Places to follow up:
https://www.reddit.com/r/PromptCoding/comments/16wn8r4/proposal_for_high_level_library_to_make_prompt/
https://www.reddit.com/r/aipromptprogramming/comments/16wna4a/for_programmers_proposal_for_high_level_library/
https://www.reddit.com/r/ChatGPTCoding/comments/16wn6jo/proposal_for_high_level_library_to_make_prompt/

Test out to see how close they get to what I want:
https://docs.flowiseai.com/
https://kani.readthedocs.io/en/latest/engine_reference.html
https://github.com/zhudotexe/kani
https://github.com/hero-page/hero-ml
https://python.langchain.com/docs/expression_language/cookbook/multiple_chains

Proposal

I have created prompt chains, and have discussed the best ways to do it with multiple clients (prospective and current). So I'm proposing a high-level prompt chaining library, open source. Built in Python and then in JS, so it can be used in Node, React, Typescript, etc.

I'm imagining the structure of nodes being stored, as code to start, then very soon as text in a database for easier editing and versioning. The library would also manage which node the user is currently at, running prompts/code, and movement between nodes. More on those later, for now let's talk about the most important part:

The most important part of the library

"Node" (or maybe "Step"?) object

- Inputs: Input values to use, wrap them in [] to use them, e.g. `subject`
- Display to user: e.g. "Ready to write a poem about [subject]"
- Run prompt or custom code: "write a poem about [subject] like Dr. Seuss" or `writePoem(subject)`
- Display to user: "Here is your poem: [result]"
- Outputs to Child Prompt: `result` (prompt completion)
- Child nodes: 1-n node IDs, along with characters to watch for in `result`, to stop streaming to user *and* indicate which child node to choose

I need a cleaner term to differentiate between what is displayed to the user before, versus after, running something. But, you get the idea.

Questions for you

Things to think about **while you read the use cases below**:

1. Would this kind of library be useful for you?
2. Would you be willing to contribute an example use case, if I build it?
3. Does a library that operates at this level already exist?
    a. I know this functionality can be built in many different libraries, e.g. Langchain, Kani, aiwaves-cn/agents, etc.
    b. They are all lower level than what I'm envisioning.
    c. I'm envisioning something high level enough that storing the nodes as code would be a similar length as the use cases above.

Illustrative use cases

Use case 1: simple chain to gather information

Node 1:

- Inputs: none (starting node)
- Display to user: "welcome, please give me a subject for your poem"
- Run prompt: n/a
- Display to user: n/a
- Outputs to Child Prompt: subject (user's response)
- Child nodes: Node 2

Node 2:

- Input: subject
- Display to user: "sounds good, how about a setting for your poem?"
- Run prompt: n/a
- Display to user: n/a
- Outputs to Child Prompt: setting (user's response)
- Child nodes: node 3

Node 3:

- Inputs: subject, setting
- Display to user: n/a
- Run prompt: "Generate a poem, in the style of Dr. Seuss, about [subject]. Poem is set in [setting]"
- Display to user: response from prompt
- Outputs to Child Prompt: n/a
- Child nodes: none (end)

Use case 2: branching based on user input or prompt completion

Node 1:

- Inputs: none (starting node)
- Display to user: "what would you like to do? 1: play a game, 2: write a poem"
- Run prompt: n/a
- Display to user: n/a
- Outputs to Child Prompt: n/a
- Child nodes: Using user's response: Node 2 if 1, Node 3 if 2, otherwise Node 1

Node 2:

- Input: n/a
- Display to user: "You are standing in an open field west of a white house, with a boarded front door. There is a small mailbox here. What do you do?"
- Run prompt: "analyze the following text: ```[user's response]``` If text is about moving west, going to a house, or going to a door, say: 'A'. Otherwise, if text is about a mailbox, or interacting with one, say 'B'. Otherwise, say 'C'.
- Display to user: n/a
- Outputs to Child Prompt: action (user's response)
- Child nodes: Using prompt completion: Node 4 if A, Node 5 if B, Node 6 if C

Node 3:

- Input: n/a
- Display to user: "welcome, please give me a subject for your poem"
- Run prompt: n/a
- Display to user: n/a
- Outputs to Child Prompt: subject (user's response)
- Child nodes: Node 7

(Not showing further nodes because this demonstrates the use case well enough.)

Use case 3: running custom code

Node 1:

- Inputs: none (starting node)
- Display to user: n/a
- Run prompt: n/a
- Display to user: "Hi, I'm a binary mathematician. My only job is to double numbers. What would you like to double?"
- Outputs to Child Prompt: number (user's response)
- Child nodes: Node 2

Node 2:

- Input: number
- Display to user: n/a
- Run code: doubleNumber([number])
- Display to user: result from function
- Outputs to Child Prompt: n/a
- Child nodes: n/a

Thank you for reading and your feedback!