ASSIGNMENT-18.1

NAME: VITTAM VENKATESH.

HALLTICKET NUMBER: 2403A52419

BATCH: 15

SUBJECT: AI CODING.

TASK-1

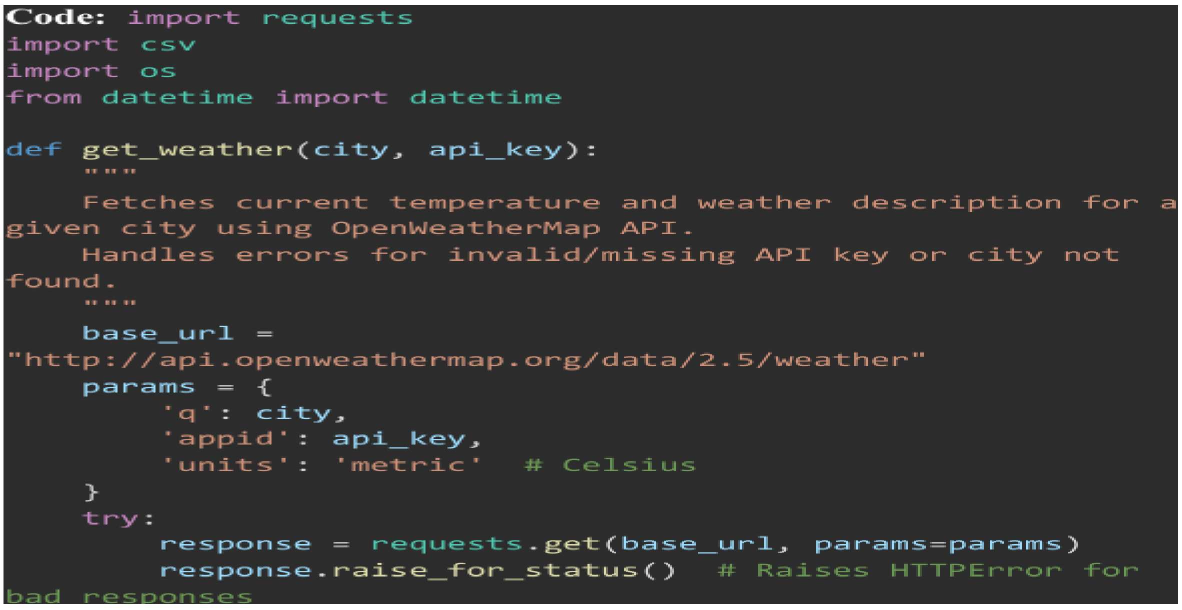
QUESTION 1:

Weather Forecasting API A travel company wants to show real-time weather updates for its customers. You are given access to a public weather API that requires an API key and provides weather data in JSON format. • Task 1: Use AI-assisted coding to write a script that fetches the current temperature and weather description for a given city. The script should handle errors if the API key is invalid or missing. • Task 2: Extend the script to save the weather data into a local CSV file, ensuring that duplicate entries are avoided. Implement error handling for file I/O exceptions.

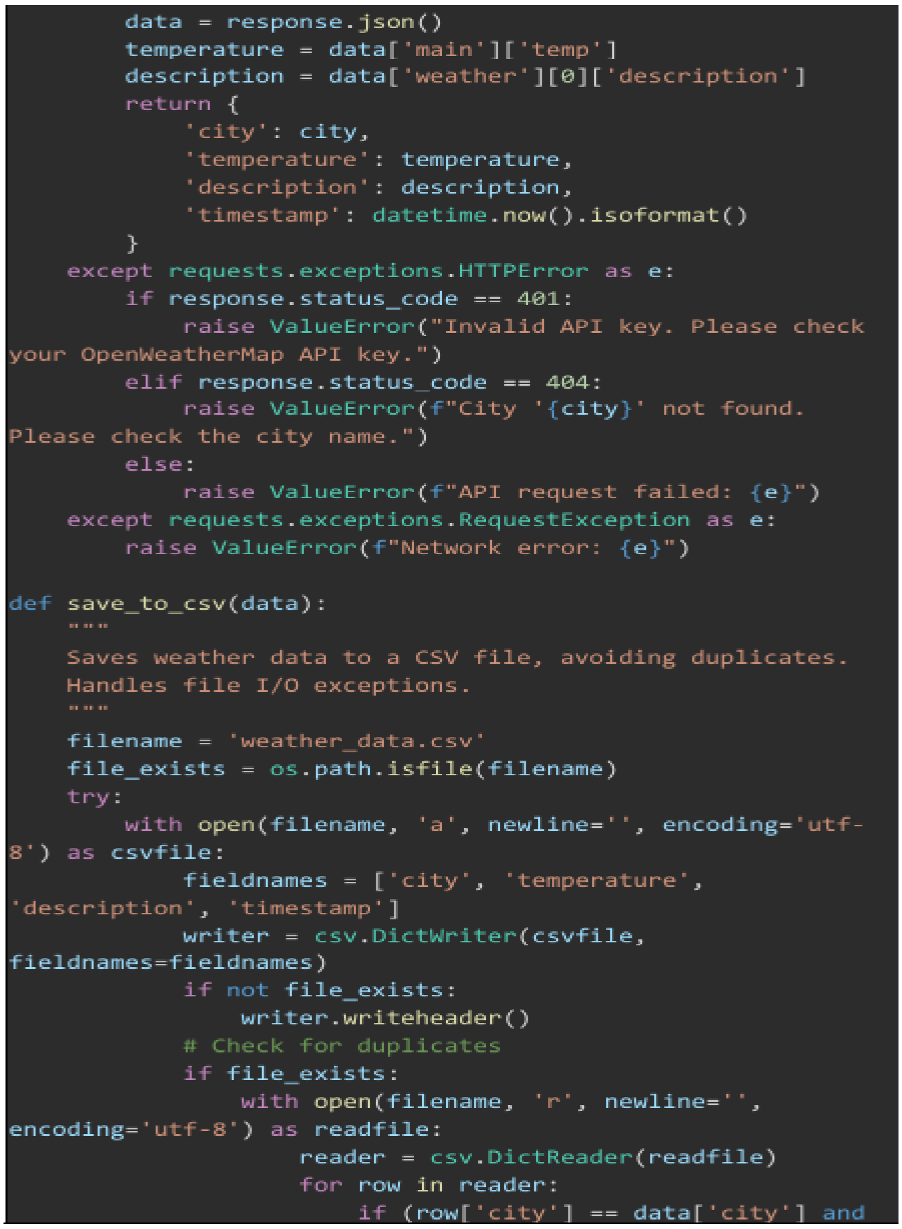
PROMPT:

Write a Python script that fetches current weather (temperature + description) for a city using a public API with full error handling, and then saves the data into a CSV file without duplicates. Include handling for invalid API keys, network issues, and file I/O exceptions.

CODE:

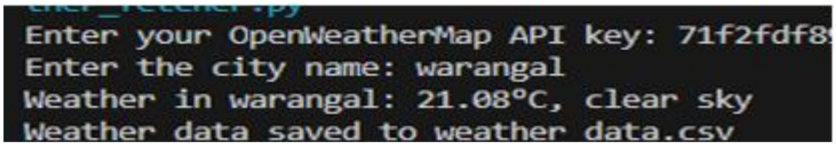


```
Code: import requests
import csv
import os
from datetime import datetime

def get_weather(city, api_key):
    """
    Fetches current temperature and weather description for a
given city using OpenWeatherMap API.
    Handles errors for invalid/missing API key or city not
found.
    """
    base_url =
"http://api.openweathermap.org/data/2.5/weather"
    params = {
        'q': city,
        'appid': api_key,
        'units': 'metric'   # Celsius
    }
    try:
        response = requests.get(base_url, params=params)
        response.raise_for_status()   # Raises HTTPError for
bad responses
```

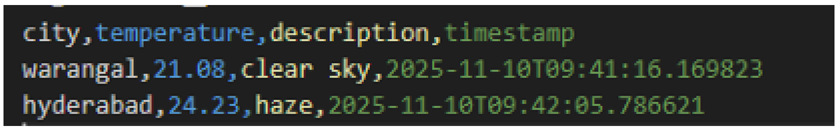


```
        data = response.json()
        temperature = data['main']['temp']
        description = data['weather'][0]['description']
        return {
            'city': city,
            'temperature': temperature,
            'description': description,
            'timestamp': datetime.now().isoformat()
        }
    except requests.exceptions.HTTPError as e:
        if response.status_code == 401:
            raise ValueError("Invalid API key. Please check
your OpenWeatherMap API key.")
        elif response.status_code == 404:
            raise ValueError(f"City '{city}' not found.
Please check the city name.")
        else:
            raise ValueError(f"API request failed: {e}")
    except requests.exceptions.RequestException as e:
        raise ValueError(f"Network error: {e}")

def save_to_csv(data):
    """
    Saves weather data to a CSV file, avoiding duplicates.
    Handles file I/O exceptions.
    """
    filename = 'weather_data.csv'
    file_exists = os.path.isfile(filename)
    try:
        with open(filename, 'a', newline='', encoding='utf-
8') as csvfile:
            fieldnames = ['city', 'temperature',
'description', 'timestamp']
            writer = csv.DictWriter(csvfile,
fieldnames=fieldnames)
            if not file_exists:
                writer.writeheader()
            # Check for duplicates
            if file_exists:
                with open(filename, 'r', newline='',
encoding='utf-8') as readfile:
                    reader = csv.DictReader(readfile)
                    for row in reader:
                        if (row['city'] == data['city'] and
```

OUTPUT:



```
Enter your OpenWeatherMap API key: 71f2fdf8
Enter the city name: warangal
Weather in warangal: 21.08°C, clear sky
Weather data saved to weather_data.csv
```



```
city,temperature,description,timestamp
warangal,21.08,clear sky,2025-11-10T09:41:16.169823
hyderabad,24.23,haze,2025-11-10T09:42:05.786621
```

OBSERVATION:

The script successfully retrieves real-time weather data for a given city using an API and handles errors such as invalid keys or network failures. It also stores the fetched data in a CSV file while preventing duplicate entries and managing file I/O exceptions effectively

TASK-2

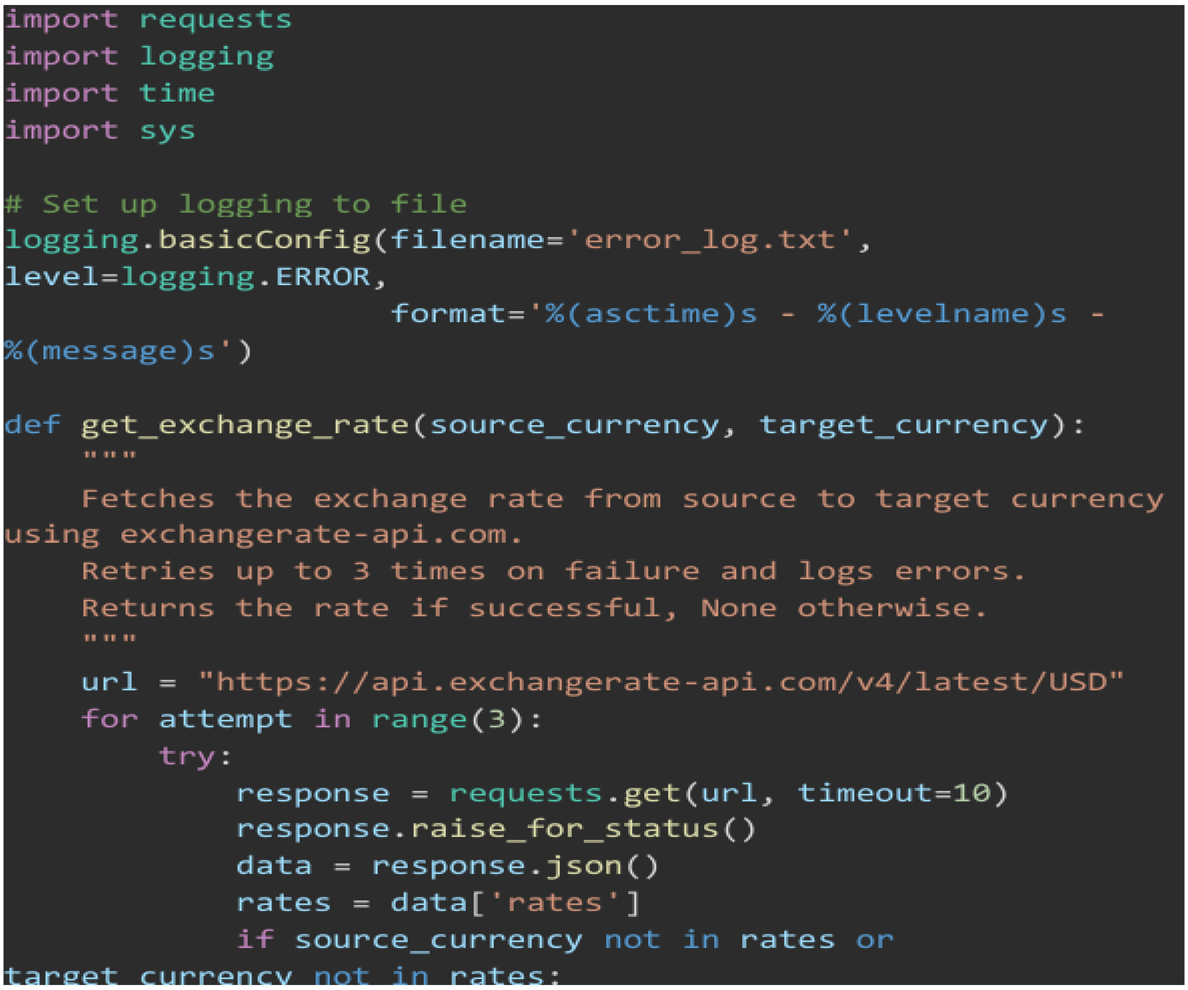
QUESTION 2:

Currency Exchange Rate API A financial startup needs a tool to convert amounts between currencies using an exchange rate API. However, the API occasionally fails due to server downtime. • Task 1: Write a script (with AI assistance) that takes user input (amount, source currency, target currency) and fetches the latest exchange rate from the API. Include errors in handling invalid currency codes. • Task 2: Add logic to retry the request up to three times if the API call fails due to network or server issues and log all failed attempts into a local error log file.

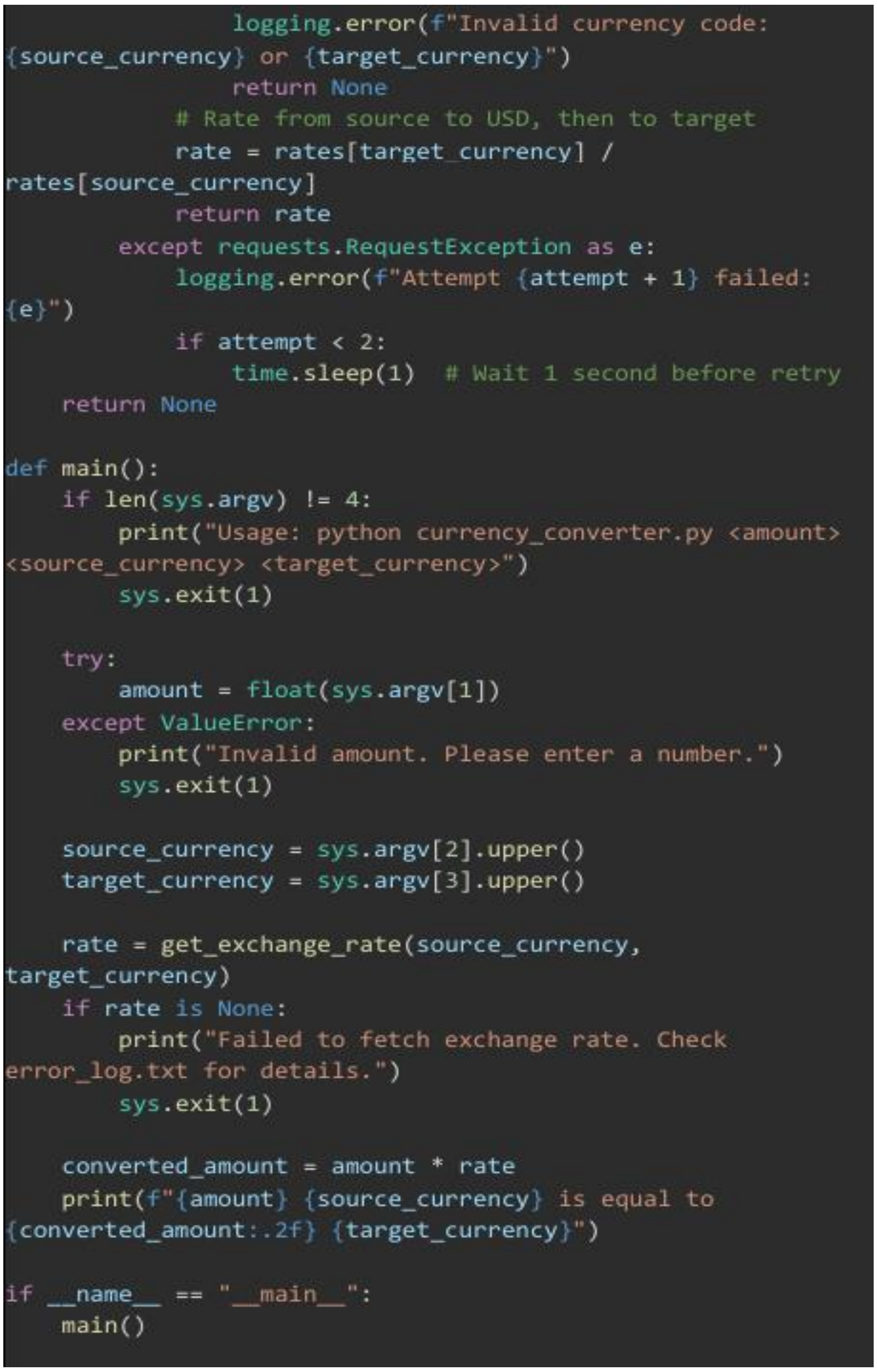
PROMPT:

Write a Python script that takes an amount, source currency, and target currency, then fetches the latest exchange rate from an API with error handling for invalid currency codes. Add retry logic (up to three attempts) for API failures and log all failed attempts into a local error log file

CODE:

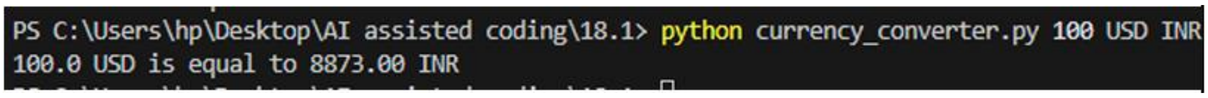


```
import requests
import logging
import time
import sys

# Set up logging to file
logging.basicConfig(filename='error_log.txt',
level=logging.ERROR,
                    format='%(asctime)s - %(levelname)s -
%(message)s')

def get_exchange_rate(source_currency, target_currency):
    """
    Fetches the exchange rate from source to target currency
using exchangerate-api.com.
    Retries up to 3 times on failure and logs errors.
    Returns the rate if successful, None otherwise.
    """
    url = "https://api.exchangerate-api.com/v4/latest/USD"
    for attempt in range(3):
        try:
            response = requests.get(url, timeout=10)
            response.raise_for_status()
            data = response.json()
            rates = data['rates']
            if source_currency not in rates or
target_currency not in rates:
```



```
                logging.error(f"Invalid currency code:
{source_currency} or {target_currency}")
                return None
            # Rate from source to USD, then to target
            rate = rates[target_currency] /
rates[source_currency]
            return rate
        except requests.RequestException as e:
            logging.error(f"Attempt {attempt + 1} failed:
{e}")
            if attempt < 2:
                time.sleep(1)  # Wait 1 second before retry
    return None

def main():
    if len(sys.argv) != 4:
        print("Usage: python currency_converter.py <amount>
<source_currency> <target_currency>")
        sys.exit(1)

    try:
        amount = float(sys.argv[1])
    except ValueError:
        print("Invalid amount. Please enter a number.")
        sys.exit(1)

    source_currency = sys.argv[2].upper()
    target_currency = sys.argv[3].upper()

    rate = get_exchange_rate(source_currency,
target_currency)
    if rate is None:
        print("Failed to fetch exchange rate. Check
error_log.txt for details.")
        sys.exit(1)

    converted_amount = amount * rate
    print(f"{amount} {source_currency} is equal to
{converted_amount:.2f} {target_currency}")

if __name__ == "__main__":
    main()
```

OUTPUT:



```
PS C:\Users\hp\Desktop\AI assisted coding\18.1> python currency_converter.py 100 USD INR
100.0 USD is equal to 8873.00 INR
```

OBSERVATION:

The script accurately converts currency values using real-time exchange rates while handling invalid inputs gracefully. It includes a retry mechanism for API failures and records all unsuccessful attempts in an error log, ensuring reliability even during server issues

TASK-3

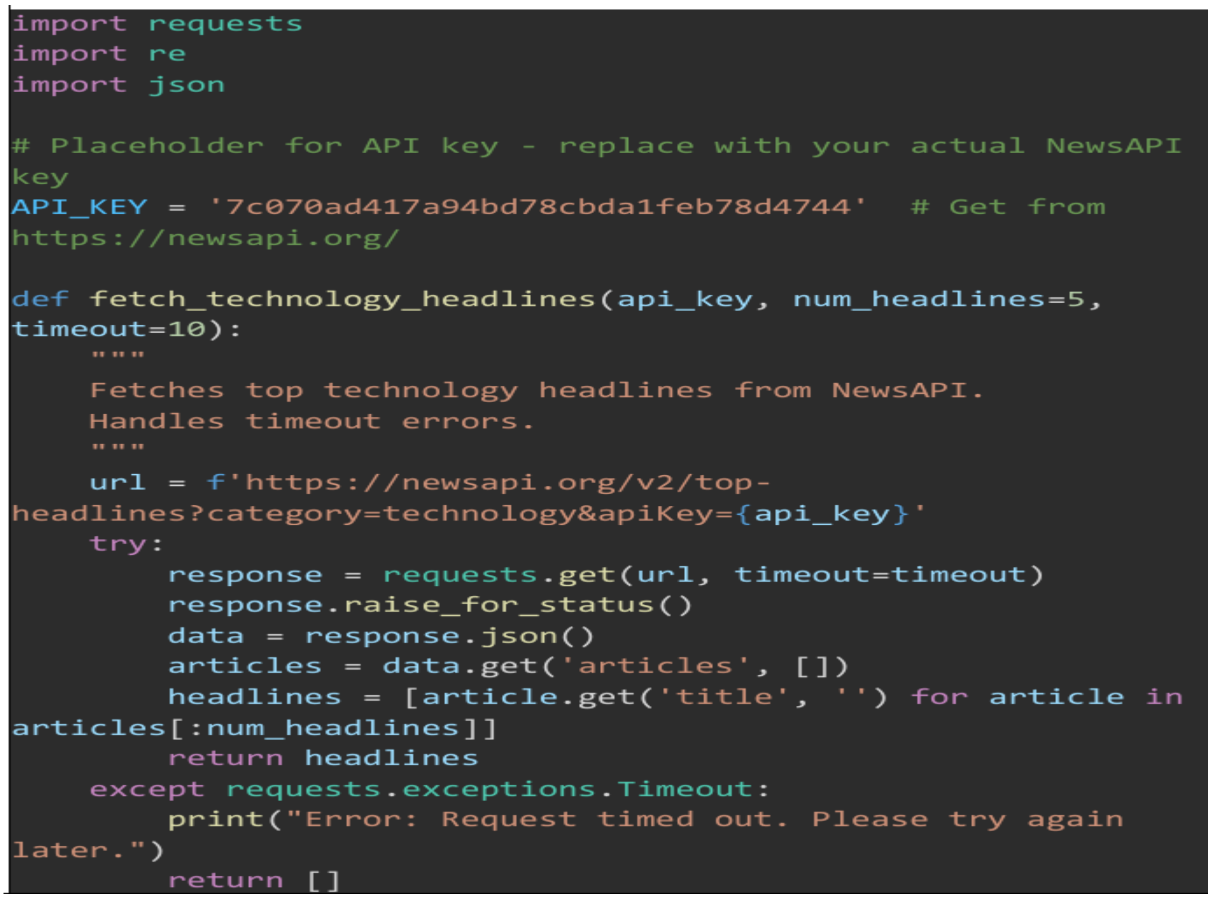
QUESTION 3:

: News Headlines API A news aggregator wants to display the latest technology news headlines using a news API. Sometimes, the API responds slowly or returns incomplete data. • Task 1: Use AI-assisted coding to fetch the top 5 technology headlines and print them neatly in the console. Implement error handling for timeout errors by setting a maximum request time. • Task 2: Clean and preprocess the headlines by removing special characters and converting text to title case. Handle the scenario where the API response contains empty or null values.

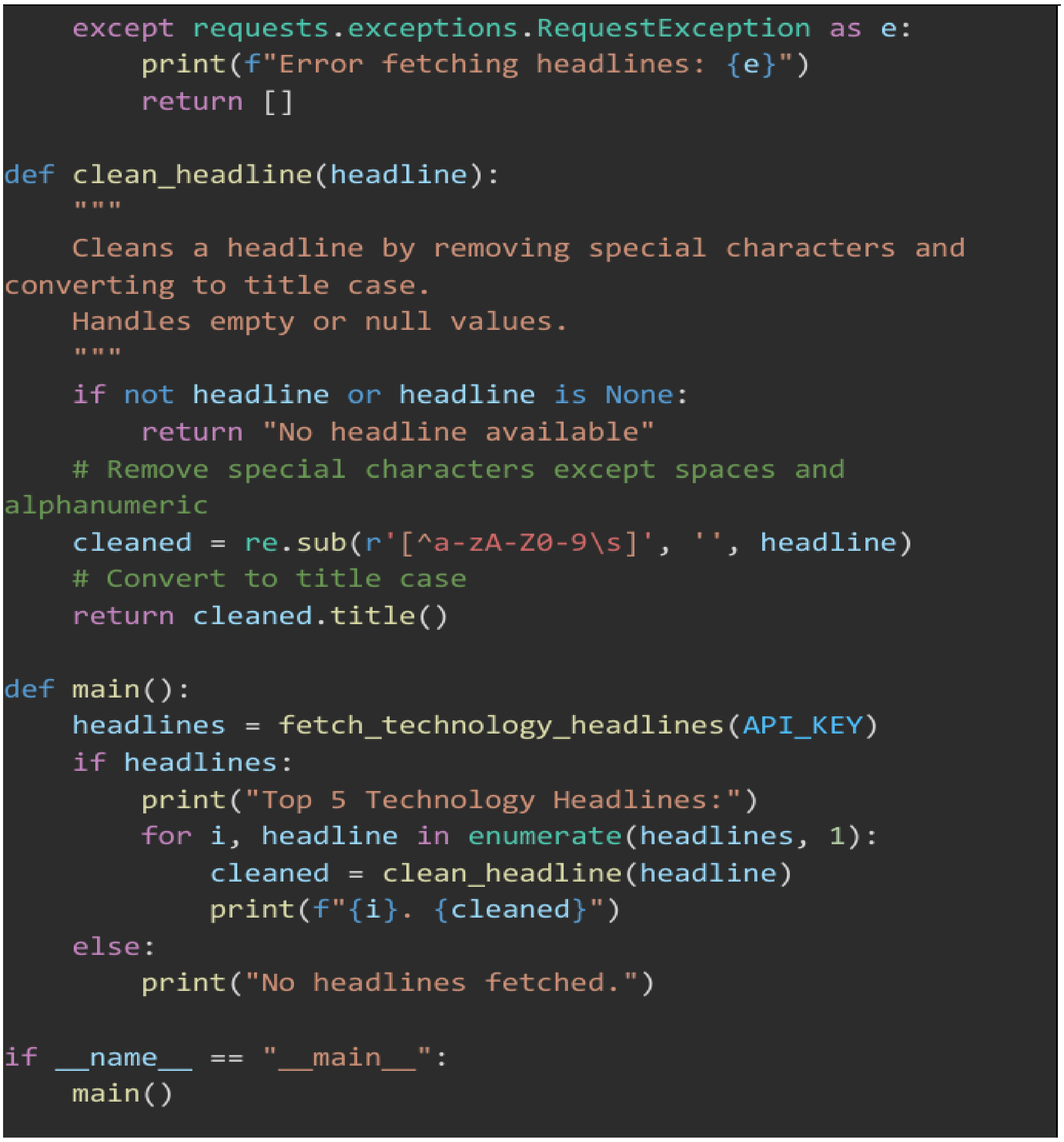
PROMPT:

Write a Python script that fetches the top 5 technology news headlines from a news API with timeout error handling. Clean the headlines by removing special characters, converting them to title case, and handle any empty or null values in the API response.

CODE:

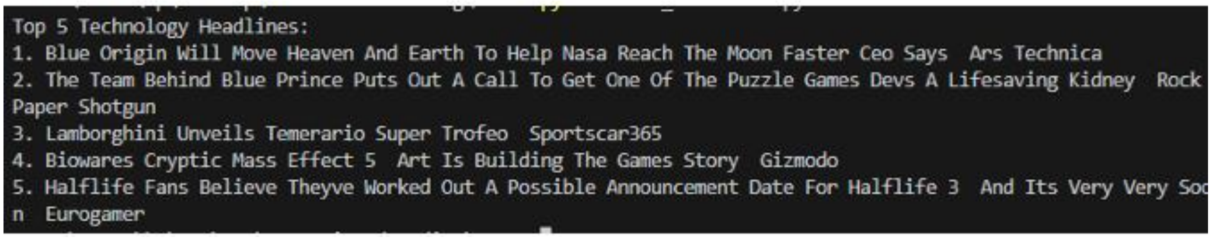


```
import requests
import re
import json

# Placeholder for API key - replace with your actual NewsAPI
key
API_KEY = '7c070ad417a94bd78cbda1feb78d4744'  # Get from
https://newsapi.org/

def fetch_technology_headlines(api_key, num_headlines=5,
timeout=10):
    """
    Fetches top technology headlines from NewsAPI.
    Handles timeout errors.
    """
    url = f'https://newsapi.org/v2/top-
headlines?category=technology&apiKey={api_key}'
    try:
        response = requests.get(url, timeout=timeout)
        response.raise_for_status()
        data = response.json()
        articles = data.get('articles', [])
        headlines = [article.get('title', '') for article in
articles[:num_headlines]]
        return headlines
    except requests.exceptions.Timeout:
        print("Error: Request timed out. Please try again
later.")
        return []
```



```python
    except requests.exceptions.RequestException as e:
        print(f"Error fetching headlines: {e}")
        return []

def clean_headline(headline):
    """
    Cleans a headline by removing special characters and 
converting to title case.
    Handles empty or null values.
    """
    if not headline or headline is None:
        return "No headline available"
    # Remove special characters except spaces and 
alphanumeric
    cleaned = re.sub(r'[^a-zA-Z0-9\s]', '', headline)
    # Convert to title case
    return cleaned.title()

def main():
    headlines = fetch_technology_headlines(API_KEY)
    if headlines:
        print("Top 5 Technology Headlines:")
        for i, headline in enumerate(headlines, 1):
            cleaned = clean_headline(headline)
            print(f"{i}. {cleaned}")
    else:
        print("No headlines fetched.")

if __name__ == "__main__":
    main()
```

OUTPUT:



```
Top 5 Technology Headlines:
1. Blue Origin Will Move Heaven And Earth To Help Nasa Reach The Moon Faster Ceo Says  Ars Technica
2. The Team Behind Blue Prince Puts Out A Call To Get One Of The Puzzle Games Devs A Lifesaving Kidney  Rock
Paper Shotgun
3. Lamborghini Unveils Temerario Super Trofeo  Sportscar365
4. Biowares Cryptic Mass Effect 5  Art Is Building The Games Story  Gizmodo
5. Halflife Fans Believe Theyve Worked Out A Possible Announcement Date For Halflife 3  And Its Very Very Soo
n  Eurogamer
```

OBSERVATION:

The script retrieves technology headlines reliably even during slow or incomplete API responses. It also cleans and formats each headline, ensuring clear and readable output while managing empty or missing values