SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE	**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

Program Name: B. Tech	**Assignment Type: Lab**	**Academic Year:**2025-2026

Course Coordinator Name	Venkataramana Veeramsetty		
Instructor(s) Name	Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)		
Course Code	24CS002PC215	**Course Title**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week5 - Monday	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	
Assignment Number: 18.5(Present assignment number)/**24**(Total number of assignments)			

Q.No.	**Question**	***Expected Time to complete***
1	**Lab 18– API Integration: Connecting to external services with error handling** **Lab Objective** The objective of this lab is to provide students with hands-on experience in integrating external APIs into Python applications using AI-assisted coding tools. Students will: 1. Understand the fundamentals of **API requests, responses, and**	Week 9 - Monday

authentication mechanisms.

2. Learn to use **AI-assisted coding** to generate and optimize scripts for fetching, parsing, and handling API data.
3. Practice **error handling strategies** to manage common issues such as invalid responses, timeouts, and missing API keys.
4. Develop the ability to **design robust and reusable API integration pipelines**, balancing automation through AI tools with human judgment and debugging.

Lab Question 1: Weather Forecasting API

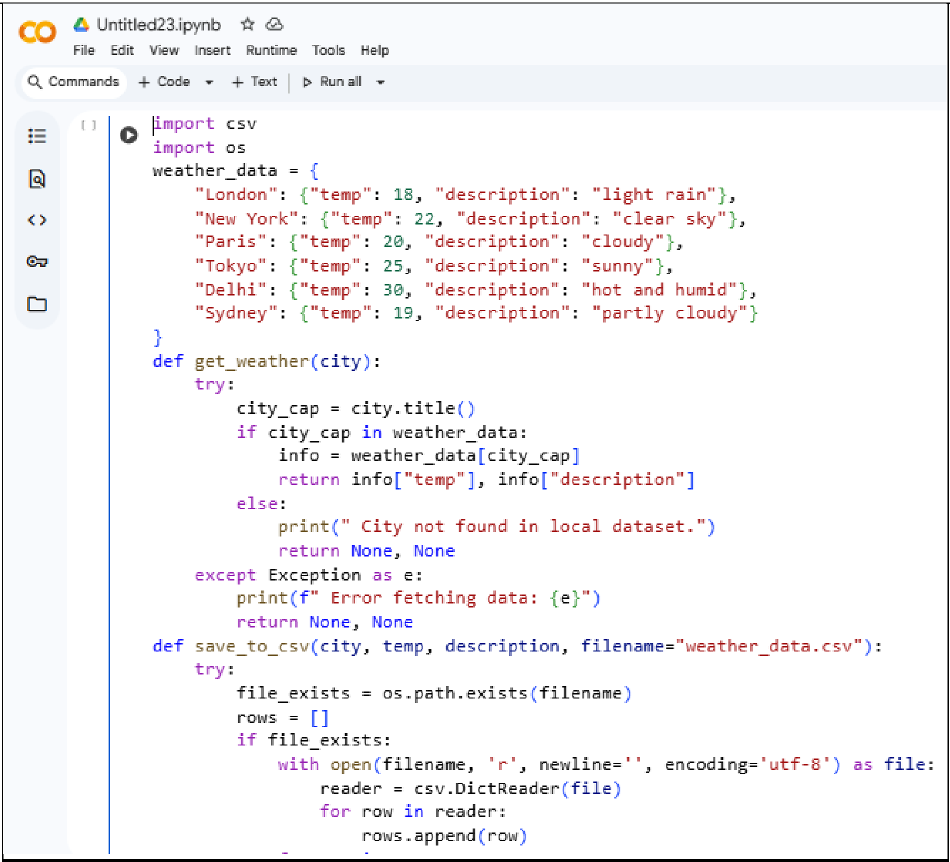
A travel company wants to show real-time weather updates for its customers. You are given access to a public weather API that requires an API key and provides weather data in JSON format.

- **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.
- **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.

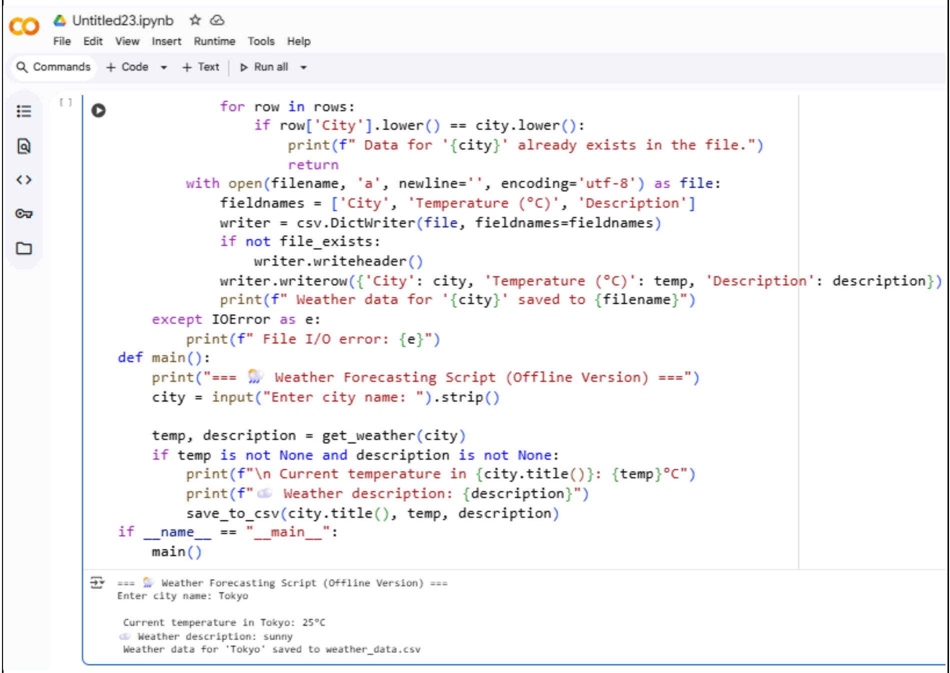
Prompt:

Write a Python program that gives weather info for a city
using hardcoded data.
Let the user enter a city name and show its temperature
and description.
If the city isn’t found, display a friendly error message.
Save the result to a CSV file, avoiding duplicate entries.
Handle any file errors so the program runs smoothly

Code:


Untitled23.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all

```python
import csv
import os
weather_data = {
    "London": {"temp": 18, "description": "light rain"},
    "New York": {"temp": 22, "description": "clear sky"},
    "Paris": {"temp": 20, "description": "cloudy"},
    "Tokyo": {"temp": 25, "description": "sunny"},
    "Delhi": {"temp": 30, "description": "hot and humid"},
    "Sydney": {"temp": 19, "description": "partly cloudy"}
}
def get_weather(city):
    try:
        city_cap = city.title()
        if city_cap in weather_data:
            info = weather_data[city_cap]
            return info["temp"], info["description"]
        else:
            print(" City not found in local dataset.")
            return None, None
    except Exception as e:
        print(f" Error fetching data: {e}")
        return None, None
def save_to_csv(city, temp, description, filename="weather_data.csv"):
    try:
        file_exists = os.path.exists(filename)
        rows = []
        if file_exists:
            with open(filename, 'r', newline='', encoding='utf-8') as file:
                reader = csv.DictReader(file)
                for row in reader:
                    rows.append(row)
```


Untitled23.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all

```python
            for row in rows:
                if row['City'].lower() == city.lower():
                    print(f" Data for '{city}' already exists in the file.")
                    return
        with open(filename, 'a', newline='', encoding='utf-8') as file:
            fieldnames = ['City', 'Temperature (°C)', 'Description']
            writer = csv.DictWriter(file, fieldnames=fieldnames)
            if not file_exists:
                writer.writeheader()
            writer.writerow({'City': city, 'Temperature (°C)': temp, 'Description': description})
            print(f" Weather data for '{city}' saved to {filename}")
    except IOError as e:
        print(f" File I/O error: {e}")
def main():
    print("=== 🌦 Weather Forecasting Script (Offline Version) ===")
    city = input("Enter city name: ").strip()

    temp, description = get_weather(city)
    if temp is not None and description is not None:
        print(f"\n Current temperature in {city.title()}: {temp}°C")
        print(f"☁ Weather description: {description}")
        save_to_csv(city.title(), temp, description)
if __name__ == "__main__":
    main()
```

```
=== 🌦 Weather Forecasting Script (Offline Version) ===
Enter city name: Tokyo

 Current temperature in Tokyo: 25°C
☁ Weather description: sunny
 Weather data for 'Tokyo' saved to weather_data.csv
```

Lab Question 2: Currency Exchange Rate API

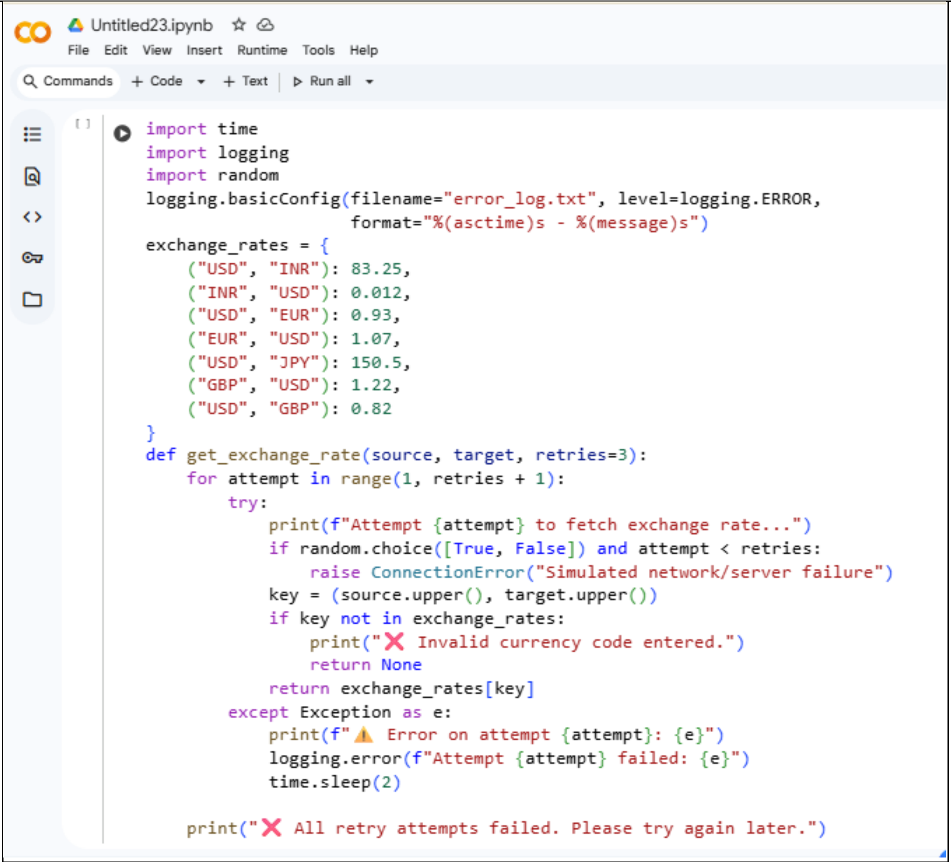
A financial startup needs a tool to convert amounts between currencies using an exchange rate API. However, the API occasionally fails due to server downtime.

- **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.

Prompt:

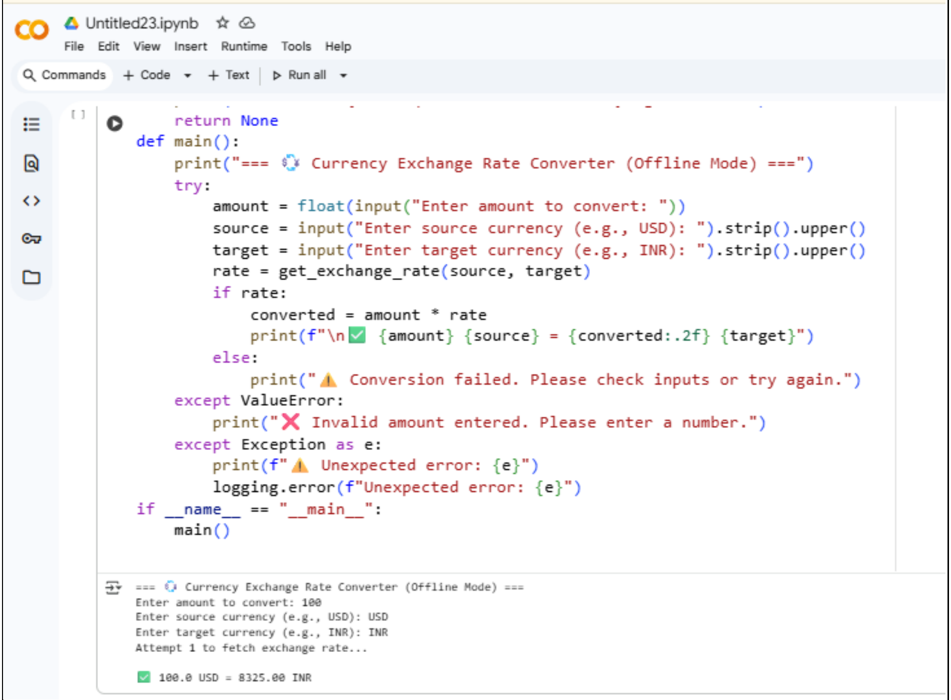
Write a Python program that simulates a currency converter without using any API. The user should enter an amount, the currency they're converting from, and the one they're converting to. Use hardcoded exchange rates to calculate and show the converted value. If a simulated network error happens, the program should retry up to three times. Any failed attempts should be logged in an error_log.txt file. Also, make sure the program handles invalid currency codes or incorrect inputs gracefully.

Code:



```python
import time
import logging
import random
logging.basicConfig(filename="error_log.txt", level=logging.ERROR,
                    format="%(asctime)s - %(message)s")
exchange_rates = {
    ("USD", "INR"): 83.25,
    ("INR", "USD"): 0.012,
    ("USD", "EUR"): 0.93,
    ("EUR", "USD"): 1.07,
    ("USD", "JPY"): 150.5,
    ("GBP", "USD"): 1.22,
    ("USD", "GBP"): 0.82
}
def get_exchange_rate(source, target, retries=3):
    for attempt in range(1, retries + 1):
        try:
            print(f"Attempt {attempt} to fetch exchange rate...")
            if random.choice([True, False]) and attempt < retries:
                raise ConnectionError("Simulated network/server failure")
            key = (source.upper(), target.upper())
            if key not in exchange_rates:
                print("❌ Invalid currency code entered.")
                return None
            return exchange_rates[key]
        except Exception as e:
            print(f"⚠️ Error on attempt {attempt}: {e}")
            logging.error(f"Attempt {attempt} failed: {e}")
            time.sleep(2)

    print("❌ All retry attempts failed. Please try again later.")
```



```python
    return None
def main():
    print("=== 🔄 Currency Exchange Rate Converter (Offline Mode) ===")
    try:
        amount = float(input("Enter amount to convert: "))
        source = input("Enter source currency (e.g., USD): ").strip().upper()
        target = input("Enter target currency (e.g., INR): ").strip().upper()
        rate = get_exchange_rate(source, target)
        if rate:
            converted = amount * rate
            print(f"\n✅ {amount} {source} = {converted:.2f} {target}")
        else:
            print("⚠️ Conversion failed. Please check inputs or try again.")
    except ValueError:
        print("❌ Invalid amount entered. Please enter a number.")
    except Exception as e:
        print(f"⚠️ Unexpected error: {e}")
        logging.error(f"Unexpected error: {e}")
if __name__ == "__main__":
    main()
```

```
=== 🔄 Currency Exchange Rate Converter (Offline Mode) ===
Enter amount to convert: 100
Enter source currency (e.g., USD): USD
Enter target currency (e.g., INR): INR
Attempt 1 to fetch exchange rate...

✅ 100.0 USD = 8325.00 INR
```

Lab Question 3: News Headlines API

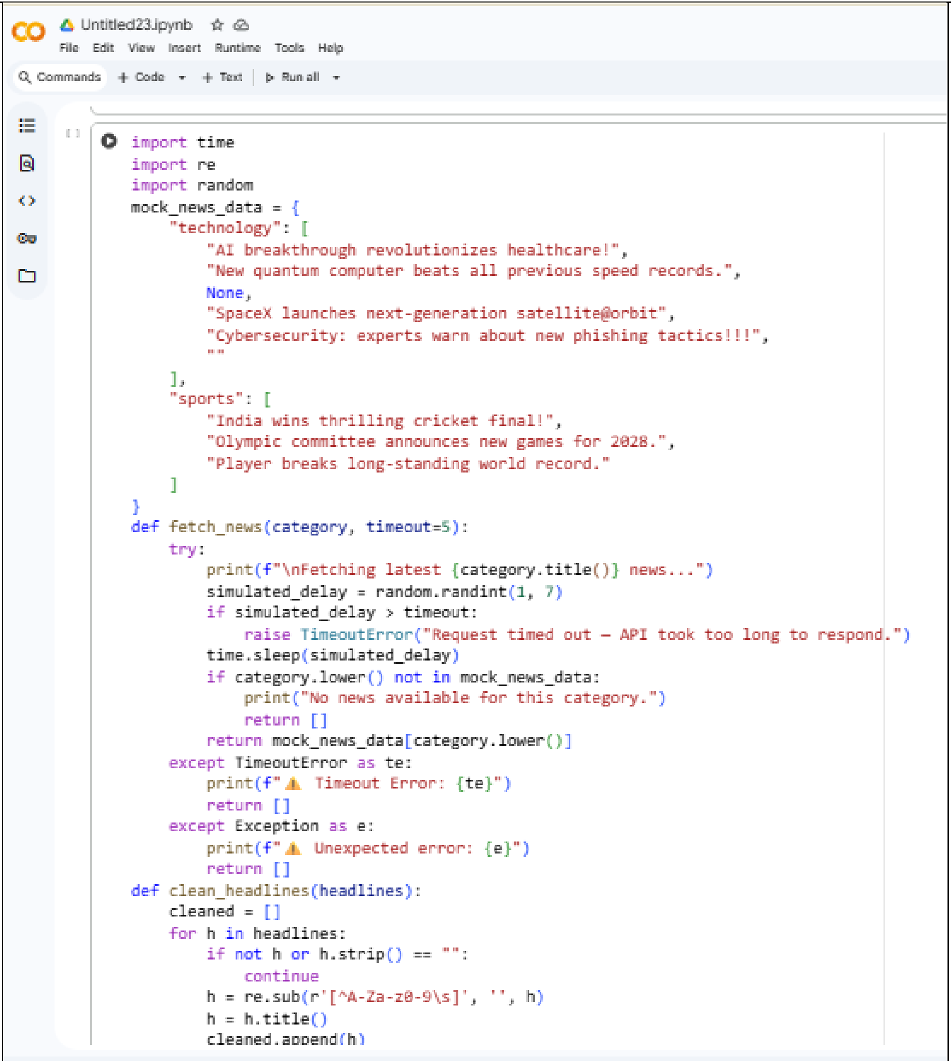
A news aggregator wants to display the latest technology news headlines using a news API. Sometimes, the API responds slowly or returns incomplete data.

- **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.

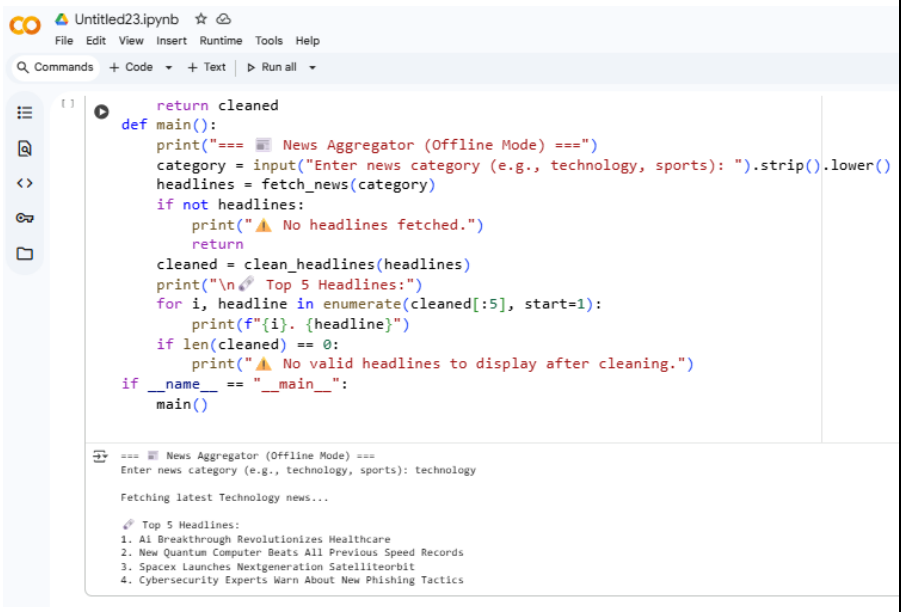
Prompt:

Write a Python program that acts like a simple news aggregator without using any API key. The user should be able to enter a news category like "technology" or "sports," and the program will pull from a set of hardcoded headlines stored in the code. It should simulate an API delay or timeout to mimic real-world behavior. Once it fetches the news, display the top 5 headlines in a clean, readable format — removing any special characters and converting the text to title case. Make sure the program handles missing or empty values gracefully so it doesn't crash.

Code:


Untitled23.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all

```python
import time
import re
import random
mock_news_data = {
    "technology": [
        "AI breakthrough revolutionizes healthcare!",
        "New quantum computer beats all previous speed records.",
        None,
        "SpaceX launches next-generation satellite@orbit",
        "Cybersecurity: experts warn about new phishing tactics!!!",
        ""
    ],
    "sports": [
        "India wins thrilling cricket final!",
        "Olympic committee announces new games for 2028.",
        "Player breaks long-standing world record."
    ]
}
def fetch_news(category, timeout=5):
    try:
        print(f"\nFetching latest {category.title()} news...")
        simulated_delay = random.randint(1, 7)
        if simulated_delay > timeout:
            raise TimeoutError("Request timed out – API took too long to respond.")
        time.sleep(simulated_delay)
        if category.lower() not in mock_news_data:
            print("No news available for this category.")
            return []
        return mock_news_data[category.lower()]
    except TimeoutError as te:
        print(f"⚠️ Timeout Error: {te}")
        return []
    except Exception as e:
        print(f"⚠️ Unexpected error: {e}")
        return []
def clean_headlines(headlines):
    cleaned = []
    for h in headlines:
        if not h or h.strip() == "":
            continue
        h = re.sub(r'[^A-Za-z0-9\s]', '', h)
        h = h.title()
        cleaned.append(h)
```


Untitled23.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all

```python
    return cleaned
def main():
    print("=== 📰 News Aggregator (Offline Mode) ===")
    category = input("Enter news category (e.g., technology, sports): ").strip().lower()
    headlines = fetch_news(category)
    if not headlines:
        print("⚠️ No headlines fetched.")
        return
    cleaned = clean_headlines(headlines)
    print("\n🗞️ Top 5 Headlines:")
    for i, headline in enumerate(cleaned[:5], start=1):
        print(f"{i}. {headline}")
    if len(cleaned) == 0:
        print("⚠️ No valid headlines to display after cleaning.")
if __name__ == "__main__":
    main()
```

```
=== 📰 News Aggregator (Offline Mode) ===
Enter news category (e.g., technology, sports): technology

Fetching latest Technology news...

🗞️ Top 5 Headlines:
1. Ai Breakthrough Revolutionizes Healthcare
2. New Quantum Computer Beats All Previous Speed Records
3. Spacex Launches Nextgeneration Satelliteorbit
4. Cybersecurity Experts Warn About New Phishing Tactics
```