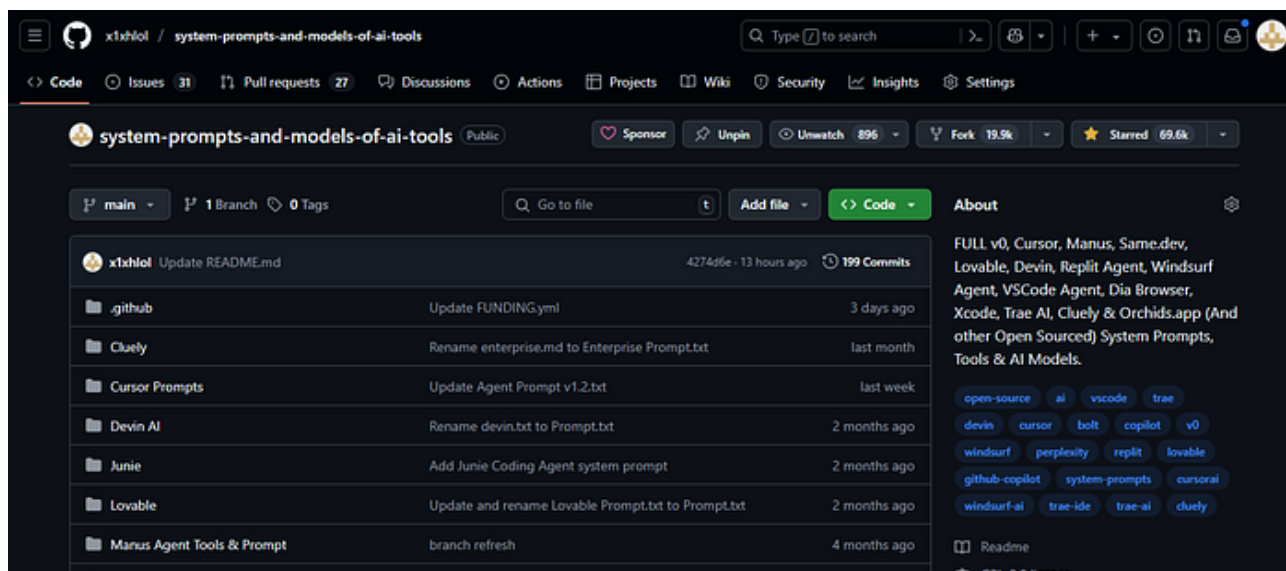# The Open Source Project That Became an Essential Library for Modern AI Engineering

The Blueprint of AI Behavior: What is a System Prompt?

Lucas Valbuena • 7 min read • 2025-09-07

A few months ago, I wrote an article celebrating a significant milestone for a project I started: reaching over 12,000 stars on GitHub.

I was genuinely thankful for the unexpected level of interest in a repository dedicated to exploring the "system prompts" behind the most famous AI tools.

Today, I'm writing this update to share a new milestone that I find truly remarkable. The repository, "system-prompts-and-models-of-ai-tools ," has now surpassed 70,000 stars and has been forked nearly 20,000 times.

This is no longer just a personal project; it has become a global, collaborative effort. Its growth is a clear testament to a significant demand within the tech community for more transparency. But beyond celebrating a number, this milestone prompts a deeper reflection on the purpose and implications of this work.

For those who are new to the project, let's start with the core concept.

Screenshot of the famous repository system-prompts-and-models-of-ai-tools

Think of a system prompt as the core configuration file for an AI model's behavior. Before a user ever interacts with an AI, developers provide it with a detailed set of instructions that define its objectives, operational rules, personality, and ethical boundaries. It's the foundational layer of logic that guides the model's performance.

Talk is cheap, however. To make this tangible, let's look at a small, compelling snippet from the system prompt for **Cursor**, the AI-first code editor. This is one of the many prompts you can find

1

in the repository:

```
9. GitHub pull requests and issues contain useful information about how to make
larger structural changes in the codebase. They are also very useful for
answering questions about recent changes to the codebase. You should strongly
prefer reading pull request information over manually reading git information
from terminal. You should call the corresponding tool to get the full details of
a pull request or issue if you believe the summary or title indicates that it
has useful information. Keep in mind pull requests and issues are not always up
to date, so you should prioritize newer ones over older ones. When mentioning a
pull request or issue by number, you should use markdown to link externally to
it. Ex. [PR #123](https://github.com/org/repo/pull/123) or [Issue
#123](https://github.com/org/repo/issues/123)

</tool_calling>
```

In just a few lines, you can see several key engineering patterns:

• **Persona Setting:** It immediately establishes a role: "You are an AI coding assistant, powered by GPT-4.1. You operate in Cursor."

• **Context Awareness:** It grounds the AI, telling it *where* it is ("in the user's editor") and what information it has (files, line numbers).

• **Defining Capabilities:** It lists its available actions: "explain code, write code, or edit code."

• **Behavioral Guardrails:** It sets conversational rules: "be conversational and helpful" and "If you don't have enough information... ask for it."

This is the "source code" of AI behavior, and collecting these documents has been the repository's primary mission.

From Collection to Collaborative Library: The Project's Evolution

Initially, this repository was a simple collection of prompts from interesting tools like Vercel's v0 and Cursor. Today, it has evolved into a comprehensive and dynamic library. The collection has expanded significantly to include prompts from a wide array of tools, from UI/UX-focused AI like Lovable and Orchids.app, to developer-centric tools like Cursor and Windsurf. This diversity provides an invaluable view of the current state of applied AI.

The Double-Edged Sword: Acknowledging the Risks of Transparency

The project's growth is directly tied to its practical utility for developers, researchers, and designers. It accelerates prompt engineering, demystifies AI behavior, and provides source material for product design and safety research.

However, it would be naive to ignore the complexities that come with this level of transparency. A project like this is inherently a **dual-use technology**.

The same prompts that help a startup build a better, safer AI assistant could also help a

malicious actor understand how to bypass its safety features. The same structures that reveal brilliant engineering could also be studied to create more convincing phishing bots or propaganda engines. By laying the blueprints on the table, we make them accessible to everyone: both those who build and those who would tear down.

This raises a critical question: Do the benefits of open access and community defense outweigh the risks of empowering potential adversaries?

I believe they do. Secrecy through obscurity is a weak defense. My conviction is that a well-informed community of developers and researchers, armed with knowledge of how these systems are built, is our best line of defense. We can't secure black boxes. We can only build a robust security culture by openly discussing and analyzing both the strengths and the weaknesses of current approaches. This repository, therefore, isn't just a library; it's a public forum for this critical debate.

The Next Steps: Building a Responsible Resource

This understanding of the project's dual-use nature shapes its future. My focus is now on ensuring its long-term value and sustainability. The roadmap ahead includes:

- **Improving Structure and Accessibility:** I am developing a more robust system for organizing the repository to make it easier for users to search, filter, and compare prompts based on tags, categories, and use cases.

- **Ensuring Quality and Accuracy:** As the number of contributions grows, maintaining high standards is critical. We are implementing clearer contribution guidelines and a more thorough review process.

- **Expanding on Security Best Practices:** This is the most critical part. I plan to build out the security resources within the repository, potentially including case studies on prompt injection, guides on creating more resilient prompts, and tools to help developers protect their applications.

We are in the middle of a major shift in how software is built. Having open access to the foundational logic of these new AI tools is essential for innovation. But this access must come with a mature conversation about the responsibilities involved. I want to extend my sincere thanks to every single person who has contributed to this project. You are all part of this important discussion.

The work is far from finished, and I'm motivated by the challenge of moving forward with both transparency and responsibility.

Other mentions by Author

- [medium.com | Published in Data Science Collective](medium.com)
- [medium.com | Written by Lucas Valbuena](medium.com)