Lab 18– API Integration: Connecting to external services with error handling

---

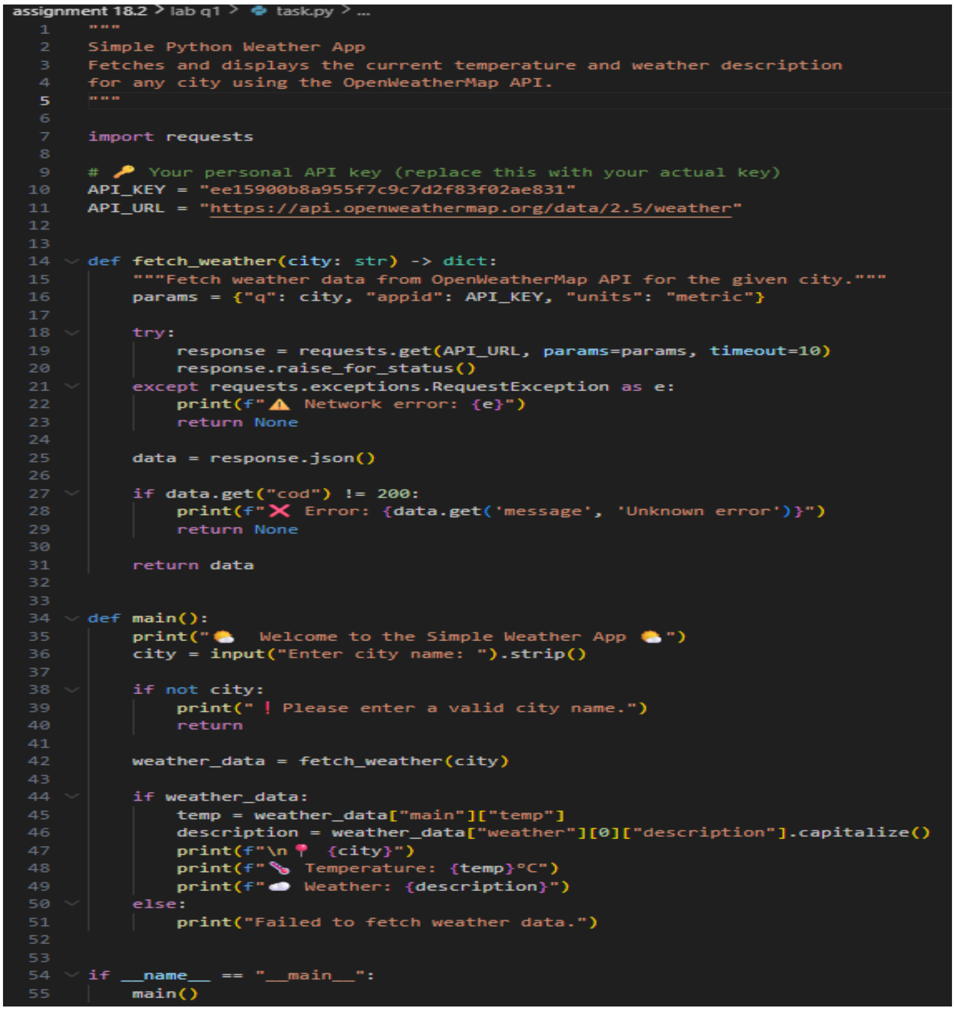
Lab Question 1 : Weather Forecasting API

Task 1

Prompt:

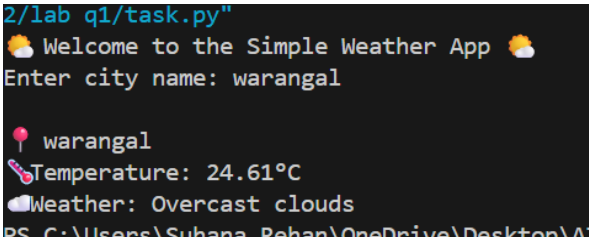
Write a Python script using the OpenWeatherMap API to fetch and display current temperature and weather description for a given city with error handling for missing or invalid API key.

Code:



```python
"""
Simple Python Weather App
Fetches and displays the current temperature and weather description
for any city using the OpenWeatherMap API.
"""

import requests

# 🔑 Your personal API key (replace this with your actual key)
API_KEY = "ee15900b8a955f7c9c7d2f83f02ae831"
API_URL = "https://api.openweathermap.org/data/2.5/weather"


def fetch_weather(city: str) -> dict:
    """Fetch weather data from OpenWeatherMap API for the given city."""
    params = {"q": city, "appid": API_KEY, "units": "metric"}

    try:
        response = requests.get(API_URL, params=params, timeout=10)
        response.raise_for_status()
    except requests.exceptions.RequestException as e:
        print(f"⚠ Network error: {e}")
        return None

    data = response.json()

    if data.get("cod") != 200:
        print(f"❌ Error: {data.get('message', 'Unknown error')}")
        return None

    return data


def main():
    print("🌤  Welcome to the Simple Weather App 🌤")
    city = input("Enter city name: ").strip()

    if not city:
        print(" ❗ Please enter a valid city name.")
        return

    weather_data = fetch_weather(city)

    if weather_data:
        temp = weather_data["main"]["temp"]
        description = weather_data["weather"][0]["description"].capitalize()
        print(f"\n📍  {city}")
        print(f"🌡 Temperature: {temp}°C")
        print(f"☁ Weather: {description}")
    else:
        print("Failed to fetch weather data.")


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

Output:



```
2/lab q1/task.py"
⛅ Welcome to the Simple Weather App ⛅
Enter city name: warangal

📍 warangal
🌡️Temperature: 24.61°C
☁️Weather: Overcast clouds
PS C:\Users\Suhana Rehan\OneDrive\Desktop\A
```

Observations:

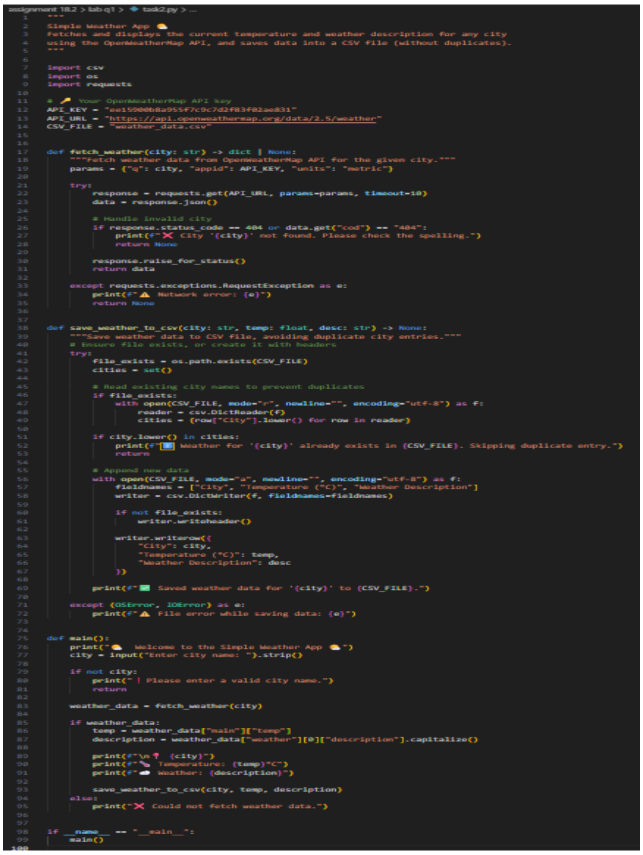
- I used AI to help me understand how to send API requests using the requests module.
- It also showed me how to check if my API key is missing or wrong.
- I learned that JSON responses can be accessed like dictionaries in Python.

Task 2

Prompt:

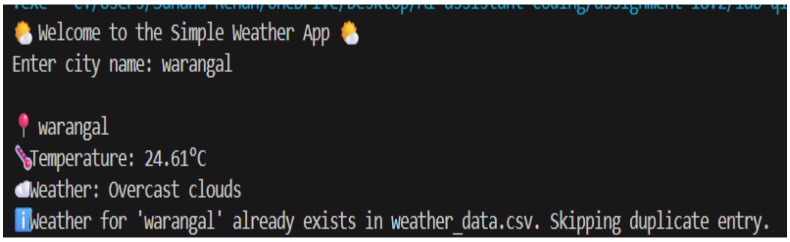
Extend the weather script to save weather data into a CSV file without duplicates and handle file read/write errors.

Code:

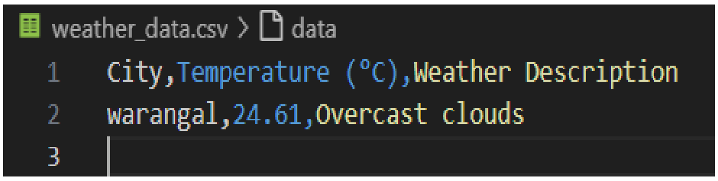


```python
"""
Simple Weather App ⛅
Fetches and displays the current temperature and weather description for any city
using the OpenWeatherMap API, and saves data into a CSV file (without duplicates).
"""

import csv
import os
import requests

# 🔑 Your OpenWeatherMap API key
API_KEY = "ee15900b8a955f7c9c7d2f83f02ae831"
API_URL = "https://api.openweathermap.org/data/2.5/weather"
CSV_FILE = "weather_data.csv"


def fetch_weather(city: str) -> dict | None:
    """Fetch weather data from OpenWeatherMap API for the given city."""
    params = {"q": city, "appid": API_KEY, "units": "metric"}

    try:
        response = requests.get(API_URL, params=params, timeout=10)
        data = response.json()

        # Handle invalid city
        if response.status_code == 404 or data.get("cod") == "404":
            print(f"❌ City '{city}' not found. Please check the spelling.")
            return None

        response.raise_for_status()
        return data

    except requests.exceptions.RequestException as e:
        print(f"⚠️ Network error: {e}")
        return None


def save_weather_to_csv(city: str, temp: float, desc: str) -> None:
    """Save weather data to CSV file, avoiding duplicate city entries."""
    # Ensure file exists, or create it with headers
    try:
        file_exists = os.path.exists(CSV_FILE)
        cities = set()

        # Read existing city names to prevent duplicates
        if file_exists:
            with open(CSV_FILE, mode="r", newline="", encoding="utf-8") as f:
                reader = csv.DictReader(f)
                cities = {row["City"].lower() for row in reader}

        if city.lower() in cities:
            print(f"ℹ️ Weather for '{city}' already exists in {CSV_FILE}. Skipping duplicate entry.")
            return

        # Append new data
        with open(CSV_FILE, mode="a", newline="", encoding="utf-8") as f:
            fieldnames = ["City", "Temperature (°C)", "Weather Description"]
            writer = csv.DictWriter(f, fieldnames=fieldnames)

            if not file_exists:
                writer.writeheader()

            writer.writerow({
                "City": city,
                "Temperature (°C)": temp,
                "Weather Description": desc
            })

        print(f"✅ Saved weather data for '{city}' to {CSV_FILE}.")

    except (OSError, IOError) as e:
        print(f"⚠️ File error while saving data: {e}")


def main():
    print("⛅  Welcome to the Simple Weather App ⛅")
    city = input("Enter city name: ").strip()

    if not city:
        print("❗ Please enter a valid city name.")
        return

    weather_data = fetch_weather(city)

    if weather_data:
        temp = weather_data["main"]["temp"]
        description = weather_data["weather"][0]["description"].capitalize()

        print(f"\n📍  {city}")
        print(f"🌡️ Temperature: {temp}°C")
        print(f"☁️ Weather: {description}")

        save_weather_to_csv(city, temp, description)
    else:
        print("❌ Could not fetch weather data.")


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

Output:



```
🌥Welcome to the Simple Weather App 🌥
Enter city name: warangal

📍 warangal
🌡Temperature: 24.61°C
☁Weather: Overcast clouds
ℹWeather for 'warangal' already exists in weather_data.csv. Skipping duplicate entry.
```



weather_data.csv > data

```
City,Temperature (°C),Weather Description
warangal,24.61,Overcast clouds

```

Observations:

- I learned how to use csv.DictWriter to store data safely.
- The AI helped me avoid duplicate city entries by checking before writing.
- I also handled file errors using try and except blocks.

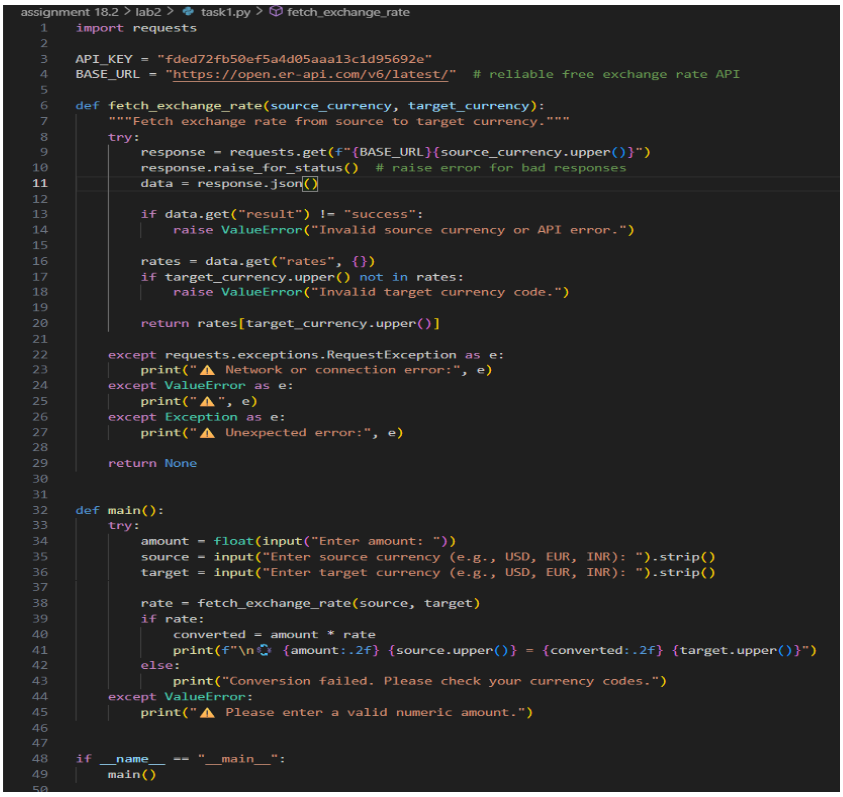
Lab Question 2: Currency Exchange Rate API

Task 1

Prompt:

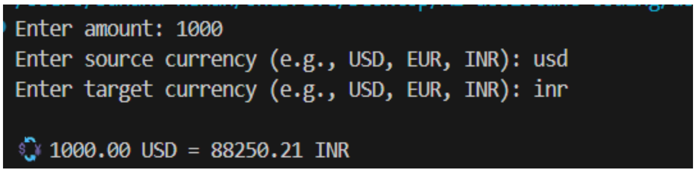
Write a Python script that takes user input (amount, source, target currency) and fetches the latest exchange rate with error handling for invalid currency codes.

Code:



```python
import requests

API_KEY = "fded72fb50ef5a4d05aaa13c1d95692e"
BASE_URL = "https://open.er-api.com/v6/latest/"  # reliable free exchange rate API

def fetch_exchange_rate(source_currency, target_currency):
    """Fetch exchange rate from source to target currency."""
    try:
        response = requests.get(f"{BASE_URL}{source_currency.upper()}")
        response.raise_for_status()  # raise error for bad responses
        data = response.json()

        if data.get("result") != "success":
            raise ValueError("Invalid source currency or API error.")

        rates = data.get("rates", {})
        if target_currency.upper() not in rates:
            raise ValueError("Invalid target currency code.")

        return rates[target_currency.upper()]

    except requests.exceptions.RequestException as e:
        print("⚠ Network or connection error:", e)
    except ValueError as e:
        print("⚠", e)
    except Exception as e:
        print("⚠ Unexpected error:", e)

    return None


def main():
    try:
        amount = float(input("Enter amount: "))
        source = input("Enter source currency (e.g., USD, EUR, INR): ").strip()
        target = input("Enter target currency (e.g., USD, EUR, INR): ").strip()

        rate = fetch_exchange_rate(source, target)
        if rate:
            converted = amount * rate
            print(f"\n💱 {amount:.2f} {source.upper()} = {converted:.2f} {target.upper()}")
        else:
            print("Conversion failed. Please check your currency codes.")
    except ValueError:
        print("⚠ Please enter a valid numeric amount.")


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

Output:



```
Enter amount: 1000
Enter source currency (e.g., USD, EUR, INR): usd
Enter target currency (e.g., USD, EUR, INR): inr

💱 1000.00 USD = 88250.21 INR
```

Observations:

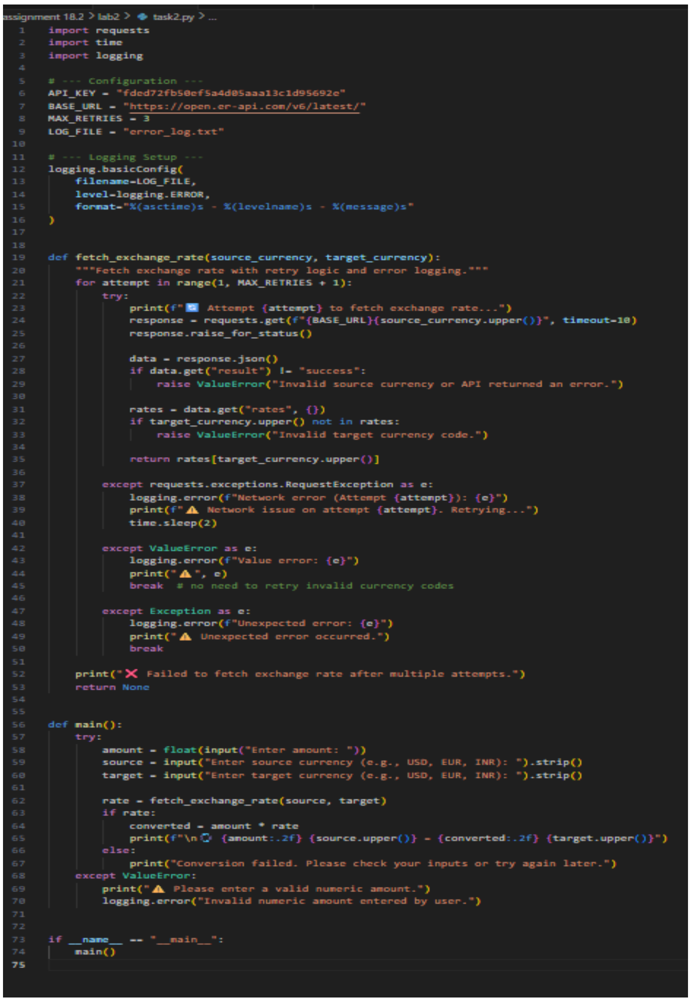
- I used AI to help create input prompts and handle wrong currency codes.
- I understood how to check if the API response was valid before using it.
- The script now gives a clear error message when something goes wrong.

Task 2

Prompt:

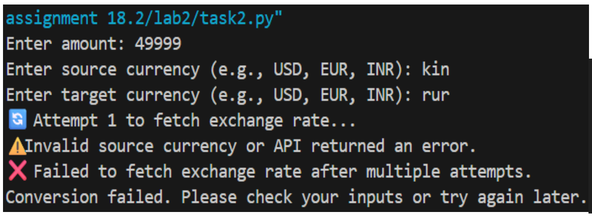
Add retry logic to the currency script to attempt the API call up to three times if it fails, and log all errors into a local file.

Code:

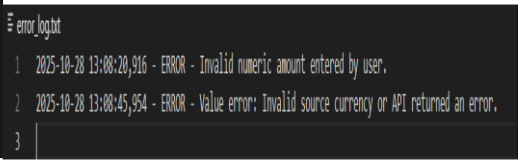


```python
import requests
import time
import logging

# --- Configuration ---
API_KEY = "fded72fb50ef5a4d05aaa13c1d95692e"
BASE_URL = "https://open.er-api.com/v6/latest/"
MAX_RETRIES = 3
LOG_FILE = "error_log.txt"

# --- Logging Setup ---
logging.basicConfig(
    filename=LOG_FILE,
    level=logging.ERROR,
    format="%(asctime)s - %(levelname)s - %(message)s"
)


def fetch_exchange_rate(source_currency, target_currency):
    """Fetch exchange rate with retry logic and error logging."""
    for attempt in range(1, MAX_RETRIES + 1):
        try:
            print(f"🔄 Attempt {attempt} to fetch exchange rate...")
            response = requests.get(f"{BASE_URL}{source_currency.upper()}", timeout=10)
            response.raise_for_status()

            data = response.json()
            if data.get("result") != "success":
                raise ValueError("Invalid source currency or API returned an error.")

            rates = data.get("rates", {})
            if target_currency.upper() not in rates:
                raise ValueError("Invalid target currency code.")

            return rates[target_currency.upper()]

        except requests.exceptions.RequestException as e:
            logging.error(f"Network error (Attempt {attempt}): {e}")
            print(f"⚠ Network issue on attempt {attempt}. Retrying...")
            time.sleep(2)

        except ValueError as e:
            logging.error(f"Value error: {e}")
            print("⚠", e)
            break  # no need to retry invalid currency codes

        except Exception as e:
            logging.error(f"Unexpected error: {e}")
            print("⚠ Unexpected error occurred.")
            break

    print("❌ Failed to fetch exchange rate after multiple attempts.")
    return None


def main():
    try:
        amount = float(input("Enter amount: "))
        source = input("Enter source currency (e.g., USD, EUR, INR): ").strip()
        target = input("Enter target currency (e.g., USD, EUR, INR): ").strip()

        rate = fetch_exchange_rate(source, target)
        if rate:
            converted = amount * rate
            print(f"\n🔄 {amount:.2f} {source.upper()} = {converted:.2f} {target.upper()}")
        else:
            print("Conversion failed. Please check your inputs or try again later.")
    except ValueError:
        print("⚠ Please enter a valid numeric amount.")
        logging.error("Invalid numeric amount entered by user.")


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

Output:



```
assignment 18.2/lab2/task2.py"
Enter amount: 49999
Enter source currency (e.g., USD, EUR, INR): kin
Enter target currency (e.g., USD, EUR, INR): rur
🔄 Attempt 1 to fetch exchange rate...
⚠Invalid source currency or API returned an error.
❌ Failed to fetch exchange rate after multiple attempts.
Conversion failed. Please check your inputs or try again later.
```



```
2025-10-28 13:08:20,916 - ERROR - Invalid numeric amount entered by user.
2025-10-28 13:08:45,954 - ERROR - Value error: Invalid source currency or API returned an error.
```

Observations:

- AI showed me how to use a simple retry loop with for and try.
- I created an error log file using basic file handling.
- I learned that logging helps when APIs fail due to server issues.

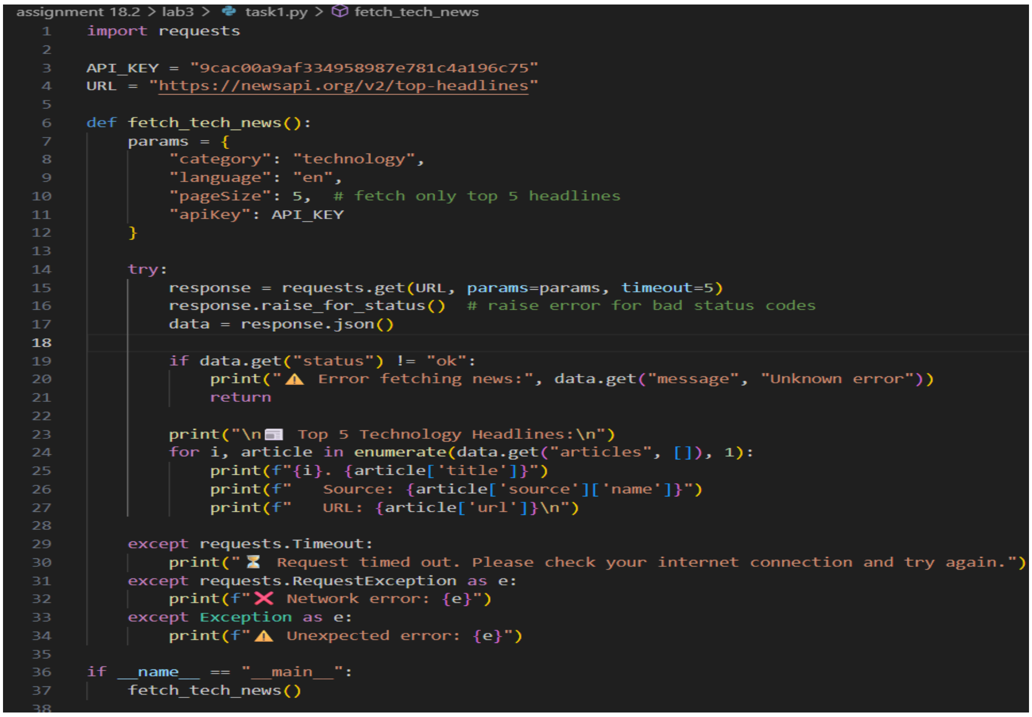
Lab Question 3 – News Headlines API

Task 1

Prompt:

Write a Python script to fetch and print the top 5 technology headlines from a news API, with timeout error handling.

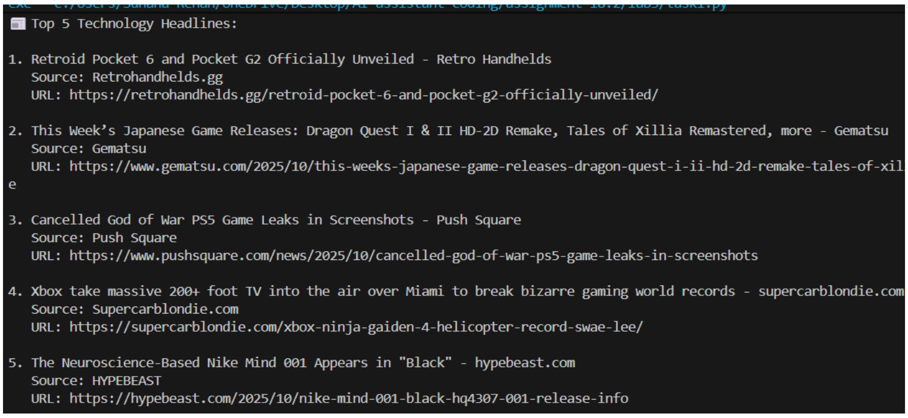
Code:



assignment 18.2 > lab3 > task1.py > fetch_tech_news

```python
import requests

API_KEY = "9cac00a9af334958987e781c4a196c75"
URL = "https://newsapi.org/v2/top-headlines"

def fetch_tech_news():
    params = {
        "category": "technology",
        "language": "en",
        "pageSize": 5,  # fetch only top 5 headlines
        "apiKey": API_KEY
    }

    try:
        response = requests.get(URL, params=params, timeout=5)
        response.raise_for_status()  # raise error for bad status codes
        data = response.json()

        if data.get("status") != "ok":
            print("⚠️ Error fetching news:", data.get("message", "Unknown error"))
            return

        print("\n📰 Top 5 Technology Headlines:\n")
        for i, article in enumerate(data.get("articles", []), 1):
            print(f"{i}. {article['title']}")
            print(f"   Source: {article['source']['name']}")
            print(f"   URL: {article['url']}\n")

    except requests.Timeout:
        print("⏳ Request timed out. Please check your internet connection and try again.")
    except requests.RequestException as e:
        print(f"❌ Network error: {e}")
    except Exception as e:
        print(f"⚠️ Unexpected error: {e}")

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

Output:



```
📰 Top 5 Technology Headlines:

1. Retroid Pocket 6 and Pocket G2 Officially Unveiled - Retro Handhelds
   Source: Retrohandhelds.gg
   URL: https://retrohandhelds.gg/retroid-pocket-6-and-pocket-g2-officially-unveiled/

2. This Week’s Japanese Game Releases: Dragon Quest I & II HD-2D Remake, Tales of Xillia Remastered, more - Gematsu
   Source: Gematsu
   URL: https://www.gematsu.com/2025/10/this-weeks-japanese-game-releases-dragon-quest-i-ii-hd-2d-remake-tales-of-xil
e

3. Cancelled God of War PS5 Game Leaks in Screenshots - Push Square
   Source: Push Square
   URL: https://www.pushsquare.com/news/2025/10/cancelled-god-of-war-ps5-game-leaks-in-screenshots

4. Xbox take massive 200+ foot TV into the air over Miami to break bizarre gaming world records - supercarblondie.com
   Source: Supercarblondie.com
   URL: https://supercarblondie.com/xbox-ninja-gaiden-4-helicopter-record-swae-lee/

5. The Neuroscience-Based Nike Mind 001 Appears in "Black" - hypebeast.com
   Source: HYPEBEAST
   URL: https://hypebeast.com/2025/10/nike-mind-001-black-hq4307-001-release-info
```

Observations:

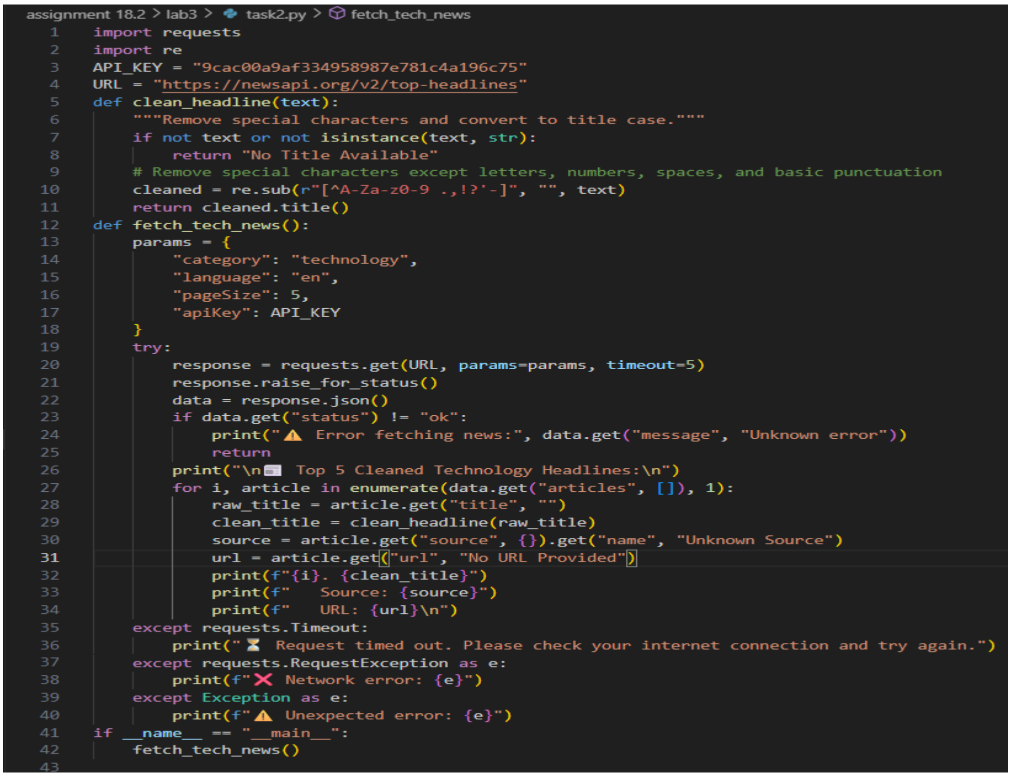
- AI helped me understand how to use the timeout parameter in requests.
- The script now prints headlines clearly in the console.
- I learned how to handle slow or failed responses without crashing.

Task 2

Prompt:

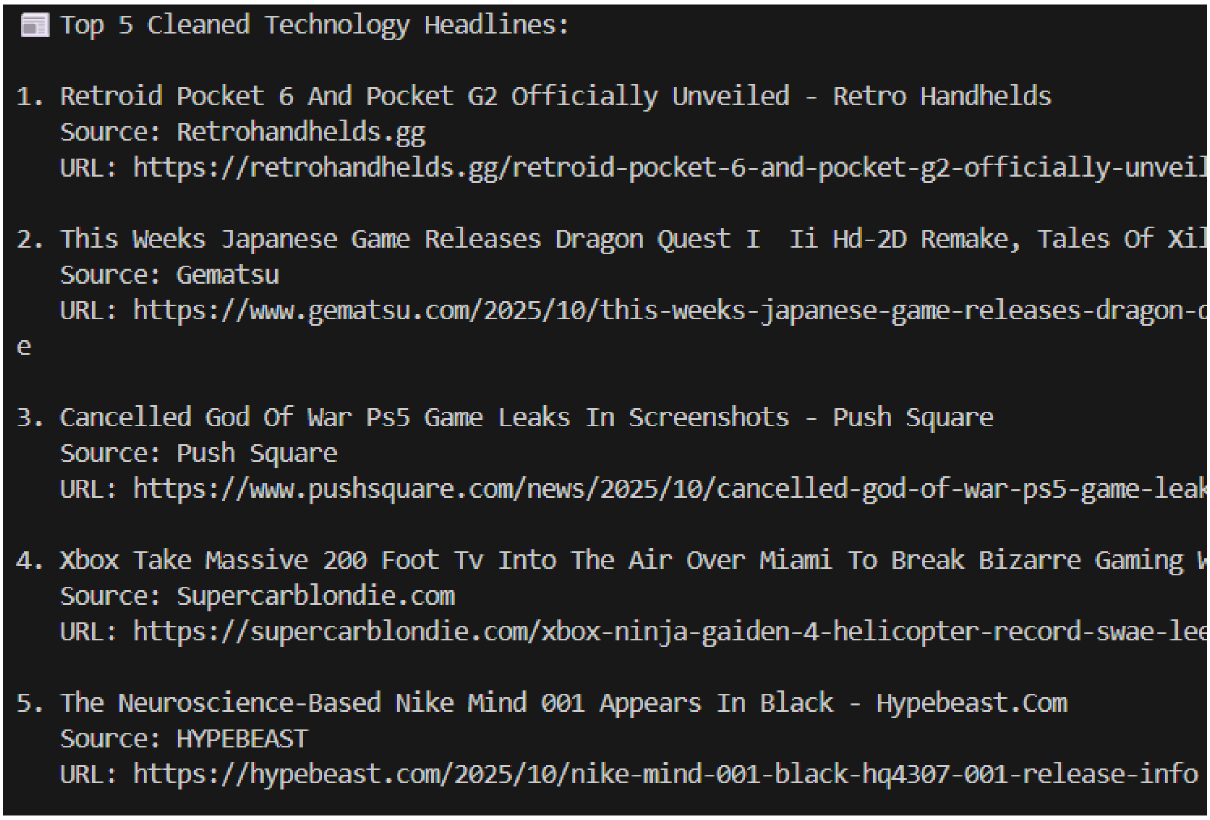
Clean and preprocess the headlines by removing special characters and converting them to title case, handling empty or null values.

Code:



```python
import requests
import re
API_KEY = "9cac00a9af334958987e781c4a196c75"
URL = "https://newsapi.org/v2/top-headlines"
def clean_headline(text):
    """Remove special characters and convert to title case."""
    if not text or not isinstance(text, str):
        return "No Title Available"
    # Remove special characters except letters, numbers, spaces, and basic punctuation
    cleaned = re.sub(r"[^A-Za-z0-9 .,!?'-]", "", text)
    return cleaned.title()
def fetch_tech_news():
    params = {
        "category": "technology",
        "language": "en",
        "pageSize": 5,
        "apiKey": API_KEY
    }
    try:
        response = requests.get(URL, params=params, timeout=5)
        response.raise_for_status()
        data = response.json()
        if data.get("status") != "ok":
            print("⚠️ Error fetching news:", data.get("message", "Unknown error"))
            return
        print("\n📰 Top 5 Cleaned Technology Headlines:\n")
        for i, article in enumerate(data.get("articles", []), 1):
            raw_title = article.get("title", "")
            clean_title = clean_headline(raw_title)
            source = article.get("source", {}).get("name", "Unknown Source")
            url = article.get("url", "No URL Provided")
            print(f"{i}. {clean_title}")
            print(f"   Source: {source}")
            print(f"   URL: {url}\n")
    except requests.Timeout:
        print("⏳ Request timed out. Please check your internet connection and try again.")
    except requests.RequestException as e:
        print(f"❌ Network error: {e}")
    except Exception as e:
        print(f"⚠️ Unexpected error: {e}")
if __name__ == "__main__":
    fetch_tech_news()
```

Output:



```
📰 Top 5 Cleaned Technology Headlines:

1. Retroid Pocket 6 And Pocket G2 Officially Unveiled - Retro Handhelds
   Source: Retrohandhelds.gg
   URL: https://retrohandhelds.gg/retroid-pocket-6-and-pocket-g2-officially-unveil

2. This Weeks Japanese Game Releases Dragon Quest I  Ii Hd-2D Remake, Tales Of Xil
   Source: Gematsu
   URL: https://www.gematsu.com/2025/10/this-weeks-japanese-game-releases-dragon-q
e

3. Cancelled God Of War Ps5 Game Leaks In Screenshots - Push Square
   Source: Push Square
   URL: https://www.pushsquare.com/news/2025/10/cancelled-god-of-war-ps5-game-leak

4. Xbox Take Massive 200 Foot Tv Into The Air Over Miami To Break Bizarre Gaming W
   Source: Supercarblondie.com
   URL: https://supercarblondie.com/xbox-ninja-gaiden-4-helicopter-record-swae-lee

5. The Neuroscience-Based Nike Mind 001 Appears In Black - Hypebeast.Com
   Source: HYPEBEAST
   URL: https://hypebeast.com/2025/10/nike-mind-001-black-hq4307-001-release-info
```

Observations:

- I used AI to clean text using regular expressions and title() function.
- It helped me skip empty or None headlines safely.
- The cleaned headlines look much more readable now.

  -