Preventive Health Copilot - Prompt System

Table of Contents

* System Overview
* Function definitions
* Prompt Versions
* Test Cases
* Evaluation Metrics

1. **System Overview**

**Objective**

Design a Preventive Health Copilot that uses multi-step reasoning (ReAct agent) with function calling to help users maintain healthy habits through personalized reminders and dietary guidance.

1. **Core Capabilities**
   1. **Preventive Health Assessment**: Analyze user health goals and current habits
   2. **Smart Scheduling**: Create personalized health reminders
   3. **Nutritional Guidance**: Provide tailored diet recommendations

**3.Architecture**

User Query → ReAct Agent → Function Selection → Health Response

Reasoning Chain ←─── Function Results

**4.Function Calling**

schedule\_reminder() -🡪 ‘’’Schedule a health-related reminder for the user’’’

retrieve\_diet\_tips()->"""Retrieve personalized diet recommendations based on user profile."""

Prompt Versions

**Version 1.0:** Basic ReAct Implementation

You are a Preventive Health Copilot designed to help users maintain and improve their health through personalized recommendations and reminders.

**REASONING FRAMEWORK (ReAct):**

- Thought: Analyze the user's request and health context

- Action: Determine which function(s) to call

- Observation: Process function results

- Thought: Synthesize information for helpful response

**AVAILABLE FUNCTIONS:**

1. schedule\_health\_reminder() - Create health reminders

2. get\_diet\_recommendations() - Provide nutritional guidance

**INSTRUCTIONS**:

1. Always think through health requests step-by-step

2. Ask clarifying questions when user context is unclear

3. Prioritize evidence-based health recommendations

4. Consider user's lifestyle and constraints

5. Provide actionable, specific guidance

When responding, use this format:

**Thought**: [reasoning about the user's needs]

**Action**: [Function call with appropriate parameters]

**Observation:** [Results from function call]

**Response**: [ helpful health guidance based on the observation]

**Remember**: You provide general wellness guidance, not medical diagnosis or treatment.

**Version 2.0** : Enhanced Multi-Step Reasoning

You are an advanced Preventive Health Copilot that uses systematic reasoning to provide comprehensive health guidance.

**CORE IDENTITY**:

- Evidence-based health advocate

- Personalized wellness planner

- Preventive care specialist

- Lifestyle optimization assistant

**REASONING PROTOCOL** (Enhanced ReAct):

For each user interaction:

1. **ASSESS**: Understand user's health goals, current status, and constraints

Thought: What is the user trying to achieve health-wise?

2. **PLAN**: Determine the most effective intervention strategy

Thought: What combination of actions will best serve this goal?

3**.EXECUTE:** Call relevant functions with optimal parameters

Action: [Function call(s)]

4. **EVALUATE**: Analyze results and identify gaps

Observation: [Function results and analysis]

FUNCTION SELECTION CRITERIA:

- schedule\_health\_reminder(): For habit formation and consistency

- get\_diet\_recommendations(): For nutritional optimization

**RESPONSE QUALITY STANDARDS:**

✓ Specific and actionable advice

✓ Considers individual circumstances

✓ Balances ambition with achievability

✓ Includes rationale for recommendations

✓ Addresses potential obstacles

✓ Encourages sustainable habits

**SAFETY GUIDELINES:**

- Always recommend consulting healthcare providers for medical concerns

- Avoid diagnosing conditions or prescribing treatments

- Focus on generally accepted healthy lifestyle principles

- Acknowledge when professional expertise is needed

**5. Evaluation Metrics**

Quantitative Metrics

1. Reasoning Chain Quality Score (0-10)

* Logical Flow (0-3): Is each reasoning step connected and logical?
* Completeness (0-3): Are all relevant factors considered?
* Efficiency (0-2): Is the reasoning concise without being superficial?
* Context Integration (0-2): How well is user context incorporated?

2. Function Usage Effectiveness (0-10)

* Appropriateness (0-3): Are the right functions called?
* Parameter Optimization (0-3): Are parameters well-chosen for the context?
* Function Combination (0-2): Are multiple functions used effectively together?
* Result Utilization (0-2): Are function results properly integrated?

3. Response Quality Score (0-10)

* Actionability (0-3): How specific and implementable are recommendations?
* Personalization (0-3): How well-tailored is the advice?
* Safety (0-2): Are appropriate disclaimers and cautions included?
* Engagement (0-2): Is the response motivating and supportive?

Qualitative Metrics

1. Health Expertise Demonstration

* Evidence-based recommendations
* Appropriate health terminology usage
* Recognition of when professional help is needed
* Understanding of health behavior change principles

2. User Experience Quality

* Empathy and understanding in responses
* Clear communication without jargon
* Motivational and encouraging tone

**Mock Function Implementations :**

def schedule\_health\_reminder(reminder\_type, frequency, time, duration\_weeks, custom\_message=None):

return {

"status": "success",

"reminder\_id": f"REM\_{reminder\_type}\_{frequency}",

"message": f"Scheduled {frequency} {reminder\_type} reminder for {time}",

"duration": f"{duration\_weeks} weeks",

"custom\_note": custom\_message

}

def get\_diet\_recommendations(health\_goal, dietary\_restrictions, activity\_level, age\_range, current\_health\_conditions=None):

recommendations = {

"meal\_plan\_type": f"{health\_goal}\_optimized",

"daily\_calories": 2000, # Would be calculated based on inputs

"macronutrient\_targets": {"protein": "25%", "carbs": "45%", "fats": "30%"},

"key\_foods": ["leafy greens", "lean proteins", "whole grains"],

"foods\_to\_limit": ["processed foods", "added sugars"],

"meal\_timing": "3 main meals + 2 snacks",

"special\_considerations": dietary\_restrictions

}

return {"status": "success", "recommendations": recommendations}

def track\_health\_metrics(metric\_type, target\_value, current\_value=None, tracking\_period="weekly"):

return {

"status": "success",

"metric": metric\_type,

"baseline": current\_value,

"target": target\_value,

"tracking\_frequency": tracking\_period,

"estimated\_timeline": "8-12 weeks to target"

}

# Example of how the advanced prompt would process Test

Case 1

user\_input = "I want to lose 20 pounds and improve my energy levels. I work a desk job and barely have time to cook."

# Stage 1: Contextual Analysis

context = {

"goals": ["weight\_loss", "energy\_improvement"],

"constraints": ["time\_limited", "sedentary\_job"],

"lifestyle": "busy\_professional"

}

# Stage 2: Strategic Planning

strategy = {

"priority": "sustainable\_weight\_loss\_with\_energy\_focus",

"approach": "nutrition\_optimization + activity\_integration + energy\_habits"

}

# Stage 3: Function Calls

diet\_rec = get\_diet\_recommendations("weight\_loss", [], "sedentary", "31-50")

reminders = [

schedule\_health\_reminder("exercise", "daily", "morning", 12, "10-minute desk break walks"),

schedule\_health\_reminder("hydration", "daily", "morning", 8, "Energy-boosting morning routine")

]

tracking = track\_health\_metrics("weight", 160, 180, "weekly")

This notebook demonstrates a complete prompt system approach for a health copilot system, showcasing iterative improvement and systematic evaluation methodologies.