
네카라쿠배-Java

Encore Playdata : Team 2

이재호, 김도희, 나예지, 윤슬아, 이수영, 정우철, 하가영

목차

1 계획

2 분석 및 설계

3 구현

Part 1

계획

계획

1.1. 주제선정 배경

개발자 맞춤 채용사이트
부재

채용사이트 매번 비교하기
번거로움



직관적인 기업별 채용정보
제공 필요성 인식



직관적인 채용 정보 제공
+
채용 트렌드 제공
사이트

계획

1.2. 주제 및 목표일정

<주제>

빅테크 기업에 취직을 원하는 구직자들이 채용사이트를 매번 비교해야하는 번거로움을 덜기 위해 기업별 채용 공고를 직관적으로 비교할 수 있게 모아놓고, 동시에 트렌드를 확인할 수 있는 웹사이트 만들어보자!!!

		기간 (일)	수(4/21)	목(4/22)	금(4/23)	월(4/26)	화(4/27)	수(4/28)	목(4/29)	금(4/30)	월(5/3)	화(5/4)	수(5/5)	목(5/6)	금(5/7)
프로젝트 계획	주제선정	1													
	역량파악	1													
	간트차트	1													
프로젝트 분석	아키텍처 구조	1													
	다이어그램	1													
	요구사항명세서	2													
프로젝트 설계	DB Modeling	1													
	Front UI	1													
	Usecase Diagram	1													
	Business Logic Template	1													
	데이터 수집 및 전처리	2													
프로젝트 구현	개발	7								센터 휴관			어린이날		
	취합 및 테스트	4													



Part 2

분석 및 설계

분석 및 설계

2.1. 요구사항 분석

- 검색 기능을 통해 원하는 키워드와 조건에 따른 공고를 검색할 수 있다.
- 빅테크 기업 취업 준비를 위해 상세한 채용공고 정보를 알 수 있다.
- 랭킹 페이지에서 빅테크 기업의 채용추세(랭킹) 파악할 수 있다.
- 관심 있는 공고를 스크랩(저장)하고 채용정보를 간단하게 비교할 수 있다.



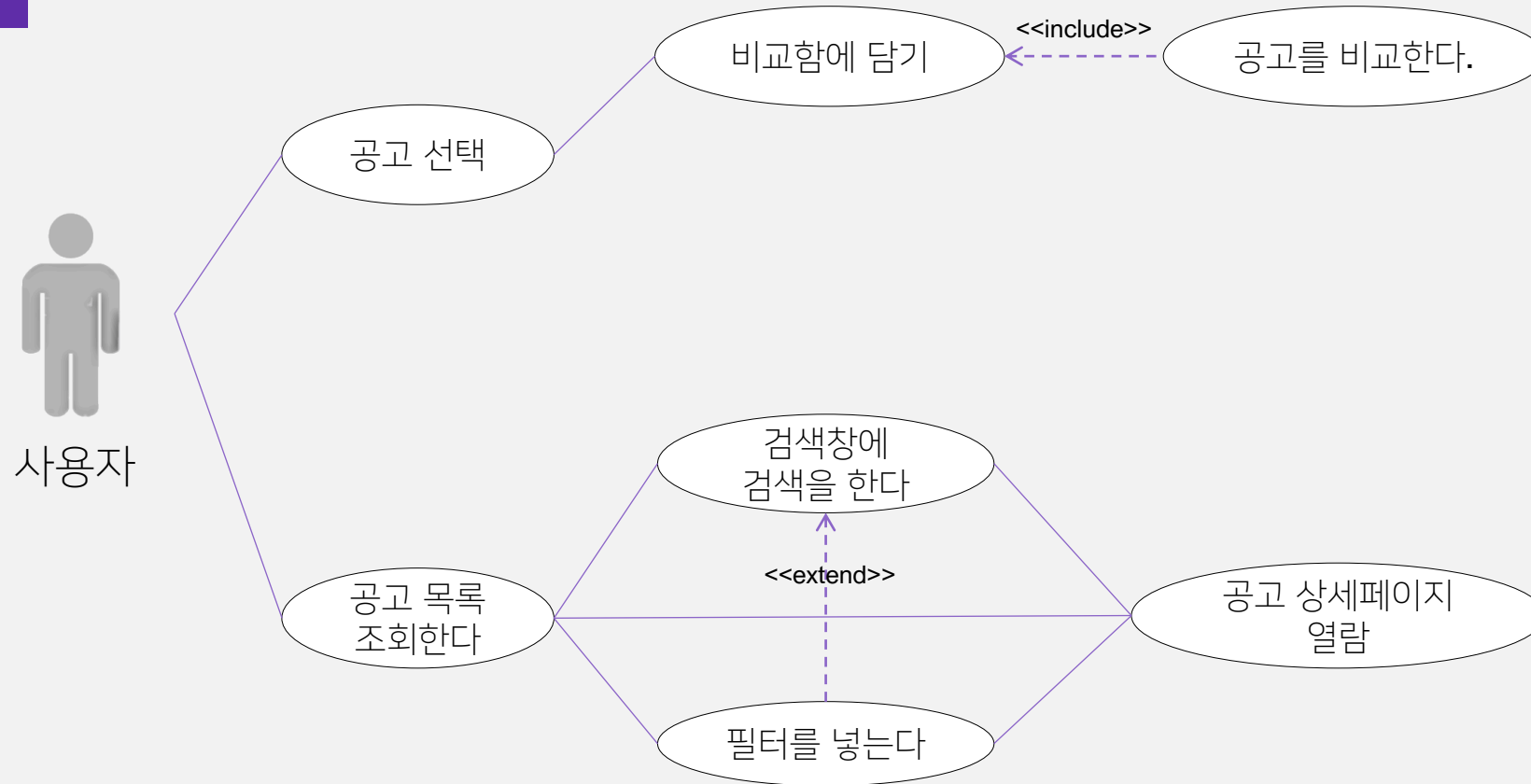
분석 및 설계

2.2. 요구사항 명세서 (SRS)

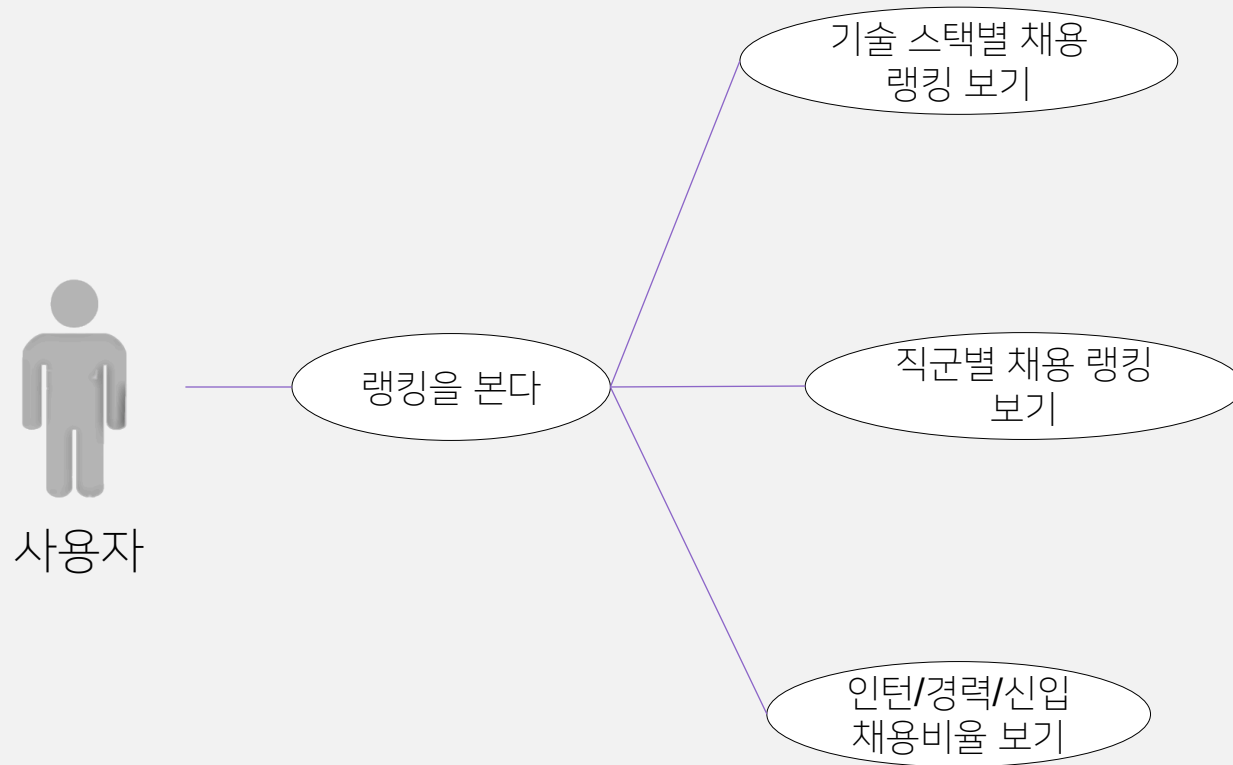
code
RL: Recruit list
RA: Rank
U: User

대분류	분류	ID	요구사항명	요구사항내용	담당자	
채용공고	공고목록	RL01	전체 공고 목록 조회	시작페이지 접속시 전체 공고 목록을 보여준다	이재호, 나예지	
		RL02	공고 마우스 호버	공고에 대한 간략한 내용을 보여준다.	이재호, 나예지	
		RL03	공고 클릭	공고 클릭시 공고 상세페이지를 새창으로 보여준다.	이재호, 나예지	
	네비게이션	RL04	네비게이션 바	로그인 페이지, 스크랩 페이지, 랭킹 페이지, 홈페이지로 돌아간다.	이수영, 김도희, 하가영	
		RL05	채용사이트로 이동	(상세 페이지 내)해당 공고의 공식 채용 사이트를 새 창으로 보여준다	이수영, 김도희, 하가영	
	필터링	RL06	공고 필터링	필터(직군, 경력, 코테여부,학력 (근무지역,근무시간)) 를 넣으면 그 값을 기준으로 화면을 정렬한다. 필터 해제를 하는 기능도 있다.	이재호, 나예지	
	검색	RL07	공고 검색	검색어에 대한 공고를 출력한다.(회사이름, 언어)	이재호, 나예지	
	비교	RL08	공고 비교하기	비교 페이지를 새창으로 띄워준다.(회사이름, 직군, 기술스택, 채용형태, 마감일자)	이수영, 김도희, 하가영	
		RL09	비교 카트에 추가	선택한 공고들의 목록을 화면 비교 카트에서 보여준다.	이수영, 김도희, 하가영	
		RL10	비교 카트 삭제	비교 카트에서 공고를 삭제한다.	이수영, 김도희, 하가영	
		RL11	비교 카트 접기/펼치기	비교 카트를 접거나 펼친다.	이수영, 김도희, 하가영	
랭킹	랭킹	RA01	기술 스택별 순위	채용 공고에 많이 기재된 언어 순위를 나타낸다.	윤슬아, 정우철	
		RA02	직군별 공고 수	채용 공고에서 직군별 공고 수가 많은 순서로 정렬.	윤슬아, 정우철	
		RA03	경력/신입/인턴 비율	경력 사항별 정렬	윤슬아, 정우철	
사용자	사용자	U01	로그인	로그인을 한다.	이수영, 김도희, 하가영	
		U02	로그아웃	로그아웃을 한다.	이수영, 김도희, 하가영	
		U03	정보수정	회원의 정보를 수정한다.	이수영, 김도희, 하가영	
	회원가입	U04	비밀번호 확인	비밀번호를 확인한다.	이수영, 김도희, 하가영	
		U05	아이디 중복확인	아이디 중복확인을 한다.	이수영, 김도희, 하가영	
		U06	비밀번호 암호화	비밀번호가 db에 저장될때 암호화를 시켜 넣는다	이수영, 김도희, 하가영	
		U07	이메일 형식확인	이메일 형식확인을 한다.	이수영, 김도희, 하가영	
	스크랩	U08	공고 스크랩하기	로그인한 상태에서 스크랩 버튼을 선택하여 스크랩 목록에 추가 및 삭제한다. 로그인이 되어있지 않다면, 로그인 컨펌창을 띄운다	이수영, 김도희, 하가영	
		U09	바로 지원하기	스크랩한 공고의 지원사이트를 새 창으로 연다.	이수영, 김도희, 하가영	

채용공고 및 검색



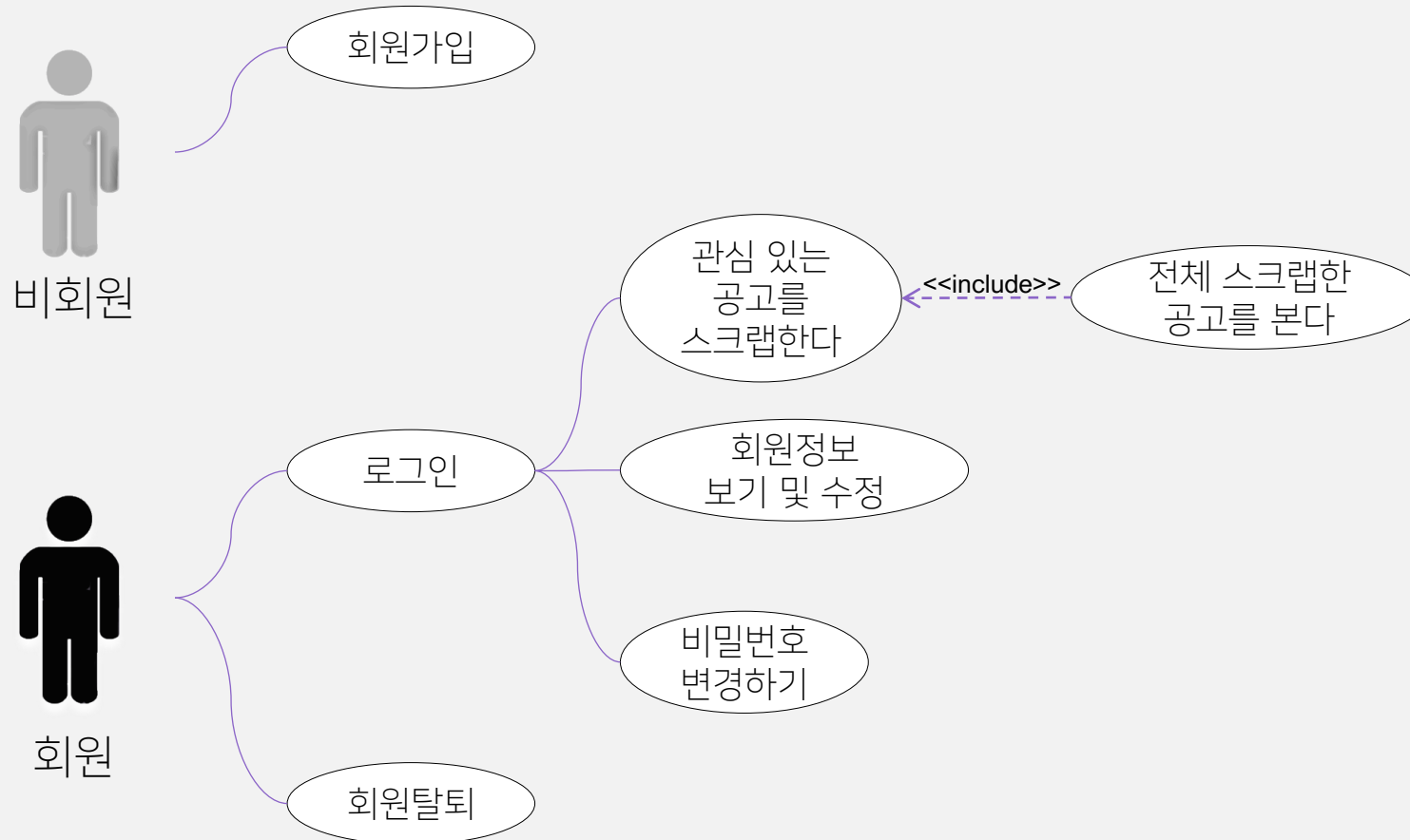
랭킹



분석 및 설계

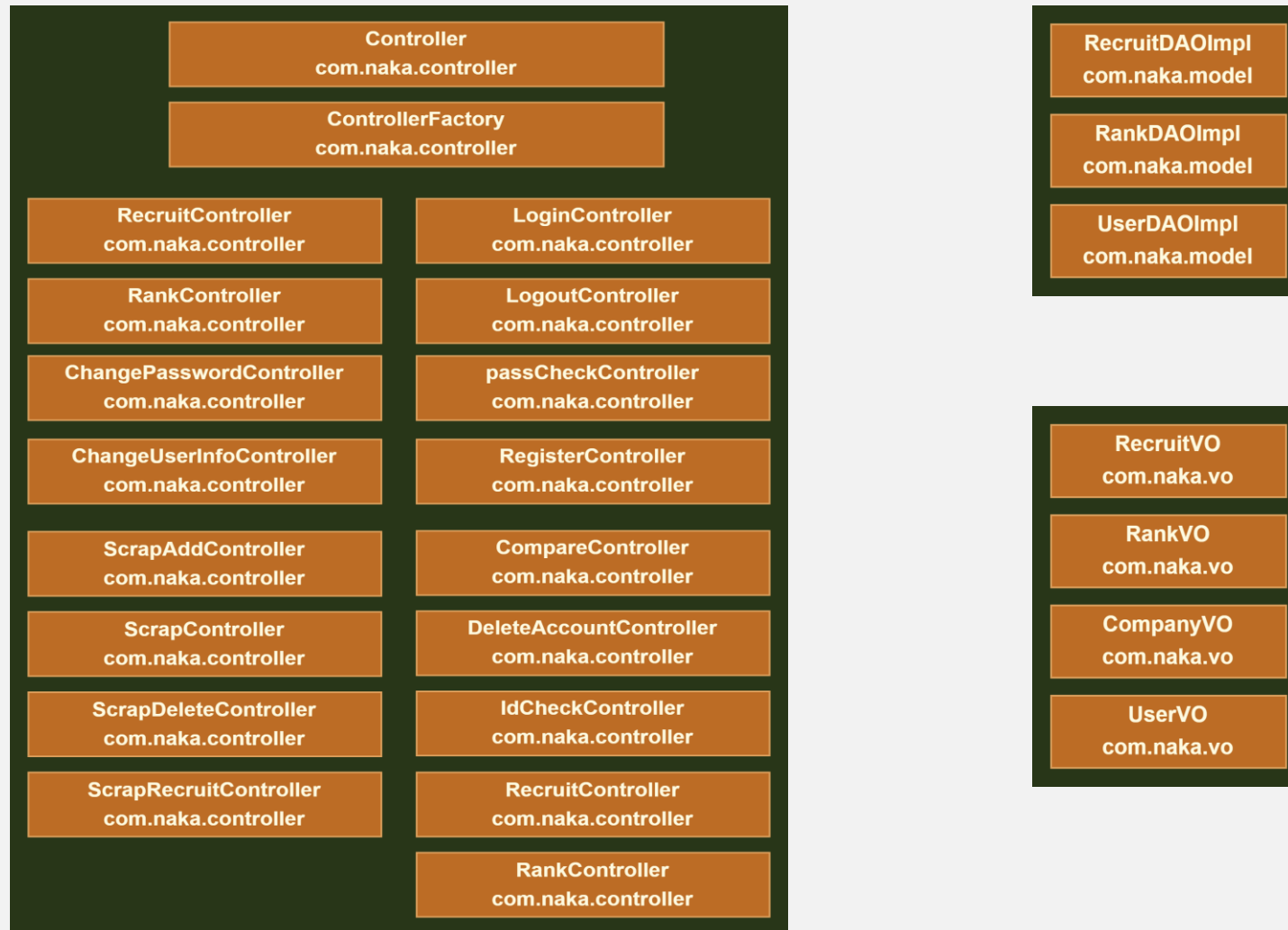
2.3.3. Usecase Diagram

사용자 기능



분석 및 설계

2.4. Package Hierarchy (패키지 계층구조)



분석 및 설계

2.5. DB - 분석

user

- 회원 정보 관리
- 스크랩한 공고에 대한 id값 저장

recruit

- 공고 데이터 관리
- 공고 키워드 검색
- 공고 필터 검색

ranking

- 공고 주제별 랭킹 관리
- 공고 데이터 키워드를 기반으로 데이터 수집후 랭킹 정렬
- 그래프 생성

company

- 회사 데이터 관리

분석 및 설계

2.6. DB Modeling

User						
Table logical name						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓	+		아이디	u_id	VARCHAR(45)
		+		비밀번호	password	VARCHAR(100)
		+		이름	name	VARCHAR(45)
		+	✓	주소	address	VARCHAR(45)
		+		전화번호	phone	VARCHAR(45)
		+	✓	구직상태	status	VARCHAR(45)
		+		이메일	email	VARCHAR(45)
		+	✓	분야	job_field	VARCHAR(45)
		+	✓	학력	education	VARCHAR(45)
		+	✓	생년월일	birth_day	DATE
		+	✓	즐거찾기	scrap	VARCHAR(5000)
		+			salt	VARCHAR(45)

Recruit						
공고						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓	+		아이디	r_id	INT
		+	✓	직군	position	VARCHAR(45)
		+	✓	기술스택	tech	VARCHAR(500)
		+	✓	채용형태	job_type	VARCHAR(45)
		+	✓	학력	education	VARCHAR(45)
		+		공고 이미지	img	VARCHAR(100)
		+		공고주소	link	VARCHAR(200)
		+		회사아이디	c_id	INT
		+	✓	마감일자	exp_date	DATE
		+	✓	시작일자	start_date	DATE
		+		공고제목	title	VARCHAR(120)

Company						
회사정보						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓	✓			c_id	INT
		+		회사이름	company_name	VARCHAR(45)
		+		로고 이미지	logo_img	VARCHAR(45)

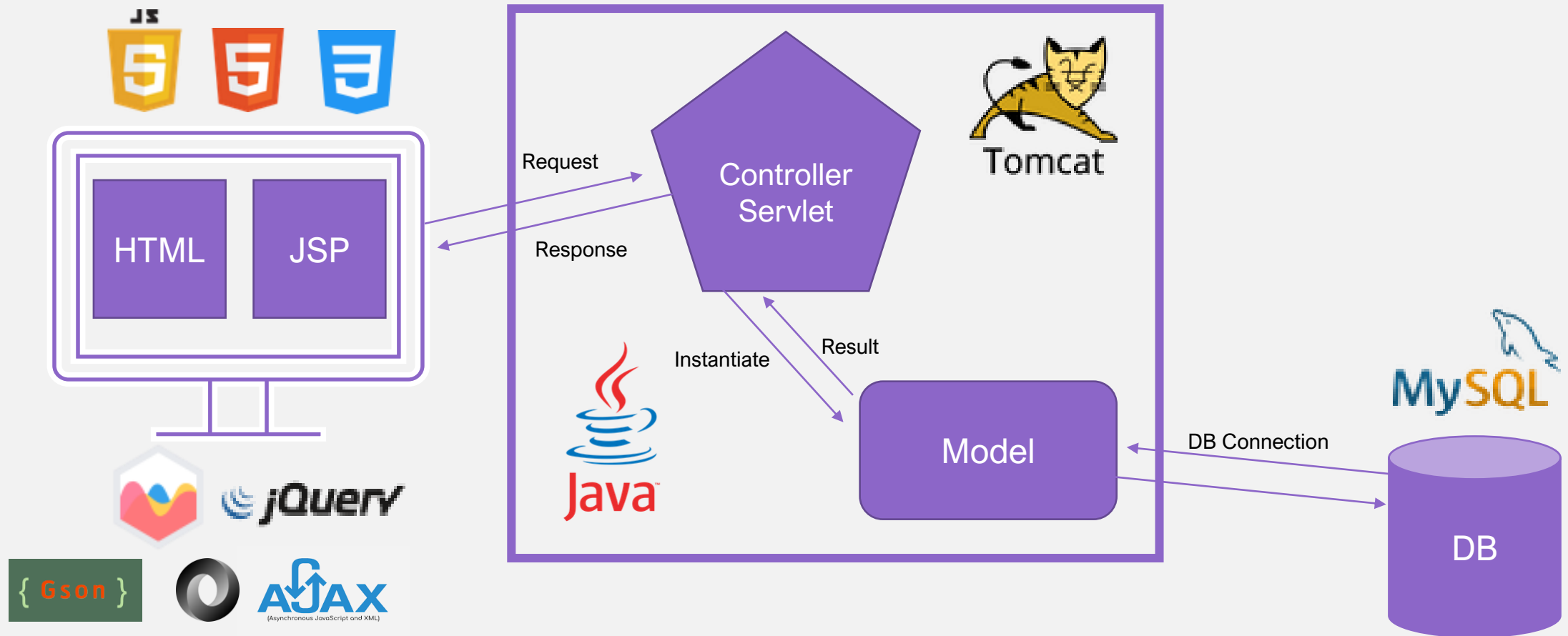
Ranking						
Table logical name						
PK	AI	FK	Null	Logical Name	Name	Type
✓		+			id	VARCHAR(45)
		+			name	VARCHAR(200)
		+			value	INT

Memo name						
T_000 => 기술별						
P_000 => 직군별						
J_000 => 채용형태별						

분석 및 설계

2.7. 프로젝트 아키텍처

기술 스택: Java, JavaScript, HTML5, CSS3, MySQL, Apache, jQuery, Json, Gson, chart.js, Ajax



Part 3

구현

채용공고 목록

- 채용공고의 제목이 표시



- Ajax 이용하여 객체를 받아와서 출력
- 공고의 목록 수를 페이징에 이용

```
$.ajax({
  url: "recruit.do",
  type: "post",
  data: {num: pageNumber, position: $("#select1").val(), job_type: $("#select2").val(), education: $("#select3").val(),
    keyword: ($("#keyword").val()), scrap: ($("#scrap").val()), command: "getRecruits",
  },
  success: function(data) {
    var jsonObject = JSON.parse(data);
    $("#recruit_container div.row").html('');
    if(jsonObject.length > 0) {
      var str = "";
      var scrapList = [];
      var r_vo = $("#rvo");
      var u_id = $("#u_id");

      if(u_id != "") {
        scrapList = scrap(u_id);
      }

      for(key in jsonObject) {
        var src;
        var svg;
        if(scrapList.includes(jsonObject[key].r_id) == false) {
          src = "url(assets/img/elements/heart-regular.svg)";
          svg = "svg1";
        } else {
          src = "url(assets/img/elements/heart-solid.svg)";
          svg = "svg2";
        }
        var title = jsonObject[key].title;
        if(title.length > 24) title = title.substr(0, 24) + "...";
        var str = '<div class="col-lg-3 col-md-6 col-sm-6" style="overflow: hidden;"><div class="single-location mb-30" name = "location"></div><div class="location"><div class="location-content"><div class="location-title"><h3>' + title + '</h3></div><div class="location-description"><p>' + jsonObject[key].description + '</p></div></div></div></div>';
        $("#recruit_container div.row").append(str);
        $("#"+jsonObject[key].r_id).data('info', jsonObject[key]);
      }
    }
  }
});
```

```
function setTotalCount() {
  return $.ajax({
    url: "recruit.do",
    type: "post",
    data: {position: $("#select1").val(), job_type: $("#select2").val(), education: $("#select3").val(),
      keyword: ($("#keyword").val()), scrap: ($("#scrap").val()), command: "pageCount",
    },
    success: function(data) {
      totalPageCount = (JSON.parse(data).pageCount / 16);
      if(totalPageCount < 1) totalPageCount++;
    }
  });
}
```


채용공고 목록

```
String[][] inputs = new String[][] { {"position",position}, {"job_type",job_type}, {"education",education}, {"title"}, {"tech"}, {"company_name"}

ArrayList<RecruitVO> list = new ArrayList<RecruitVO>();
try {
    con = getConnection();
    String query = "select * from (select r.* from recruit r inner join company c on (c.c_id = r.c_id) where r.c_id = c.c_id ";
    for(int i=0;i<3;i++) {
        if(inputs[i][1]!=null&&!inputs[i][1].equals("")) {
            query+="and ";
            query+=inputs[i][0]+" like '%"+inputs[i][1]+'% ' ";
        }
    }

    if(keyword!=null&&keyword.equals("")) {
        query+="and ";
        query+="(";
        for(int i=0;i<6;i++) {
            query+= inputs[i][0]+" like '%"+keyword+'% ' ";
            if(i<5)
                query+="or ";
        }
        query+=")";
    }

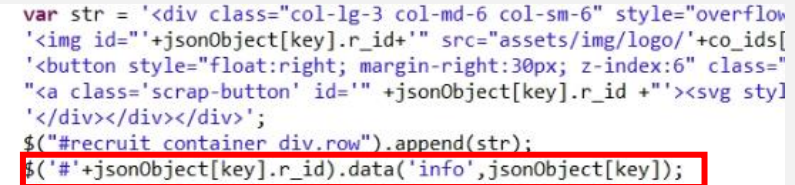
    if(scrap!=null&&!scrap.equals("")) {
        query+= "scrap in (";
        String[] s = scrap.split(",");
        for(int i=0;i<s.length;i++) {
            query+= "'"+s[i]+(i==s.length-1?"','":"'")";
        }
    }
    query+= ") T limit ?,?";
    ps = con.prepareStatement(query);
    ps.setInt(1, 4*4*(pageNumber-1));
    ps.setInt(2, 4*4);
    rs = ps.executeQuery();
}
```

```
public RecruitDAOImpl() {
    try {
        InitialContext ic = new InitialContext();
        ds = (DataSource)ic.lookup("java:comp/env/jdbc/mysql");
        setTotalCount();
    } catch (Exception e) {
        System.err.println("Datasource create Error");
    }
}
```

DAOImpl 최초 생성시 전체 공고 수를 세어 놓고
아무런 검색 조건이 없을 시 미리 세어 놓은 값을 반환

Query를 작성할때 검색 키워드 존재 여부를
따져서 이어 붙여서 결과 리스트를 받음

- 마우스를 올릴 시 간략한 공고 내용이 표기
- 클릭하면 공고 상세페이지로 이동



```

    $(this).on('mouseenter', '.single-location.mb-30', function() {
      $(this).find('.content').addClass('recruit-hover');
      var id = $(this).find('img').attr('id');
      var recruitInfo = $(this).find('#'+id).data('info');
      var tech = recruitInfo.tech;
      if(tech.length>24) tech = tech.substr(0,20)+"...";
      var str = "<div>직무 : "+ recruitInfo.position + "<br> 채용유형 : "+recruitInfo.job_type + "<br>경력 : "+recruitInfo.education+"<br>기술스택 : "+tech;
      $(this).find('p').html("");
      $(this).find('.content').html(str);
    });
  });

```

```

$('.location-img').click(function() {
    var id = $(this).find('img').attr('id');
    var win = window.open("recruit_detail.do?id="+id);
});
$('.content').click(function() {
    var id = $(this).parent().parent().find('img').attr('id');
    var win = window.open("recruit_detail.do?id="+id);
});
}

```

©Saebyeol Yu. Saebyeol's PowerPoint

3.1.2. 주요 기능 - 채용공고 비교하기



Manifest	Key	Value
Service Workers	compareStorage	[{"job_type": "경력(1~4년)", "tech": "Javascript, WebGL, Canvas, Svg, WebAssembly", "ir"}, {"job_type": "경력(1~4년)", "tech": "Javascript, WebGL, Canvas, Svg, WebAssembly", "ir"}, {"job_type": "무관", "tech": "React, Vue, Angular, HTML, CSS, Javascript, Git", "ir"}]
Storage		
Local Storage		
http://localhost:8888		

- Ajax를 이용해 공고를 비교함에 담는다.
- 로컬 스토리지를 사용하여 저장한다.
- 최소 2개, 최대 3개까지 비교가능

```

$('.row').on('click', 'button[name=addItemCart]', function () {
    //로컬 스토리지에 저장 setItem()
    console.log("click");
    var arr = []
    if(localStorage.getItem('compareStorage') != null){
        var output = localStorage.getItem('compareStorage');
        arr = JSON.parse(output);
    }

    if(arr.length >= 3){
        alert("비교항이 가득 찼습니다. 슬롯을 비운 후 다시 시도해 주세요");
    }else{
        var flag = false;
        for(var i = 0; i<arr.length;i++){
            if(arr[i].r_id==$(this).attr('id')){
                flag = true;
            }
        }
        if(flag){
            alert("비교함에 이미 존재하는 공고입니다.");
        }else{
            var id = $(this).val();

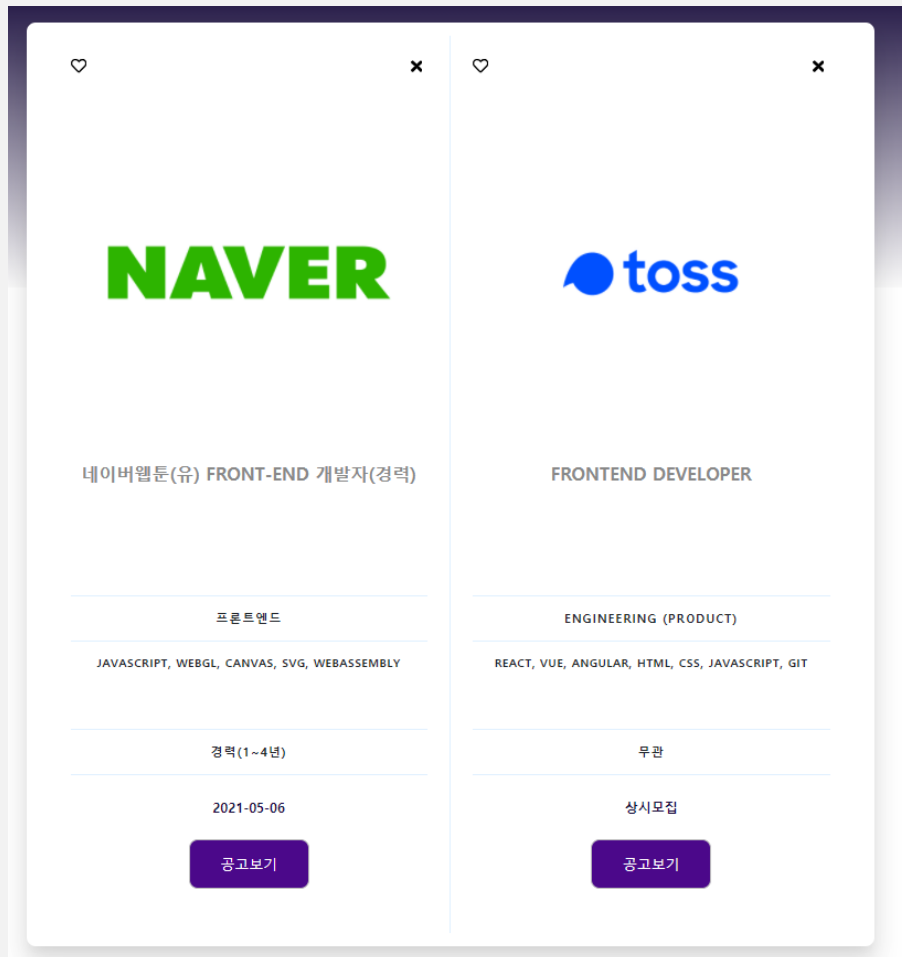
            $.ajax({
                type: "post",
                url: "compare.do",
                data: "id=" + id,

                success: function(result) {
                    arr.push(JSON.parse(result));
                    localStorage.setItem('compareStorage', JSON.stringify(arr));
                }
            });

            |
            setTimeout(() => { arrLengthCheck(); }, 100);
            setTimeout(() => { showCart(); }, 100);
        }
    }
});

```


3.1.2. 주요 기능 - 채용공고 비교하기



- 비교함에 담긴 공고의 회사 로고, 공고제목, 직군, 기술 스택, 경력, 마감일자를 비교

```
function refresh(){
  var output = localStorage.getItem('compareStorage');
  var compare_list = JSON.parse(output);
```

비교함에 담긴 공고 가져오기

```
$('#compare-table').on('click', '.delete', function() {
  var arr = []
  if(localStorage.getItem('compareStorage') != null){
    var output = localStorage.getItem('compareStorage');
    arr = JSON.parse(output);
  }
  if(arr.length !=null && arr.length > 2){
    var idx = $(this).attr('id');
    for(var i=0; i<arr.length; i++){
      if(arr[i].r_id == idx){
        arr.splice(i, 1);
      }
    }

    localStorage.setItem('compareStorage',JSON.stringify(arr));
  }else{
    alert("공고는 2개 이상 비교가 가능합니다.");
  }

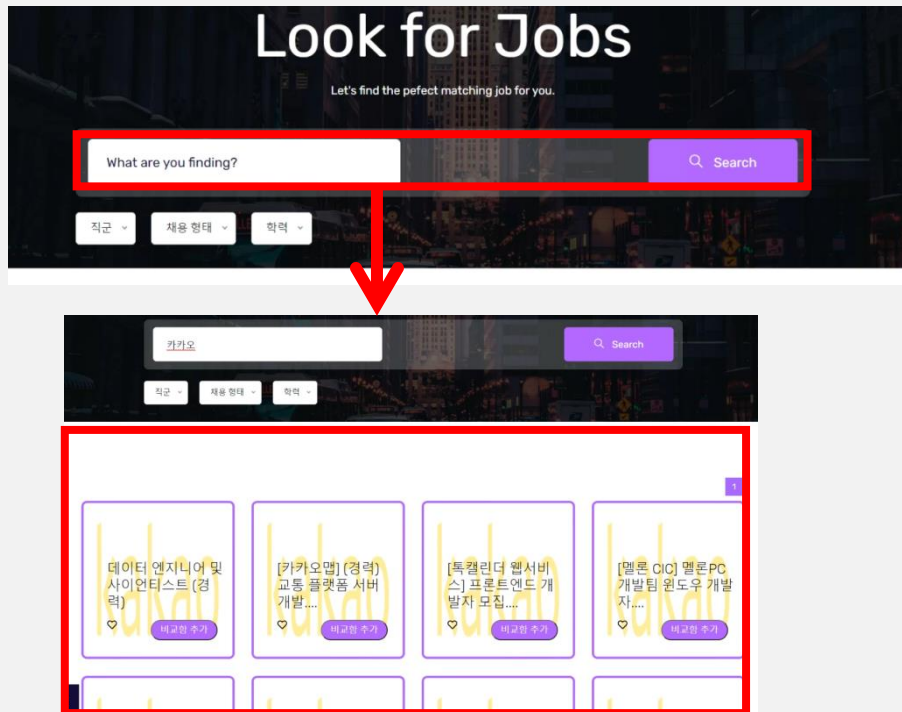
  refresh();
});
```

비교함에서 공고 삭제

3.2.1. 주요 기능 - 검색창에 검색하기

검색창

- 검색어를 입력 후 Enter 또는 Search 버튼을 클릭

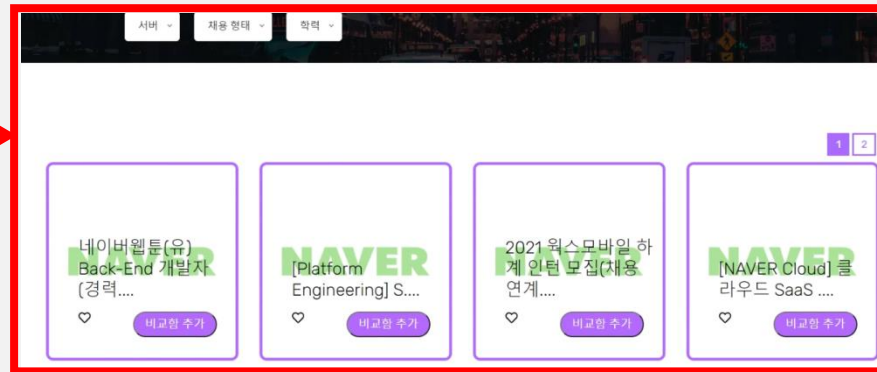
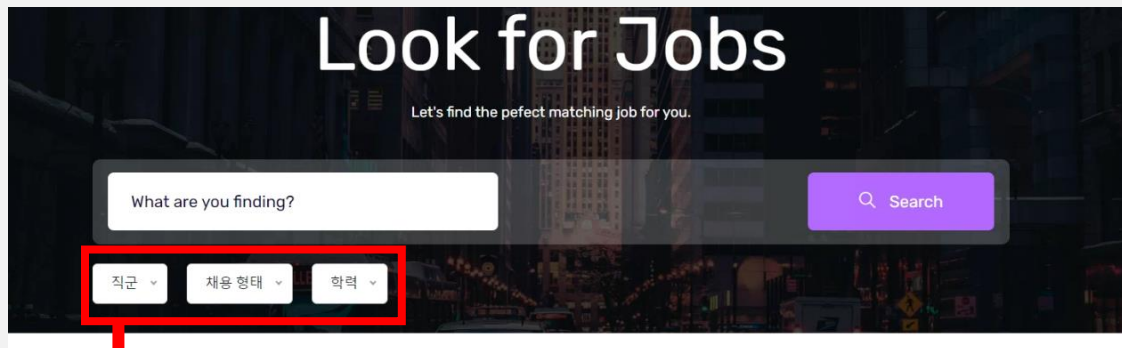


```
$("#keyword").keyup(function(e){
    if(e.keyCode==13){
        e.preventDefault();
        pageNumber=1;
        $.when(setTotalCount()).done(function(){
            pageCountModify();
        });
    }
});
$("#searchBtn").click(function(){
    pageNumber=1;
    $.when(setTotalCount()).done(function(){
        pageCountModify();
    });
});
```

Id가 keyword인 input에 엔터키(keycode =13)이 입력될 경우 기존의 submit 이벤트가 페이지를 이동시키기 때문에 삭제 하고 현재 페이지를 1로 초기화

필터링

- 선택창의 키워드 선택 시 선택된 분야의 키워드로 검색



```

$('select').change(function(){
    pageNumber=1;
    $.when(setTotalCount()).done(function(){
        pageCountModify();
    });
});

```

Select 변경시 서버에 필터된 공고수를 전달 받고 페이징 버튼들 수정

```

function pageCountModify(){
    $("#page_num").html('');
    var str = "";
    pageNumber = new Number($(this).text());
    pageNumber = pageNumber==0?1:pageNumber;
    for(var i=(pageNumber-2<1?1:pageNumber-2);i<=pageNumber+2&&i<=totalPageCount;i++){
        str += "<div class='pageBtn'>"+i+"</div>";
    }
    $("#page_num").html(str);
    $(".pageBtn").css("color", "black");
    $(".pageBtn:contains("+pageNumber+")").addClass("pageBtnA");
    addLoadedRecruits();
    $(document).off('click', ".pageBtn", pageCountModify);
    $(document).on('click', ".pageBtn", pageCountModify);
}

```

Select 변경시 서버에 필터된 공고수를 전달 받고 페이징 버튼들 수정

3.3. 주요 기능 - 랭킹

```

@Override
public void refreshRank() {
    ArrayList<RecruitVO> list = new ArrayList<RecruitVO>();
    list.add(null);

    HashMap<String, Integer> tech_map = new HashMap<String, Integer>();
    HashMap<String, Integer> position_map = new HashMap<String, Integer>();
    HashMap<String, Integer> job_type_map = new HashMap<String, Integer>();

    for (int i = 1; list.size() != 0; i++) {
        list = RecruitDAOImpl.getInstance().getRecruits(i);
        for (RecruitVO vo : list) {
            for (String s : vo.getTech().split(", ")) {
                if(s.equals(""))continue;
                tech_map.put(s, tech_map.getOrDefault(s, 0) + 1);
            }
            for (String s : vo.getPosition().split(", ")) {
                if(s.equals(""))continue;
                position_map.put(s, position_map.getOrDefault(s, 0) + 1);
            }
            for (String s : vo.getJob_type().split("/")) {
                if(s.equals(""))continue;
                job_type_map.put(s, job_type_map.getOrDefault(s, 0) + 1);
            }
        }
    }

    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    int i = 1;

    try {
        con = getConnection();
        String query = "INSERT INTO ranking (id, name, value) VALUES(?,?,?)";
        ps = con.prepareStatement(query);

        for (Map.Entry<String, Integer> entry : tech_map.entrySet()) {
            ps.setString(1, "t_" + i++);
            ps.setString(2, entry.getKey());
            ps.setInt(3, entry.getValue());
            ps.executeUpdate();
        }
        for (Map.Entry<String, Integer> entry : position_map.entrySet()) {
            ps.setString(1, "p_" + i++);
            ps.setString(2, entry.getKey());
            ps.setInt(3, entry.getValue());
            ps.executeUpdate();
        }
        for (Map.Entry<String, Integer> entry : job_type_map.entrySet()) {
            ps.setString(1, "j_" + i++);
            ps.setString(2, entry.getKey());
            ps.setInt(3, entry.getValue());
            ps.executeUpdate();
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeAll(rs, ps, con);
    }
}

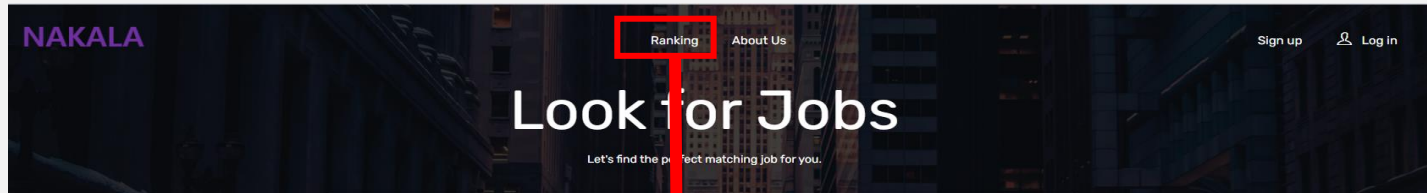
```

RankDAOImpl
RefreshRank() 메소드

recruit 테이블 내 데이터를 불러와 ranking table 생성
id 값으로 각각 그래프에 들어갈 데이터를 구분해줌.



	id	name	value
▶	j_347	경력(4년 이상)	3
	j_348	경력(1년 이상)	9
	j_349	경력(10년 이상)	4
	j_350	경력(7년 이상)	6
	j_351	인턴	3
	j_352	경력(8년 이상)	5
	j_353	경력(1~5년)	1
	j_354	경력(1~4년)	8
	j_355	경력(6개월 이상)	2
	j_356	경력(5년 이상)	34
	j_357	경력(2년 이상)	35
	j_358	경력(3~5년)	1
	j_359	경력(5년 미만)	2
	j_360	경력(3년 이상)	79
	j_361	무관	26
	j_362	경력(6년 이상)	2
	j_363	경력(2~5년)	2
	j_364	경력	48
	j_365	신입	9
	j_366	채용연계형 인턴	3
	p_275	네트워크	2
	p_276	머신러닝	4
	p_277	DBA	5
	p_278	Server	21



선택한 메뉴에 따라 데이터 교체

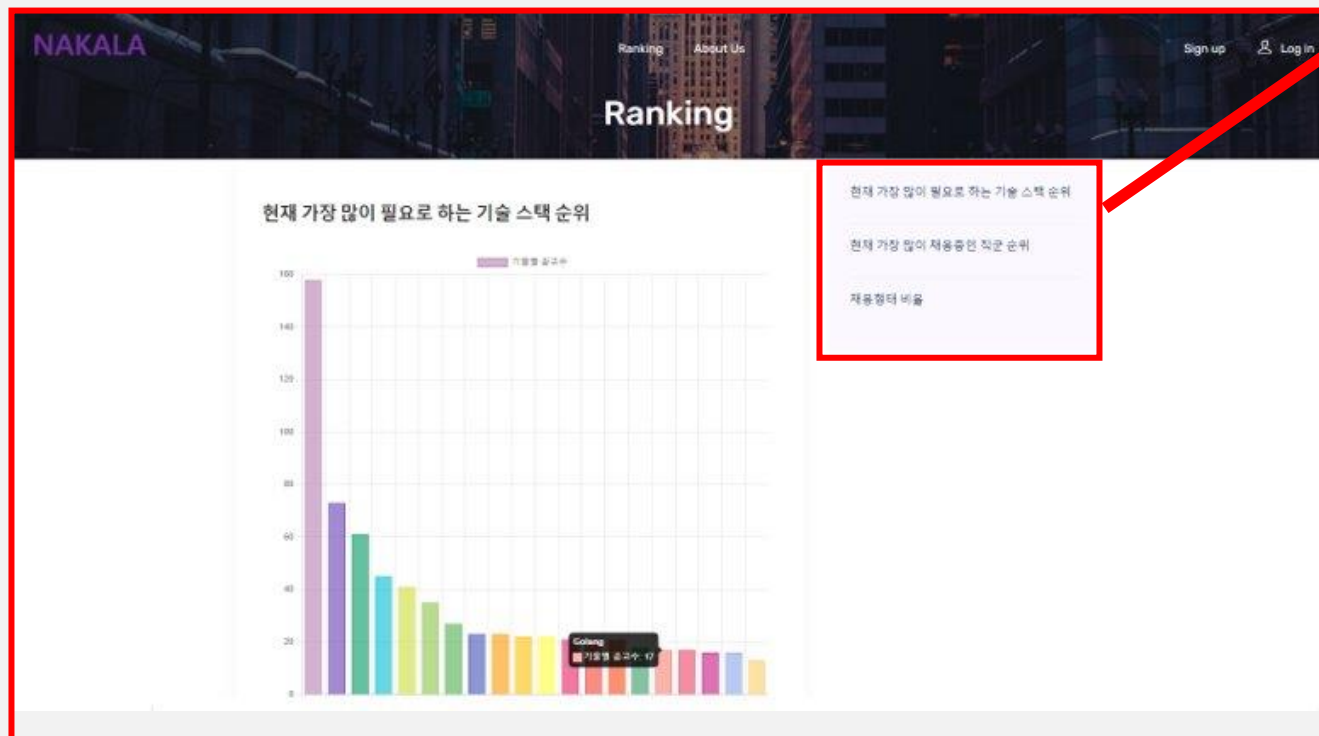
```
$(function(){
  var els = $(".rank-list li");
  var values = ["tech", "position", "job_type"];
  loadData("tech");
  $(''.blog-head').html(els[0].text());
  for(var i=0; i<3; i++){
    $(els[i]).data("command", values[i]);
    $(els[i]).on("click", function(e){
      loadData($(this).data("command"));
      var title = $(this).text();
      $(''.blog-head').html(title);
    });
  }
});
```

json형태의 랭킹 데이터를
chart 라이브러리를 사용하여 그려줌

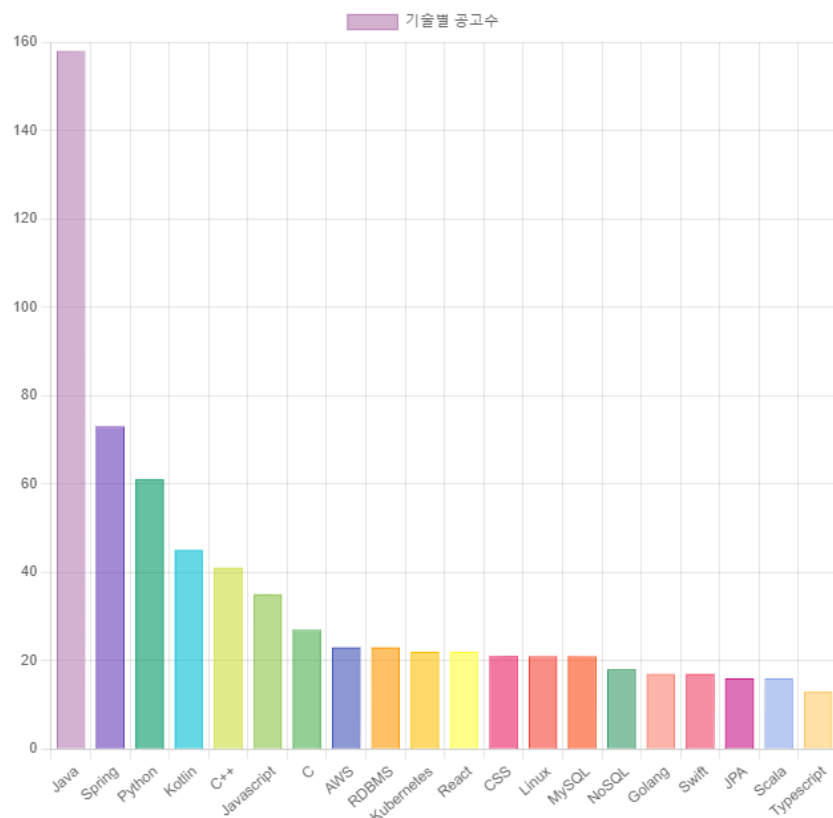
```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script type="text/javascript">
var myChart = null;
function loadData(command){
  var labels = [];
  var input_data = [];
  var types = {"tech":["bar", "기술"], "position":["bar", "직군"], "job_type":["doughnut", "채용 유형"]};

  $.ajax({
    url: "ranking.do",
    type: "post",
    data: {rank_command: command},
    success: function(data){
      var jsonObject = JSON.parse(data);
      for(key in jsonObject){
        labels.push(jsonObject[key].name);
        input_data.push(jsonObject[key].value);
      }
      var ctx = document.getElementById("myChart");
      if(myChart!=null){
        myChart.destroy();
      }

      myChart = new Chart(ctx, {
        type: types[command][0],
        data: {
          labels: labels,
          datasets: [{
            label: types[command][1]+'별 순위',
            data: input_data,
            backgroundColor: [
              'rgba(181, 127, 179, 0.6)',
            ]
          }]
        }
      });
    }
  });
}
```



현재 가장 많이 필요로 하는 기술 스택 순위



```
@Override
public ArrayList<RankVO> getTechRank() {
    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ArrayList<RankVO> list = new ArrayList<RankVO>();
    try {
        con = getConnection();
        String query = "SELECT name,value FROM ranking WHERE id LIKE 't%' ORDER BY value DESC LIMIT 20;";
        ps = con.prepareStatement(query);

        rs = ps.executeQuery();
        while (rs.next()) {
            list.add(new RankVO(rs.getString("name"), rs.getInt("value")));
        }

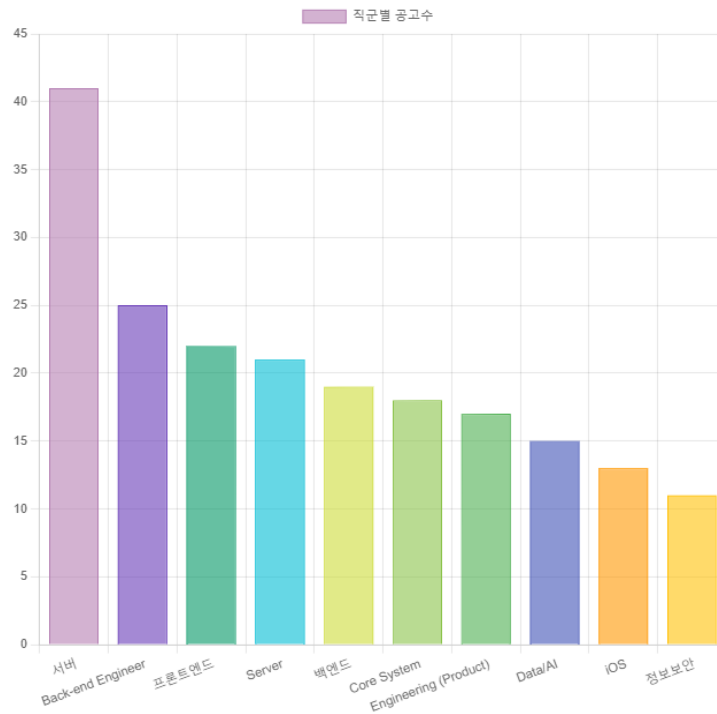
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeAll(rs, ps, con);
    }
    return list;
}
```

RankDAOImpl

getTechRank() 메소드

ranking table의 t_00 id 값을 불러와 각 공고에 명시된 기술스택(name)의 중복값(value)을 내림차순으로 정렬

현재 가장 많이 채용중인 직군 순위



```
@Override
public ArrayList<RankVO> getPositionRank() {
    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    ArrayList<RankVO> list = new ArrayList<RankVO>();
    try {
        con = getConnection();
        String query = "SELECT name,value FROM ranking WHERE id LIKE 'p%' ORDER BY value DESC LIMIT 10;";
        ps = con.prepareStatement(query);
        rs = ps.executeQuery();

        while (rs.next()) {
            list.add(new RankVO(rs.getString("name"), rs.getInt("value")));
        }

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeAll(rs, ps, con);
    }
    return list;
}
```

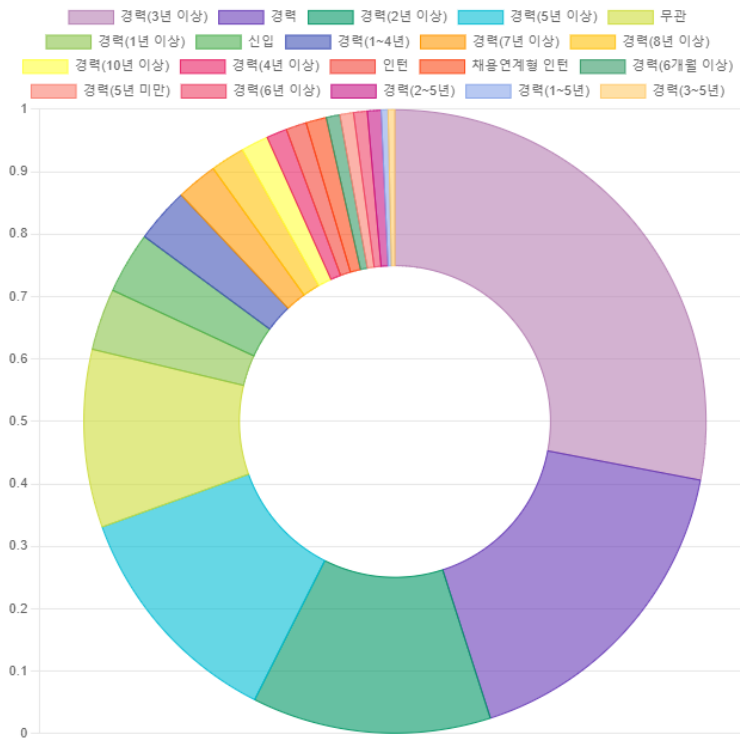
RankDAOImpl

getPositionRank() 메소드

ranking table의 p_00 id 값을 불러와 각 공고에 명시된 직군(name)의 중복값(value)을 내림차순으로 정렬

3.3. 주요 기능 - 랭킹 (채용형태별 비율)

채용형태 비율



```

@Override
public ArrayList<RankVO> getTypeRate() {
    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    ArrayList<RankVO> list = new ArrayList<RankVO>();
    try {
        con = getConnection();
        String query = "SELECT name,value FROM ranking WHERE id LIKE 'j%' ORDER BY value DESC;";
        ps = con.prepareStatement(query);
        rs = ps.executeQuery();
        while (rs.next()) {
            list.add(new RankVO(rs.getString("name"), rs.getInt("value")));
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeAll(rs, ps, con);
    }
    return list;
}
  
```

RankDAOImpl

getTypeRate() 메소드

ranking table의 j_00 id 값을 불러와 각 공고에 명시된
채용형태(name)의 중복값(value)을 내림차순으로 정렬

3.4.1. 주요 기능 - 회원가입

NAKALA

ID

password

password check

name

address

phone number

구직상태 ☐ 학생 ☐ 구직자 ☐ 채용담당자

email

분야

최종학력

birthday 연도-월-일

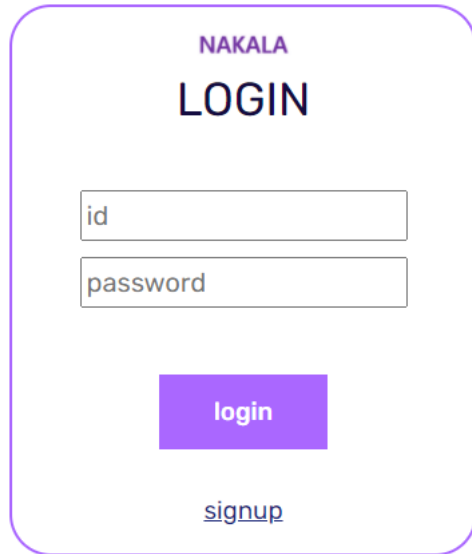
회원가입

```
$(function () {
    $("form").submit(function(){
        var flag = passReg() | checkPass() | checkId();

        if(flag == 0){
            var queryString = $("form[name=loginFormData]").serialize() ;
            $.ajax({
                type: "post",
                url: "register.do",
                data: queryString,
                encode: true,
                success: function(result) {
                    alert("회원가입이 성공적으로 완료되었습니다.");
                    window.location.href = 'login.jsp';
                },
                error: function(err){
                    alert("error");
                }
            });
        }else{
            alert("ID 또는 비밀번호를 다시 입력해주세요.");
        }
    }
});
```

- 회원가입을 하면 회원의 정보가 DB에 저장됨.
- 비밀번호는 그대로 저장되지 않고 salt값을 통해 암호화가 되어 저장됨.
- id 중복 체크를 하고, 비밀번호는 regex를 이용해 안전한 비밀번호를 생성하도록 만들.
- 회원가입에 성공하면 login페이지로 이동함.

u_id	password	name	addr	phone	status	email	job_field	education	birth_day	scrap	salt
111	ebddd60247e053f358719de03de993ee38c91adad69d12ce0c42b5529fa1...	수영	SF	01011111111	구직자	sooyoung.lee712@gmail.com	서버		2021-05-04		abb942fc5b99c12d
123	d04e1d297887fcac917942d62022f2f8031ec0698ccd2c46f01ed5bc9c21311f	수영	SF	01011111111	구직자	sooyoung.lee712@gmail.com	서버	대학교 졸업	2021-05-06		8654d8d717621d92
1234	ece1113ba27f32917d65bae72b616246f9b40abe930d2b41500fe948fe1a...	수영	NY	01011111111	구직자	sooyoung.lee712@gmail.com	서버	대학교 졸업	2021-05-06		c04b49371eb9f62a
222	5d68dd2897243d0ea6247f19532fa2fde53eebed8d768f76e67201604427...	123		123	NULL	ditndud100@naver.com			2021-05-02	302, 125,	98ca0eb0aeb15e32
sooyoung	7ff0911782fc97299d0baf16f5621d0912817a4017d7ef0329048c8cf5d8a...	수영	123	123	구직자	ditndud100@naver.com	프론트...		NULL	2,	0501ec7e643c390a
sooyoung1	c0ee8fe7b23fafd4a5312743e411d5b9fe81b2baa48c8e373ca79ba9c7338...	수영	123	123	학생	sooyoung.lee712@gmail.com			NULL		d74ff2b961f99150
sooyoung2	4583fb9b1c93989472e4341de14f1536066a3135a5f7a67d067a4b433a1...	123		123	구직자	sooyoung.lee712@gmail.com	서버	대학교 졸업 예정	2021-05-03		80280dab014afc99
sooyoung22	4627bc89cc28e9d88244f3c552e480679154b2766cfb6d6fd52835ffb7767...	123	SF	111	학생	sooyoung.lee712@gmail.com			2021-05-03		3cf0a5a3ff5911ec
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL




NAKALA
LOGIN

id

password

login

[signup](#)

Sign up  Log in

Log Out  Cloud님

- 로그인을 진행하면 폼에서 아이디와 비밀번호를 받아와 DB에 저장된 비밀번호와 비교.
- DB에는 암호화 된 비밀번호가 저장되어 있어 사용자의 원문 비밀번호 값을 암호화하여 비교.
- 로그인 상태와 로그아웃 상태의 헤더를 다르게 표시

```
conn = getConnection();

String query = "SELECT salt FROM user WHERE u_id=?";
ps = conn.prepareStatement(query);

ps.setString(1, id);

rs = ps.executeQuery();
if(rs.next()) salt = rs.getString("salt");
password = SHA256Util.getEncrypt(o_password, salt);

query = "SELECT name, address, phone, status, email, job_field, education, birth_day FROM user WHERE u_id=? AND password=?";
ps = conn.prepareStatement(query);
```


3.4.1. 주요 기능 - 회원정보 보기 및 수정

NAKALA Log Out 수영님

회원 정보

아이디	123
이름	수영
주소	SF
휴대전화	01011111111
구직상태	구직자
이메일	sooyoung.lee712@gmail.com
분야	서버
최종학력	대학교 졸업
생일	2021-05-06

회원 정보 보기
 회원 정보 수정
 비밀번호 변경
 회원 탈퇴

NAKALA Log Out 수영님

회원 정보 수정

아이디	123
이름	수영 <input type="text" value="수영"/>
주소	SF <input type="text" value="SF"/>
휴대전화	01011111111 <input type="text" value="0101111111"/>
구직상태	구직자 ○ 학생 ● 구직자 ○ 채용담당자
이메일	sooyoung.lee712@gmail.com <input type="text" value="sooyoung.lee712@gmail.c"/>
분야	서버 <input type="text" value="서버"/>

회원 정보 보기
 회원 정보 수정
 비밀번호 변경
 회원 탈퇴

- 로그인 후 본인의 정보를 볼 수 있는 기능이다.
- Session을 이용한다.

```
<% UserVO rvo = (UserVO)session.getAttribute("rvo"); %>
```

- 회원정보를 수정할 수 있는 기능이다.
- DB에 수정한 정보를 넣어주고, 수정한 정보를 다시 Session에 넣어준다.

```
function changeUser(){
    var queryString = $("form[name=changeUserInfo]").serialize();
    $.ajax({
        type: "post",
        url: "changeUserInfo.do",
        data: queryString,
        encode: true,
        success: function(result) {
            alert('<%=rvo.getName() %>'+ "님의 회원정보가 성공적으로 수정되었습니다");
            window.location.href = 'myPage.jsp';
        }
    });
}
```

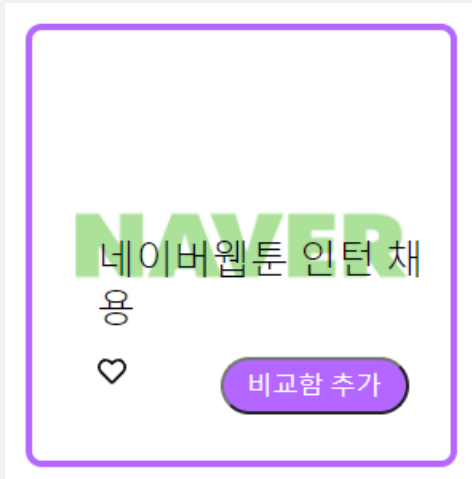
3.4.1. 주요 기능 - 비밀번호 변경, 회원 탈퇴

- 비밀번호를 변경할 수 있는 기능이다.
- 기존의 비밀번호와 같은 비밀번호로는 수정을 하지 못한다.

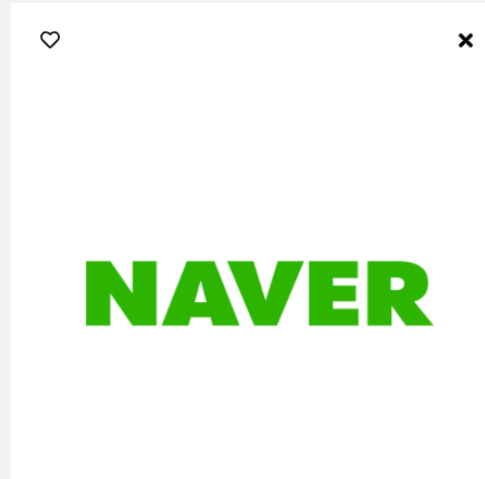
- DB에서 회원의 정보를 삭제하는 회원 탈퇴 기능이다.

```
$("#deleteSubmit").click(function(){
    $.ajax({
        type: "post",
        url: "deleteAccount.do",
        data: "id="+$('#id').val()+"&password="+$('#password').val(),
        success: function(result) {
            alert("탈퇴가 성공적으로 완료되었습니다.");
            window.location.href = 'index.jsp';
        }
    });
});
```

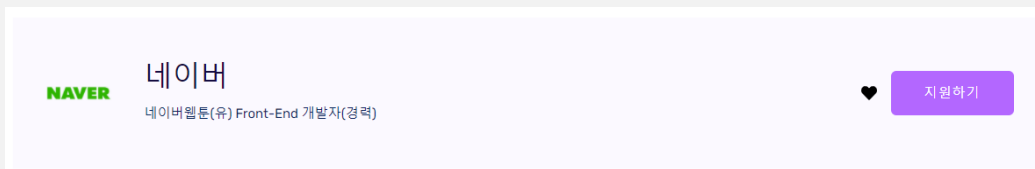
Main Page



비교 Page



상세 Page



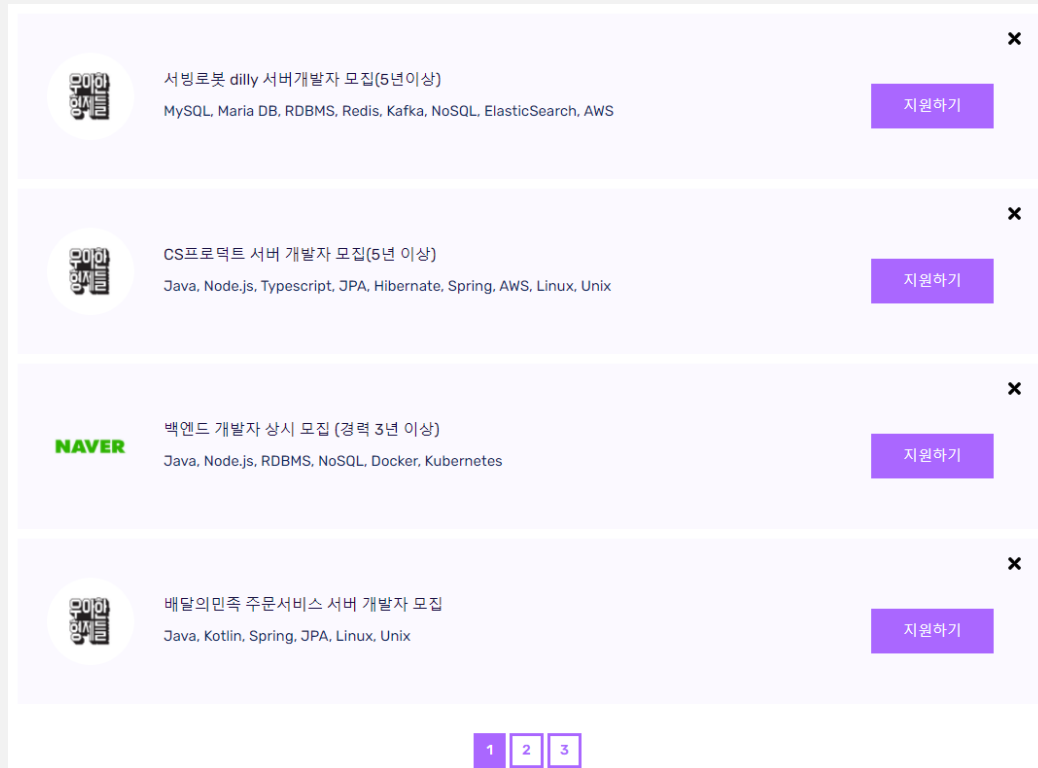
- main, 비교, 상세 Page에서 하트 버튼을 누르면 광고가 scrap된다.
- scrap된 광고는 검은색 하트로 표시된다.

```
$('#compare-table').on('click', '.scrap-button', function() {

    var r_vo = '${rvo}';
    var u_id = '${rvo.u_id}';

    if(u_id == ""){
        alert("로그인이 필요한 서비스 입니다.");
        return;
    }

    var r_id = $(this).attr('id');
    var s = $('#'+ r_id + " svg").attr('id');
    if(s == "svg1"){
        $('#'+ r_id + ' svg').css("background-image", "url(assets/img/elements/heart-solid.svg)");
        $('#'+ r_id + " svg").attr('id', 'svg2');
        $.ajax({
            type: 'post',
            url: 'scrapadd.do', /*응답하는 데이터 타입이 객체일 때 json 이라고 지정*/
            data: 'r_id=' + r_id + '&u_id=' + u_id,
            success: function(result) {
                //callback
            }
        });
    }else{
        $('#'+ r_id + ' svg').css("background-image", "url(assets/img/elements/heart-regular.svg)");
        $('#'+ r_id + " svg").attr('id', 'svg1');
        $.ajax({
            type: 'post',
            url: 'scrapdelete.do?', /*응답하는 데이터 타입이 객체일 때 json 이라고 지정*/
            data: 'r_id=' + r_id + '&u_id=' + u_id,
            success: function(result) {
                //callback
            }
        });
    }
});
```



- Scrap된 공고는 Scrap Page에서 보여준다.

```
var scrap_list = [];
function scrap(u_id){

    $.ajax({
        type: 'post',
        url: 'scrap.do?u_id='+u_id, /*응답하는 데이터 타입이 객체일 때 json 이라고 지정*/
        async: false,
        success: function(result) {
            str = result.split(",");
            scrap_list = str.map(i=>Number(i));
        } //callback
    }); //ajax

    return scrap_list;
}
```

```
var scrapList = [];
scrapList = scrap('${rvo.u_id}');
```


감사합니다

