**Data Scraping and Research :**   
  
Initial research on Selenium and Beautifulsoup libraries :

| **Feature** | **Selenium** | **Beautiful Soup** |
| --- | --- | --- |
| **Type** | Browser automation tool | HTML/XML parser |
| **Use Case** | Scraping dynamic websites (JavaScript-heavy pages) | Scraping static websites (pre-rendered HTML) |
| **Rendering** | Uses a real browser (e.g., Chrome, Firefox) | Works with raw HTML, does not execute JavaScript |
| **Speed** | Slower due to browser automation | Faster as it only parses HTML |
| **Interaction** | Can interact with forms, buttons, scrolling, etc. | Only extracts data, cannot interact with elements |
| **Installation** | Requires a WebDriver (e.g., ChromeDriver) | Requires only BeautifulSoup library |
| **Best For** | Dynamic content, complex navigation | Quick parsing of well-structured HTML |

**Use Cases :**

1. Beautiful Soup: Ideal for extracting structured data from static web pages using simple parsing.
2. Selenium: Useful for scraping JavaScript-rendered content and interacting with websites dynamically.

For robust web scraping, both can be combined: Selenium fetches dynamic content, and Beautiful Soup extracts relevant data from the rendered page.

Went through tutorials of these to understand how to use them.

https://www.youtube.com/watch?v=XVv6mJpFOb0

Performed web scraping on one of the source pages provided by the BU Law client, given that it is publicly available, presumed that it is within the ethical limits :   
  
**Source :**

Massachusetts Laws, since our project is related to evictions I took the **Chapter 239: SUMMARY PROCESS FOR POSSESSION OF LAND** to get data from

<https://malegislature.gov/Laws/GeneralLaws/PartIII/TitleIII/Chapter239>

### **Approach :**

1. Extract all section links from the main Chapter 239 page.
2. Visit each section link, extract the legal text.
3. Within each section, find additional links (if any).
4. Visit those additional links recursively to ensure complete data collection.
5. Save all the scraped content into a structured text file.

**\*Code and Output have been attached in the same folder\***

### **Features of the code :**

Recursively follows links within each section to get full legal text.  
 Avoids duplicate scraping using a set of visited links.  
 Waits properly for JavaScript-rendered content before extracting.  
 Saves structured data into a file (chapter\_239\_evictions\_full.txt).