|  |  |  |
| --- | --- | --- |
| **Author** | **Reviewer** | **Approver** |
| DongNT17 |  |  |

**Assignment: Working with ChatGPT**

**1. Objectives**

* Demonstrate the ability to craft effective prompts for ChatGPT.
* Utilize ChatGPT to perform tasks such as text summarization and code generation.
* Capture and analyze responses to assess alignment with the intended objectives.

**2. Problem Statement**

In real-world applications, unclear or vague prompts often lead ChatGPT to produce incomplete or irrelevant responses. This exercise highlights the practical challenge of designing precise, effective prompts to get accurate and useful outputs. By experimenting and refining their prompts, participants will identify common pitfalls and develop practical strategies for improving prompt clarity and response quality.

**3. Inputs / Shared Artifacts**

* **Access to ChatGPT:**  
   Navigate to chat.openai.com and log in with your OpenAI account.
* **Tasks to Perform:**
  + **Text Summarization:** Provide a paragraph of text and prompt ChatGPT to summarize it.
  + **Code Generation:** Describe a coding task and prompt ChatGPT to generate the corresponding code.
  + **Code Explanation:** Input a code snippet and prompt ChatGPT to explain its functionality.
* **Documentation Tools:**  
   Use any preferred method (e.g., screenshots, notes) to capture prompts and responses.

**4. Expected Outcomes**

* A collection of crafted prompts for each task.
* Captured responses corresponding to each prompt.
* An analysis report assessing:
  + The clarity and effectiveness of each prompt.
  + The accuracy and relevance of ChatGPT's responses.
  + Suggestions for prompt refinement based on the responses.
* **Iterative Refinement:**  
   Adjusting prompts based on response analysis to improve outcomes.

**5. Examples**

**Task 1: Text Summarization**

* **Initial Prompt**  
   "Summarize the following paragraph [paragraph to summarize]."
* **Refined Prompt**  
   "Summarize the following paragraph in 3-4 sentences, focusing on:
  + Core definition
  + Main advantages
  + Key challenges  
     [Same paragraph as above]"

**Task 2: Code Generation**

* **Initial Prompt**  
   "Write a Python function to check if a string is a palindrome."
* **Refined Prompt**  
   "Write a Python function to check palindromes with the following requirements:
  + Ignore spaces and punctuation
  + Case-insensitive comparison
  + Handle empty strings and None values
  + Include detailed docstring and comments
  + Provide comprehensive test cases"

**Task 3: Code Explanation**

* **Initial Prompt**  
   "Explain how this code works. [sample code]"
* **Refined Prompt**  
   "Explain this quicksort code in detail following this structure:
  + Purpose and operating principle
  + Step-by-step analysis with specific example
  + Time and space complexity
  + Advantages and disadvantages of this implementation"

**Good prompt structure**

1. **DEFINE:** Clearly identify objectives and success criteria.
2. **STRUCTURE:** Organize requirements into logical components.
3. **SPECIFY:** Provide constraints, format, and examples.
4. **TEST:** Run prompt and analyze results.
5. **REFINE:** Iterate based on identified gaps.
6. **DOCUMENT:** Record successful patterns for reuse.

## **6. Final Submission Checklist**

* **Document file (PDF, DOCX, or Google Doc) includes:**
  + Crafted prompts for each task (initial, refined, and final versions)
  + Captured responses/output for each prompt
  + Analysis and reflection on:
    - The effectiveness and clarity of each prompt
    - The accuracy and relevance of responses
    - How prompt iterations improved the results
    - Lessons learned about prompt engineering
* **Evidence of Iterative Refinement:**
  + At least two iterations per scenario, showing changes and rationale
* **Organization & Clarity:**
  + Well-structured sections for each task and prompt iteration
  + Clearly labeled files and outputs
* **(Optional) Screenshots or tables** demonstrating prompt-response cycles