scraping Udemy courses and coupons that are free

==========

==========

i will do it in 2 steps

first step : scraping all courses names, links, coupons found after the links and save them in .csv file for easy visualization

main.py > ...

```python
import requests
from bs4 import BeautifulSoup
import re
import csv

# Step 1: Fetch the page content
url = 'https://yofreesamples.com/courses/free-discounted-udemy-courses-list/'
response = requests.get(url)
if response.status_code == 200:
    soup = BeautifulSoup(response.text, 'html.parser')

    # Open a CSV file to save the extracted data
    with open('udemy_courses.csv', mode='w', newline='', encoding='utf-8') as file:
        writer = csv.writer(file)

        # Write the header row
        writer.writerow(['Course Name', 'Course URL', 'Coupon Code'])

        # Step 2: Find all course containers with the specified class
        courses = soup.find_all('h4', class_='wp-block-heading')

        # Step 3: Extract name, URL, and coupon code
        for course in courses:
            # Extract course name
            course_name = course.get_text(strip=True)

            # Extract link
            link_tag = course.find('a', class_='external_link_title')
            if link_tag and 'href' in link_tag.attrs:
                course_url = link_tag['href']

                # Extract coupon code from the URL
                coupon_code_match = re.search(r'couponCode=([A-Z0-9]+)', course_url)
                coupon_code = coupon_code_match.group(1) if coupon_code_match else 'N/A'

                # Write the extracted information to the CSV file
                writer.writerow([course_name, course_url, coupon_code])

    print("Data saved to udemy_courses.csv")
else:
    print(f"Failed to retrieve page. Status code: {response.status_code}")

```

as an example for the output today :

	A	B	C
1	Course Name	Course URL	Coupon Code
2	Mastering Conversio	https://www.udemy.c	93646E3ECD40BC5
3	AI for All – Getting st	https://www.udemy.c	59F556FB02B51363
4	CapCut Mastery: Fro	https://www.udemy.c	359E682E0BA791B
5	RPA Project: Gather	https://www.udemy.c	51EE27250FFB0622
6	RPA Project: Proces	https://www.udemy.c	C7C2C030BF33893
7	RPA Project: Mappir	https://www.udemy.c	45FDC2884DBF2FF
8	ML for Business Man	https://www.udemy.c	FREENOV24
9	Artificial Intelligence	https://www.udemy.c	5B2E4404D8D9FD1
10	Artificial Intelligence	https://www.udemy.c	FE3ECCD115D1DE8
11	Entrepreneurship an	https://www.udemy.c	FF387060163821E3
12	GIS Software & Appl	https://www.udemy.c	0E6AA3A67F02EFB
1	edical Robotics Cc	https://www.udemy.c	1D7550B8092D8F3D
1	Solar Cell Technolo	https://www.udemy.c	F9BA4A50B06009F
15	Introduction to Finar	https://www.udemy.c	INTROFINPRSER01
16	Vigilant Leader: Lea	https://www.udemy.c	7E08146CAFDD228
17	Professional Certifica	https://www.udemy.c	B56340B4F4134F78
18	Professional Certifica	https://www.udemy.c	F8ECDB1EBE7212B
19	Diploma: Human Re	https://www.udemy.c	7A20123E25D23B63
20	Professional Certifica	https://www.udemy.c	0326C7EF3B7DBEC
21	Sales Operations the	https://www.udemy.c	CC98AE8C5AF93E9
22	Accelerate Your Lea	https://www.udemy.c	091278F74747C2C5
23	Time Management A	https://www.udemy.c	CB7DA4DB5C8D64E
24	Developing successf	https://www.udemy.c	DEVSUCPROFREL1
25	Ultimate Guide to Ca	https://www.udemy.c	810A45B15A011BC
26	Medical and Cosmet	https://www.udemy.c	ICANVA
27	Professional Certifica	https://www.udemy.c	A06D80C9F1FD7958
28	Advanced Project M	https://www.udemy.c	0851823A5F191BD3

from this step i created a new GUI using tkinter that do the following

first imports i need

```
import tkinter as tk
from tkinter import messagebox
from tkinter import ttk
import csv
import random
import re
from selenium import webdriver
import time
import logging
```

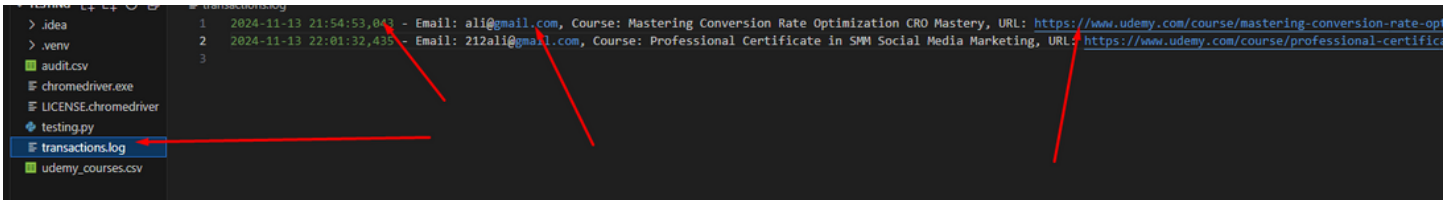
second describing the used functions

setup_logging()

this function setup logging to file named transactions.log that contain
gmail loggined , url and timestamp that he logged in

```
# Set up logging to log all transactions
def setup_logging():
    logging.basicConfig(
        filename='transactions.log',
        level=logging.INFO,
        format='%(asctime)s - %(message)s',
        filemode='a'
    )
```

here is an example of transactions.log file



this function i use to select a random course and go to the url of it but i didnt use it as i made a combo box contain all urls for all courses

```python
# Function to read data from the CSV file and choose a random course
def select_random_course():
    try:
        with open('udemy_courses.csv', mode='r', encoding='utf-8') as file:
            reader = csv.DictReader(file)
            courses = list(reader)
            if courses:
                return random.choice(courses)
            else:
                messagebox.showerror("Error", "No courses found in udemy_courses.csv.")
                return None
    except FileNotFoundError:
        messagebox.showerror("Error", "The file 'udemy_courses.csv' was not found.")
        return None
```

save_selection() : this function that saves email, URL on the audit.csv file

```python
# Function to save selected course with email to a new CSV file
def save_selection(email, course_name, course_url):
    try:
        with open('audit.csv', mode='a', newline='', encoding='utf-8') as file:
            writer = csv.writer(file)
            writer.writerow([email, course_name, course_url])  # Save email, course name, and URL
        messagebox.showinfo("Success", f"Course saved for {email}")
        # Log the transaction in the log file
        log_transaction(email, course_name, course_url)
    except Exception as e:
        messagebox.showerror("Error", f"Error saving data: {e}")
```

as shown here in audit.csv file it saves the gmail trying to enter the url in the database

now log the transactions to log file early created

then open the link in google chrom that you can apply or add to cart

```python
# Log the transaction in the log file
def log_transaction(email, course_name, course_url):
    logging.info(f"Email: {email}, Course: {course_name}, URL: {course_url}")

# Function to open the course link using Selenium
def open_course_link(url):
    driver = webdriver.Chrome()  # Ensure ChromeDriver is in your PATH or provide its path he
    driver.get(url)  # Just open the URL without clicking anything
    time.sleep(20)  # Wait for a few seconds to ensure the page loads fully

    driver.quit()  # Close the browser after opening the link
```

function update_combobox() : that gets all urls in the .csv file for the next step to choose between them

```python
# Function to update the combo box with the available course names
def update_combobox():
    try:
        with open('udemy_courses.csv', mode='r', encoding='utf-8') as file:
            reader = csv.DictReader(file)
            courses = list(reader)
            course_names = [course['Course Name'] for course in courses]
            url_mapping = {course['Course Name']: course['Course URL'] for course in courses}
            url_combobox['values'] = course_names  # Populate the combobox with course names
            if course_names:
                url_combobox.set(course_names[0])  # Set the default value of the combobox
            else:
                messagebox.showerror("Error", "No course names found in udemy_courses.csv.")
            return url_mapping
    except FileNotFoundError:
        messagebox.showerror("Error", "The file 'udemy_courses.csv' was not found.")
        return {}
```

now open the course link i chooses

also i add some error handling as if you typed an email that not end with @gmail.com it will not execute

```python
# Button function to process the email, validate, and open the course link
def process_email():
    email = email_entry.get().strip()
    if not re.match(r'^[\w\.-]+@gmail\.com$', email):
        messagebox.showerror("Invalid Email", "Please enter a valid Gmail address ending with @gmail.com.")
        return

    # Save the Gmail address when validated
    global user_email
    user_email = email

    selected_course_name = url_combobox.get()  # Get the selected course name from the combobox
    if not selected_course_name:
        messagebox.showerror("Error", "Please select a valid course.")
        return

    # Find the course with the selected name
    course_url = url_mapping.get(selected_course_name)
    if not course_url:
        messagebox.showerror("Error", "Could not find the selected course.")
        return

    # Open the course link without clicking anything
    open_course_link(course_url)

    # Save the selected course along with the email and course URL to the CSV file
    save_selection(user_email, selected_course_name, course_url)
```

setting up gui
note: ChatGPT helped a lot in optimization and good visualization

```python
# Setting up the Tkinter GUI
root = tk.Tk()
root.title("Udemy Course Selector")
root.geometry("500x350")
root.config(bg="#f0f0f0")  # Light gray background

# Adding a title label with a larger font
title_label = tk.Label(root, text="Udemy Course Selector", font=("Helvetica", 18, "bold"), bg="#f0f0f0")
title_label.pack(pady=20)

# Frame for better layout management
frame = tk.Frame(root, bg="#f0f0f0")
frame.pack(pady=20)

# Label and entry for email input with custom styling
email_label = tk.Label(frame, text="Enter your Gmail address:", font=("Helvetica", 12), bg="#f0f0f0")
email_label.grid(row=0, column=0, padx=10, pady=10)

email_entry = tk.Entry(frame, width=30, font=("Helvetica", 12), bd=2)
email_entry.grid(row=0, column=1, padx=10, pady=10)

# Label for combobox
url_label = tk.Label(frame, text="Select a course:", font=("Helvetica", 12), bg="#f0f0f0")
url_label.grid(row=1, column=0, padx=10, pady=10)

# Combobox for selecting a course name
url_combobox = ttk.Combobox(frame, width=30, font=("Helvetica", 12))
url_combobox.grid(row=1, column=1, padx=10, pady=10)

# Button to submit email and select a course with a hover effect
def on_enter(event):
    submit_button.config(bg="#4CAF50")
```

Setting up the buttons

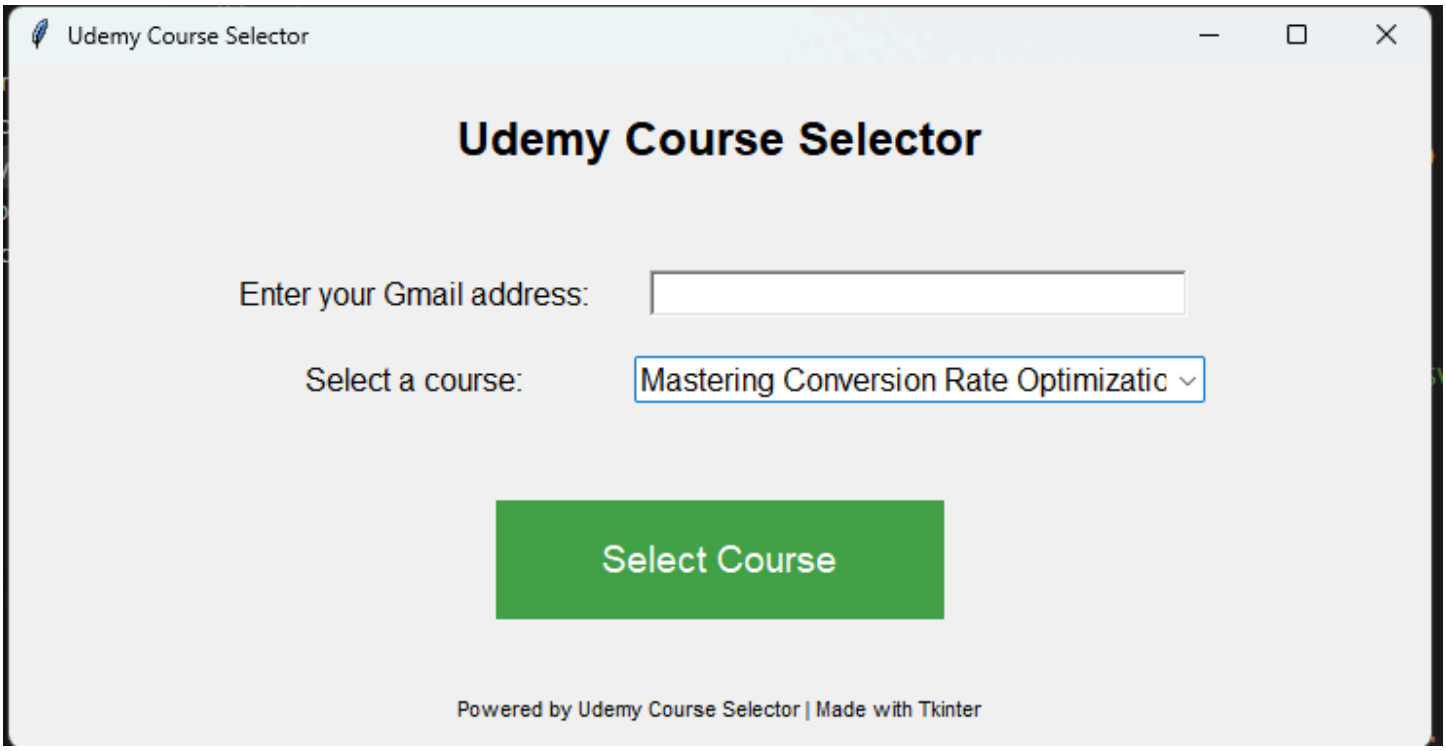
```python
# Button to submit email and select a course with a hover effect
def on_enter(event):
    submit_button.config(bg="#4CAF50")

def on_leave(event):
    submit_button.config(bg="#45a049")

submit_button = tk.Button(root, text="Select Course", font=("Helvetica", 14), bg="#45a049", fg="white", command=process_email, relief="flat", height=2, width=20)
submit_button.pack(pady=20)
submit_button.bind("<Enter>", on_enter)
submit_button.bind("<Leave>", on_leave)

# Footer Label for additional information or instructions
footer_label = tk.Label(root, text="Powered by Udemy Course Selector | Made with Tkinter", font=("Helvetica", 8), bg="#f0f0f0")
footer_label.pack(side="bottom", pady=10)

# Update the combobox with the course names when the program starts
url_mapping = update_combobox()

# Initialize logging at the start
setup_logging()

# Keep the window open until the user manually closes it
root.mainloop()
```

==========

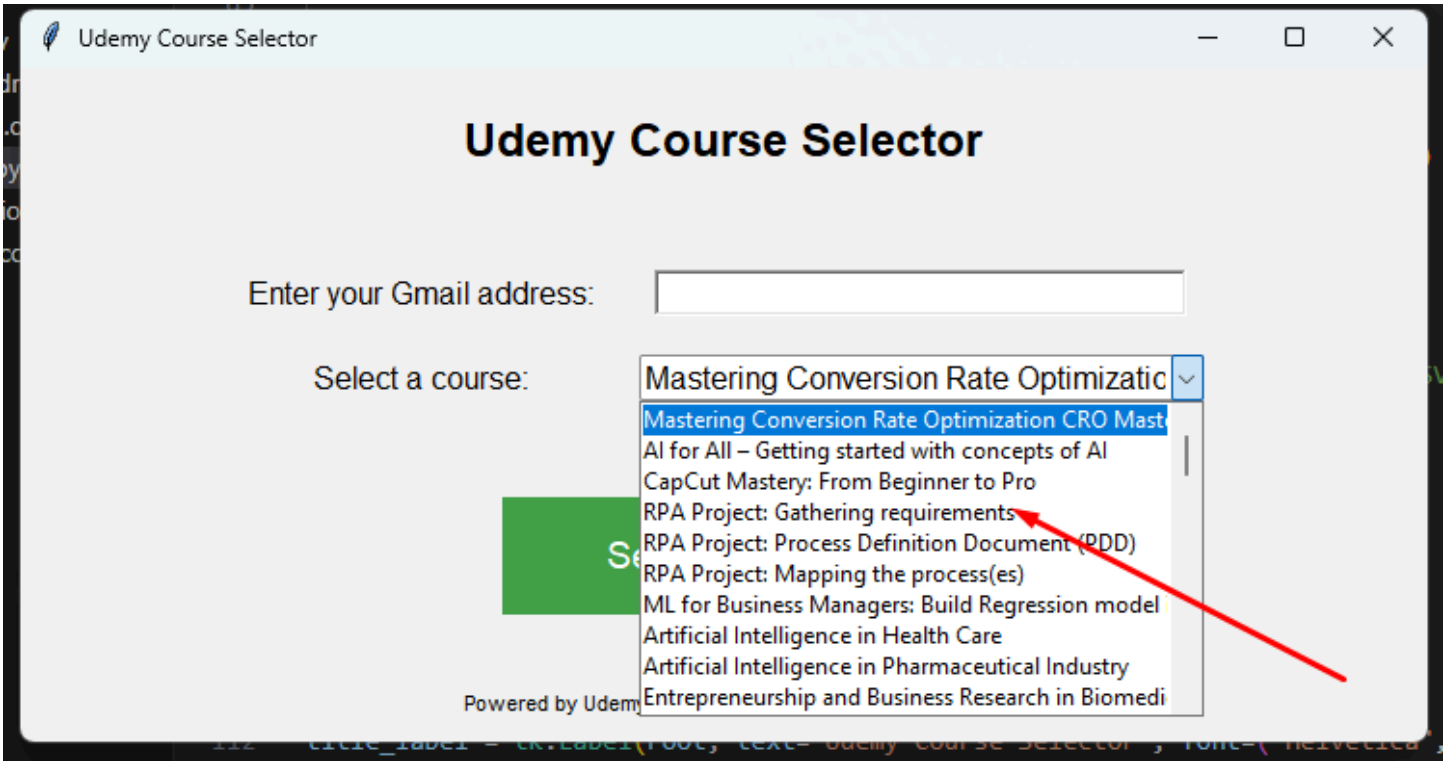
Live action for the GUI

==========

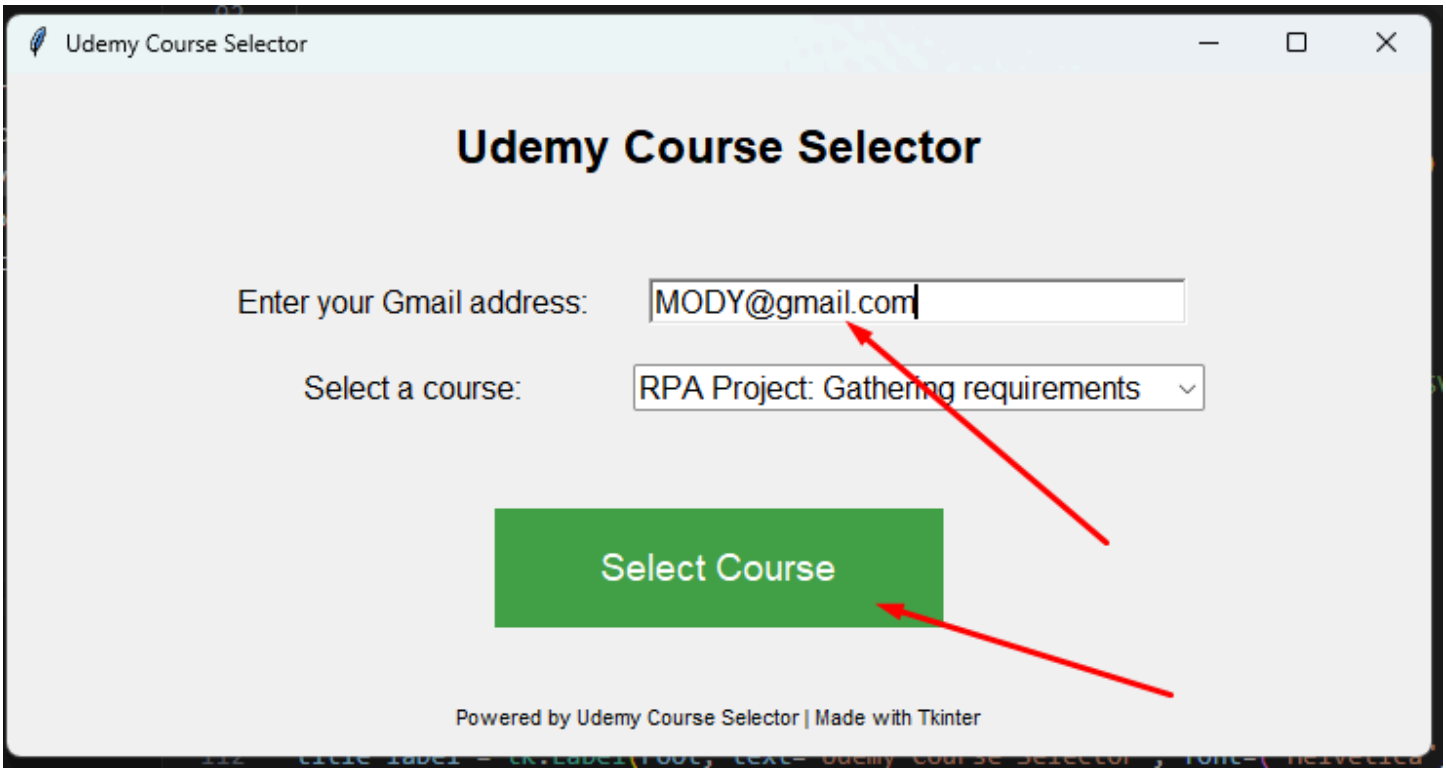
here is a The GUI



from this combo box i choose between courses i scrapped

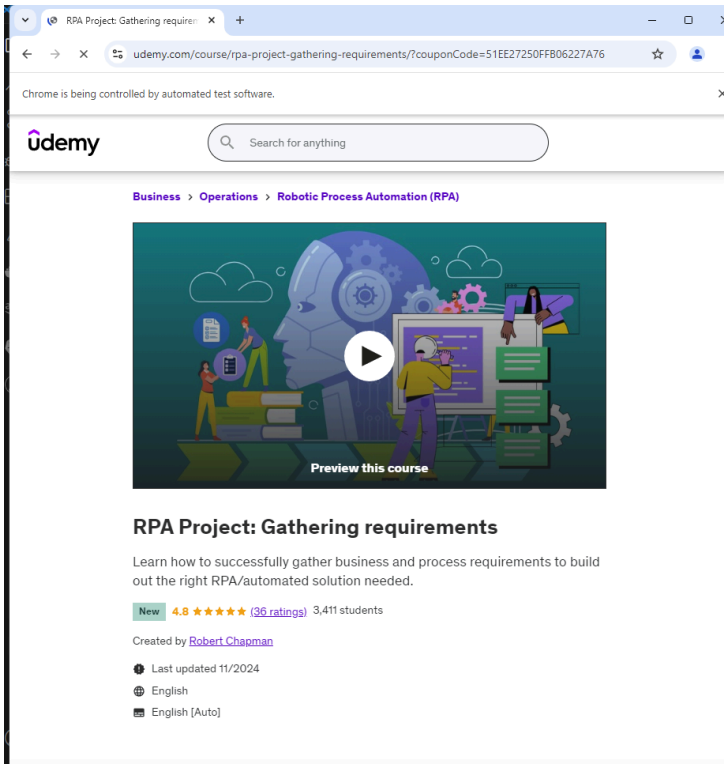


add my email here

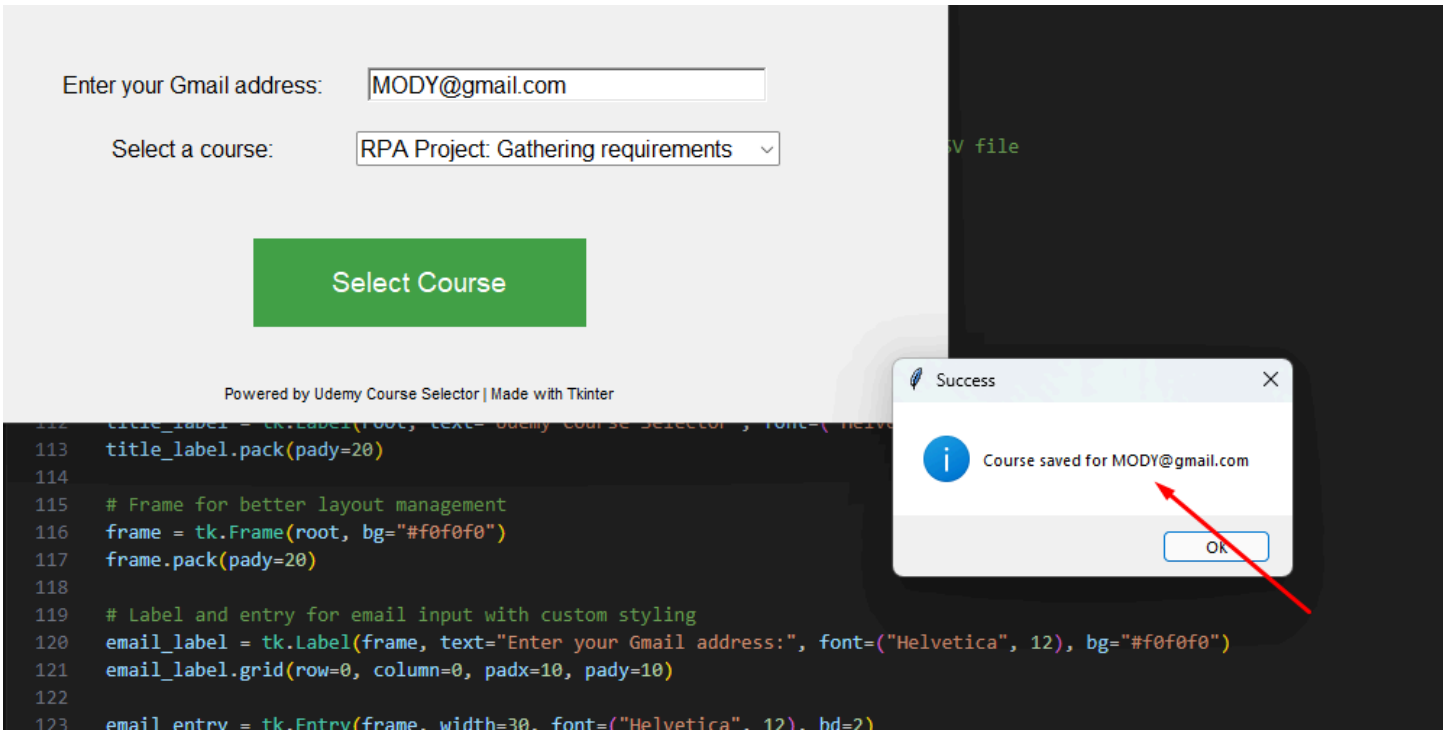


here it enters the course note that : i chooses rpa course in combo and it enters it

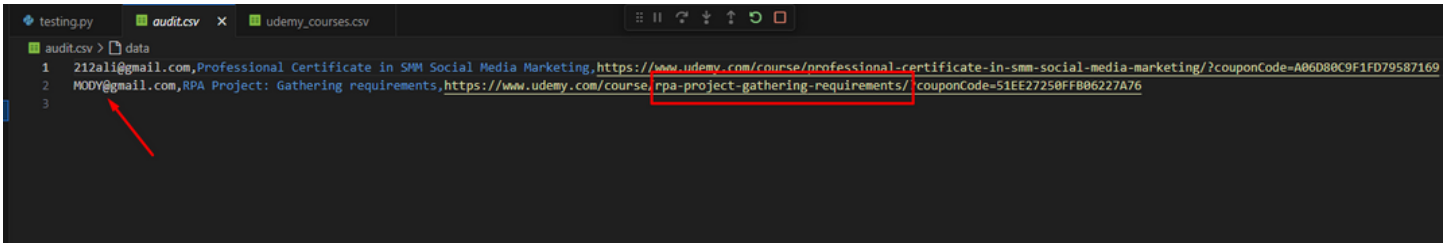
now i can add interactive like add it to cart but i didn't as cloud fare get me out



now the gmail added to the audit.csv

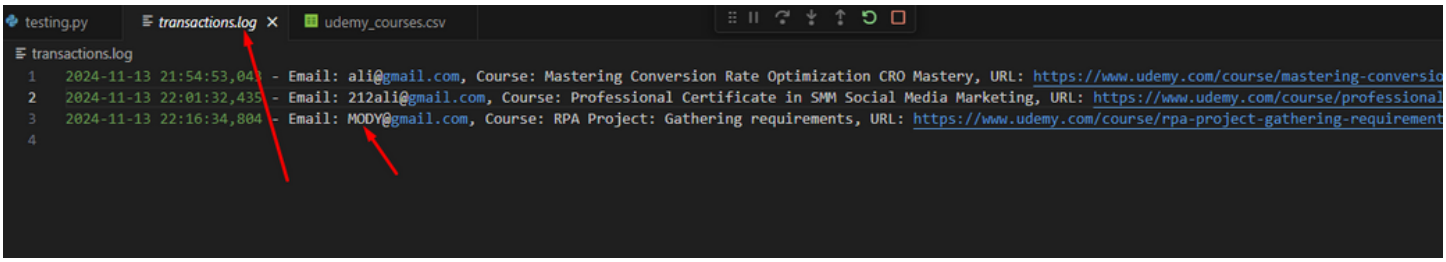


clear

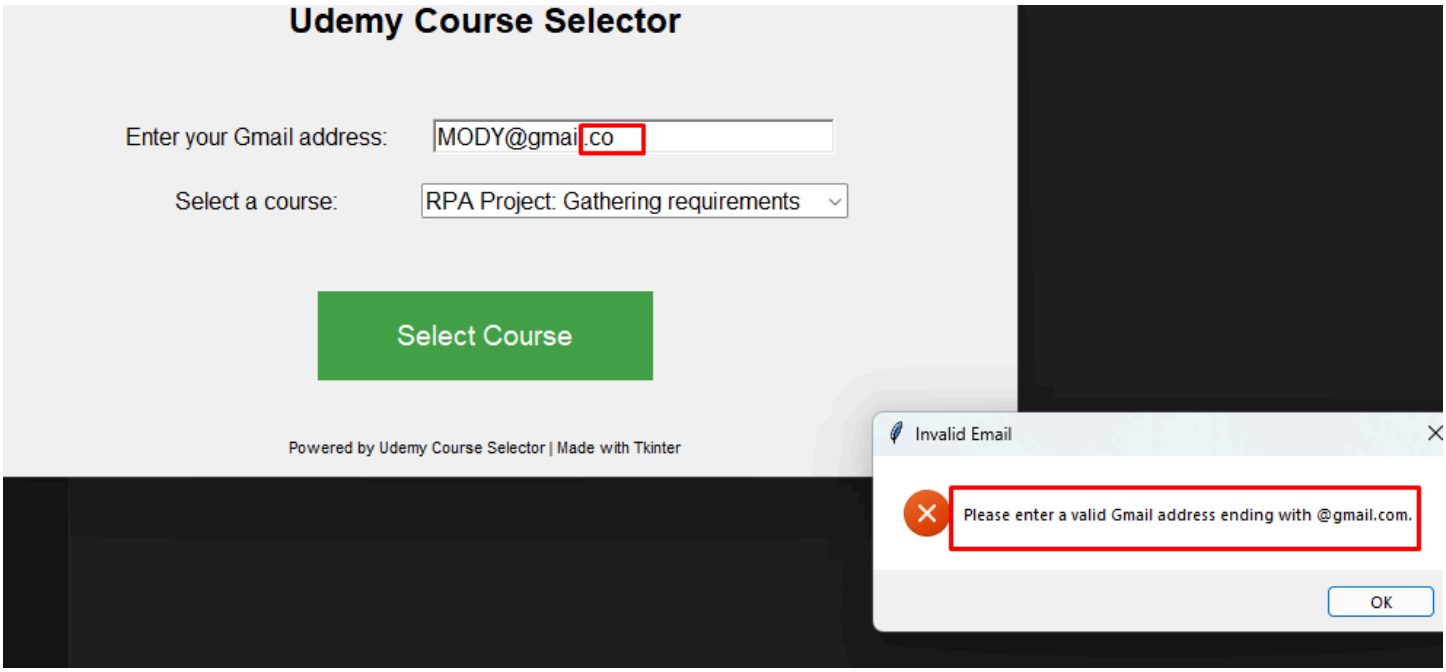


also transactions.log

note only 1 transaction added as only this GUI enters the URL but you do the other work



error handling as when u forget @gmail.com it gives you an error



google drive link : udemy_scraper - Google Drive