SCRAPING THE PRODUCT DETAILS OF US FOOD & DRUG ADIMINISTRATION WEBSITE USING PYTHON

1. INTRODUCTION

The United States Food and Drug Administration (FDA or US FDA) is a federal agency of the Department of Health and Human Services. The FDA is responsible for protecting,promoting public health through the control and supervision of food safety, tobacco products, caffeine products, dietary supplements, prescription and over-the-counter pharmaceutical drugs (medications), vaccines, biopharmaceuticals, blood transfusions, medical devices, electromagnetic radiation emitting devices (ERED), cosmetics, animal foods & feed and veterinary products.

The objectives of this project is to scrap the details of a product given by user. User enters the name of the product or upload the list of the products and internal process gets started. Different details (i.e Approval Date, Different Patents, Strength & Expiration Date etc) of the product are scraped. After the process is completed the data is downloaded to the internal storage.

2. WHAT IS WEB SCRAPING ?

There are mainly two ways to extract data from a website

- Use the API of the website (if it exists).
- Access the HTML of the webpage and extract useful information/data from it. This technique is called **web scraping** or web harvesting or web data extraction.

Web scraping is the process of using bots to extract content and data from a website.Unlike screen scraping, which only copies pixels displayed onscreen, web scraping extracts underlying HTML code and, with it, data stored in a database. The scraper can then replicate entire website content elsewhere.

Steps involved in web scraping:

- Send an HTTP request to the URL of the webpage you want to access. The server responds to the request by returning the HTML content of the webpage.

For this task, we will use a third-party HTTP library for python-**requests**.

- Once we have accessed the HTML content, we are left with the task of parsing the data. Since most of the HTML data is nested, we cannot extract data simply through string processing. One needs a parser which can create a nested/tree structure of the HTML data. There are many HTML parser libraries available but the most advanced one is lxml or html.parser.
- Now, all we need to do is navigating and searching the parse tree that we created, i.e. tree traversal. For this task, we will be using third-party python library, **BeautifulSoup** It is a Python library for pulling data out of HTML and XML files.

3. WHY USE PYTHON FOR WEB SCRAPING ?

Python is an excellent choice for developers for building web scrapers because it includes native libraries designed exclusively for web scraping. Easy to Understand- Reading a Python code is similar to reading an English statement, making Python syntax simple to learn.
Requests, BeautifulSoup, Scrapy, and Selenium, are some popular libraries used for web scraping in Python.

4. METHODOLOGY

Python Libraries Used: [BeautifulSoup, Requests, Tkinter , Pandas etc. **]**
BeautifulSoup : *Beautiful Soup* provides simple methods for navigating, searching, and modifying a parse tree in HTML, XML files. It transforms a complex HTML document into a tree of Python objects. It also automatically converts the document to Unicode, so you don't have to think about encodings. This tool not only helps you scrape but also to clean the data. Beautiful Soup supports the HTML parser included in Python's standard library, but it also supports several third-party Python parsers like lxml or hml5lib.

Third-Party library integration: Beautiful Soup can be easily integrated with other Python libraries to improve the functionality of web scraping projects. For example, you

can use Requests or Selenium for making HTTP requests and then use Beautiful Soup to parse the web page's content.

Requests: Requests library is one of the integral part of Python for making HTTP requests to a specified URL. Whether it be REST APIs or Web Scraping, requests is must to be learned for proceeding further with these technologies. When one makes a request to a URI, it returns a response. Python requests provides inbuilt functionalities for managing both the request and response.

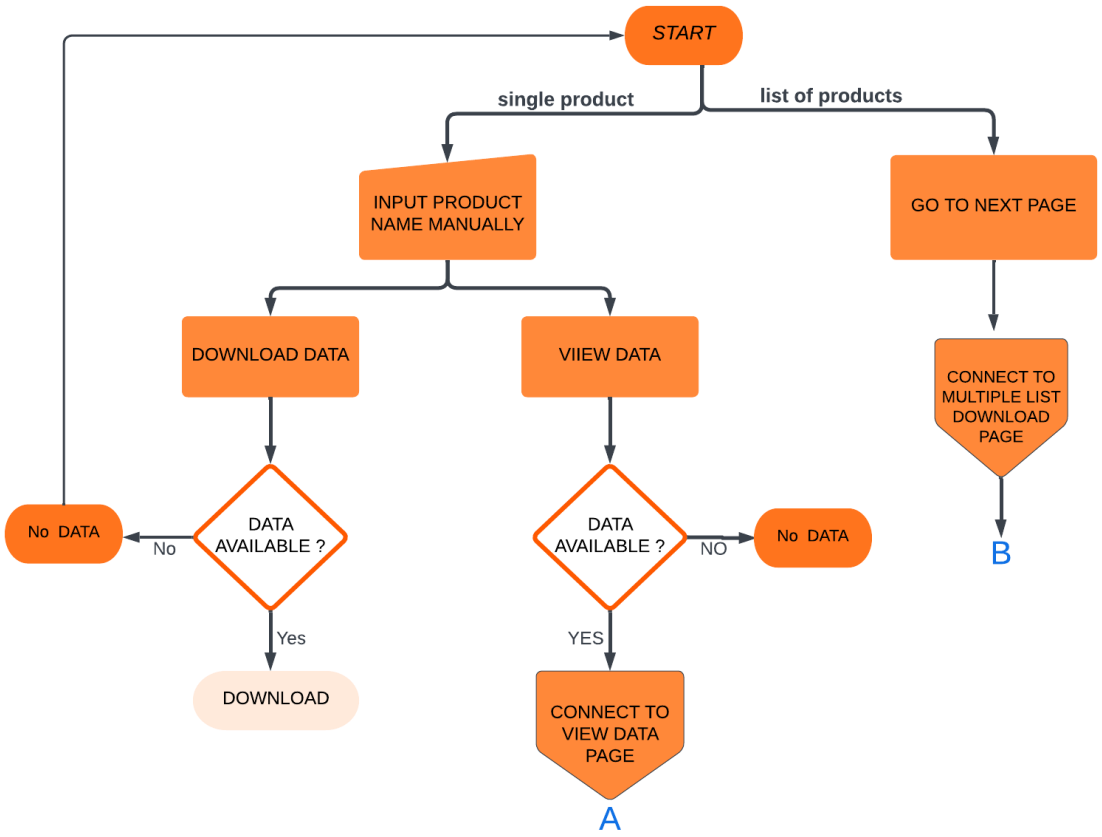
Requests library can be used to scrape the data from the website. Using requests, you can get, post, delete, update the data for the URL given. The handling of cookies and session is very easy. The security is also taken care of the help of authentication module support.

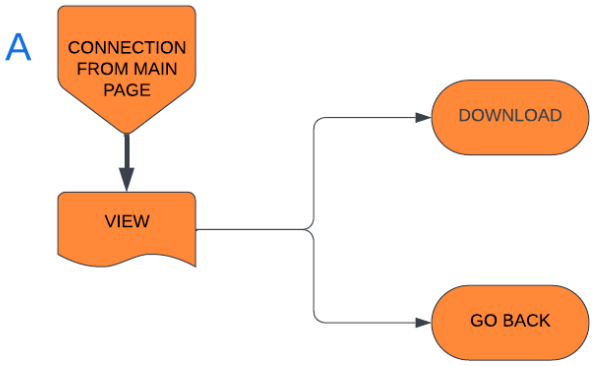
Tkinter: Python provides various options for developing graphical user interfaces (GUIs). **Tkinter** is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

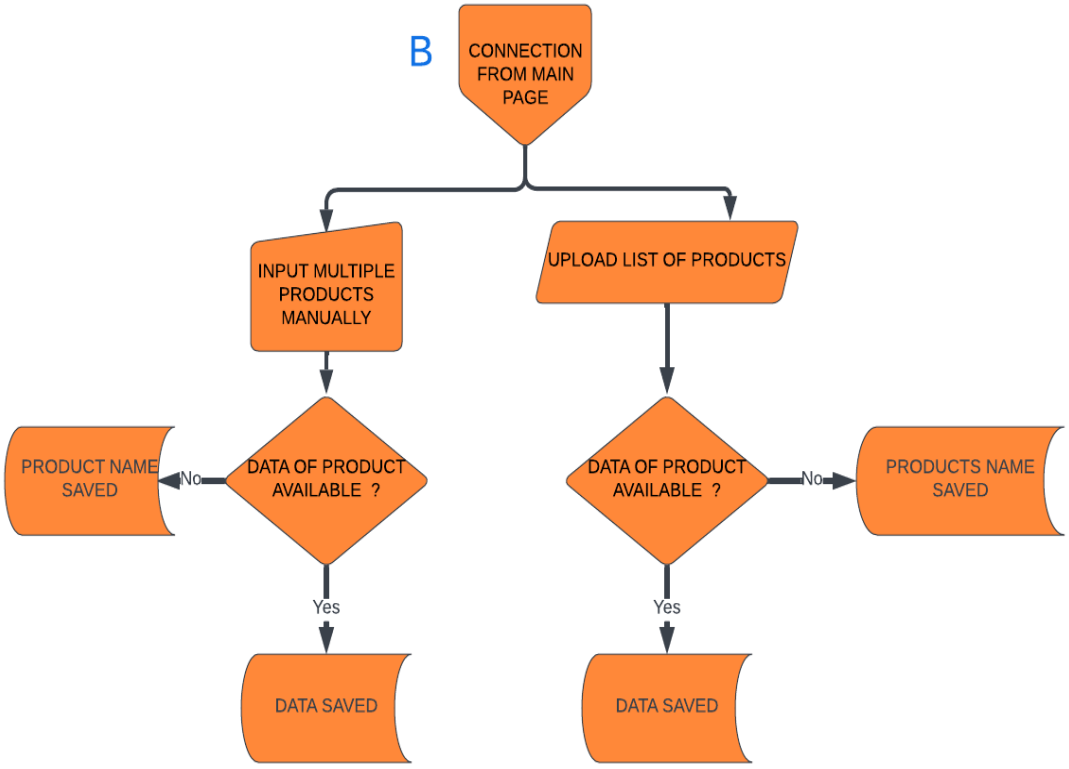
- Tkinter is easy and fast to implement as compared to any other GUI toolkit.
- Tkinter is more flexible and stable.
- Tkinter is included in Python, so nothing extra need to download.
- Tkinter provides a simple syntax.
- Tkinter is really easy to understand and master.
- Tkinter provides three geometry managers: place, pack, and grid. That is much more powerful and easy to use.

5. FLOWCHART

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. Here in this project there are several process so this created these flowcharts which shows the overall workflow of the project.


START
single product
list of products
INPUT PRODUCT NAME MANUALLY
GO TO NEXT PAGE
DOWNLOAD DATA
VIIEW DATA
CONNECT TO MULTIPLE LIST DOWNLOAD PAGE
No DATA
No
DATA AVAILABLE ?
DATA AVAILABLE ?
NO
No DATA
B
Yes
YES
DOWNLOAD
CONNECT TO VIEW DATA PAGE
A


A
CONNECTION FROM MAIN PAGE
VIEW
DOWNLOAD
GO BACK



6. HOW THINGS WORK

When the user starts working with the projects , first of all the main page pops up to the screen . Shown in the image(1).There are mutiple functionalities in the main page. Let us Suppose a user have a product name and he wants to download the details of that product. So the user will have to enter the the product name to the entry field. Now there are two buttons we see on the page , first one is **Download Data Button** and the second one is **View Data Button.**

image(1)-main page

- When the user clicks on the **Download Data Button.** There are some conditions checked first, like if the button is clicked with less than 3 character in the entry field or with a empty entry field, a warning message pops up to the screen which says that there should be more than 3 character of a product name. And if the entered product name is more than 3 characters than it is been checked that whether the data is available or not on the website. if the data is not available a message pops up to the screen saying that there is no data available, and if the data is available than a new window pops up for choosing the path for where to save the downloaded data, Shown in the image(2). After choosing the path for saving data downloading process gets started and it takes some time to process and ends with the message to the screen saying that the data is downloaded.

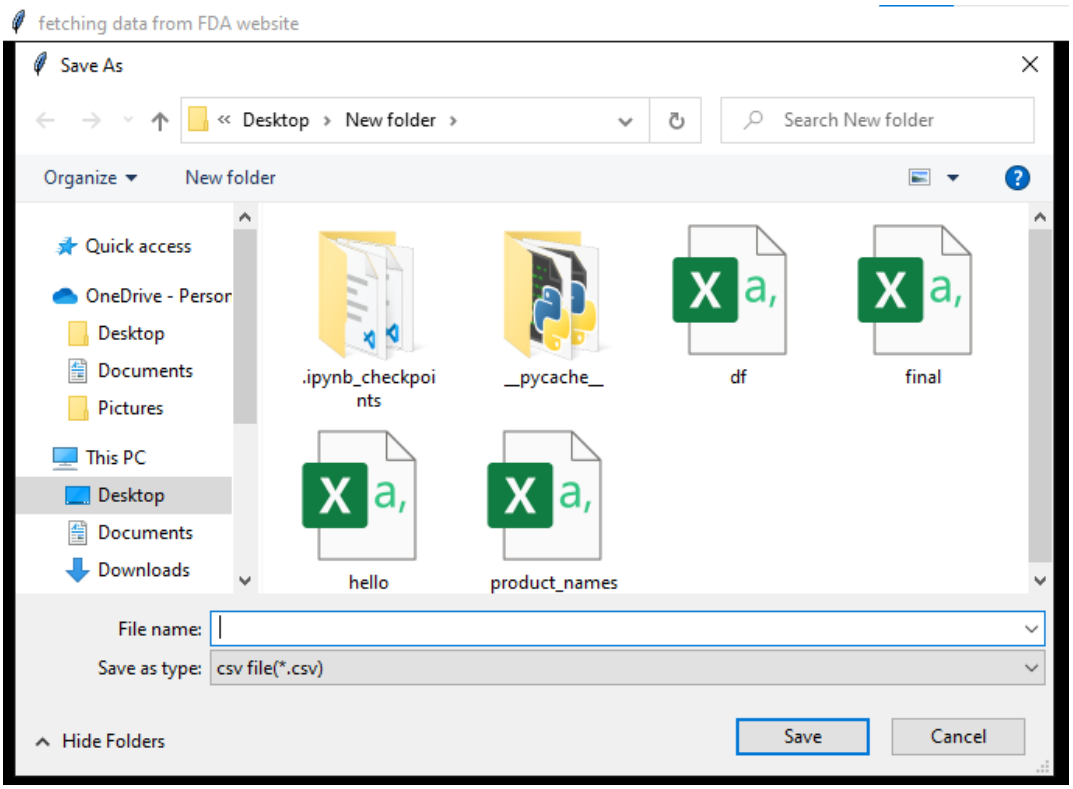
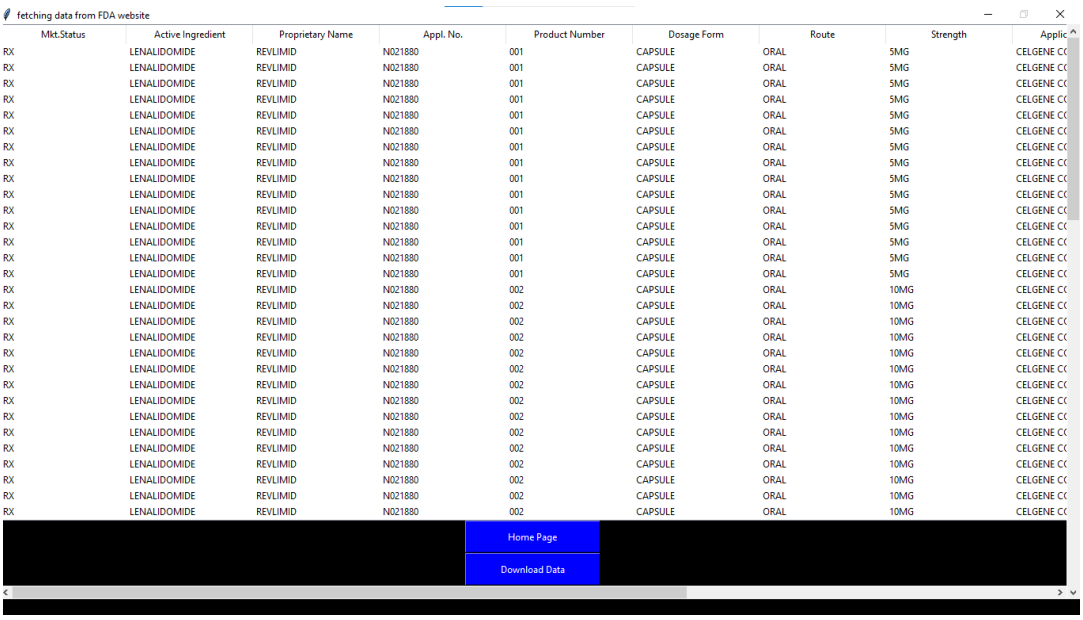


image (2)- [Choosing path for save data]



Image(3)- [Downloaded Data]

- On the other hand , when the user clicks on the **View Data Button.** The backend process gets started , different conditions for data is available or not are checked . If the data is available the data for the given product name is fetched from the website and a new page pops up with a table containig different details of the product like approval date , expiry date , strength of the product etc. Image(3) shows the data fetched for a product here user can also download the data which is in the table.
- There is another **Multiple Product Button** on the main page which redirect to the next page , where we can download the multiple product's data at a time. In Image(4) next page is shown.

image(4)--Multiple Products Page

Here in this page , there are two fields one for entering multiple products name manually and other one is for uploading the list of products from the internal memory.

- In the backend process when working with the multiple products, each product is being checked whether it has data available on the FDA website or not.
- If the data is not available on the website for a product , that product is saved to a list and returned to the log file
- If the data is available on the website , Each product data is downloaded and saved as a CSV file format. And the time for the entire process is calculated.

6. CONCLUSION

In conclusion, the web scraping project has proven to be a valuable and effective means of extracting relevant data of a given product from the US FDA official website. Throughout the process, we successfully gathered a vast amount of information, enabling us to gain valuable insights, make informed decisions, and derive meaningful patterns and trends.

Despite the success of this web scraping initiative, we must acknowledge that website's structures and content can change over time, affecting the scraping process. Regular maintenance and updates to the scraping scripts will be necessary to ensure the continued reliability and relevance of the data.