PROGRAMMING ASSESSMENT

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 os

import requests

from urllib.parse import urljoin, urlparse

from bs4 import BeautifulSoup

def download\_file(url, folder):

# Ensure the directory exists

if not os.path.isdir(folder):

os.makedirs(folder)

local\_filename = os.path.join(folder, url.split('/')[-1])

# Download the file and write it to disk

with requests.get(url, stream=True) as r:

r.raise\_for\_status()

with open(local\_filename, 'wb') as f:

for chunk in r.iter\_content(chunk\_size=8192):

f.write(chunk)

return local\_filename

def download\_all\_files\_from\_url(page\_url, folder="downloads"):

response = requests.get(page\_url)

response.raise\_for\_status() # Ensure we notice bad responses

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

for link in soup.find\_all('a', href=True):

file\_url = link['href']

# Ensure the link is a valid URL

if not urlparse(file\_url).scheme:

file\_url = urljoin(page\_url, file\_url)

# Download the file

try:

print(f"Downloading {file\_url}")

download\_file(file\_url, folder)

except Exception as e:

print(f"Failed to download {file\_url}: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

page\_url = input("Enter the URL : ")

download\_all\_files\_from\_url(page\_url)

