**Building Rental Property Dataset Using Web Scraping**

**Team Members:**

Madhu Bandru – MB4236

Karthikreddy Kuna – KK3375

Varun Kumar B – VB475

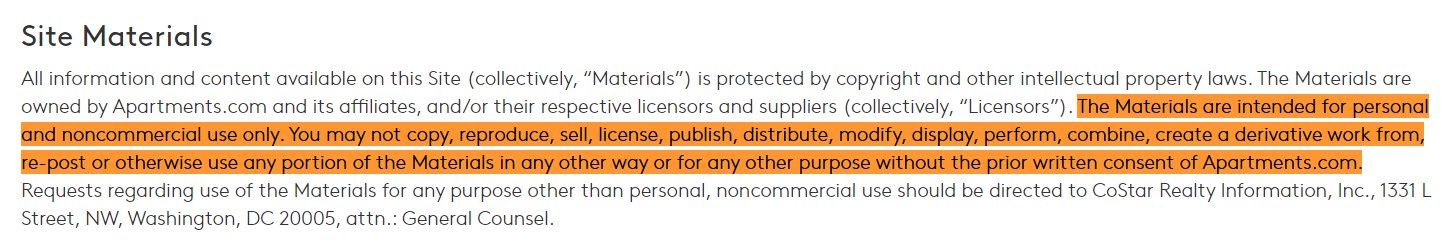
Pradeep Kumar Kankipati – PK593

The objective of the project is to gather the data available over rental site apartments.com using web scrapping technique and put that data into csv format so that data can be used to perform analysis and derive meaningful insights for prediction of pricing.

We have followed step by step procedure to achieve this objective.

**Step 1:**

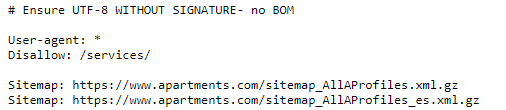
Before we scrap a website, we need to check the legal terms to confirm that web scraping can be performed on site or not. So, based on this we found that we can perform web scrapping on apartments.com site when we do it for personal or non-commercial purpose.



**Step 2:**

In addition to legal terms, we need to take a look at their robots.txt. This file tells us which services can be scraped within the website. To find the robots.txt we need to add “/robots.txt” to the base url. As, we want to crawl through apartments.com , we performed this action on <https://www.apartments.com/robots.txt>

For apartments.com, the robots.txt contains the following, as of today.



From this, we can crawl all the sections on the site except for the ones with apartments.com/services/ in the URL.

**Step 3:**

Getting the necessary libraries.

**Beautiful Soup:**

We have chosen Beautiful Soup which provides simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree using Python parsers like lxml and html5lib. It automatically converts incoming documents to Unicode and outgoing documents to UTF-8.

**Pandas:**

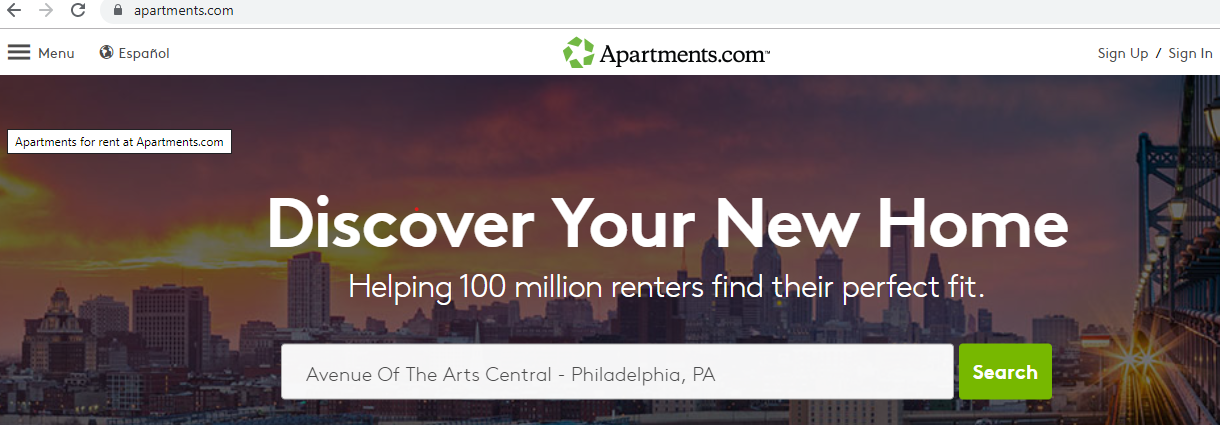
**pandas** is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labelled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

A screenshot of a cell phone

Description automatically generated

**Step 4:**

We inspected the website for different cities to figure out the structure of the URL.



Philadelphia:

<https://www.apartments.com/philadelphia-pa/>

Cincinnati:

<https://www.apartments.com/cincinnati-oh/>

City URL with Page numbers:

<https://www.apartments.com/philadelphia-pa/2/>

**URL Pattern:**

So it can be observed that for each city, URL pattern will be “[https://www.](https://www.apartments.com/philadelphia-pa/1/)apartments.com/cityname-state/page-number"

Note: For first page there is no page number in URL.

**Step 5:**

Extracting the list of major cities below.



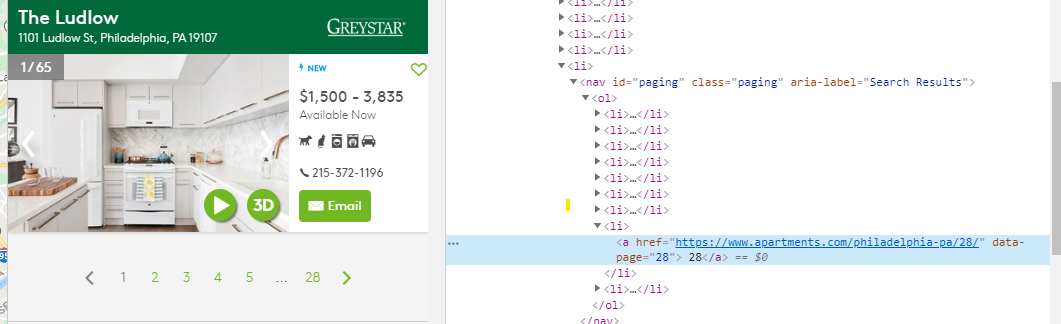


A screenshot of a cell phone

Description automatically generated

**Step 6:**

To crawl through all pages in the websites we need to figure out the last page number for each city. By inspecting the paging section.





While extracting data the last page number of the city, we observed that few cities has different patterns of showing last page number.

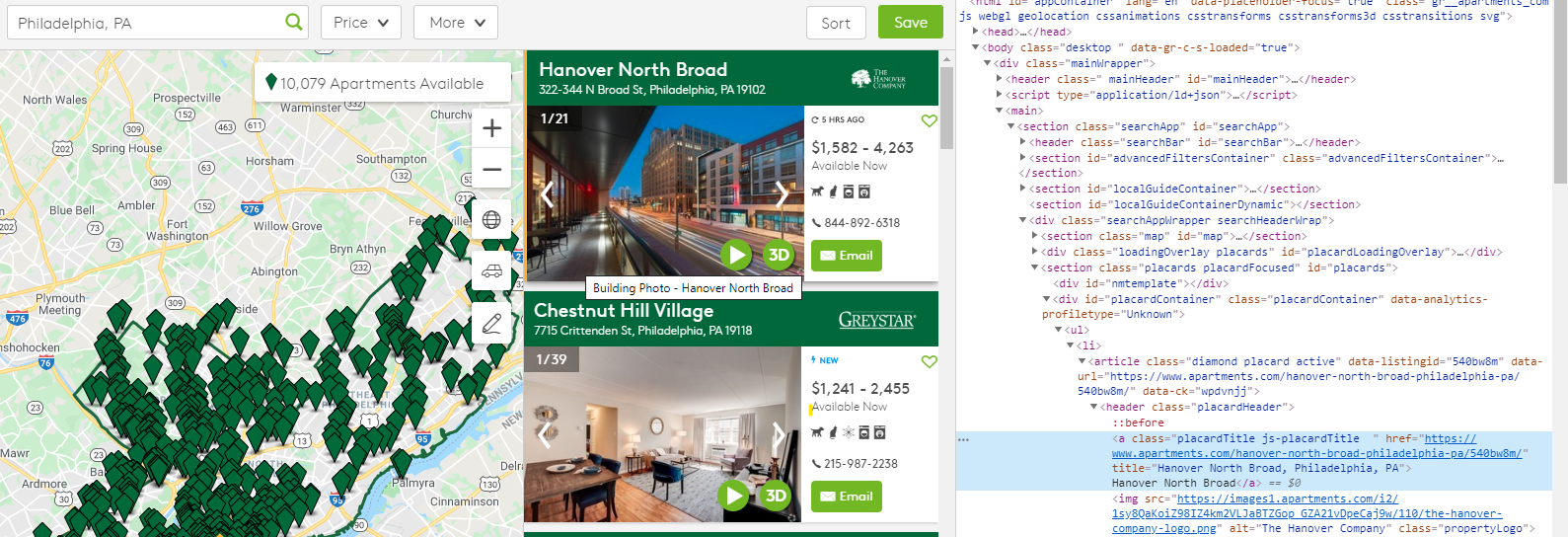
There wasn’t a common blueprint followed for the hovering of page numbers.



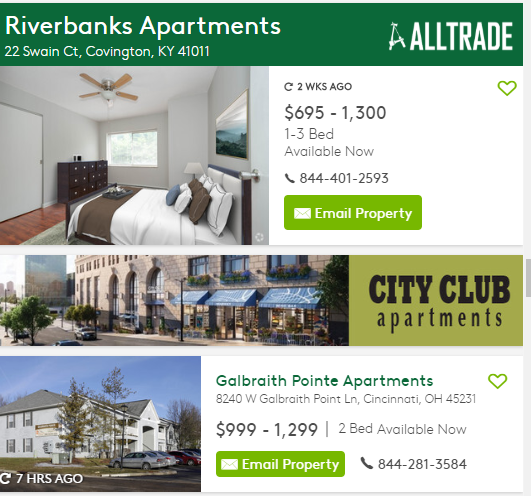


**Step 7:**

Extracting the list of Property URLs for each page using which we can able to get the full information of each property.



While extracting the URL of each property, we could see few properties have different banner style due to which the property URL is stored in different tags.



**Step 8:**

Getting the information of following fields for each property and storing the data in csv using pandas.

1. property name

2. property street

3. property city

4. property state

5. property zipcode

6. Studio

7. 1 Bedroom

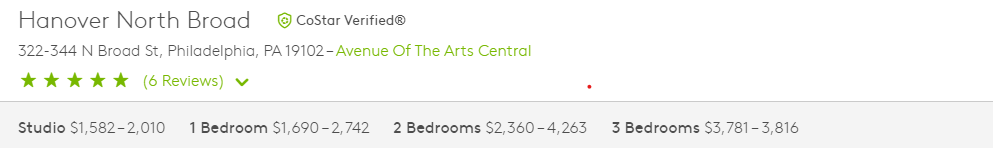
8. 2 Bedrooms

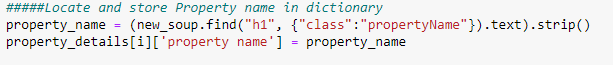
9. 3 Bedrooms

10. Latitude

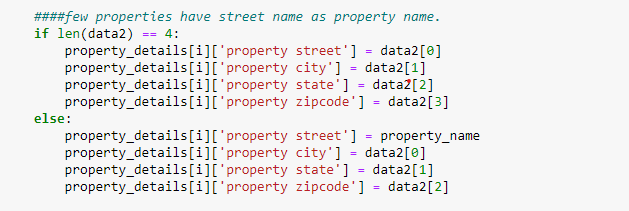
11. Longitude

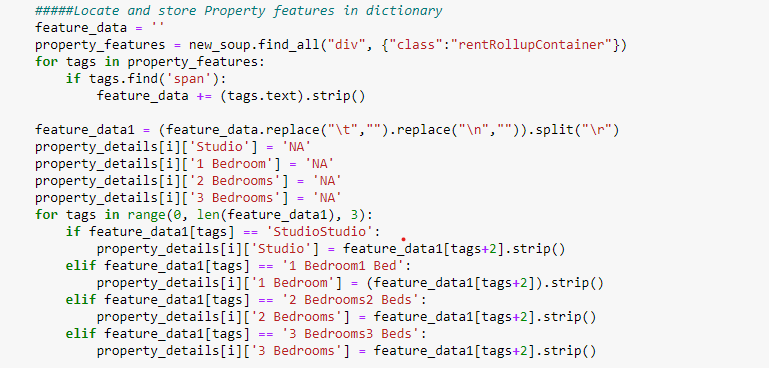
12. URL





Few properties do not have street names so we have used property name as street name for such cases.







Data is stored in CSV format using pandas framework.

A close up of a logo

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Readme.md link below:

https://github.com/Madhu407/DSCI-511-Final-Project.git