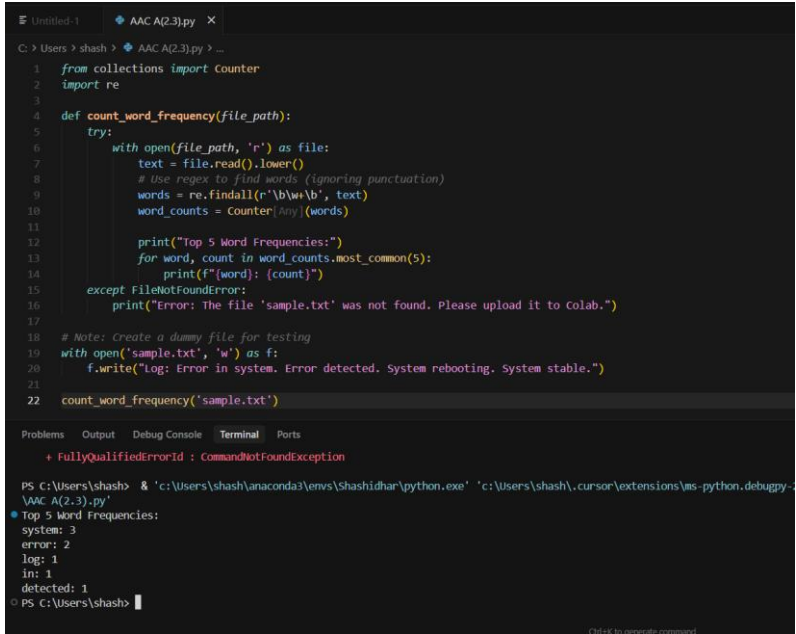


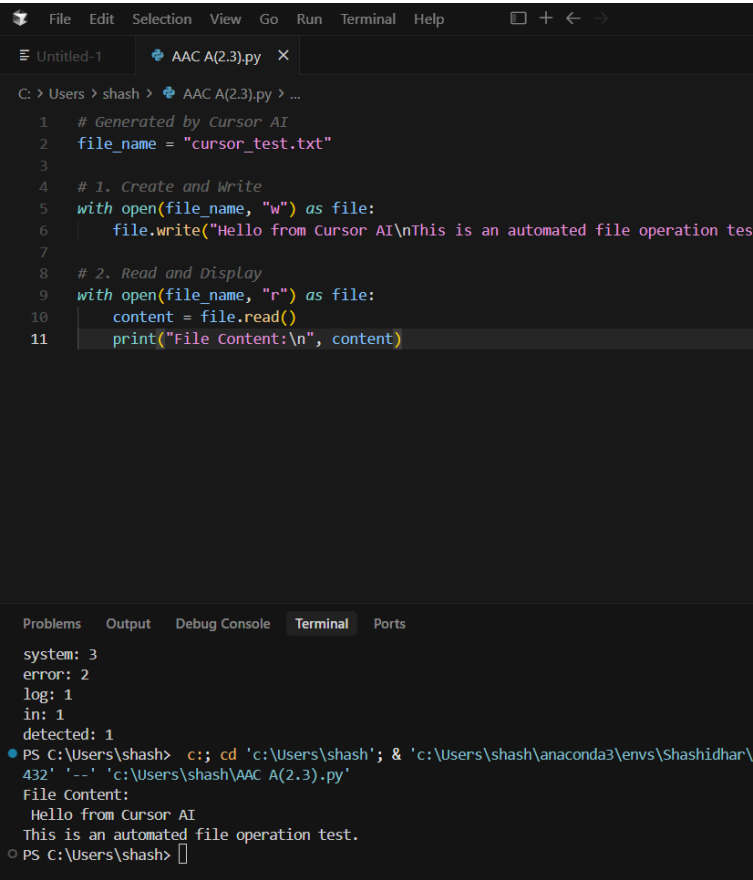
NAME:M.AKASH

H.NO:2303A51820

BATCH:26

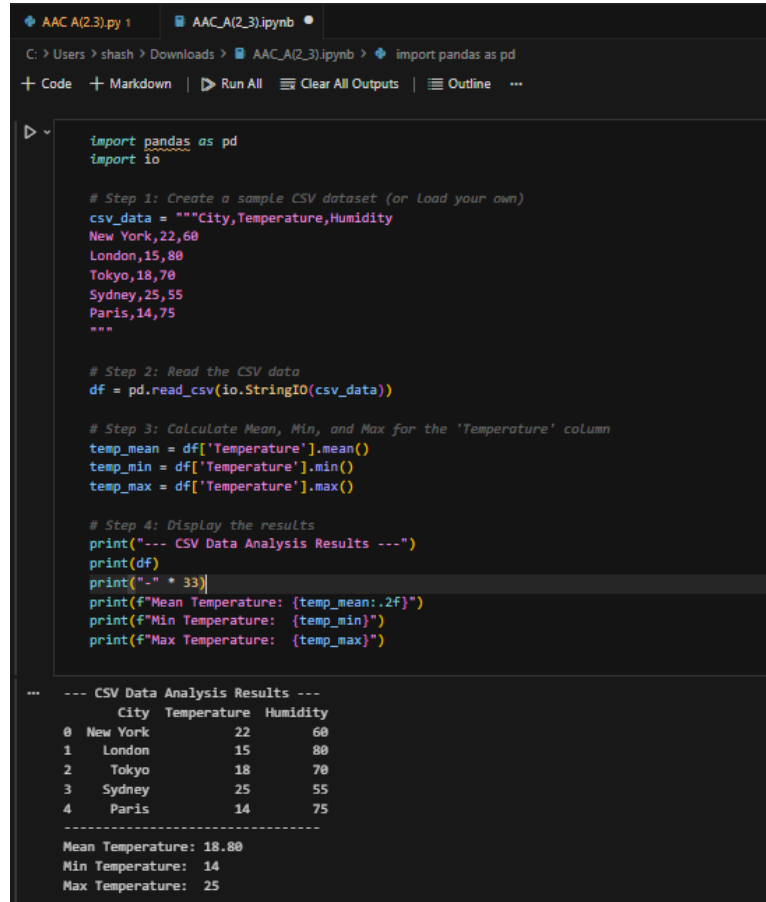
<b>SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE</b>		<b>DEPARTMENT OF COMPUTER SCIENCE ENGINEERING</b>																		
<b>Program Name:</b> B. Tech		<b>Assignment Type:</b> Lab	<b>Academic Year:</b> 2025-2026																	
<b>Course Coordinator Name</b>		Dr. Rishabh Mittal																		
<b>Instructor(s) Name</b>		<table border="1"> <tr><td>Mr. S Naresh Kumar</td></tr> <tr><td>Ms. B. Swathi</td></tr> <tr><td>Dr. Sasanko Shekhar Gantayat</td></tr> <tr><td>Mr. Md Sallauddin</td></tr> <tr><td>Dr. Mathivanan</td></tr> <tr><td>Mr. Y Srikanth</td></tr> <tr><td>Ms. N Shilpa</td></tr> <tr><td>Dr. Rishabh Mittal (Coordinator)</td></tr> <tr><td>Dr. R. Prashant Kumar</td></tr> <tr><td>Mr. Ankushavali MD</td></tr> <tr><td>Mr. B Viswanath</td></tr> <tr><td>Ms. Sujitha Reddy</td></tr> <tr><td>Ms. A. Anitha</td></tr> <tr><td>Ms. M.Madhuri</td></tr> <tr><td>Ms. Katherashala Swetha</td></tr> <tr><td>Ms. Velpula sumalatha</td></tr> <tr><td>Mr. Bingi Raju</td></tr> </table>		Mr. S Naresh Kumar	Ms. B. Swathi	Dr. Sasanko Shekhar Gantayat	Mr. Md Sallauddin	Dr. Mathivanan	Mr. Y Srikanth	Ms. N Shilpa	Dr. Rishabh Mittal (Coordinator)	Dr. R. Prashant Kumar	Mr. Ankushavali MD	Mr. B Viswanath	Ms. Sujitha Reddy	Ms. A. Anitha	Ms. M.Madhuri	Ms. Katherashala Swetha	Ms. Velpula sumalatha	Mr. Bingi Raju
Mr. S Naresh Kumar																				
Ms. B. Swathi																				
Dr. Sasanko Shekhar Gantayat																				
Mr. Md Sallauddin																				
Dr. Mathivanan																				
Mr. Y Srikanth																				
Ms. N Shilpa																				
Dr. Rishabh Mittal (Coordinator)																				
Dr. R. Prashant Kumar																				
Mr. Ankushavali MD																				
Mr. B Viswanath																				
Ms. Sujitha Reddy																				
Ms. A. Anitha																				
Ms. M.Madhuri																				
Ms. Katherashala Swetha																				
Ms. Velpula sumalatha																				
Mr. Bingi Raju																				
<b>CourseCode</b>	23CS002PC304	<b>Course Title</b>	AI Assisted Coding																	
<b>Year/Sem</b>	III/II	<b>Regulation</b>	R23																	
<b>Date and Day of Assignment</b>	Week1 – Wednesday	<b>Time(s)</b>	23CSBTB01 To 23CSBTB52																	
<b>Duration</b>	2 Hours	<b>Applicable to Batches</b>	All batches																	
<b>Assignment Number:1.3</b> (Present assignment number)/ <b>24</b> (Total number of assignments)																				
<b>Q.No.</b>	<b>Question</b>		<b>Expected Time to complete</b>																	
1	Lab 2: Exploring Additional AI Coding Tools beyond Copilot – Gemini (Colab) and Cursor AI  <b>Lab Objectives:</b> ❖ To explore and evaluate the functionality of Google Gemini for AI-		Week1 - Monday																	

	<p>assisted coding within Google Colab.</p> <ul style="list-style-type: none"> <li>❖ To understand and use Cursor AI for code generation, explanation, and refactoring.</li> <li>❖ To compare outputs and usability between Gemini, GitHub Copilot, and Cursor AI.</li> <li>❖ To perform code optimization and documentation using AI tools.</li> </ul> <p><b>Lab Outcomes (LOs):</b> After completing this lab, students will be able to:</p> <ul style="list-style-type: none"> <li>❖ Generate Python code using Google Gemini in Google Colab.</li> <li>❖ Analyze the effectiveness of code explanations and suggestions by Gemini.</li> <li>❖ Set up and use Cursor AI for AI-powered coding assistance.</li> <li>❖ Evaluate and refactor code using Cursor AI features.</li> <li>❖ Compare AI tool behavior and code quality across different platforms.</li> </ul>	
	<p><b>Task 1: Word Frequency from Text File</b></p> <ul style="list-style-type: none"> <li>❖ <b>Scenario:</b> You are analyzing log files for keyword frequency.</li> <li>❖ <b>Task:</b> Use Gemini to generate Python code that reads a text file and counts word frequency, then explains the code.</li> <li>❖ <b>Expected Output:</b> <ul style="list-style-type: none"> <li>➤ Working code</li> <li>➤ Explanation</li> <li>➤ Screenshot</li> </ul> </li> </ul>  <p>The screenshot displays a code editor with a Python script named 'AAC A(2.3).py'. The script defines a function 'count_word_frequency' that takes a file path as input. It attempts to open the file, read its contents, and use a regular expression to find all words, ignoring punctuation. It then counts the frequency of each word and prints the top 5 most common words. An exception handler catches 'FileNotFoundError' and prints an error message. A comment suggests creating a dummy file for testing, which is then created and written to with a log message. Finally, the 'count_word_frequency' function is called with the path to the dummy file.</p> <p>The terminal output shows the execution of the script. It prints the top 5 word frequencies: 'system: 3', 'error: 2', 'log: 1', 'in: 1', and 'detected: 1'. The prompt 'PS C:\Users\shash&gt;' is visible at the bottom.</p>	

	<div><h3>Task 2: File Operations Using Cursor AI</h3><div><div>❖ <b>Scenario:</b> You are automating basic file operations.</div><div><div>❖ <b>Task:</b> Use Cursor AI to generate a program that:</div><div><div>➤ Creates a text file</div><div>➤ Writes sample text</div><div>➤ Reads and displays the content</div></div></div><div><div>❖ <b>Expected Output:</b></div><div><div>➤ Functional code</div><div>➤ Cursor AI screenshots</div></div></div><div><p>The screenshot displays the Cursor AI code editor with a Python script named 'AAC A(2.3).py'. The code performs two tasks: creating a text file and reading its content. The terminal at the bottom shows the execution output, including system statistics and the file's content.</p><pre>1 # Generated by Cursor AI 2 file_name = "cursor_test.txt" 3 4 # 1. Create and Write 5 with open(file_name, "w") as file: 6     file.write("Hello from Cursor AI\nThis is an automated file operation test") 7 8 # 2. Read and Display 9 with open(file_name, "r") as file: 10     content = file.read() 11     print("File Content:\n", content)</pre><p>Terminal Output:</p><pre>system: 3 error: 2 log: 1 in: 1 detected: 1 PS C:\Users\shash&gt; c:: cd 'c:\Users\shash'; &amp; 'c:\Users\shash\anaconda3\envs\Shashidhar\432' '--' 'c:\Users\shash\AAC A(2.3).py' File Content: Hello from Cursor AI This is an automated file operation test. PS C:\Users\shash&gt;</pre></div></div></div>	
	<div><h3>Task 3: CSV Data Analysis</h3><div><div>❖ <b>Scenario:</b> You are processing structured data from a CSV file.</div><div><div>❖ <b>Task:</b> Use Gemini in Colab to read a CSV file and calculate mean, min, and max.</div></div></div></div>	

❖ **Expected Output:**

- Correct output
- Screenshot



```
import pandas as pd
import io

# Step 1: Create a sample CSV dataset (or Load your own)
csv_data = """City,Temperature,Humidity
New York,22,60
London,15,80
Tokyo,18,70
Sydney,25,55
Paris,14,75
"""

# Step 2: Read the CSV data
df = pd.read_csv(io.StringIO(csv_data))

# Step 3: Calculate Mean, Min, and Max for the 'Temperature' column
temp_mean = df['Temperature'].mean()
temp_min = df['Temperature'].min()
temp_max = df['Temperature'].max()

# Step 4: Display the results
print("--- CSV Data Analysis Results ---")
print(df)
print("-" * 33)
print(f"Mean Temperature: {temp_mean:.2f}")
print(f"Min Temperature: {temp_min}")
print(f"Max Temperature: {temp_max}")
```

---

	City	Temperature	Humidity
0	New York	22	60
1	London	15	80
2	Tokyo	18	70
3	Sydney	25	55
4	Paris	14	75

-----

Mean Temperature: 18.80  
Min Temperature: 14  
Max Temperature: 25

**Task 4: Sorting Lists – Manual vs Built-in**

❖ **Scenario:**

You are reviewing algorithm choices for efficiency.

❖ **Task:**

Use **Gemini** to generate:

- Bubble sort
- Python's built-in sort()
- Compare both implementations.

❖ **Expected Output:**

- Two versions of code
- Short comparison

```
File Edit Selection View Go Run Terminal Help
AAC A(2.3).py X

C:\Users\shash> shash > AAC A(2.3).py > ...

1 def bubble_sort(arr):
2     n = len(arr)
3     # Outer loop to traverse through all array elements
4     for i in range(n):
5         # Last i elements are already in place, so we ignore them
6         for j in range(0, n - i - 1):
7             # Swap if the element found is greater than the next
8             if arr[j] > arr[j + 1]:
9                 arr[j], arr[j + 1] = arr[j + 1], arr[j]
10    return arr
11
12 # Example usage
13 data = [64, 34, 25, 12, 22, 11, 90]
14 print(f"Manual Bubble Sort: {bubble_sort(data.copy())}")

Problems Output Debug Console Terminal Ports

import pandas as pd
ModuleNotFoundError: No module named 'pandas'
PS C:\Users\shash> c:: cd 'c:\Users\shash'; & 'c:\Users\shash\anaconda3\envs\Shashid
'c:\Users\shash\AAC A(2.3).py'
Traceback (most recent call last):
  File "c:\Users\shash\AAC A(2.3).py", line 1, in <module>
    import pandas as pd
ModuleNotFoundError: No module named 'pandas'
PS C:\Users\shash> c:: cd 'c:\Users\shash'; & 'c:\Users\shash\anaconda3\envs\Shashid
'c:\Users\shash\AAC A(2.3).py'
Manual Bubble Sort: [11, 12, 22, 25, 34, 64, 90]
PS C:\Users\shash> 
```

**Note:** Report should be submitted as a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots.