

필요한 쿼리문

[웹에서 사용하는 기능]

페이지

- 회원가입
- 로그인 (로그아웃 버튼)
- 보고 싶은 채용공고 선택하는 페이지(로그인이 안되었으면 메인?)
- 조건에 맞춰 선택된 채용공고를 보여주는 페이지
- 사용자가 자신의 관심사를 입력하는 페이지
- 사용자의 관심사를 보여주는 페이지
- 관심사에 따라서 채용공고를 보여주는 페이지 (로그인 되어 있으면 메인?)
- 찜 버튼 - 찜한 공고를 모두 보여주는 페이지

기능

- 알림 기능

1. '회원가입' 페이지

: 처음에는 user_id를 auto increament하려고 했으나, 아래와 같은 이유로 관리자 측에서 코드 상에서 이를 해줘야 할 듯 보인다.

```
1 ALTER TABLE user MODIFY user_id INT NOT NULL AUTO_INCREMENT;
```



Query Favorites ▾

Query History ▾

Last Error Message

Cannot change column 'user_id': used in a foreign key constraint 'userID' of table 'JobRecommendSystem.favorite'

```

# idx변수 : user_id에 들어갈 값(회원가입 한번 일어날때마다 idx+=1)
# name변수 : 사용자 이름 담는 input값 ex) 'Paul'
# age변수 : 사용자 나이 담는 input값 ex) 22
# region변수 : 사용자 지역 담는 input값 ex) '서울 종로구'

SET @usr_region = (SELECT region_id FROM region WHERE '서울 종로구' = region);
INSERT INTO user(user_id, `name`, age, region_id) VALUES(101, 'Paul', 22,
@usr_region);

```

실제 파이썬으로 DB연동부 확인

```

import sys
import requests
import base64
import json
import logging
import pymysql

host = "project.catth3znjejo.ap-northeast-2.rds.amazonaws.com"
port = 3306
username = "admin"
password = "[비밀번호]"
database = "JobRecommendSystem"

def main():
    conn = pymysql.connect(host, user=username, passwd=password, db=database, \
        port=port, use_unicode=True, charset='utf8')
    cursor = conn.cursor()

    idx = 14
    name = 'Tom'
    age = 25
    region = '서울 종로구'

    # query = "SET @usr_region = (SELECT region_id FROM region WHERE '{}' = region);
    INSERT INTO user(user_id, name, age, region_id) VALUES({}, '{}', {},
@usr_region);".format(region, idx, name, age)
    query1 = "SET @usr_region = (SELECT region_id FROM region WHERE '{}' = region);
".format(region)
    query2 = "INSERT INTO user(user_id, name, age, region_id) VALUES({}, '{}', {},
@usr_region);".format(idx, name, age)
    cursor.execute(query1)
    cursor.execute(query2)
    conn.commit()
    idx += 1
    print(cursor.fetchall()) # fetchall() 메서드는 모든 데이터를 한꺼번에 클라이언트로 가져올
때 사용된다

```

```

print("success")
exit(0)

main()

```

JobRecommendSys
Select Database
TABLES

- career
- company
- favorite
- holiday_tp_nm
- jobs_cd
- license
- notice
- post
- region

```

ubuntu@ip-172-31-34-164:~/DB$ python3 test.py
()
success
ubuntu@ip-172-31-34-164:~/DB$

```

user_id	name	age	region_id
0	Paul	22	2 →
1	Paul	22	2 →
14	Tom	25	2 →
148	Paul	22	2 →
13847	Paul	22	2 →
13848	Paul	22	2 →

2. '채용공고' 페이지

1) 우선 예시로 notice테이블에 정보를 하나 넣고,

```

INSERT INTO notice(notice_id, company_id, title, sal_tp_nm, max_sal, min_sal,
holiday_tp_nm, min_edubg, career, wanted_info_url) VALUES(2, 3, '전산 담당자 모집', '연
봉', 32000000, 29000000, '주5일근무', '학력무관', '경력',
'http://www.work.go.kr/empDetailRedirect.do?wantedAuthNo=K151422006020059')

```

2) 다음의 내용을 조회한다.

- 회사 이름
- 채용 제목
- 급여 형태
- 연봉(최대, 최소)
- 근무 형태
- 학력 조건
- 경력 여부

```

SELECT nt.title, cp.company, nt.sal_tp_nm, nt.max_sal, nt.min_sal, nt.holiday_tp_nm,
nt.min_edubg, nt.career FROM notice AS nt
JOIN company AS cp
ON nt.company_id = cp.company_id

```

title	company	sal_tp_nm	max_sal	min_sal	holiday_tp_nm	min_edubg	career
전산 담당자 모집	(주)엄지식품	연봉	32000000	29000000	주5일근무	학력무관	경력

3. '찜한 공고' 페이지

: favorite테이블에서 appliedDate에 대한 컬럼이 사용자가 '좋아요' 를 누른 time이라고 판단되어, 기존 테이블을 만드는 쿼리문을 저장해둔 채, appliedDate컬럼을 CURRENT_TIMESTAMP로 변경하였다.

[기존 favorite 테이블 생성 쿼리문]

Field: "Create Table" – VARCHAR(1024) NOT NULL utf8

Edit All Fields in Pop-up Sheet

```
CREATE TABLE `favorite` (
  `user_id` int(11) NOT NULL,
  `notice_id` int(11) NOT NULL,
  `appliedDate` datetime DEFAULT NULL,
  PRIMARY KEY (`user_id`,`notice_id`),
  KEY `noticeID_idx` (`notice_id`),
  CONSTRAINT `noticeID` FOREIGN KEY (`notice_id`) REFERENCES `notice` (`notice_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `userID` FOREIGN KEY (`user_id`) REFERENCES `user` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

```
ALTER TABLE favorite DROP appliedDate;
ALTER TABLE favorite ADD COLUMN appliedDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP();
INSERT INTO favorite(user_id, notice_id) VALUES(0, 2)
SELECT * FROM favorite
```

[결과]

user_id	notice_id	appliedDate
0	2	2020-06-03 11:42:08