```python
from selenium.webdriver.common.by import By as by
import json, time, os, random
from random_user_agent.user_agent import UserAgent
from random_user_agent.params import SoftwareName, HardwareType
from importlib.resources import path
from webdriver_manager.chrome import ChromeDriverManager as CM
import random, string, io, os, shutil, platform, subprocess, sys, zipfile, time
import threading
import pyaudio
import re

from selenium import webdriver
from selenium.webdriver.chrome.options import Options

import undetected_chromedriver._compat as uc
from undetected_chromedriver.patcher import Patcher

from pathlib import Path
from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage, messagebox
from tkinter import *
from tkinter import ttk

import random
import json
import time
import requests
import string
from datetime import datetime
import gc
import sys
import os
import traceback
import threading
import uuid


import undetected_chromedriver as uc
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities
from selenium.webdriver.support.ui import Select
import shutil
import wave

global stop_event
stop_event = threading.Event()

def random_user_agent():
 return UserAgent(software_names=[SoftwareName.CHROME.value], hardware_types={HardwareType.COMPUTER.value},
limit=100).get_random_user_agent()

"""
def get_random_name():
    return json.loads(requests.get("https://api.namefake.com/").text)["name"]

def check_phone_verification(driver, api_key):
    if driver.current_url == "https://twitter.com/account/access":
        try:
            driver.find_element(by.XPATH, "//select[@id='country_code']/option[@value='7']")
            pv = PhoneVerification(api_key)
            number, _id = pv.get_number()
            phone_number = driver.find_element(by.XPATH, "//input[@name='']"); phone_number.click();
phone_number.send_keys(number)
            time.sleep(0.5)
            driver.find_element(by.XPATH, "//input[@value='Next']").click(); time.sleep(5.5)
            ver = driver.find_element(by.XPATH, "//input[@name='pin']")
            time.sleep(5.5)
            ver_code = pv.get_sms(_id)
            ver.click(); ver.send_keys(ver_code); time.sleep(1.5)
            driver.find_element(by.XPATH, "//input[@value='Next']").click(); time.sleep(5.5)
        except Exception:
            pass
"""
```

```
def init_driver(proxy_address=None, headless=False):
    chrome = Chrome()
    return chrome.webdriver(proxy_address=proxy_address)

def signup(gameNum=None, proxyy=None):
    try:
        if not stop_event.is_set():
            canvas.itemconfig(texti, text="TOOL STOPPED!",fill="#FCFCFC")
            try:
                driver.quit()
            except:
                pass
            return
        try:
            print(str(gameNum))
            gameNum1 = str(gameNum[0])
            timeSleepi = int((gameNum[1]))
            print(str(gameNum1))
            print(str(timeSleepi))
            gameNum1 = [gameNum1]
            print(str(gameNum1))
        except:
            ee = traceback.format_exc()
            print(ee)
            canvas.itemconfig(texti, text="ERROR!",fill="#FCFCFC")
            messagebox.showerror("ERROR!", "Error - Game number and time to wait not in the right order!
Should be '5,10' for example.")
            stop_event.clear()
            try:
                driver.quit()
            except:
                pass
            return
        for gamiNumi in gameNum1:
            if not stop_event.is_set():
                canvas.itemconfig(texti, text="TOOL STOPPED!",fill="#FCFCFC")
                try:
                    driver.quit()
                except:
                    pass
                return
            canvas.itemconfig(texti, text="CHECKING FOR TICKETS ON GAME NUMBER:
"+str(gamiNumi),fill="#FCFCFC")
            gamiNumiFinal = int(gamiNumi)-1
            if gameNum1 == None:
                print("gameNum vazio")
                canvas.itemconfig(texti, text="ERROR!",fill="#FCFCFC")
                messagebox.showerror("ERROR!", "Error - No Game Number was Provided.")
                stop_event.clear()
                try:
                    driver.quit()
                except:
                    pass
                return
            elif len(gameNum1)==0:
                print("gameNum lista vazia")
                canvas.itemconfig(texti, text="ERROR!",fill="#FCFCFC")
                messagebox.showerror("ERROR!", "Error - No Game Number was Provided.")
                stop_event.clear()
                try:
                    driver.quit()
                except:
                    pass
                return
            if proxyy == None:
                print("proxy dado vazio")
                canvas.itemconfig(texti, text="ERROR!",fill="#FCFCFC")
                messagebox.showerror("ERROR!", "Error - No Proxy was Provided.")
                stop_event.clear()
                try:
                    driver.quit()
                except:
                    pass
                return
            proxy = random.choice(proxyy)
            driver = init_driver(proxy_address=proxy)
            #driver.execute_cdp_cmd("Network.setUserAgentOverride", {"userAgent": f"{random_user_agent()}"})
            queueDone = False
            firt = False
            while(True):
                if not firt:
                    driver.get("https://tickets.rugbyworldcup.com/en/resale_france_new_zealand")
                    firt = True
                else:
                    driver.get("https://tickets.rugbyworldcup.com/en/")
                try:
                    #QUEUEEEEEEEEE
                    try:#check se tem queue
                        WebDriverWait(driver, .5).until(EC.element_to_be_clickable((By.XPATH,
'//h1[contains(text(), "403 ERROR")]')))
                        driver.get("https://tickets.rugbyworldcup.com/en/")
                        try:
                            WebDriverWait(driver, 3).until(EC.element_to_be_clickable((By.XPATH,
'/html/body/div[3]/div/div/div[3]/button'))).click()
                            try:
                                WebDriverWait(driver, 5).until(EC.element_to_be_clickable((By.XPATH, '//*
[@id="root"]/div/div[2]/div[1]/h2')))
                            except Exception as e:
                                print("wait for queue not found the first time, error maybe?")
                                canvas.itemconfig(texti, text="ERROR!",fill="#FCFCFC")
                                messagebox.showerror("ERROR!", "ERROR OCCURED: "+str(e))
                                stop_event.clear()
                                try:
                                    driver.quit()
                                except:
                                    pass
                                return
                            while(True):#You are in the queue. Please wait...
                                try:
                                    WebDriverWait(driver, 2).until(EC.element_to_be_clickable((By.XPATH, '//*
[@id="root"]/div/div[2]/div[1]/h2')))
                                    time.sleep(1)
                                except:
                                    break
                            try:#cookies
                                WebDriverWait(driver, 5).until(EC.element_to_be_clickable((By.XPATH, '//*
[@id="onetrust-accept-btn-handler"]'))).click()
                            except:
                                print("no cookies!")
                        except:
                            if not queueDone:
                                try:#cookies
                                    WebDriverWait(driver, 5).until(EC.element_to_be_clickable((By.XPATH, '//*
[@id="onetrust-accept-btn-handler"]'))).click()
                                except:
                                    pass
                    except:
                        print("no queue")
                        if not queueDone:
                            try:#cookies
                                WebDriverWait(driver, 5).until(EC.element_to_be_clickable((By.XPATH, '//*
[@id="onetrust-accept-btn-handler"]'))).click()
                            except:
                                print("no cookies!")
                    queueDone = True
                    #MAIN MENUS AREA
                    #//*[@id="block-hometicketing"]/div[2]/div[2]/fieldset[1]/div/div/div/div[3]/div[1]
                    #//*[@id="block-hometicketing"]/div[2]/div[2]/fieldset[2]/div/div[1]/div/div[3]/div[1]
                    #//*[@id="block-hometicketing"]/div[2]/div[2]/fieldset[2]/div/div[2]/div/div[3]/div[1]
                    if not stop_event.is_set():
                        canvas.itemconfig(texti, text="TOOL STOPPED!",fill="#FCFCFC")
                        try:
                            driver.quit()
                        except:
                            pass
                        return
                    try:
                        tryAdd = WebDriverWait(driver, 1.5).until(EC.element_to_be_clickable((By.XPATH, '//*
[@id="block-hometicketing"]/div[2]/div[2]/fieldset[2]/div/div[1]/div/div[3]/div[1]')))
                        #ActionChains(driver).move_to_element(tryAdd).click(tryAdd).perform()
                        try:
                            #WebDriverWait(driver, 5).until(EC.presence_of_element_located((By.XPATH,
'//a[contains(@class, "btn-resale")]')))
                            tryAdd = driver.find_elements(By.XPATH, '//a[contains(@class, "btn-resale")]')
                            lili = 0
                            liliEncon = False
                            """for tryied in tryAdd:
                                if lili==gamiNumiFinal:
                                    driver.execute_script("arguments[0].click();", tryied)
                                    liliEncon = True
                                    break
                                else:
                                    lili+=1"""
                            if not liliEncon:
                                tryAdd = driver.find_elements(By.XPATH, '//h3[contains(@class, "match-
label")]')
                                lili = 0
                                liliEncon = False
                                for tryied in tryAdd:
                                    if lili==gamiNumiFinal:
                                        print(str(tryied.text))
                                        list_of_words = re.split('v(?=[A-Z])', str(tryied.text))
                                        lowercase_words = [word.lower().replace(" ","_") for word in
list_of_words]
                                        print(str(lowercase_words))
                                        my_href =
"https://tickets.rugbyworldcup.com/en/resale_"+str(lowercase_words[0])+"_"+str(lowercase_words[1])
                                        tryAdd55 = driver.find_elements(By.XPATH, '//div[contains(@class,
"btn btn-primary noloader unavailable")]')
                                        lili55 = 0
                                        alili = False
                                        for tryied55 in tryAdd55:
                                            if lili55==gamiNumiFinal:
                                                driver.execute_script("""
                                                    var div = arguments[0];
                                                    var href = arguments[1];
                                                    var a = document.createElement('a');
                                                    a.setAttribute('href', href);
                                                    div.parentNode.insertBefore(a, div);
                                                    a.appendChild(div);
                                                """, tryied55, my_href)
                                                driver.execute_script("arguments[0].click();", tryied55)
                                                alili = True
                                                break
                                            else:
                                                lili55+=1
                                        break
                                    else:
                                        lili+=1
                                if not alili:
                                    print("jogo dado nao encontrado!")
                                    canvas.itemconfig(texti, text="WARNING!",fill="#FCFCFC")
                                    messagebox.showwarning("WARNING!", "WARNING - Game Number:
"+str(gamiNumiFinal)+" was not found! Press OK to try the next games.")
                                    try:
                                        driver.quit()
                                    except:
                                        pass
                                    continue
                        except:
                            ee = traceback.format_exc()
                            print(ee)
                            print("couldn't find resale tickets link!")
                            canvas.itemconfig(texti, text="FATAL ERROR!",fill="#FCFCFC")
                            messagebox.showerror("FATAL ERROR!", "FATAL ERROR OCCURED WHILE TRYING TO FIND THE
BUY RESALE TICKETS BUTTON: "+str(e))
                            stop_event.clear()
                            try:
                                driver.quit()
                            except:
                                pass
                            return
                    except:
                        print("couldn't find game!")
                        if not firt:
                            canvas.itemconfig(texti, text="WARNING!",fill="#FCFCFC")
                            messagebox.showwarning("WARNING!", "WARNING -Game Number: "+str(gamiNumiFinal)+"
was not found! Press OK to try the next games.")
                            try:
                                driver.quit()
                            except:
                                pass
                            continue
                    while(True):
                        if not stop_event.is_set():
                            canvas.itemconfig(texti, text="TOOL STOPPED!",fill="#FCFCFC")
                            try:
                                driver.quit()
                            except:
                                pass
                            return
                        try:
                            #BUY TICKET, CHECK TICKET AREA
                            try:
                                WebDriverWait(driver,
1.25).until(EC.element_to_be_clickable((By.CSS_SELECTOR, '[class="nb-tickets"]')))
                                catAEscolher = 1
                                totalCats = 0
                                breaka = False
                                categories_available = driver.find_elements(By.XPATH, '//*[@id="edit-tickets-
list"]/tbody/tr')
                                totalCats = len(categories_available)
                                while(True):
                                    if breaka:
                                        break
                                    if catAEscolher > totalCats:
                                        break
                                    try:
                                        categories_available = driver.find_elements(By.XPATH, '//*[@id="edit-
tickets-list"]/tbody/tr')
                                    except Exception as e:
                                        ee = traceback.format_exc()
                                        print(ee)
                                        print("couldn't find ticket categories, but tickets seem to be
available?")
                                        canvas.itemconfig(texti, text="FATAL ERROR!",fill="#FCFCFC")
                                        messagebox.showerror("FATAL ERROR!", "FATAL ERROR OCCURED, IT SEEMS
THAT TICKETS WERE AVAILABLE BUT I WAS UNABLE TO OPEN THE CATEGORIES: "+str(e))
                                        stop_event.clear()
                                        try:
                                            driver.quit()
                                        except:
                                            pass
                                        return
                                    inCat = 1
                                    print(str(len(categories_available)))
                                    for categoria in categories_available:
                                        if inCat < catAEscolher:
                                            inCat+=1
                                            continue
                                        categoria.click()
                                        try:
                                            eflie = 1
                                            altsjdsa = False
                                            try:
                                                WebDriverWait(driver,
1).until(EC.element_to_be_clickable((By.XPATH, '//div[@class="resale-listing-action"]/button'))).click()
                                            except:
                                                try:
                                                    cartiss =
driver.find_elements(By.XPATH,'//div[@class="resale-listing-action"]/button')
                                                except:
                                                    cartiss =
driver.find_elements(By.XPATH,'//div[@class="resale-listing-action"]/button')
                                                for carti in cartiss:
                                                    if eflie < catAEscolher:
                                                        eflie+=1
                                                        continue
                                                    try:
                                                        carti.click()
                                                    except:
                                                        print("no ticket?")
                                                        altsjdsa = True
                                                        break
                                            if altsjdsa:
                                                continue
                                            try:#add to cart
                                                WebDriverWait(driver,
2).until(EC.element_to_be_clickable((By.XPATH, '//a[@data-drupal-selector="edit-show-product-
cart"]'))).click()
                                                breaka = True
                                                break
                                            except:
                                                try:#check if already purchased
                                                    WebDriverWait(categoria, 
5).until(EC.element_to_be_clickable((By.XPATH, '//button[@class="ui-dialog-titlebar-close"]'))).click()
                                                    breaka = False
                                                    catAEscolher+=1
                                                    break
                                                except:
                                                    pass
                                                try:#add to cart
                                                    WebDriverWait(driver, 
8).until(EC.element_to_be_clickable((By.XPATH, '//a[@data-drupal-selector="edit-show-product-
cart"]'))).click()
                                                    breaka = True
                                                    break
                                                except Exception as e:
                                                    ee = traceback.format_exc()
                                                    print(ee)
                                                    print("couldn't find the cart!")
                                                    canvas.itemconfig(texti, text="FATAL 
ERROR!",fill="#FCFCFC")
                                                    messagebox.showerror("FATAL ERROR!", "FATAL ERROR OCCURED 
WHILE TRYING TO ADD THE TICKETS TO THE CART: "+str(e))
                                                    stop_event.clear()
                                                    try:
                                                        driver.quit()
                                                    except:
                                                        pass
                                                    return
                                        except Exception as e:
                                            ee = traceback.format_exc()
                                            print(ee)
                                            print("couldn't add to cart!")
                                            canvas.itemconfig(texti, text="FATAL ERROR!",fill="#FCFCFC")
                                            messagebox.showerror("FATAL ERROR!", "FATAL ERROR OCCURED WHILE 
TRYING TO VIEW THE CART: "+str(ee))
                                            stop_event.clear()
                                            try:
                                                driver.quit()
                                            except:
                                                pass
                                            return
                                if not breaka:
                                    print("no tickets are available!")
                                    #driver.quit()
                                    time.sleep(timeSleepi)
                                    driver.refresh()
                                    continue
                                else:
                                    break
                            except:
                                print("no tickets are available (error)!")
                                #driver.quit()
                                time.sleep(timeSleepi)
                                driver.refresh()
                                continue
                        except:
                            ee = traceback.format_exc()
                            print(ee)
                            with open("errorLog.txt", "a") as f: f.write(f"{datetime.now().strftime('%Y-%m-%d 
%H:%M:%S')} - "+str(ee)+"\n")
                    print("done and added ticket!")
                    canvas.itemconfig(texti, text="TICKET(s) FOUND FOR GAME NUMBER: "+str(gamiNumi), 
fill="#90EE90")
                    #play sound!
                    while not stop_sound_loop:
                        # open the file for reading.
                        wf = wave.open(relative_to_assets("ale.wav"), 'rb')
                        chunk = 1024

                        # create an audio object
                        p = pyaudio.PyAudio()

                        # open stream based on the wave object which has been input.
                        stream = p.open(format =
                                        p.get_format_from_width(wf.getsampwidth()),
                                        channels = wf.getnchannels(),
                                        rate = wf.getframerate(),
                                        output = True)

                        # read data (based on the chunk size)
                        data = wf.readframes(chunk)

                        # play stream (looping from beginning of file to the end)
                        while data:
                            # writing to the stream is what *actually* plays the sound.
                            stream.write(data)
                            data = wf.readframes(chunk)


                        # cleanup stuff.
                        wf.close()
                        stream.close()
                        p.terminate()
                        time.sleep(2)
                    while stop_event.is_set():
                        time.sleep(1)
                    stop_event.clear()
                    canvas.itemconfig(texti, text="FINISHED! Press Start to Start Again!",fill="#FCFCFC")
                    return
                except Exception as e:
                    canvas.itemconfig(texti, text="FATAL ERROR!",fill="#FCFCFC")
                    messagebox.showerror("FATAL ERROR!", "FATAL ERROR OCCURED: "+str(e))
                    stop_event.clear()
                    try:
                        driver.quit()
                    except:
                        pass
                    return
    except:
        ee = traceback.format_exc()
        print(ee)
        messagebox.showerror("FATAL ERROR!", "ERROR: "+str(ee))

def setup_useragent(driver):
    driver.execute_cdp_cmd("Network.setUserAgentOverride", {"userAgent": f"{random_user_agent}"})

def Proxy(PROXY_HOST, PROXY_PORT, PROXY_USER, PROXY_PASS, i):
 try:
  manifest_json = """
  {
   "manifest_version": 2,
   "name": "Proxy Manager",
   "version": "3.0.11",
   "permissions": [
    "proxy",
    "tabs",
    "unlimitedStorage",
    "storage",
    "<all_urls>",
    "webRequest",
    "webRequestBlocking"
   ],
   "background": {
    "scripts": ["background.js"]
   },
   "minimum_chrome_version":"22.0.0"
  }
  """

  background_js = string.Template(
  """
  var config = {
    mode: "fixed_servers",
    rules: {
    singleProxy: {
     scheme: "http",
     host: "${PROXY_HOST}",
     port: parseInt(${PROXY_PORT})
    },
    bypassList: ["foobar.com"]
    }
   };
  chrome.proxy.settings.set({value: config, scope: "regular"}, function() {});
  function callbackFn(details) {
   return {
    authCredentials: {
     username: "${PROXY_USER}",
     password: "${PROXY_PASS}"
    }
   };
  }
  chrome.webRequest.onAuthRequired.addListener(
     callbackFn,
     {urls: ["<all_urls>"]},
     ['blocking']
  );
  """
  ).substitute(
   PROXY_HOST=PROXY_HOST,
   PROXY_PORT=PROXY_PORT,
   PROXY_USER=PROXY_USER,
   PROXY_PASS=PROXY_PASS)

  if not os.path.exists("data/extension"):
   os.makedirs("data/extension")

  with zipfile.ZipFile(f'data/extension/proxy_auth_plugin_{i}.zip', 'w', zipfile.ZIP_DEFLATED, False) as zp:
   zp.writestr('manifest.json', manifest_json)
   zp.writestr('background.js', background_js)
  return f"data/extension/proxy_auth_plugin_{i}.zip"
 except Exception as e:
  return False
  now = datetime.now().strftime('%H:%M:%S')
  print(f'[{now}] - {e}')

class bcolors:
    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKCYAN = '\033[96m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'

CHROME = ['{8A69D345-D564-463c-AFF1-A69D9E530F96}',
          '{8237E44A-0054-442C-B6B6-EA0509993955}',
          '{401C381F-E0DE-4B85-8BD8-3F3F14FBDA57}',
          '{4ea16ac7-fd5a-47c3-875b-dbf4a2008c20}']

def download_driver():
    OSNAME = platform.system()
    print(bcolors.WARNING + 'Getting Chrome Driver...' + bcolors.ENDC)
    if OSNAME == 'Linux':
        OSNAME = 'lin'
        EXE_NAME = ""
        with subprocess.Popen(['google-chrome', '--version'], stdout=subprocess.PIPE) as proc:
            version = proc.stdout.read().decode('utf-8').replace('Google Chrome', '').strip()
    elif OSNAME == 'Darwin':
        OSNAME = 'mac'
        EXE_NAME = ""
        process = subprocess.Popen(['/Applications/Google Chrome.app/Contents/MacOS/Google Chrome', '--
version'], stdout=subprocess.PIPE)
        version = process.communicate()[0].decode('UTF-8').replace('Google Chrome', '').strip()
    elif OSNAME == 'Windows':
        OSNAME = 'win'
        EXE_NAME = ".exe"
        version = None
        try:
            process = subprocess.Popen(['reg', 'query',
'HKEY_CURRENT_USER\\Software\\Google\\Chrome\\BLBeacon', '/v', 'version'], stdout=subprocess.PIPE,
stderr=subprocess.DEVNULL, stdin=subprocess.DEVNULL)
            version = process.communicate()[0].decode(
                'UTF-8').strip().split()[-1]
        except:
            for i in CHROME:
                for j in ['opv', 'pv']:
                    try:
                        command = ['reg', 'query',
f'HKEY_LOCAL_MACHINE\\Software\\Google\\Update\\Clients\\{i}', '/v', f'{j}', '/reg:32']
                        process = subprocess.Popen(command, stdout=subprocess.PIPE,
stderr=subprocess.DEVNULL, stdin=subprocess.DEVNULL)
```

```
                        version = process.communicate()[0].decode('UTF-8').strip().split()[-1]
                    except:
                        pass
        if not version:
            print(bcolors.WARNING + "Couldn't find your Google Chrome version automatically!" + bcolors.ENDC)
            version = input(bcolors.WARNING + 'Please input your google chrome version (ex: 91.0.4472.114) : '
+ bcolors.ENDC)
    else:
        print('{} OS is not supported.'.format(OSNAME))
        sys.exit()
    uc.install()

class Chrome():
    CHROMEDRIVER = None
    user_dir = None

    def __init__(self):
        # uc.install(); time.sleep(2.5)
        path = CM(path="data/driver").install()
        # if sys.platform == "win32":
        #     # shutil.move("chromedriver.exe", "data/driver")
        #     cd = os.path.abspath("data/driver/chromedriver.exe")
        # else:
        #     # shutil.move("chromedriver", "data/driver")
        #     time.sleep(2.5)
        #     #cd = os.path.abspath(path)
        # Patcher(executable_path=cd).patch_exe()
        self.CHROMEDRIVER = path
        Patcher.patch_exe = self.monkey_patch_exe

    @staticmethod
    def gen_random_cdc():
        cdc = random.choices(string.ascii_lowercase, k=26)
        cdc[-6:-4] = map(str.upper, cdc[-6:-4])
        cdc[2] = cdc[0]
        cdc[3] = "_"
        return "".join(cdc).encode()

    def monkey_patch_exe(self):
        linect = 0
        replacement = self.gen_random_cdc()
        replacement = f"  var key = '${replacement.decode()}_';\n".encode()
        with io.open(self.CHROMEDRIVER, "r+b") as fh:
            for line in iter(lambda: fh.readline(), b""):
                if b"var key = " in line:
                    fh.seek(-len(line), 1)
                    fh.write(replacement)
                    linect += 1
            return linect

    def webdriver(self, i=None, proxy=False, headless=False, browser_profile=None, proxy_address=None):
        options = self.options(i=i, proxy=proxy, headless=headless, browser_profile=browser_profile,
proxy_address=proxy_address)
        return webdriver.Chrome(executable_path=self.CHROMEDRIVER, options=options)

    def close(self, driver):
        driver.quit()

    def options(self, i=None, proxy=False, headless=False, browser_profile=None, proxy_address=None):
        chrome_options = Options()
        if proxy_address is not None and len(proxy_address.split(":")) == 2:
            chrome_options.add_argument("--proxy-server="+proxy_address)
        if proxy_address is not None and len(proxy_address.split(":")) == 4:
            i=random.randint(1000, 9999999)
            proxy = Proxy(*proxy_address.split(":"), i); time.sleep(1.5)
            chrome_options.add_extension(proxy)
            # chrome_options.add_argument(f'--load-extension='+proxy)
        if browser_profile is not None:
            os.makedirs("data/browser-profiles") if not os.path.exists("data/browser-profiles") else False
            # user_data_dir = user_data_dir = f'data/browser-profiles/{random.randint(1000, 9999)}-
{"".join(random.choice(string.ascii_letters) for i in range(8))}'
            user_data_dir = f'data/browser-profiles/{browser_profile}'
            os.makedirs(user_data_dir) if not os.path.exists(user_data_dir) else False
            self.user_dir = user_data_dir
            chrome_options.add_argument("--user-data-dir=%s" % user_data_dir)
        if not headless:
            chrome_options.add_extension(os.path.join(os.path.dirname(os.path.abspath(__file__)),
"extensions", "always_active.zip"))
```

```
            chrome_options.add_extension(os.path.join(os.path.dirname(os.path.abspath(__file__)),
"extensions", "fingerprint_defender.zip"))
            chrome_options.add_extension(os.path.join(os.path.dirname(os.path.abspath(__file__)),
"extensions", "spoof_timezone.zip"))
            chrome_options.add_extension(os.path.join(os.path.dirname(os.path.abspath(__file__)),
"extensions", "webrtc_control.zip"))
        chrome_options.add_argument('--mute-audio')
        chrome_options.add_argument("--start-maximized")
        chrome_options.add_experimental_option('prefs', {'intl.accept_languages': 'en,en_US'})
        chrome_options.add_argument('Content-Type="text/html"')
        if headless:
            chrome_options.add_argument("--headless")
        chrome_options.add_argument('chartset=utf-8')
        chrome_options.add_argument("--no-sandbox")
        chrome_options.add_argument("--disable-gpu")
        chrome_options.add_argument("--disable-crash-reporter")
        chrome_options.add_argument("--disable-in-process-stack-traces")
        chrome_options.add_argument("--disable-logging")
        chrome_options.add_argument("--disable-dev-shm-usage")
        chrome_options.add_argument("--log-level=3")
        chrome_options.add_argument("--output=/dev/null")
        if proxy!=False and i!=None:
            chrome_options.add_extension(f"data/extension/proxy_auth_plugin_{i}.zip")
        chrome_options.add_experimental_option("excludeSwitches", ["enable-automation", "enable-logging"])
        chrome_options.add_experimental_option('useAutomationExtension', False)
        chrome_options.add_experimental_option('prefs', {'intl.accept_languages': 'en_US,en'})
        chrome_options.add_argument('--disable-features=UserAgentClientHint')
        webdriver.DesiredCapabilities.CHROME['loggingPrefs'] = {'driver': 'OFF', 'server': 'OFF', 'browser':
'OFF'}
        webdriver.DesiredCapabilities.CHROME['acceptSslCerts']=True
        return chrome_options

def relative_to_assets(relative_path):
    try:
        base_path = sys._MEIPASS
    except Exception:
        base_path = os.path.dirname(__file__)
    return os.path.join(base_path, relative_path)

def external_function(text2_content, entry1_content):#prox,game
    global stop_event
    global stop_sound_loop
    stop_sound_loop = False
    stop_event.set()
    # Process entry1_content
    try:
        if entry1_content.strip():
            entry1_list = entry1_content.split(',')
            if all(element.strip().isdigit() for element in entry1_list):
                entry1_int_list = [int(element.strip()) for element in entry1_list]
            else:
                messagebox.showerror("Error with Game Number(s)", "Error - Game number(s) provided was not in
the correct format of '1,2,3,...' or '1'.")
                stop_event.clear()
                return
        else:
            messagebox.showerror("No Game(s) Number(s)!", "Error - No Game Number was Provided.")
            stop_event.clear()
            return
        # Process text2_content
        if text2_content.strip():
            text2_list = text2_content.splitlines()
        else:
            messagebox.showerror("No Proxy!", "Error - No Proxy was Provided.")
            stop_event.clear()
            return
        thread = SignupThread(target=signup, kwargs={"gameNum": entry1_int_list, "proxyy": text2_list})
        thread.start()
    except Exception as e:
        messagebox.showerror("Fatal Error", "Fatal Error 67: "+str(e))
        stop_event.clear()
        return

def stop_sound():
    global stop_sound_loop
    stop_sound_loop = True
    messagebox.showinfo("Sounds Stopped", "All Sounds Stopped!")
```

```python
def stop_tool():
    global stop_event
    if not stop_event.is_set():
        messagebox.showinfo("Nothing to Stop!", "Tool not running!")
    else:
        stop_event.clear()
        canvas.itemconfig(texti, text="STOPPING THE TOOL.... PLEASE WAIT.....",fill="#FCFCFC")

class SignupThread(threading.Thread):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)

    def run(self):
        global stop_event
        super().run()
        stop_event.clear()

class app:
    def __init__(self, master):
        self.master = master
        self.master.title("Rugby Ticket Getter")
        self.master.resizable(False, False)
        self.master.geometry("862x519")
        self.master.iconbitmap(relative_to_assets('potatowatts.ico'))
        self.master.configure(bg = "#3A7FF6")
        self.page1()

    def save_text(self):
        if not os.path.exists(APP_DATA_FOLDER):
            os.makedirs(APP_DATA_FOLDER)

        # Save text from self.text_2
        file_path_text_2 = os.path.join(APP_DATA_FOLDER, "saved_text_2.txt")
        with open(file_path_text_2, "w") as file:
            file.write(self.text_2.get(1.0, "end-1c"))

        # Save text from self.entry_1
        file_path_entry_1 = os.path.join(APP_DATA_FOLDER, "saved_entry_1.txt")
        with open(file_path_entry_1, "w") as file:
            file.write(self.entry_1.get())

    def load_text(self):
        # Load text for self.text_2
        file_path_text_2 = os.path.join(APP_DATA_FOLDER, "saved_text_2.txt")
        if os.path.exists(file_path_text_2):
            with open(file_path_text_2, "r") as file:
                self.text_2.delete(1.0, "end")
                self.text_2.insert(1.0, file.read())

        # Load text for self.entry_1
        file_path_entry_1 = os.path.join(APP_DATA_FOLDER, "saved_entry_1.txt")
        if os.path.exists(file_path_entry_1):
            with open(file_path_entry_1, "r") as file:
                self.entry_1.delete(0, "end")
                self.entry_1.insert(0, file.read())

    def on_closing(self):
        global stop_event
        stop_event.clear()
        self.save_text()
        self.master.destroy()

    def starti(self):
        if not stop_event.is_set():
            text2_content = self.text_2.get(1.0, "end-1c")
            entry1_content = self.entry_1.get()
            external_function(text2_content, entry1_content)
        else:
            messagebox.showinfo("Already Running!", "Tool is already running!")

    def page1(self):
        for i in self.master.winfo_children():
            i.destroy()
        self.frame1 = Frame(self.master)

        self.canvas = Canvas(
            self.master,
            bg = "#3A7FF6",
            height = 519,
            width = 862,
            bd = 0,
            highlightthickness = 0,
            relief = "ridge"
        )

        self.canvas.place(x = 0, y = 0)
        self.canvas.create_rectangle(
            430.9999999999999,
            0.0,
            861.9999999999999,
            519.0,
            fill="#FCFCFC",
            outline="")

        self.canvas.create_text(
            450.9999999999999,
            238.0,
            anchor="nw",
            text="By Pressing Start you Agree to the Terms & Disclaimer",
            fill="#000000",
            font=("AdventPro Regular", 16 * -1)
        )

        self.button_image_1 = PhotoImage(
            file=relative_to_assets("button_11.png"))
        self.button_1 = Button(
            image=self.button_image_1,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: self.page2(),
            relief="flat"
        )
        self.button_1.place(
            x=556.9999999999999,
            y=272.0,
            width=180.0,
            height=55.0
        )

        self.button_image_2 = PhotoImage(
            file=relative_to_assets("button_12.png"))
        self.button_1 = Button(
            image=self.button_image_2,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: messagebox.showinfo("Disclaimer", """This tool is provided 'as is' without any
warranties or guarantees, either express or implied. The developer disclaims all liability for any direct,
indirect, incidental, special, punitive, consequential, or other damages arising from the use of this tool. By
using this tool, you agree to comply with Rugby World Cup's terms of service, any applicable laws, and assume
full responsibility for your actions. You also acknowledge that the developer is not affiliated with, endorsed
by, or in any way associated with Rugby World Cup."""),
            relief="flat"
        )
        self.button_1.place(
            x=556.9999999999999,
            y=372.0,
            width=180.0,
            height=55.0
        )

        self.canvas.create_text(
            39.999999999999886,
            127.0,
            anchor="nw",
            text="Rugby Ticket Getter",
            fill="#FCFCFC",
            font=("Roboto Bold", 24 * -1)
        )

        self.canvas.create_text(
            519.9999999999999,
            203.0,
            anchor="nw",
            text="Press start when ready.",
            fill="#505485",
            font=("Roboto Bold", 24 * -1)
        )

        self.canvas.create_rectangle(
            39.999999999999886,
            160.0,
            99.99999999999989,
            165.0,
            fill="#FCFCFC",
            outline="")

        self.canvas.create_text(
            39.999999999999886,
            175.0,
            anchor="nw",
            text="Tool made by PotatoWatts",
            fill="#FCFCFC",
            font=("AdventPro Regular", 24 * -1)
        )

        self.canvas.create_text(
            30.999999999999886,
            471.0,
            anchor="nw",
            text="I am not liable for anything that results from the use",
            fill="#FCFCFC",
            font=("AdventPro Regular", 16 * -1)
        )

        self.canvas.create_text(
            30.999999999999886,
            490.0,
            anchor="nw",
            text="of this tool. Use at your own risk.        Version 1.1.0",
            fill="#FCFCFC",
            font=("AdventPro Regular", 16 * -1)
        )

        self.canvas.create_text(
            40.999999999999886,
            203.0,
            anchor="nw",
            text="@PotatoWatts on Telegram",
            fill="#FCFCFC",
            font=("AdventPro Regular", 14 * -1)
        )
        root.mainloop()

    def page2(self):
        global canvas
        global texti
        self.master.protocol("WM_DELETE_WINDOW", self.on_closing)
        for i in self.master.winfo_children():
            i.destroy()

        self.frame1 = Frame(self.master)

        canvas = Canvas(
            self.master,
            bg = "#3A7FF6",
            height = 519,
            width = 862,
            bd = 0,
            highlightthickness = 0,
            relief = "ridge"
        )

        canvas.place(x = 0, y = 0)
        canvas.create_rectangle(
            492.9999999999999,
            7.105427357601002e-15,
            865.9999999999999,
            531.0,
            fill="#FCFCFC",
            outline="")

        entry_image_1 = PhotoImage(
            file=relative_to_assets("entry_1.png"))
        entry_bg_1 = canvas.create_image(
            327.9999999999999,
            21.500000000000007,
            image=entry_image_1
        )
        self.entry_1 = Entry(
            bd=0,
            bg="#EEEEEE",
            fg="#000716",
            highlightthickness=0
        )
        self.entry_1.place(
            x=175.9999999999999,
            y=8.000000000000007,
            width=304.0,
            height=25.0
        )

        entry_image_2 = PhotoImage(
            file=relative_to_assets("entry_2.png"))
        entry_bg_2 = canvas.create_image(
            326.4999999999999,
            77.0,
            image=entry_image_2
        )
        # Create the Text widget with vertical and horizontal scrollbars
        self.text_2 = Text(self.master, wrap="none", bd=0, bg="#EEEEEE", fg="#000716", highlightthickness=0)
        self.text_2.place(x=172.9999999999999, y=46.00000000000001, width=287.0, height=60.0)  # Adjust the
width to 287.0

        # Create the vertical scrollbar
        v_scrollbar = Scrollbar(self.master, orient="vertical", command=self.text_2.yview)
        v_scrollbar.place(x=460, y=46.00000000000001, height=60.0)  # Adjust the x position to 460

        # Create the horizontal scrollbar
        h_scrollbar = Scrollbar(self.master, orient="horizontal", command=self.text_2.xview)
        h_scrollbar.place(x=172.9999999999999, y=106, width=287.0)  # Adjust the y position to 106

        # Configure the text widget to use the scrollbars
        self.text_2.config(yscrollcommand=v_scrollbar.set, xscrollcommand=h_scrollbar.set)


        canvas.create_text(
            522.9999999999999,
            460.0,
            anchor="nw",
            text="Looking for Extra Features? Extra Options?",
            fill="#000000",
            font=("AdventPro Regular", 16 * -1)
        )

        canvas.create_text(
            522.9999999999999,
            438.0,
            anchor="nw",
            text="Looking for New Automation Services/Tools?",
            fill="#000000",
            font=("AdventPro Regular", 16 * -1)
        )

        canvas.create_text(
            506.9999999999999,
            10.000000000000007,
            anchor="nw",
            text="If Tickets are Found a Sound will Ring",
            fill="#505485",
            font=("Roboto Bold", 20 * -1)
        )

        canvas.create_text(
            522.9999999999999,
            46.00000000000001,
            anchor="nw",
            text="Until you Press the Below Button:",
            fill="#505485",
            font=("Roboto Bold", 20 * -1)
        )

        button_image_1 = PhotoImage(
            file=relative_to_assets("button_1.png"))
        button_1 = Button(
            image=button_image_1,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: messagebox.showinfo(title='AUTOMATION SERVICES - PotatoWatts', message='I am 
available on several places, feel free to Contact Me via Telegram: @ PotatoWatts'),
            relief="flat"
        )
        button_1.place(
            x=583.9999999999999,
            y=483.0,
            width=183.0,
            height=27.0
        )

        button_image_2 = PhotoImage(
            file=relative_to_assets("button_2.png"))
        button_2 = Button(
            image=button_image_2,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: self.starti(),
            relief="flat"
        )
        button_2.place(
            x=150.9999999999999,
            y=217.0,
            width=180.0,
            height=55.0
        )

        button_image_3 = PhotoImage(
            file=relative_to_assets("button_3.png"))
        button_3 = Button(
            image=button_image_3,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: stop_tool(),
            relief="flat"
        )
        button_3.place(
            x=59.999999999999886,
            y=389.0,
            width=120.0,
            height=55.0
        )

        button_image_4 = PhotoImage(
            file=relative_to_assets("button_4.png"))
        button_4 = Button(
            image=button_image_4,
            borderwidth=0,
            highlightthickness=0,
            command=lambda: stop_sound(),
            relief="flat"
        )
        button_4.place(
            x=606.9999999999999,
            y=85.0,
            width=145.0,
            height=45.0
        )

        canvas.create_text(
            29.99999999999989,
            171.0,
            anchor="nw",
            text="START RUGBY TICKET SEARCH",
            fill="#FCFCFC",
            font=("Roboto Bold", 28 * -1)
        )

        canvas.create_text(
            6.999999999999886,
            10.000000000000007,
            anchor="nw",
            text="Games,Wait Time:",
            fill="#FCFCFC",
            font=("Roboto Bold", 18 * -1)
        )

        canvas.create_text(
            20.999999999999886,
            64.0,
            anchor="nw",
            text="Proxies to Use:",
            fill="#FCFCFC",
            font=("Roboto Bold", 20 * -1)
        )

        canvas.create_text(
            102.99999999999989,
            301.0,
            anchor="nw",
            text="Current Action Being Performed:",
            fill="#FCFCFC",
            font=("Roboto Bold", 20 * -1)
        )

        canvas.create_text(
            651.9999999999999,
            270.0,
            anchor="nw",
            text="pW Software",
            fill="#505485",
            font=("Roboto Bold", 24 * -1)
        )

        canvas.create_rectangle(
            504.9999999999999,
            161.0,
            853.9999999999999,
            167.0,
            fill="#3A7FF6",
            outline="")

        canvas.create_rectangle(
            504.9999999999999,
            428.0,
            853.9999999999999,
            433.0,
            fill="#3A7FF6",
            outline="")

        canvas.create_rectangle(
            12.999999999999886,
            452.0,
            479.9999999999999,
            457.0,
            fill="#FCFCFC",
            outline="")

        canvas.create_rectangle(
            12.999999999999886,
            377.0,
            479.9999999999999,
            382.0,
            fill="#FCFCFC",
            outline="")

        canvas.create_rectangle(
            12.999999999999886,
            138.0,
            479.9999999999999,
            143.0,
            fill="#FCFCFC",
            outline="")

        canvas.create_text(
            30.999999999999886,
            471.0,
            anchor="nw",
            text="I am not liable for anything that results from the use",
            fill="#FCFCFC",
            font=("AdventPro Regular", 16 * -1)
        )

        texti = canvas.create_text(
            49.999999999999886,
            335.0,
            anchor="nw",
            text="WAITING ON START...",
            fill="#FCFCFC",
            font=("AdventPro Regular", 18 * -1)
        )

        canvas.create_text(
            198.9999999999999,
            392.0,
            anchor="nw",
            text="Press this button to force stop the tool.",
            fill="#FCFCFC",
            font=("AdventPro Regular", 16 * -1)
        )

        canvas.create_text(
            198.9999999999999,
            411.0,
            anchor="nw",
            text="Sometimes it takes a few minutes to stop, be ",
            fill="#FCFCFC",
            font=("AdventPro Regular", 14 * -1)
        )

        canvas.create_text(
            198.9999999999999,
            425.0,
            anchor="nw",
            text="patient or you might get corrupted data files.",
            fill="#FCFCFC",
            font=("AdventPro Regular", 14 * -1)
        )

        canvas.create_text(
            30.999999999999886,
            490.0,
            anchor="nw",
            text="of this tool. Use at your own risk.                                  Version 1.1.0",
            fill="#FCFCFC",
            font=("AdventPro Regular", 16 * -1)
        )

        image_image_1 = PhotoImage(
            file=relative_to_assets("image_1.png"))
        image_1 = canvas.create_image(
            601.9999999999999,
            284.0,
            image=image_image_1
        )
        self.load_text()
        root.mainloop()

os.makedirs("data") if not os.path.exists("data") else False
os.makedirs("data/driver") if not os.path.exists("data/driver") else False
os.makedirs("data/browser-profiles") if not os.path.exists("data/browser-profiles") else False
APP_DATA_FOLDER = os.path.join(Path.home(), "AppData", "Local", "RugbyTicketGetterData")


root = Tk()
app(root)
"""
game=["1","2"]
proxyDado="86.104.165.2:12323:14a0e87864ba8:0ff20683b3"#rotating.proxyempire.io:9000:ccGHbESP6IoGbkGA:wifi;gb;;;

signup(gameNum=game,proxy=proxyDado)
"""
```