

# **3460-678 : DATA INTEGRATION**

## **Project Report**

Prof. En Cheng  
Team *Seeker*  
Sai Jyothi Ongole  
so70@uakron.edu

**Project Title:** *Career Compass*

### **Abstract:**

A web application is developed using Python web scraping technique and dataframes. Jobs data is extracted in automated way by integrating that information from multiple websites (Indeed and LinkedIn) into one. The user will be able to easily identify job postings from multiple websites and also whether any of his/her LinkedIn connections are working in the job postings' company or not. The only limitation of this project is that the user must have LinkedIn account to use this application. The goal of this project is to enable user to easily identify his/her friends from LinkedIn to get referrals and to know more about the job posted details to go further in the interview process even though the job postings are not from LinkedIn website. Job posting details include job title, company name, posted date and application link.

**Data Source:** Job posting websites – Indeed, LinkedIn

**Front End:** Flask, bootstrap, css, html

**Programming Language:** Python, jQuery

**Python Libraries:** requests, BeautifulSoup, pandas, numpy, json, Selenium

### **Introduction:**

Here is the brief introduction of the techniques used in the project.

#### Webscraping:

Webscraping is the process of gathering information from the internet. Not every website has an API to fetch content. Without an API, extracting the HTML, or scraping, might be the only way to get the website content.

#### BeautifulSoup:

Beautiful Soup is a Python library that makes it easy to scrape information from web pages by pulling data out of HTML and XML files and store the scraped data in Python dataframes or lists.

#### Selenium:

Selenium is a powerful tool for controlling web browsers through programs and performing browser automation. It mimics user actions like scrolling, clicking buttons or links etc.

## Implementation:

### Webscraping using BeautifulSoup:

- BeautifulSoup and Selenium are not available in Python by default. They have to be installed separately. For my project, I have installed using *pip* command.
- Basic steps involved in webscraping through BeautifulSoup.
  - Extract the HTML content using the *requests* library.
  - Analyze the HTML structure and identify the tags which are required to get our data.
  - Extract the tags using BeautifulSoup and store the data in Python dataframes.
- Fetching data from Indeed. Below is the sample code snippet. 5 pages of job posting will be iterated and the data fetched is posted in the past 24 hours.

```
def fetch_indeed_jobs(what, where):  
    indeed_url = fetch_indeed_URL(what, where)  
    final_jobData = []  
    for p in range(5):  
        url=indeed_url+'&start={}'.format(p*10)  
        #print(url)  
        response = requests.get(url)  
        #print(indeed_url)  
        soup = BeautifulSoup(response.text, 'html.parser')  
        job_cards = soup.find_all('a', class_='tapItem')  
        #print(job_cards)  
        for each_card in job_cards:  
            jobData = get_indeed_job_details(each_card)  
            #print(jobData)  
            final_jobData.append(jobData)  
        p+=1  
  
    indeed_jobs = pd.DataFrame(final_jobData)  
    return indeed_jobs
```

- URL of indeed is fetched using requests library. Then BeautifulSoup is used to get the HTML tags required for our content which are job title, company name, job posted date, application link in this project.
- Separate functions have been defined for code reusability and readability for each task it does.
- Function *fetch\_indeed\_URL()* retrieves Indeed website URL which is to be scraped by taking inputs from user like Company name/Role and Location. URL can be modified as per our requirement like in this project data fetched is data posted in the past 24 hours and this is done by inserting '*fromage=1*' in the URL.
- Function *fetch\_indeed\_jobs()* iterates through multiple pages to scrape the data. In this project, up to 5 pages data is fetched.

- Function *get\_indeed\_job\_details()* extracts further more tags required for the content like job title, company name, job posted date, application url and stores this data into dataframe by adding each job posting data row wise.
- LinkedIn job posting data is also fetched in the similar manner and the functions are defined separately for each task. *fetch\_linkedin\_URL()*, *fetch\_linkedin\_jobs()* and *get\_linkedin\_job\_details()* are the functions used to scrape the LinkedIn website.

### **Webscraping using Selenium:**

- Scraping job postings data is a dynamic process where as scraping LinkedIn connections is like a batch process. This is because a user's LinkedIn connections do not change regularly and this batch process can be done once in a week or month based on user's choice.
- Chrome webdriver needs to be installed for Selenium and is to be saved in the folder where the project is saved.
- Several functions have been defined for each task to maintain code reusability and to understand code easily.
- Function *login()* does the intial and basic step in scraping LinkedIn connections which is logging into user's LinkedIn. If the given username and password doesn't match, it will show an error.
- Function *scrap\_basic()* retrieves all the connections details like name, headline, profile url.
- For company names they are working, every profile needs to be opened and check for Experience section of their LinkedIn profile to fetch the company names. This will be done using the function *scrap\_companies()*.
- Tasks like scrolling and waiting is implemented in the code so that data fetching can be done only after page is loaded completely. Basically we are telling Selenium to wait for the page to load completely before data scraping.
- For some profiles company name doesn't exist because they are either students or who are currently not working. In that case instead of reporting null, 'Not Working' is appended to the final data for such cases.
- After retrieving all the data of LinkedIn Connections, the data is stored in the form of CSV file from where data can be integrated with the data fetched from job postings data of Indeed and LinkedIn.

### **Implementation of Data Integration:**

- After retrieving the job postings data based on the user input checkboxes and if the user wants to see LinkedIn connections, data will be merged using merge method in python dataframes pandas library.

- The data is joined based on column Company and it is left join because we want all the job postings to be displayed and the connections are to be displayed only if any opening is there in any connection's currently working company.

```

if(connectionsFlag == 'true'):
    df_conn = pd.read_csv('connections.csv')
    df_conn['Companies'] = df_conn['Companies'].str.replace('\n', '').str.replace('<!-->', '').str.strip()
    df_conn['Name'] = df_conn['Name'].str.replace('\n', '')
    df_conn.drop('Headline', axis=1, inplace=True)
    #df_conn['Headline'] = df_conn['Headline'].str.replace('occupation','').str.replace('\n','')
    df_conn = df_conn.rename(columns={'Companies': 'Company', 'Name': 'Connection Name', 'Link': 'Profile URL'})
    df_conn['Profile URL'] = '<a href="' + df_conn['Profile URL'].astype(str) + '" target="_blank"> url </a>'
    result = pd.merge( result,df_conn, how='left', on = 'Company')

result = result.replace(np.nan, '', regex=True) #.drop_duplicates(inplace=True)
return jsonify(
    my_table=json.loads(result.to_json(orient="split"))["data"],
    columns=[{"title": str(col)} for col in json.loads(result.to_json(orient="split"))["columns"]])

```

- This is similar to SQL left join on the condition matching company names in both tables which are dataframes here.
- Instead of directly displaying the integrated data, I have used jsonify function in Flask to convert the data frame into json format response.
- Data cleaning like trimming, removing special characters, job applications urls are done just before displaying the data for efficiency.
- Any job application URL or LinkedIn profile URL will open in new tab when user clicks on that because of attribute 'target=\_blank'.
- Duplicate rows are removed using *drop\_duplicates* method in python and also *numpy* is used to remove duplicate rows.
- Search functionality is provided using jQuery DataTable in Flask so that user can easily search for anything even after getting all job details.
- To display new results every time, old table is destroyed and then new table is shown in UI.

```

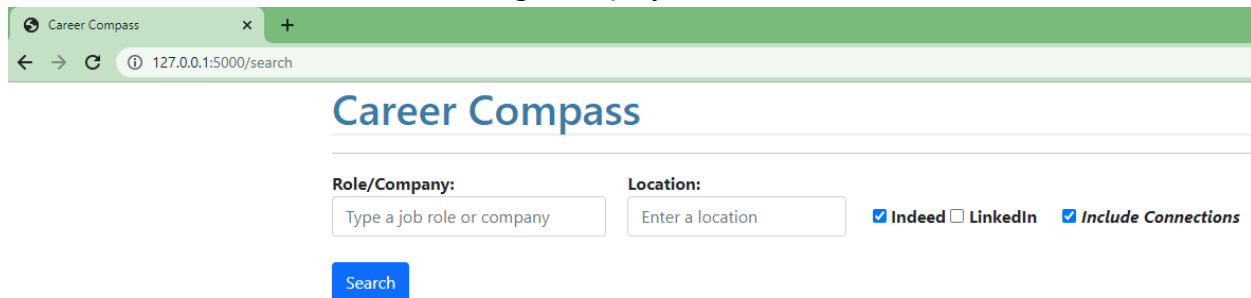
$(document)
    .ajaxStart(function () {
        $loading.show();
    })
    .ajaxStop(function () {
        $loading.hide();
    });
$.fn.dataTable.ext.errMode = 'throw';
$(document).ready(function() {
    var table = null;
    $('#search').bind('click', function() {
        $.getJSON('/result', {
            job: $('input[name="job"]').val(),
            loc: $('input[name="location"]').val(),
            indeed: $('input[name="indeed"]').is(':checked'),
            linkedin: $('input[name="linkedin"]').is(':checked'),
            connections: $('input[name="connections"]').is(':checked')

```

```
}, function(data) {  
  // $("#elements").text(data.number_elements);  
  //alert(data.data)  
  if (table !== null) {  
    table.destroy();  
    table = null;  
    $("#result_table").empty();  
  }  
  table = $("#result_table").DataTable({  
    data: data.my_table,  
    columns: data.columns  
  });  
});
```

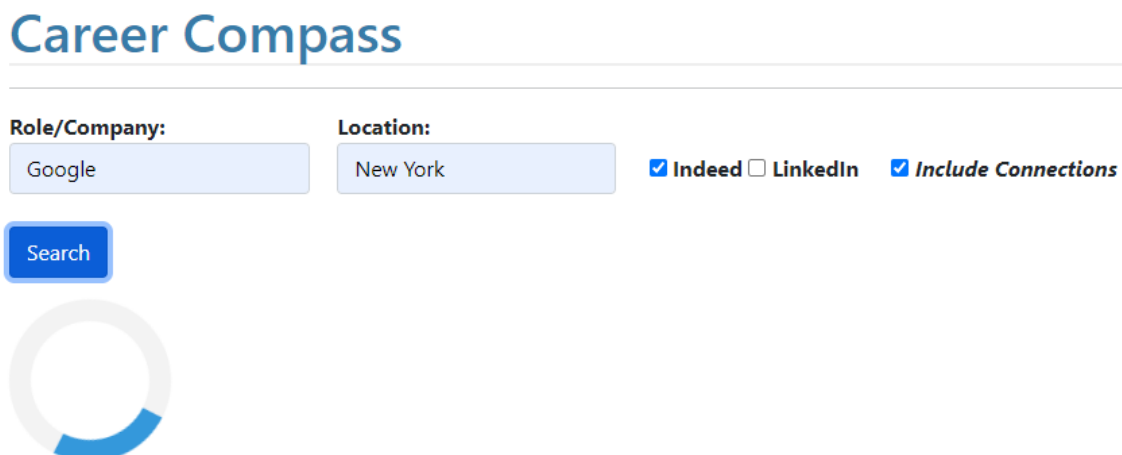
### Implementation of User Interface and Application Functionalities:

- Flask application is run in virtual environment. My project runs on the URL <http://127.0.0.1:5000/search>.
- *search.html* file contains the form elements and jQuery datatable.
- Home screen contains total 5 user choices, out of which two are text boxes and the other three are checkboxes. The results will get displayed based on the user selection.



*Application Home Screen*

- All these checks are performed in *app.py* file which is in Flask application.
- *app.css* file is used for loading div functionality. Instead of null response while data loading, user can be able to know that the data is being loaded.



*Home screen when data loading is in process*

- Loading animation is always displayed below the search button and above the data table if present.


## Career Compass

**Role/Company:**

**Location:**

☒ Indeed ☒ LinkedIn ☒ Include Connections

Search



Show  entries

Search

Role	Company	Location	Job URL
(USA) Software Engineer II	Walmart	Bentonville, AR	<a href="#">url</a>
(USA) Software Engineer II	Walmart	Bentonville, AR	<a href="#">url</a>

- Bootstrap is used to define table properties like on mouse hover, row color will change and alternate rows are displayed in different color.

Role	Company	Location	Job URL	Posted	Connection Name	Profile URL
CPU Verification Infrastructure Engineer	Google	Hybrid remote in Mountain View, NY	<a href="#">url</a>	PostedToday	Sandhya Munagala	<a href="#">url</a>
Data Scientist, Business and Marketing, Google Customer Solu...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Data Scientist, Google Cloud Finance Business Intelligence	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedToday	Sandhya Munagala	<a href="#">url</a>
Director, User Experience Researcher, Coherence Initiative	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Financial Analyst, Global Customer Solutions, Ads Finance Ac...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedToday	Sandhya Munagala	<a href="#">url</a>
Financial Crimes Investigations Analyst, Payments Compliance	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Global Manager, Industry Experts, Customer Success Accelerat...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Lead Product Growth Analyst, Ads Marketing Automation	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedJust posted	Sandhya Munagala	<a href="#">url</a>

- Sorting the data based on column can be done by clicking the up arrow and down arrows symbols present beside each column name.

Role	Company	Location	Job URL	Posted	Connection Name	Profile URL

- Showing limited number of entries per page and displaying page numbers functionalities are implemented using jQuery DataTable in Flask.

Show  entries

Role <span>↑↓</span>	Company <span>↑↓</span>	Location

Showing 1 to 10 of 23 entries

Previous **1** 2 3 Next

- User can search any particular word using search functionality present just above the table after retrieving the job postings details. This search functionality is very quick in nature.

### Career Compass

Role/Company:  Location:  ☒ Indeed ☐ LinkedIn ☒ Include Connections

Show  entries Search:

Role <span>↑↓</span>	Company <span>↑↓</span>	Location <span>↑↓</span>	Job URL <span>↑↓</span>	Posted <span>↑↓</span>	Connection Name <span>↑↓</span>	Profile URL <span>↑↓</span>
Financial Analyst, Global Customer Solutions, Ads Finance Ac...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedToday	Sandhya Munagala	<a href="#">url</a>
Financial Crimes Investigations Analyst, Payments Compliance	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Lead Product Growth Analyst, Ads Marketing Automation	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedJust posted	Sandhya Munagala	<a href="#">url</a>
Strategic Financial Analyst, Google Customer Solutions, Fina...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>

Showing 1 to 4 of 4 entries (filtered from 23 total entries)

Previous **1** Next

- Table display is dynamic in nature based on user inputs. LinkedIn connections data is displayed only when user selects the LinkedIn Connections checkbox. Otherwise the related columns are not displayed.

---- to be continued on next page

## Career Compass

**Role/Company:**  **Location:**  ☒ Indeed ☐ LinkedIn ☒ Include Connections

Show  entries Search:

Role	Company	Location	Job URL	Posted	Connection Name	Profile URL
CPU Verification Infrastructure Engineer	Google	Hybrid remote in Mountain View, NY	<a href="#">url</a>	PostedToday	Sandhya Munagala	<a href="#">url</a>
Data Scientist, Business and Marketing, Google Customer Solu...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Data Scientist, Google Cloud Finance Business Intelligence	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedToday	Sandhya Munagala	<a href="#">url</a>
Director, User Experience Researcher, Coherence Initiative	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Financial Analyst, Global Customer Solutions, Ads Finance Ac...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedToday	Sandhya Munagala	<a href="#">url</a>
Financial Crimes Investigations Analyst, Payments Compliance	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Global Manager, Industry Experts, Customer Success Accelerat...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Lead Product Growth Analyst, Ads Marketing Automation	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedJust posted	Sandhya Munagala	<a href="#">url</a>
Manager, Customer Success Leads, gCare Product Support	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>
Manager, Display and Programmatic, gCare	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago	Sandhya Munagala	<a href="#">url</a>

Showing 1 to 10 of 23 entries

Previous **1** 2 3 Next

*Data displayed when LinkedIn Connections checkbox is checked.*

## Career Compass

**Role/Company:**  **Location:**  ☒ Indeed ☐ LinkedIn ☐ Include Connections

Show  entries Search:

Role	Company	Location	Job URL	Posted
CPU Verification Infrastructure Engineer	Google	Hybrid remote in Mountain View, NY	<a href="#">url</a>	PostedToday
Data Scientist, Business and Marketing, Google Customer Solu...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago
Data Scientist, Google Cloud Finance Business Intelligence	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedToday
Director, User Experience Researcher, Coherence Initiative	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago
Financial Analyst, Global Customer Solutions, Ads Finance Ac...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedToday
Financial Crimes Investigations Analyst, Payments Compliance	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago
Global Manager, Industry Experts, Customer Success Accelerat...	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago
Lead Product Growth Analyst, Ads Marketing Automation	Google	New York, NY (Chelsea area)	<a href="#">url</a>	PostedJust posted
Manager, Customer Success Leads, gCare Product Support	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago
Manager, Display and Programmatic, gCare	Google	New York, NY (Chelsea area)	<a href="#">url</a>	Posted1 day ago

Showing 1 to 10 of 23 entries

Previous **1** 2 3 Next

*Data displayed when LinkedIn Connections checkbox is not checked.*



- If no one from user's LinkedIn connections work in the job position company empty cells are displayed in that place.

## Career Compass

Role/Company:

Location:

☐ Indeed ☒ LinkedIn ☒ Include Connections

Show  entries

Search:

Role	Company	Location	Job URL	Posted	Connection Name	Profile URL
(USA) Software Engineer II	Walmart	Bentonville, AR	<a href="#">url</a>	17 minutes ago	Jake Schaller	<a href="#">url</a>
(USA) Software Engineer II	Walmart	Bentonville, AR	<a href="#">url</a>	24 hours ago	Jake Schaller	<a href="#">url</a>
(USA) Software Engineer III	Walmart	Bentonville, AR	<a href="#">url</a>	24 hours ago	Jake Schaller	<a href="#">url</a>
2022 Digital Academy Intern – Software Engineer	Adobe	Arkansas, United States	<a href="#">url</a>	6 hours ago		
Angular Front End Developer	Arrow Electronics, Inc.	Remote in Fayetteville, AR 72701	<a href="#">url</a>	PostedJust posted		
Application Engineer	Trane Technologies	Fort Smith, AR	<a href="#">url</a>	23 hours ago		
Arkansas Administrative Office of the Courts - Website Devel...	Arkansas Judiciary	Little Rock, AR	<a href="#">url</a>	Posted1 day ago		

## Challenges:

- First thought about webscraping is using only BeautifulSoup, but while scraping LinkedIn connections it is difficult to do user actions like scrolling and clicking. So, Selenium is used to scrape LinkedIn connections data.
- Initially worked on retrieving job position last date to apply details also, but most of the job positions do not have deadlines to apply. So, removed that scraping part for application efficiency.
- Initially application was developed based on form submission, later modified it to ajax query for quick request/response and user understandability.
- Instead of implementing separate search functionality, jQuery DataTable search functionality is implemented as it is very quick and can get results from all pages.
- Loading animation, error messages display functionalities are implemented instead of throwing random errors for user.
- Initially data cleaning is done row wise while fetching the data from the websites which is very slow, but later modified it by doing data cleaning just before displaying the data in the UI. This process is very fast and efficient way.
- Some deprecation warnings may be displayed while running Selenium which may be ignored.

### **Limitations of the application:**

- The only limitation in this project is that the user must have LinkedIn profile. Because the basic idea of this project is to know whether user have any contacts in any company for job positions that are posted on internet. LinkedIn provides this feature by default, we can go to that company page in LinkedIn and check for any connections working there. But its not possible for the job positions that are posted in websites other than LinkedIn. This application provides that feature.

### **Future scope:**

- Application development can be extended by fetching data from more job posting websites like Monster, Glassdoor etc.
- Instead of mapping the job openings data only with LinkedIn connections, it can also be mapped with the profile we are following in LinkedIn. Because we cannot connect to everyone in LinkedIn, we just follow some profiles for job updates so that we can reach out to them later.

### **What I have learned in this project:**

- I have learnt new technologies/concepts like Flask, webscraping techniques using BeautifulSoup and Selenium, ajax request/response, jQuery DataTable, usage of jsonify function.

### **References:**

- <https://beautiful-soup-4.readthedocs.io/en/latest/>
- <https://www.geeksforgeeks.org/selenium-python-tutorial/>