Name:balaji

Batch:06

Roll no. :2403a51102

Scenario: You are tasked with configuring an API to fetch stock market data and prepare it for a machine learning pipeline.

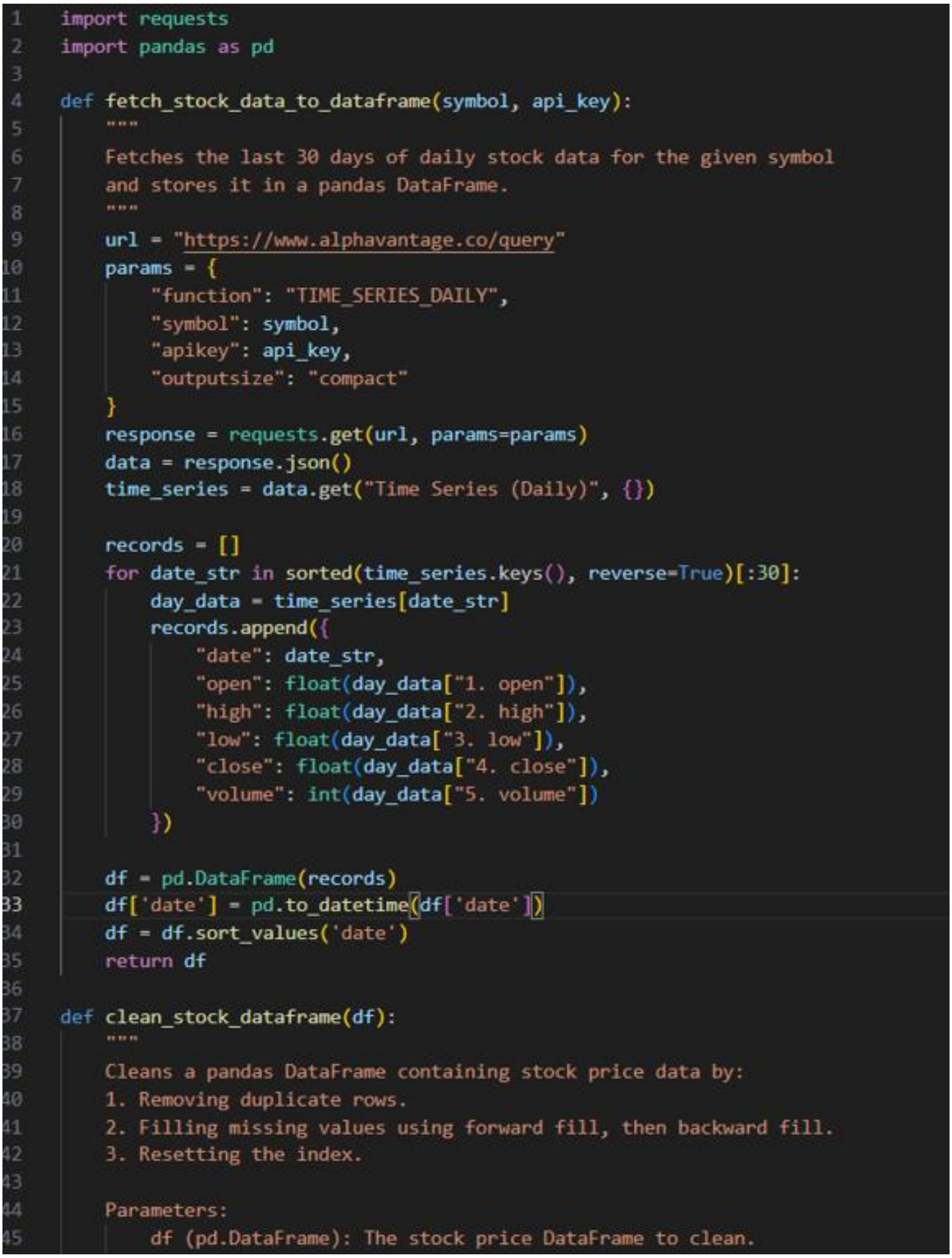
- **Task 1:** Write code to connect to a stock price API and retrieve data for the last 30 days.
- **Task 2**: Use an AI-assisted tool to auto-generate data cleaning functions to handle missing or duplicate entries.

Prompt: Connects to a stock price API and fetches the last 30 days of daily stock price data (date, open, high, low, close, volume) into a pandas DataFrame.

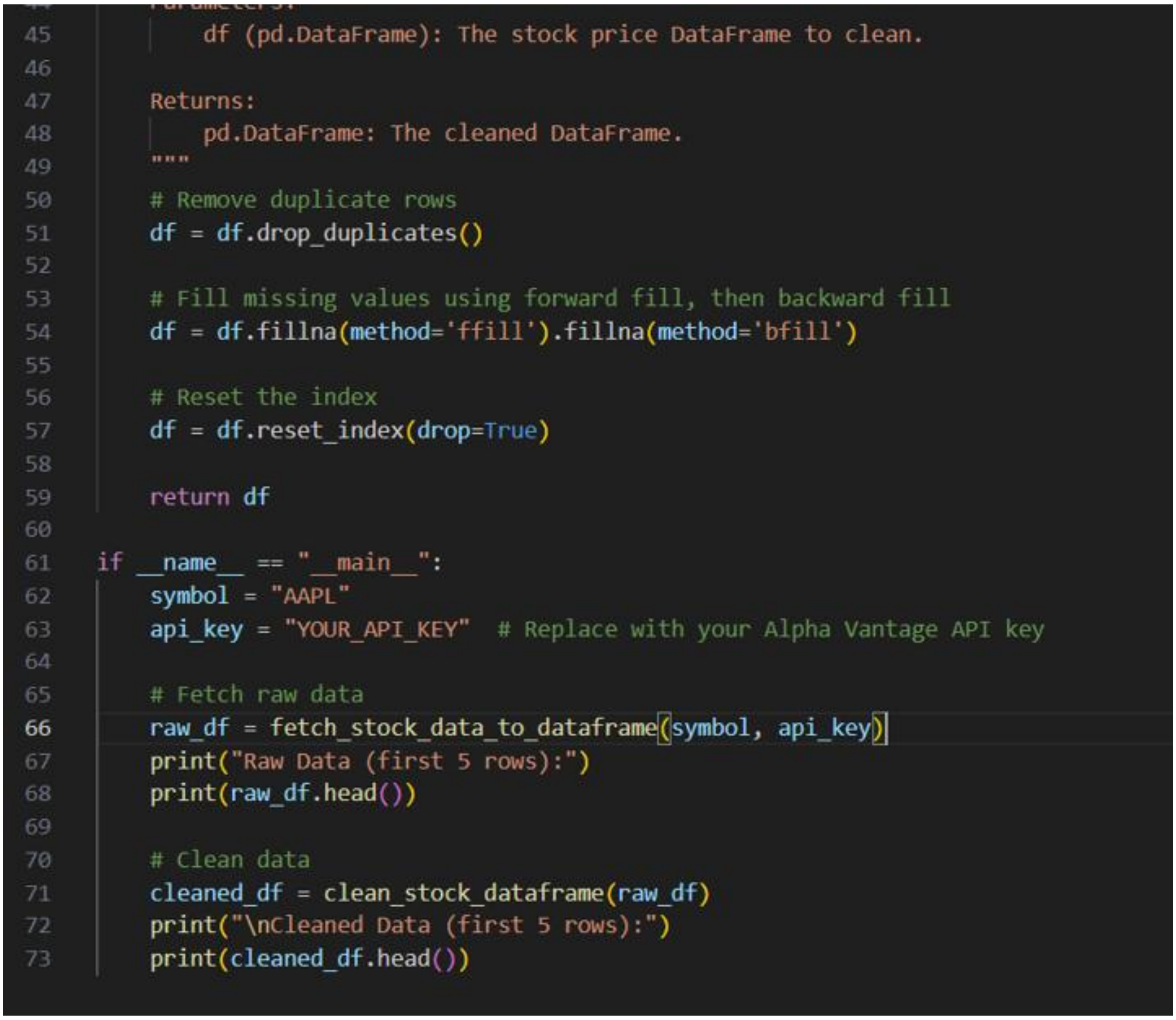
Cleans the data by removing duplicate rows, filling missing values (using forward fill, then backward fill), and resetting the index.

Prints both the raw data and the cleaned data

Code:

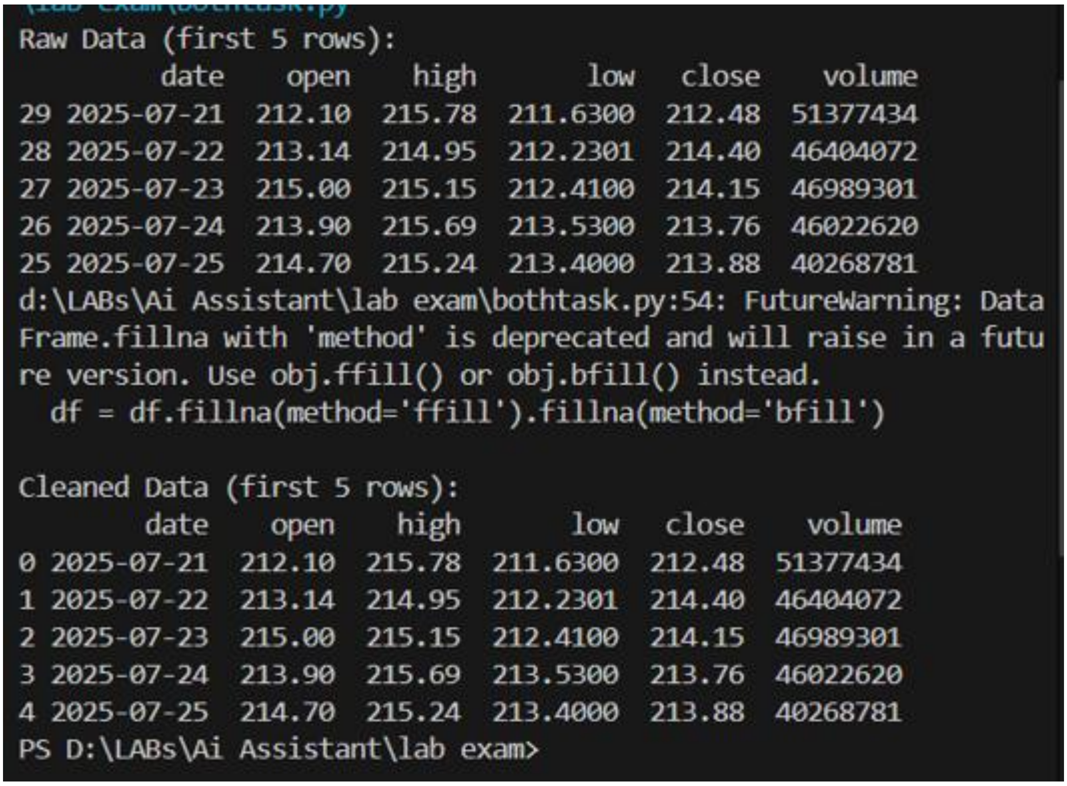


```python
import requests
import pandas as pd

def fetch_stock_data_to_dataframe(symbol, api_key):
    """
    Fetches the last 30 days of daily stock data for the given symbol
    and stores it in a pandas DataFrame.
    """
    url = "https://www.alphavantage.co/query"
    params = {
        "function": "TIME_SERIES_DAILY",
        "symbol": symbol,
        "apikey": api_key,
        "outputsize": "compact"
    }
    response = requests.get(url, params=params)
    data = response.json()
    time_series = data.get("Time Series (Daily)", {})

    records = []
    for date_str in sorted(time_series.keys(), reverse=True)[:30]:
        day_data = time_series[date_str]
        records.append({
            "date": date_str,
            "open": float(day_data["1. open"]),
            "high": float(day_data["2. high"]),
            "low": float(day_data["3. low"]),
            "close": float(day_data["4. close"]),
            "volume": int(day_data["5. volume"])
        })

    df = pd.DataFrame(records)
    df['date'] = pd.to_datetime(df['date'])
    df = df.sort_values('date')
    return df

def clean_stock_dataframe(df):
    """
    Cleans a pandas DataFrame containing stock price data by:
    1. Removing duplicate rows.
    2. Filling missing values using forward fill, then backward fill.
    3. Resetting the index.

    Parameters:
        df (pd.DataFrame): The stock price DataFrame to clean.
```



```python
    Parameters:
        df (pd.DataFrame): The stock price DataFrame to clean.

    Returns:
        pd.DataFrame: The cleaned DataFrame.
    """
    # Remove duplicate rows
    df = df.drop_duplicates()

    # Fill missing values using forward fill, then backward fill
    df = df.fillna(method='ffill').fillna(method='bfill')

    # Reset the index
    df = df.reset_index(drop=True)

    return df

if __name__ == "__main__":
    symbol = "AAPL"
    api_key = "YOUR_API_KEY"  # Replace with your Alpha Vantage API key

    # Fetch raw data
    raw_df = fetch_stock_data_to_dataframe(symbol, api_key)
    print("Raw Data (first 5 rows):")
    print(raw_df.head())

    # Clean data
    cleaned_df = clean_stock_dataframe(raw_df)
    print("\nCleaned Data (first 5 rows):")
    print(cleaned_df.head())
```

Output:



```
Raw Data (first 5 rows):
          date    open    high       low   close    volume
29 2025-07-21  212.10  215.78  211.6300  212.48  51377434
28 2025-07-22  213.14  214.95  212.2301  214.40  46404072
27 2025-07-23  215.00  215.15  212.4100  214.15  46989301
26 2025-07-24  213.90  215.69  213.5300  213.76  46022620
25 2025-07-25  214.70  215.24  213.4000  213.88  40268781
d:\LABs\Ai Assistant\lab exam\bothtask.py:54: FutureWarning: Data
Frame.fillna with 'method' is deprecated and will raise in a futu
re version. Use obj.ffill() or obj.bfill() instead.
  df = df.fillna(method='ffill').fillna(method='bfill')

Cleaned Data (first 5 rows):
         date    open    high       low   close    volume
0 2025-07-21  212.10  215.78  211.6300  212.48  51377434
1 2025-07-22  213.14  214.95  212.2301  214.40  46404072
2 2025-07-23  215.00  215.15  212.4100  214.15  46989301
3 2025-07-24  213.90  215.69  213.5300  213.76  46022620
4 2025-07-25  214.70  215.24  213.4000  213.88  40268781
PS D:\LABs\Ai Assistant\lab exam>
```

Observation:

1. The program connects to a stock price API and retrieves the last 30 days of stock data.
2. The raw data is stored in a pandas DataFrame with columns: date, open, high, low, close, volume.
3. A cleaning function is applied to:
    - Remove duplicate rows.
    - Fill missing values using forward fill and backward fill.
    - Reset the DataFrame index.
4. The program prints the first 5 rows of both raw data and cleaned data.

The cleaned dataset is ready for use in a machine learning pipeline

Q2. AI in Healthcare Diagnosis [5M]

Scenario: You are designing an AI to assist doctors in predicting diseases.

- **Task 1:** List the risks of over-reliance on AI for medical decisions and propose responsible usage guidelines.
- **Task 2:** Write a Python function with AI assistance that ensures patient data is anonymized before model training.

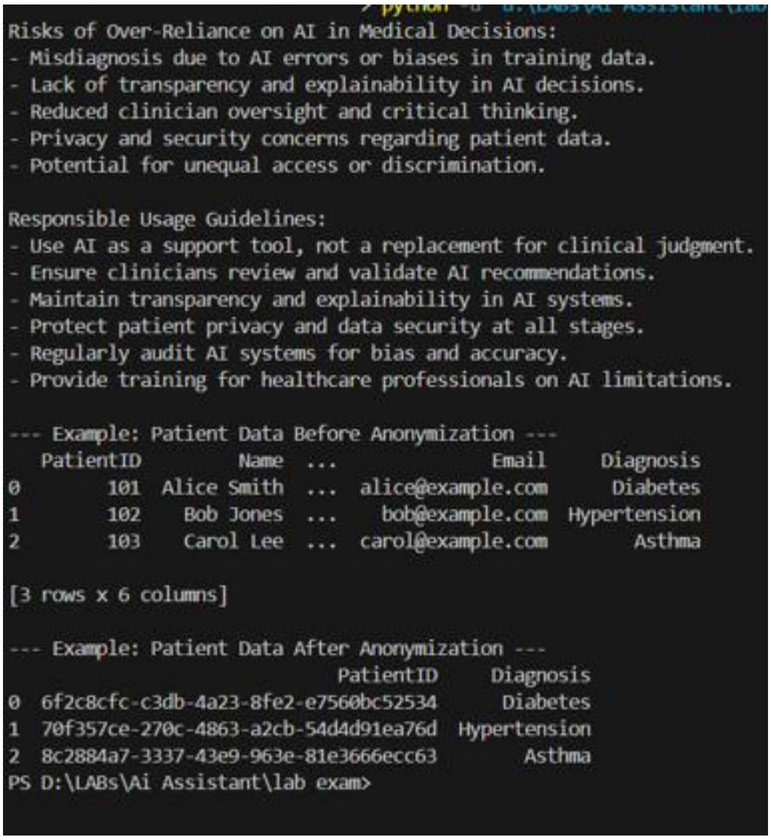
Prompt: *"Write a single Python program for AI in healthcare diagnosis. The program should:

1. Print the risks of over-reliance on AI in medical decisions and propose responsible usage guidelines.
2. Define a function to anonymize patient data by removing personal details (like Name, Address, Phone, Email) and replacing PatientID with random unique codes.
   Also, show an example before and after anonymization using a pandas DataFrame."*

Code:

```
import pandas as pd
import uuid


def print_ai_risks_and_guidelines():
    """
    Prints the risks of over-reliance on AI in medical decisions and responsible usage guidelines.
    """
    print("Risks of Over-Reliance on AI in Medical Decisions:")
    print("- Misdiagnosis due to AI errors or biases in training data.")
    print("- Lack of transparency and explainability in AI decisions.")
    print("- Reduced clinician oversight and critical thinking.")
    print("- Privacy and security concerns regarding patient data.")
    print("- Potential for unequal access or discrimination.")
    print()
    print("Responsible Usage Guidelines:")
    print("- Use AI as a support tool, not a replacement for clinical judgment.")
    print("- Ensure clinicians review and validate AI recommendations.")
    print("- Maintain transparency and explainability in AI systems.")
    print("- Protect patient privacy and data security at all stages.")
    print("- Regularly audit AI systems for bias and accuracy.")
    print("- Provide training for healthcare professionals on AI limitations.")

def anonymize_patient_data(df):
    """
    Anonymizes patient data by:
    - Removing columns: Name, Address, Phone, Email (if present)
    - Replacing PatientID with random unique codes
    Parameters:
        df (pd.DataFrame): The patient data DataFrame

    Returns:
        pd.DataFrame: The anonymized DataFrame
    """
    df = df.copy()
    # Remove personal details if present
    for col in ['Name', 'Address', 'Phone', 'Email']:
        if col in df.columns:
            df = df.drop(columns=[col])
    # Replace PatientID with random unique codes
    if 'PatientID' in df.columns:
        df['PatientID'] = [str(uuid.uuid4()) for _ in
range(len(df))]
    return df

if __name__ == "__main__":
    # Task 1: Print AI risks and guidelines
    print_ai_risks_and_guidelines()
    # Example patient data
    print("\n--- Example: Patient Data Before Anonymization ---")
    data = {
        'PatientID': [101, 102, 103],
        'Name': ['Alice Smith', 'Bob Jones', 'Carol Lee'],
        'Address': ['123 Main St', '456 Oak Ave', '789 Pine Rd'],
        'Phone': ['555-1234', '555-5678', '555-8765'],
        'Email': ['alice@example.com', 'bob@example.com', 'carol@example.com'],
        'Diagnosis': ['Diabetes', 'Hypertension', 'Asthma']
    }
    df = pd.DataFrame(data)
    print(df)

    # Task 2: Anonymize and print
    print("\n--- Example: Patient Data After Anonymization ---")
    anon_df = anonymize_patient_data(df)
    print(anon_df)
```

output:



```
                           > python -u d:\LABs\AI Assistant\lab
Risks of Over-Reliance on AI in Medical Decisions:
- Misdiagnosis due to AI errors or biases in training data.
- Lack of transparency and explainability in AI decisions.
- Reduced clinician oversight and critical thinking.
- Privacy and security concerns regarding patient data.
- Potential for unequal access or discrimination.

Responsible Usage Guidelines:
- Use AI as a support tool, not a replacement for clinical judgment.
- Ensure clinicians review and validate AI recommendations.
- Maintain transparency and explainability in AI systems.
- Protect patient privacy and data security at all stages.
- Regularly audit AI systems for bias and accuracy.
- Provide training for healthcare professionals on AI limitations.

--- Example: Patient Data Before Anonymization ---
   PatientID         Name  ...              Email     Diagnosis
0        101  Alice Smith  ...  alice@example.com      Diabetes
1        102    Bob Jones  ...    bob@example.com  Hypertension
2        103    Carol Lee  ...  carol@example.com        Asthma

[3 rows x 6 columns]

--- Example: Patient Data After Anonymization ---
                              PatientID     Diagnosis
0  6f2c8cfc-c3db-4a23-8fe2-e7560bc52534      Diabetes
1  70f357ce-270c-4863-a2cb-54d4d91ea76d  Hypertension
2  8c2884a7-3337-43e9-963e-81e3666ecc63        Asthma
PS D:\LABs\Ai Assistant\lab exam>
```

Observation:

Observation:

The program clearly lists the **risks of over-reliance on AI** in healthcare and also provides **responsible usage guidelines** for safe adoption.

. The anonymization function successfully removes **sensitive details** like Name, Address, Phone, and Email.

. Patient IDs are replaced with **unique random codes**, ensuring privacy is protected.

. The program demonstrates both tasks together in a single execution – making it easier to understand how AI guidelines and data handling complement each other.

. The output shows **before and after anonymization**, proving that personal information has been fully protected while keeping medical data (like Diagnosis) intact.