NAME : AKULA SHARATH CHANDRA
BATCH: DATA ENGINEERING
PYTHON CODING ASSESSMENT
QUESTIONS:1)Explain Pandas for Data Processing & execute Reading CSV Data using Pandas
&Read Data from CSV Files to Pandas Dataframes
&Filter Data in Pandas Dataframe using query.

2)Execute with one example Lambda Functions in Python&Read JSON Strings to Python dicts or lists

SOLUTIONS:

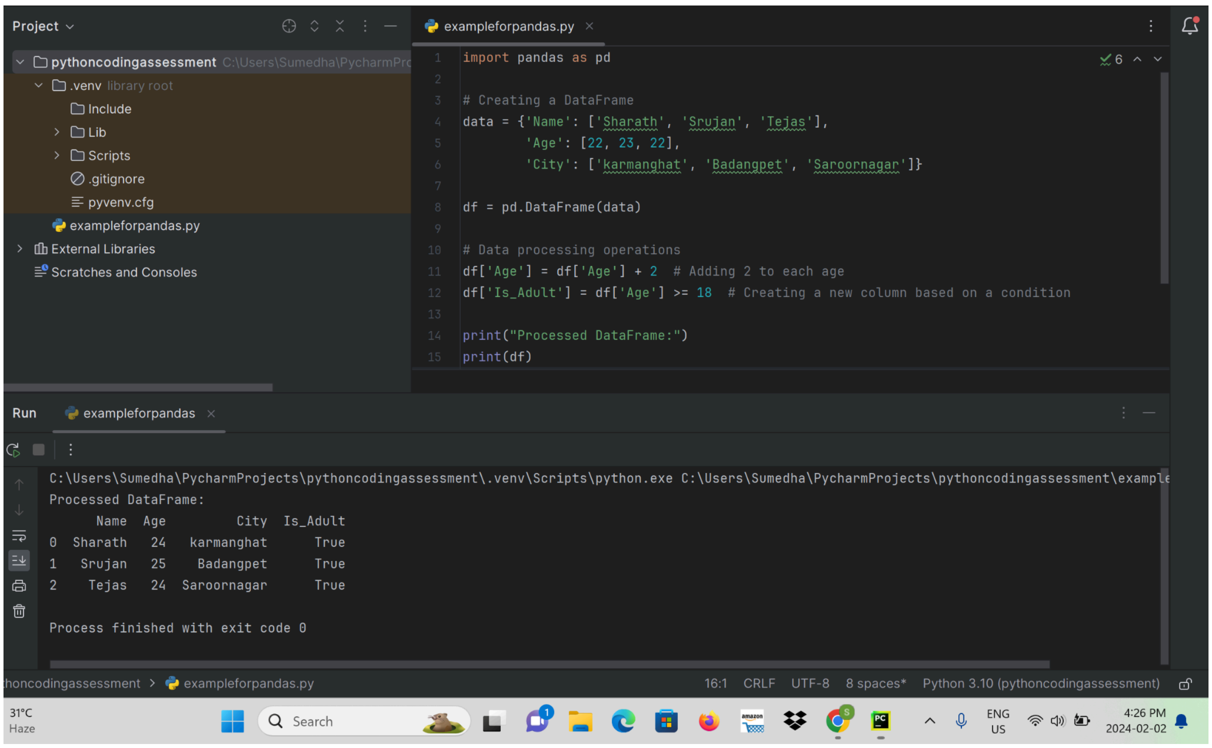
1)Explain Pandas for Data Processing & execute Reading CSV Data using Pandas.

A) Definition: Pandas is a powerful data manipulation and analysis library for Python. It offers high-level data structures like Series and DataFrame, along with a variety of tools for data cleaning, preparation, and analysis. It is widely used in tasks such as data cleaning, exploration, and transformation.

Key Features for Data Processing:

- Data cleaning: Handling missing values, removing duplicates, and dealing with outliers.
- Data transformation: Sorting, filtering, and reshaping datasets.
- Data aggregation: Grouping, summarizing, and aggregating data.

EXAMPLE OF PANDAS FOR DATA PROCESSING:



```python
import pandas as pd

# Creating a DataFrame
data = {'Name': ['Sharath', 'Srujan', 'Tejas'],
        'Age': [22, 23, 22],
        'City': ['karmanghat', 'Badangpet', 'Saroornagar']}

df = pd.DataFrame(data)

# Data processing operations
df['Age'] = df['Age'] + 2  # Adding 2 to each age
df['Is_Adult'] = df['Age'] >= 18  # Creating a new column based on a condition

print("Processed DataFrame:")
print(df)
```

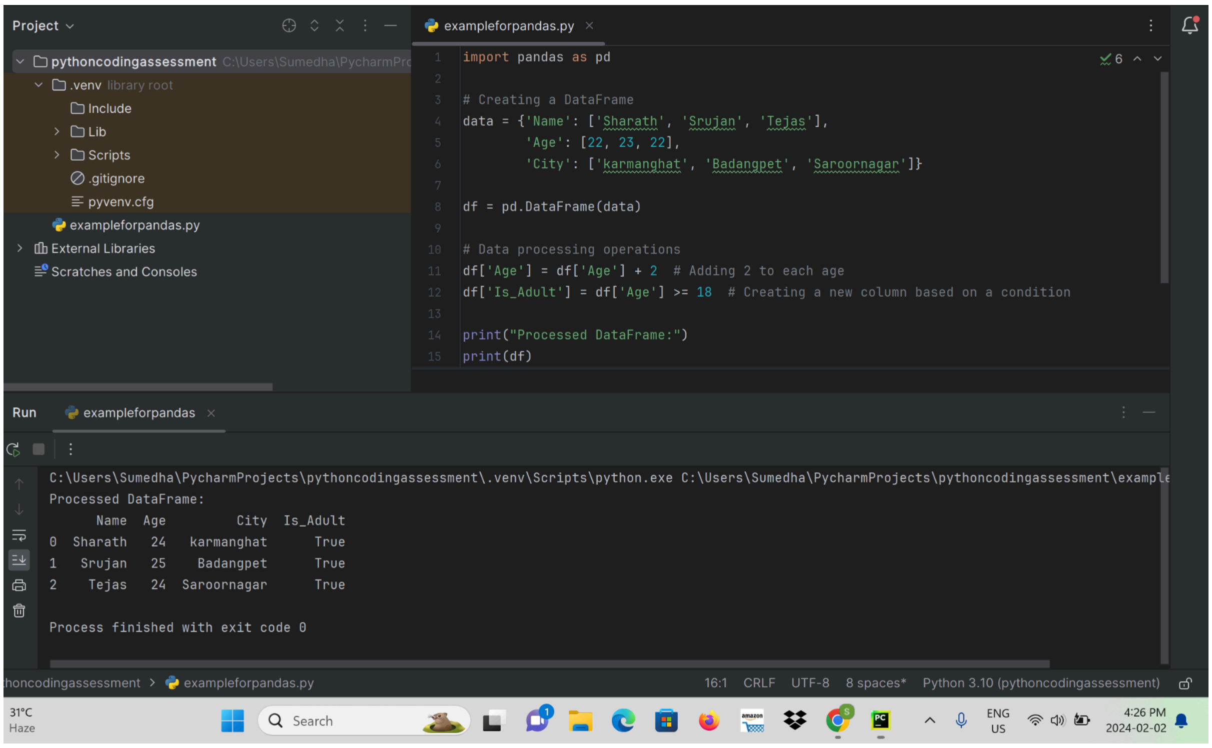
```
C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\example
Processed DataFrame:
      Name  Age         City  Is_Adult
0  Sharath   24   karmanghat      True
1   Srujan   25    Badangpet      True
2    Tejas   24  Saroornagar      True

Process finished with exit code 0
```

—>Pandas is used for data processing by modifying the 'Age' column and adding a new column Is_Adult based on a condition.

Reading CSV Data using Pandas:

Definition: Pandas provides convenient functions to read data from various file formats, and reading CSV (Comma-Separated Values) files is a common use case.

Example:**Execute Reading CSV Data using Pandas.**



```python
import pandas as pd

# Creating a DataFrame
data = {'Name': ['Sharath', 'Srujan', 'Tejas'],
        'Age': [22, 23, 22],
        'City': ['karmanghat', 'Badangpet', 'Saroornagar']}

df = pd.DataFrame(data)

# Data processing operations
df['Age'] = df['Age'] + 2  # Adding 2 to each age
df['Is_Adult'] = df['Age'] >= 18  # Creating a new column based on a condition

print("Processed DataFrame:")
print(df)
```

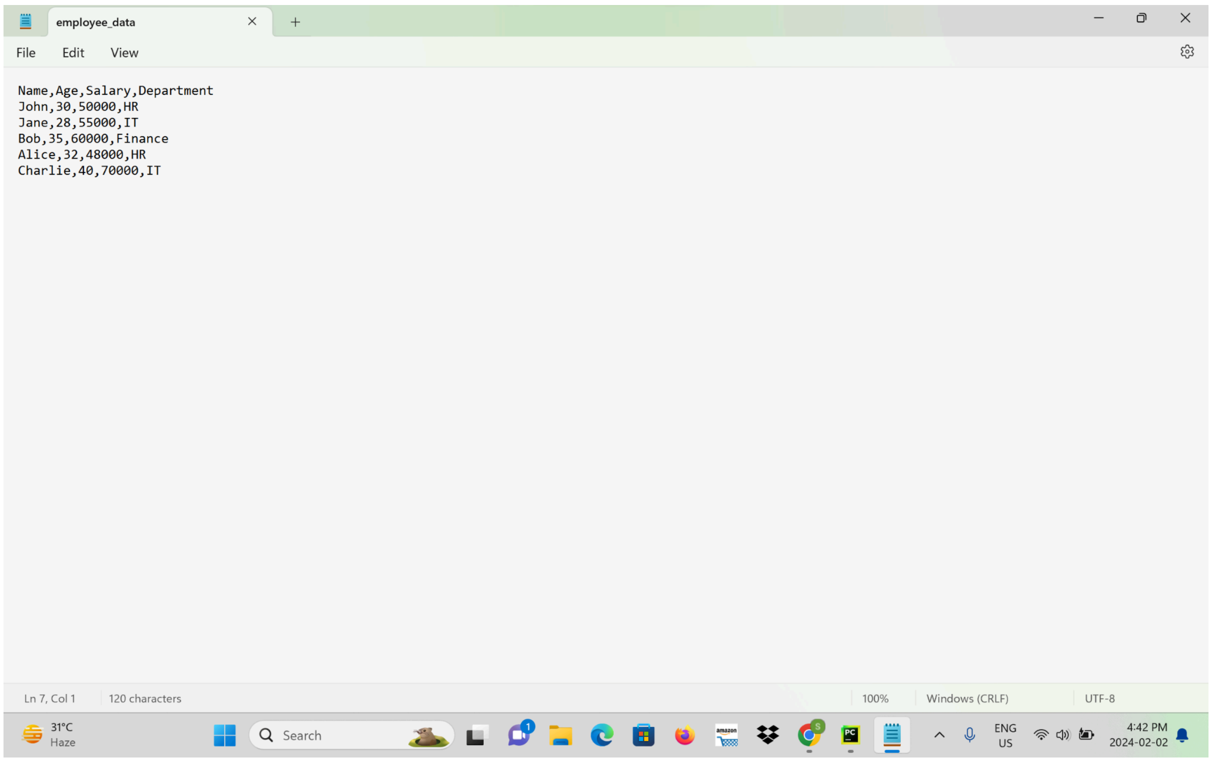
```
C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\example
Processed DataFrame:
      Name  Age         City  Is_Adult
0  Sharath   24   karmanghat      True
1   Srujan   25    Badangpet      True
2    Tejas   24  Saroornagar      True

Process finished with exit code 0
```

ii)Read Data from CSV Files to Pandas Dataframes:

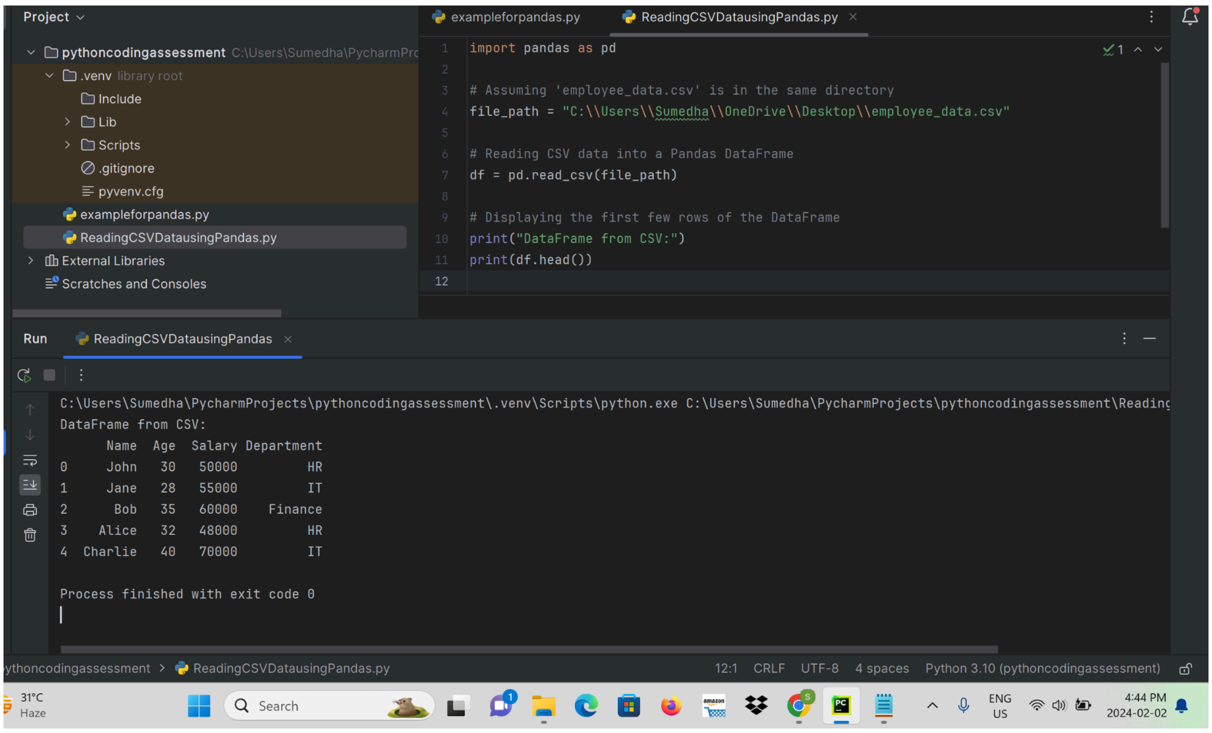
Definition: Pandas provides a read_csv()function that allows users to efficiently read tabular data from CSV (Comma-Separated Values) files into a DataFrame. This function provides various options to handle different CSV file structures, including specifying delimiters, handling missing values, and more.

Example: Let's consider a CSV file named "employee_data.csv" with columns like 'Name', 'Age', 'Salary', and 'Department'. Here's an example of reading this CSV data into a Pandas DataFrame:

→creation of csv file :



```
Name,Age,Salary,Department
John,30,50000,HR
Jane,28,55000,IT
Bob,35,60000,Finance
Alice,32,48000,HR
Charlie,40,70000,IT
```



```python
import pandas as pd

# Assuming 'employee_data.csv' is in the same directory
file_path = "C:\\Users\\Sumedha\\OneDrive\\Desktop\\employee_data.csv"

# Reading CSV data into a Pandas DataFrame
df = pd.read_csv(file_path)

# Displaying the first few rows of the DataFrame
print("DataFrame from CSV:")
print(df.head())
```

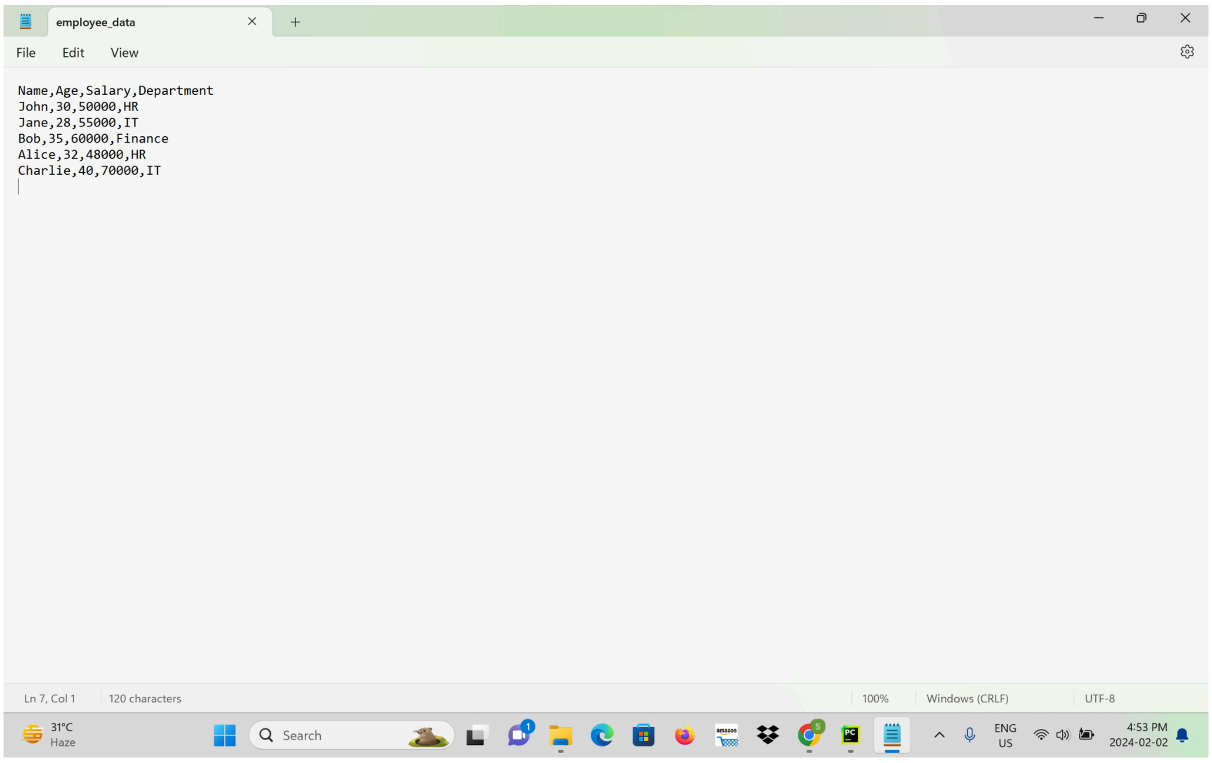
```
C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\Reading
DataFrame from CSV:
      Name  Age  Salary Department
0     John   30   50000         HR
1     Jane   28   55000         IT
2      Bob   35   60000    Finance
3    Alice   32   48000         HR
4  Charlie   40   70000         IT

Process finished with exit code 0
```

SUMMARY: It is used to read data from a CSV file named "employee_data.csv" into a DataFrame. The file path is specified, and the pd.read_csv()function is used to load the CSV data into the

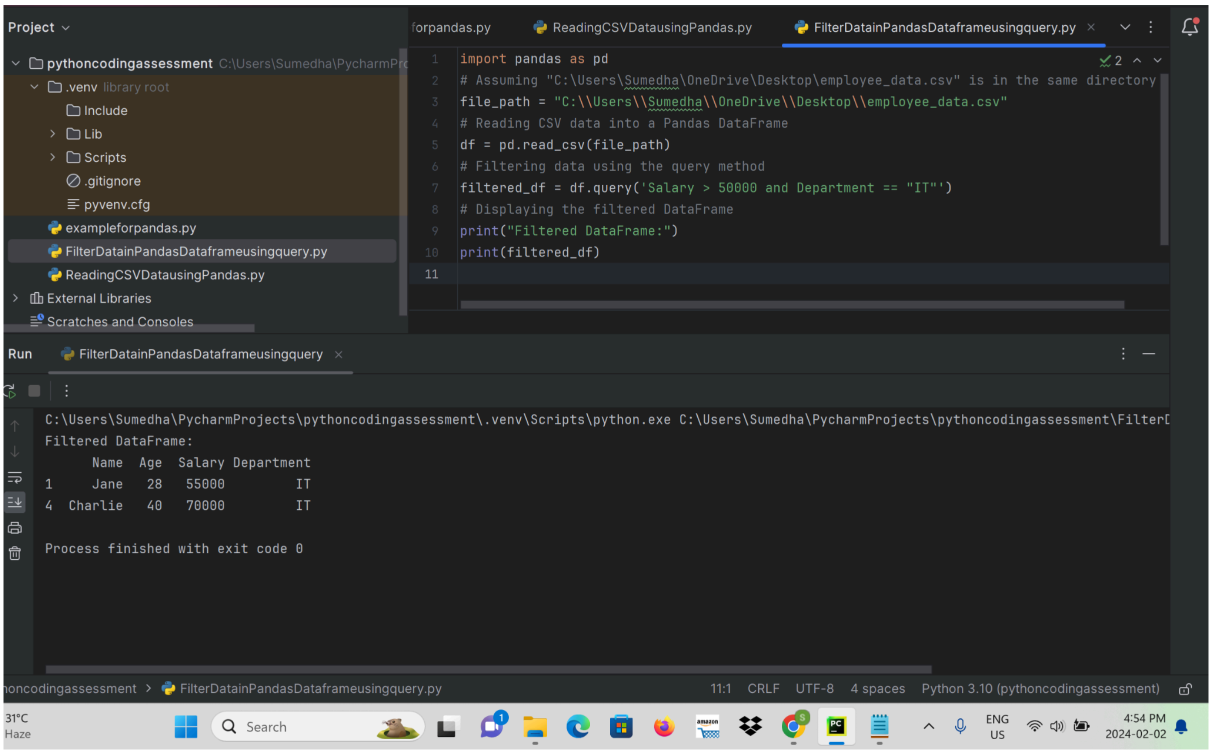
DataFrame named df. The script then prints a message indicating that it's displaying the DataFrame from the CSV file, followed by the first few rows of the DataFrame using df.head().

iii)Filter Data in Pandas Dataframe using query:
Definition: Filtering data in a Pandas DataFrame using the query() method involves selecting rows based on specified conditions. The query() method takes a string expression that represents the filtering conditions.
Example:: Let's consider a CSV file named "employee_data.csv" with columns like 'Name', 'Age', 'Salary', and 'Department'. Here's an example of Filter Data in Pandas Dataframe using query:

→creation of csv file :



```
Name,Age,Salary,Department
John,30,50000,HR
Jane,28,55000,IT
Bob,35,60000,Finance
Alice,32,48000,HR
Charlie,40,70000,IT
```



```
import pandas as pd
# Assuming "C:\Users\Sumedha\OneDrive\Desktop\employee_data.csv" is in the same directory
file_path = "C:\\Users\\Sumedha\\OneDrive\\Desktop\\employee_data.csv"
# Reading CSV data into a Pandas DataFrame
df = pd.read_csv(file_path)
# Filtering data using the query method
filtered_df = df.query('Salary > 50000 and Department == "IT"')
# Displaying the filtered DataFrame
print("Filtered DataFrame:")
print(filtered_df)
```

```
C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\FilterD
Filtered DataFrame:
      Name  Age  Salary Department
1     Jane   28   55000         IT
4  Charlie   40   70000         IT

Process finished with exit code 0
```

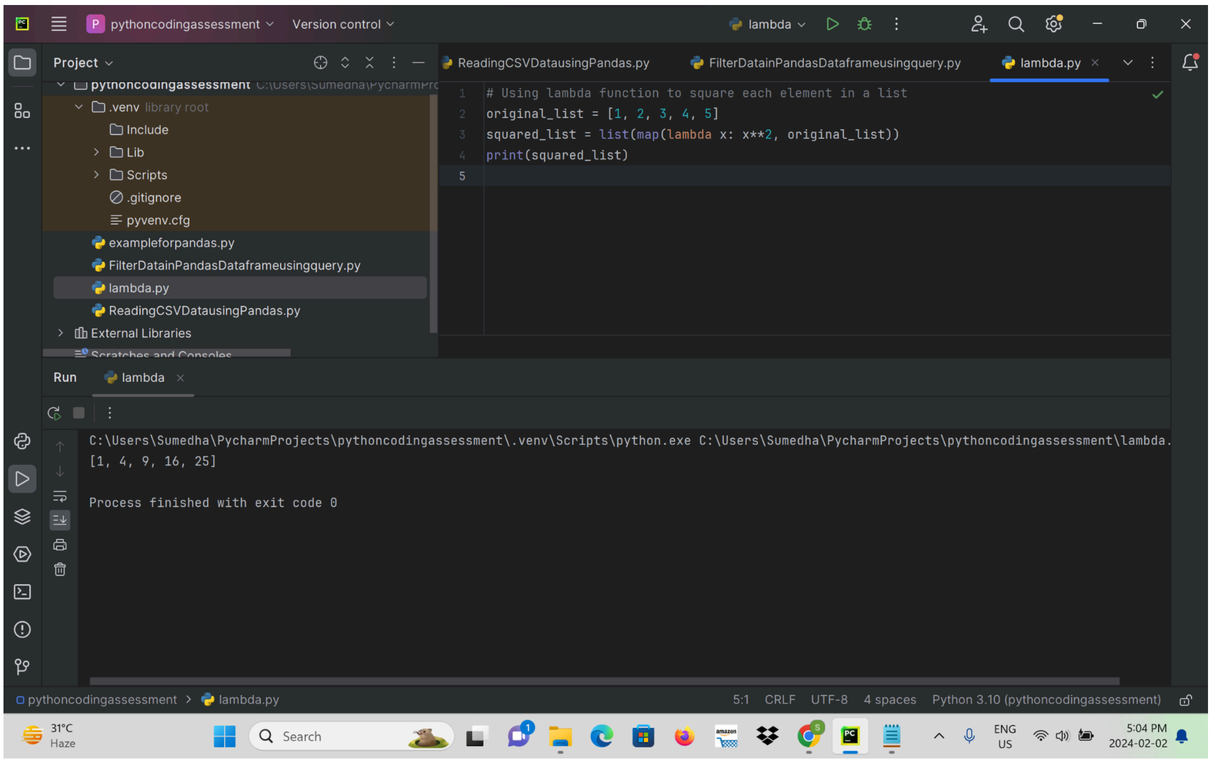
SUMMARY: Firstly I have used an above csv file and then the code reads a CSV file into a Pandas DataFrame, and then filters the data to include only rows where the salary is greater than 50,000 and the department is IT. The resulting DataFrame (filtered_df) contains the filtered data.

2)

i)Execute with one example Lambda Functions in Python:

Definition: A lambda function is an anonymous function defined using the lambda keyword. It is often used for short-lived, one-time operations.

Example:lambda with the map function to square each element in a list.



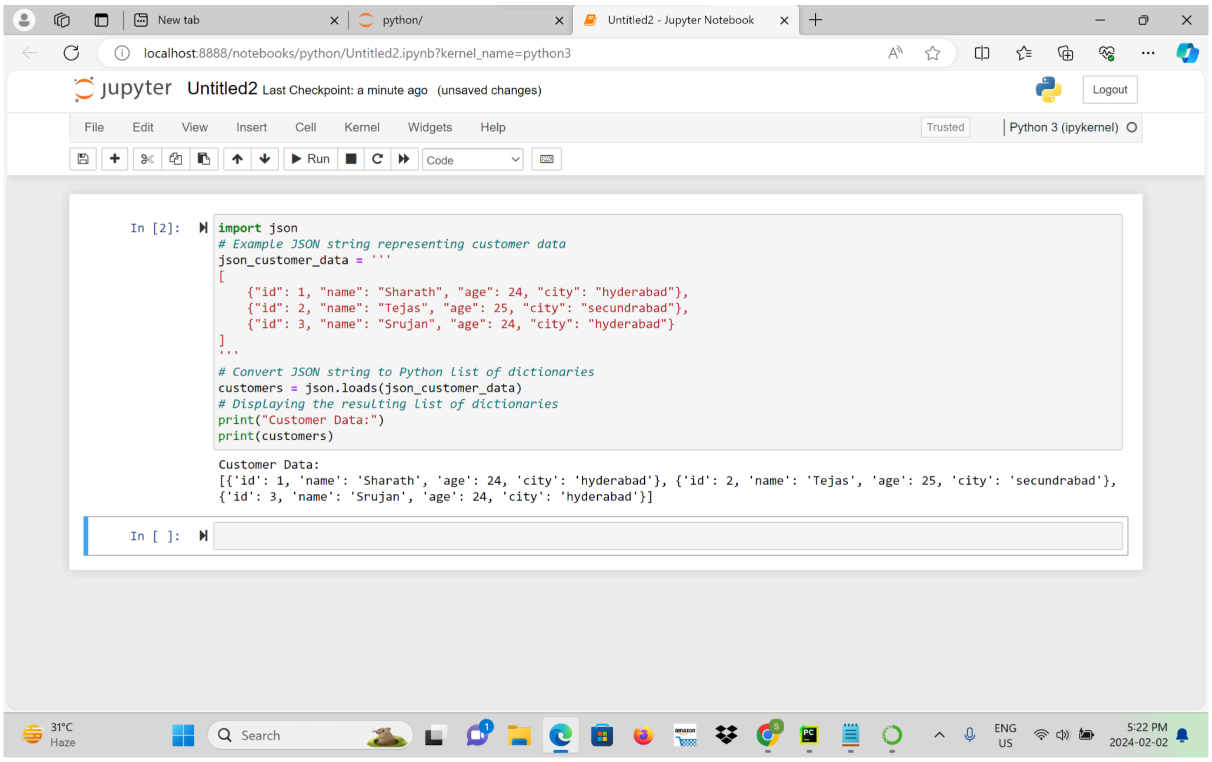
```python
# Using lambda function to square each element in a list
original_list = [1, 2, 3, 4, 5]
squared_list = list(map(lambda x: x**2, original_list))
print(squared_list)
```

```
C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythoncodingassessment\lambda.
[1, 4, 9, 16, 25]

Process finished with exit code 0
```

SUMMARY: a list of numbers is defined, and a lambda function is utilized with the built-in map function. The lambda function takes a single argument and returns its square. The map function applies this lambda function to each element in the list of numbers, resulting in a new list named squared_numbers containing the squared values of the original numbers.

ii)Read JSON Strings to Python dicts or lists: The json module to read JSON strings into Python dictionaries or lists. The json.loads()function is specifically designed for this purpose.

Example:



```
In [2]:  import json
         # Example JSON string representing customer data
         json_customer_data = '''
         [
             {"id": 1, "name": "Sharath", "age": 24, "city": "hyderabad"},
             {"id": 2, "name": "Tejas", "age": 25, "city": "secundrabad"},
             {"id": 3, "name": "Srujan", "age": 24, "city": "hyderabad"}
         ]
         '''
         # Convert JSON string to Python list of dictionaries
         customers = json.loads(json_customer_data)
         # Displaying the resulting list of dictionaries
         print("Customer Data:")
         print(customers)

         Customer Data:
         [{'id': 1, 'name': 'Sharath', 'age': 24, 'city': 'hyderabad'}, {'id': 2, 'name': 'Tejas', 'age': 25, 'city': 'secundrabad'},
         {'id': 3, 'name': 'Srujan', 'age': 24, 'city': 'hyderabad'}]

In [ ]:
```

(MAM JSON.LOADS() FUNCTION IS NOT WORKING IN PYCHARM MAM THATS THE REASON I HAVE EXECUTED IN JUPYTER NOTEBOOK)

SUMMARY: The json module is used to work with a JSON string representing customer data. The json_customer_data variable contains a JSON array, where each element is a dictionary representing information about a customer. The json.loads() function is employed to convert this JSON string into a Python list of dictionaries, creating the customers variable.