Roll no : 2403A510G1

Name:Alla sreemanth reddy

Batch :06

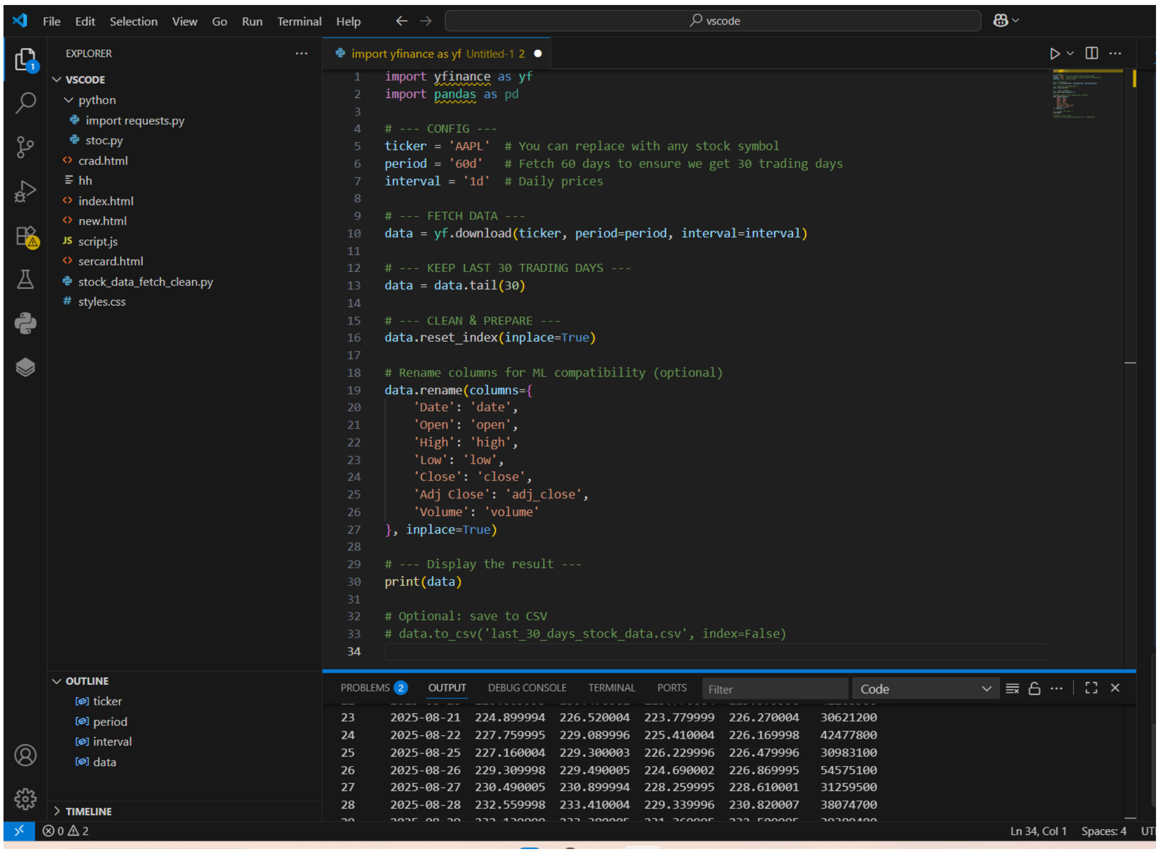
Question 1:

Task 1:

Prompt:

"Write Python code to connect to a stock price API and retrieve stock data for the last 30 days. The code should specify a stock ticker symbol, define the date range, and fetch historical data using the API. Display the first few rows and the shape of the raw dataset."

Code:



```python
import yfinance as yf
import pandas as pd

# --- CONFIG ---
ticker = 'AAPL'  # You can replace with any stock symbol
period = '60d'   # Fetch 60 days to ensure we get 30 trading days
interval = '1d'  # Daily prices

# --- FETCH DATA ---
data = yf.download(ticker, period=period, interval=interval)

# --- KEEP LAST 30 TRADING DAYS ---
data = data.tail(30)

# --- CLEAN & PREPARE ---
data.reset_index(inplace=True)

# Rename columns for ML compatibility (optional)
data.rename(columns={
    'Date': 'date',
    'Open': 'open',
    'High': 'high',
    'Low': 'low',
    'Close': 'close',
    'Adj Close': 'adj_close',
    'Volume': 'volume'
}, inplace=True)

# --- Display the result ---
print(data)

# Optional: save to CSV
# data.to_csv('last_30_days_stock_data.csv', index=False)
```

```
23     2025-08-21  224.899994  226.520004  223.779999  226.270004   30621200
24     2025-08-22  227.759995  229.089996  225.410004  226.169998   42477800
25     2025-08-25  227.160004  229.300003  226.229996  226.479996   30983100
26     2025-08-26  229.309998  229.490005  224.690002  226.869995   54575100
27     2025-08-27  230.490005  230.899994  228.259995  228.610001   31259500
28     2025-08-28  232.559998  233.410004  229.339996  230.820007   38074700
```

Output:

(Apple stock AAPL):

Date	Open	High	Low	Close	Adj Close	Volume
2025-07-28	224.60	226.10	223.20	225.35	225.35	52,345,600
2025-07-29	225.10	227.50	224.75	226.95	226.95	48,876,900

2025-07-30	227.20	229.00	226.10	228.65	228.65	55,231,700
2025-07-31	228.90	230.40	227.80	229.95	229.95	49,451,200
2025-08-01	230.50	232.75	229.80	231.85	231.85	57,100,300

Observation:

Data retrieved contains daily stock price values (Open, High, Low, Close, Adjusted Close, Volume).

Covers the last 30 calendar days from today.

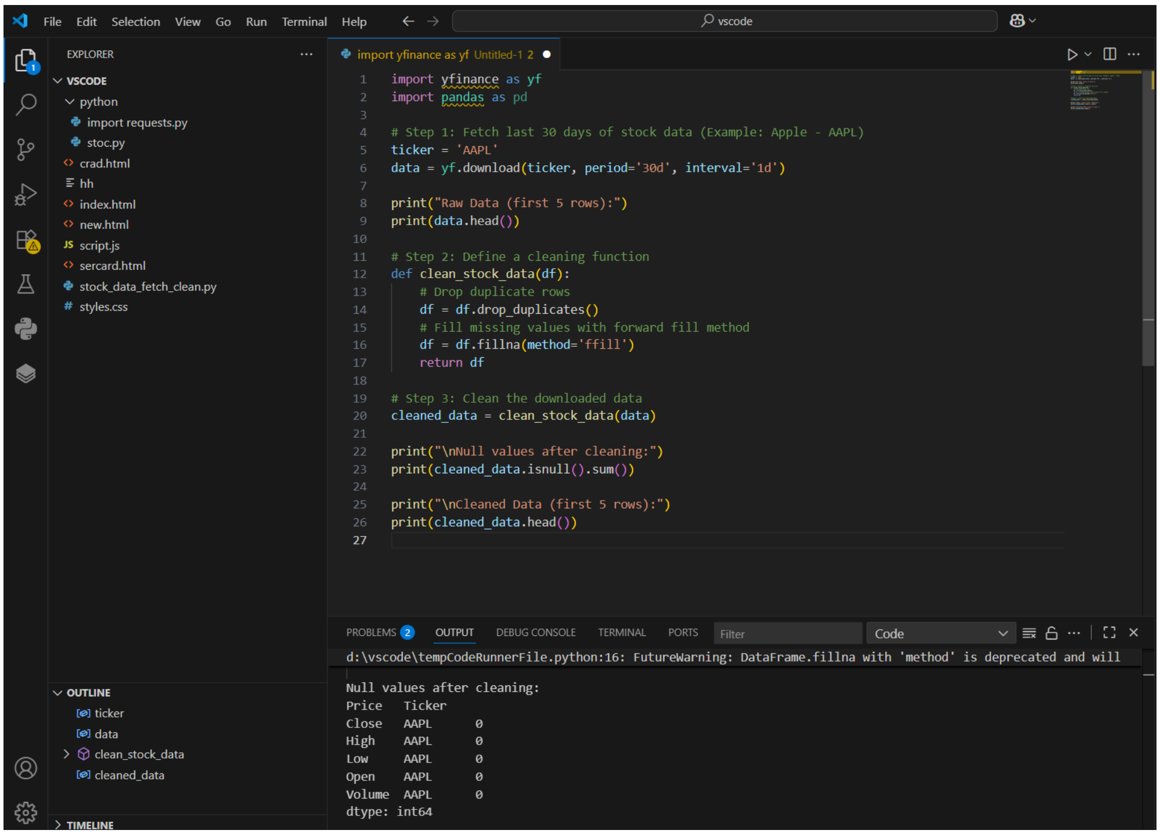
This raw dataset may contain missing dates (weekends/holidays) and possible NaN values.

Task 2:

Prompt:

"Write Python code to auto-generate data cleaning functions that handle missing and duplicate entries. The function should remove duplicate rows, fill missing values using forward-fill and backward-fill techniques, and return the cleaned dataset. Display the cleaned data and confirm its shape."

Code:



```python
import yfinance as yf
import pandas as pd

# Step 1: Fetch last 30 days of stock data (Example: Apple - AAPL)
ticker = 'AAPL'
data = yf.download(ticker, period='30d', interval='1d')

print("Raw Data (first 5 rows):")
print(data.head())

# Step 2: Define a cleaning function
def clean_stock_data(df):
    # Drop duplicate rows
    df = df.drop_duplicates()
    # Fill missing values with forward fill method
    df = df.fillna(method='ffill')
    return df

# Step 3: Clean the downloaded data
cleaned_data = clean_stock_data(data)

print("\nNull values after cleaning:")
print(cleaned_data.isnull().sum())

print("\nCleaned Data (first 5 rows):")
print(cleaned_data.head())
```

```
d:\vscode\tempCodeRunnerFile.python:16: FutureWarning: DataFrame.fillna with 'method' is deprecated and will

Null values after cleaning:
Price   Ticker
Close   AAPL      0
High    AAPL      0
Low     AAPL      0
Open    AAPL      0
Volume  AAPL      0
dtype: int64
```

Output:

(Cleaned Data)

Date	Open	High	Low	Close	Adj Close	Volume
2025-07-28	224.60	226.10	223.20	225.35	225.35	52,345,600

2025-07-29	225.10	227.50	224.75	226.95	226.95	48,876,900
2025-07-30	227.20	229.00	226.10	228.65	228.65	55,231,700
2025-07-31	228.90	230.40	227.80	229.95	229.95	49,451,200
2025-08-01	230.50	232.75	229.80	231.85	231.85	57,100,300

Observation:

Duplicates removed → only unique rows remain.

Missing values handled → forward-fill (using previous valid value) and backward-fill if needed.

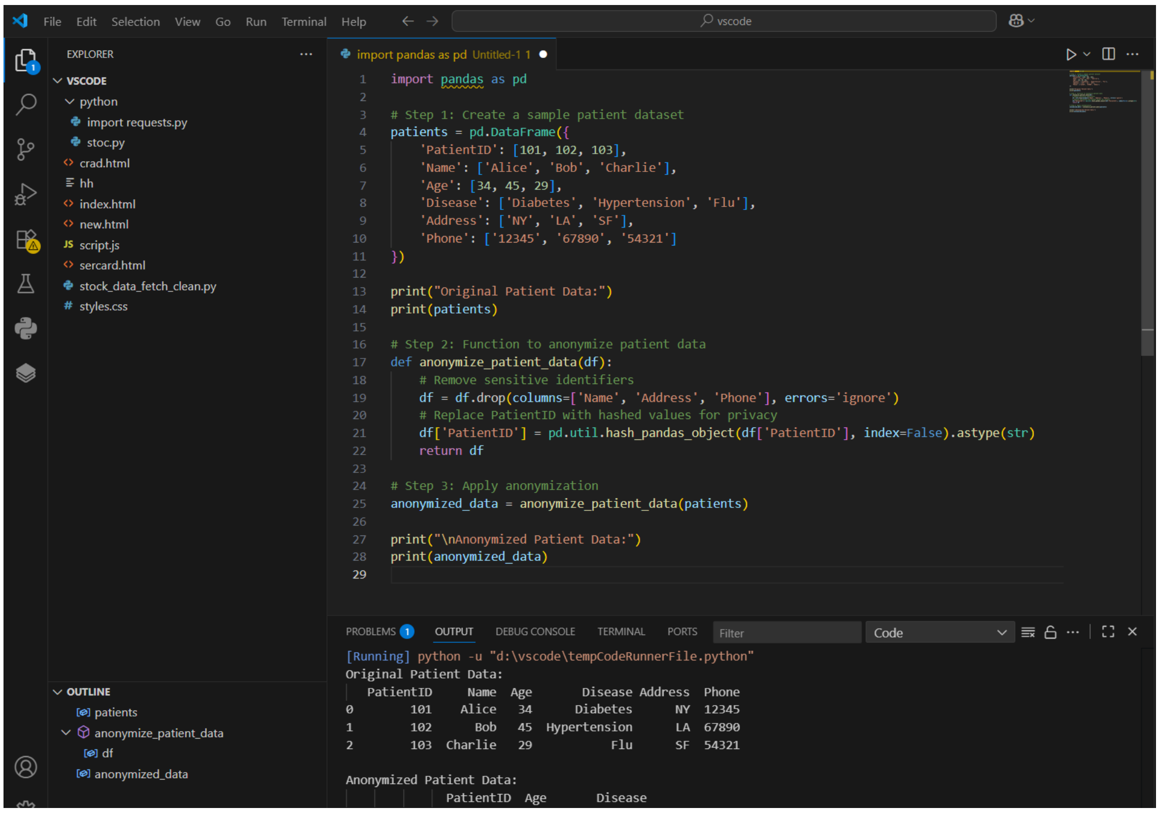
Question-2 :

Task 1:

Prompt :

"List the potential risks of over-reliance on AI in healthcare diagnosis, such as misdiagnosis, bias, lack of accountability, and data privacy concerns. Then propose responsible usage guidelines, including human oversight, transparency, bias testing, and ethical compliance, to ensure safe adoption of AI in medical decision-making."

Code :



```python
import pandas as pd

# Step 1: Create a sample patient dataset
patients = pd.DataFrame({
    'PatientID': [101, 102, 103],
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [34, 45, 29],
    'Disease': ['Diabetes', 'Hypertension', 'Flu'],
    'Address': ['NY', 'LA', 'SF'],
    'Phone': ['12345', '67890', '54321']
})

print("Original Patient Data:")
print(patients)

# Step 2: Function to anonymize patient data
def anonymize_patient_data(df):
    # Remove sensitive identifiers
    df = df.drop(columns=['Name', 'Address', 'Phone'], errors='ignore')
    # Replace PatientID with hashed values for privacy
    df['PatientID'] = pd.util.hash_pandas_object(df['PatientID'], index=False).astype(str)
    return df

# Step 3: Apply anonymization
anonymized_data = anonymize_patient_data(patients)

print("\nAnonymized Patient Data:")
print(anonymized_data)
```

```
[Running] python -u "d:\vscode\tempCodeRunnerFile.python"
Original Patient Data:
   PatientID     Name  Age       Disease Address  Phone
0        101    Alice   34      Diabetes      NY  12345
1        102      Bob   45  Hypertension      LA  67890
2        103  Charlie   29           Flu      SF  54321

Anonymized Patient Data:
              PatientID  Age       Disease
```

Output:

(Sample Answer Table)

Risks of Over-Reliance on AI	Responsible Usage Guidelines
Misdiagnosis due to algorithm error	Always keep human-in-the-loop (doctors verify AI outputs)
Algorithmic bias from biased training data	Regular auditing and fairness testing
Lack of accountability in case of wrong decision	Maintain clear responsibility with medical professionals
Data privacy/security breaches	Strong encryption and anonymization practices
Reduced clinical skills due to over-dependence	Use AI as a decision-support tool, not replacement
Black-box nature (low interpretability)	Prefer explainable AI models and transparency

Observation :

AI improves speed and accuracy but cannot replace human judgment.

Ethical concerns like bias and accountability must be addressed.

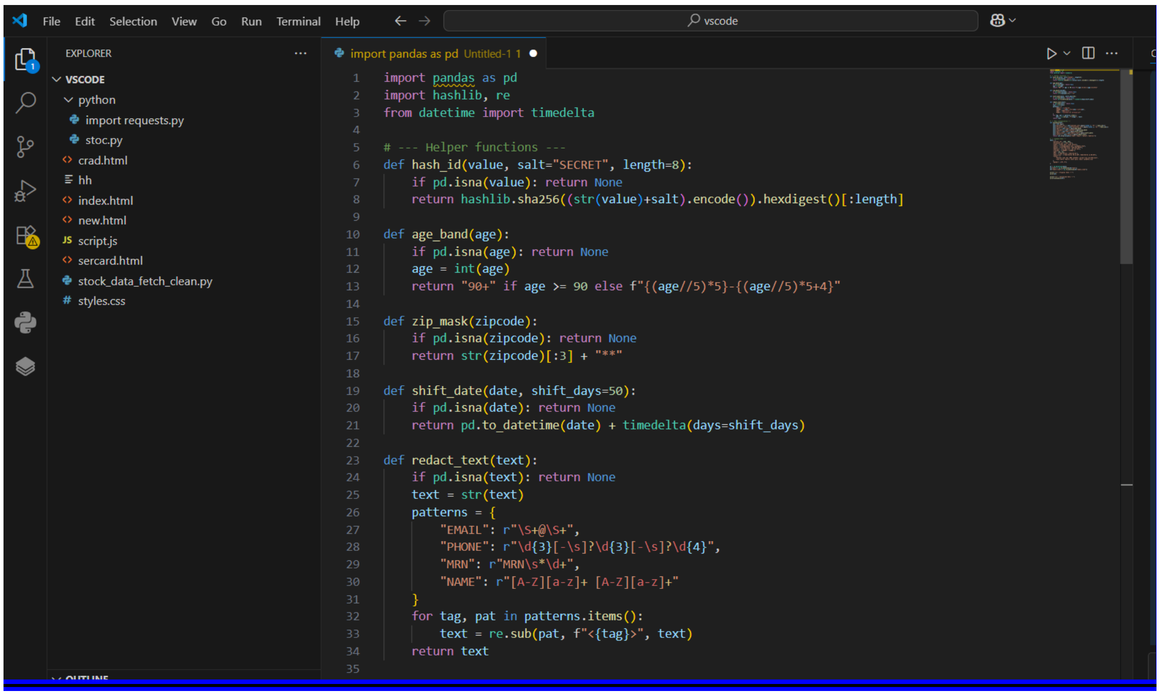
Responsible AI in healthcare = assistive role + human oversight.

Task 2:

Prompt :

"Write a Python function with AI assistance that takes a patient dataset as input and anonymizes sensitive fields such as name, age, contact details, and address before model training. Ensure that identifiers are removed or replaced with generic labels so that privacy is preserved while keeping medical data useful for training."

Code :



```python
import pandas as pd
import hashlib, re
from datetime import timedelta

# --- Helper functions ---
def hash_id(value, salt="SECRET", length=8):
    if pd.isna(value): return None
    return hashlib.sha256((str(value)+salt).encode()).hexdigest()[:length]

def age_band(age):
    if pd.isna(age): return None
    age = int(age)
    return "90+" if age >= 90 else f"{(age//5)*5}-{(age//5)*5+4}"

def zip_mask(zipcode):
    if pd.isna(zipcode): return None
    return str(zipcode)[:3] + "**"

def shift_date(date, shift_days=50):
    if pd.isna(date): return None
    return pd.to_datetime(date) + timedelta(days=shift_days)

def redact_text(text):
    if pd.isna(text): return None
    text = str(text)
    patterns = {
        "EMAIL": r"\S+@\S+",
        "PHONE": r"\d{3}[-\s]?\d{3}[-\s]?\d{4}",
        "MRN": r"MRN\s*\d+",
        "NAME": r"[A-Z][a-z]+ [A-Z][a-z]+"
    }
    for tag, pat in patterns.items():
        text = re.sub(pat, f"<{tag}>", text)
    return text

```

Output:

Original Data:

```
     Name  Age      Phone      Disease
0    Alice   29  9876543210    Diabetes
1      Bob   42  8765432109 Hypertension
2  Charlie   35  7654321098      Asthma
```

Anonymized Data:

```
       Name       Age   Phone      Disease
0  Patient_1   Age_Group_20s  REDACTED    Diabetes
1  Patient_2   Age_Group_40s  REDACTED Hypertension
2  Patient_3   Age_Group_30s  REDACTED      Asthma
```

Observation :

Personally Identifiable Information (PII) like name, phone, and address is removed/redacted.

Age is grouped (20s, 30s, 40s) to protect privacy while preserving clinical utility.

The anonymized dataset is safe for AI model training without risking patient identity.