Job Recommendation Engine - Hunt4Jobs

Group 6
Ankita Naikdalal
Heet Palod
Kulraj Singh Kohli
Vaidehi Patil
Varun Kandukuri

UNIVERSITY of WASHINGTON

Problem Statement

Design an online job portal that will provide a platform for job-seekers to get job recommendations based on the job-seekers skillset.

Motivation

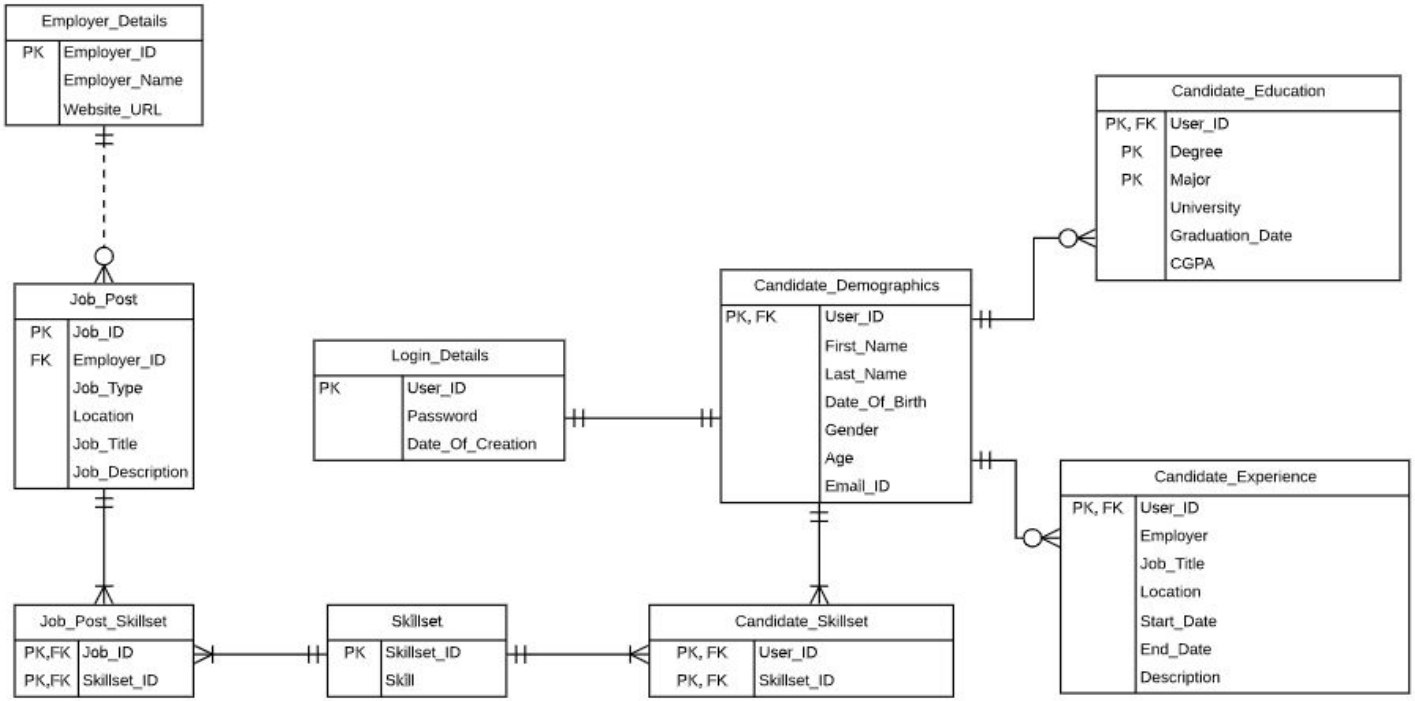
One of the primary functions of websites like LinkedIn, Glassdoor, etc. is to try and match job-seekers to appropriate postings based on their interests and skill sets. This is an exercise that many of us students go through on a frequent basis in our job-hunt.

Given that there are thousands of postings every single day, we wanted to create a platform that would help job-seekers to navigate through these listings and find appropriate jobs for them in an efficient manner.

Core Functionalities

- Candidate will be able to sign-up and register on the website.
- Candidate will be able to login and enter his academic experience, work experience and skillset on the website.
- Candidate will be able to view a list of job recommendations matching his own skill set.
- Website administrator will be able to perform basic CRUD operations on candidate profile/personal details.

Entity-Relationship Diagram



Method and Technologies Used

- Job portal website will be created using HTML5, CSS3, PHP, JavaScript and XAMPP
- phpMyAdmin (MySQL) for managing and storing data
- Python Selenium for web scraping job posting data from job portals

Code Snippet for Web Scraping

```
"source": [
 "from selenium.webdriver import Chrome\n",
 "import time\n",
 "#connect to chrome\n",
 "\n",
 "browser = Chrome()\n",
 "\n",
 "#open web page\n",
 "\n",
 "browser.get('https://www.indeed.com/jobs?q=&l=Seattle%2C+WA')\n",
 "time.sleep(10)\n",
 "\n",
 "# perform navigation clicks to desired page\n",
 "\n",
 "browser.find_element_by_xpath('//button[@class=\"mat-button guided-search-button ng-star-inserted\"]').click()\n",
 "time.sleep(2)\n",
 "browser.find_element_by_xpath('//div[@class=\"guided-search-option-name\" and text()=\" Medical Specialties \"]').click()\n",
 "time.sleep(2)\n",
 "browser.find_element_by_xpath('//div[@class=\"guided-search-option-name\" and text()=\" Medical Care \"]').click()\n",
 "time.sleep(2)\n",
 "browser.find_element_by_xpath('//div[@class=\"guided-search-option-name\" and text()=\" Hospitals \"]').click()\n",
 "time.sleep(5)\n",
 "browser.find_element_by_xpath('//div[@class=\"dropdown-container applied-filters\"]').click()\n",
 "time.sleep(2)\n",
 "browser.find_element_by_xpath('//span[@class=\"ng-star-inserted\" and text()=\" 100 miles \" ]').click()\n",
```

Code Snippet for Job Matching

Pulling data to Python to perform matching

In [ ]:

```
cnx = mysql.connector.connect(host = '127.0.0.1',database='test',user='root',password='')
conn = cnx.cursor()
jobs_desc = """
SELECT * FROM job_post;
"""
skills = """SELECT * FROM skillset;"""

def ret_results(q):
    conn.execute(q)
    ls = []
    for x in conn:
        ls.append(x)
    return ls

jds = ret_results(jobs_desc)
skill_ls = ret_results(skills)
```

Code Snippet for Job Matching

Skills for each job ID

In [10]:
```
import collections
cdt = collections.defaultdict(list)
x = 0
for id in job_ids:
    jd = test[x]
    x += 1
    jd_skills = []
    for word in jd.split():
        if word in skls_st:
            jd_skills.append(word)
    cdt[id] = jd_skills

cdt
```

Out[10]:
```
defaultdict(list,
            {711: ['Fitness',
              'PHP',
              'SQL',
              'Tableau',
              'Azure',
              'AWS',
              'Leadership'],
             712: ['PHP', 'CSS', 'NoSQL', 'SSIS', 'MongoDB', 'Django'],
```

Code Snippet for Job Matching

Inserting skills for each job

In [ ]:

```
st = ''
for job_id, skills in cdt.items():
    for skl in skills:
        qr = ("INSERT INTO job_post_skillset (Job_ID, Skillset_ID) VALUES ({},{})".format(job_id, skill_id_dt[skl]))
        st += qr
        conn = cnx.cursor()
        conn.execute(qr)
```

Product Demo

Limitations

- Due to time constraints, we have currently web-scraped about 1000 job-postings from the Seattle area for our project.
- We have created a portal only to be used by candidates and not the employers. Our employer data is auto-generated.
- We have created the job portal website on local host and plan to host the same on cloud in due course of time.

Future Work

- Migration to AWS; making the web portal centralized.
- Employer registration and management.

Thank You

*Have a **PAWsome Summer** Break!*
Stay healthy, stay safe!