**2.Write a Python function that takes a list of URLs, attempts to download their content, and retries up to 3 times if an error occurs. Use appropriate error handling to manage different types of exceptions.**

import requests

from time import sleep

def download\_content(urls):

results = {}

for url in urls:

attempts = 3

while attempts > 0:

try:

response = requests.get(url, timeout=5)

response.raise\_for\_status()

results[url] = response.text

print(f"Successfully downloaded content from {url}")

break

except requests.exceptions.HTTPError as http\_err:

print(f"HTTP error occurred for {url}: {http\_err}")

except requests.exceptions.ConnectionError as conn\_err:

print(f"Connection error occurred for {url}: {conn\_err}")

except requests.exceptions.Timeout as timeout\_err:

print(f"Timeout error occurred for {url}: {timeout\_err}")

except requests.exceptions.RequestException as req\_err:

print(f"Error occurred for {url}: {req\_err}")

attempts -= 1

if attempts > 0:

print(f"Retrying {url} ({3 - attempts}/3)...")

sleep(2) # Wait for 2 seconds before retrying

else:

print(f"Failed to download content from {url} after 3 attempts")

return results

# Example usage

urls = [

'https://www.google.com',

'https://www.facebook.com', # This will fail

'https://google.us/404', # This will return a server error

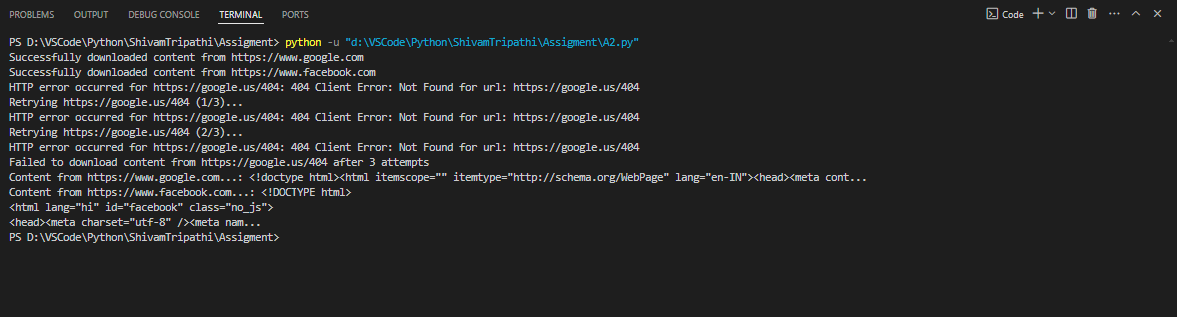
]

content = download\_content(urls)

for url, data in content.items():

print(f"Content from {url[:30]}...: {data[:100]}...") # Print first 100 chars of content

**OUTPUT:**

****