# AI ASSISTED CODING LAB
# ASSIGNMENT-18

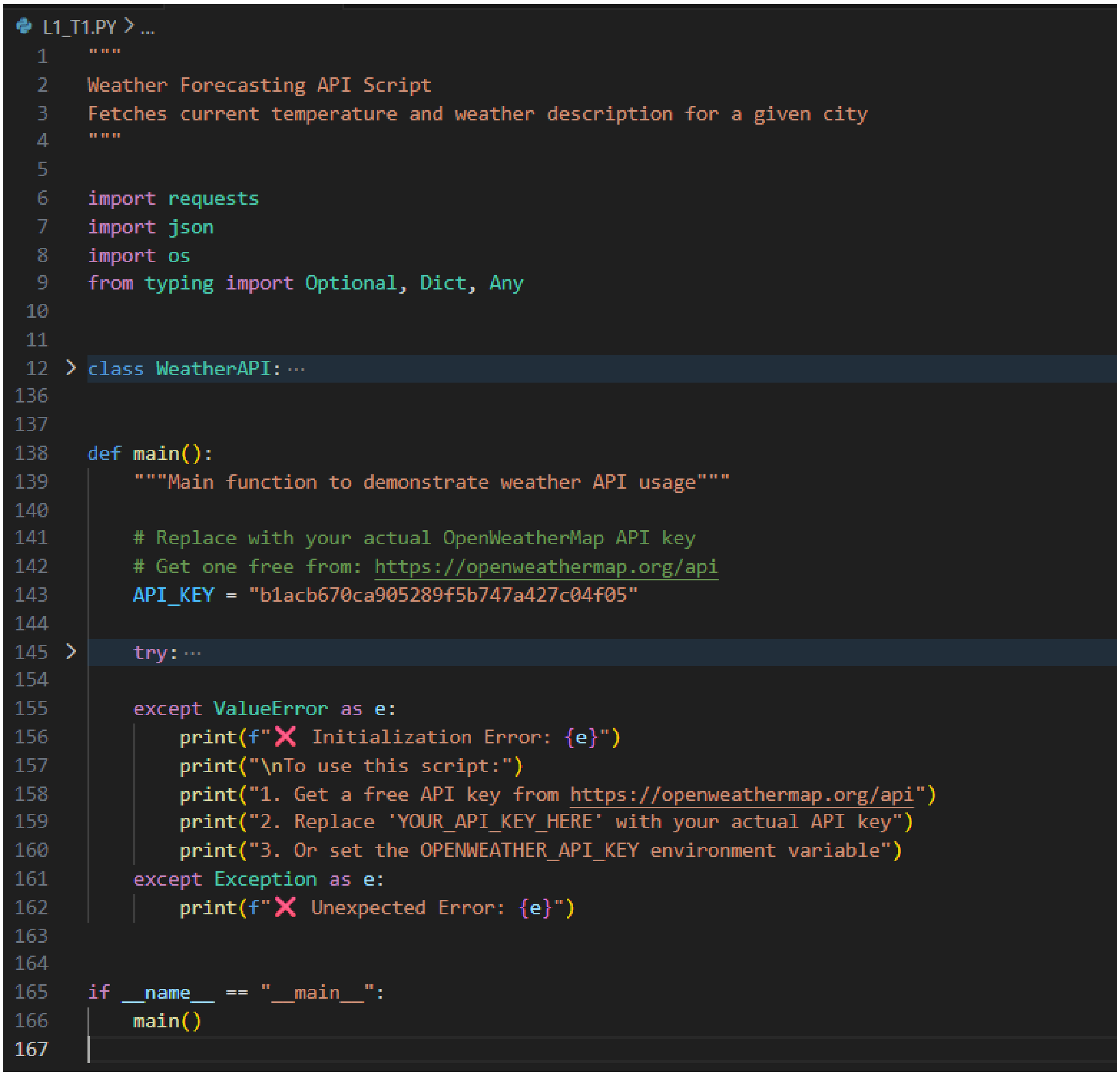
Name: K. Jashuva<br>
Enrollment No: 2503a51l16<br>
Batch No: 19

**Lab 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.

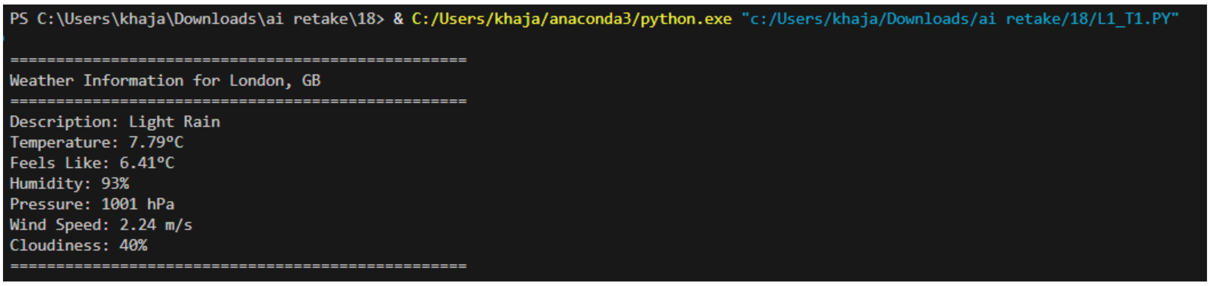
**CODE:-**



L1_T1.PY > ...

```python
"""
Weather Forecasting API Script
Fetches current temperature and weather description for a given city
"""

import requests
import json
import os
from typing import Optional, Dict, Any


class WeatherAPI:···


def main():
    """Main function to demonstrate weather API usage"""

    # Replace with your actual OpenWeatherMap API key
    # Get one free from: https://openweathermap.org/api
    API_KEY = "b1acb670ca905289f5b747a427c04f05"

    try:···

    except ValueError as e:
        print(f"❌ Initialization Error: {e}")
        print("\nTo use this script:")
        print("1. Get a free API key from https://openweathermap.org/api")
        print("2. Replace 'YOUR_API_KEY_HERE' with your actual API key")
        print("3. Or set the OPENWEATHER_API_KEY environment variable")
    except Exception as e:
        print(f"❌ Unexpected Error: {e}")


if __name__ == "__main__":
    main()

```

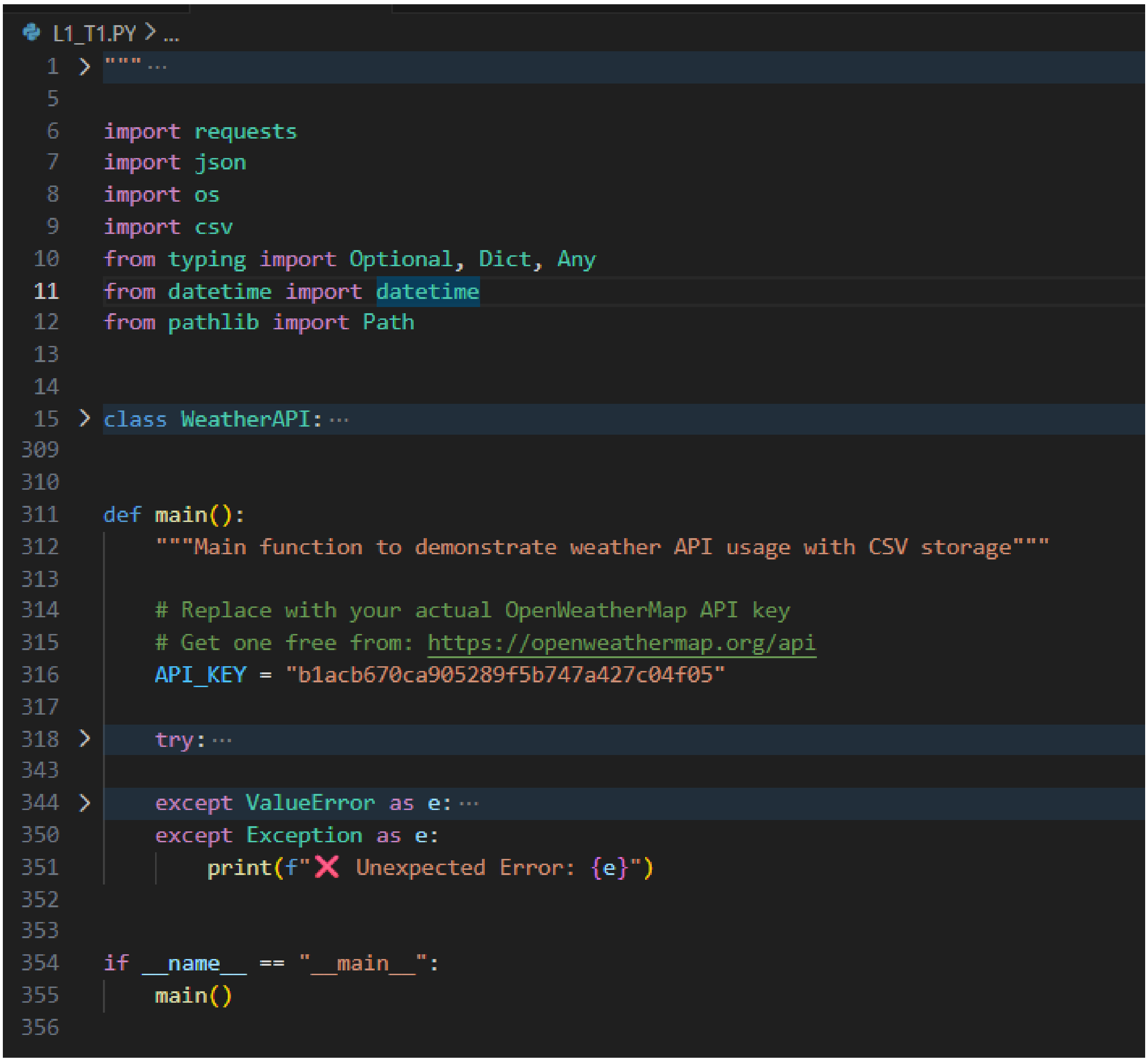
OUTPUT:-



```
PS C:\Users\khaja\Downloads\ai retake\18> & C:/Users/khaja/anaconda3/python.exe "c:/Users/khaja/Downloads/ai retake/18/L1_T1.PY"

==================================================
Weather Information for London, GB
==================================================
Description: Light Rain
Temperature: 7.79°C
Feels Like: 6.41°C
Humidity: 93%
Pressure: 1001 hPa
Wind Speed: 2.24 m/s
Cloudiness: 40%
==================================================
```

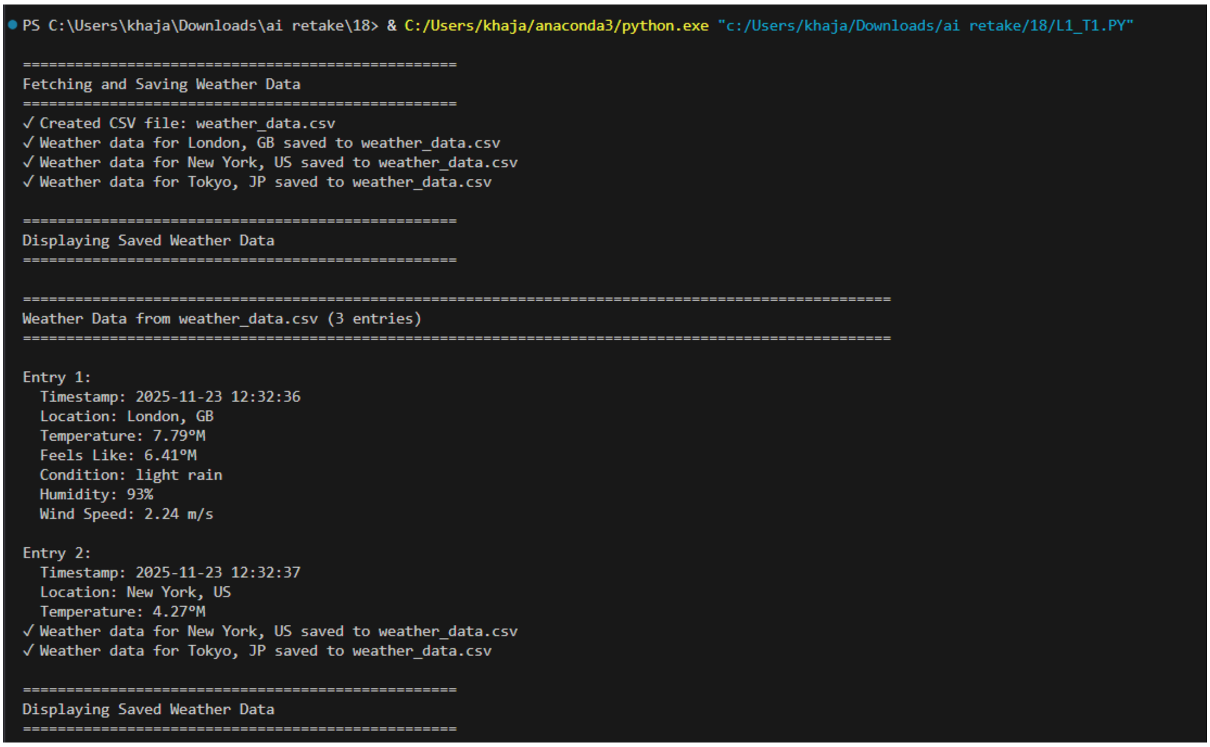
- **Task-2:** Extend the script to save the weather data into a local CSV file, ensuring that duplicate entries are avoided. Implement error handling for file I/O exceptions.

**CODE:-**



```python
"""...

import requests
import json
import os
import csv
from typing import Optional, Dict, Any
from datetime import datetime
from pathlib import Path


class WeatherAPI:···


def main():
    """Main function to demonstrate weather API usage with CSV storage"""

    # Replace with your actual OpenWeatherMap API key
    # Get one free from: https://openweathermap.org/api
    API_KEY = "b1acb670ca905289f5b747a427c04f05"

    try:···

    except ValueError as e:···
    except Exception as e:
        print(f"❌ Unexpected Error: {e}")


if __name__ == "__main__":
    main()

```

**OUTPUT:-**



```
PS C:\Users\khaja\Downloads\ai retake\18> & C:/Users/khaja/anaconda3/python.exe "c:/Users/khaja/Downloads/ai retake/18/L1_T1.PY"

==================================================
Fetching and Saving Weather Data
==================================================
✓ Created CSV file: weather_data.csv
✓ Weather data for London, GB saved to weather_data.csv
✓ Weather data for New York, US saved to weather_data.csv
✓ Weather data for Tokyo, JP saved to weather_data.csv

==================================================
Displaying Saved Weather Data
==================================================

====================================================================================================
Weather Data from weather_data.csv (3 entries)
====================================================================================================

Entry 1:
  Timestamp: 2025-11-23 12:32:36
  Location: London, GB
  Temperature: 7.79°M
  Feels Like: 6.41°M
  Condition: light rain
  Humidity: 93%
  Wind Speed: 2.24 m/s

Entry 2:
  Timestamp: 2025-11-23 12:32:37
  Location: New York, US
  Temperature: 4.27°M
✓ Weather data for New York, US saved to weather_data.csv
✓ Weather data for Tokyo, JP saved to weather_data.csv

==================================================
Displaying Saved Weather Data
==================================================
```

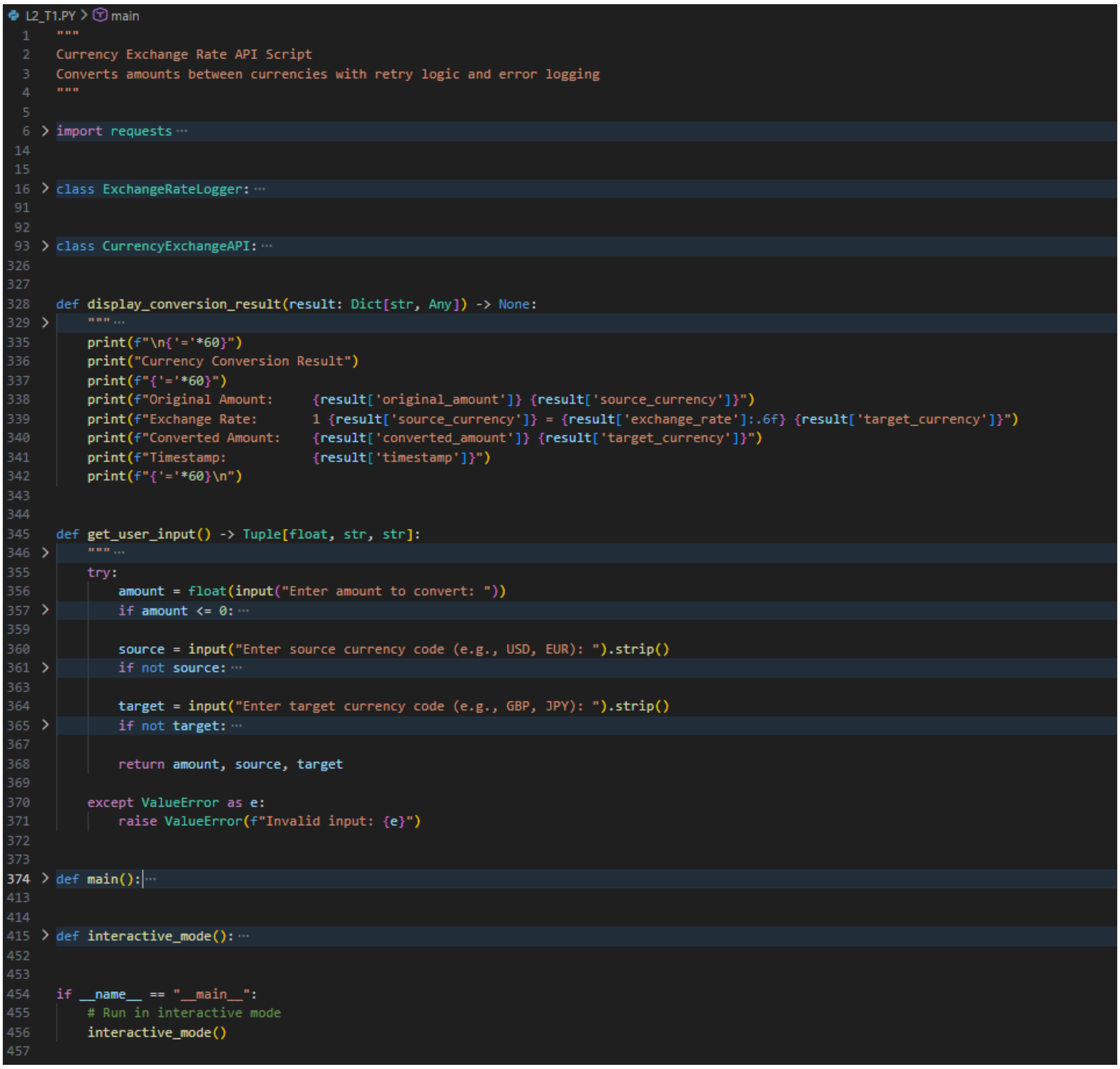
**Observation:** The program successfully retrieved real-time weather details, including temperature and conditions, for the specified city using the Weather API. It handled invalid or missing API keys effectively and saved the data into a local CSV file while avoiding duplicates and managing file I/O errors smoothly.

**Lab 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.

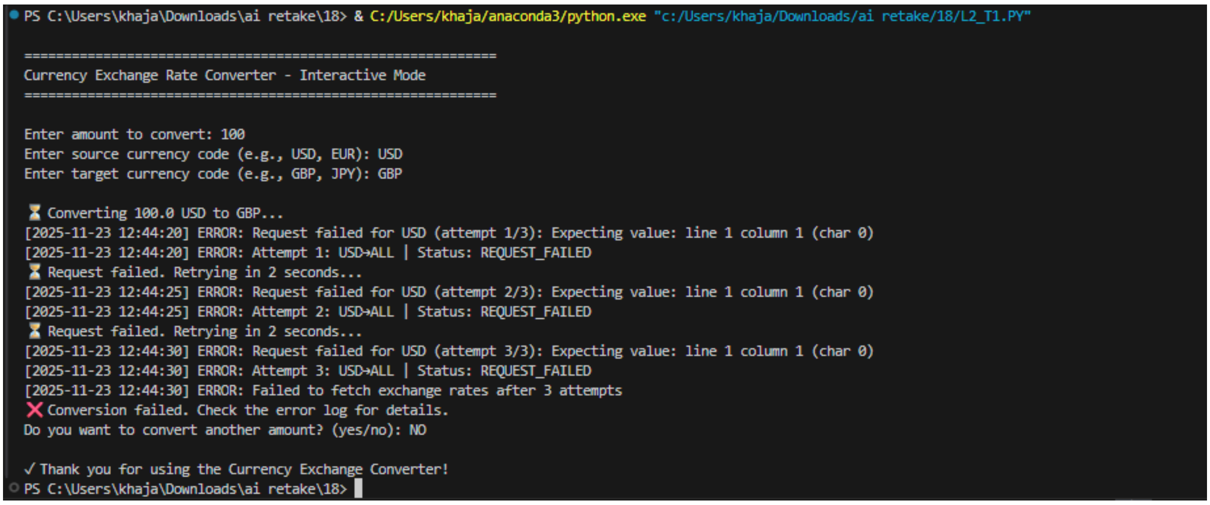
- **Task-1:** Write a script (with AI assistance) that takes user input (amount, source currency, target currency) and fetches the latest exchange rate from the API. Include errors in handling invalid currency codes.
- **Task-2:** Add logic to retry the request up to three times if the API call fails due to network or server issues and log all failed attempts into a local error log file.

**Code:**



```
L2_T1.PY > main
"""
Currency Exchange Rate API Script
Converts amounts between currencies with retry logic and error logging
"""

import requests ···


class ExchangeRateLogger: ···


class CurrencyExchangeAPI: ···


def display_conversion_result(result: Dict[str, Any]) -> None:
    """···
    print(f"\n{'='*60}")
    print("Currency Conversion Result")
    print(f"{'='*60}")
    print(f"Original Amount:     {result['original_amount']} {result['source_currency']}")
    print(f"Exchange Rate:       1 {result['source_currency']} = {result['exchange_rate']:.6f} {result['target_currency']}")
    print(f"Converted Amount:    {result['converted_amount']} {result['target_currency']}")
    print(f"Timestamp:           {result['timestamp']}")
    print(f"{'='*60}\n")


def get_user_input() -> Tuple[float, str, str]:
    """···
    try:
        amount = float(input("Enter amount to convert: "))
        if amount <= 0: ···

        source = input("Enter source currency code (e.g., USD, EUR): ").strip()
        if not source: ···

        target = input("Enter target currency code (e.g., GBP, JPY): ").strip()
        if not target: ···

        return amount, source, target

    except ValueError as e:
        raise ValueError(f"Invalid input: {e}")


def main(): ···


def interactive_mode(): ···


if __name__ == "__main__":
    # Run in interactive mode
    interactive_mode()
```

OUTPUT:-



```
PS C:\Users\khaja\Downloads\ai retake\18> & C:/Users/khaja/anaconda3/python.exe "c:/Users/khaja/Downloads/ai retake/18/L2_T1.PY"

============================================================
Currency Exchange Rate Converter - Interactive Mode
============================================================

Enter amount to convert: 100
Enter source currency code (e.g., USD, EUR): USD
Enter target currency code (e.g., GBP, JPY): GBP

⏳ Converting 100.0 USD to GBP...
[2025-11-23 12:44:20] ERROR: Request failed for USD (attempt 1/3): Expecting value: line 1 column 1 (char 0)
[2025-11-23 12:44:20] ERROR: Attempt 1: USD→ALL | Status: REQUEST_FAILED
⏳ Request failed. Retrying in 2 seconds...
[2025-11-23 12:44:25] ERROR: Request failed for USD (attempt 2/3): Expecting value: line 1 column 1 (char 0)
[2025-11-23 12:44:25] ERROR: Attempt 2: USD→ALL | Status: REQUEST_FAILED
⏳ Request failed. Retrying in 2 seconds...
[2025-11-23 12:44:30] ERROR: Request failed for USD (attempt 3/3): Expecting value: line 1 column 1 (char 0)
[2025-11-23 12:44:30] ERROR: Attempt 3: USD→ALL | Status: REQUEST_FAILED
[2025-11-23 12:44:30] ERROR: Failed to fetch exchange rates after 3 attempts
❌ Conversion failed. Check the error log for details.
Do you want to convert another amount? (yes/no): NO

✓ Thank you for using the Currency Exchange Converter!
PS C:\Users\khaja\Downloads\ai retake\18> 
```

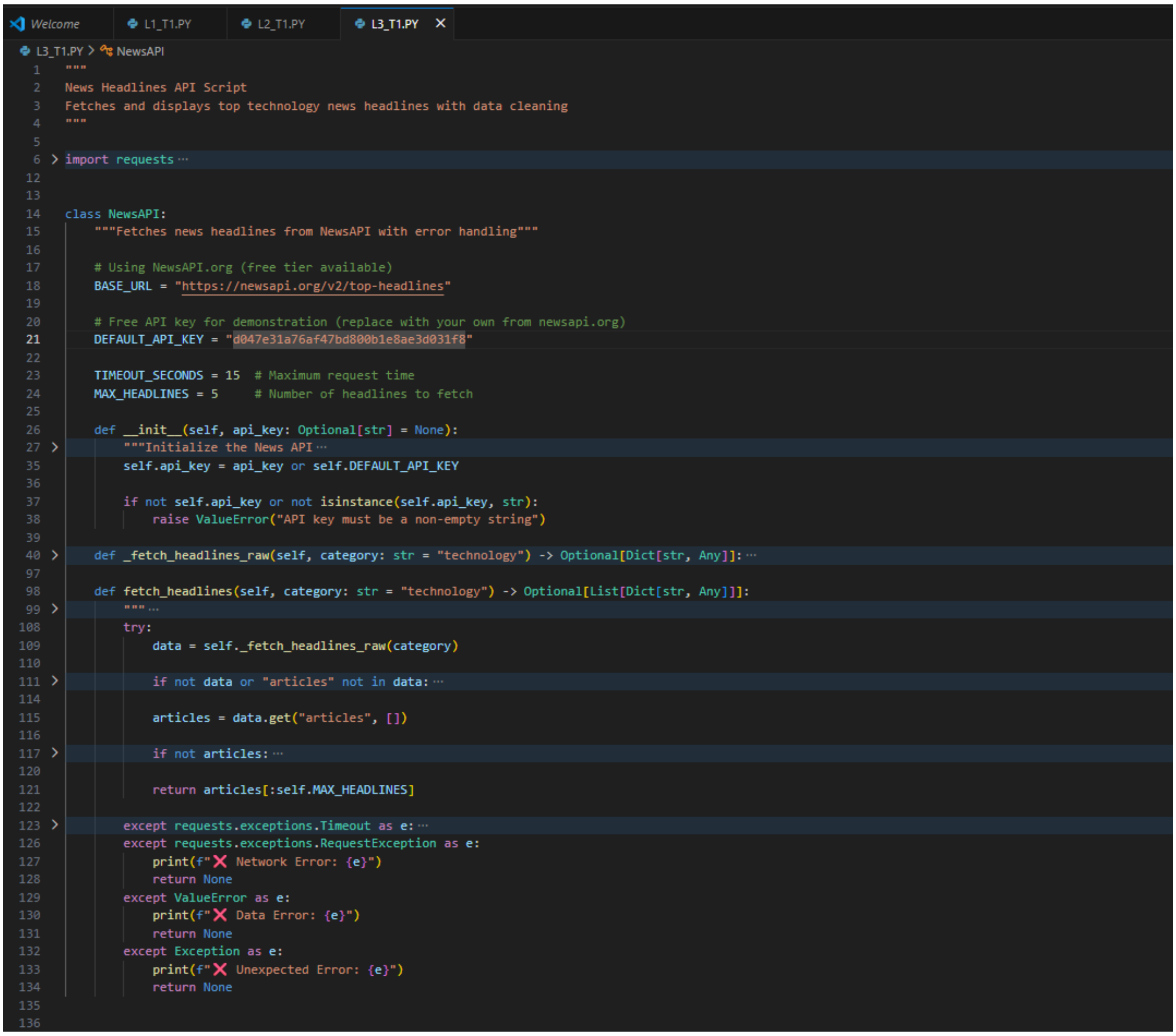
**Observation:** The script efficiently converted currencies by fetching real-time exchange rates from the API while validating user inputs for incorrect codes. It also implemented a retry mechanism to handle network or server issues and logged failed attempts, ensuring reliable performance and smooth execution.

## Lab Question-3: News Headlines API

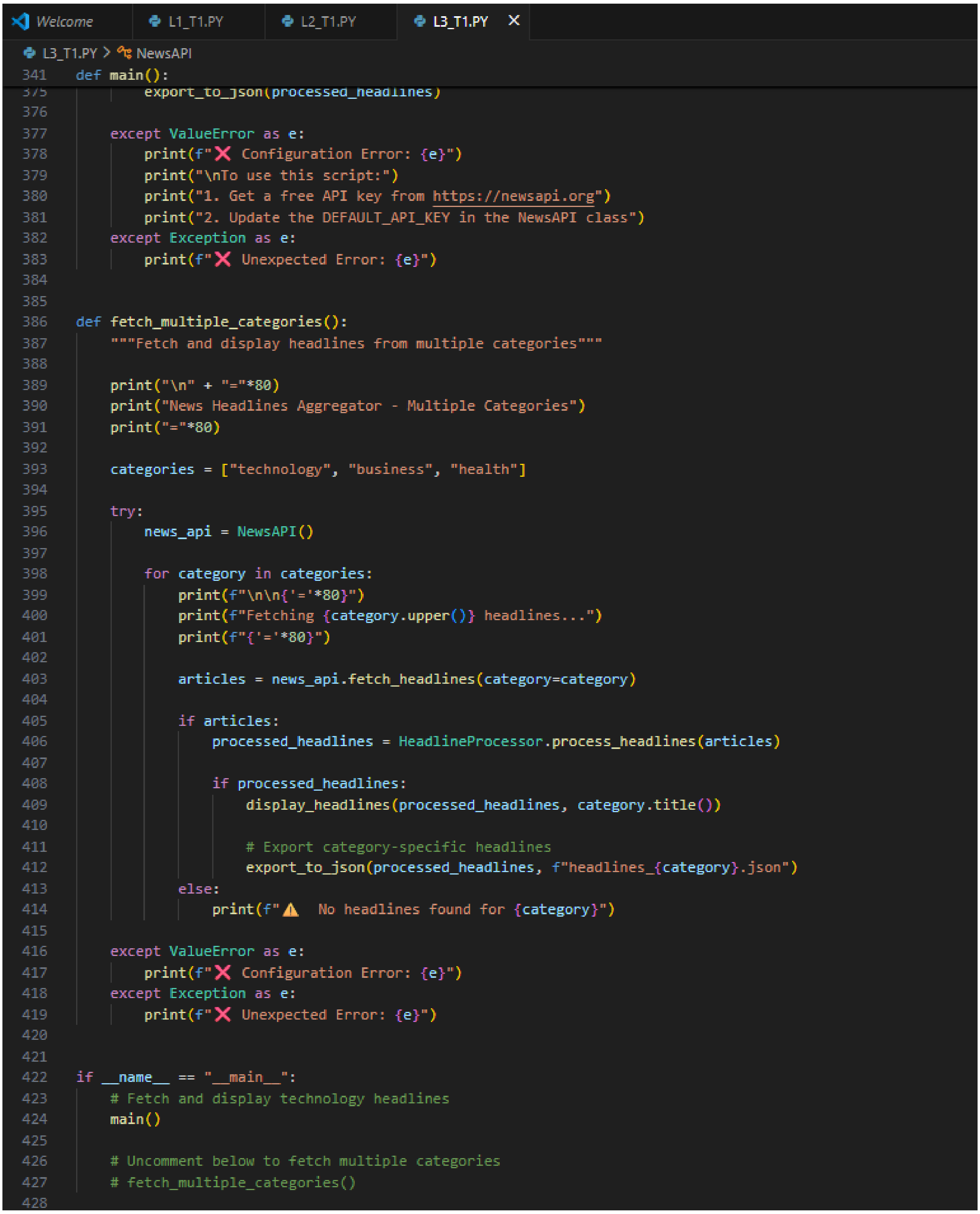
A news aggregator wants to display the latest technology news headlines using a news API. Sometimes, the API responds slowly or returns incomplete data.

- **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.

## Code:

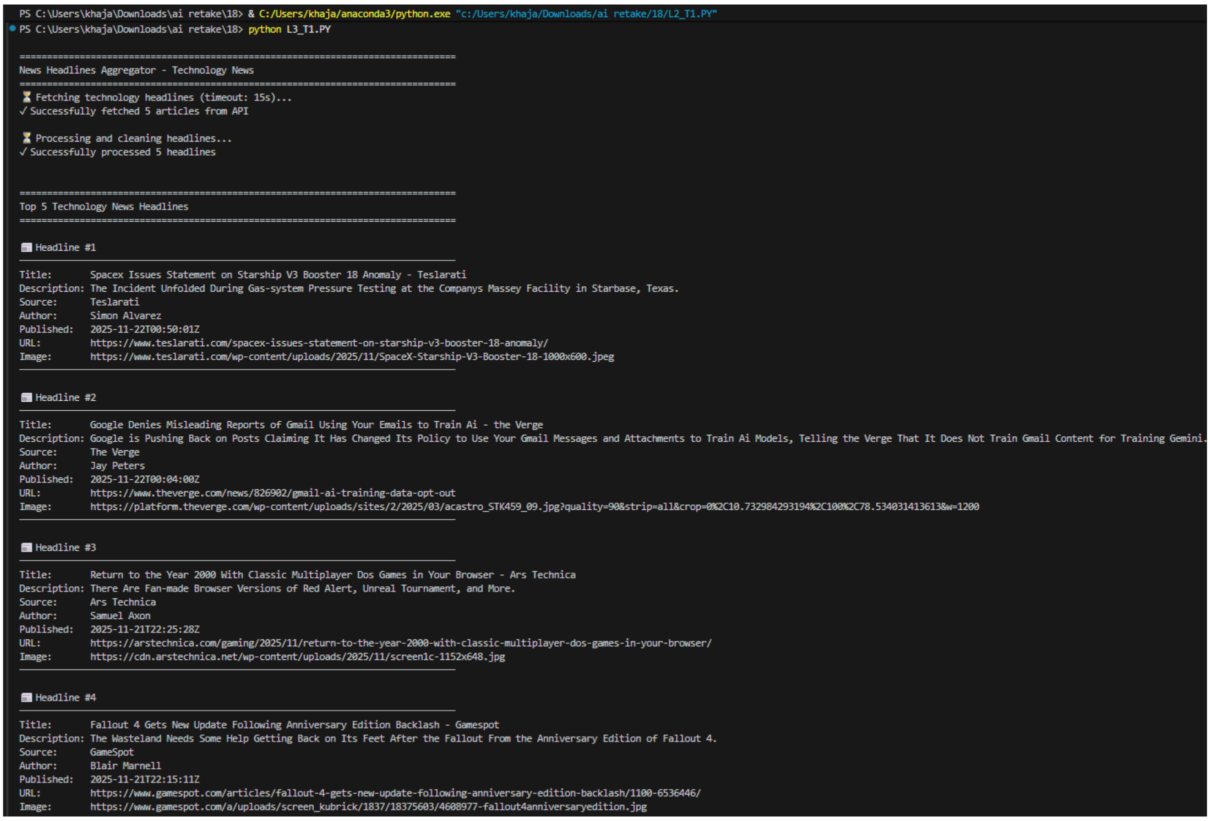


```python
"""
News Headlines API Script
Fetches and displays top technology news headlines with data cleaning
"""

import requests ···


class NewsAPI:
    """Fetches news headlines from NewsAPI with error handling"""

    # Using NewsAPI.org (free tier available)
    BASE_URL = "https://newsapi.org/v2/top-headlines"

    # Free API key for demonstration (replace with your own from newsapi.org)
    DEFAULT_API_KEY = "d047e31a76af47bd800b1e8ae3d031f8"

    TIMEOUT_SECONDS = 15  # Maximum request time
    MAX_HEADLINES = 5     # Number of headlines to fetch

    def __init__(self, api_key: Optional[str] = None):
        """Initialize the News API ···
        self.api_key = api_key or self.DEFAULT_API_KEY

        if not self.api_key or not isinstance(self.api_key, str):
            raise ValueError("API key must be a non-empty string")

    def _fetch_headlines_raw(self, category: str = "technology") -> Optional[Dict[str, Any]]: ···

    def fetch_headlines(self, category: str = "technology") -> Optional[List[Dict[str, Any]]]:
        """···
        try:
            data = self._fetch_headlines_raw(category)

            if not data or "articles" not in data: ···

            articles = data.get("articles", [])

            if not articles: ···

            return articles[:self.MAX_HEADLINES]

        except requests.exceptions.Timeout as e: ···
        except requests.exceptions.RequestException as e:
            print(f"❌ Network Error: {e}")
            return None
        except ValueError as e:
            print(f"❌ Data Error: {e}")
            return None
        except Exception as e:
            print(f"❌ Unexpected Error: {e}")
            return None


```



```python
        export_to_json(processed_headlines)

    except ValueError as e:
        print(f"❌ Configuration Error: {e}")
        print("\nTo use this script:")
        print("1. Get a free API key from https://newsapi.org")
        print("2. Update the DEFAULT_API_KEY in the NewsAPI class")
    except Exception as e:
        print(f"❌ Unexpected Error: {e}")


def fetch_multiple_categories():
    """Fetch and display headlines from multiple categories"""

    print("\n" + "="*80)
    print("News Headlines Aggregator - Multiple Categories")
    print("="*80)

    categories = ["technology", "business", "health"]

    try:
        news_api = NewsAPI()

        for category in categories:
            print(f"\n\n{'='*80}")
            print(f"Fetching {category.upper()} headlines...")
            print(f"{'='*80}")

            articles = news_api.fetch_headlines(category=category)

            if articles:
                processed_headlines = HeadlineProcessor.process_headlines(articles)

                if processed_headlines:
                    display_headlines(processed_headlines, category.title())

                    # Export category-specific headlines
                    export_to_json(processed_headlines, f"headlines_{category}.json")
            else:
                print(f"⚠️  No headlines found for {category}")

    except ValueError as e:
        print(f"❌ Configuration Error: {e}")
    except Exception as e:
        print(f"❌ Unexpected Error: {e}")


if __name__ == "__main__":
    # Fetch and display technology headlines
    main()

    # Uncomment below to fetch multiple categories
    # fetch_multiple_categories()

```

## Output:



```
PS C:\Users\khaja\Downloads\ai retake\18> & C:/Users/khaja/anaconda3/python.exe "c:/Users/khaja/Downloads/ai retake/18/L2_T1.PY"
● PS C:\Users\khaja\Downloads\ai retake\18> python L3_T1.PY

================================================================================
News Headlines Aggregator - Technology News
================================================================================
⏳ Fetching technology headlines (timeout: 15s)...
✓ Successfully fetched 5 articles from API

⏳ Processing and cleaning headlines...
✓ Successfully processed 5 headlines


================================================================================
Top 5 Technology News Headlines
================================================================================

📰 Headline #1
────────────────────────────────────────────────────────────────────────────────
Title:       Spacex Issues Statement on Starship V3 Booster 18 Anomaly - Teslarati
Description: The Incident Unfolded During Gas-system Pressure Testing at the Companys Massey Facility in Starbase, Texas.
Source:      Teslarati
Author:      Simon Alvarez
Published:   2025-11-22T00:50:01Z
URL:         https://www.teslarati.com/spacex-issues-statement-on-starship-v3-booster-18-anomaly/
Image:       https://www.teslarati.com/wp-content/uploads/2025/11/SpaceX-Starship-V3-Booster-18-1000x600.jpeg
────────────────────────────────────────────────────────────────────────────────

📰 Headline #2
────────────────────────────────────────────────────────────────────────────────
Title:       Google Denies Misleading Reports of Gmail Using Your Emails to Train Ai - the Verge
Description: Google is Pushing Back on Posts Claiming It Has Changed Its Policy to Use Your Gmail Messages and Attachments to Train Ai Models, Telling the Verge That It Does Not Train Gmail Content for Training Gemini.
Source:      The Verge
Author:      Jay Peters
Published:   2025-11-22T00:04:00Z
URL:         https://www.theverge.com/news/826902/gmail-ai-training-data-opt-out
Image:       https://platform.theverge.com/wp-content/uploads/sites/2/2025/03/acastro_STK459_09.jpg?quality=90&strip=all&crop=0%2C10.732984293194%2C100%2C78.534031413613&w=1200
────────────────────────────────────────────────────────────────────────────────

📰 Headline #3
────────────────────────────────────────────────────────────────────────────────
Title:       Return to the Year 2000 With Classic Multiplayer Dos Games in Your Browser - Ars Technica
Description: There Are Fan-made Browser Versions of Red Alert, Unreal Tournament, and More.
Source:      Ars Technica
Author:      Samuel Axon
Published:   2025-11-21T22:25:28Z
URL:         https://arstechnica.com/gaming/2025/11/return-to-the-year-2000-with-classic-multiplayer-dos-games-in-your-browser/
Image:       https://cdn.arstechnica.net/wp-content/uploads/2025/11/screen1c-1152x648.jpg
────────────────────────────────────────────────────────────────────────────────

📰 Headline #4
────────────────────────────────────────────────────────────────────────────────
Title:       Fallout 4 Gets New Update Following Anniversary Edition Backlash - Gamespot
Description: The Wasteland Needs Some Help Getting Back on Its Feet After the Fallout From the Anniversary Edition of Fallout 4.
Source:      GameSpot
Author:      Blair Marnell
Published:   2025-11-21T22:15:11Z
URL:         https://www.gamespot.com/articles/fallout-4-gets-new-update-following-anniversary-edition-backlash/1100-6536446/
Image:       https://www.gamespot.com/a/uploads/screen_kubrick/1837/18375603/4608977-fallout4anniversaryedition.jpg
```

**Observation:** The program successfully fetched and displayed the latest technology headlines from the News API with proper timeout and error handling. It also cleaned and formatted the headlines by removing special characters, converting text to title case, and ignoring null entries for a neat and readable output.