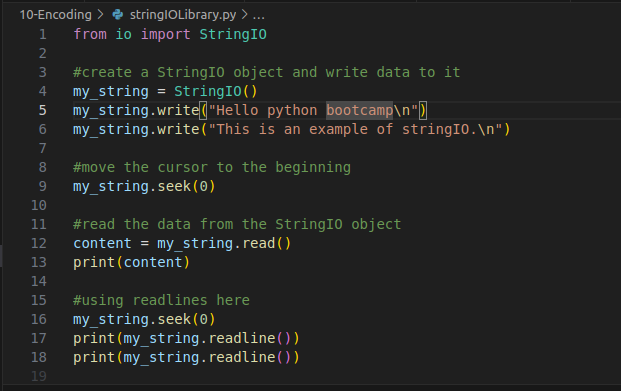
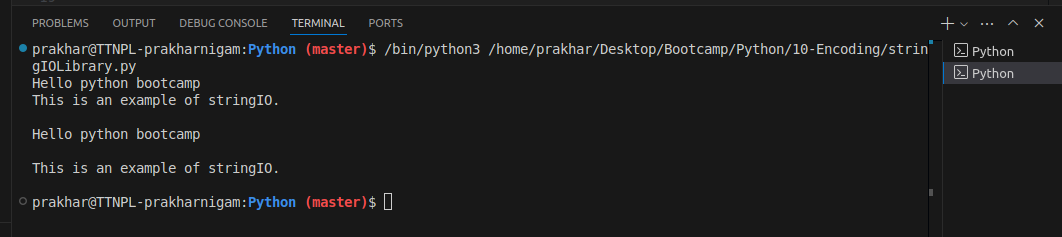
Assignment (Data Encoding & Processing)

1. **What is StringIO library for? explain with an example.**

The StringIO library in Python is used to treat strings as file-like objects. It is mainly used for using in memory file operations by using strings instead of using actual files. This means we can perform file operations such as reading and writing on an in-memory string, same function as doing on files, but without actually using it. It's particularly useful when you need to simulate file handling but don't want to work with actual files on the disk. It is useful for testing or when manipulating data temporarily in memory. Lets take a look at the following example:



In this example I used StringIO to convert my\_string into a stringIO object. After that I performed operations just like I did on the files. Following is the output

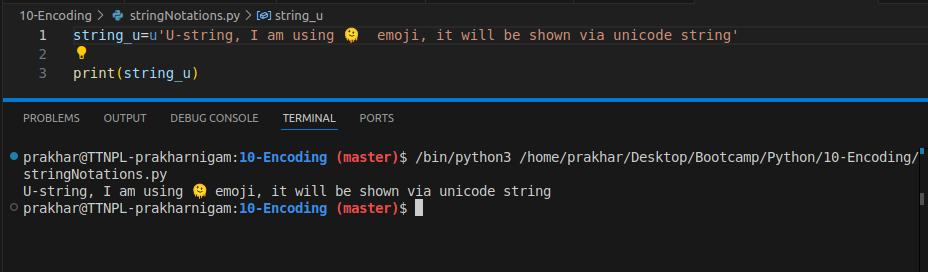


Output

1. **What is u"", r"" and "", f"" string notations in Python. Give examples?**

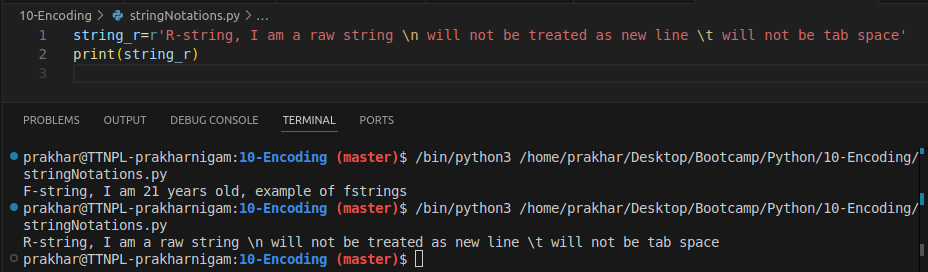
### Unicode String (u””)

The u"" notation was used in Python 2 to define strings that support Unicode (which is just a fancy way of saying that it can handle characters from almost any language or special symbols). In Python 3: All strings are Unicode by default, so you generally don’t need u"" anymore. In this case, the u"" helps Python understand that this string can hold emojis or characters from other languages.



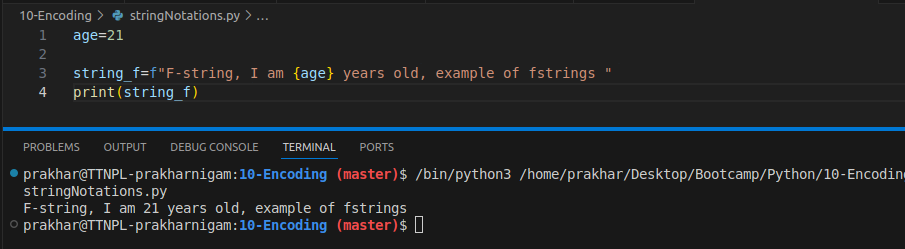
### Raw String (r””)

The r"" notation is used when you want to define a raw string. This means that Python will ignore escape characters like \n (newlines), \t (tabs), or \\ (backslashes). This is super helpful when you’re dealing with regular expressions or file paths (because backslashes are common in those). Raw strings are used majorly for defining directories in the strings.



### Formatted String (f””)

The f"" notation, also called f-strings, lets us put variables or expressions directly inside the string. This is really convenient when you want to combine text with numbers or other variables, and it's super readable.



1. **Part-1: Write a simple script to translate CSV file into JSON file input:**

**example CSV file output: .json file - each line is valid JSON equivalent to a row in CSV file**

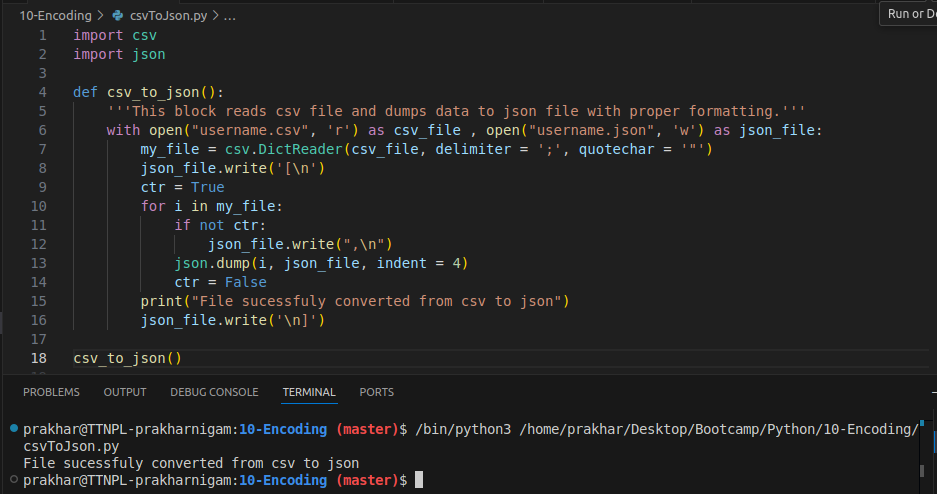
**Part-2: Write a script to translate a simple JSON file (white each line is a valid JSON) into .CSV file Assume that JSONs are not nested**

**Part-3: Write a simple function using StringIO which can translate a "dict" into a comma-separated string**

**Download CSV Example:** [**https://support.staffbase.com/hc/en-us/articles/360007108391-CSV-File-Examples**](https://support.staffbase.com/hc/en-us/articles/360007108391-CSV-File-Examples)

Part - 1

I used the username.csv file available on the internet. I used csv.DictReader to convert the data from the csv files into dictionaries. Since we need to convert this into a json file, DictReader serves the purpose as JSON files have key-value pairs and dictionaries are a perfect way to convert it. Then for format purposes, I used ‘.write’ to make the json file a list.

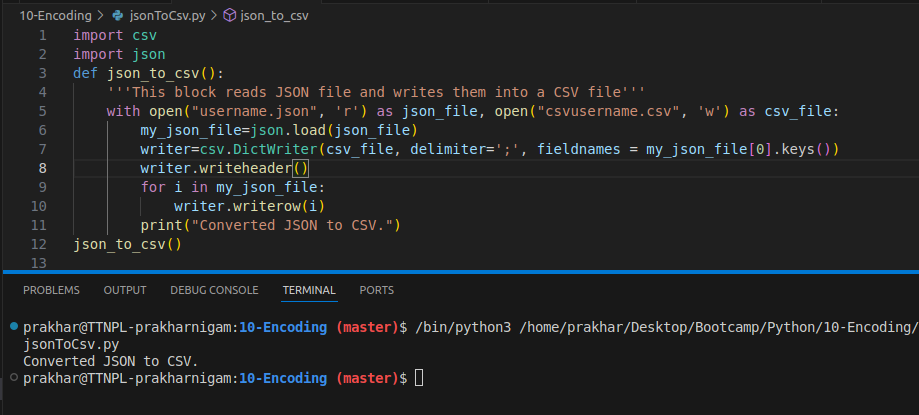


Code

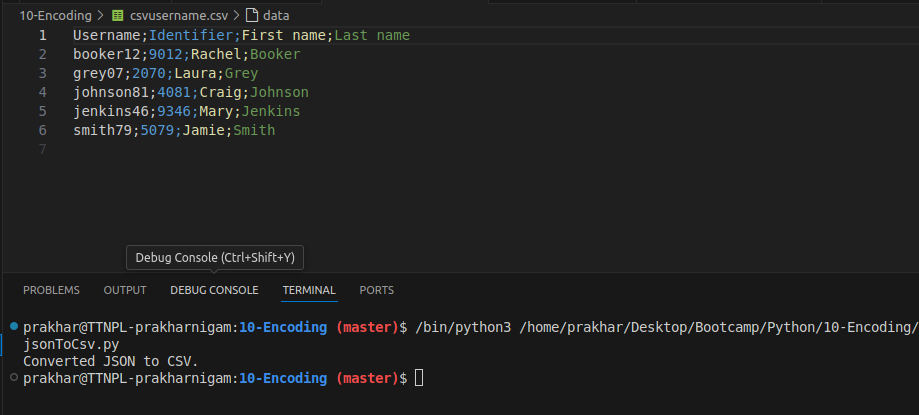


Output - “username.json” file

Part - 2

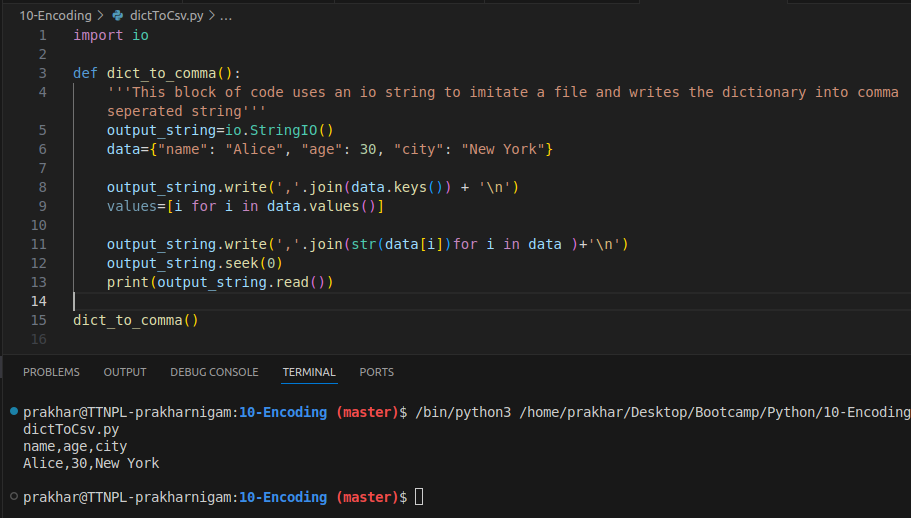


Code



Output - Csvusername.csv

Part-3

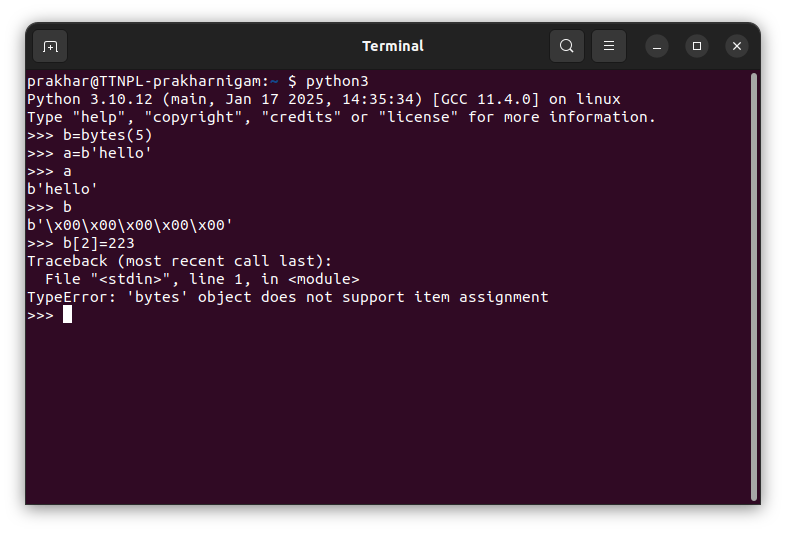


Code & Output

1. **In Python, what is `bytes` and `bytearray` ? What is the use of both and when to use both explain with a use-case code example.**

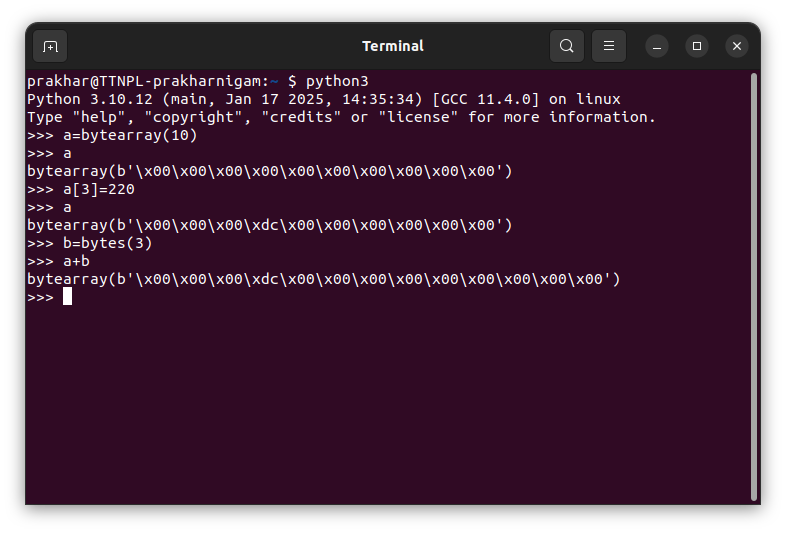
In Python, both bytes and bytearray are types used to represent sequences of byte data.

The bytes type represents an immutable sequence of bytes. Once a bytes object is created, its contents cannot be modified. It is typically used when the data needs to be treated as raw byte data, and you do not need to change the contents after creation. Since bytes are immutable, they are usually more memory-efficient for fixed data

**

*Used bytes and b’string’ to create bytes, upon changing the byte, we see an error*

The bytearray type represents a mutable sequence of bytes. You can modify its content after creation (e.g., append, change elements). It is used when you need a mutable sequence of bytes that can be modified during program execution. You can create a bytearray object using the bytearray() constructor. Since they are mutable, they are appropriate for modifying like invalid header, however they take more memory.

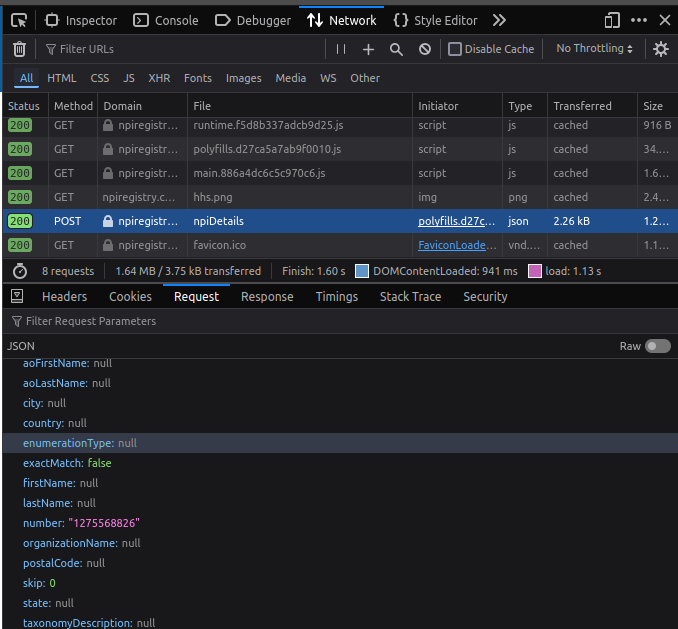
**

*Created a bytearray and changed the value and also adding byte and bytearray will result in bytearray*

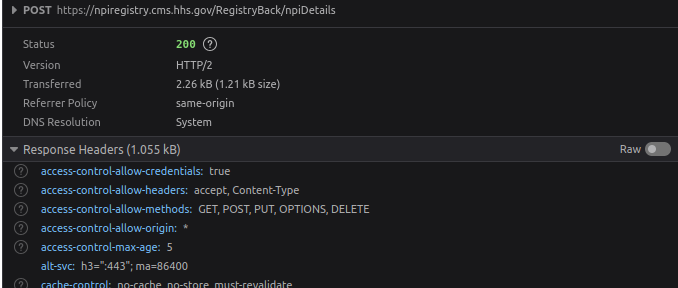
Assignment (Regular Expressions)

1. **Learn about “National Provider Identifier (NPI)” https://en.wikipedia.org/wiki/National\_Provider\_Identifier    On the portal https://npiregistry.cms.hhs.gov/search, one can search for details associated with an NPI. For example, use 1114473527 for the ‘NPI Number’ input field and click on the “Search” button The task in this assignment is to create a Python API to fetch details corresponding to an NPI as JSON. On the search result page, details are displayed as a table with a column name, and its value should form a JSON key-value pair. If a column has a subfield field, for example, Taxonomy, then create a nested JSON object On the result page, there is 2 address fields “Mailing Address” and “Primary Practice Address”, parse the address field into JSON as Stree-1, Stree-2, state, city, pin, phone, fax, zip as nested field.**

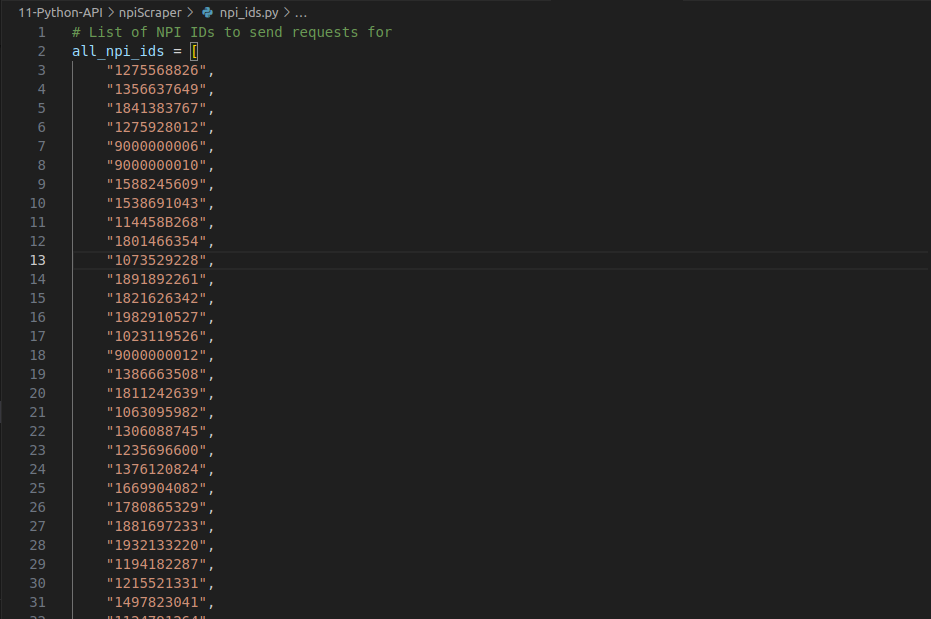
Example list of NPI numbers: "1275568826" "1356637649" "1841383767" "1275928012" "9000000006" "9000000010" "1588245609" "1538691043" 114458B268 "1801466354" "1073529228" "1891892261" "1821626342" "1982910527" "1023119526" 9000000012 "1386663508" "1811242639" "1063095982" "1306088745" "1235696600" "1376120824" "1669904082" "1780865329" "1881697233" "1932133220" "1194182287" "1215521331" "1497823041" 1124791264 "1578550679" "1174755672" "9000000013" "1790366169" "1952332165" "9000000008" null null "1750725362" null "1174944508" "1033421938" "1235149022" "1487662748" null "9000000002" "1477513158" null "1548350648" "1801320627" "1740748789" "1093858516" "9000000018" "1033537410" "1285675173" "1659633097" "1730161290" "1609974211" "1134518574" "9000000010" "9000000006" null null "9000000008" "1255316527" "1700062866" "1295753515" "1447756937" "1235452590" "1003472366" "1255733895" null "1912000613" "1033870274" "1417404039" "1295723005" "1013159763" "1538729314" "1003349648" "1073515011" "1073941159" "12345678" "1679532154" "1508380858" "1386793552" null "1407480791" "1588625511" null "1033525571" "1275594277" "1780601773" "1073579835" "1760561377" "1467946681" "1487072014" "1578237277" "1891754891" "1194457622" "1689818197" null "1316949563" "1558610576" "9000000007" "1780072868" "9000000001" "1154464410" "1699749374" "1740733880" "1326799016" "3746538476" 12557289994 "1689629875" "1114144417" "1073277695" "1659391050" "1689177271" "1609893072" "1932598729" "1255633525" "1144589268" "9000000081" "1487127734" "9000000010" "1013273671" null "1245793058" "1285602698" "1497945968" "1467938985" "1417330796" "1619401932" "1407341977" "1609864446" "1821251968" "9000000004" "1811776719" "1811555840" "1396302964" "1447623038" "1043335300" "1114402443" "1922629104" "1073985834" "1801856125" "1528027356" "1437491685" "9000000004" "9000000005" "1629325857" null "1245987833" "1265877732" "1356482905" "1174554000" "1467421602" "1073978979" "1740834993" null "9191920155" "1538150271" "1326275421" "1538691746" "9191919192" "1548314263" "1255728994" "1013558287" "1255945861" "9000000019" "1326406695"



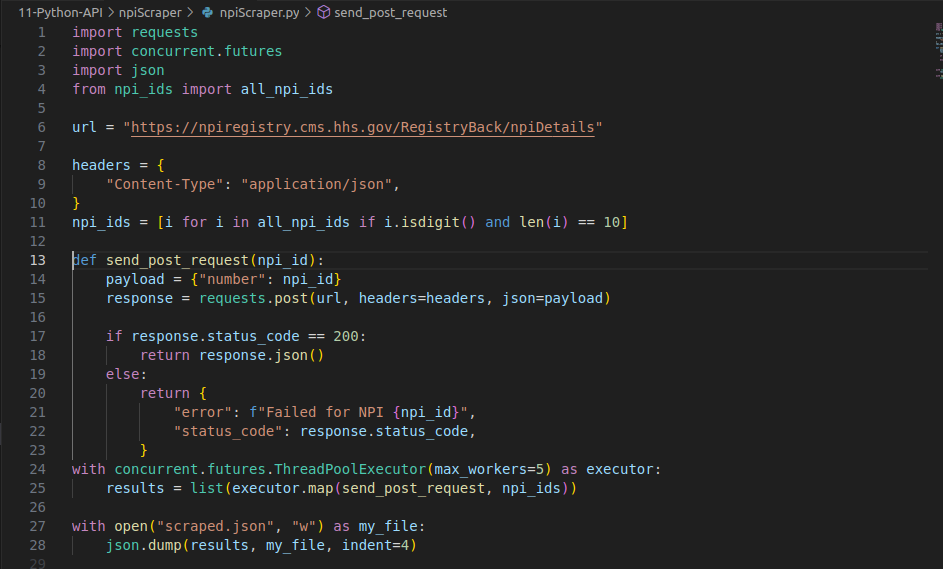
Checked the request on the website



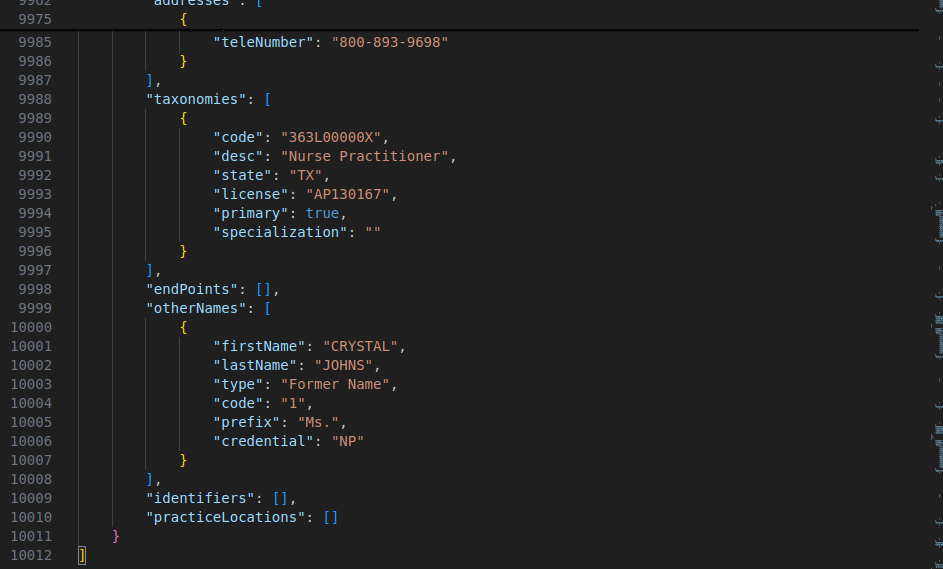
Got the post request url along with the header



Given IDs

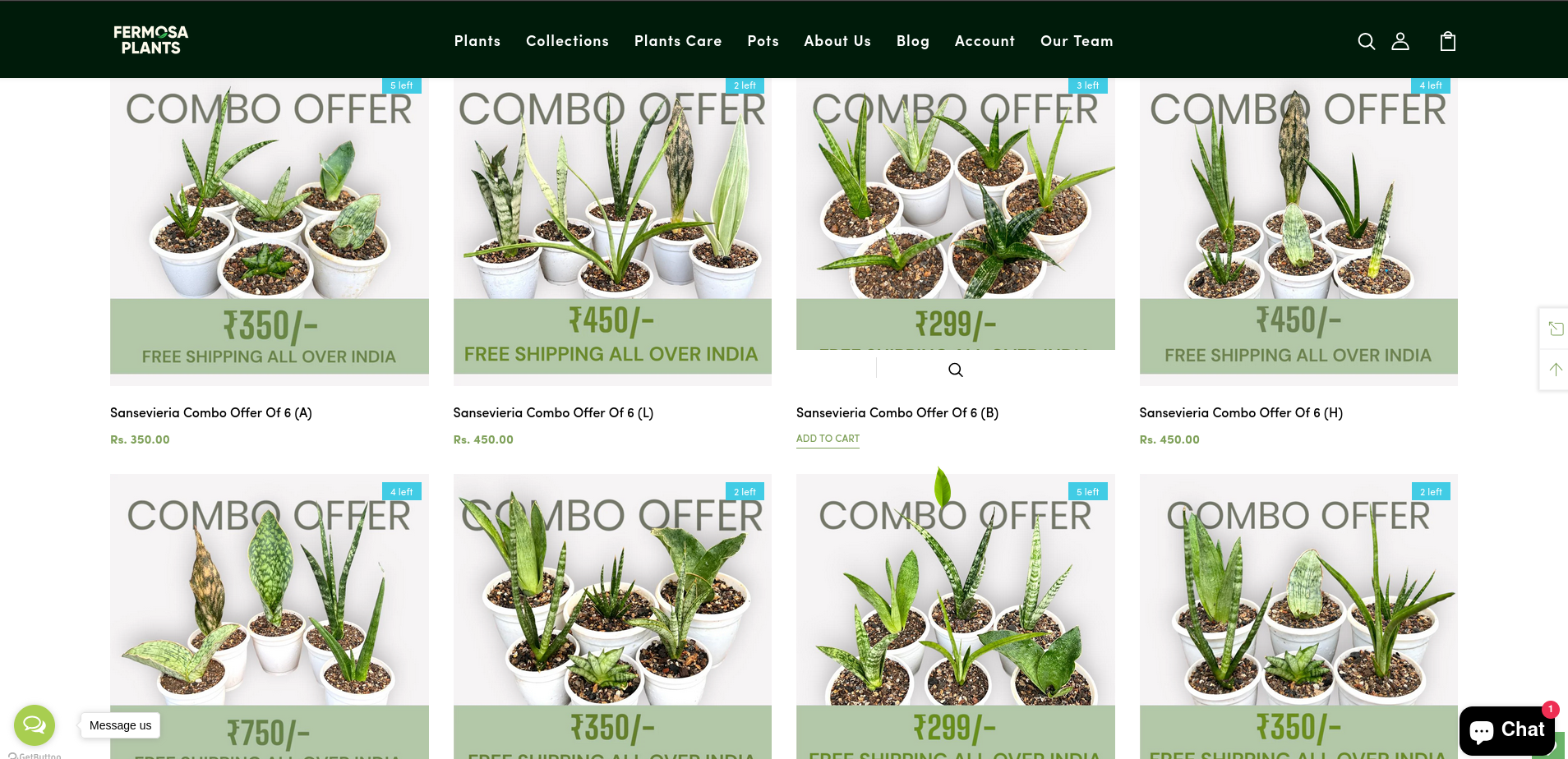


Code 1/1

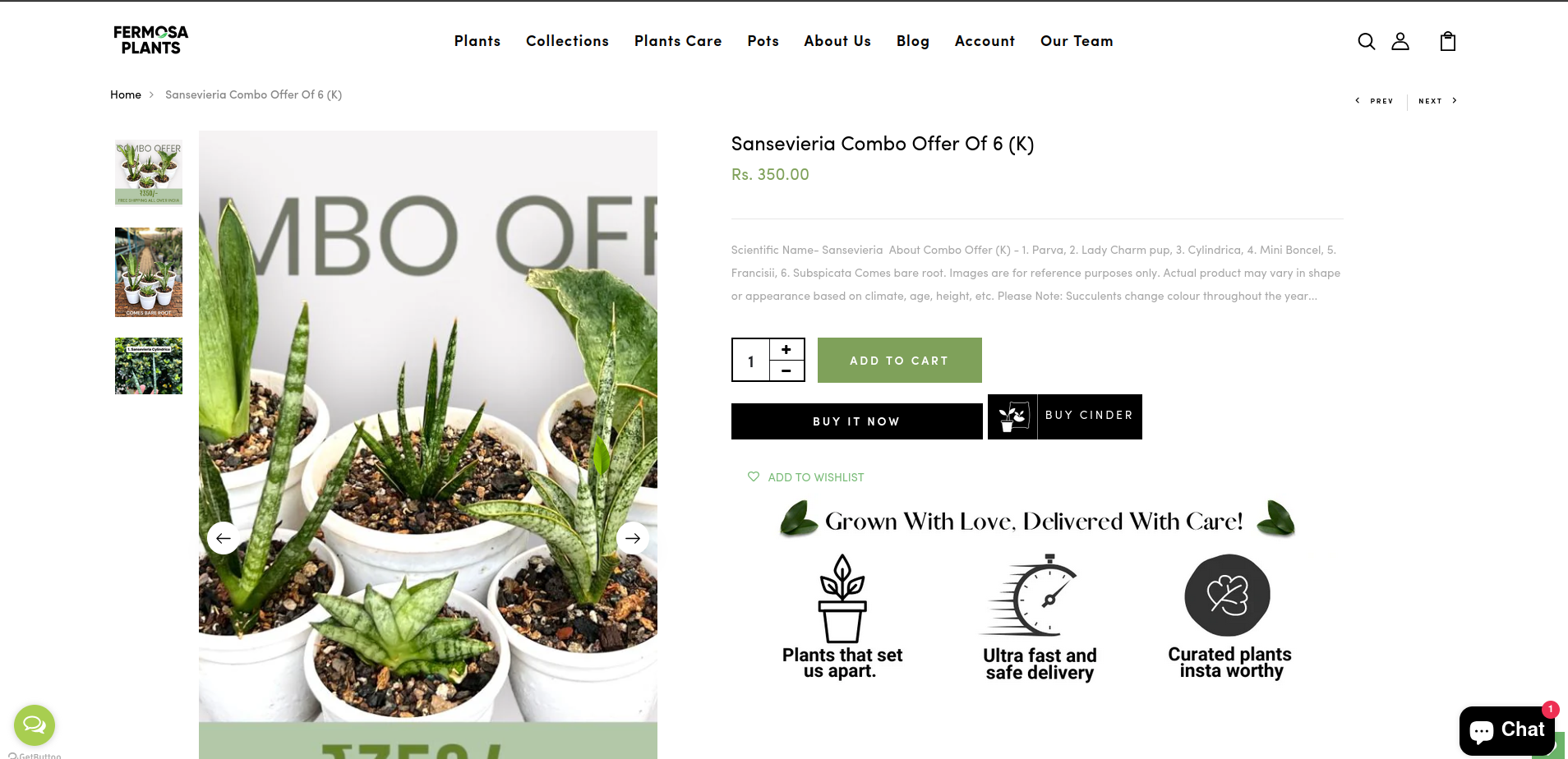


Data was scraped and dumped into a json file with over 10012 lines from the post requests.

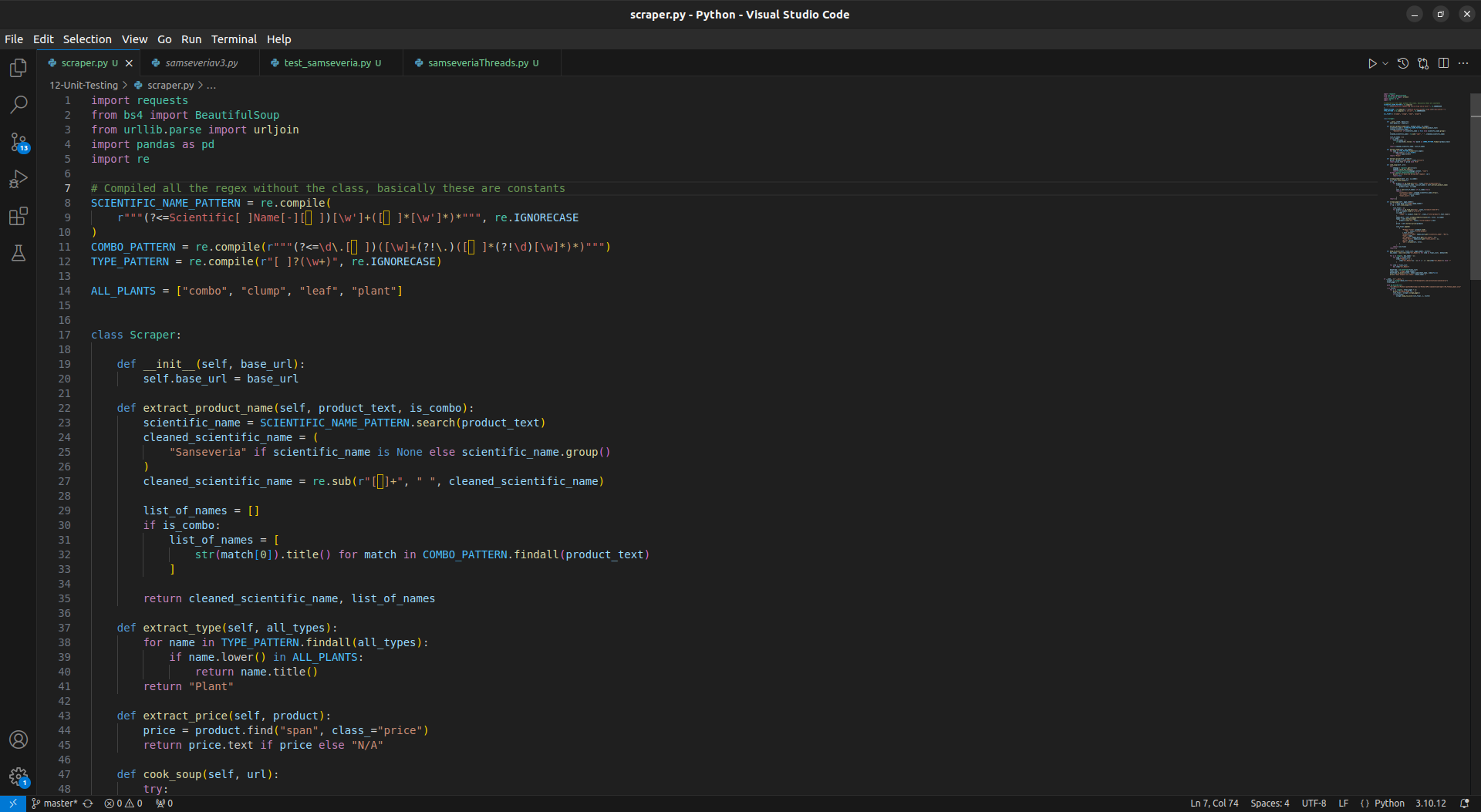
1. **Fermoa Plant is an online portal from where one can order different types of sansevierias, either in combo or as individual: https://fermosaplants.com/collections/sansevieria  This assignment is about scrapping the portal to prepare an Excel (not CSV) dataset.  Go through all pages of items: “https://fermosaplants.com/collections/sansevieria?page=<number>”   Visit each item and download the following information in Excel.  \* 1 Item in 1 row \* Columns:  <URL> <TYPE> <Price> <Number> URL: link of page Type: combo, clump, Leaf, plant, pub Price: Price in rupees Number: number of plant in combo Name: Plant names (one plant name in one column)**

****

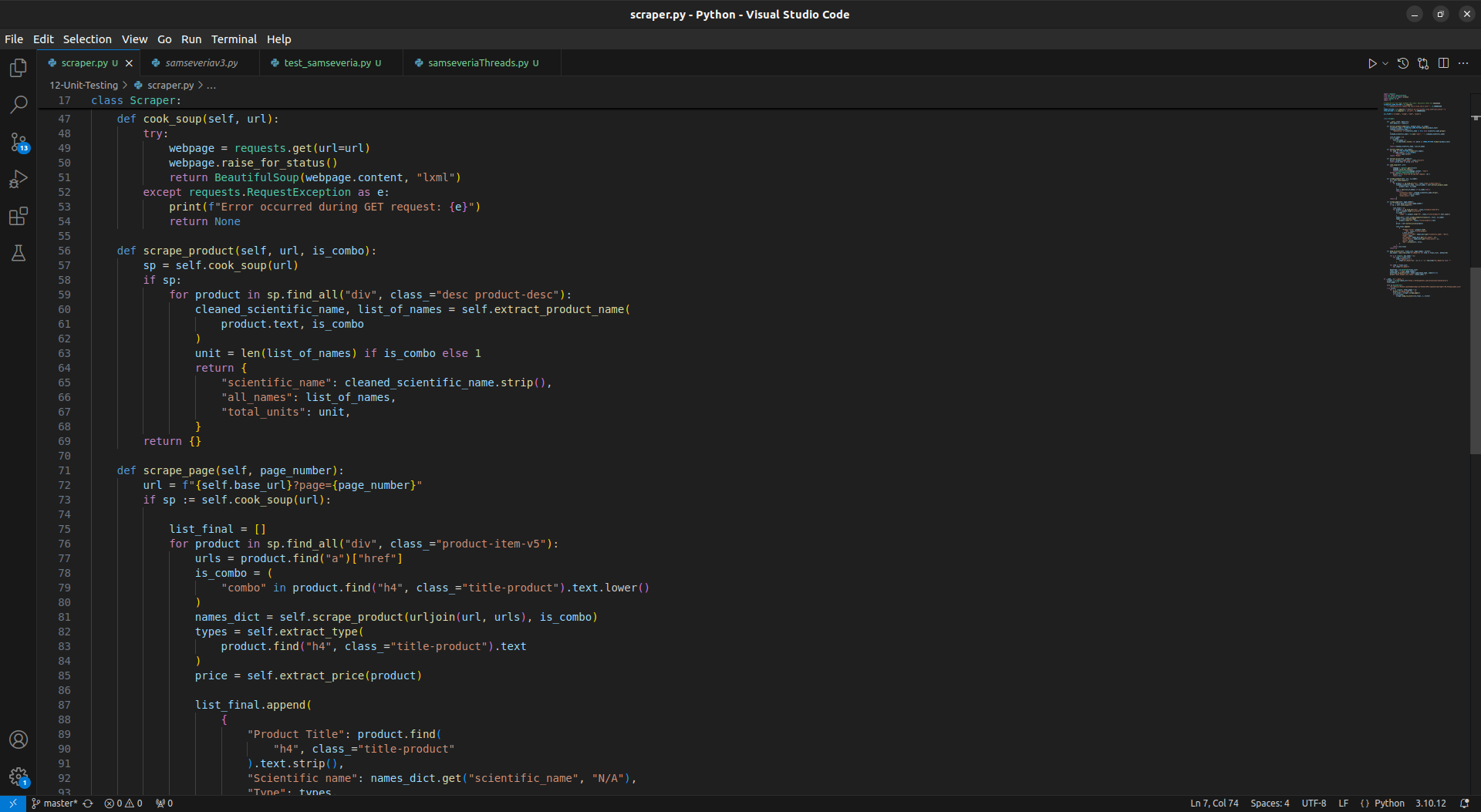
Page which will be scraped



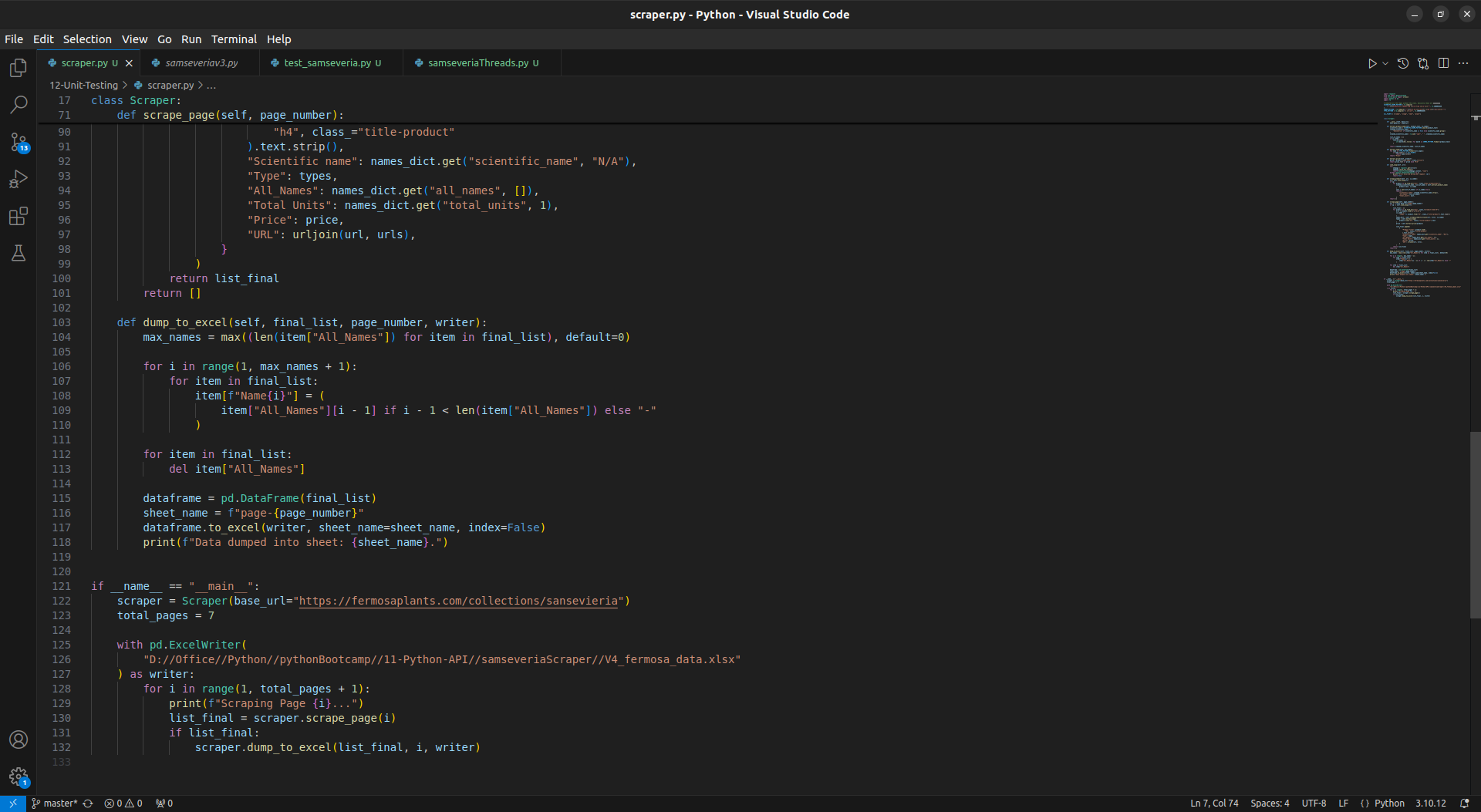
Each product which would be scraped while scraping page<number> from the website



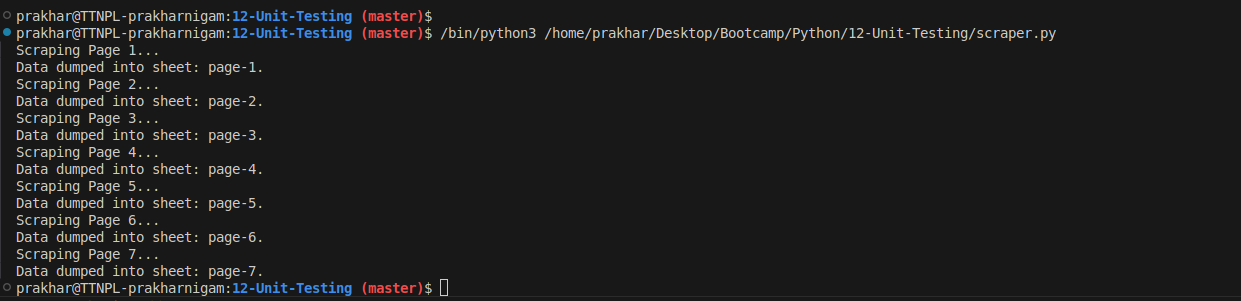
Code 1/3



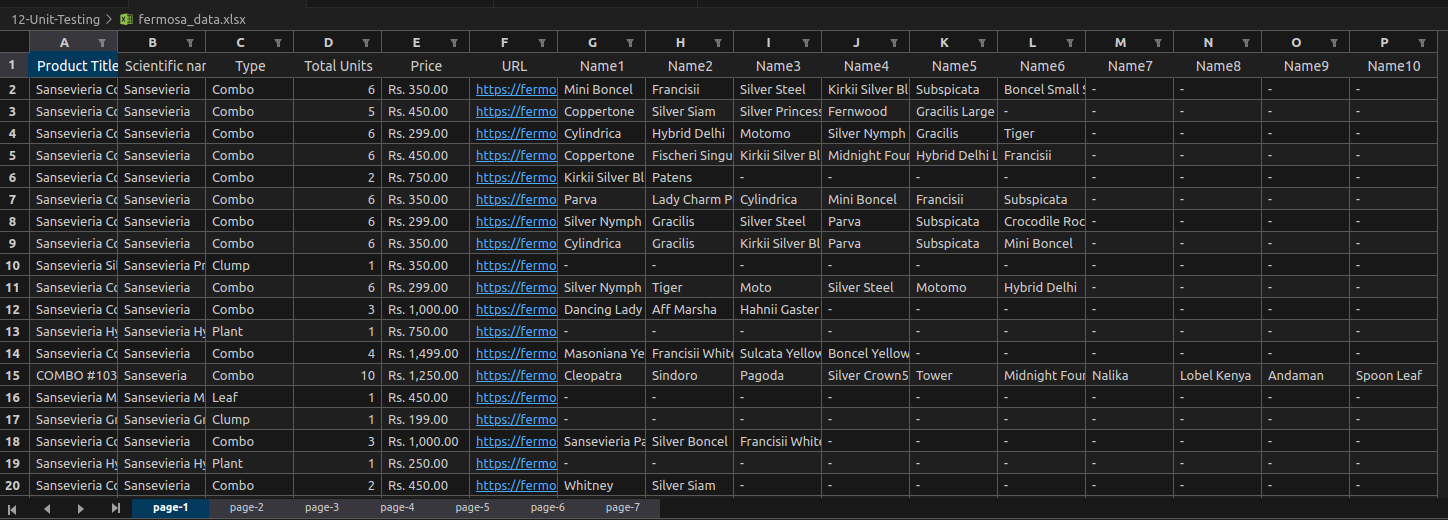
Code 2/3



Code 3/3



Output (terminal)



Excel file into which the scraped data was dumped.