Task : 4 Author:- Sagar Bhoi

Task4: Digital Marketing Automation

Input Value Type text- Item name-Data Science Basic Training Program for Everyone(Age=10 to 60) Input Value Type link : https://yoshops.com/products/data-science-basic-training-program-for-everyone-age-10-t0-60

Output:

1.Create Banner(Jpg,Png),Tag Line for topic(notepad),Topic description file(notepad),keyword file(notepad),

2.Create 20sec 🎥 shorts videos(MP4) on Data Science Basic Training Program for Everyone

3.Write a blog on Data Science Basic Training Program for Everyone.

In [1]:

```
!pip install bs4
!pip install python-docx
!pip install moviepy
!pip install opencv-python
```

```
Requirement already satisfied: bs4 in e:\python\lib\site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in e:\python\lib\site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in e:\python\lib\site-packages (from beautifulsoup4->bs4) (2.3.2.post1)
Requirement already satisfied: python-docx in e:\python\lib\site-packages (0.8.11)
Requirement already satisfied: lxml>=2.3.2 in e:\python\lib\site-packages (from python-docx) (4.9.1)
Requirement already satisfied: moviepy in e:\python\lib\site-packages (1.0.3)
Requirement already satisfied: requests<3.0,>=2.8.1 in e:\python\lib\site-packages (from moviepy) (2.28.1)
Requirement already satisfied: numpy in e:\python\lib\site-packages (from moviepy) (1.23.5)
Requirement already satisfied: proglog<=1.0.0 in e:\python\lib\site-packages (from moviepy) (0.1.10)
Requirement already satisfied: imageio-ffmpeg>=0.2.0 in e:\python\lib\site-packages (from moviepy) (0.4.8)
Requirement already satisfied: imageio<3.0,>=2.5 in e:\python\lib\site-packages (from moviepy) (2.19.3)
Requirement already satisfied: tqdm<5.0,>=4.11.2 in e:\python\lib\site-packages (from moviepy) (4.64.1)
Requirement already satisfied: decorator<5.0,>=4.0.2 in e:\python\lib\site-packages (from moviepy) (4.4.2)
Requirement already satisfied: pillow>=8.3.2 in e:\python\lib\site-packages (from imageio<3.0,>=2.5->moviepy) (9.2.0)
Requirement already satisfied: idna<4,>=2.5 in e:\python\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in e:\python\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (2022.9.24)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in e:\python\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (1.26.13)
Requirement already satisfied: charset-normalizer<3,>=2 in e:\python\lib\site-packages (from requests<3.0,>=2.8.1->moviepy) (2.0.4)
Requirement already satisfied: colorama in e:\python\lib\site-packages (from tqdm<5.0,>=4.11.2->moviepy) (0.4.5)
Requirement already satisfied: opencv-python in e:\python\lib\site-packages (4.7.0.72)
Requirement already satisfied: numpy>=1.17.3 in e:\python\lib\site-packages (from opencv-python) (1.23.5)
```

In [2]:

```
import requests
from bs4 import BeautifulSoup as soup
from urllib.request import urlopen as uReq
import numpy as np
import pandas as pd
from PIL import Image,ImageFont, ImageDraw
from moviepy.editor import AudioFileClip, ImageClip
import cv2
import os
from os.path import isfile, join
import docx
from time import sleep
#The sleep() function suspends execution of the current thread for a given number of seconds
from random import randint
#The randint Python function is a built-in method that lets you generate random integers using the random module
```

In [3]:

```
print("Enter 1 to - Create a banner of yoshops sale \n")
print("Enter 2 to - Create a video \n")
print("Enter 3 to - Create a blog in word document  \n")

# Taking input from the user as integer
num = int(input("Enter a number: "))

if(num==1):
    # Extracting Images From Yoshops Site
    video_url = "https://yoshops.com/products?keywords=data+science"
    video_uClient = uReq(video_url)
    video_page_html = video_uClient.read()
    video_uClient.close()
    video_page_soup = soup(video_page_html,"html.parser")

    video_containers = video_page_soup.findAll("div", {"class":"col-sm-3 col-xs-6"})
    video_container =  video_containers[0]
    image1_link_video = video_container.img['src']
    im1 = Image.open(requests.get(image1_link_video, stream=True).raw)
    im1.save("task4_img1_video.webp")

    video_container =  video_containers[1]
    image2_link_video = video_container.img['src']
    im2 = Image.open(requests.get(image2_link_video, stream=True).raw)
    im2.save("task4_img2_video.webp")

elif(num==2):
    #Extacting images via webscraping
    width = 1280
    height = 720
    channel = 3

    fps = 1
    sec = 20

    fourcc = cv2.VideoWriter_fourcc(*'MP42')

    video = cv2.VideoWriter('image_to_video.avi', fourcc, float(fps), (width, height))

    directory = r'/Users/Sagar/Yoshops Data Science Intern/Task_4_week_4/'

    img_name_list = os.listdir(directory)

    for frame_count in range(fps*sec):

        img_name = np.random.choice(img_name_list)
        img_path = os.path.join(directory, img_name)
        img = cv2.imread(img_path)
        img_resize = cv2.resize(img, (width, height))

        video.write(img_resize)

    video.release()

elif(num==3):
    url = "https://yoshops.com/products/hammer-sting-lite-in-ear-wireless-bluetooth-neckband-earphones-black"
    uClient = uReq(url)
    page_html = uClient.read()
    uClient.close()
    page_soup = soup(page_html,"html.parser")
    #Extracting image of the product
    containers1 = page_soup.findAll("div", {"class":"col-sm-4 single-product-img-col"})
    container1 =  containers1[0]
    image_link = container1.div.img['src']
    im = Image.open(requests.get(image_link, stream=True).raw)
    im.save("task4_img.jpg")
    heading = container1.div.img['alt']
    containers2 = page_soup.findAll("div", {"class":"col-sm-5"})
    container2 =  containers2[0]
    heading = container2.findAll("h1",{"class":"single-product-title"})
    available = container2.findAll("font",{"color":"#000000"})
    containers3 = page_soup.findAll("div", {"class":"col-sm-3"})
    container3 =  containers3[0]
    price = container3.findAll("span",{"id":"regular-price"})
    old_price=price[0].text
```

```
dis_price = container3.findAll("span",{"id":"sale-price"})
new_price=dis_price[0].text
shipping = container3.div.div.div.div['class']

mydoc = docx.Document()

mydoc.add_heading(heading[0].text, 0)

mydoc.add_picture("task4_img.jpg", width=docx.shared.Inches(4), height=docx.shared.Inches(5))

mydoc.add_heading("Available Feature", 1)
mydoc.add_paragraph(available[0].text)
mydoc.add_paragraph(available[1].text)
mydoc.add_paragraph(available[2].text)
mydoc.add_paragraph(available[3].text)

mydoc.add_heading(available[4].text, 1)
mydoc.add_paragraph(available[5].text)
mydoc.add_paragraph(available[6].text)
mydoc.add_paragraph(available[7].text)
mydoc.add_paragraph(available[8].text)
mydoc.add_paragraph(available[9].text)
mydoc.add_paragraph(available[10].text)
mydoc.add_paragraph(available[11].text)
mydoc.add_paragraph(available[12].text)
mydoc.add_paragraph(available[13].text)
mydoc.add_paragraph(available[14].text)
mydoc.add_paragraph(available[15].text)
mydoc.add_paragraph(available[16].text)
mydoc.add_paragraph(available[17].text)
mydoc.add_paragraph(available[18].text)
mydoc.add_paragraph(available[19].text)
mydoc.add_paragraph(available[20].text)
mydoc.add_paragraph(available[21].text)
mydoc.add_paragraph(available[22].text)
mydoc.add_paragraph(available[23].text)
mydoc.add_paragraph(available[24].text)
mydoc.add_paragraph(available[25].text)
mydoc.add_paragraph(available[26].text)

mydoc.add_heading(available[27].text, 1)
mydoc.add_paragraph(available[28].text)
mydoc.add_paragraph(available[29].text)
mydoc.add_paragraph(available[30].text)
mydoc.add_paragraph(available[31].text)
mydoc.add_paragraph(available[32].text)
```

```
    mydoc.save("my_written_file.docx")
```

```
Enter 1 to - Create a banner of yoshops sale

Enter 2 to - Create a video

Enter 3 to - Create a blog in word document

Enter a number: 3
```

In [ ]:

```

```