

Intermediate Techniques

System Prompt Variables & Message Pairs

- System Prompt Variables: Wrap camelCase string in {{...}} to setup a variable within your system prompt. Could be used to customize the prompt for specific purposes or to further modularize prompts.
- Message Pairs: Most common use-case is for chat history, but fundamentally adds context to this specific request.
- **NOTE:** My personal preference is to avoid things like system prompt variables and instead systematize user requests. You'll see examples of how I do this in the Prompt Engineering module. This is just what makes more sense to me.

The screenshot shows the Claude AI interface. At the top, there are three buttons: a menu icon, a plus sign for creating new models, and a cube icon labeled "Default". Below the buttons, the title "Testing New System" is shown with a "Last saved Nov 16 at 11:13 AM" timestamp. To the right of the title are several icons: a file icon, a "1" indicating one message pair, a gear icon, "Examples", and an "Improve prompt" button. The main area is divided into sections. The first section is titled "System Prompt" and contains the text: "This is an example prompt, just print out {{selectedResponse}} This is an LLM response from {{}} no matter what I send to you." The second section is titled "User" and contains the message "Trying this out!". The third section is titled "Assistant" and contains the message "This is an LLM response from Anthropic API". The fourth section is titled "User" and contains the message "Yeah, why do you keep saying that?". At the bottom of the interface, there are two buttons: "Pre-fill response" and "Add message pair".

Thoughts on Frontend/Backend

You will need to select which languages/frameworks you want to use

- **NOTE:** Do NOT expose your API keys on the frontend AT ALL. This is a massive security vulnerability. This is why third-party LLM API calls will be made on the backend. Remember, your bank accounts and/or cards are linked to these API keys. Set proper limits and don't expose yourself to bad actors.
- I'll be using TypeScript/Vite for frontend, Flutter for mobile, and Go/Gin for backend in this tutorial. AI-Powered IDEs will be a necessity for the non-coders (even if you know how to code, they boost productivity significantly). Cursor is my go-to.
- Again, code will not be a prominent factor in this course, but I'll give concrete steps for setting up a basic frontend and backend that you can use to iterate on for your project.