ASSIGNMENT 2.3

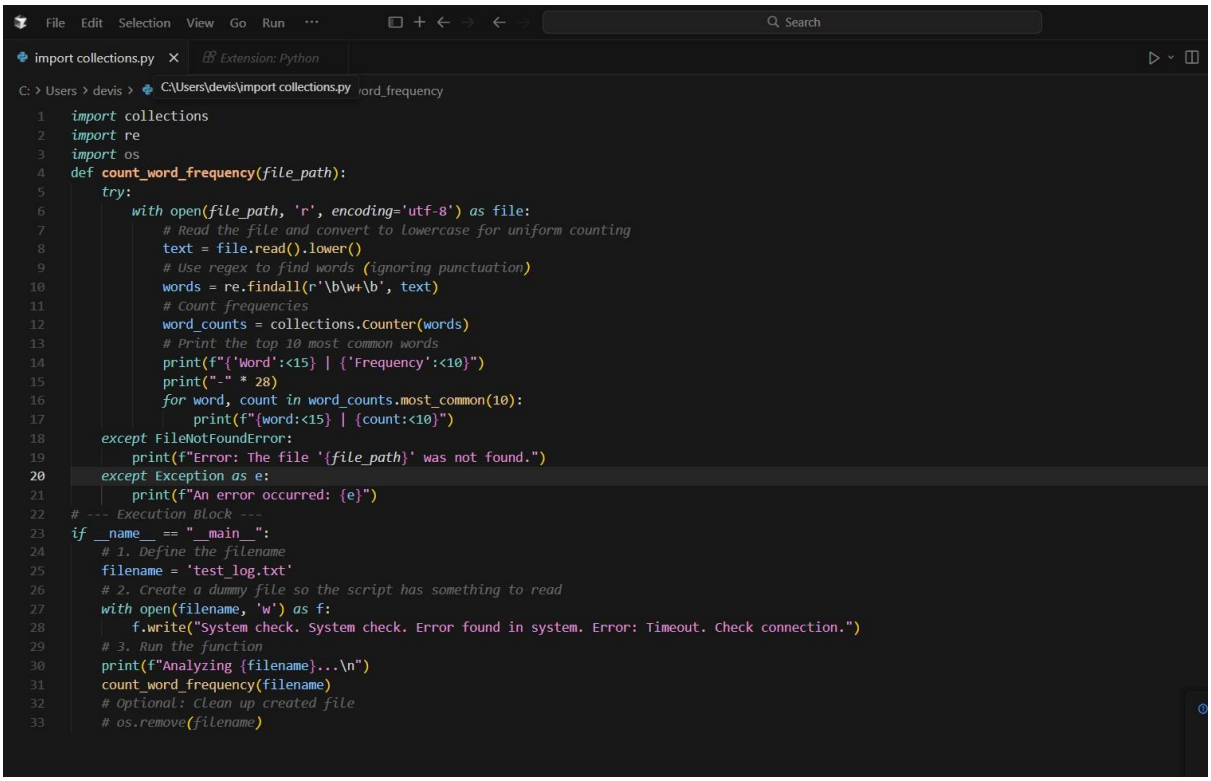
2303A51314

Batch – 05

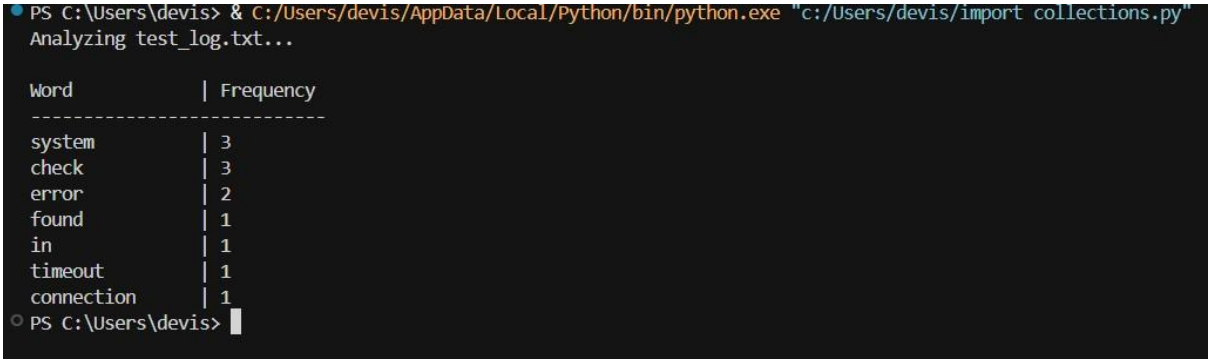
Task – 01

Prompt : Write a Python script to analyze a log file and count keyword frequency.

Code :



Output :



Explanation :

- Imports libraries for word counting (Counter), pattern matching (re), and file handling (os).
- Reads a text file and converts all text to lowercase.
- Uses regex to extract words while ignoring punctuation.
- Counts how many times each word appears.
- Displays the top 10 most frequent words in table format.
- Uses try-except to handle missing files or runtime errors.
- Creates a sample file for testing the function.
- Calls the function to analyze the file content.
- Optional line can delete the test file after execution.

TASK – 02

Prompt : Write a Python program that creates a text file, writes sample text into it, then reads and prints the file content.

Code :

```python
filename = "sample.txt"

# Write sample text to the file
with open(filename, 'w', encoding='utf-8') as f:
    f.write("Hello, this is a sample text file written by Python.\nSecond line of the file.")
# Read the content and print it
with open(filename, 'r', encoding='utf-8') as f:
    content = f.read()
    print(f"\nContent of '{filename}':\n{content}")
```

Output :

```
PS C:\Users\devis> & C:/Users/devis/AppData/Local/Python/bin/python.exe "c:/Users/devis/import collections.py"

Content of 'sample.txt':
Hello, this is a sample text file written by Python.
Second line of the file.
PS C:\Users\devis>
```

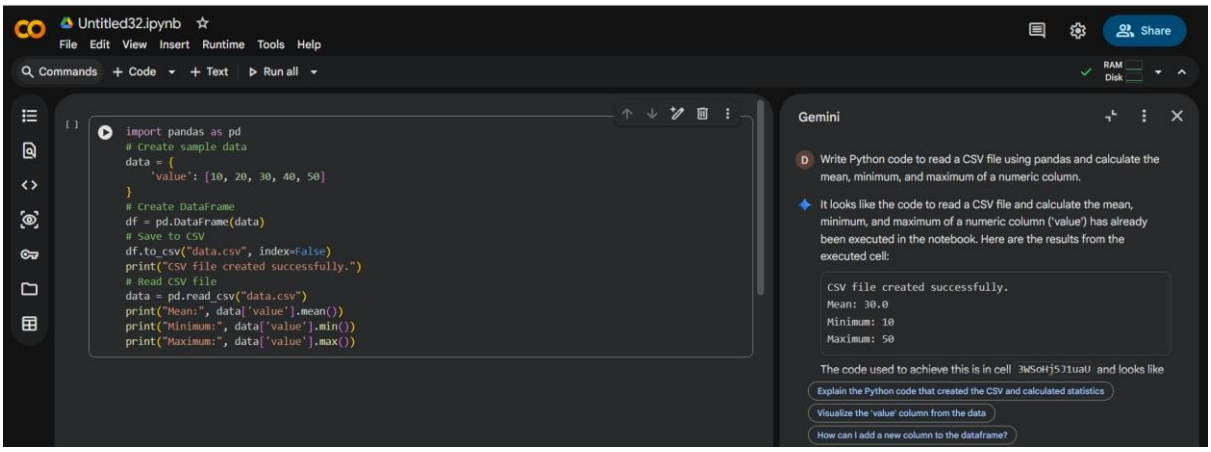
Explanation :

- Stores the file name in a variable.
- Opens the file in write mode and writes sample text into it.
- Automatically closes the file after writing.
- Reopens the file in read mode.
- Reads the entire content and prints it to the screen.

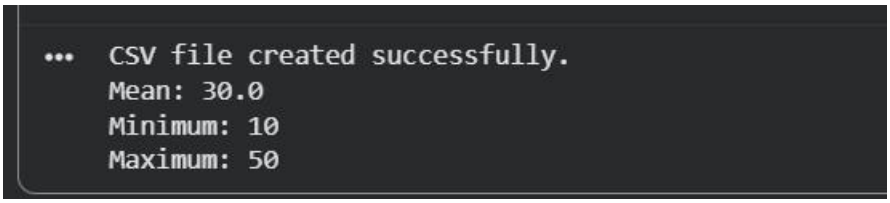
Task – 03

Prompt : Write Python code to read a CSV file using pandas and calculate the mean, minimum, and maximum of a numeric column**.**

Code :



Output :



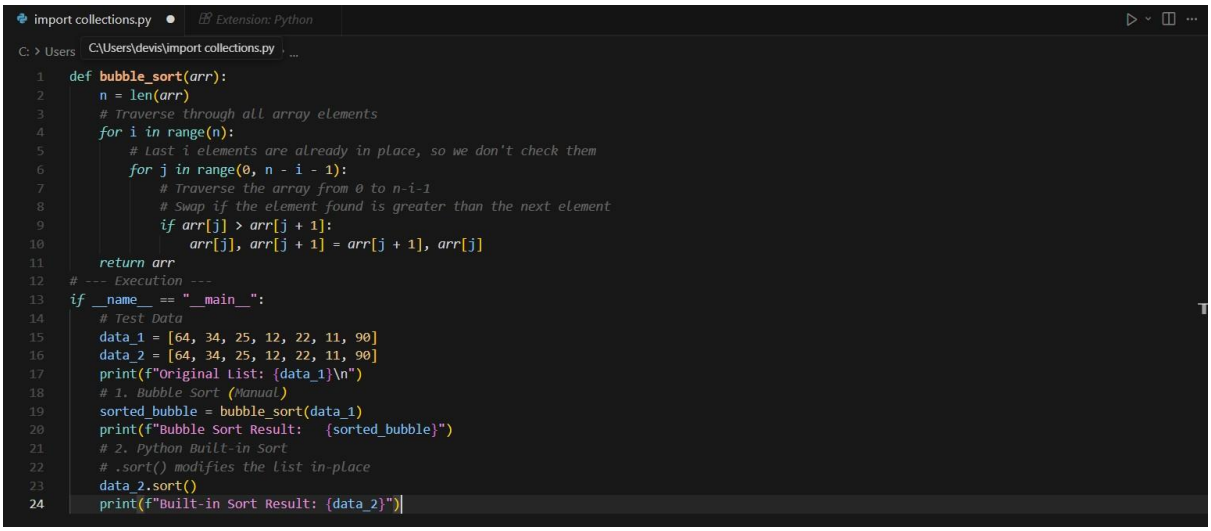
Explanation :

- Imports the pandas library for data handling.
- Creates sample numeric data in a dictionary.
- Converts the data into a DataFrame.
- Saves the DataFrame as a CSV file.
- Reads the CSV file back into the program.
- Calculates and prints the mean, minimum, and maximum values.

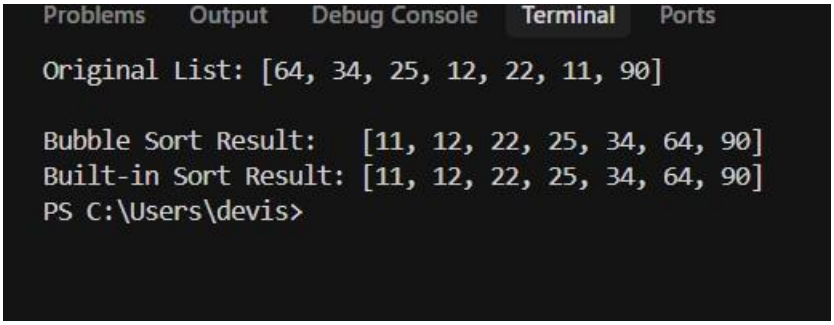
TASK – 04

Prompt : Write Python code for Bubble Sort and Python's built-in sort() method, then give a short comparison of both in terms of efficiency and simplicity.

Code :



Output :



Explanation :

- Bubble Sort is a manual sorting algorithm that repeatedly swaps adjacent elements, it is easy to understand but slow for large lists.
- Built-in sort() is optimized and much faster because it uses Python's efficient internal algorithm.
- Bubble Sort has higher time complexity ($O(n^2)$), while built-in sort is more efficient ($\approx O(n \log n)$).
- Built-in sort() is preferred in real applications due to better performance and simplicity.