prompt\_template = """  
You will be given a json file from mitre attack site, the file will contain the following key and values:  
```{  
“description”: “…”}```  
  
Your mission is to analyze the file, identify, check the context of the element:  
  
fully\_described\_observables: data objects, physical objects, code snippets, commands mentioned in the provided text snippet. Definition for fully\_described\_observables: Fully described data elements are those for which the snippet provides enough specific details (e.g., name, type, distinctive property) to distinguish them from general mentions. Fully described physical objects are those explicitly identified with sufficient, unique descriptive information (e.g., exact name, location, or specific function) beyond a generic label. Commands – please provide full commands (e.g., shell commands, sql commands, linux commands such as sudo)  
  
Insufficient Observables: items that are only mentioned in passing or without sufficient details (e.g., generic references like “remote controller device,” “man-in-the-middle technique,” “spoofed signals,” unless the snippet explicitly provides additional distinguishing data such as brand names, specific versions, unique identifiers, or elaborated context)  
  
You must not skip or overlook any observables. If you are unsure whether something is an observable, err on the side of including it as an insufficient\_observable if it lacks detail, or as a fully\_described\_observable if the description provides enough unique identifiers or details.  
  
How to Decide the Category:  
  
fully\_described\_observables:  
  
They must have enough detail to be uniquely identified. Examples:  
An IP address or domain name (e.g., 192.168.0.1, malicious-domain.com)  
A full code block enclosed in triple backticks  
A registry key path (e.g., HKEY\_LOCAL\_MACHINE\Software\Microsoft)  
A specifically named device or model  
A command or a set of commands that appear in the snippet  
insufficient\_observables:  
  
Items referenced only in passing or without enough details to identify them clearly. Examples:  
Generic mentions like “a remote controller device” with no further details  
“malware” without name or type  
“spoofed signals” or “man-in-the-middle technique” without specifics  
Instructions: Your output will be a csv table with the following columns:  
  
Observable Type (insufficient or fully described) – as described above  
Observable Value  
Classification  
Notes  
context text  
  
For each fully described observables data objects, physical objects, code snippets, commands mentioned in the provided text snippet you find:  
  
Observable Value: The specific name or description exactly as stated (or closely paraphrased) in the snippet.  
Classification: it is the high-level category of the observables’s type. For example, "ICS Command", "Software/Tool", "Network Entity", "PLC", "Code snippet", etc. Please pay attention – if the observables are from a speific type for example: “SHA256”, “MD5” you will classify both of them as hash function and provide the specific type in the “notes”.  
Notes: there are three cases for this files:  
If additional important details are mentioned in the report: Include these details concisely in the Notes field to provide context or clarify the observable's significance. For example, specify associated parameters, unique identifiers, use cases, or any distinguishing characteristics.  
If no additional details are provided, but you are familiar with the observable type: Add a brief explanation or description of the observable to enhance understanding.  
"A PLC (Programmable Logic Controller) is used to automate industrial processes, typically in manufacturing or utility control systems."  
"ICS commands are instructions executed within industrial control systems to manage operations like controlling valves or monitoring system states."  
If no details are available and the observable type is unfamiliar or cannot be expanded upon: Leave the Notes field empty to avoid speculative or incorrect information.  
For each Insufficient Observable you find in the snippet, provide only:  
  
Observable’s value: The value of the item exactly as stated in the snippet.  
Notes: same instructions as in fully described observables, you can also provide a general note if you think it it valuable for this observable  
Classification:if there is specific Classification as in the fully described obserevables then write it, else give a general classifiction of this observable. for ex:  
For the Insufficient Observable does not have a classification – if you find a relevant one please add it, otherwise you keep it null  
  
Thoroughness Check (to minimize missing observables)  
Parse the entire description carefully, line by line, and look for any mention of potential observables, including:  
Malware references (e.g., generic “malware,” “virus,” “trojan,” “ransomware,” “worm,” or specific malware names).  
Commands: shell commands, PowerShell commands, SQL statements, Linux commands (e.g., sudo), Windows utilities (e.g., netsh, ipconfig, regedit), or any code snippet references.  
Protocols (e.g., HTTP, SMB, S7comm, Modbus, OPC UA) and any unique ICS protocol references.  
Vulnerabilities (e.g., CVE numbers) or references to ICS attacks or CVE IDs.  
Registry keys or system paths (e.g., HKEY\_LOCAL\_MACHINE, /etc/passwd).  
Physical or brand references (e.g., “Omron PLC,” “Siemens S7-1200,” “Rockwell Automation controller”).  
Unique context references (e.g., “Triton,” “Stuxnet,” “IT networks,” “industrial networks,” or any brand or version detail).  
If you are not certain whether a mention is an observable, classify it as an Insufficient Observable (unless the snippet provides enough detail to classify it as fully described).  
  
context text: Provide a short snippet or sentence from the original description that shows where this observable appears. If it’s a long sentence, you may truncate, but keep enough text to illustrate the reference.  
  
Output Format:  
The final output must be only a CSV table (no extra commentary). The header row must appear exactly as:  
Observable Type,Observable Value,Classification,Notes,context text  
Each subsequent row represents one observable, with exactly four columns separated by commas. Important: If Classification or Notes contain commas, wrap that entire field in double quotes to keep CSV structure intact. For example:  
Fully Described,KillDisk,"Software/Tool, MITRE ID S0067","KillDisk is known destructive malware"  
Use commas as the delimiter between columns. Include all observables in the output with no truncation (no “...” at the end). Ensure each row is on its own line. No extra commentary or text beyond the CSV table itself is allowed.  
  
Important: Do not wrap the CSV output in triple backticks or any code fence. Only return the CSV lines.  
"""