

Python을 이용한 'HRD-Net' 크롤링 및 JSP 기반의 지도 정보 제공 웹사이트 구축

대한상공회의소 서울기술교육센터
자바기반 빅데이터 시각화 시스템 개발

4조

박재경 · 양재모 · 이재호 · 임선우



주제 및 목표 (03)

일정 및 역할 분담 (04)

Database 구축 (05)

웹페이지 화면 설계 (09)

검색 테이블 생성 (10)

화면 구성 (11)

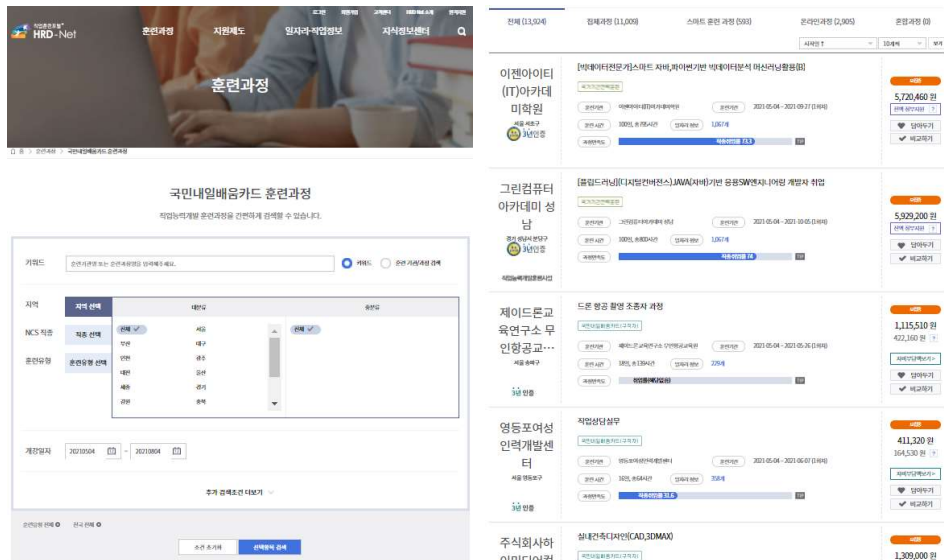
시연 영상 (13)

기대효과 및 한계점 (14)



주제 선정 배경

현재 고용노동부가 운영하는 직업훈련포털인 HRD-Net의 경우,
텍스트 위주의 정보를 제공하고 있기 때문에 가독성·직관성·용이성이 떨어지고,
기관 위치를 따로 검색해야 하는 불편함이 있음
→ 이를 해결하고자 지도 정보를 포함한 훈련과정 검색 웹사이트를 구축하고자 함



프로젝트 목표

- ① **Python**: HRD-Net의 훈련과정 및 기관 정보 수집
- ② **HTML5·CSS3·JavaScript**: 웹페이지 디자인 및 지도 정보 제공
- ③ **JSP·MariaDB**: DB와 서버 연결을 통한 동적 웹사이트 구현

프로젝트 개요

- **사용 언어**: Python / HTML5 / CSS / JavaScript / JSP
- **개발 환경**: Jupyter Notebook / Visual Studio Code / Eclipse
- **데이터베이스**: MariaDB
- **개발 기간**: 21/04/20 - 21/05/07



프로젝트 일정

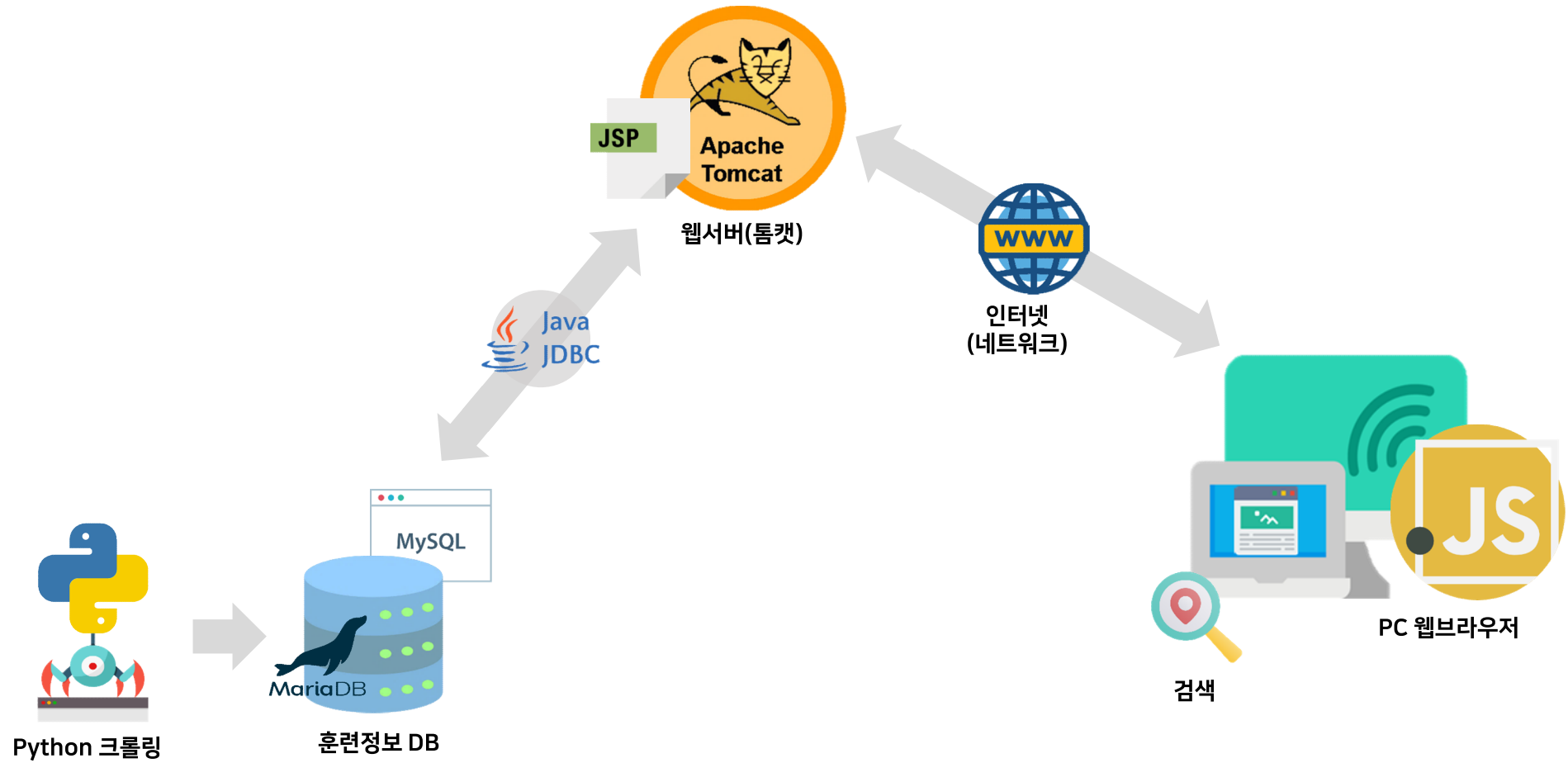
	내용	1	2	3	4	5	6	7	8	9	10	11	12
기획	주제 선정												
	계획 수립 및 역할 분담												
	화면 구조 설계												
개발 (내부)	HRD-Net 크롤링												
	DB 구축												
	서버 연동 및 데이터 입력												
	디버깅												
개발 (외부)	화면 구현												
	PPT 작성												

역할 분담

박 재 경	조장, 웹페이지 디자인(CSS), 발표, 아뱅(아이디어 बैं크라는 뜻),
양 재 모	화면 구현(HTML, JavaScript), 구글맵 달인, 데이터 시각화
이 재 호	서버 연동, DB 구축, 디버깅
임 선 우	크롤링, HTML 구조짜기, PPT 제작



프로젝트 구성도





HRD-Net div 구성

(디지털컨버전스) UI/UX 스마트 콘텐츠 웹기반 융합 응용SW개발자 양성과정(3)

국가기간전략훈련

훈련기관 KH정보교육원 강남지원 훈련기간 2021-05-04 ~ 2021-12-06 (1회차)

훈련시간 150일, 총900시간 일자리 정보 1,086개

과정만족도 직종취업률 81.6 TIP

HRD-Net 크롤링: Python

```
# 빈 리스트 생성
list = []

# 크롤링 할 페이지 수
pagenum = 1
lastpage = 1397

#### 크롤링 후 list에 내용 저장
while(pagenum <= lastpage):

    # url 및 html 정보
    url = f"https://www.hrd.go.kr/hrdp/ti/ptiao/PTIA00100L.do?pageIndex={pagenum}"
    res = requests.get(url, headers = headers)
    res.raise_for_status()
    soup = BeautifulSoup(res.text, "html.parser")

    # div.content
    hrds = soup.find_all("div", "content")

    # 한 페이지 당 10개씩 보기
    pagenum += 1
```

HRD-Net 크롤링: Python

```
for hrd in hrds:
    # 과정
    title = hrd.find("a", "tool_tip_btn")

    if title:
        title = title.get_text().strip()
        title = title.replace(", ", " ")

    # 기관
    place = hrd.find("dl", "item term").get_text().rstrip()[6:]
    place = place.replace(", ", " ")

    # 기간
    term = hrd.find_all("dl", "item term")[1].get_text()
    term = term.replace(" ", "")
    term = term.replace("\n", "")
    term = term.replace(" ", "")
    term = term.replace("2021", "21")
    term = term.replace("-", "/")
    term = term[4:21]

    term_tmp = term.replace("/", "")
    term_start = term_tmp[0:6]
    term_end = term_tmp[7:13]

    # 시간
    time = hrd.find("dl", "item time").get_text()[7:-1]
    time = time.replace("\n", " ")
    time = time.replace(", ", " ")

    # 취업률
    emp_rate = hrd.find("span", "bar")
    if emp_rate:
        emp_rate = emp_rate.get_text().strip()[6:]
    else:
        emp_rate = "-"
    emp_rate = emp_rate.replace("없음", "-")

    # 링크
    val = hrd.find("a", "tool_tip_btn")["onclick"]
    val1 = val[19:36]
    val2 = val[39]
    val3 = val[43:48]
    val4 = val[51:63]
    link = f"https://www.hrd.go.kr/hrdp/co/pcobo/PC0B00100P.do?tracsel={val1}&tracse"
```



훈련기관 위·경도 수집: Python

```
# 구글맵 키
gmaps_key = "AIzaSyBb-qqA00eik_9JZZkQyeQUhVcgG_5Xw"
gmaps = googlemaps.Client(key = gmaps_key)

# 빈 리스트 생성(위도, 경도)
place_adr = []
place_lat = []
place_lng = []

# 위경도 최대최소값
max_lat = 38.0
min_lat = 33.0
max_lng = 132.0
min_lng = 126.0

# 위경도 검색
for place in df_list["place"]:
    tmp = gmaps.geocode(place, language = "ko")

# 구글맵 검색 될 경우
if tmp:
    tmp_adr = tmp[0].get("formatted_address")
    tmp_adr = tmp_adr.replace(", ", "")
    tmp_adr = tmp_adr.replace("'", "")
    tmp_adr = replaceAdr(tmp_adr)
    place_adr.append(tmp_adr)

    tmp_loc = tmp[0].get("geometry")
    tmp_lat = tmp_loc["location"]["lat"]
    tmp_lng = tmp_loc["location"]["lng"]

# 한국 범위 벗어나는 경우 0으로 입력
if(tmp_lat > max_lat or tmp_lat < min_lat or tmp_lng > max_lng or tmp_lng < min_lng):
    place_lat.append("0")
    place_lng.append("0")
# list에 위경도 추가
else:
    place_lat.append(tmp_lat) # 위도
    place_lng.append(tmp_lng) # 경도

# 검색 안 될 경우 0으로 입력
else:
    place_adr.append("0")
    place_lat.append("0")
    place_lng.append("0")
```

csv 파일 생성: Python

```
# 크롤링 결과 dataframe 생성
df_list = pd.DataFrame(list)
df_list = df_list.rename(columns = {0: "title", 1: "place", 2: "term",
                                    3: "term_start", 4: "term_end", 5: "time", 6: "emp_rate", 7: "link"})

# 위경도 수집 결과 dataframe 생성
df_list2 = pd.DataFrame(place_lat)
df_list2 = df_list2.rename(columns = {0: "lat"})
df_list3 = pd.DataFrame(place_lng)
df_list3 = df_list3.rename(columns = {0: "lng"})

# 크롤링, 위경도 dataframe 결합
df = pd.concat([df_list, df_list2, df_list3], axis = 1)

# 이상 데이터 삭제(위경도 0인 행)
df = df.query("lat != '0'")

### 최종 csv 파일 생성
df.to_csv("data_0504.csv", encoding = "utf-8")
```

데이터프레임 예시

```
df.sample(6)
```

	title	place	term	term_start	term_end	time	emp_rate	link	lat	lng
	쉽게 쓰고 편하게 읽히는 2page 보고 서 만들기	원스택원 격평생고 육원	21/05/05-21/06/03	210505	210603	일 중 28시 간	-	https://www.hrd.go.kr/hrdp/co/pcoba/PCOBO0100P...	37.504	127.127
	실무에서 인정받는 신의 한 수 기획/보 고의 기술	에듀튜어	21/05/05-21/06/04	210505	210604	일 중 17시 간	-	https://www.hrd.go.kr/hrdp/co/pcoba/PCOBO0100P...	37.6547	126.773
	컴퓨터활용능력2 급 자격증 취득	우리전산 학원	21/05/06-21/06/03	210506	210603	20일 중80 시간	45.5	https://www.hrd.go.kr/hrdp/co/pcoba/PCOBO0100P...	35.25	128.523
	요양보호사 자격 취득과정	연세요양 보호사고 육원	21/05/06-21/08/05	210506	210805	81일 중240 시간	-	https://www.hrd.go.kr/hrdp/co/pcoba/PCOBO0100P...	37.2336	127.209
	핵심만 콕 바로 쓰 는 문화예술 기획 전략	에듀튜어	21/05/05-21/06/04	210505	210604	일 중 17시 간	-	https://www.hrd.go.kr/hrdp/co/pcoba/PCOBO0100P...	37.6547	126.773
	대기환경기사(필 기)	(주)중앙고 육	21/05/05-21/07/04	210505	210704	일 중 80시 간	-	https://www.hrd.go.kr/hrdp/co/pcoba/PCOBO0100P...	37.3857	126.932



DB 및 테이블 생성: Maria DB

MariaDB

└ python (database)

└ hrd (table): 전체 훈련정보 데이터가 들어있는 테이블

└ tmp_hrd (table): 검색된 훈련정보 데이터를 넣을 테이블

```

MariaDB [(none)]> create database python;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use python;
Database changed
MariaDB [python]> CREATE TABLE hrd(
  -> index1 INT(7) NOT NULL,
  -> title VARCHAR(100) NOT NULL,
  -> place VARCHAR(30) NOT NULL,
  -> term VARCHAR(17) NOT NULL,
  -> term_start INT(6) NOT NULL,
  -> term_end INT(6) NOT NULL,
  -> term_time VARCHAR(20) NOT NULL,
  -> emp_rate VARCHAR(10) NOT NULL,
  -> link VARCHAR(150) NOT NULL,
  -> lat VARCHAR(20) NOT NULL,
  -> lng VARCHAR(20) NOT NULL);
Query OK, 0 rows affected (0.008 sec)

MariaDB [python]> LOAD DATA LOCAL INFILE 'C:/Users/aaaa/Desktop/data_0504.csv'
INTO Table python.hrd FIELDS TERMINATED BY ',';
Query OK, 13310 rows affected, 1061 warnings (0.153 sec)
Records: 13310 Deleted: 0 Skipped: 0 Warnings: 1061

MariaDB [python]> create table tmp_hrd as select * from hrd;
Query OK, 981 rows affected (0.050 sec)
Records: 981 Duplicates: 0 Warnings: 0

```

hrd 테이블 구조 및 예시

```

MariaDB [python]> desc hrd;

```

Field	Type	Null	Key	Default	Extra
index1	int(7)	NO		NULL	
title	varchar(100)	NO		NULL	
place	varchar(30)	NO		NULL	
term	varchar(17)	NO		NULL	
term_start	int(6)	NO		NULL	
term_end	int(6)	NO		NULL	
term_time	varchar(20)	NO		NULL	
emp_rate	varchar(10)	NO		NULL	
link	varchar(150)	NO		NULL	
lat	varchar(20)	NO		NULL	
lng	varchar(20)	NO		NULL	

```

MariaDB [python]> select place, term, emp_rate, lat, lng from hrd limit 20;

```

place	term	emp_rate	lat	lng
IH정보교육원 강남지원	21/05/04-21/12/06	81.6	37.49899300000001	127.032909
인천직업전문학교	21/05/04-21/08/20	56.4	37.4749963	126.6524296
이천아이티(IT)아카데미학원	21/05/04-21/09/27	73.3	37.5021576	127.0243763
한국환경산업기술원	21/05/04-21/07/23	62	37.6104898	126.9335998
디지털융합기술원	21/05/04-21/10/19	-	37.2680919	127.0003279
인원트랜드평생교육원	21/05/04-21/09/14	-	37.4920278	126.7235791
하이미디어아카데미	21/05/04-21/06/16	-	37.4988362	127.0262103
테라컴퓨터학원	21/05/04-21/07/02	43.8	35.8430799	128.6949088
군산영문요리전문학교	21/05/04-21/06/23	-	35.9710234	126.6985066
에프비아이제과제빵커피학원	21/05/04-21/07/07	34	37.6755799	126.7477019
한솔유리커피제과제빵아카데미	21/05/04-21/06/09	45.7	37.654755	127.0600445
경상직업전문학교	21/05/04-21/08/27	36.4	35.8341435	128.5545237
이천IT컴퓨터아카데미	21/05/04-21/07/08	58.7	37.4327035	127.1296961
미래경영교육원	21/05/04-21/06/22	61.8	35.8706744	128.5943736
명문직업전문학교	21/05/04-21/06/18	37.5	37.5438275	126.7236928
이천IT컴퓨터아카데미	21/05/04-21/06/30	55.1	37.4327035	127.1296961
일산에드모소직업전문학교	21/05/04-21/06/15	65.7	37.6351555	126.8313903
대경녹색기술직업전문학교	21/05/04-21/09/28	60	35.8767557	128.5987104
주식회사하이미디어컴퓨터학원	21/05/04-21/09/01	70	37.2783487	127.0178475
주식회사하이미디어컴퓨터학원	21/05/04-21/08/23	-	37.2783487	127.0178475

화면 설계서

과정 ▼	빅데이터	5			
서울 ▼	강서구 ▼				
21-03-01	21-09-01				
검색					
훈련 과정 훈련 기관 훈련 기간					
훈련 과정 훈련 기관 훈련 기간					

1	과정 및 기관 키워드 검색	전체, 과정, 기관의 기준에 따라 키워드를 포함한 훈련과정 검색
2	기관 위치 검색	선택한 주소에 위치한 기관의 훈련과정 검색
3	시작, 종료 기간 검색	선택한 시작, 종료 기간 사이의 훈련과정 검색
4	검색 결과 목록	검색된 훈련과정에 대한 간단한 정보를 목록으로 만들어 표시 클릭 시 기관의 위치로 지도 이동
5	지도 정보	검색된 훈련과정을 지도에 마커로 표시 마커 클릭 시 자세한 정보와 HRD-Net 연결 링크 표시



tmp_hrd 테이블 생성: JSP

```
// 검색 변수 받아오기
String standard = request.getParameter("standard");
String keyword = request.getParameter("keyword");

String loc1 = request.getParameter("loc1");
String loc2 = request.getParameter("loc2");

String ts = request.getParameter("term_start");
int term_start = 0;
if(!ts.equals("")) {
    ts = ts.substring(8,10) + ts.substring(0,2) + ts.substring(3,5);
    term_start = Integer.parseInt(ts);
}

String te = request.getParameter("term_end");
int term_end = 0;
if(!te.equals("")) {
    te = te.substring(8,10) + te.substring(0,2) + te.substring(3,5);
    term_end = Integer.parseInt(te);
}

// sql 쿼리문 전송을 위한 statement 객체 생성
ResultSet rs = null;
Statement stmt = conn.createStatement();

// sql 쿼리문 - 검색 조건에 따른 where절
String sql = "";
sql = kwdSearch(sql, keyword, standard, loc1, loc2, term_start, term_end);

// sql 쿼리문 - 검색 조건에 따라 tmp_hrd 테이블 생성
String sql2 = "CREATE TABLE tmp_hrd AS SELECT * FROM hrd ";
sql2 += sql + ";";

// 쿼리문 실행
stmt.executeUpdate("DROP TABLE tmp_hrd;");
stmt.executeUpdate(sql2);
```

검색 관련 쿼리문 작성 함수: JSP

```
// 혼련 기간
String termSearch(String sql, int term_start, int term_end) {
    if(term_start != 0 && term_end != 0) {
        sql += "AND term_start >= " + term_start + " ";
        sql += "AND term_end <= " + term_end + " ";
    } else if(term_start != 0) {
        sql += "AND term_start >= " + term_start + " ";
    } else if(term_end != 0) {
        sql += "AND term_end <= " + term_end + " ";
    }
    return sql;
}

// 혼련 기관 위치
String adrSearch(String sql, String loc1, String loc2, int term_start, int term_end) {
    if(loc1.equals("0")) {
        sql = termSearch(sql, term_start, term_end);
    } else if(!loc1.equals("0") && !loc2.equals("종분류")) {
        sql += "AND adr LIKE '" + loc1 + "%' ";
        sql += "AND adr LIKE '%" + loc2 + "%' ";
        sql = termSearch(sql, term_start, term_end);
    } else if(!loc1.equals("0")) {
        sql += "AND adr LIKE '" + loc1 + "%' ";
        sql = termSearch(sql, term_start, term_end);
    }
    return sql;
}

// 혼련 과정 및 기관 키워드
String kwdSearch(String sql, String keyword, String standard, String loc1, String loc2,
    int term_start, int term_end) {
    if(keyword.equals("")) {
        sql += "WHERE title IS NOT NULL ";
        sql = adrSearch(sql, loc1, loc2, term_start, term_end);
    } else if(standard.equals("undefined") || standard.equals("all")) {
        sql += "WHERE (title LIKE '%" + keyword + "%' ";
        sql += "OR place LIKE '%" + keyword + "%' ) ";
        sql = adrSearch(sql, loc1, loc2, term_start, term_end);
    } else if(standard.equals("title")) {
        sql += "WHERE title LIKE '%" + keyword + "%' ";
        sql = adrSearch(sql, loc1, loc2, term_start, term_end);
    } else if(standard.equals("place")) {
        sql += "WHERE place LIKE '%" + keyword + "%' ";
        sql = adrSearch(sql, loc1, loc2, term_start, term_end);
    }
    return sql;
}
```



검색창 만들기: JavaScript

```
// 사용자가 검색하는 주소와 키워드
var adr1;
var adr2;

// **** 카테고리 선택 -> 카테고리
function categoryChange(e) {
    var loc_0 = ["종분류"];
    var loc_a = ["종분류", "강남구", "강동구", "강북구", "강서구", "관악구", "광진구",
        "구로구", "금천구", "노원구", "도봉구", "동대문구", "동작구",
        "마포구", "서대문구", "서초구", "성동구", "성북구", "송파구",
        "양천구", "영등포구", "용산구", "은평구", "영등포구", "종로", "중랑구"];
    var loc_b = ["종분류", "강서구", "금정구", "기장군", "남구", "동구", "동래구",
        "부산진구", "북구", "사상구", "사하구", "서구", "수영구",
        "연제구", "영도구", "중구", "해운대구"];
    var loc_c = ["종분류", "남구", "달서구", "달성군", "동구", "북구", "서구", "수성구", "중구"];

    var target = document.getElementById("loc-2");

    if(e.value == "0") var locs = loc_0;
    else if(e.value == "서울") var locs = loc_a;
    else if(e.value == "부산") var locs = loc_b;
    else if(e.value == "대구") var locs = loc_c;

    target.options.length = 0;

    for (loc in locs) {
        var opt = document.createElement("option");
        opt.value = locs[loc];
        opt.innerHTML = locs[loc];
        target.appendChild(opt);
    }

    // 사용자가 선택한 주소 대분류
    adr1 = e.options[e.selectedIndex].text;

    function adrChange(e) {
        // 사용자가 선택한 주소 중분류
        adr2 = e.options[e.selectedIndex].text;
    }

    // **** 달력 관련 ****
    $(function(){
        $("#date1").datepicker();
        $("#date2").datepicker();
    });
}
```

구글맵 띄우기: JavaScript

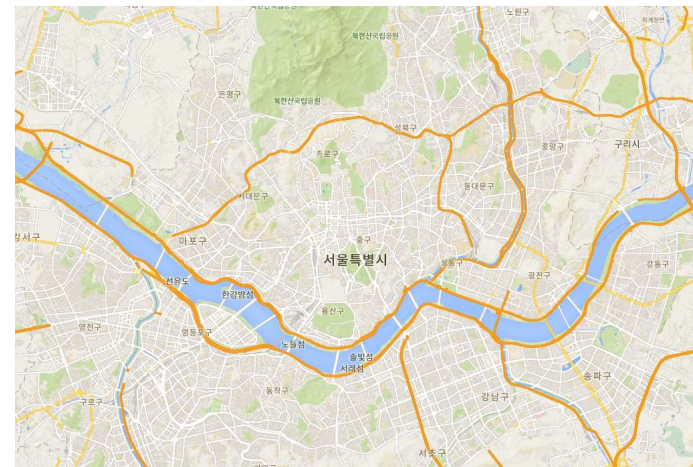
```
// **** 지도 관련 ****
var map;

function initMap() {

    //지도가 보여질 요소 찾아오기
    var e = document.getElementById('map');

    // Google지도 객체 생성 및 요소 보여주기!! (시작위치: 서울 중구)
    map = new google.maps.Map(e, {
        center: {
            lat: 37.5579452,
            lng: 126.9941904
        },
        zoom: 12
    });

    // 검색 버튼 클릭시 세션스토리지에 주소 저장
    document.getElementById('search-btn').addEventListener('click', function() {
        adr = adr1 + " " + adr2;
        sessionStorage.setItem("address", adr);
    });
}
```





검색 결과 마커 찍기: JavaScript

```
// 지오코딩을 위한 지오코더
var geocoder = new google.maps.Geocoder();
// 지도 중심위치 세션스토리지에서 불러오기
var address = sessionStorage.getItem("address");

geocoder.geocode({'address': address}, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
        // 중심위치 위경도 알아내서 위치 이름
        map.setCenter(results[0].geometry.location);
        // **** 인포윈도우
        var infowindow = new google.maps.InfoWindow();

        var marker;
        var locations = [];
        <%
        // db에서 정보 받아오기
        ResultSet rs = null;
        Statement stmt = conn.createStatement();

        String sql = "SELECT * FROM tmp_hrd;";
        rs = stmt.executeQuery(sql);

        int i = 0;
        while(rs.next()) {
            String place = rs.getString("place");
            String lat = rs.getString("lat");
            String lng = rs.getString("lng");
            String link = rs.getString("link");
            String title = rs.getString("title");

            // 위경도 정보를 통해 마커 그리기
            marker = new google.maps.Marker({
                position: new google.maps.LatLng(<%=lat%>, <%=lng%>),
                map: map
            });
            // 클릭 시 팝업 뜨는 것
            google.maps.event.addListener(marker, 'click', (function(marker, i) {
                return function() {
                    //html로 표시할 인포 윈도우의 내용
                    infowindow.setContent("과정: <%=title%><br>기관: <%=place%><br><a href='<%=link%>'>해당사이트이동</a>");
                    //인포윈도우가 표시될 위치
                    infowindow.open(map, marker);
                }
            })(marker, i));
            // 마커 늘렸을 때 지도 변화
            if(marker) {
                marker.addListener('click', function() {
                    // 중심 위치를 클릭된 마커의 위치로 변경
                    map.setCenter(this.getPosition());
                    // 마커 클릭 시의 줌 변화
                    map.setZoom(16);
                });
            }
        }
    }
});
```

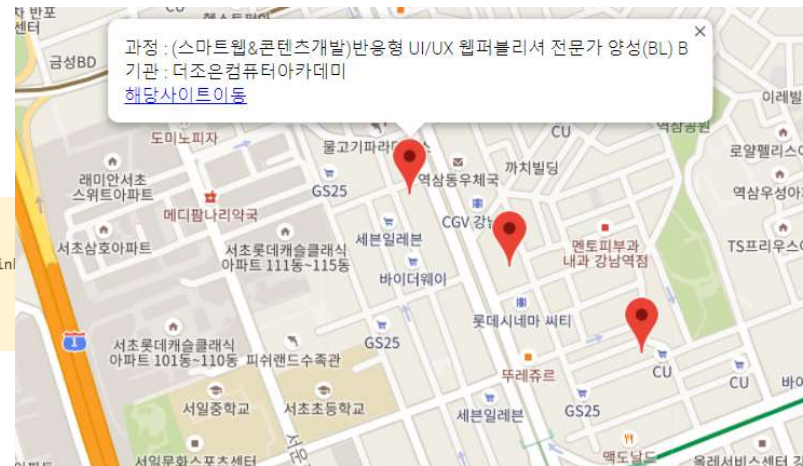
검색 결과 목록 표시: JSP

```
// db에서 리스트에 띄울 정보 가져옴
String sql2 = "SELECT * FROM tmp_hrd;";
rs = stmt.executeQuery(sql2);

while(rs.next()){
    String title = rs.getString("title");
    String place = rs.getString("place");
    String term = rs.getString("term");
    String lat = rs.getString("lat");
    String lng = rs.getString("lng");
    String term_time = rs.getString("term_time");
    String emp_rate = rs.getString("emp_rate");

    // 리스트 출력 부분
    out.println("<div onclick='moveMap(" + lat + ", " + lng + ")'><h4>" + title + "</h4>");
    out.println("<h5>" + place + "<br>");
    out.println(term + " (" + term_time + "<br>취업률: "+emp_rate+"%</h5></div><hr>");
}

stmt.close();
conn.close();
```



[빅데이터전문가]자바 파이썬기반 빅데이터 분석 머신러닝활용

이젠컴퓨터학원
21/05/06~21/09/30 (100일 총795시간)
취업률: 53.6%

파이썬(Python)을 활용한 빅데이터(분석 시각화) 양성 과정

바이트컴퓨터학원
21/05/06~21/07/30 (60일 총240시간)
취업률: -%

[빅데이터 UI 전문가]파이썬과 R을 활용한 빅데이터 UI 개발자

이젠컴퓨터학원
21/05/06~21/10/12 (107일 총850시간)
취업률: -%



```

140    $("#date2").datepicker();
141    });
142
143    // ***** 지도 관련 *****
144    // 지도를 가져옴
145    var map;
146    var geocoder;
147
148    function initMap() {
149
150        //지도가 로드된 후에 실행하기
151        var e = document.getElementById('map');
152
153        // Google지도 검색 창을 드래그로 조작하기!! (←자바: 지을 필요)
154        map = new google.maps.Map(e, {
155            center: {
156                lat: 37.5579452,
157                lng: 126.9941904
158            },
159            zoom: 12
160        });
161    }
162
163    // 검색 창 클릭시 장소검색결과를 표시
164    document.getElementById('search-btn').addEventListener('click', function() {
165        adr = adr1 + " " + adr2;
166        sessionStorage.setItem("address", adr);
167    });
168
169    }
170    </script>
171
172
173    <!-- 지도 영역 -->
174    <div map id="map">
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Console:

```

Tomcat v9.0 Server at localhost (Apache Tomcat/9.0.44@localhost) (2021.5.6 오후 10:45:45)
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual

```



기대효과

- UI를 단순화하여 정보 전달의 용이성을 높임
- 키워드, 지역, 날짜의 검색 과정을 단순화하여 정보 검색 시간을 단축함
- HRD-Net 크롤링 및 연결 링크 제공을 통해 정보의 정확성을 높임
- 검색 결과를 지도와 함께 제공하여 필요한 정보를 한 눈에 볼 수 있음
(따로 지도 검색을 할 필요가 없음)

한계점

- Python과 JavaScript 연동의 어려움으로 실시간 데이터를 통한 자동 최신화 작업을 완료하지 못함
- 시간 부족으로 인해 다양한 Geocoder 기능을 활용하지 못함
(훈련기관과 내 장소의 거리 비교 기능 추가, 마커 클러스터 등)

감사합니다

Python을 이용한 'HRD-Net' 크롤링 및
JSP 기반의 지도 정보 제공 웹사이트 구축

4조

박재경 · 양재모 · 이재호 · 임선우