To achieve the tasks outlined, you'll typically need to scrape video data from a platform like YouTube or similar. However, such scraping must comply with the website's terms of service. You can use Python libraries like BeautifulSoup and requests for web scraping, or automation tools like selenium for dynamic content.

Here’s a complete implementation:

**Prerequisites**

1. Install necessary libraries:
2. pip install requests beautifulsoup4 pandas
3. Make sure scraping complies with the platform's policy.

**Python Code Implementation**

import requests

from bs4 import BeautifulSoup

import pandas as pd

# Function to extract video details

def scrape\_video\_data(url):

response = requests.get(url)

if response.status\_code != 200:

print("Failed to retrieve webpage.")

return

soup = BeautifulSoup(response.text, 'html.parser')

# Extract video elements

video\_elements = soup.select('ytd-video-renderer')[:5]

if not video\_elements:

print("No videos found.")

return

data = {

"Video URL": [],

"Thumbnail URL": [],

"Title": [],

"Views": [],

"Posted Time": []

}

for video in video\_elements:

# Extracting video URL

link\_tag = video.find('a', id='thumbnail')

video\_url = "https://www.youtube.com" + link\_tag['href'] if link\_tag else "N/A"

data["Video URL"].append(video\_url)

# Extracting thumbnail URL

thumbnail\_url = link\_tag.find('img')['src'] if link\_tag else "N/A"

data["Thumbnail URL"].append(thumbnail\_url)

# Extracting title

title\_tag = video.find('a', id='video-title')

title = title\_tag.text.strip() if title\_tag else "N/A"

data["Title"].append(title)

# Extracting views

views\_tag = video.find('span', class\_='view-count')

views = views\_tag.text.strip() if views\_tag else "N/A"

data["Views"].append(views)

# Extracting posted time

time\_tag = video.find('span', class\_='published-time-text')

posted\_time = time\_tag.text.strip() if time\_tag else "N/A"

data["Posted Time"].append(posted\_time)

# Save to CSV

df = pd.DataFrame(data)

df.to\_csv("video\_data.csv", index=False)

print("Data saved to video\_data.csv")

# Replace this URL with the webpage you want to scrape

scrape\_video\_data("https://www.youtube.com/results?search\_query=python+programming")

**Explanation of Code**

1. **Video URL**: Extracted from the <a> tag's href attribute under the id="thumbnail" tag.
2. **Thumbnail URL**: Retrieved from the <img> tag inside the thumbnail <a> tag.
3. **Title**: Extracted from the <a> tag with id="video-title".
4. **Views**: Scraped from the <span> tag with the class view-count.
5. **Posted Time**: Extracted from the <span> tag with the class published-time-text.
6. **CSV Export**: Data is saved using the Pandas DataFrame to ensure a structured format.

**Notes**

* Dynamic sites may require tools like selenium for accurate scraping.
* Always check and adhere to the terms of service of the website.