from selenium import webdriver

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.by import By

from webdriver\_manager.chrome import ChromeDriverManager

import pandas as pd

# Setup WebDriver

service = Service(ChromeDriverManager().install())

driver = webdriver.Chrome(service=service)

# Define the base URL and the class we are interested in

class\_name = "style\_\_font-bold\_\_\_1k9Dl"

# List to store the extracted values

extracted\_values = []

x = int(input("Enter start page no:"))

y = int(input("Enter end page no:"))

label=input("enter the label letter :")

# Loop through pages x to y

for page in range(x, y+1):

url = f"https://www.1mg.com/drugs-all-medicines?page={page}&label={label}"

driver.get(url)

elements = driver.find\_elements(By.CLASS\_NAME, class\_name)

for element in elements:

# Split the text by line breaks and check if "ADD" is present

lines = element.text.split("\n")

if "ADD" not in lines:

# Append medication name to extracted\_values

extracted\_values.append(lines[0])

# Close the WebDriver

driver.quit()

# Get the file path from the user

file\_path = input("Enter the file path to save the Excel sheet (e.g., 'C:/Users/username/Documents/medications.xlsx'): ")

# Convert the extracted values to a DataFrame

df = pd.DataFrame({"Medication": extracted\_values})

# Write the DataFrame to an Excel file

df.to\_excel(file\_path, index=False)

print("Excel file created successfully.")

# ----------------------------------------------------------------------------------------------------------------------------

# ----------------------------------------------------------------------------------------------------------------------------

import openpyxl

# Load the source Excel workbook

source\_workbook = openpyxl.load\_workbook(r'')

# Load the destination Excel workbook

destination\_workbook = openpyxl.load\_workbook(r'')

# Select the source worksheet

source\_sheet = source\_workbook.active

# Select the destination worksheet

destination\_sheet = destination\_workbook.active

# Initialize an empty list to store values from the first column

first\_column\_values = []

# Function to categorize medicine types

def categorize\_medicine\_type(value):

if value is not None: # Added check for None value

if "Capsule" in value:

return "Capsule"

elif "Tablet" in value:

return "Tablet"

elif "Syrup" in value:

return "Syrup"

elif "Suspension" in value:

return "Suspension"

elif "Drop" in value:

return "Drop"

elif "Injection" in value:

return "Injection"

elif "Ointment" in value:

return "Ointment"

elif "Cream" in value:

return "Cream"

elif "Powder" in value:

return "Powder"

elif "Soap" in value:

return "Soap"

elif "Lotion" in value:

return "Lotion"

elif "Suppository" in value:

return "Suppository"

elif "Gel" in value:

return "Gel"

return "Other" # Return "Other" for None values

# Function to process medicine names

def process\_medicine\_name(name, category):

if name.endswith(category):

return name[:-len(category)].strip()

else:

return name

# Iterate through the rows in the first column of the source sheet

for row\_index, row in enumerate(source\_sheet.iter\_rows(min\_row=1, max\_row=source\_sheet.max\_row, min\_col=1, max\_col=1), start=1):

original\_value = row[0].value

first\_column\_values.append(original\_value)

categorized\_value = categorize\_medicine\_type(original\_value)

processed\_value = process\_medicine\_name(original\_value, categorized\_value)

destination\_sheet.cell(row=row\_index, column=1, value=original\_value) # Write original value

destination\_sheet.cell(row=row\_index, column=2, value=categorized\_value) # Write categorized value

destination\_sheet.cell(row=row\_index, column=3, value=processed\_value) # Write processed value

# Save the changes to the destination workbook

destination\_workbook.save(r'C:\Users\mshar\OneDrive\Desktop\Automation projects\cbi.xlsx')