CitySearch Web Scraping

Importing libraries

In [ ]:

```
import pandas as pd
import time
import re

from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException
```

Navigating to the main page of CitySearch

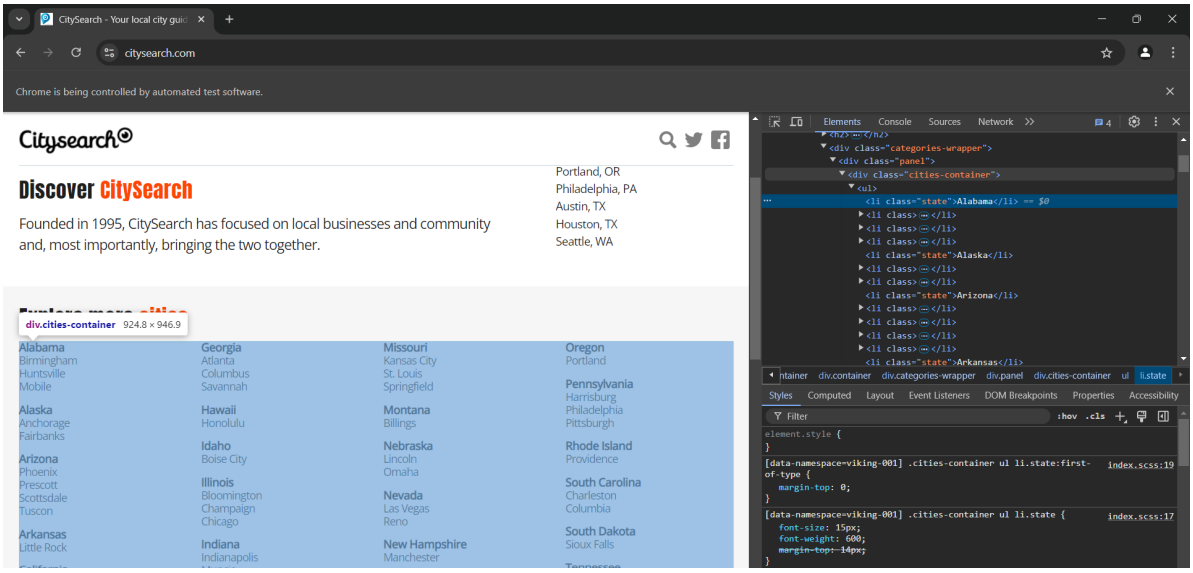
In [ ]:

```
driver = webdriver.Chrome()
driver.get("https://www.citysearch.com/")
```

Extracting the links to individual cities

In [ ]:

```
container = driver.find_element(By.CSS_SELECTOR, "div.cities-container
cities = container.find_elements(By.CSS_SELECTOR, "li:not([class*='sta

city_links = [city.get_attribute("href") for city in cities]
state, city = re.search("(?<=\.com\/).*", city_links[0]).group().split
```

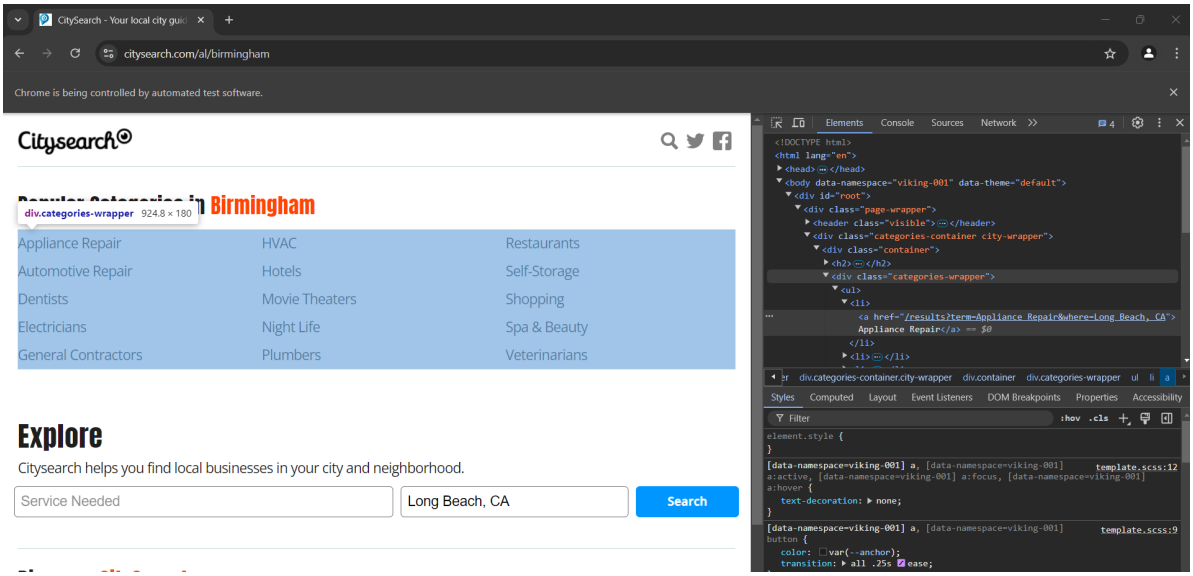


Navigating to a city link and gathering popular jobs

If we have keywords of specific industries we're interested in, I can iterate over them instead of iterating over popular industries. Also, if we have a list of states or cities we're interested in, I can also iterate over those.

In [ ]:

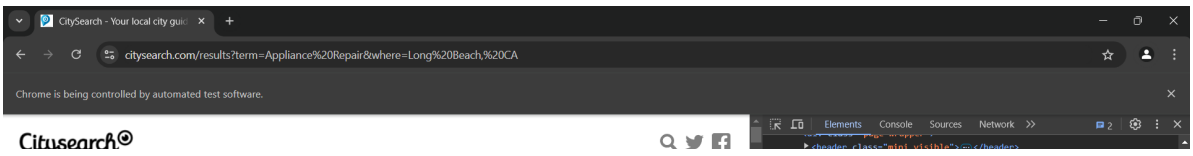
```
driver.get(city_links[0]) # as an example will be going through the jo
try:
    elem = WebDriverWait(driver, 10).until(EC.presence_of_element_loca
except TimeoutException:
    print("Timed out waiting for page to load")

popular_jobs = driver.find_elements(By.CSS_SELECTOR, 'div.categories-w
popular_jobs_links = [job.get_attribute("href") for job in popular_job
```

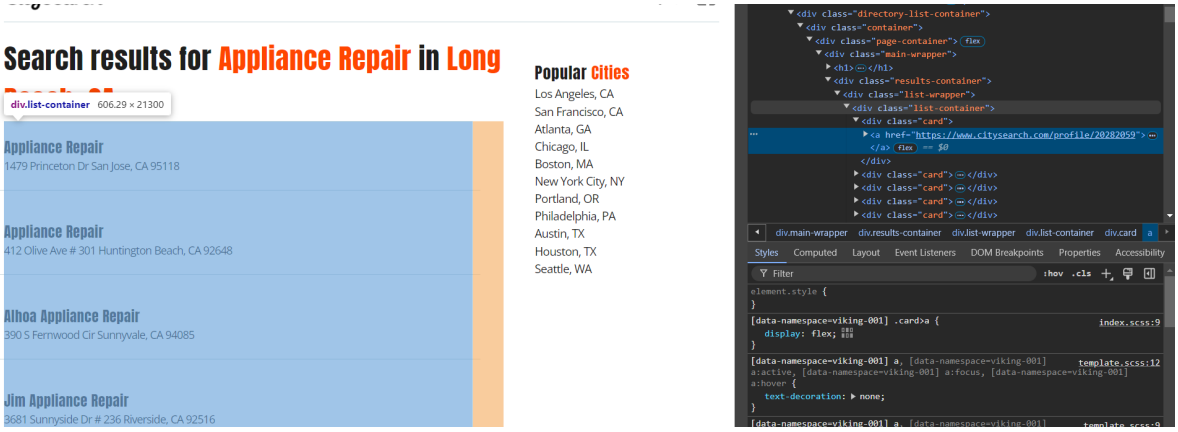


Navigating to first popular job and extracting links to jobs

In [ ]:

```
driver.get(popular_jobs_links[0])

try:
    elem = WebDriverWait(driver, 10).until(EC.presence_of_element_loca
except TimeoutException:
    print("Timed out waiting for page to load")

job_cards = driver.find_elements(By.CSS_SELECTOR, "div.list-container
job_cards_links = [job.get_attribute("href") for job in job_cards]
```

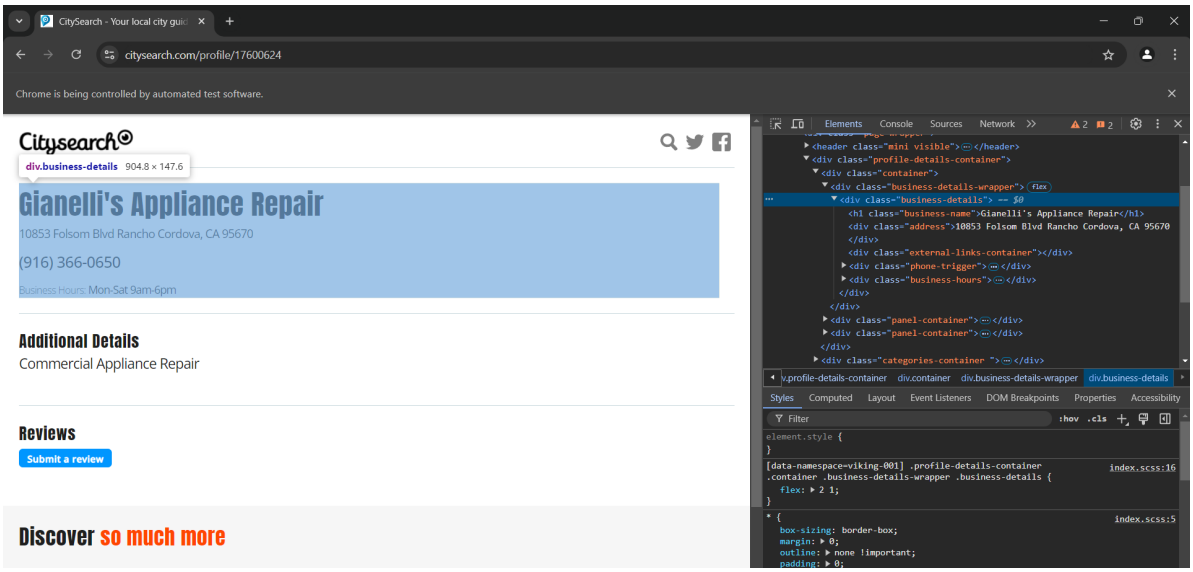




Scraping job description

In [ ]:

```
business_list = []

for i in range(0, 5): #5 should be job_cards_link's length when implem

    driver.get(job_cards_links[i])

    try:
        elem = WebDriverWait(driver, 10).until(EC.presence_of_element_
    except TimeoutException:
        print("Timed out waiting for page to load")

    # not sure if all business have all their contact info so creating
    business_details = driver.find_elements(By.CSS_SELECTOR, 'div.busi

    business_details_dict = {
        entry.get_attribute("class"): entry.text
        for entry in business_details
    }

    business_list.append(business_details_dict)
    time.sleep(3)
```



Appliance Repair HVAC Restaurants

Converting to dataframe, renaming columns and exporting to csv

In [ ]:
```
df = pd.DataFrame.from_dict(business_list)

df.rename(columns={
    "business-name": "business name",
    "external-links-container": "external link",
    "phone-trigger": "phone number",
    "business-hours": "business hours"
}, inplace=True)

df.to_csv(f'./{state}_{city}.csv', index=False) # example al_birmingha
```

In [ ]:
```
driver.quit()
```

Possible Improvements and Changes

Depending on the company's needs, I can store these values elsewhere instead of a CSV. Possibly in MongoDB or an SQL database.

When scraping large amounts of data, the current script may run into memory issues. During full implementation, I'll refactor the script into something more modular or OOP. Selenium has something called Page Object Model (POM). I'm not too familiar with POM but I am more than willing to try!