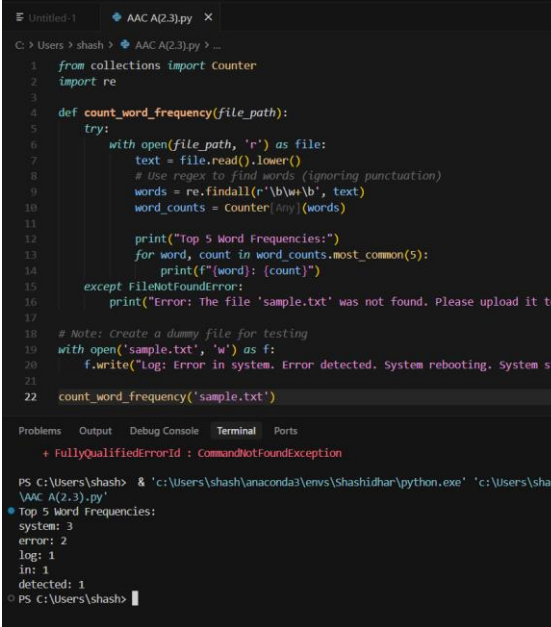
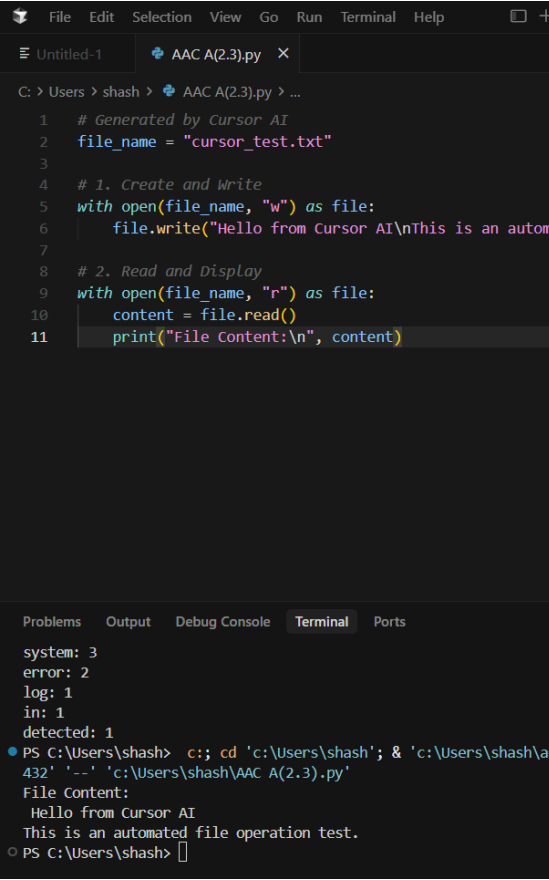
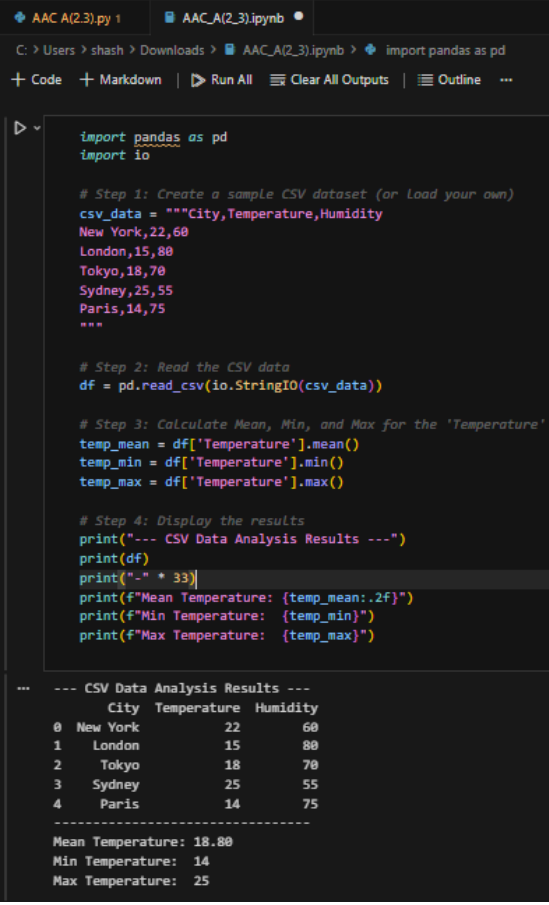


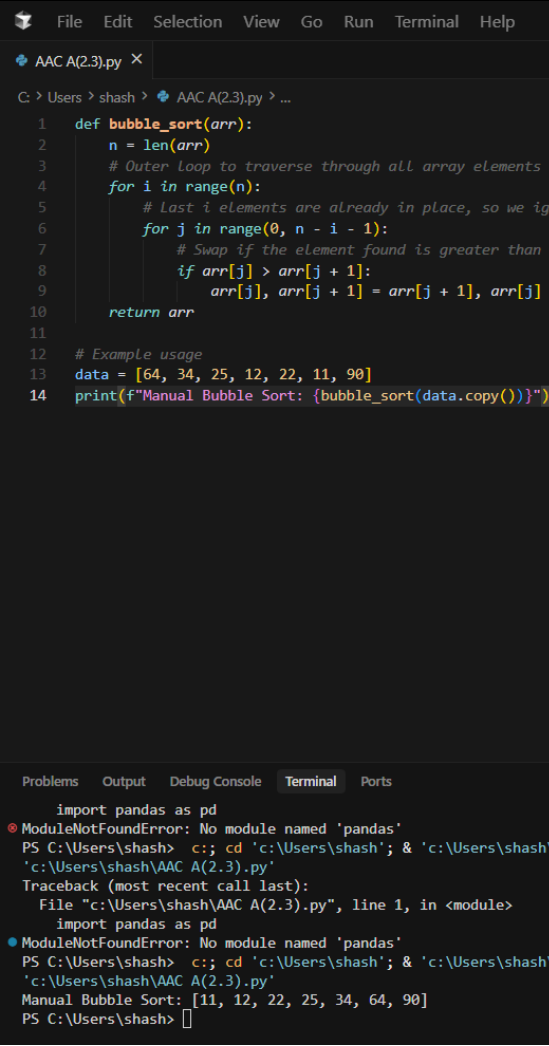
NAME:G.Bhagath H.NO:2303A51807 BATCH:26

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE				DEPART	
Program Name: B. Tech				Assignment Type: Lab	
Course Coordinator Name				Dr. Rishabh Mittal	
Instructor(s) Name					
				Mr. S Naresh Kumar	
				Ms. B. Swathi	
				Dr. Sasanko Shekhar Gantaya	
				Mr. Md Sallauddin	
				Dr. Mathivanan	
				Mr. Y Srikanth	
				Ms. N Shilpa	
				Dr. Rishabh Mittal (Coordinat	
				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					
CourseCode	23CS002PC304			Course Title	AI Assisted
Year/Sem	III/II			Regulation	R23
Date and Day of Assignment	Week1 – Wednesday			Time(s)	23CSBTB0
Duration	2 Hours			Applicable to Batches	All batches
			Assignment Number:1.3(Present assignment number)/24(Total n		
Q.No.		Question			
1		Lab 2: Exploring Additional AI Coding Tools beyond C and Cursor AI  <b>Lab Objectives:</b>			

		<div>❖ To explore and evaluate the functionality of AI-assisted coding within Google Colab.</div> <div>❖ To understand and use Cursor AI for code generation and refactoring.</div> <div>❖ To compare outputs and usability between Gemini and Cursor AI.</div> <div>❖ To perform code optimization and documentation.</div> <div><b>Lab Outcomes (LOs):</b> After completing this lab, students will be able to:<div>❖ Generate Python code using Google Gemini in a Jupyter Notebook.</div><div>❖ Analyze the effectiveness of code explanations generated by Gemini.</div><div>❖ Set up and use Cursor AI for AI-powered coding.</div><div>❖ Evaluate and refactor code using Cursor AI features.</div><div>❖ Compare AI tool behavior and code quality across different scenarios.</div></div> <div><b>Task 1: Word Frequency from Text File</b><div>❖ <b>Scenario:</b> You are analyzing log files for keyword frequency.</div><div>❖ <b>Task:</b> Use Gemini to generate Python code that reads a text file, calculates word frequency, then explains the code.</div><div>❖ <b>Expected Output:</b><div>➤ Working code</div><div>➤ Explanation</div><div>➤ Screenshot</div></div><div><pre>1 from collections import Counter 2 import re 3 4 def count_word_frequency(file_path): 5     try: 6         with open(file_path, 'r') as file: 7             text = file.read().lower() 8             # Use regex to find words (ignoring punctuation) 9             words = re.findall(r'\b\w+\b', text) 10            word_counts = Counter(words) 11 12            print("Top 5 Word Frequencies:") 13            for word, count in word_counts.most_common(5): 14                print(f"{word}: {count}") 15        except FileNotFoundError: 16            print("Error: The file 'sample.txt' was not found. Please upload it to the workspace.") 17 18    # Note: Create a dummy file for testing 19    with open('sample.txt', 'w') as f: 20        f.write("Log: Error in system. Error detected. System rebooting. System started.") 21 22    count_word_frequency('sample.txt')</pre><p>PS C:\Users\shash&gt; &amp; 'c:\Users\shash\anaconda3\envs\Shashidhar\python.exe' 'c:\Users\shash\VAAC A(2.3).py'</p><p>VAAC A(2.3).py</p><p>Top 5 Word Frequencies:</p><p>system: 3</p><p>error: 2</p><p>log: 1</p><p>in: 1</p><p>detected: 1</p><p>PS C:\Users\shash&gt;</p></div></div>
--	--	---

		<div><div>Task 2: File Operations Using Cursor AI</div><div><div>❖ Scenario:</div><div>You are automating basic file operations.</div></div><div><div>❖ Task:</div><div>Use Cursor AI to generate a program that:<ul style="list-style-type: none"><li>➤ Creates a text file</li><li>➤ Writes sample text</li><li>➤ Reads and displays the content</li></ul></div></div><div><div>❖ Expected Output:</div><div><ul style="list-style-type: none"><li>➤ Functional code</li><li>➤ Cursor AI screenshots</li></ul></div></div><div><div></div><div>➤</div></div><div><div>Task 3: CSV Data Analysis</div><div><div>❖ Scenario:</div><div>You are processing structured data from a CSV file</div></div><div><div>❖ Task:</div><div>Use Gemini in Colab to read a CSV file and calculat</div></div></div></div>
--	--	--

		<div><div><div>❖ Expected Output:</div><div><div>➤ Correct output</div><div>➤ Screenshot</div></div></div><div><p>The screenshot shows a Jupyter Notebook interface with a file named 'AAC_A(2_3).ipynb'. The code is as follows:</p><pre>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' 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("\n * 33]") print(f"Mean Temperature: {temp_mean:.2f}") print(f"Min Temperature: {temp_min}") print(f"Max Temperature: {temp_max}")</pre><p>The output of the code is displayed below the code cell:</p><pre>--- --- CSV Data Analysis Results ---    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</pre></div><div><div>➤</div></div></div>
		<div><div>Task 4: Sorting Lists – Manual vs Built-in</div><div><div>❖ Scenario:</div><div>You are reviewing algorithm choices for efficiency.</div></div><div><div>❖ Task:</div><div>Use <b>Gemini</b> to generate:<div><div>➤ Bubble sort</div><div>➤ Python's built-in sort()</div><div>➤ Compare both implementations.</div></div></div></div><div><div>❖ Expected Output:</div><div><div>➤ Two versions of code</div><div>➤ Short comparison</div></div></div></div>

		<div data-bbox="1078 191 1624 1230"><pre>File Edit Selection View Go Run Terminal Help AAC A(2.3).py X C: &gt; Users &gt; shash &gt; AAC A(2.3).py &gt; ... 1 def bubble_sort(arr): 2     n = len(arr) 3     # Outer Loop to traverse through all array element 4     for i in range(n): 5         # Last i elements are already in place, so we ig 6         for j in range(0, n - i - 1): 7             # Swap if the element found is greater than 8             if arr[j] &gt; 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&gt; c:; cd 'c:\Users\shash'; &amp; 'c:\Users\shash\ '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 &lt;module&gt;     import pandas as pd ModuleNotFoundError: No module named 'pandas' PS C:\Users\shash&gt; c:; cd 'c:\Users\shash'; &amp; 'c:\Users\shash\ 'c:\Users\shash\AAC A(2.3).py' Manual Bubble Sort: [11, 12, 22, 25, 34, 64, 90] PS C:\Users\shash&gt; </pre></div> <p><b>Note:</b> Report should be submitted as a word document in a single document with prompts, comments &amp; code explanation and if required, screenshots.</p>
--	--	---