Instructions

All terminal commands are highlighted.
Make sure you first have python 3 installed.
You can check this by running "python -V" in the terminal.
If the version it writes is not 3, download python 3 and for the instructions, use python3 everywhere instead of python.

Steps for the very first time

1. unzip the downloaded folder
2. Open the terminal
3. Type in cd and space
4. open the terminal and drag (click, hold down and move mouse) the unzipped folder onto the terminal
5. It should have pasted the folders path after cd onto the terminal. Press enter
6. type python -m pip install -r requirements.txt and press enter

Every other time you would like to run the script, you need to redo steps 2-4 from the first time, so your terminal is running in the folder of the script.

Now that your terminal is in the folder of the code, you can run the script. You can type python frontendy into the terminal and press enter to start the GUI.

This will launch the GUI. You can navigate the pages of filters through the arrow buttons in the top right corner, or by pressing on the page name. Once you have entered all the filters you would like, you can press on the start button to take you to the start page. The start button is displayed below. Then press on the button to apply the filters and the script will start. Information about it's status will be displayed in the terminal. The filter validation on the GUI is not very lackluster. If a filter is in the wrong format, it will write an error message in the console and the GUI will close. Make sure the inputs are in the correct format. The format for all the dates is MM/DD/YYYY. In Canada you may use the format of DD/MM/YYYY but the GUI uses the format I mentioned. For the city, enter in only the city name, such as 'Burnaby', and press submit. And for the price simply enter in numbers.

Output while running

While the script is running, it will output certain information about what it is doing. For every location, it will output “starting on location” where location is the starting link of that location.

After that, the script will output “listings found: number” where number is the amount of listings found for that city.
If the option to continue the file from the options page is used, there is a slight difference in that for every city it will instead output “new listings (non duplicates) found: number” where number is the number of new listings that were found. The information is actually the same for the flag or not: the number of listings found. But with the flag it is more useful as you can know which cities have had new posting, since many may not have if the script has been run recently again. If it outputs 0 as the number, that means all the listings were already contained in the output file and nothing new has been posted. The way listings are differentiated is through their unique ID on the website.

Then it will continually output a number that keeps increasing. Each number is one listing being written to the file.

Every so often, it will continually output “sleeping” instead of a number. This means the script has temporarily stopped making requests to the website's server because they block too many requests. Do not worry, it will continue after 20-30 seconds once the server allows it once more.

Finally it will output “FINISHED”. It is done running and you can now open the output file to view the results. You cannot have the file open in something such as excel while the script is running or it will error as it will not be able to write to it.

If it ever displays something cryptic such as

```
Traceback (most recent call last):
  File "C:\Users\\main.py", line 99, in <module>
    main()
  File "C:\Users\main.py", line 35, in main
    with open(outFile, 'r+' if continue_file else 'w', newline = '', encoding = 'utf-8') as
```

csvfile:
PermissionError: [Errno 13] Permission denied: 'output.csv'
and stops running, that means an error has occurred. It is unlikely for any unaccounted errors to occur, since I addressed any that had the possibility of occurring during my testing, but something unexpected can always happen. To address this, copy paste the entire error message, or take a screenshot, and contact me. I will fix it and get back to you.

The output does not need to be monitored, it is just auxiliary information while it is running. Since there are many total listings for all the cities involved, and each needs to be meticulously scraped individually, it takes some time to finish running. It takes a second or two per listing, and in my testing I found there to be around 1000 for all the cities, although this number will always vary. So that would take about 30 minutes. I recommend simply leaving the script running and coming back when it's done.
If you run into any issues, or have any additional questions, feel free to reach out to me again.