort=recent">
    <div class="bot-content">
        <div class="b-content">
            <div>Contract</div>
            <div>계약직</div>
        </div>
    </div>
</a>
<a href="/okky/jobs/jobList.do?contractType=fulltime&sort=recent">
    <div class="bot-content">
        <div class="b-content">
            <div>Fulltime</div>
            <div>정규직</div>
        </div>
    </div>
</a>
</div>
</div>
<div id="talent">
    <div id="talent-content">
        <div class="bot-ment">
            <div>Talent</div>
            <span>멋진 인재가 모여있는 곳</span>
        </div>
        <div class="bot-menu">
            <c:if test="${sessionScope.type eq '1'}">
                <a href="/okky/member/applyList.do">
                    <div id="talent-menu">
                        <div class="t-content">
                            <div>Resume</div>
                            <div>이력서 등록</div>
                        </div>
                    </div>
                </a>
            </c:if>
            <c:if test="${sessionScope.type eq '2'}">
                <a href="/okky/jobs/jobRecruit.do">
                    <div id="talent-menu">
                        <div class="t-content">
                            <div>Recruit</div>
                            <div>공고 등록</div>
                        </div>
                    </div>
                </a>
            </c:if>
        </div>
    </div>
</div>
```

JSP

```
<div id="count">
    <ul>
        <li id="count-ment">
            <h4>지금 이 순간에도 수많은 개발자와 일자리가 OKKY Jobs를 통해 활발히 만나고 있습니다.</h4>
        </li>
        <li class="content-count">
            <div>${total_member} 명</div>
            <span>가입 개인회원 수</span>
        </li>
        <li class="content-count">
            <div>${total_company} 사</div>
            <span>가입 기업회원 수</span>
        </li>
        <li class="content-count">
            <div>${total_position} 개</div>
            <span>오늘 하루 등록된 포지션 수</span>
        </li>
    </ul>
</div>
```

Controller

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    int total_member = 0;
    int total_company = 0;
    int total_position = 0;

    JobDAO dao = new JobDAO();
    total_member = dao.getMemberCount();
    total_company = dao.getCompanyCount();
    total_position = dao.getDayRecruitCount();

    dao.close();

    req.setAttribute("total_member", total_member);
    req.setAttribute("total_company", total_company);
    req.setAttribute("total_position", total_position);

    req.getRequestDispatcher("./jobMain.jsp").forward(req, resp);
}
```

04

프로젝트 수행 경과

주요 컨텐츠

구인구직
-통합 검색,
정렬, 페이징
(1)

담당 - 조현기

기능 및 적용 기술

- 검색어를 통해 회사명을 검색
- 존재하는 공고의 포지션, 지역, 경력 별로 필터링 검색 가능
- 최신순, 오래된순 선택에 따른 공고 게시글 정렬
- 공고 갯수에 따라 페이지 처리 기능
- 기업 회원만 공고 등록할 수 있는 버튼 존재

구현 이미지

The screenshot displays a job search interface with the following components:

- Search Bar:** A top bar with a search icon, "Search Keywords" input field, and "Search" button.
- Filter Buttons:** Below the search bar are three buttons: "포지션" (Position), "지역" (Region), and "경력" (Experience).
- Job Listings:** A grid of eight job cards, each featuring the "OKKYJOBS" logo. The cards provide details such as company name, position, experience level, and location.
- Page Navigation:** A bottom navigation bar with page numbers (1, 2) and arrows (<<, <, >, >>).
- Search Bar (Bottom):** A second search bar at the bottom with the same layout as the top one.
- Filter Sidebar:** A sidebar on the right side for filtering results by position, region, and experience.
 - Position:** Includes checkboxes for Frontend, Backend, DBA, Plastek, PM, and 기타 (Others).
 - Region:** Includes radio buttons for Seoul, Gyeonggi, Incheon, Daegu, Busan, and Gwangju.
 - Experience:** Includes checkboxes for 1~2 years, 3~4 years, 5~6 years, and 7 years or more.
- Sort Options:** Buttons for "정렬" (Sort) and "최신순" (Newest) and "오래된순" (Oldest) to sort the job listings.
- Buttons:** "공고등록" (Post Job) and "정렬" (Sort) buttons located at the bottom right.

주요 컨텐츠

구인구직
-통합 검색,
정렬, 페이징
(2)

담당 - 조현기

기능 및 적용 기술

- 검색어를 통해 회사명을 검색
- 존재하는 공고의 포지션, 지역, 경력 별로 검색 가능
- 최신순, 오래된순 선택에 따른 공고 게시글 정렬
- 공고 갯수에 따라 페이지 처리 기능
- 기업 회원은 공고 등록할 수 있는 버튼 존재

Controller

```

int total_count = 0;
int total_page = 1;
int page_no = 1;
int page_size = 8;
int page_skip_cnt = 8;
int page_block_size = 5;
int page_block_start = 1;
int page_block_end = 1;

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

String search_word = req.getParameter("search_word");
String pc = "";
String ac = "";
String cc = "";

if(search_word != null && !search_word.isEmpty()) {
    search_word = search_word.trim();

    maps.put("search_word", search_word);
}

if(req.getParameterValues("position_category") != null) {
    String[] position_category = req.getParameterValues("position_category");

    for(int i=0; i<position_category.length; i++) {
        if(i == position_category.length-1) {
            pc += "''" + position_category[i] + "'";
        } else {
            pc += "''" + position_category[i] + "', ";
        }
    }
}

if(req.getParameter("addr_category") != null) {
    ac = req.getParameter("addr_category");
}

if(req.getParameterValues("career_category") != null) {
    String[] career_category = req.getParameterValues("career_category");

    for(int i=0; i<career_category.length; i++) {
        if(i == career_category.length-1) {
            cc += "''" + career_category[i] + "'";
        } else {
            cc += "''" + career_category[i]+ "', ";
        }
    }
}

```

주요 코드

DAO

```

// (장규직, 계약직) 공고 리스트 조회
public List<JobDTO> getJobList(Map<String, Object> maps){
    List<JobDTO> list = new Vector<>();

    StringBuilder sb = new StringBuilder();
    sb.append("SELECT m.companyName, m.type, m.companyAddr, r.recruitIdx, r.position, r.career, r.contractType ");
    sb.append("FROM recruit AS r INNER JOIN member AS m ON m.memberIdx=r.memberIdx ");
    sb.append("WHERE companyName IS NOT NULL AND contractType = '" + maps.get("contractType") + "' ");

    // 검색어
    if(maps.get("search_word") != null) {
        sb.append("AND m.companyName LIKE '%" + maps.get("search_word") + "%' ");
    }

    // 필터 검색
    if(maps.get("position_category") != "") {
        sb.append("AND r.position IN (" + maps.get("position_category") + ") ");
    }

    if(maps.get("addr_category") != "") {
        sb.append("AND m.companyAddr LIKE '" + maps.get("addr_category") + "%' ");
    }

    if(maps.get("career_category") != "") {
        sb.append("AND r.career IN (" + maps.get("career_category") + ") ");
    }

    // 정렬 방법(최신순 기본)
    if(maps.get("sort").equals("old")) {
        sb.append(" ORDER BY regDate, r.recruitIdx ");
    } else {
        sb.append(" ORDER BY regDate DESC, r.recruitIdx DESC ");
    }

    // 페이지
    if(maps.get("page_size") != null && maps.get("page_skip_cnt") != null) {
        sb.append(" LIMIT " + maps.get("page_skip_cnt") + ", " + maps.get("page_size"));
    }
}

```

04

프로젝트 수행 경과

주요 컨텐츠

구인구직
-공고 등록

담당 - 조현기

기능 및 적용 기술

- 기업회원만 접근할 수 있는 공고 등록 페이지
- 회원가입 시 저장된 기업의 정보를 바탕으로 추가 정보를 입력해 공고를 등록하는 기능
- 모든 입력 칸 미입력 시 메세지 출력

회사 공고 등록

사용자23의 회사의 정보를 입력하고 공고를 등록하세요.

공고 제목

공고 제목을 입력해주세요.

회사명

사용자23의 회사

담당자 이름

김일오

담당자 연락처

01012345615

회사 주소

충남 천안시

구현 이미지

JSP - 미입력 시 메세지

```
let submit = document.querySelector("#submitBtn");
submit.addEventListener("click", (e) => {
    const recruitTitle = document.querySelector("#recruitTitle");
    const companyName = document.querySelector("#companyName");
    const managerName = document.querySelector("#managerName");
    const managerPhone = document.querySelector("#managerPhone");
    const companyAddr = document.querySelector("#companyAddr");
    const dueDate = document.querySelector("#dueDate");
    const salary = document.querySelector("#salary");
    const position = document.querySelector("#position");
    const contractType = document.querySelector("#contractType");
    const career = document.querySelector("#career");
    const recruitContent = document.querySelector("#recruitContent");

    if (recruitTitle.value == "") {
        document.querySelector("#recruitTitleCheck").style.display = "block";
        recruitTitle.focus();
        e.preventDefault();
        return false;
    }
    if (companyName.value == "") {
        document.querySelector("#companyNameCheck").style.display = "block";
        companyName.focus();
        e.preventDefault();
        return false;
    }
    if (managerName.value == "") {
        document.querySelector("#managerNameCheck").style.display = "block";
        managerName.focus();
        e.preventDefault();
        return false;
    }
})
```

주요 코드

Controller - 등록 처리

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    HttpSession session = req.getSession();
    int result = 0;
    int memberIdx = 0;
    String recruitTitle = req.getParameter("recruitTitle");
    String dueDate = req.getParameter("dueDate");
    String salary = req.getParameter("salary");
    String position = req.getParameter("position");
    String contractType = req.getParameter("contractType");
    contractType = contractType.equals("정규직") ? "fulltime" : "contract";
    String career = req.getParameter("career");
    String recruitContent = req.getParameter("recruitContent");

    if (session.getAttribute("memberIdx") != null) {
        String strMemberIdx = session.getAttribute("memberIdx").toString();
        memberIdx = strMemberIdx != null ? Integer.parseInt(strMemberIdx) : 0;

        JobDTO dto = new JobDTO();
        JobDAO dao = new JobDAO();

        dto.setRecruitTitle(recruitTitle);
        dto.setDueDate(dueDate);
        dto.setSalary(salary);
        dto.setPosition(position);
        dto.setContractType(contractType);
        dto.setCareer(career);
        dto.setRecruitContent(recruitContent);
        dto.setMemberIdx(memberIdx);

        result = dao.insertRecruit(dto);
    }
}
```

04

프로젝트 수행 경과

주요 컨텐츠

구인구직 -상세화면

담당 - 조현기

기능 및 적용 기술

- 등록된 정보를 보여주는 페이지
- 기업 회원 중 본인이 작성한 공고는 수정/삭제 버튼이 존재 (우측 이미지)
- 삭제 버튼 클릭 시 바로 삭제 가능

구현 이미지

◆ 차세대 형사사법시스템 구축

user10의 회사

마감일: 2024-04-05 조회수: 1

포지션	필요경력	근무지역	연봉
풀스택	무관	경남 경주시	면접 시 협의

프로젝트 소개

◆ 차세대 형사사법시스템 구축

- 업무 및 역할 : 파일전환팀 개발자 기술리딩
- * 내부적으로 각 서브시스템별 담당자 6인과 협력/가이드 필요
- * 외부적으로 데이터전환, TA/DA/DBA/보안/응용 등 의사소통
- * 고객 의사소통은 PL 및 풍괄이 담당하므로 테크니컬 이슈 관리에만 집중
- 스릴 및 스팩 (필수)
- * PL역할 경험자(의사소통 능력)
- * 데이터 전환 경험
- * JAVA프로그래밍 경험
- * ORACLE DB 프로시저 작성, SQL튜닝
- 기간 : 투입 ~ 11월

※ 지원 시 스킬인벤토리 포함한 양식의 이력서와 함께 최장급여(10일) 및 투입 날짜 기재 부탁드리겠습니다

대형쇼핑몰/마트 운영인력

사용자23의 회사

마감일: 2024-04-17 조회수: 3

포지션	필요경력	근무지역	연봉
백엔드	무관	충남 천안시	2000~2999만 원

프로젝트 소개

대형쇼핑몰/마트 운영인력 / 4월초~1년단위 연장 / 필수스킬 : ASP.NET MVC (with C#) / MSSQL, 쇼핑몰 또는 마트 경험자 우대 / 중급 1명 / 장기근무 가능자(최소2~3년) / 중급 1명 /

대형쇼핑몰/마트 운영인력

.기간 : 4월초~

.필수스킬 : ASP.NET MVC (with C#) / MSSQL, 쇼핑몰 또는 마트 경험자 우대

.기타 : 장기근무 가능자

주요 코드

```
// 꿈고 꿈 조회
public JobDTO getJobDetail(int recruitIdx){
    JobDTO dto = new JobDTO();

    StringBuilder sb = new StringBuilder();
    sb.append("SELECT m.memberIdx, m.companyName, m.companyAddr, m.managerName, m.managerPhone, ");
    sb.append("r.recruitTitle, r.dueDate, r.readCnt, r.salary, r.position, r.contractType, r.career, r.recruitContent ");
    sb.append("FROM recruit r INNER JOIN member AS m ON m.memberIdx = r.memberIdx ");
    sb.append("WHERE r.recruitIdx=?");

    try {
        psmt = conn.prepareStatement(sb.toString());
        psmt.setInt(1, recruitIdx);
        rs = psmt.executeQuery();

        if(rs.next()) {
            dto.setMemberIdx(rs.getInt("memberIdx"));
            dto.setCompanyName(rs.getString("companyName"));
            dto.setCompanyAddr(rs.getString("companyAddr"));
            dto.setManagerName(rs.getString("managerName"));
            dto.setManagerPhone(rs.getString("managerPhone"));
            dto.setRecruitIdx(rs.getInt("recruitIdx"));
            dto.setRecruitTitle(rs.getString("recruitTitle"));
            dto.setDueDate(rs.getString("dueDate"));
            dto.setReadCnt(rs.getInt("readCnt"));
            dto.setSalary(rs.getString("salary"));
            dto.setPosition(rs.getString("position"));
            dto.setContractType(rs.getString("contractType"));
            dto.setCareer(rs.getString("career"));
            dto.setRecruitContent(rs.getString("recruitContent"));
        }
    } catch(Exception e) {
        e.printStackTrace();
        System.out.println("꿈고 꿈 조회 시 예외 발생");
    }

    return dto;
}

<c:if test="${sessionScope.memberIdx eq jobView.memberIdx}">
    <div class="btns">
        <button class="btn-update" id="btn_update">수정하기</button>
        <button class="btn-delete" id="btn_delete">삭제하기</button>
    </div>
</c:if>
<div class="info-top">
    <div id="title" class="info">${jobView.recruitTitle}</div>
    <div id="companyName" class="info">${jobView.companyName}</div>
    <div class="info" style="float: left; margin-right: 50px;">마감일: ${jobView.dueDate}</div>
    <div class="info">조회수: ${jobView.readCnt}</div>
</div>
<div class="info-content">
    <ul>
        <li>포지션</li>
        <span>${jobView.position}</span>
    </div>
    <div>
        <li>필요경력</li>
        <span>${jobView.career}</span>
    </div>
    <div>
        <li>근무지역</li>
        <span>${jobView.companyAddr}</span>
    </div>
    <div>
        <li>연봉</li>
        <span>${jobView.salary}</span>
    </div>
    </ul>
</div>
<hr style="border: 5px solid gray">
<div class="company-intro">
    <div class="intro-content">
        <p>${jobView.recruitContent}</p>
    </div>
</div>

```

04

프로젝트 수행 경과

주요 컨텐츠

구인구직
-수정화면

담당 - 조현기

기능 및 적용 기술

- 등록된 정보를 수정하기 위한 페이지
- 기존에 작성한 내용들을 불러와서 추가 수정 작업 가능
- 회사 이름 및 주소, 담당자 이름 및 연락처는 수정 불가능

구현 이미지

공고 제목
대형쇼핑몰/마트 운영인력

회사명
사용자23의 회사

담당자 이름
김일오

담당자 연락처
01012345615

회사 주소
충남 천안시

마감 기한
2024-04-17

2000~2999만원 ▾ 백엔드 ▾ 정규직 ▾ 무관 ▾

공고 본문
프로젝트 소개
대형쇼핑몰/마트 운영인력 / 4월초~1년단위 연장 / 필수스킬 : ASP.NET MVC (with C#) / MSSQL , 쇼핑몰 또는 마트 경험자 우대 / 중급 1명 / 장기근무 가능자(최소2~3년) / 중급 1명 /
대형쇼핑몰/마트 운영인력
기간 : 4월초~
필수스킬 : ASP.NET MVC (with C#) / MSSQL , 쇼핑몰 또는 마트 경험자 우대
기타 : 장기근무가능자

주요 코드

```
@Override  
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
    HttpSession session = req.getSession();  
    int recruitIdx = req.getParameter("recruitIdx") != null ? Integer.parseInt(req.getParameter("recruitIdx")) : 0;  
  
    if (session.getAttribute("memberIdx") != null) {  
        String strMemberIdx = session.getAttribute("memberIdx").toString();  
        int memberIdx = strMemberIdx != null ? Integer.parseInt(strMemberIdx) : 0;  
  
        if(recruitIdx > 0) {  
            JobDAO dao = new JobDAO();  
            JobDTO jobView = dao.getJobDetail(recruitIdx);  
            dao.close();  
  
            req.setAttribute("jobView", jobView);  
  
            req.getRequestDispatcher("/jobs/jobModify.jsp?recruitIdx=" + recruitIdx).forward(req, resp);  
        } else {  
            resp.sendRedirect("../jobs/jobMain.do");  
        }  
    } else {  
        resp.sendRedirect("../jobs/jobMain.do");  
    }  
}
```

04

프로젝트 수행 경과

주요 컨텐츠

QnA, 커뮤니티
-게시판 댓글

담당 - 조현기

기능 및 적용 기술

- 게시판 댓글 작성 기능
- 댓글 작성한 회원 닉네임, 작성 일자, 작성 내용의 목록을 보여 주는 기능

3 개의 댓글

안녕하세요~

구현 이미지

3 개의 댓글

유저닉네임23 · 2024-04-01

안녕하세요!

반가워요

user5 닉네임 · 2024-03-27

안녕

유저닉네임24 · 2024-03-25

녕!

댓글작성

DAO - 댓글 작성

```
public int writeQnaComment(CommentDTO inputDTO) {  
    int result = 0;  
  
    StringBuilder sb = new StringBuilder();  
    sb.append("INSERT INTO qna_comment(commentContent, regDate, memberIdx, qnaIdx) ");  
    sb.append("VALUES (?, NOW(), ?, ?)");  
  
    try {  
        psmt = conn.prepareStatement(sb.toString());  
        psmt.setString(1, inputDTO.getCommentContent());  
        psmt.setInt(2, inputDTO.getMemberIdx());  
        psmt.setInt(3, inputDTO.getQnaIdx());  
          
        result = psmt.executeUpdate();  
    } catch(Exception e) {  
        e.printStackTrace();  
        System.out.println("댓글 작성 중 예외 발생");  
    }  
  
    return result;  
}
```

주요 코드

DAO - 댓글 목록

```
public List<CommentDTO> getQnaCommentList(int qnaIdx) {  
    List<CommentDTO> list = new Vector<CommentDTO>();  
  
    StringBuilder sb = new StringBuilder();  
    sb.append("SELECT c.commentContent, c.regDate, m.memberIdx, m.nickname, q.qnaIdx ");  
    sb.append("FROM qna_comment AS c ");  
    sb.append("INNER JOIN member AS m ON m.memberIdx = c.memberIdx ");  
    sb.append("INNER JOIN qna AS q ON q.qnaIdx = c.qnaIdx ");  
    sb.append("WHERE q.qnaIdx = ? ");  
    sb.append("ORDER BY c.regDate desc, c.qnaCommentIdx desc");  
  
    try {  
        psmt = conn.prepareStatement(sb.toString());  
        psmt.setInt(1, qnaIdx);  
  
        rs = psmt.executeQuery();  
        while(rs.next()) {  
            CommentDTO dto = new CommentDTO();  
  
            dto.setCommentContent(rs.getString("commentContent"));  
            dto.setRegDate(rs.getString("regDate"));  
            dto.setMemberIdx(rs.getInt("memberIdx"));  
            dto.setNickname(rs.getString("nickname"));  
            dto.setQnaIdx(rs.getInt("qnaIdx"));  
  
            list.add(dto);  
        }  
    } catch(Exception e) {  
        e.printStackTrace();  
    }  
}
```

04

프로젝트 수행 경과

주요 컨텐츠

일반 회원
-마이페이지
(프로필)

담당 - 전주연

기능 및 적용 기술

- 개인 회원이 로그인 후 마이페이지 클릭 시 보여지는 화면
- 닉네임 중복 체크 후 변경 가능
- 스킬 태그는 선택 사항으로 체크 후 저장 가능
- JSP 및 데이터베이스 사용 홈페이지 구현, 정보 저장 및 수정
- 중복 확인에 대한 부분을 고민하고 가장 최선의 방법을 찾아 적용시킬 수 있었음

구현 이미지



주요 코드

```


<p></p></div>
<div class="label" for="nickname">닉네임</label><div>
<div id="nicknameBox">
  <c:if test="${not empty checkNickname}">
    <input type="text" name="nickname" id="nickname" value="${checkNickname}">
  </c:if>
  <c:if test="${empty checkNickname}">
    <input type="text" name="nickname" id="nickname" value="${nickname}">
  </c:if>
  <input type="button" id="checkBtn" value="중복 확인" disabled>
</div>
<c:if test="${not empty nNameErrMsg}">
  <p style="color: red; font-weight: bold;">${nNameErrMsg}</p>
</c:if>
<c:if test="${not empty nNameOkayMsg}">
  <p style="color: blue; font-weight: bold;">${nNameOkayMsg}</p>
</c:if>
<div class="label" for="skill1">선택 1</div>
<div id="selectCon1">
  <select name="skill1" id="skill1_1st">
    <option value=""><c:if test="${skill1 eq ''}">selected</c:if>선택 1</option>
    <option value="프론트엔드"><c:if test="${skill1 eq '프론트엔드'}">selected</c:if>프론트엔드</option>
    <option value="백엔드"><c:if test="${skill1 eq '백엔드'}">selected</c:if>백엔드</option>
    <option value="풀스택"><c:if test="${skill1 eq '풀스택}">selected</c:if>풀스택</option>
  </select>
</div>
<div id="selectCon2">
  <select name="skill12" id="skill1_2nd">
    <option value=""><c:if test="${skill2 eq ''}">selected</c:if>선택 2</option>
    <option value="DB"><c:if test="${skill2 eq 'DB'}">selected</c:if>DB</option>
    <option value="SQL"><c:if test="${skill2 eq 'SQL}">selected</c:if>SQL</option>
    <option value="스크립트"><c:if test="${skill2 eq '스크립트}">selected</c:if>스크립트</option>
    <option value="자바"><c:if test="${skill2 eq '자바'}">selected</c:if>자바</option>
    <option value="자바스크립트"><c:if test="${skill2 eq '자바스크립트}">selected</c:if>자바스크립트</option>
    <option value="HTML"><c:if test="${skill2 eq 'HTML'}">selected</c:if>HTML</option>
  </select>
</div>


```

```

@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
  HttpSession session = req.getSession();
  String memberId = session.getAttribute("memberId") == null ? "" : session.getAttribute("memberId").toString();
  String name = session.getAttribute("name") == null ? "" : session.getAttribute("name").toString();
  String pwd = session.getAttribute("pwd") == null ? "" : session.getAttribute("pwd").toString();

  String nicknameF = req.getParameter("nickname");
  String nickname = session.getAttribute("nickname") == null ? "" : session.getAttribute("nickname").toString();
  String skill1 = session.getAttribute("skill1") == null ? "" : session.getAttribute("skill1").toString();
  String skill2 = session.getAttribute("skill2") == null ? "" : session.getAttribute("skill2").toString();
  String skill3 = session.getAttribute("skill3") == null ? "" : session.getAttribute("skill3").toString();

  MemberDAO dao = new MemberDAO();
  MemberDTO dto = dao.getMemberInfo(memberId, pwd);
  dao.close();

  memberId = dto.getMemberId();
  pwd = dto.getPwd();
  name = dto.getName();
  nickname = dto.getNickname();
  skill1 = dto.getSkill1();
  skill2 = dto.getSkill2();
  skill3 = dto.getSkill3();

  req.setAttribute("nickname", nickname);
  req.setAttribute("name", name);
  req.setAttribute("skill1", skill1);
  req.setAttribute("skill2", skill2);
  req.setAttribute("skill3", skill3);
  req.getRequestDispatcher("./mypage.jsp").forward(req, resp);
}

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
  HttpSession session = req.getSession();
  String email = session.getAttribute("email") == null ? "" : session.getAttribute("email").toString();
  String emailN = req.getParameter("email") == null ? "" : req.getParameter("email");

  MemberDAO dao = new MemberDAO();
  int Email_check = dao.getEmailCount(emailN);
  dao.close();

  if (Email_check == 0) {
    req.setAttribute("checkEmail", emailN);
    req.setAttribute("emailOkayMsg", "사용 가능한 닉네임입니다.");
    req.getRequestDispatcher("./account.jsp").forward(req, resp);
  } else {
    req.setAttribute("email", emailN);
    req.setAttribute("emailMsg", "중복된 이메일 주소입니다.");
    req.getRequestDispatcher("./account.jsp").forward(req, resp);
  }
  req.getRequestDispatcher("./mypage.jsp").forward(req, resp);
}
}

```

04

프로젝트 수행 경과

주요 컨텐츠

기업 회원
-마이페이지
(프로필)

담당 - 전주연

기능 및 적용 기술

- 기업 회원이 로그인 후 마이페이지 클릭 시 보여지는 페이지
- 담당자명과 담당자 연락처 수정 가능
- 조건에 맞으면 초록색 테두리 체크가 되며 저장 버튼 활성화
- 조건에 맞지 않으면 저장 버튼을 클릭했을 때 변경이 되지 않는 기능 적용 시킴

구현 이미지

회원 정보

담당자명

회사명

사업자 등록번호

담당자 연락처

저장

저장

주요 코드

```
<ul>
    <li><h3>내 계정</h3></li>
    <a href="#"><li class="detail">프로필</li></a>
    <a href="#"><li class="detail">계정 관리</li></a>
</ul>
<hr>
<ul>
    <li><h3>JOBS</h3></li>
    <a href="#"><li class="detail">구직 내역 관리</li></a>
</ul>
</div>
<aside>
<div id="outer">
    <div id="innerContainer">
        <h3>회원 정보</h3>
        <form action="mypage_co.do" name="infoForm" id="infoForm" method="post">
            <div class="label"><label for="managerName">담당자명</label></div>
            <div class="label"><label for="companyName">회사명</label></div>
            <div class="label"><label for="companyNo">사업자 등록번호</label></div>
            <div class="label"><label for="companyPhone">담당자 연락처</label></div>
            <div><input type="text" name="managerName" value="${managerName }" readonly></div>
            <div><input type="text" name="companyName" value="${companyName }" readonly></div>
            <div><input type="text" name="companyNo" value="${companyNo }" readonly></div>
            <div><input type="text" name="companyPhone" value="${managerPhone }" placeholder="제외 번호" disabled></div>
            <div id="errPno" style="display: none;"></div>
            <div class="saveBtnArea">
                <input type="submit" id="saveBtn" name="saveBtn" value="저장" disabled>
            </div>
        </form>
    </div>
</div>
```

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    HttpSession session = req.getSession();
    String memberId = session.getAttribute("memberId") == null ? "" : session.getAttribute("memberId").toString();
    String name = session.getAttribute("name") == null ? "" : session.getAttribute("name").toString();
    String managerName = req.getParameter("managerName");
    String companyName = req.getParameter("companyName");
    String companyNo = req.getParameter("companyNo");
    String managerPhone = req.getParameter("managerPhone");
    String type = req.getParameter("type");

    MemberDAO dao = new MemberDAO();
    MemberDTO dto = dao.getCompanyInfo(memberId, name);

    dao.close();

    memberId = dto.getMemberId();
    managerName = dto.getManagerName();
    companyName = dto.getCompanyName();
    companyNo = dto.getCompanyNumber();
    managerPhone = dto.getManagerPhone();
    type = dto.getType();

    req.setAttribute("memberId", memberId);
    req.setAttribute("managerName", managerName);
    req.setAttribute("companyName", companyName);
    req.setAttribute("companyNo", companyNo);
    req.setAttribute("managerPhone", managerPhone);
    req.setAttribute("type", type);

    req.getRequestDispatcher("./mypage_co.jsp").forward(req, resp);
}
```

04

프로젝트 수행 경과

주요 컨텐츠

회원 공통
-마이페이지
(계정 관리)

담당 - 전주연

기능 및 적용 기술

- 회원 공통으로 계정 관리 클릭 시 나타나는 페이지
- 비밀번호 변경/ 이메일 변경/ 회원 탈퇴 기능 有
- 닉네임 중복 체크 후 변경 가능
- 스킬 태그는 선택 사항으로 체크 후 저장 가능
- JSP 및 데이터베이스 사용으로 정보 저장 및 수정
- 비밀번호 변경 버튼 활성화 후, 조건에 맞지 않으면 안내 문구 출력

구현 이미지

주요 코드

```
as follows:


<div id="innerContainer1">
    <h3>비밀번호 변경</h3>
    <form name="pwdForm" id="pwdForm" action="changepwd.do" method="post">
      <div class="label"><label for="pwd">현재 비밀번호</label></div>
      <div><input type="password" name="pwd" id="pwd" value="" placeholder="현재 사용중인 비밀번호를 입력해주세요"></div>
      <p style="display: none" id="ment1"></p>
      <div class="label"><label for="newPwd">신규 비밀번호</label></div>
      <div><input type="password" name="newPwd" id="newPwd" value="" placeholder="영문+숫자+특수문자 포함 6~20자 이내로 입력"></div>
      <div class="label"><label for="newPwdCk">비밀번호 확인</label></div>
      <div><input type="password" name="newPwdCk" id="newPwdCk" value="" placeholder="새로운 비밀번호를 다시 한번 입력해주세요"></div>
      <p style="display: none" id="ment2"></p>
      <div class="saveBtnArea">
        <input type="submit" id="changeBtn" name="changeBtn" value="비밀번호 변경" disabled>
      </div>
    </form>
  </div>


```

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
  HttpSession session = req.getSession();
  String memberId = session.getAttribute("memberId") == null ? "" : session.getAttribute("memberId").toString();
  String pwd = session.getAttribute("pwd") == null ? "" : session.getAttribute("pwd").toString();
  String pwd1 = req.getParameter("pwd");
  String pwd2 = req.getParameter("newPwd");
  String pwd2Ck = req.getParameter("newPwdCk");

  MemberDTO dto = new MemberDTO();
  MemberDAO dao = new MemberDAO();
  if(pwd2.equals(pwd2Ck)) {
    dto.setPwd(pwd2);
    dto.setMemberId(memberId);
  }
  int result = dao.changePwd(dto);
  dao.close();
  session.setAttribute("pwd", pwd2);
  if(result > 0) {
    resp.sendRedirect("./account.do");
  } else {
    req.getRequestDispatcher("./account.jsp").forward(req, resp);
  }
}
```

04

프로젝트 수행 경과

주요 컨텐츠

회원 공통
-마이페이지
(계정 관리)

담당 - 전주연

기능 및 적용 기술

- 이메일 정보 변경 가능
- 중복 체크 후 변경 가능
- 사용가능 한 닉네임일 경우
이메일 변경 가능
- 이메일 형식에 맞지 않을 경우
중복 확인 불가능
- JSP 및 데이터베이스 사용으로
정보 저장 및 수정

구현 이미지

이메일 정보

kim1@gmail.com

증복 확인

저장

이메일 정보

kim1@naver.com

증복 확인

사용가능한 닉네임입니다.

증복 확인

kim1@nave.

증복 확인

저장

주요 코드

```
<div id="innerContainer2">
    <form name="emailForm" id="emailForm" action="changeEmail.do" method="post">
        <div class="label"><label for="email">이메일 정보</label></div>
        <div id="emailCkBox">
            <c:if test="${not empty checkEmail}">
                <input type="text" name="email" id="email" value="${checkEmail}">
            </c:if>
            <c:if test="${empty checkEmail}">
                <input type="text" name="email" id="email" value="${sessionScope.email}">
            </c:if>
            <input type="button" id="checkBtn" value="중복 확인" disabled></div>
            <c:if test="${not empty emailErrMsg}">
                <p style="color: red; font-weight: bold;">${emailErrMsg}</p>
            </c:if>
            <c:if test="${not empty emailOkayMsg}">
                <p style="color: blue; font-weight: bold;">${emailOkayMsg}</p>
            </c:if>
            <p style="display: none" id="emailment"></p>
        <div class="saveBtnArea">
            <input type="submit" id="saveBtn" name="saveBtn" value="저장" <c:if test="${empty emailOkayMsg}">disabled</c:if>
        </div>
    </form>
</div>
```

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    HttpSession session = req.getSession();
    String memberId = session.getAttribute("memberId") == null ? "" : session.getAttribute("memberId").toString();
    String name = session.getAttribute("name") == null ? "" : session.getAttribute("name").toString();
    String email = session.getAttribute("email") == null ? "" : session.getAttribute("email").toString();
    String emailN = req.getParameter("email");
    MemberDTO dto = new MemberDTO();
    MemberDAO dao = new MemberDAO();

    if(!email.equals(emailN)) {
        dto.setEmail(emailN);
        dto.setMemberId(memberId);
    }
    int result = dao.changeMail(dto);
    dao.close();
    session.setAttribute("email", emailN);
    if(result > 0) {
        resp.sendRedirect("./account.do");
    } else {
        req.getRequestDispatcher("./account.jsp").forward(req, resp);
    }
}
```

04

프로젝트 수행 경과

주요 컨텐츠

회원 공통 -마이페이지 (계정관리)

담당 - 전주연

기능 및 적용 기술

- 사이트 회원 탈퇴 기능
- 개인정보 처리 방침 클릭 시 관련 정보 확인 가능
- 체크 박스 확인 시 탈퇴 버튼 활성화
- 데이터베이스 내에 저장되어 있는 회원 정보 삭제

계정 삭제

회원 탈퇴일로부터 계정과 닉네임을 포함한 계정 정보(아이디/이메일/닉네임)는 [개인정보 처리방침](#)에 따라 **60일간 보관(잠김)**되며, 60일 경과된 후에는 모든 개인 정보는 완전히 삭제되며 더 이상 복구할 수 없게 됩니다.
작성된 게시물은 삭제되지 않으며, 익명처리 후 OKKY로 소유권이 귀속됩니다.

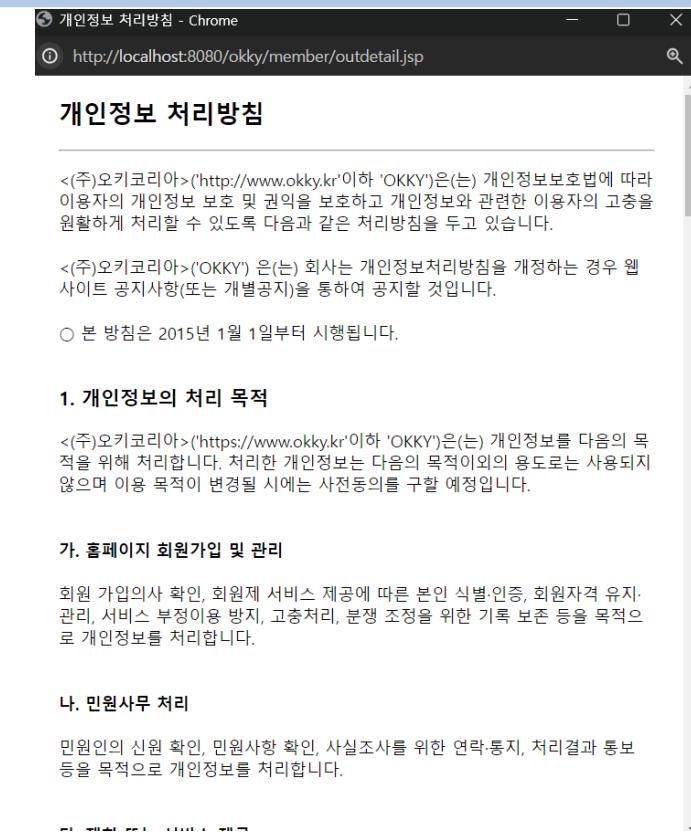
계정 삭제에 관한 정책을 읽고 이에 동의합니다.

탈퇴

계정 삭제에 관한 정책을 읽고 이에 동의합니다.

탈퇴

구현 이미지



주요 코드

```
<div id="innerContainer3">
    <form name="deleteForm" id="deleteForm" action="delete.do" method="post">
        <div class="label"><label for="accountDel">계정 삭제</label></div>
        <div id="delText">
            <p>회원 탈퇴일로부터 계정과 닉네임을 포함한 계정 정보(아이디/이메일/닉네임)는<br>
                <a href="#">javascript:openWindowPop('./outdetail.jsp', 'popup');</a><span class="blue underline">개인정보 처리방침</span><br>
                60일 경과된 후에는 모든 개인 정보는 완전히 삭제되며 더 이상 복구할 수 없게 됩니다.<br>
                작성된 게시물은 삭제되지 않으며, 익명처리 후 OKKY로 소유권이 귀속됩니다.</p>
            </div>
            <div><input type="checkbox" id="accountCkbox" name="accountCkbox" checked="" /></div>
            <div class="saveBtnArea">
                <input type="submit" id="outBtn" name="outBtn" value="탈퇴" disabled="disabled" />
            </div>
        </div>
    </form>
</div>
```

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    int result = 0;
    HttpSession session = req.getSession();
    String memberId = session.getAttribute("memberId") == null ? "" : session.getAttribute("memberId").toString();
    MemberDTO dto = new MemberDTO();
    MemberDAO dao = new MemberDAO();
    if(memberId != "") {
        dto.setMemberId(memberId);
        result= dao.deleteAccount(dto);
    }
    dao.close();
    session.invalidate();
    if(result > 0) {
        resp.sendRedirect("../main.jsp");
    } else {
        req.getRequestDispatcher("../main.jsp").forward(req, resp);
    }
}
```

04

프로젝트 수행 경과

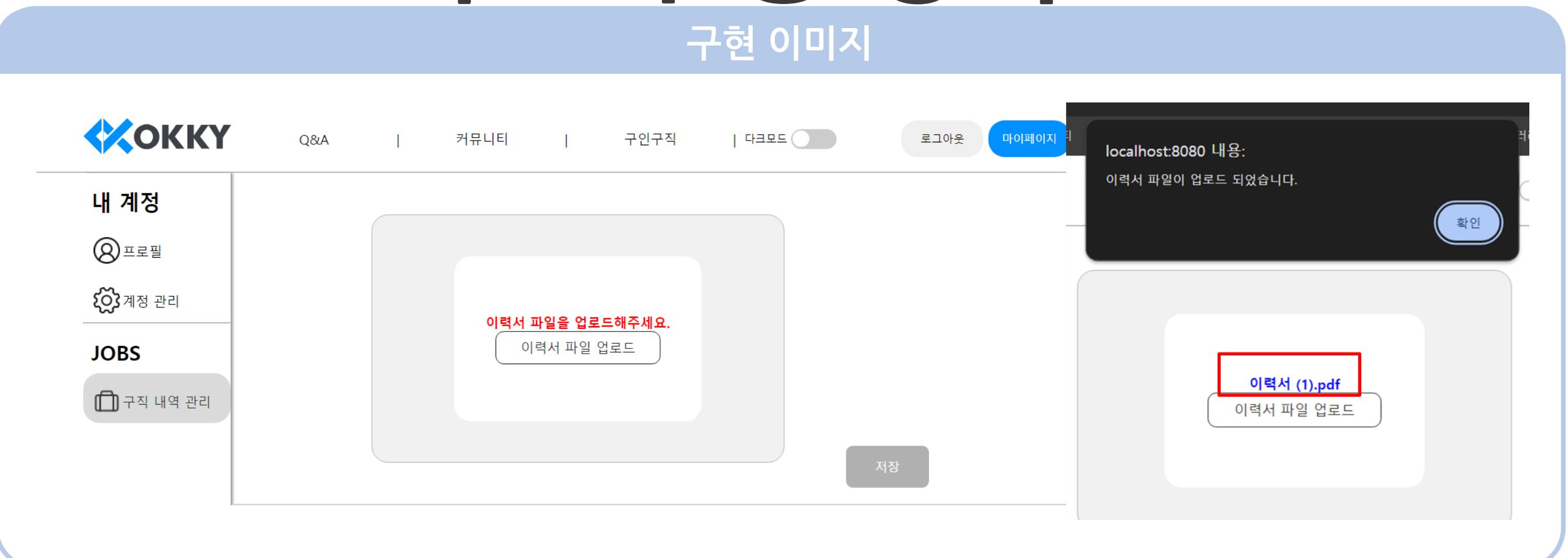
주요 컨텐츠

일반 회원
-마이페이지
(구직 내역 관리)
담당 - 전주연

기능 및 적용 기술

- 개인 회원 로그인 시 구직 내역관리 클릭 시 나타나는 페이지
- Pdf의 이력서 파일 업로드 가능 구현
- 업로드 전에는 업로드 하라는 문구, 업로드 후에는 업로드 된 파일명 표시

구현 이미지



주요 코드

```
<div id="outer">
    <div id="container1">
        <form id="resumeRegiFrm" name="resumeRegiFrm" action="regiResume.do" method="post" enctype="multipart/form-data">
            <div id="innerContainer">
                <div id="resumeBox">
                    <div id="resumeBoxBG">
                        <c:if test="${sessionScope.orgCompanyFile eq null}">
                            <p id="fileMent" style="color: red; font-weight:bold;">이력서 파일을 업로드해주세요.</p>
                        </c:if>
                        <c:if test="${not empty sessionScope.orgCompanyFile}">
                            <p id="fileMent" style="color: blue; font-weight:bold;">${sessionScope.orgCompanyFile}</p>
                        </c:if>
                        <label for="fileUpload">
                            <div id="filebox">이력서 파일 업로드</div>
                        </label>
                        <input type="file" id="fileUpload" name="file" accept=".pdf">
                    </div>
                </div>
                <div class="saveBtnArea">
                    <input type="submit" id="saveBtn" name="saveBtn" value="저장" disabled>
                </div>
            </div>
        </form>
    </div>
```

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    int result = 0;
    HttpSession session = req.getSession();
    Member member = session.getAttribute("memberId") == null ? "" : session.getAttribute("memberId").toString();
    MemberDTO dto = new MemberDTO();
    MemberDAO dao = new MemberDAO();

    String saveDir = req.getServletContext().getRealPath("resumeFiles");
    String saveDir2 = "D:\\java\\workspace\\okkyFile";
    String orgFileName = "";

    try {
        orgFileName = FileUtil2.uploadFile(req, saveDir);
    } catch (Exception e) {
        e.printStackTrace();
        req.setAttribute("uploadResumeError", "파일 업로드 에러 발생");
        req.getRequestDispatcher("./applyList.jsp").forward(req, resp);
    }

    if(orgFileName != "") {
        String saveFileName = FileUtil2.renameFile(req, saveDir, orgFileName);
        dto.setOrgCompanyFile(saveFileName);
        dto.setSaveCompanyFile(saveFileName);
        dto.setMemberId(memberId);
        req.setAttribute("orgFileName", orgFileName);
        System.out.println(orgFileName);
    }
    result = dao.registerResume(dto);
    dao.close();
    if(result > 0) {
        session.setAttribute("orgCompanyFile", orgFileName);
        resp.sendRedirect("./applyList.do");
    } else {
        req.getRequestDispatcher("./applyList.jsp").forward(req, resp);
    }
}
```

04

프로젝트 수행 경과

주요 컨텐츠

일반 회원
-마이페이지
(구직 내역 관리)

담당 - 전주연

기능 및 적용 기술

- 개인 회원 로그인 시 구직 내역관리 클릭 시 나타나는 페이지
- 개인회원이 지원한 공고목록을 볼 수 있음
- 공고 지원 전에는 목록이 없음 표시, 공고 지원 후에는 공고의 번호와 제목, 마감일, 공고페이지로 넘어갈 수 있는 버튼 확인 가능

구현 이미지

공고번호	지원 공고 제목	마감일	공고조회
지원한 공고 목록이 없습니다.			

공고번호	지원 공고 제목	마감일	공고조회
9	S사 SCM 고도화	2024-04-30	<button>조회</button>

주요 코드

```
<div class="divLine"></div>
<div id="container2">
  <div id="joblist">
    <table id="listTbl">
      <thead>
        <tr>
          <th>공고번호</th>
          <th>지원 공고 제목</th>
          <th>마감일</th>
          <th>공고조회</th>
        </tr>
      </thead>
      <tbody>
        <c:choose>
          <c:when test="${not empty applyList }">
            <c:forEach var="list" items="${applyList }" varStatus="loop">
              <tr>
                <td>${list.recruitIdx }</td>
                <td>${list.recruitTitle}</td>
                <td>${list.dueDate}</td>
                <td class="lastCol"><button class="linkBtn" name="linkBtn" onclick="location.href = '/o
                  </td>
                </tr>
              </c:forEach>
            </c:when>
            <c:otherwise>
              <tr>
                <td colspan="4" style="text-align: center;">지원한 공고 목록이 없습니다.</td>
              </tr>
            </c:otherwise>
          </c:choose>
        </tbody>
      </table>
    </div>
```

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
  //jsp 페이지에 전달할 맵 객체 설정
  Map<String, Object> params = new HashMap<>();
  HttpSession session = req.getSession();
  String id = session.getAttribute("memberId") == null ? "" : session.getAttribute("memberId").toString();
  String orgCompanyFile = session.getAttribute("orgCompanyFile") == null ? "" : session.getAttribute("orgCompanyFile").t
  params.put("memberId", id);

  //게시판 테이블 조회
  MemberDAO dao = new MemberDAO();
  List<ApplyTableDTO> applyList = dao.applyList(params);
  dao.close();
  req.setAttribute("applyList", applyList);
  req.getRequestDispatcher("./applyList.jsp").forward(req, resp);
}
```

04

프로젝트 수행 경과

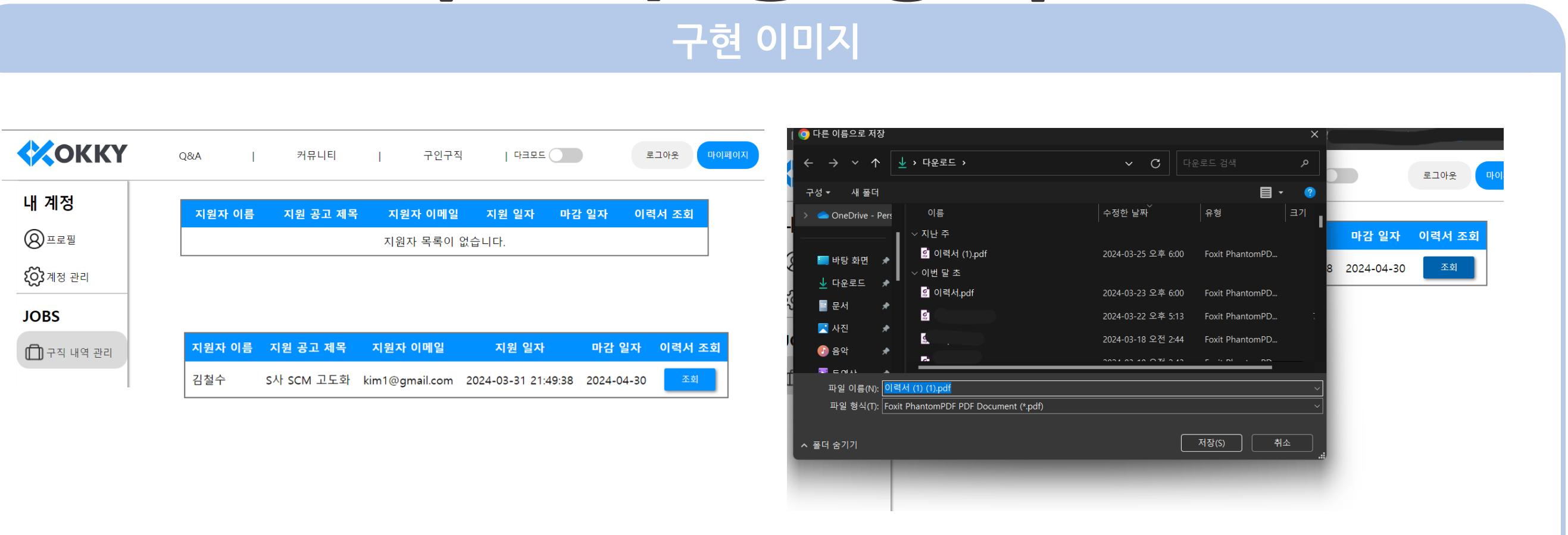
주요 컨텐츠

기업 회원
-마이페이지
(구직 내역 관리)
담당 - 전주연

기능 및 적용 기술

- 기업 회원이 구직 내역 관리를 클릭했을 때 확인할 수 있는 페이지
- 기업 회원이 올린 공고에 지원한 지원자 목록 모두 확인 가능
- 이력서 조회 클릭 시 지원자가 입력한 이력서 파일 다운 가능
- 쿼리문 작성 시 오류 발생으로 결과값이 이상하여 다시 쿼리문 작성 후, 오류를 수정했고 코드에 반영했음

구현 이미지



주요 코드

```
<div id="outer">
  <div id="container1">
    <div id="jobList">
      <table id="listTbl" >
        <thead>
          <tr>
            <th>지원자 이름</th>
            <th>지원 공고 제목</th>
            <th>지원자 이메일</th>
            <th>지원 일자</th>
            <th>마감 일자</th>
            <th>이력서 조회</th>
          </tr>
        </thead>
        <tbody>
          <c:choose>
            <c:when test="${not empty applicantList }">
              <c:forEach var="list" items="${applicantList }" varStatus="loop">
                <tr>
                  <td>${list.name}</td>
                  <td>${list.recruitTitle}</td>
                  <td>${list.email}</td>
                  <td>${list.regDate}</td>
                  <td>${list.dueDate}</td>
                  <td style="display:none;">${list.orgCompanyFile}</td>
                  <td style="display:none;">${list.saveCompanyFile}</td>
                  <td class="lastCol"><button class="linkBtn" name="linkBtn" onclick="location.href='<%=request.getContextPath()%>/jobDetail?idx=${list.idx}'>보기</button></td>
                </tr>
              </c:forEach>
            </c:when>
            <c:otherwise>
              <br>
              <td colspan="6" style="text-align: center;">지원자 목록이 없습니다.</td>
            </c:otherwise>
          </c:choose>
        </tbody>
      </table>
    </div>
  </div>

```

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    //jsp 페이지에 전달할 맵 객체 설정
    Map<String, Object> params = new HashMap<>();
    HttpSession session = req.getSession();
    String id = session.getAttribute("memberId") == null ? "" : session.getAttribute("memberId").toString();
    params.put("memberId", id);

    //게시판 테이블 조회
    MemberDAO dao = new MemberDAO();

    List<ApplyTableDTO> applicantList = dao.applicantList(params);
    dao.close();

    req.setAttribute("applicantList", applicantList);
    req.getRequestDispatcher("./applicantList.jsp").forward(req, resp);
}
```

04

프로젝트 수행 경과

주요 컨텐츠

회원/비회원 공통

- 메인 페이지(1)
(게시글 리스트)
- 담당 - 류림

기능 및 적용 기술

- TOP WRITER : 글 등록을 많이 한 작성자 아이디 순으로 나타남.
- Q&A 와 Community : 최신순으로 Q&A 또는 Community 게시판의 게시글이 나타나며 최대 5개 나타남. 각 게시글 클릭 시 해당 게시글의 상세 페이지로 이동
- 작성자 아이디, 작성날짜, 읽은 횟수, 추천수도 같이 보여줌
- Weekly Best : 인기순부터 Q&A와 Community 게시판의 게시글이 나타나며 최대 5개 나타남. (추천 수 기준으로 나오고 동률 시 조회수 순으로 출력)
- 공지사항 : 최신순부터 Community 게시판의 공지사항 카테고리의 게시글이 나타나며 최대 5개 나타남. 각 게시글 클릭 시 해당 게시글의 상세 페이지로 이동.
- 스크롤을 내리면 각 페이지로 이동하는 네비게이션 바 생성

구현 이미지

Top writers

user2	12 write
admin	9 write
user9	6 write
user3	6 write
user4	3 write

Q&A

user20	2024-03-25	42 read	14 likes
백엔드 알림 서비스에 대해...			
user15	2024-03-25	2 read	0 likes
학점 2점대 졸업예정 전공자 질문이 있습니다.			
user14	2024-03-25	5 read	0 likes
gradle build 이슈 (╯°□°)╯︵ ┻━┻			
user2	2024-03-25	3 read	0 likes
안녕하세요 질문이 있습니다!!!! 많은 분들 댓글 부탁드립니다. ㅠ			
user2	2024-03-25	1 read	0 likes
타입이 number인 input에 값을 설정하지 않을때 데이터바인딩			

커뮤니티

user20	2024-03-25	10 read	2 likes
자혹될 때마다 지방이 그립네요			
user15	2024-03-25	1 read	0 likes
개발하면서 영어실력이 늘고 있네요			
user14	2024-03-25	0 read	0 likes
인공지능 관련 서적 추천받고 싶습니다!			
user14	2024-03-25	2 read	0 likes
자주 쓰는 ai 투표			
user3	2024-03-25	11 read	0 likes
이거 아마 사무직의 대부분을 AI가 차지하지 않을까 싶습니다...			

WEEKLY BEST

user4	2024-03-22	47 read	18 likes
community제목			
user9	2024-03-22	54 read	17 likes
제목113			
user20	2024-03-25	42 read	14 likes
백엔드 알림 서비스에 대해...			
user2	2024-03-24	29 read	12 likes
dsdf			
user2	2024-03-22	29 read	11 likes
제목4			

공지사항

admin	2024-03-25	6 read	0 likes
[중요 알림] OKKY v3.1 릴리즈			
admin	2024-03-25	10 read	0 likes
[리뷰이벤트-당첨자발표] 이지스퍼블리싱 Do it! 점프 투 자바			
admin	2024-03-25	3 read	0 likes
[OKKY] 서비스 작업으로 인한 다운타임 공지			
admin	2024-03-25	1 read	0 likes
[설문조사-종료] 취업 준비생 대상 교육 수료 설문조사			
admin	2024-03-25	0 read	0 likes
[공지사항] OKKY 회원&게시물 관리 정책			

Q&A

커뮤니티

구인구직

04

프로젝트 수행 경과

주요 컨텐츠

회원/비회원 공통
- 메인 페이지(2)
(Top Writers)
담당 - 류림

기능 및 적용 기술

- Top Writers : qna의 테이블과 community의 테이블을 union all로 묶어 작성순으로 내림차순 정렬 후 5개만 보여 줌

메인 페이지의 Controller

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    Map<String, Object> qnaParams = new HashMap<>();
    Map<String, Object> communityParams = new HashMap<>();
    Map<String, Object> mainLikeParams = new HashMap<>();
    Map<String, Object> mainWriteParams = new HashMap<>();

    QnDAO dao = new QnDAO();
    List<QnDTO> qnaList = dao.mainList(qnaParams);
    dao.close();

    request.setAttribute("qnaList", qnaList);

    CommunityDAO dao2 = new CommunityDAO();
    List<CommunityDTO> communityList = dao2.mainList(communityParams);
    dao2.close();

    request.setAttribute("communityList", communityList);

    CommunityDAO notice = new CommunityDAO();
    List<CommunityDTO> noticeList = notice.mainNoticeList(communityParams);
    notice.close();

    request.setAttribute("noticeList", noticeList);

    MainDAO like = new MainDAO();
    List<MainDTO> likeList = like.likeList(mainLikeParams);
    like.close();

    request.setAttribute("likeList", likeList);

    MainDAO write = new MainDAO();
    List<MainDTO> writeList = write.writeList(mainWriteParams);
    write.close();

    request.setAttribute("writeList", writeList);

    request.getRequestDispatcher("./main.jsp").forward(request, response);
}
```

주요 코드

Top writers의 DAO

```
public List<MainDTO> writeList(Map<String, Object> map) {
    List<MainDTO> list = new Vector<>();

    String sql = "select total.memberId, count(*) as writeCnt from"
        + "("
        + " select q.qnaIdx, m.memberId"
        + " from member m"
        + " inner join qna q on m.memberIdx = q.memberIdx"
        + " union all"
        + " select c.communityIdx, m.memberId"
        + " from member m"
        + " inner join community c on m.memberIdx = c.memberIdx"
        + " ) as total"
        + " group by total.memberId"
        + " order by writeCnt desc"
        + " limit 5";

    try {
        psmt = conn.prepareStatement(sql);
        rs = psmt.executeQuery();

        while (rs.next()) {
            MainDTO dto = new MainDTO();
            dto.setMemberId(rs.getString("memberId"));
            dto.setWriteCnt(rs.getInt("writeCnt"));

            list.add(dto);
        }
    } catch (Exception e) {
        System.out.println("메인 작성자 좋아요 수 조회 실패!");
        e.printStackTrace();
    }
    return list;
}
```

메인 페이지의 Top writers의 jsp

```
<div class="side" id="Lside">
    <div id="top_title"><span><h2>Top writers</h2></span></div>
    <div><hr></div>
    <div>
        <ul>
            <c:choose>
                <c:when test="${not empty writeList }">
                    <c:forEach var="list" items="${writeList }" varStatus="loop">
                        <li>
                            <div class="top_writer">
                                <div>${list.memberId }</div>
                                <div class="smallBlue">${list.writeCnt } write</div>
                            </div>
                        </li>
                    </c:forEach>
                </c:when>
                <c:otherwise>
                    등록된 게시글이 없습니다.
                </c:otherwise>
            </c:choose>
        </ul>
    </div>
</div>
```

04

프로젝트 수행 경과

주요 컨텐츠

회원/비회원 공통
- 메인 페이지(3)
(Q&A)
담당 - 류림

기능 및 적용 기술

- Q&A : qna 테이블에서 최신순으로 내림차순 정렬 후 5개만 보여줌

메인 페이지의 Q&A JSP

```
<div class="content">
  <a href="/okky/board/qna/qna.do">
    <div class="info">
      <span>Q&A</span>
      
    </div>
  </a>
  <div>
    <ul>

      <c:choose>
        <c:when test="${not empty qnaList}">
          <c:forEach var="list" items="${qnaList}" varStatus="loop">
            <li>
              <div class="writer">
                <div>${list.memberId}</div>
                <div>${list.regDate}</div>
                <div>${list.readCnt} read</div>
                <div>${list.pageLike} likes</div>
              </div>
              <div class="content_title">
                <a href="/okky/board/qna/viewQnA.do?qnaIdx=${list.qnaIdx}&${list.title}">${list.title}</a>
              </div>
            </li>
            <li><hr></li>
          </c:forEach>
        </c:when>
        <c:otherwise>
          <p>등록된 게시물이 없습니다.</p>
        </c:otherwise>
      </c:choose>
    </ul>
  </div>
</div>
```

주요 코드

메인 페이지의 Q&A DAO

```
public List<QnADTO> mainList(Map<String, Object> map) {
  List<QnADTO> list = new Vector<QnADTO>();

  StringBuilde sb = new StringBuilde();
  sb.append("SELECT q.title, q.content, q.regDate, m.memberId, q.qnaIdx, q.memberIdx");
  sb.append(", q.modifyDate, q.skill1, q.skill2, q.skill3, q.pageLike, q.pageDislike, q.category, q.readCnt");
  sb.append(" FROM qna q inner join member m");
  sb.append(" ON q.memberIdx = m.memberIdx");
  sb.append(" ORDER BY qnaIdx DESC");
  sb.append(" LIMIT 5");

  try {
    psmt = conn.prepareStatement(sb.toString());
    rs = psmt.executeQuery();

    while (rs.next()) {
      OnADTO dto = new OnADTO();
      dto.setQnaIdx(rs.getInt("q.qnaIdx"));
      dto.setTitle(rs.getString("q.title"));
      dto.setContent(rs.getString("q.content").substring(0, 10).concat(" ..."));
      dto.setRegDate(rs.getDate("q.regDate"));
      dto.setModifyDate(rs.getDate("q.modifyDate"));
      dto.setPageLike(rs.getInt("q.pageLike"));
      dto.setPageDislike(rs.getInt("q.pageDislike"));
      dto.setCategory(rs.getString("q.category"));
      dto.setReadCnt(rs.getInt("q.readCnt"));
      dto.setMemberIdx(rs.getInt("q.memberIdx"));
      dto.setMemberId(rs.getString("m.memberId"));
      dto.setSkill1(rs.getString("q.skill1"));
      dto.setSkill2(rs.getString("q.skill2"));
      dto.setSkill3(rs.getString("q.skill3"));

      list.add(dto);
    }
  } catch (Exception e) {
    System.out.println("메인 qna 게시글 조회 실패!");
  }
}

return list;
```

04

프로젝트 수행 경과

주요 컨텐츠

회원/비회원 공통
- 메인 페이지(4)
(Community)
담당 - 류림

기능 및 적용 기술

- Community : community 테이블에서 최신순으로 내림차순 정렬 후 5개만 보여줌

주요 코드

메인 페이지의 Community JSP

```
<div class="content">
    <a href="/okky/board/commu/commu.do">
        <div class="info">
            <span>커뮤니티</span>
            
        </div>
    </a>
    <div>
        <ul>
            <c:choose>
                <c:when test="${not empty communityList}">
                    <c:forEach var="list" items="${communityList}" varStatus="loop">
                        <li>
                            <div class="writer">
                                <div>${list.memberId }</div>
                                <div></div>
                                <div>${list.regDate }</div>
                                <div></div>
                                <div>${list.readCnt } read</div>
                                <div></div>
                                <div>${list.pageLike } likes</div>
                            </div>
                            <div class="content_title">
                                <a href="/okky/board/commu/viewCommu.do?communityIdx=${list.communityIdx}">${list.title }</a>
                            </div>
                        </li>
                        <li><hr></li>
                    </c:forEach>
                </c:when>
                <c:otherwise>
                    <p> 등록된 게시물이 없습니다. </p>
                </c:otherwise>
            </c:choose>
        </ul>
    </div>
```

메인 페이지의 Community DAO

```
public List<CommunityDTO> mainList(Map<String, Object> map) {
    List<CommunityDTO> list = new Vector<>();
    StringBuilder sb = new StringBuilder();
    sb.append("SELECT c.title, c.content, c.regDate, m.memberId, c.communityIdx, c.memberIdx");
    sb.append(" FROM community c inner join member m");
    sb.append(" ON c.memberIdx = m.memberIdx");
    sb.append(" ORDER BY c.communityIdx DESC");
    sb.append(" LIMIT 5");
    try {
        psmt = conn.prepareStatement(sb.toString());
        rs = psmt.executeQuery();
        while (rs.next()) {
            CommunityDTO dto = new CommunityDTO();
            dto.setCommunityIdx(rs.getInt("c.communityIdx"));
            dto.setTitle(rs.getString("c.title"));
            dto.setContent(rs.getString("c.content").substring(0,10).concat(" ..."));
            dto.setRegDate(rs.getDate("c.regDate"));
            dto.setModifyDate(rs.getDate("c.modifyDate"));
            dto.setPageLike(rs.getInt("c.pageLike"));
            dto.setPageDislike(rs.getInt("c.pageDislike"));
            dto.setCategory(rs.getString("c.category"));
            dto.setReadCn(rs.getInt("c.readCn"));
            dto.setMemberIdx(rs.getInt("c.memberIdx"));
            dto.setMemberId(rs.getString("m.memberId"));

            dto.setSkill1(rs.getString("c.skill1"));
            dto.setSkill2(rs.getString("c.skill2"));
            dto.setSkill3(rs.getString("c.skill3"));
        }
        list.add(dto);
    } catch (Exception e) {
        System.out.println("메인 community 게시글 조회 실패!");
    }
}
return list;
```

04

프로젝트 수행 경과

주요 컨텐츠

회원/비회원 공통
- 메인 페이지(5)
(Weekly Best)
담당 - 류림

기능 및 적용 기술

- Weekly Best : qna의 테이블과 community의 테이블을 union all로 묶어 추천순, 조회 순으로 내림차순 정렬 후 5개만 보여줌

주요 코드

Weekly Best의 JSP

```
<div class="content">
    <div>
        <div class="info">
            <span>WEEKLY BEST</span>
            
        </div>
    </div>
    <ul>
        <c:choose>
            <c:when test="${not empty likeList}">
                <c:forEach var="list" items="${likeList}" varStatus="loop">
                    <li>
                        <div class="writer">
                            <div>${list.memberId}</div>
                            <div></div>
                            <div>${list.regDate}</div>
                            <div></div>
                            <div>${list.readCnt} read</div>
                            <div></div>
                            <div>${list.pageLike} likes</div>
                        </div>
                        <div class="content_title">
                            <c:if test="${list.types eq 'QnA'}" > <a href="/okky/board/qna/viewQnA.do?qnaIdx=${list.idx}">${list.title} </a> </c:if>
                            <c:if test="${list.types eq 'Community'}" > <a href="/okky/board/commu/viewCommu.do?communityIdx=${list.idx}">${list.title} </a> </c:if>
                        </div>
                    </li>
                </c:forEach>
            <c:when>
                <c:otherwise>
                    <p>등록된 게시물이 없습니다.</p>
                </c:otherwise>
            </c:otherwise>
        </c:choose>
    </ul>
</div>
```

Weekly Best의 DAO

```
// weekly best
public List<MainDTO> likeList(Map<String, Object> map) {
    List<MainDTO> list = new Vector<>();

    String sql = "select title, regDate, memberId, pageLike, readCnt, types, idx"
        + " from"
        + "("
        + " select m.memberId, q.title, q.regDate, q.pageLike, q.readCnt, 'QnA' as types, qnaIdx as idx "
        + " from qna q"
        + " inner join member m on m.memberIdx = q.memberIdx"
        + " union all"
        + " select m.memberId, c.title, c.regDate, c.pageLike, c.readCnt, 'Community' as types, communityIdx as idx"
        + " from community c"
        + " inner join member m on m.memberIdx = c.memberIdx"
        + " ) as total"
        + " order by pageLike desc, readCnt desc"
        + " LIMIT 5";

    try {
        psmt = conn.prepareStatement(sql);
        rs = psmt.executeQuery();

        while (rs.next()) {
            MainDTO dto = new MainDTO();
            dto.setTitle(rs.getString("title"));
            dto.setRegDate(rs.getDate("regDate"));
            dto.setMemberId(rs.getString("memberId"));
            dto.setPageLike(rs.getInt("pageLike"));
            dto.setReadCnt(rs.getInt("readCnt"));
            dto.setTypes(rs.getString("types"));
            dto.setIdx(rs.getInt("idx"));

            list.add(dto);
        }
    } catch(Exception e) {
        System.out.println("메인 weekly best 게시글 조회 실패!");
    }
    return list;
}
```

04

프로젝트 수행 경과

주요 컨텐츠

회원/비회원 공통
- 메인 페이지(6)
(공지사항)
담당 - 류림

기능 및 적용 기술

- 공지사항 : community의 테이블에서 [공지사항] 카테고리의 글을 최신순으로 내림차순하여 5개만 보여줌

주요 코드

공지사항의 JSP

```
<div class="content">
    <div>
        <a href="/okky/board/commu/commu.do?category_3=category_3">
            <div class="info">
                <span>공지사항</span>
                
            </div>
        </a>
    </div>
    <div>
        <ul>
            <c:choose>
                <c:when test="${not empty noticeList}">
                    <c:forEach var="list" items="${noticeList}" varStatus="loop">
                        <li>
                            <div class="writer">
                                <div>${list.memberId}</div>
                                <div></div>
                                <div>${list.regDate}</div>
                                <div></div>
                                <div>${list.readCnt} read</div>
                                <div></div>
                                <div>${list.pageLike} likes</div>
                                </div>
                                <div class="content_title">
                                    <a href="/okky/board/commu/viewCommu.do?communityIdx=${list.communityIdx}&${list.title}">${list.title}</a>
                                </div>
                            </li>
                            <li><hr></li>
                        </c:forEach>
                    </c:when>
                    <c:otherwise>
                        <p>등록된 게시물이 없습니다.</p>
                    </c:otherwise>
                </c:choose>
            </ul>
        </div>
    </div>
```

공지사항의 DAO

```
public List<CommunityDTO> mainNoticeList(Map<String, Object> map) {
    List<CommunityDTO> list = new Vector<>();

    StringBuider sb = new StringBuider();
    sb.append("SELECT c.title, c.content, c.regDate, m.memberId, c.communityIdx, c.memberIdx");
    sb.append(", c.modifyDate, c.skill1, c.skill2, c.skill3, c.pageLike, c.pageDislike, c.category, c.readCnt");
    sb.append(" FROM community c inner join member m");
    sb.append(" ON c.memberIdx = m.memberIdx");
    sb.append(" WHERE category = '공지사항'");
    sb.append(" ORDER BY c.communityIdx DESC");
    sb.append(" LIMIT 5");

    try {
        psmt = conn.prepareStatement(sb.toString());
        rs = psmt.executeQuery();

        while (rs.next()) {
            CommunityDTO dto = new CommunityDTO();
            dto.setCommunityIdx(rs.getInt("c.communityIdx"));
            dto.setTitle(rs.getString("c.title"));
            dto.setContent(rs.getString("c.content").substring(0,10).concat(" ..."));
            dto.setRegDate(rs.getDate("c.regDate"));
            dto.setModifyDate(rs.getDate("c.modifyDate"));
            dto.setPageLike(rs.getInt("c.pageLike"));
            dto.setPageDislike(rs.getInt("c.pageDislike"));
            dto.setCategory(rs.getString("c.category"));
            dto.setReadCnt(rs.getInt("c.readCnt"));
            dto.setMemberIdx(rs.getInt("c.memberIdx"));
            dto.setMemberId(rs.getString("m.memberId"));

            dto.setSkill1(rs.getString("c.skill1"));
            dto.setSkill2(rs.getString("c.skill2"));
            dto.setSkill3(rs.getString("c.skill3"));

            list.add(dto);
        }
    } catch (Exception e) {
        System.out.println("메인 공지사항 게시글 조회 실패!");
    }
}

return list;
```

04

프로젝트 수행 경과

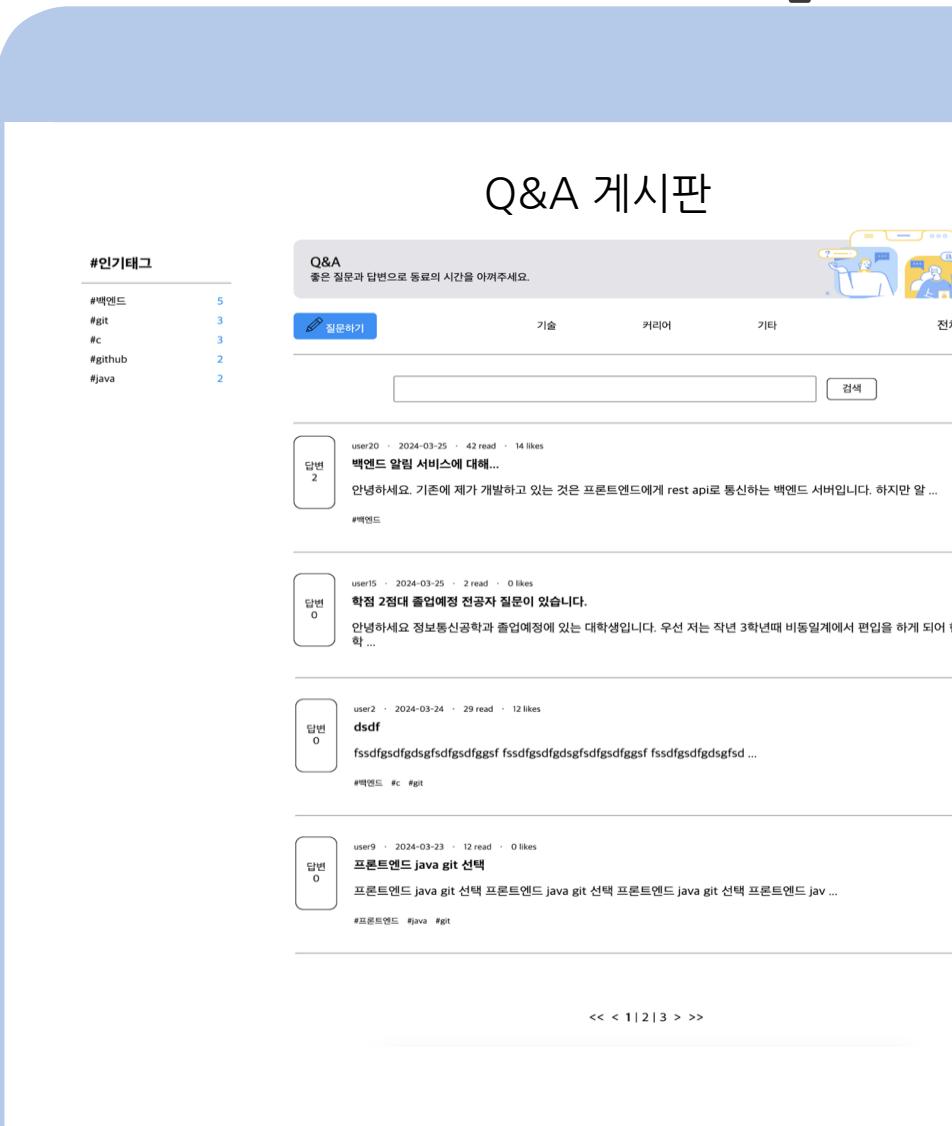
주요 컨텐츠

회원/비회원 공통
 - Q&A 게시판
 /Community 게시판
 (게시글 리스트)
 담당 - 류림

기능 및 적용 기술

- 인기태그 상위 5개 보여줌(Q&A 게시판)
- 추천글 상위 5개 보여줌(Community 게시판)
- 글 작성 기능(Q&A, Community 공통) - 회원만 가능 -> 알럿 확인 후 로그인 페이지로 이동
- 카테고리별 조회, 전체 글 보기, 정렬 기능(최신순, 추천순, 답변순, 조회순) (Q&A, Community 공통)
- 제목의 키워드로 검색하는 기능(Q&A, Community 공통)
- 현재 보여지는 페이지 번호 와 전체 페이지의 수를 나타내는 기능(Q&A, Community 공통)
- 페이지의 맨 위로 올라가는 화살표 버튼(Q&A, Community 공통)
- 페이지 기능 (10개씩 게시글이 보여짐) (Q&A, Community 공통)

구현 이미지





비회원이 질문하기/작성하기 버튼을 눌렀을 시

10.41.2.121:8080 내용:
로그인 후 접근이 가능합니다.

주요 코드

```

document.querySelector("#btn_regist").addEventListener("click", function(e) {
    let id_ck = "${(sessionScope.memberId != null)? sessionScope.memberId : 0}";

    if (id_ck == "0") {
        alert("로그인 후 접근이 가능합니다.");
        window.location.href = "/okky/member/login.do";
    } else {
        window.location.href = "writeCommu.do";
    }
}, false);

<c:choose>
    <c:when test="${not empty qnaList}">
        <c:forEach var="list" items="${qnaList}" varStatus="loop">
            <li>
                <button onclick="location.href='viewQnA.do?qnaIdx=${list.qnaIdx}'">
                    답변
                </button>
                <div>
                    ${list.answerIdx}
                </div>
            </li>
        </c:forEach>
    <c:otherwise>
        <div>
            <div><img alt="empty list icon" /></div>
            <div>등록된 글이 없습니다.</div>
        </div>
    </c:otherwise>
</c:choose>

```

04

프로젝트 수행 경과

주요 컨텐츠

회원/비회원 공통
- Q&A 게시판
/Community 게시판
(검색 / 정렬)
담당 - 류림

기능 및 적용 기술

- 검색어 입력 필수체크.
- 검색 시 where 조건절을 주어
검색어의 키워드를 %키워드
% 로 조회
- 카테고리별 조회할 때 where
조건절을 주어 해당 카테고리
의 value값에 따라 %키워드%
로 조회
- 정렬 시 orderby를 각 value별
로 다르게 주어 보여줌

주요 코드

검색어 필수 체크

```
document.querySelector("#btn_search").addEventListener("click", function(e) {  
  
    e.preventDefault();  
    e.stopPropagation();  
  
    const search_word = document.querySelector("#search_word");  
    if (search_word.value.length < 1 || search_word.value == "" || search_word.value == null) {  
        alert("검색어 입력하세요");  
        search_word.focus();  
        return false;  
    }  
  
    const frm = document.querySelector("#frmSearch");  
    frm.submit();  
  
    return;  
}, false);
```

```
StringBuilder sb = new StringBuilder();  
sb.append("SELECT");  
sb.append(" (select count(*) from qna_comment as qc where qc.qnaIdx = q.qnaIdx) as cnt,");  
sb.append(" q.title, q.content, q.regDate, m.memberId, q.qnaIdx, q.memberIdx");  
sb.append(", q.modifyDate, q.skill1, q.skill2, q.skill3, q.pageLike, q.pageDislike, q.category, q.readCnt");  
sb.append(" FROM qna q inner join member m");  
sb.append(" ON q.memberIdx = m.memberIdx");  
  
if (map.get("category_1") != null) {  
    sb.append(" WHERE category LIKE '기술'");  
}  
if (map.get("category_2") != null) {  
    sb.append(" WHERE category LIKE '커리어'");  
}  
if (map.get("category_3") != null) {  
    sb.append(" WHERE category LIKE '기타'");  
}  
  
if (map.get("search_word") != null) {  
    sb.append(" WHERE q.title");  
    sb.append(" LIKE '%" + map.get("search_word") + "%'");  
    sb.append(" ORDER BY DESC");  
}  
if (map.get("search_word") == null) {  
    if (map.get("sort") != "") {  
        sb.append(" ORDER BY ");  
        sb.append(map.get("sort"));  
        sb.append(" DESC");  
    }  
    else if (map.get("sort") == "") {  
        sb.append(" ORDER BY q.qnaIdx DESC");  
    }  
}
```

QnA DAO

- Community DAO도 동일하게 동작

04

프로젝트 수행 경과

주요 컨텐츠

회원/비회원 공통
 - Q&A 게시판
/Community 게시판
 (인기태그/추천글)
 담당 - 류림

기능 및 적용 기술

- 인기태그 : union을 사용해 해당 태그가 존재하면 count를 올리는 sql문을 사용하여 구현함
- 추천글 : 최근 일주일간의 추천 수가 높은 게시글의 상위 5개를 보여주며 제목의 길이가 7 이상이면 잘라서 보여줌

주요 코드

Qna의 인기태그 JSP

```
<div class="flex">
  <!-- 왼쪽사이드 -->
  <div class="side" id="Lside">
    <div id="top_tag"><span>#인기태그</span></div>
    <div><hr></div>
    <ul>
      <c:choose>
        <c:when test="${not empty tagList }">
          <c:forEach var="list" items="${tagList }">
            <li>
              <div class="top_tag">
                <div>#${list.tags}</div>
                <div class="smallBlue">${list.count} </div>
              </div>
            </li>
          </c:forEach>
        </c:when>
        <c:otherwise>
          등록된 게시글이 없습니다.
        </c:otherwise>
      </c:choose>
    </ul>
  </div>
```

Community의 추천글 JSP

```
<div class="side" id="Rside">
  <div id="top_tag"><span>추천글</span></div>
  <div><hr></div>
  <ul>
    <c:choose>
      <c:when test="${not empty likeList }">
        <c:forEach var="list" items="${likeList }" varStatus="loop">
          <li>
            <div class="top_tag">
              <div>#${list.title}</div>
              <div><a href="/okky/board/commu/viewCommu.do?communityIdx=${list.communityIdx}&${list.title.substring(0,7).concat("...")}">${list.title}</a></div>
            </div>
            <c:choose>
              <c:when test="#${list.title.length() >= 7 }">
                <a href="/okky/board/commu/viewCommu.do?communityIdx=${list.communityIdx}&${list.title}</a>
              </c:when>
              <c:otherwise>
                <a href="/okky/board/commu/viewCommu.do?communityIdx=${list.communityIdx}&${list.title}</a>
              </c:otherwise>
            </c:choose>
            <div class="smallBlue">${list.pageLike} likes</div>
          </li>
        </c:forEach>
      </c:when>
      <c:otherwise>
        등록된 게시글이 없습니다.
      </c:otherwise>
    </c:choose>
  </ul>
</div>
```

Qna의 인기태그 DAO

```
public List<TagDTO> tagList(Map<String, Object> map) {
  List<TagDTO> list = new Vector<Object>();
  String sql = "select case when skill1='백엔드' then '백엔드' else null end as tags, count(case when skill1='백엔드' then 1 else null end) as 'count' from qna where skill1 is not null "
  + " union "
  + " select case when skill1='프론트엔드' then 1 else '프론트엔드' end as tags, count(case when skill1='프론트엔드' then 1 else null end) as 'count' from qna where skill1 is not null "
  + " union "
  + " select case when skill1='풀스택' then 1 else '풀스택' end as tags, count(case when skill1='풀스택' then 1 else null end) as 'count' from qna where skill1 is not null "
  + " union "
  + " select case when skill2='c' then 'c' else null end as tags, count(case when skill2='c' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill2='java' then 1 else 'java' end as tags, count(case when skill2='java' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill2='c++' then 1 else 'c++' end as tags, count(case when skill2='c++' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill2='c#' then 1 else 'c#' end as tags, count(case when skill2='c#' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill2='db' then 1 else 'db' end as tags, count(case when skill2='db' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill2='python' then 1 else 'python' end as tags, count(case when skill2='python' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill2='javaScript' then 1 else 'javaScript' end as tags, count(case when skill2='javaScript' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill2='html' then 1 else 'html' end as tags, count(case when skill2='html' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill2='css' then 1 else 'css' end as tags, count(case when skill2='css' then 1 else null end) as 'count' from qna where skill2 is not null "
  + " union "
  + " select case when skill3='git' then 'git' else null end as tags, count(case when skill3='git' then 1 else null end) as 'count' from qna where skill3 is not null "
  + " union "
  + " select case when skill3='github' then 1 else 'github' end as tags, count(case when skill3='github' then 1 else null end) as 'count' from qna where skill3 is not null "
  + " order by count desc"
  + " limit 5";
  try {
    psmt = conn.prepareStatement(sql);
    rs = psmt.executeQuery();
    while(rs.next()) {
      TagDTO dto = new TagDTO();
      dto.setTags(rs.getString("tags"));
      dto.setCount(rs.getInt("count"));
      list.add(dto);
    }
  } catch (Exception e) {
    System.out.println("커뮤니티 게시판 태그 조회 오류!");
    e.printStackTrace();
  }
  return list;
}
```

Community의 추천글 DAO

```
public List<CommunityDTO> likeList(Map<String, Object> map) {
  List<CommunityDTO> list = new Vector<Object>();
  String sql = "select communityIdx, title, pageLike from community where regDate between adddate(now(), interval -1 week) and now() order by pageLike desc limit 5";
  try {
    psmt = conn.prepareStatement(sql);
    rs = psmt.executeQuery();
    while (rs.next()) {
      CommunityDTO dto = new CommunityDTO();
      dto.setCommunityIdx(rs.getInt("communityIdx"));
      dto.setTitle(rs.getString("title"));
      dto.setPageLike(rs.getInt("pageLike"));
      list.add(dto);
    }
  } catch (Exception e) {
    System.out.println("community 추천글 조회 실패!");
  }
  return list;
}
```

04

프로젝트 수행 경과

주요 컨텐츠

회원 공통
 - Q&A 게시판
 /Community 게시판
 (글 작성)
 담당 - 류림

기능 및 적용 기술

- 글 작성 페이지에 들어온 회원의 이름을 보여줌
- 등록 버튼 클릭 시 모든 항목 입력하지 않았을 시 입력해달라는 안내문구 display:none에서 display:block으로 교체
- 취소 버튼 클릭시 “정말 초기화겠습니까?”라는 알럿 창

기술 궁금증 해결하기

admin님 지식공유 플랫폼 OKKY에서 최고의 개발자들과 함께 궁금증을 해결하세요.

토Pic

토Pic을 선택해주세요.

제목

제목을 입력해주세요.

태그 - 내용을 대표하는 태그를 선택해 주세요.

선택 1

선택 2

선택 3

구현 이미지

분문
본문을 10자 이상 입력해 주세요.

토Pic
✓ 토Pic을 선택해주세요.
일상
공부
[공지사항]
제목을 입력해주세요.

Community 게시판 글 작성
- 관리자 계정일 시 토픽에 [공지사항] 노출
토Pic
✓ 토Pic을 선택해주세요.
일상
공부

취소 버튼 클릭 시

10.41.2.121:8080 내용:
초기화 시키시겠습니까?

취소 확인

주요 코드

```
let submit = document.querySelector("#submitBtn");
submit.addEventListener("click", (e) => {
  let select = document.querySelector("#category");
  const title = document.querySelector("#title");
  const content = document.querySelector("#contents");
  const titleCheck = document.querySelector("#titleCheck");
  const optionCheck = document.querySelector("#optionCheck");
  const contents = document.querySelector("#content");
  const contentsCheck = document.querySelector("#contentsCheck");

  if (select.value === "") {
    optionCheck.style.display = "block";
    select.focus();
    e.preventDefault();
    return false;
  }
  if (title.value.length < 1) {
    titleCheck.style.display = "block";
    title.focus();
    titleCheck.textContent = "제목은 필수 입력 사항입니다.";
    e.preventDefault();
    return false;
  }
  if (title.value.length > 50) {
    titleCheck.style.display = "block";
    title.focus();
    e.preventDefault();
    return false;
  }
  if (contents.value.length < 10) {
    contentsCheck.style.display = "block";
    contents.focus();
    e.preventDefault();
    return false;
  }
  return true;
});
```

```
document.querySelector("#resetBtn").addEventListener("click", function(e) {
  let confirm_yn = confirm("정말 초기화 시키시겠습니까?");
  if (confirm_yn) {
    return true;
  } else {
    e.preventDefault();
    return false;
  }
});
```

```
public int qnaRegist(OnADTO dto) {
  int result = 0;

  StringBuilder sb = new StringBuilder();
  sb.append("INSERT INTO qna (title, content, category, skill1, skill2, skill3, memberIdx) VALUES(?, ?, ?, ?, ?, ?, ?)");
  try {
    pstmt = conn.prepareStatement(sb.toString());
    pstmt.setString(1, dto.getTitle());
    pstmt.setString(2, dto.getContent());
    pstmt.setString(3, dto.getCategory());
    pstmt.setString(4, dto.getSkill1());
    pstmt.setString(5, dto.getSkill2());
    pstmt.setString(6, dto.getSkill3());
    pstmt.setInt(7, dto.getMemberIdx());

    result = pstmt.executeUpdate();
  } catch (Exception e) {
    System.out.println("게시글 등록 오류!" + e.getMessage());
    e.printStackTrace();
  }
  return result;
}
```

04

프로젝트 수행 경과

주요 컨텐츠

관리자

- Community 게시판
(글 작성)
담당 - 류림

기능 및 적용 기술

- 커뮤니티 페이지 글 작성 시
[공지사항] 카테고리 노출

기술 궁금증 해결하기

admin님 지식공유 플랫폼 OKKY에서 최고의 개발자들과 함께 궁금증을 해결하세요.

토Pic

토Pic을 선택해주세요.

제목

제목을 입력해주세요.

태그 - 내용을 대표하는 태그를 선택해 주세요.

선택 1

선택 2

선택 3

구현 이미지

Community 게시판 글 작성

- 관리자 계정일 시 토픽에 [공지사항] 노출

Community 게시판 글 작성

- 관리자 계정일 시 토픽에 [공지사항] 비노출

주요 코드

```
<input type="hidden" id="memberIdx" name="memberIdx" value="<% session.getAttribute("memberIdx") %>">
<label for="category" class="">토Pic</label><br>
<select id="category" name="category" class="input_area">
    <option value="">토Pic을 선택해주세요.</option>
    <option value="일상">일상</option>
    <option value="공부">공부</option>
    <c:if test="${sessionScope.memberId eq 'admin'}"><option value="공지사항">[공지사항]</option></c:if>
        <!-- <option value="공지사항">공지사항</option> -->
    </select>
    <div id="optionCheck" style="color: red; font-size: x-small; display: none;">토Pic을 하나 이상 선택해 주세요.</div>
</div>
```

04

프로젝트 수행 경과

주요 컨텐츠

회원 공통
 - Q&A 게시판
 /Community 게시판
 (글 조회(상단))
 담당 - 류림

기능 및 적용 기술

- 해당 글을 선택하여 들어갔을 때 나오는 조회 페이지. 조회수가 +1 됨
- 해당 글의 게시판의 이름을 보여줌
- 해당 글의 작성자의 닉네임을 보여줌
- 제목, 작성자ID, 조회수, 작성일, 카테고리와 내용, 좋아요/싫어요 수를 보여줌
- 자신의 글이 아니면 자신의 글만 수정/삭제 가능하다는 알럿창
- 비회원 상태면 수정/삭제 버튼을 눌렀을 시 로그인을 하라는 알럿창 후 로그인 페이지로 이동

Q&A

작성자 : 유저닉네임20

백엔드 알림 서비스에 대해...

작성자 ID : user20

조회수 : 44

작성일 : 2024-03-25

카테고리 : 기술

구현 이미지

10.41.2.121:8080 내용:

로그인 후 접근이 가능합니다.

확인

10.41.2.121:8080 내용:

자신의 글만 수정이 가능합니다.

확인

10.41.2.121:8080 내용:

자신의 글만 삭제가 가능합니다.

확인

주요 코드

```
<div id="main">
  <div>
    <h3>Q&A</h3>
    <p>작성자 : <strong>${params.nickName}</strong></p>
    <div class="contents">
      <h3>제목 : ${params.title}</h3>
      <div>작성자 ID : ${params.memberId}</div>
      <br>
      <div>조회수 : ${params.readCnt}</div>
      <br>
      <div>작성일 : ${params.regDate}</div>
      <br>
      <div>카테고리 : ${params.category}</div>
      <div><button type="button" style="background-color: #009090; color: white;" name="btn_modify" id="btn_modify" value="수정하기"></button>
          <button type="button" style="background-color: #ffff;" name="btn_delete" id="btn_delete" value="삭제하기"><br><br></button>
        </div>
      <div name="" id="" cols="30" rows="10" readonly disabled=${params.content}><textarea></div>
      <div id="tagAndGood">
        <div id="tags">
          <ul>
            <li><c:if test="${not empty params.skill1}"> <div>#${params.skill1}</div></c:if></li>
            <li><c:if test="${not empty params.skill2}"> <div>#${params.skill2}</div></c:if></li>
            <li><c:if test="${not empty params.skill3}"> <div>#${params.skill3}</div></c:if></li>
          </ul>
        </div>
        <div id="goods">
          <ul>
            <li><button id="up" name="up"></button> </li>
            <li>${params.pageLike}</li>
            <li><button id="down" name="down"></button> </li>
            <li>${params.pageDislike}</li>
          </ul>
        </div>
      </div>
    </div>
  </div>
```

```
document.querySelector("#btn_delete").addEventListener("click", function(e) {
  let id_ck = "${(sessionScope.memberId != null)? sessionScope.memberId : 0}";
  if (id_ck == "0") {
    alert("로그인 후 접근이 가능합니다.");
    window.location.href = "/okky/member/login.do";
  } else if ("${sessionScope.memberIdx}" != "${params.memberIdx}") {
    alert("작성자와 글의 주인이 다릅니다.");
    window.location.href = "viewQnA.do?qnaidx=${params.qnaidx}";
  } else {
    let confirm_yn = confirm("현재 글을 삭제하시겠습니까?");
    if (confirm_yn) {
      window.location.href = "deleteQnA.do?qnaidx=${params.qnaidx}&memberId=${(sessionScope.memberId)}";
    }
  }, false);
document.querySelector("#btn_modify").addEventListener("click", function(e) {
  let id_ck = "${(sessionScope.memberId != null)? sessionScope.memberId : 0}";
  if (id_ck == "0") {
    alert("로그인 후 접근이 가능합니다.");
    window.location.href = "/okky/member/login.do";
  } else if ("${sessionScope.memberIdx}" != "${params.memberIdx}") {
    alert("작성자와 글의 주인이 다릅니다.");
    window.location.href = "viewQnA.do?qnaidx=${params.qnaidx}";
  } else {
    let confirm_yn = confirm("현재 글을 수정 하시겠습니까?");
    if (confirm_yn) {
      window.location.href = "modifyQnA.do?qnaidx=${params.qnaidx}";
    }
  }, false);
});
```

```
public OnADTO qnaView (int qnaIdx) {
  OnADTO dto = new OnADTO();
  StringBuilder sb = new StringBuilder();
  sb.append("SELECT");
  sb.append(" * ");
  sb.append(" FROM qna ");
  sb.append(" WHERE qnaIdx = ? ");
  sb.append(" ORDER BY regDate DESC");
  try {
    psmt = conn.prepareStatement(sb.toString());
    psmt.setInt(1, qnaIdx);
    rs = psmt.executeQuery();
    if (rs.next()) {
      dto.setQnaidx(rs.getInt("qnaIdx"));
      dto.setTitle(rs.getString("q.title"));
      dto.setContent(rs.getString("q.content"));
      dto.setRegDate(rs.getDate("q.regDate"));
      dto.setModifyDate(rs.getDate("q.modifyDate"));
      dto.setPageLike(rs.getInt("q.pageLike"));
      dto.setPageDislike(rs.getInt("q.pageDislike"));
      dto.setCategory(rs.getString("q.category"));
      dto.setMemberId(rs.getString("q.memberId"));
      dto.setNickName(rs.getString("q.nickName"));
      dto.setMemberIdx(rs.getInt("q.memberIdx"));
      dto.setSkill1(rs.getString("q.skill1"));
      dto.setSkill2(rs.getString("q.skill2"));
      dto.setSkill3(rs.getString("q.skill3"));
    }
  } catch (Exception e) {
    System.out.println("QnA 게시판 데이터 조회 오류!");
    e.printStackTrace();
  }
  return dto;
}

public void updateReadCnt(int qnaIdx) {
  StringBuilder sb = new StringBuilder();
  sb.append("UPDATE qna SET");
  sb.append(" readCnt = readCnt + 1");
  sb.append(" WHERE qnaIdx = ?");
  try {
    psmt = conn.prepareStatement(sb.toString());
    psmt.setInt(1, qnaIdx);
    psmt.executeUpdate();
  } catch (Exception e) {
    System.out.println("조회 갯수 증가 오류!");
    e.printStackTrace();
  }
}
```

04

프로젝트 수행 경과

주요 컨텐츠

회원 공통

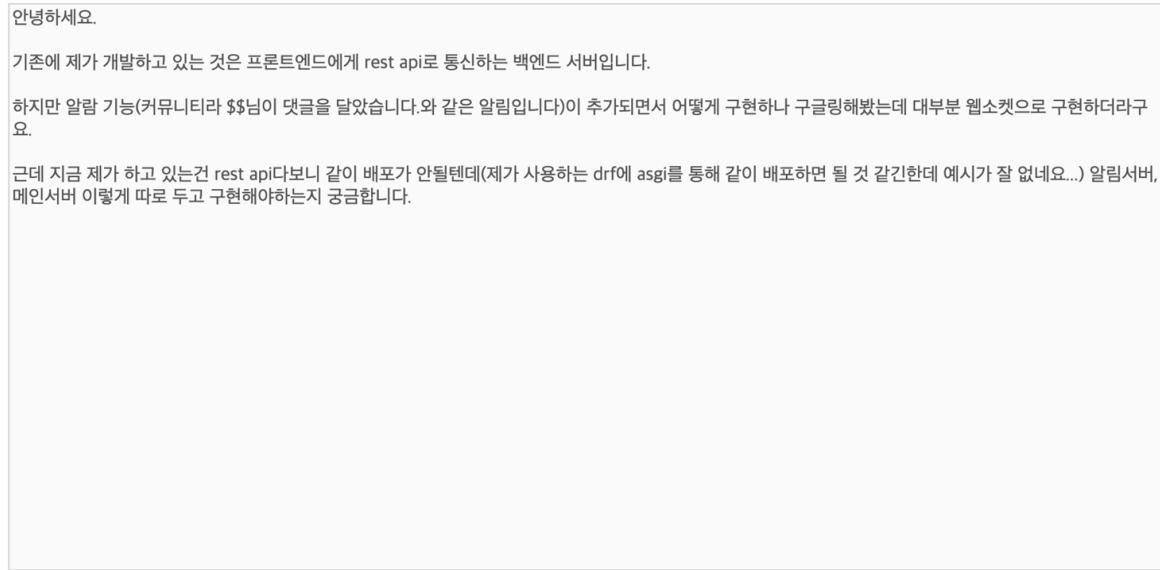
- Q&A 게시판

/Community 게시판
(글 조회(하단))
담당 - 류림

기능 및 적용 기술

- 해당 글의 태그를 보여줌
- 좋아요 / 싫어요 클릭 시
pageLike +1 또는
pageDislike + 1

구현 이미지



#백엔드

14 3

주요 코드

```
public void updatePageLike(int qnaIdx) {  
  
    StringBuilder sb = new StringBuilder();  
  
    sb.append("UPDATE qna SET");  
    sb.append(" pageLike = pageLike + 1");  
    sb.append(" WHERE qnaIdx = ?");  
  
    try {  
        psmt = conn.prepareStatement(sb.toString());  
        psmt.setInt(1, qnaIdx);  
        psmt.executeQuery();  
  
    } catch (Exception e) {  
        System.out.println("좋아요 갯수 증가 오류!");  
        e.printStackTrace();  
    }  
  
}  
  
public void updatePageDislike(int qnaIdx) {  
  
    StringBuilder sb = new StringBuilder();  
  
    sb.append("UPDATE qna SET");  
    sb.append(" pageDislike = pageDislike + 1");  
    sb.append(" WHERE qnaIdx = ?");  
  
    try {  
        psmt = conn.prepareStatement(sb.toString());  
        psmt.setInt(1, qnaIdx);  
        psmt.executeQuery();  
  
    } catch (Exception e) {  
        System.out.println("싫어요 갯수 증가 오류!");  
        e.printStackTrace();  
    }  
}
```

04

프로젝트 수행 경과

주요 컨텐츠

회원 공통
- Q&A 게시판
/Community 게시판
(글 수정)
담당 - 류림

기능 및 적용 기술

- 해당 글의 작성자가 수정하기 버튼을 눌렀을 시 수정 페이지로 이동
- 해당 글의 내용을 불러옴

10.41.2.121:8080 내용:
현재 글을 수정 하시겠습니까?

취소 확인

구현 이미지

기술 궁금증 해결하기

user20님 지식공유 플랫폼 OKKY에서 최고의 개발자들과 함께 궁금증을 해결하세요

토Pic

기술

제목

백엔드 알림 서비스에 대해...

태그 - 내용을 대표하는 태그를 입력해주세요. 태그는 공백 하나로 구분됩니다.

본문

안녕하세요.

기존에 제가 개발하고 있는 것은 프론트엔드에게 rest api로 통신하는 백엔드 서버입니다.

하지만 알람 기능(커뮤니티와 \$님이 댓글을 달았습니다.와 같은 알림입니다)이 추가되면서 어떻게 구현하나 구글링해봤는데 대부분 웹소켓으로 구현하더라구요.

근데 지금 제가 하고 있는 건 rest api다보니 같이 배포가 안될텐데(제가 사용하는 drf에 asgi를 통해 같이 배포하면 될 것 같긴한데 예시가 잘 없네요...) 알림서버, 메인서버 이렇게 따로 두고 구현해야 하는지 궁금합니다.

주요 코드

```
<div>
    <label for="category" class="..._label">
        <select id="category" name="category" class="input_area">
            <option value="">토pic 선택하세요.</option>
            <option value="...일상">일상</option>
            <option value="...공부">공부</option>
            <option value="...공지사항">공지사항</option>
        </select>
    </div>
    <div id="optionCheck" style="color: red; font-size: x-small; display: none;">토pic을 하나 이상 선택해 주세요.</div>
</div>
<div>
    <label for="title" class="..._label">
        <input type="text" id="title" name="title" placeholder="제목을 입력해주세요." class="input_area" value="${params.title}">
    </div>
    <div id="titleCheck" style="color: red; font-size: x-small; display: none;">제목은 50자 이하로 입력해 주세요.</div>
</div>
<div id="selectCon">
    <div>
        <select name="skill1" id="skill1_list">
            <option value="...백엔드">백엔드</option>
            <option value="...프론트엔드">프론트엔드</option>
            <option value="...클라우드">클라우드</option>
        </select>
    </div>
    <div>
        <select name="skill2" id="skill2_list">
            <option value="...java">java</option>
            <option value="...c">c</option>
            <option value="...c++">c++</option>
            <option value="...python">python</option>
            <option value="...html">html</option>
            <option value="...css">css</option>
            <option value="...javascript">javascript</option>
            <option value="...spring">spring</option>
        </select>
    </div>
    <div>
        <select name="skill3" id="skill3_list">
            <option value="...git">git</option>
            <option value="...github">github</option>
        </select>
    </div>
</div>
<div>
    <label for="content" class="..._label">
        <textarea id="content" name="content" cols="100" rows="20" placeholder="본문을 10자 이상 입력해 주세요...-${params.content}"></textarea>
    </div>
    <div>
        <ul>
            <li><input type="button" id="frmBtns" value="취소"></li>
            <li><input type="submit" id="submitBtn" class="frmBsns" value="수정하기"></li>
        </ul>
    </div>
</div>
```

```
public int qnaModify (QnADTO dto) {
    int result = 0;

    StringBuilder sb = new StringBuilder();
    sb.append("UPDATE qna SET title = ?, content = ?, category = ?, skill1 = ?, skill2 = ?, skill3 = ?, modifyDate = now()");
    sb.append(" WHERE qnaIdx = ?");

    try {
        psmt = conn.prepareStatement(sb.toString());
        psmt.setString(1, dto.getTitle());
        psmt.setString(2, dto.getContent());
        psmt.setString(3, dto.getCategory());
        psmt.setLong(4, dto.getSkill1());
        psmt.setLong(5, dto.getSkill2());
        psmt.setLong(6, dto.getSkill3());
        psmt.setInt(7, dto.getQnaIdx());

        result = psmt.executeUpdate();
    } catch (Exception e) {
        System.out.println("게시글 수정 중 오류 발생!" + e.getMessage());
        e.printStackTrace();
    }
    return result;
}
```

04

프로젝트 수행 경과

주요 컨텐츠

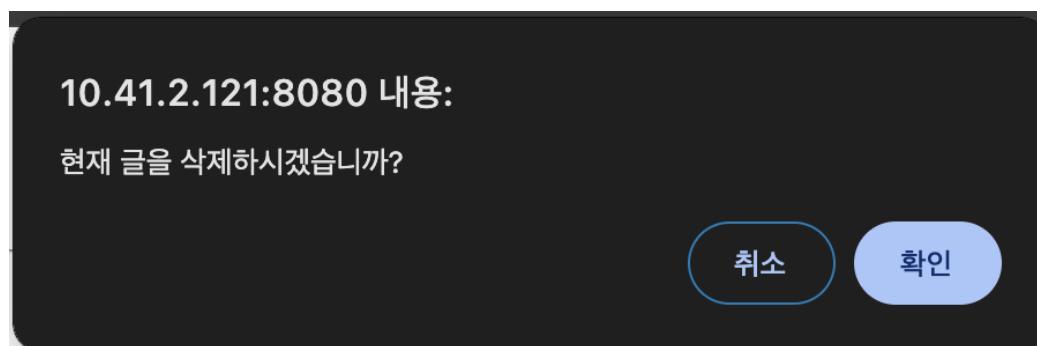
회원 공통

- Q&A 게시판
/Community 게시판
(글 삭제)
담당 - 류림

기능 및 적용 기술

- 해당 글의 작성자가 삭제하기 버튼을 눌렀을 시 삭제 컨트롤러로 이동

구현 이미지



주요 코드

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    int qnaIdx = request.getParameter("qnaIdx") != null ? Integer.parseInt(request.getParameter("qnaIdx")) : 0;

    QnDAO dao = new QnDAO();
    int result = dao.qnaDelete(qnaIdx);

    System.out.println("delete 결과" + result);
    dao.close();

    Map<String, Object> params = new HashMap<>();
    params.put("result", result);

    request.setAttribute("params", params);
    response.sendRedirect("/okky/board/qna/qna.do");
}
```

```
public int qnaDelete(int qnaIdx) {
    int result = 0;

    String sql = "DELETE FROM qna WHERE qnaIdx = ?";
    try {
        psmt = conn.prepareStatement(sql);
        psmt.setInt(1, qnaIdx);

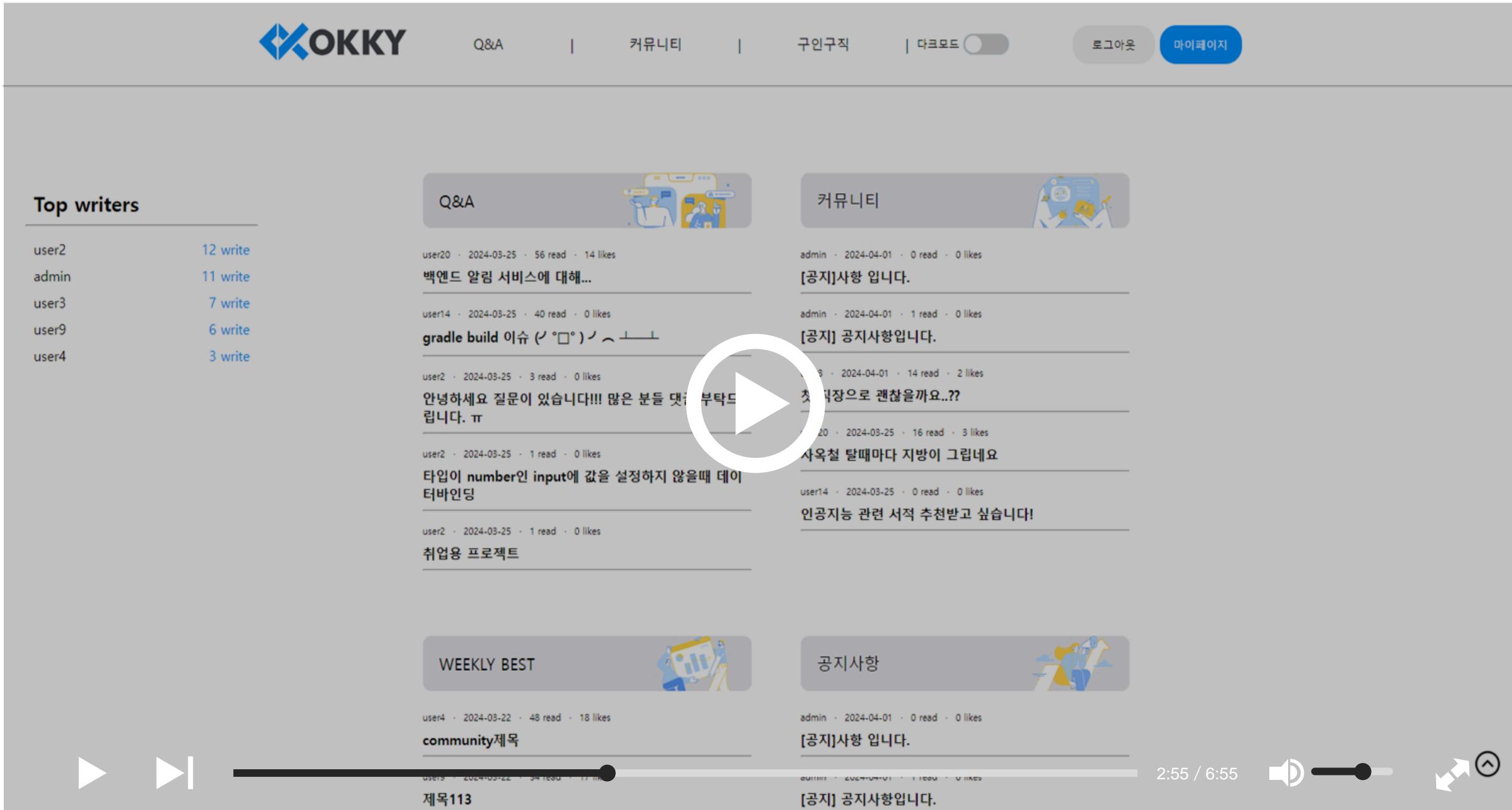
        result = psmt.executeUpdate();
    } catch (Exception e) {
        System.out.println("게시글 삭제 오류!" + e.getMessage());
        e.printStackTrace();
    }
}

return result;
}
```

04

프로젝트 수행 경과

시연 동영상



05 자체 평가 의견

- 공동 평가

완성도 : 9점/10점

아쉬운 점

1. 프로젝트 진행 전 데이터베이스 설계를 완전하게 하지 못한 점
 - 데이터베이스의 중요성을 모르고 프로젝트를 진행하며 데이터베이스를 수정하려 했으나, 나중에 데이터베이스를 수정하려고 하니, 모든 코드도 다 변경해야 했고 더 큰 일이 되어 프로젝트 기간을 맞추지 못하는 불상사가 발생함
 - 이 경험을 바탕으로 다음 프로젝트에서는 데이터베이스 설계부터 꼼꼼히 진행하자는 의견이 나옴
2. CSS를 공통으로 맞추지 못한 점
 - HTML 구조를 통일했으면 UI를 맞출 때 더 편리했을 것임
 - 각자 맡은 페이지의 UI를 잡는 데 많은 시간을 들이게 되어 아쉬움이 남음.

잘한 점

1. 분배
 - 다른 팀에 비해 인원수가 적었지만 각자 한 페이지 씩 통으로 맡아 데이터베이스부터 기능 구현까지 완료함
 - 페이지 분배로 프로젝트의 완성도에 기여한 바가 큼
2. 협동
 - 코드 구현에 어려움이 생겼을 때 같이 해결해주어 완성도를 높임

05 자체 평가 의견

공동 평가

추후 개선점, 보완할 점

1. 데이터 베이스 설계
2. 다크모드 구현
 - 다크모드를 눌렀을 시 다른 페이지로 가면 초기화가 됨
 - 다크모드의 유지 구현
3. 게시판 공지사항 상단 2개 노출
 - 기획서에 명시되어 있으나 구현하지 못함
4. 관리자 권한 확대
 - 현재 관리자 권한은 공지사항 등록밖에 없음
 - 모든 유저의 게시글 삭제 기능과 공고의 삭제, 유저 관리 하는 권한
 - 관리자 페이지 필요

05 자체 평가 의견

개인 평가 - 조현기

완성도 : 8.5점/10점

아쉬운 점

1. 공고 등록 시 선택 가능한 마감일
 - 공고 등록 시 공고 마감일을 현재일 이전으로 선택했을 때도 등록이 가능
2. 인원수
 - 다른팀에 비해 적은 인원수로 진행을 하다보니 다른 팀에 비해 일정을 맞추기가 더 힘들었음
3. 마감기한
 - 마감기한을 지키지 못한 점.

잘한 점

1. 기획서에 나와있지 않는 부분 추가
 - 공고 글의 수정/삭제
 - 공고의 제목
 - 공고의 마감 기한

05 자체 평가 의견

개인 평가 - 조현기

추후 개선점, 보완할 점

1. 공고 마감일 기준 설정
 - 현재 날짜를 가져와서 밸리데이션 체크로 이전 날짜 클릭 시 등록 막기
2. 프로젝트 마감 기한을 준수할 수 있도록 개발일정의 꾸준한 파악
3. 삭제와 같이 중요한 기능들은 사용자의 확인이 필요
4. 회원가입
 - 실패하면 입력 정보 유지

경험한 성과

1. 팀을 이끄면서 팀장의 무게 경험
2. 기획서 파악과 분석의 중요성

05 자체 평가 의견

개인 평가 - 류림

완성도 : 8.5점/10점

아쉬운 점

1. 기획서를 꼼꼼히 보지 않은 점

- 기획서를 꼼꼼하게 보고 시작하지 않아서 페이지를 구현한 후 기획서와 비교했을 때 다른점을 발견하여 수정이 필요했음.
- 코드의 수정을 하게 되면서 시간이 더 오래 걸림.
- 게시판에서 공지사항을 상단에 2개를 보여줘야 하는 것을 나중에 봄으로써 시간이 안되어 구현하지 못함.

2. 데이터베이스의 설계

- Q&A 게시판과 Community 게시판의 성격이 같고, sql 구문이 비슷함에도 테이블을 나눠놔서 구현하는데 두배의 시간이 걸림
- 테이블을 합쳐서 게시판 구분이 어딘지만 명시해 주면 테이블을 나누지 않았어도 됐을 것임.

3. 컨트롤러 갯수

- 게시판의 테이블을 합치지 않으면서 컨트롤러 갯수가 2배가 됨

잘한 점

1. 기획서에 나와있지 않는 부분 추가

- 기획서대로 메인페이지를 구현했더니 양 옆이 너무 비어있어서 top writer 부분을 추가함으로써 비어있는 부분을 채웠음

05 자체 평가 의견

개인 평가 - 류림

추후 개선점, 보완할 점

1. 공지사항 상단 2개 노출
2. 데이터 베이스 테이블 합치기
 - qna와 community의 테이블을 합쳐서 쿼리를 짜면 좀 더 수월하게 구현 가능할것 같음

경험한 성과

1. DAO와 컨트롤러의 이해도 상승
 - 좀 더 복잡한 쿼리문을 작성하면서 sql 구문 이해도 상승

05 자체 평가 의견

완성도 : 8 점 / 10 점

개인 평가 - 전주연

아쉬운 점

1. 다크모드 구현

- 다크 모드 구현하기 위해 시도는 했지만 완벽한 완성은 못했기 때문에 고도화를 할 수 있다면 다크모드를 완성시킬 수 있을 것 같음

2. 이력서 삭제

- 기획서에 없는 기능이었기 때문에 따로 신경을 쓰지 않았는데, 실제 운영되고 있는 사이트일 경우 파일을 올리고 삭제는 따로 안 된다는 점이 큰 문제가 될 수 있다고 생각함 그래서 이런 부분을 깊게 생각하지 못하고 구현하지 못한 것이 아쉬움이 남음

3. 개발 전 꼼꼼한 계획 세우기

- 개발 진행 전 기획서를 살펴보고 어떤 점을 더욱 집중해서 개발해야 할지 생각하고 계획을 세웠다면 지금보다 더 효율적으로 개발을 진행할 수 있고, 시간도 잘 활용할 수 있었을 것 같다는 생각이 들었음

잘한 점

1. 파일 업로드

- 파일 업로드 기능을 통해 지원한 이력서를 구인회사에서 볼 수 있는 기능을 구현한 것

2. 기획서에 디테일 추가

- 기획서에 적혀있지 않은 세세한 디테일 기능 추가함

05 자체 평가 의견

개인 평가 - 전주연

추후 개선점, 보완할 점

1. 다크모드 구현
 - 다크모드 클릭 시 그 값을 세션에 저장하면 구현이 가능할 것 같음
2. 이력서 삭제
 - 파일을 삭제 하지 않아 파일이 남아있을 수 있기 때문에 구현하는 것이 좋을 것 같음
3. 마이페이지 구인구직 테이블 고도화
 - 이력서가 더 많아질 경우 테이블에 페이징 또는 더보기 기능을 추가해서 사용자가 확인하기 편하게 구현하는 것이 좋을 것 같음

경험한 성과

1. JSP, DB의 동작성을 좀더 이해할 수 있었음
2. 파일 업로드 기능 이해
3. 배운 것들을 이용하여 다양한 기능을 스스로 구현해볼 수 있었음

OKKY 2조

감사합니다

천재 IT 교육센터