from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from selenium.common.exceptions import TimeoutException

import pandas as pd

import time

def setup\_driver():

# Configuration du navigateur Chrome

options = webdriver.ChromeOptions()

options.add\_argument('--headless') # Mode sans interface graphique

options.add\_argument('--lang=fr') # Définir la langue en français

return webdriver.Chrome(options=options)

def scrape\_parcoursup():

driver = setup\_driver()

try:

print("Accès au site Parcoursup...")

driver.get("https://dossierappel.parcoursup.fr/Candidat/carte")

wait = WebDriverWait(driver, 10)

# Attendre que le premier élément soit chargé

print("Attente du chargement des résultats...")

wait.until(EC.presence\_of\_element\_located((By.CLASS\_NAME, "fr-card\_\_content")))

# Récupérer tous les éléments

cards = driver.find\_elements(By.CLASS\_NAME, "fr-card\_\_content")

if not cards:

print("Aucune carte trouvée. Attente supplémentaire...")

time.sleep(5)

cards = driver.find\_elements(By.CLASS\_NAME, "fr-card\_\_content")

donnees\_etablissements = []

total = len(cards)

print(f"Nombre de cartes trouvées : {total}")

for index, card in enumerate(cards, 1):

try:

# Extraction des informations de chaque carte

info\_etablissement = {

'nom\_etablissement': card.find\_element(By.CLASS\_NAME, "psup-search-results-card\_\_school-name").text,

'nom\_formation': card.find\_element(By.CLASS\_NAME, "psup-search-results-card\_\_course-name").text,

'date\_extraction': time.strftime("%Y-%m-%d")

}

donnees\_etablissements.append(info\_etablissement)

print(f"Progression : {index}/{total} cartes traitées")

except Exception as e:

print(f"Erreur lors du traitement de la carte: {str(e)}")

continue

# Sauvegarde en Excel

print("Enregistrement des données en Excel...")

df = pd.DataFrame(donnees\_etablissements)

nom\_fichier = f'formations\_parcoursup\_{time.strftime("%Y%m%d")}.xlsx'

# Sauvegarde en Excel avec formatage

with pd.ExcelWriter(nom\_fichier, engine='openpyxl') as writer:

df.to\_excel(writer, sheet\_name='Formations', index=False)

# Ajustement automatique de la largeur des colonnes

worksheet = writer.sheets['Formations']

for idx, col in enumerate(df.columns):

max\_length = max(df[col].astype(str).apply(len).max(), len(col))

worksheet.column\_dimensions[chr(65 + idx)].width = max\_length + 2

print(f"Données sauvegardées dans {nom\_fichier}")

except Exception as e:

print(f"Une erreur est survenue: {str(e)}")

finally:

driver.quit()

if \_\_name\_\_ == "\_\_main\_\_":

print("Démarrage du script de scraping Parcoursup...")

scrape\_parcoursup()

source venv/bin/activate

pip install -r requirements.txt

python3 parcoursup\_scraper.py

# Create virtual environment

python3 -m venv venv

# Activate virtual environment (for zsh/bash)

source venv/bin/activate

# Install requirements

pip install -r requirements.txt