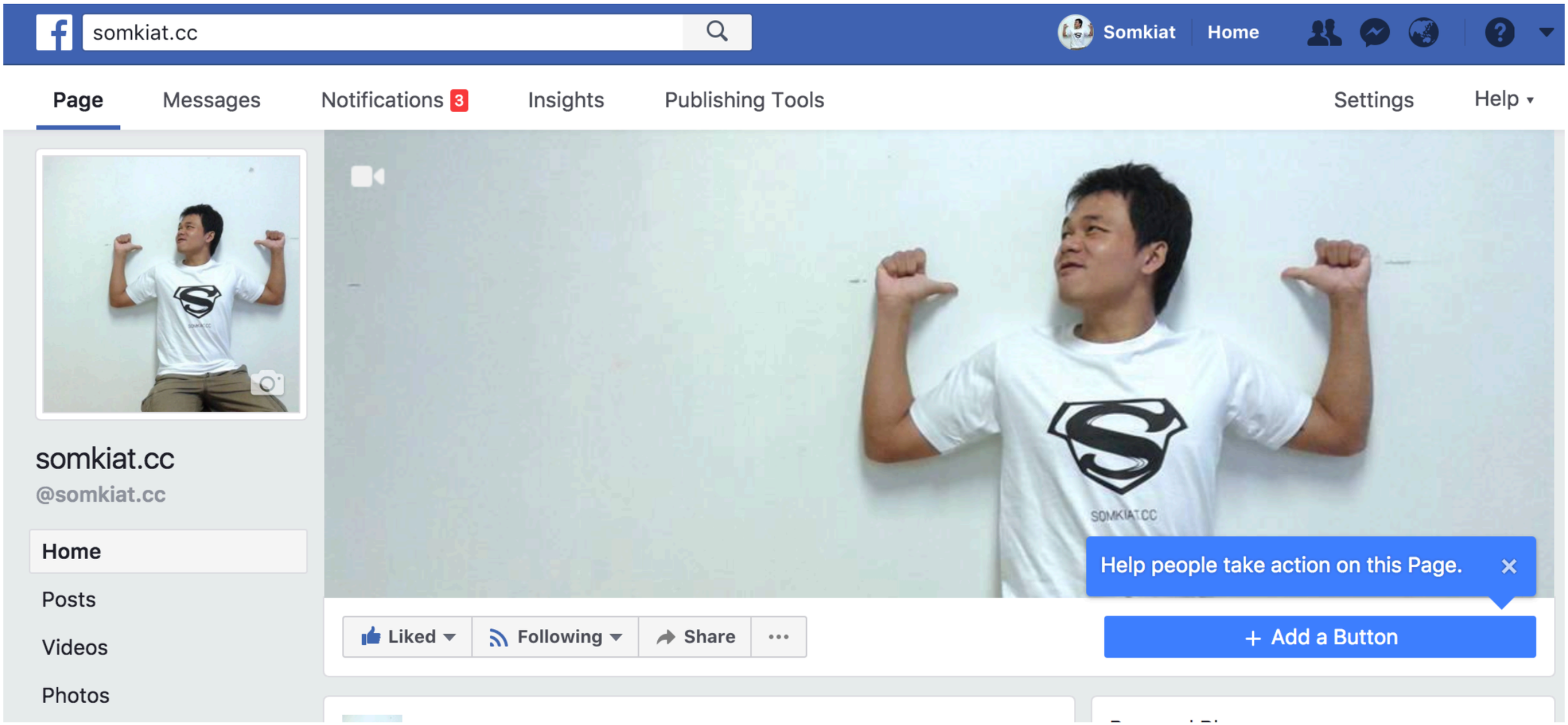
สยามชำนาญกิจ
SIAM CHAMNAN KIT

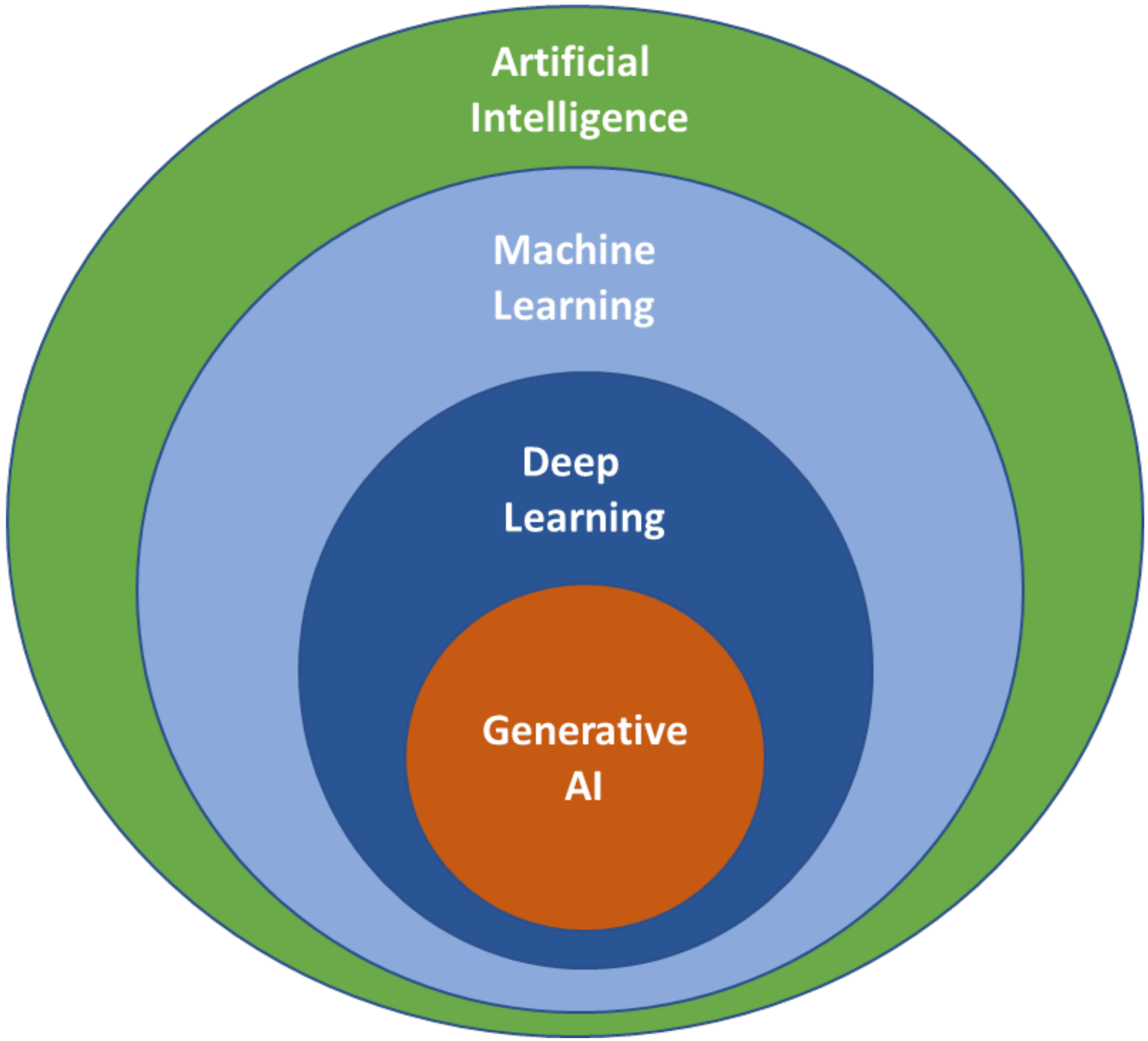
# AI for Technical Team

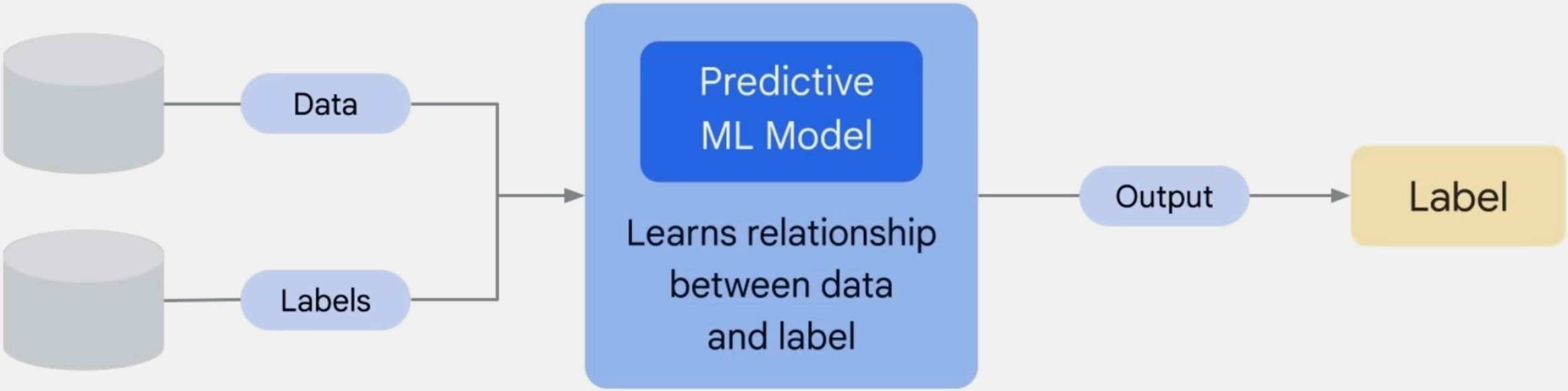
สยามชำนาญกิจ
SIAM CHAMNAN KIT

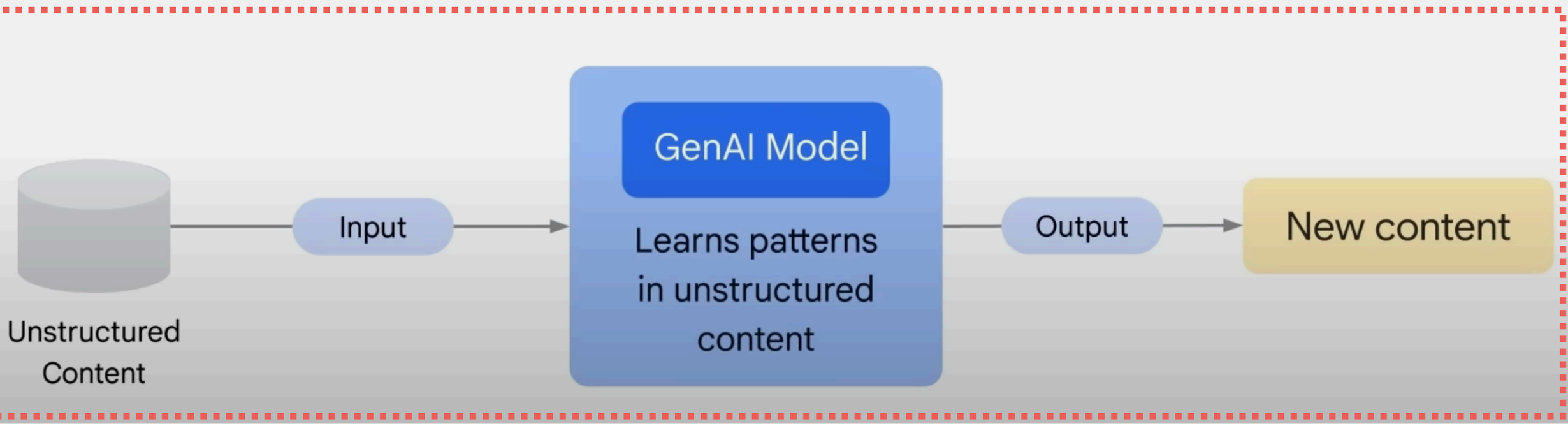
SHU HA RI
SOFTWARE DEVELOPMENT SCHOOL

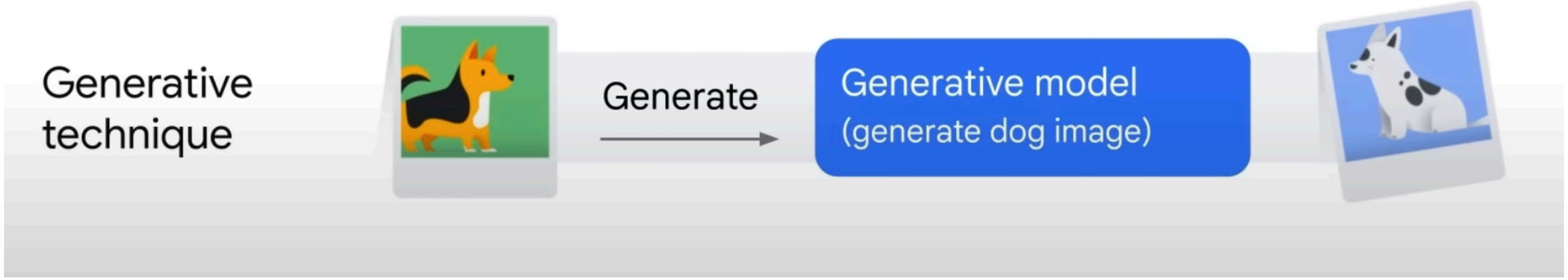


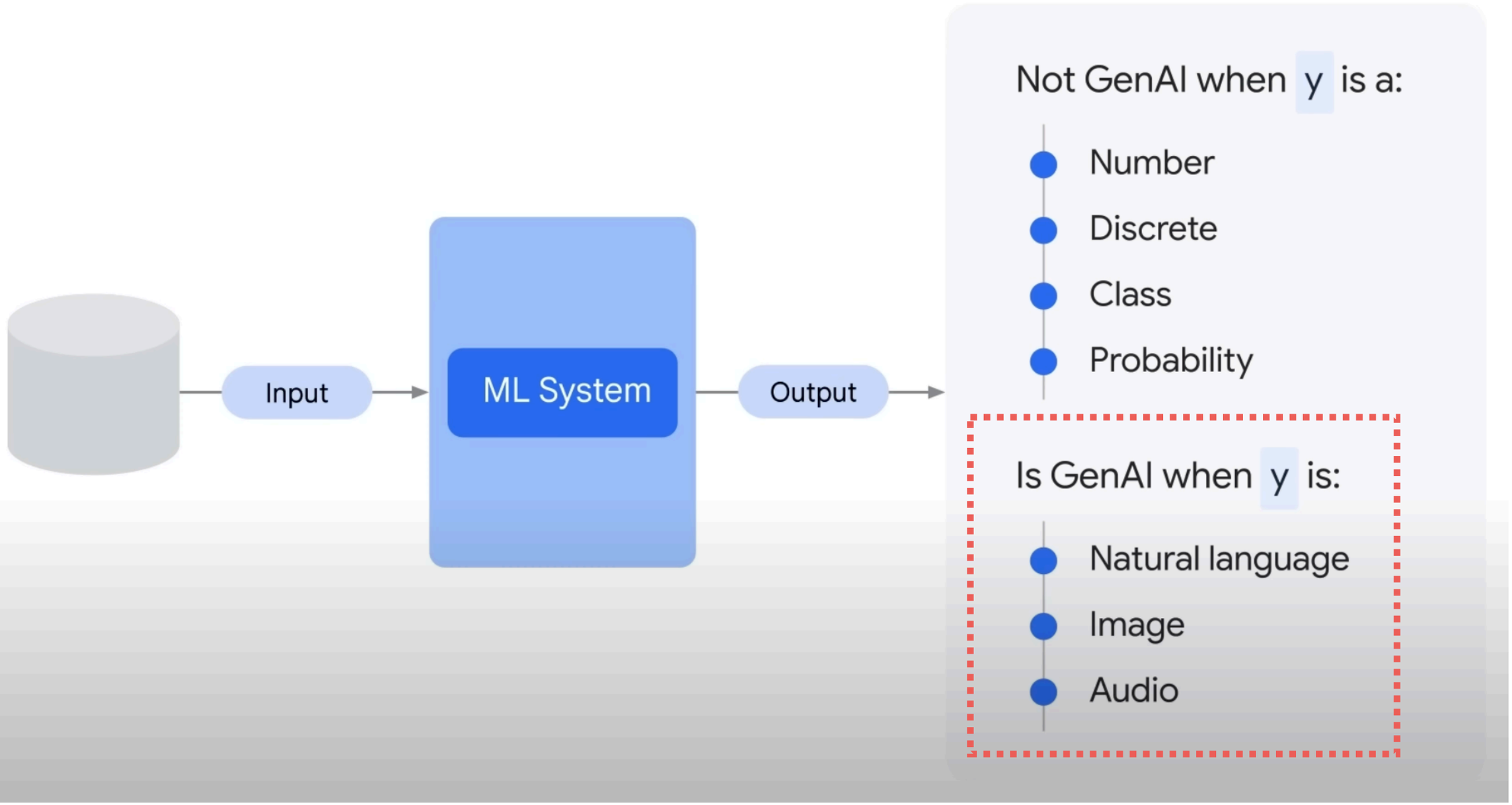
# Generative AI

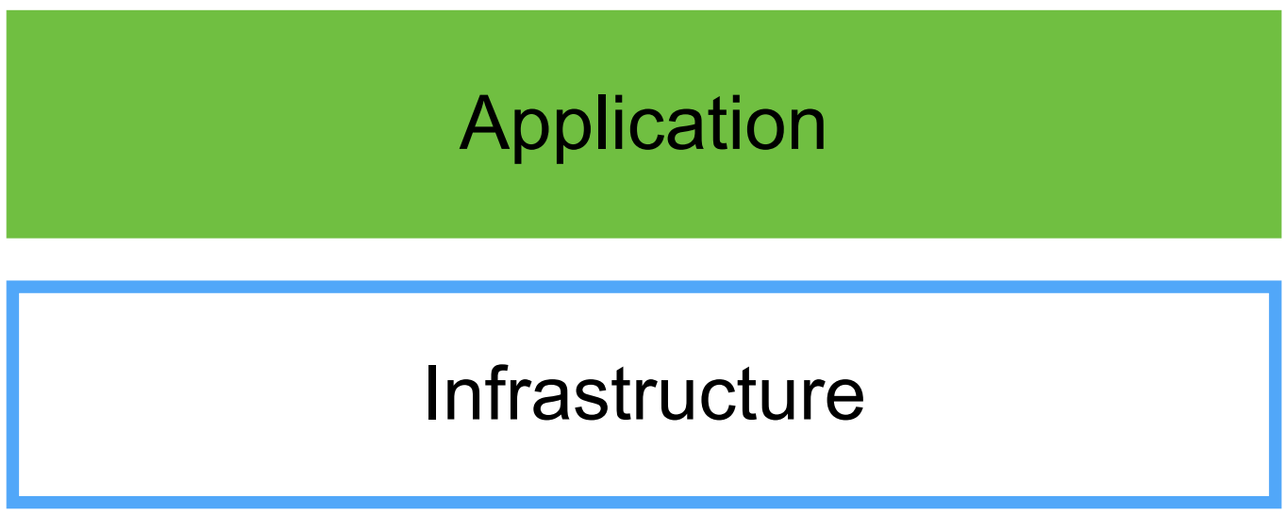








# Application
# Tool chains

# Tool chains category

Assist tasks

Interaction modes

Prompt composition

Properties of model

# Assist tasks

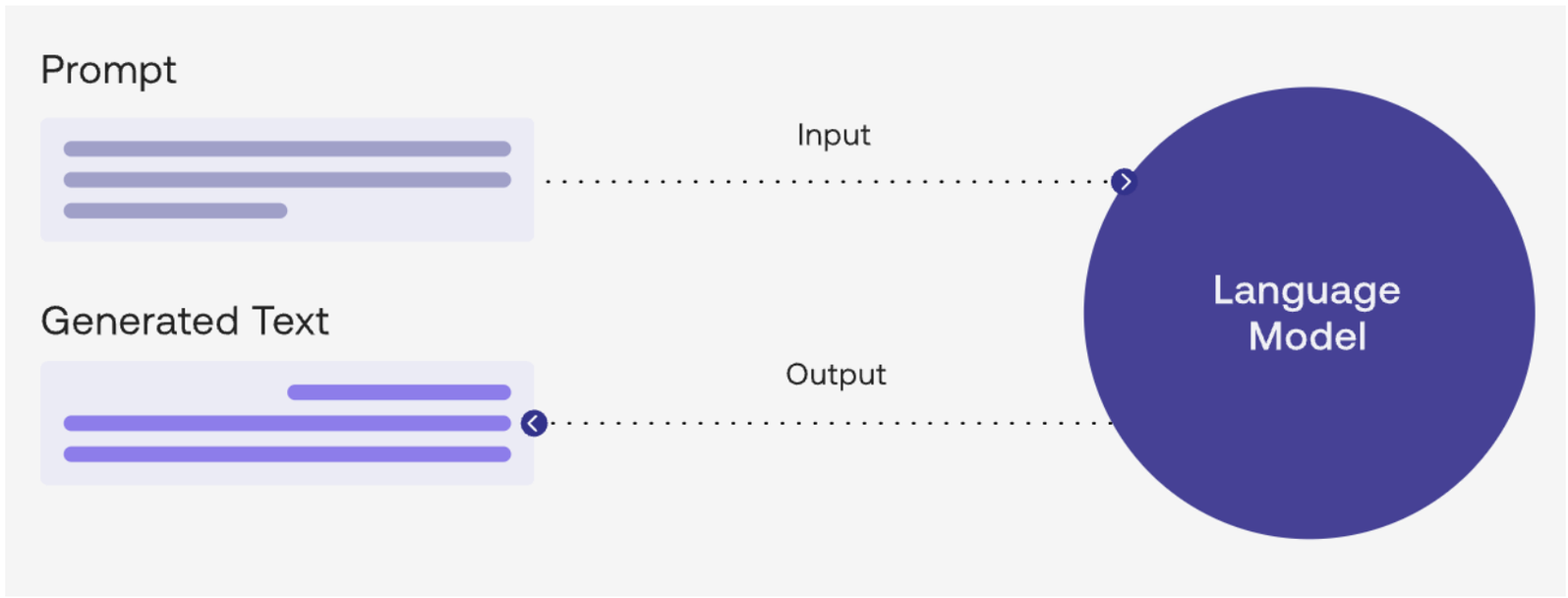
Finding information faster in context
Generating code
Reasoning about code
Transforming code into something ..

# Interaction modes

Chat interfaces
In-line assistance (typing in code editor)
CLI

# Prompt composition

Prompt engineering
Compose prompt from user input and context



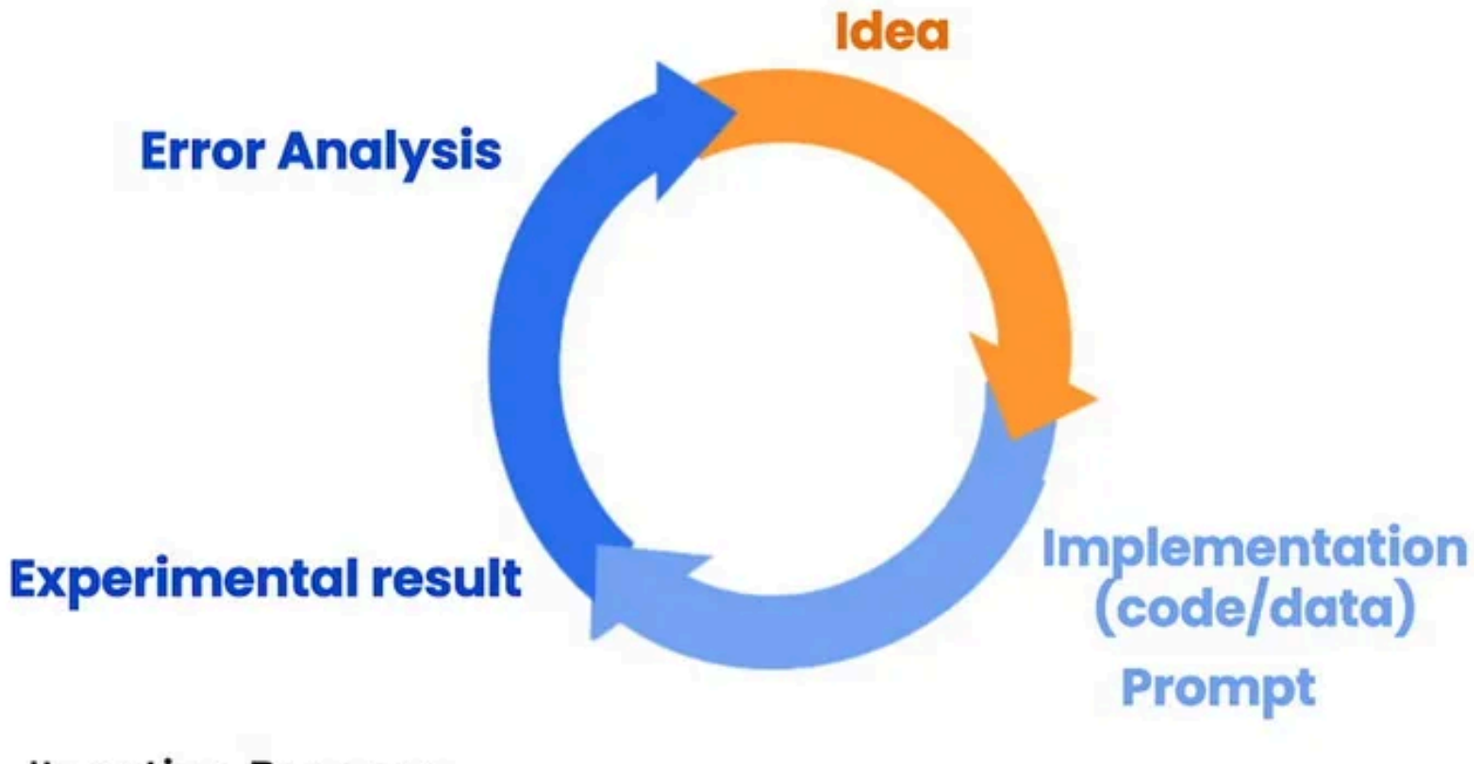

https://platform.openai.com/docs/guides/prompt-engineering

# Better Prompt

Write clear instructions
Provide reference text
Split complex tasks into simpler subtasks
Give the model time to think
Testing and improve …

https://platform.openai.com/docs/guides/prompt-engineering

# Iterative Prompt Development




Iterative Process

- Try something
- Analyze where the result does not give what you want
- Clarify instructions, give more time to think
- Refine prompts with a batch of examples

https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/

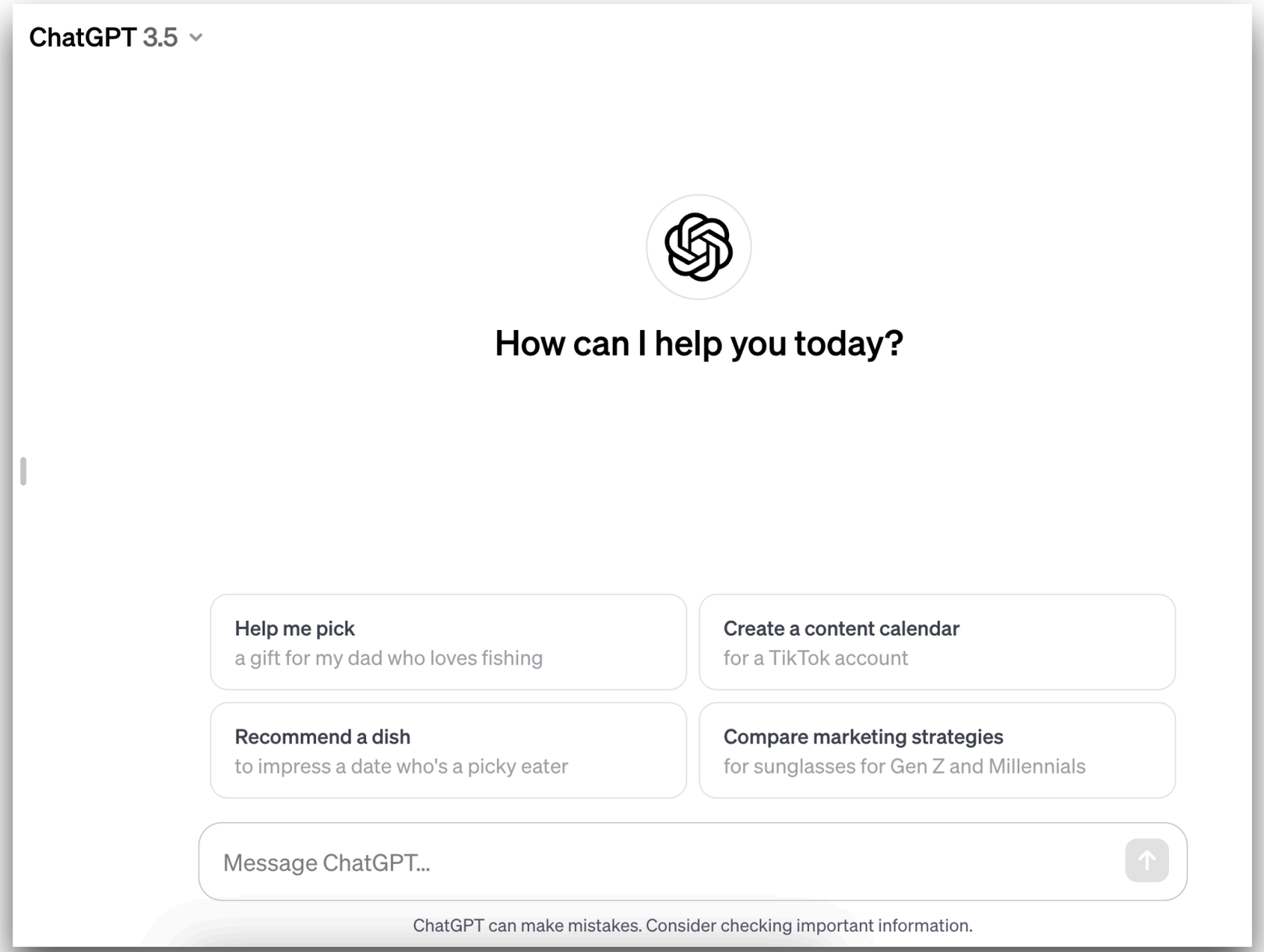
# List of tools

| Tool | Task | Interaction |
|---|---|---|
| **ChatGPT/Bard/Bing** | All | Chat |
| **GitHub Copilot** | Code generation | In-line assistance |
| **GitHub Copilot Chat** | All | Chat |
| **GPT Engineer** | Code generation | CLI |
| **WireGen** | Wireframe generation | Chat |
| **Make real** | UI generation | UI |
| **V0.dev** | UI generation | Chat |

# Chat and Search

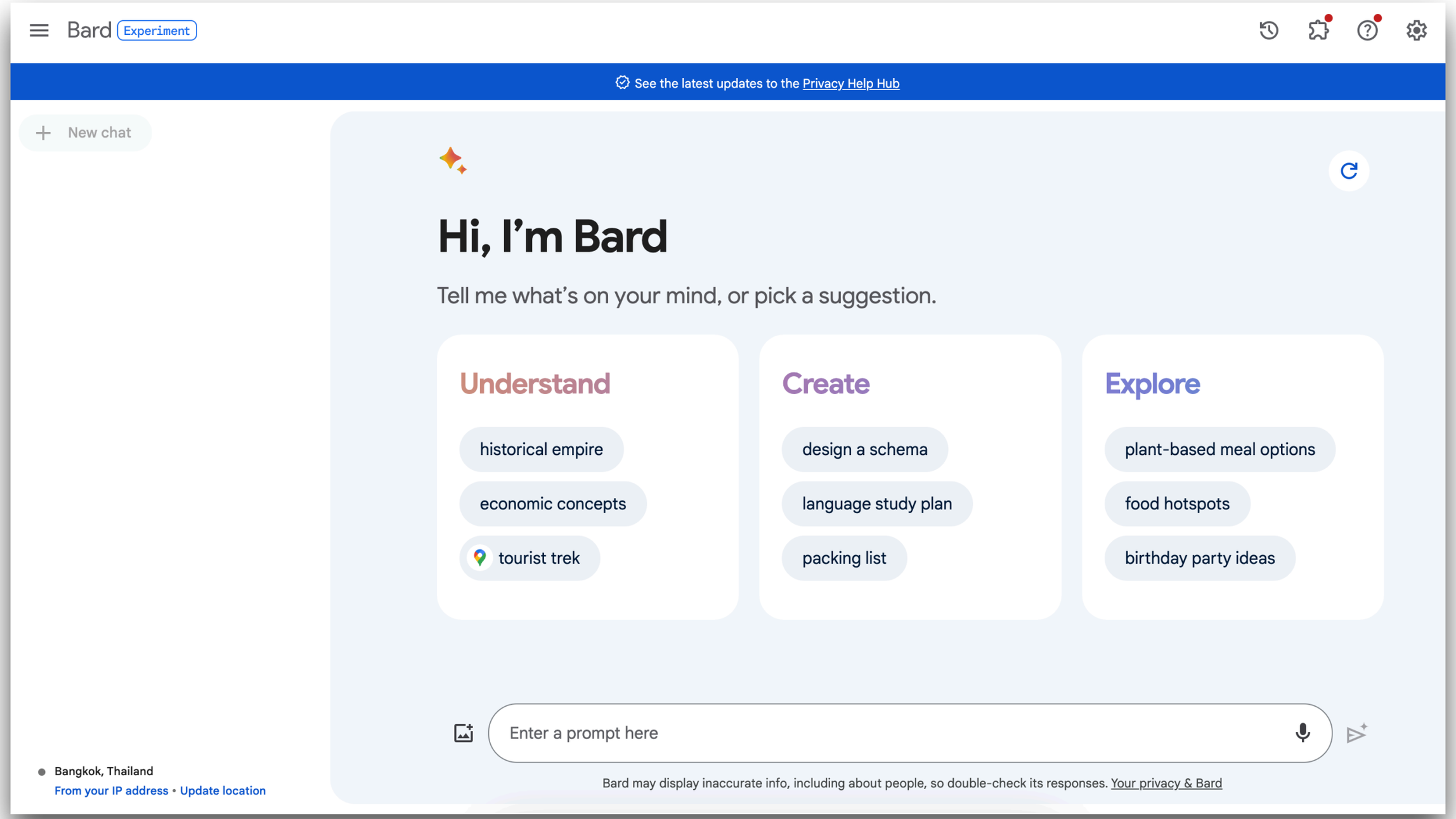
Bard vs CHAT-GPT vs BingAI

# ChatGPT from OpenAI



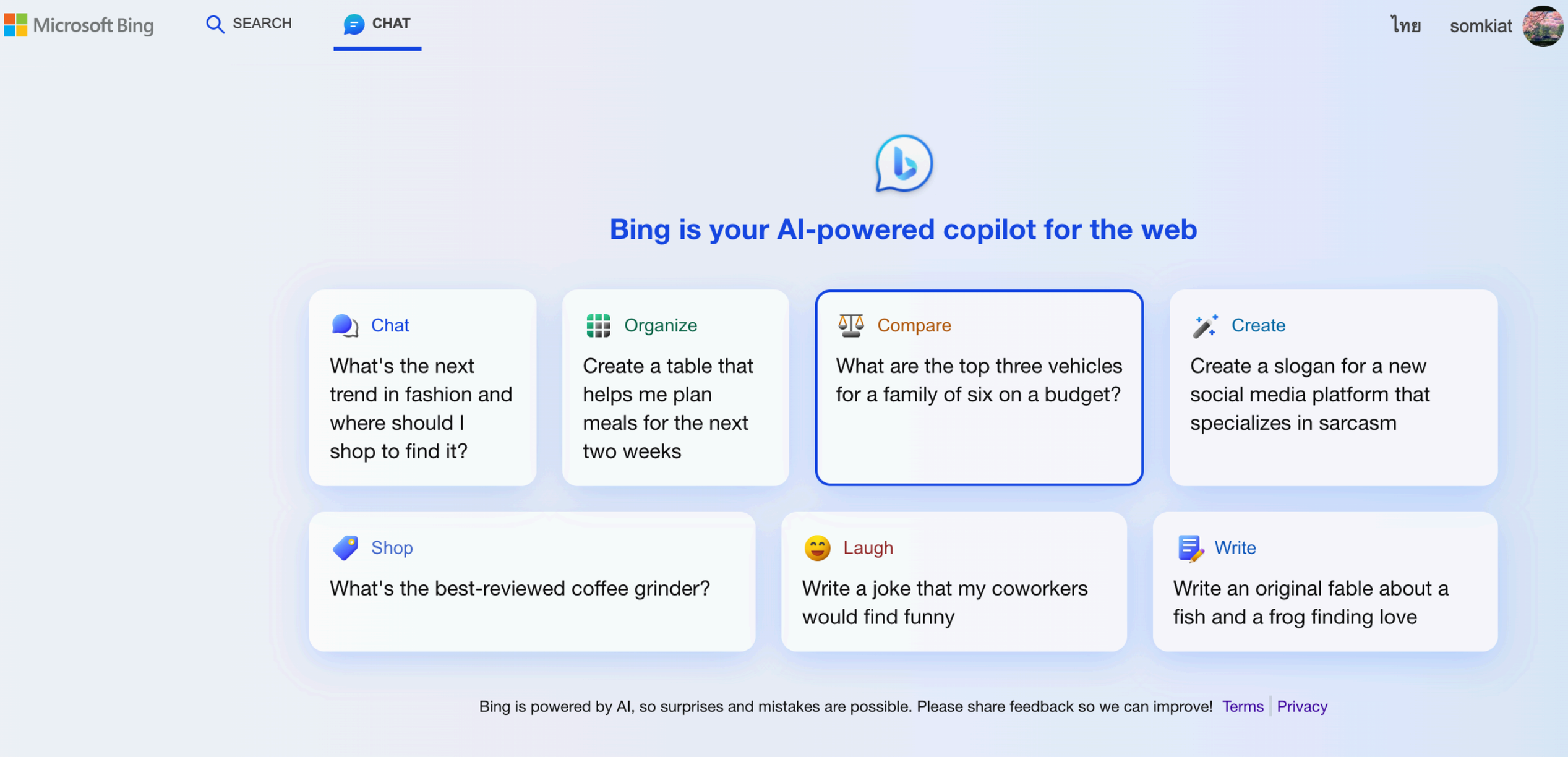

https://chat.openai.com/

# Google Bard



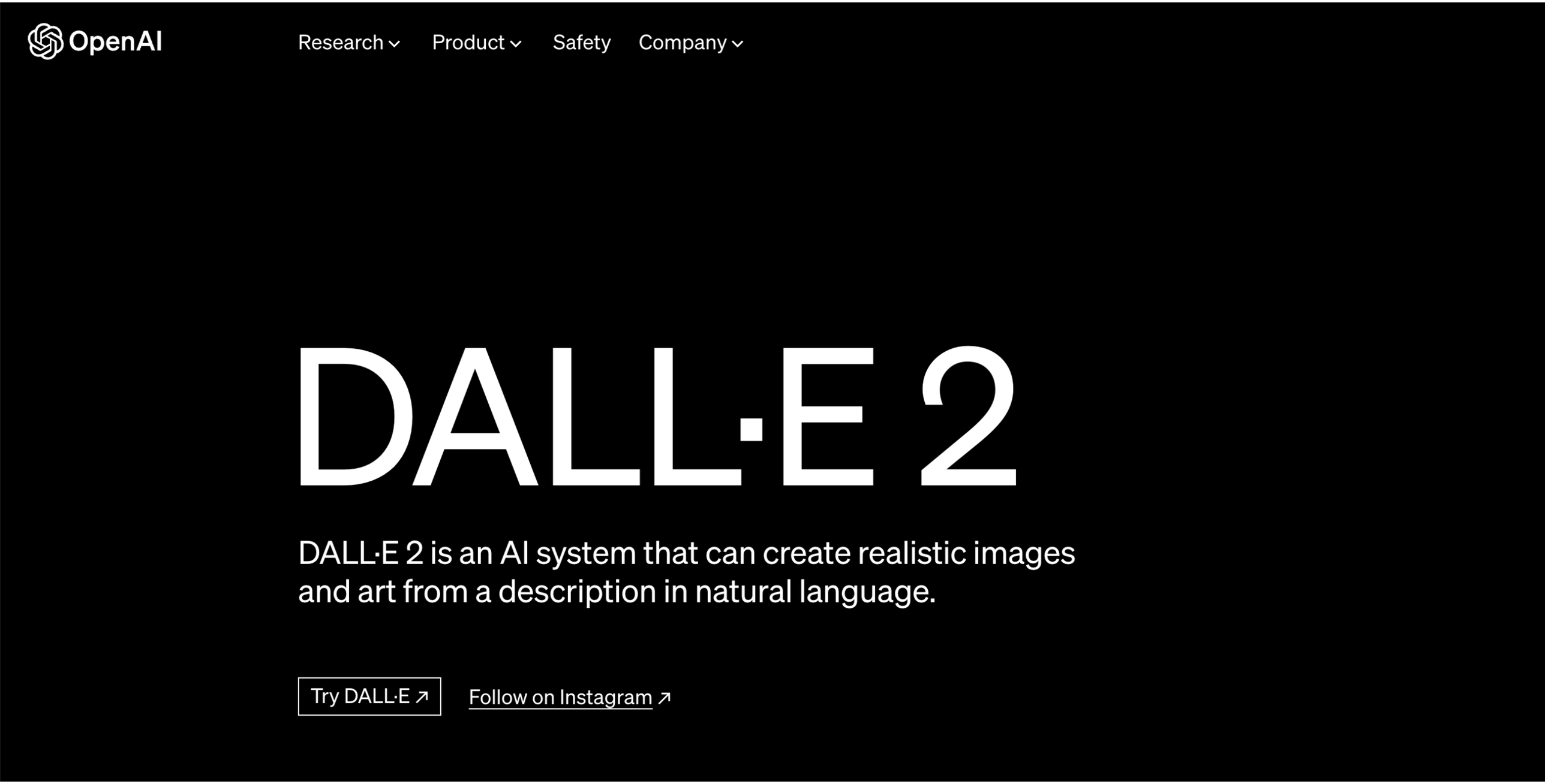

https://bard.google.com/chat

# Microsoft Bing



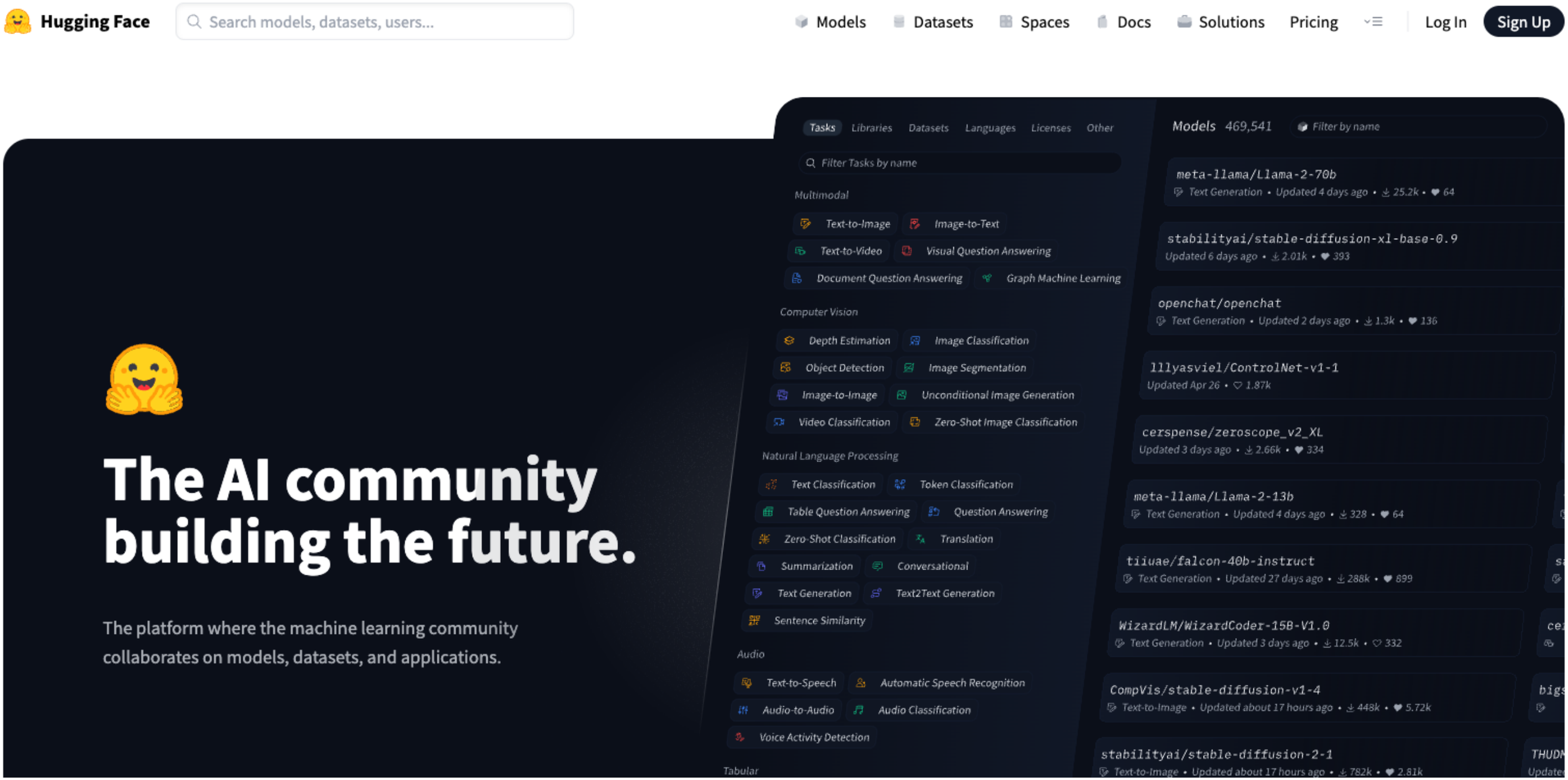

https://www.bing.com/search?q=Bing+AI&showconv=1&FORM=hpcodx

# DALL.E 2 from OpenAI




https://openai.com/dall-e-2

# Hugging Face

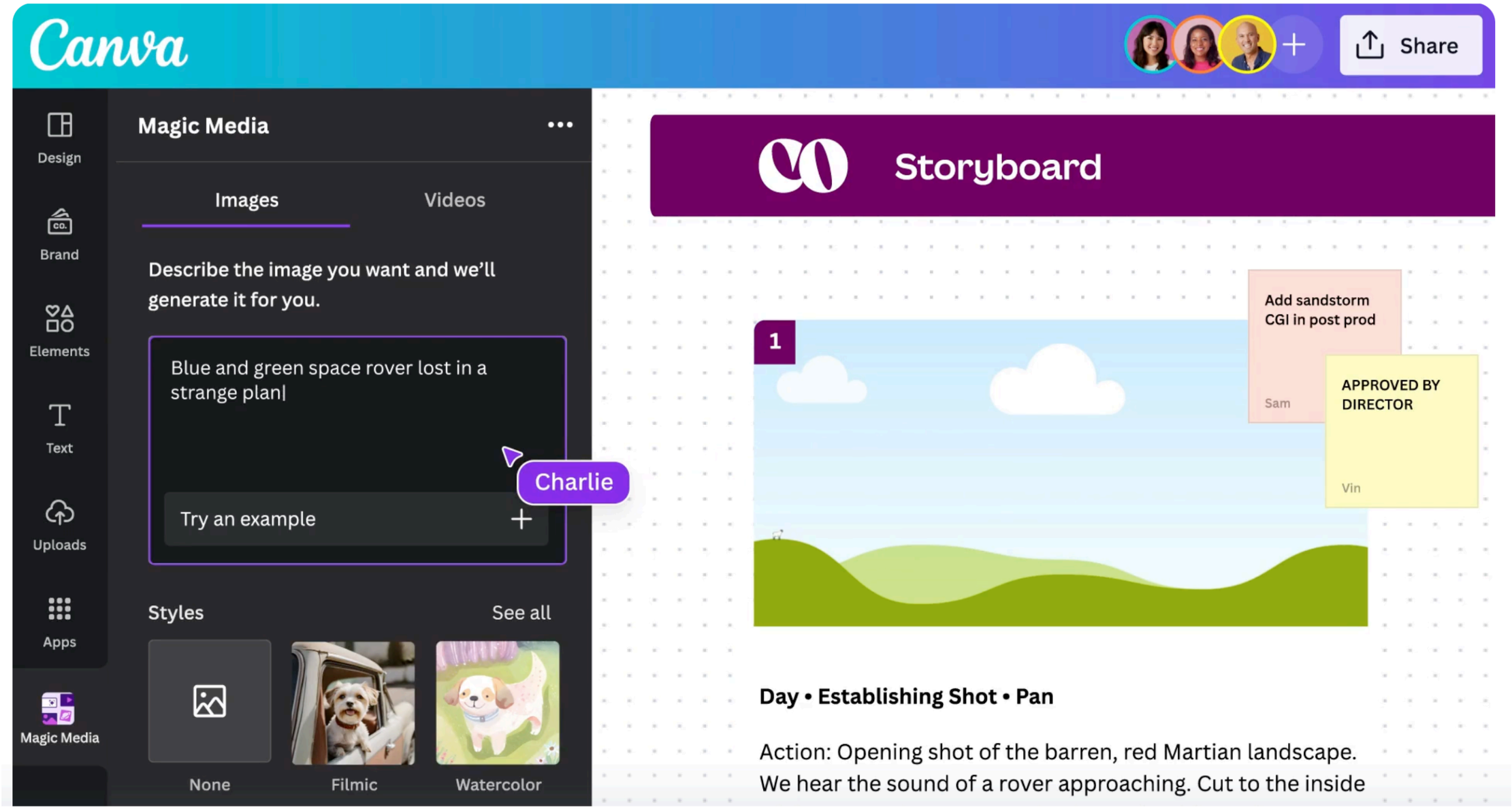



https://huggingface.co/

# Mid Journey

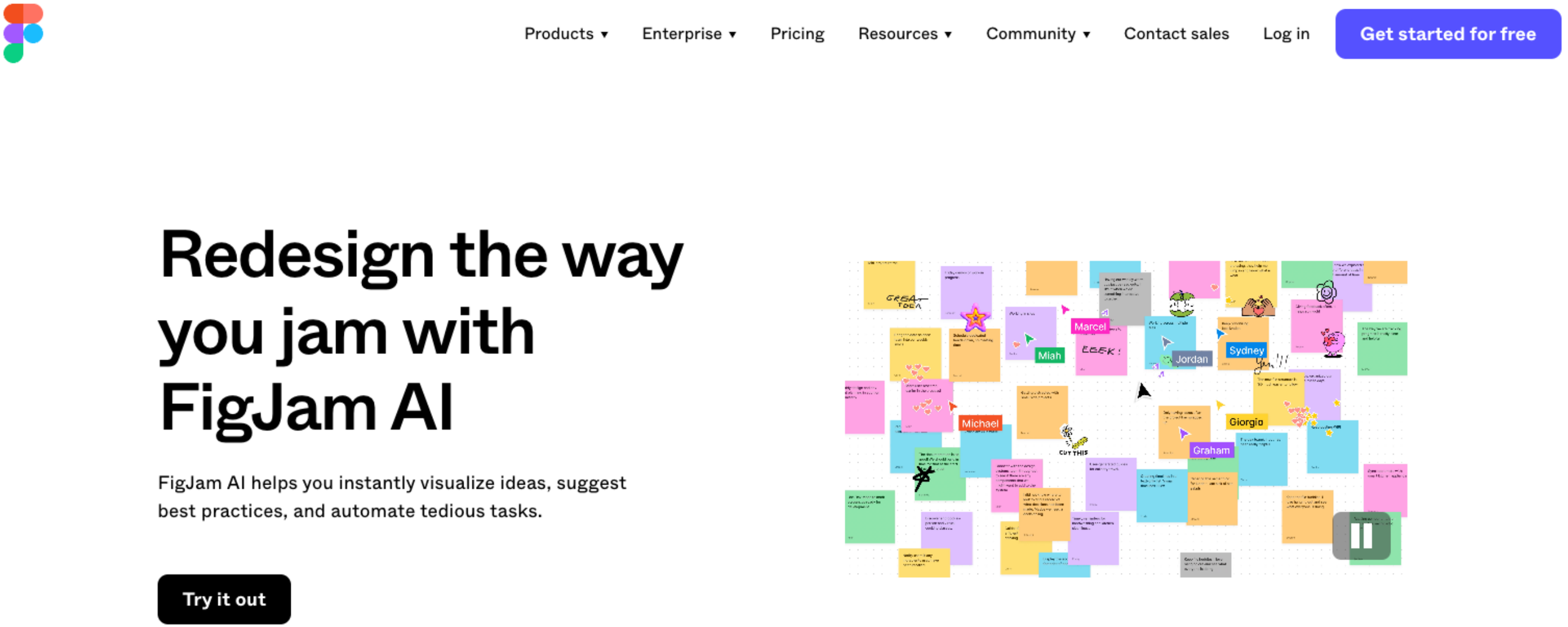
https://www.midjourney.com/

# Canva



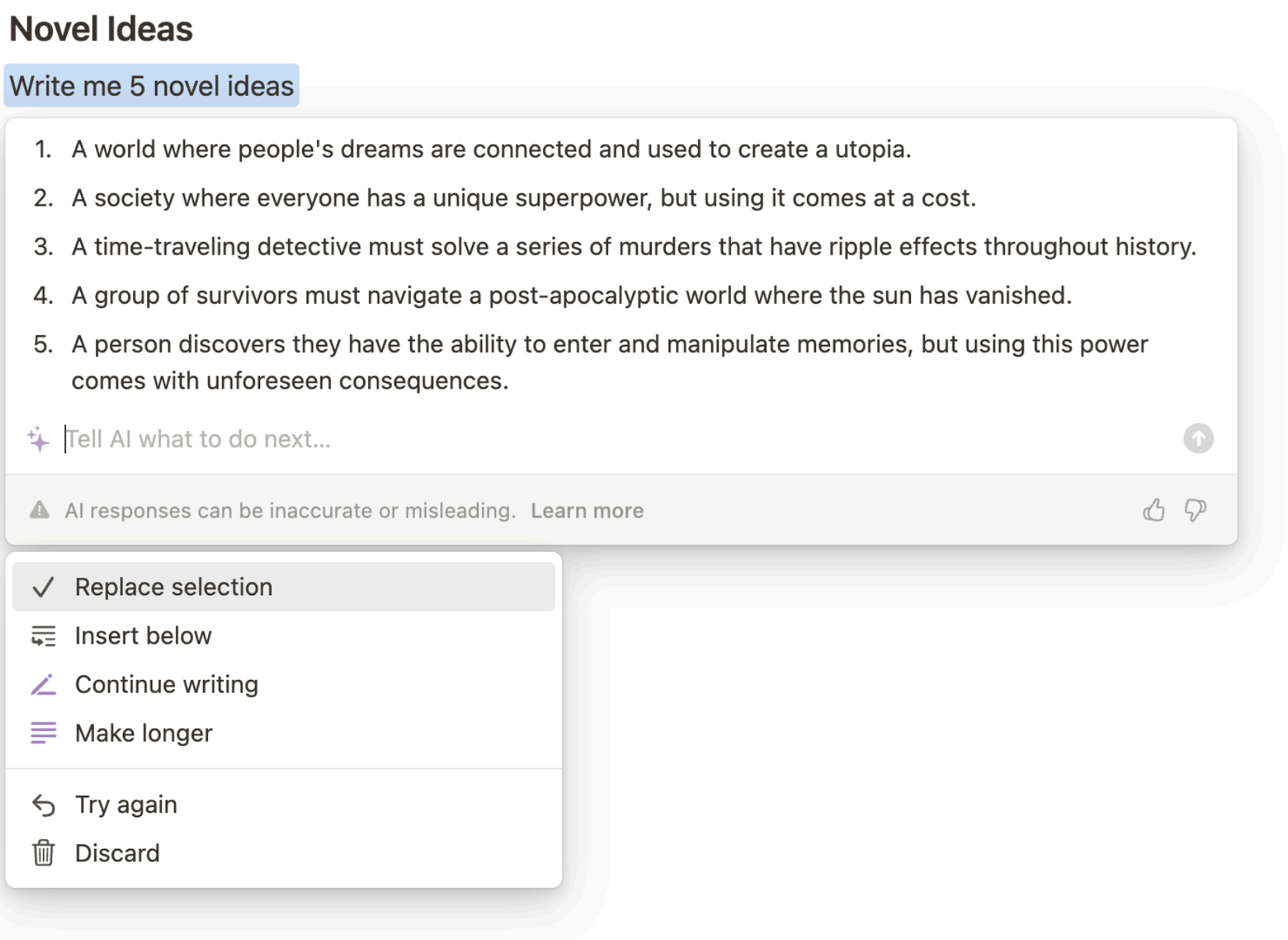

https://www.canva.com/ai-image-generator/

# FigJam AI




https://www.figma.com/figjam/ai/

# Notion AI




https://www.notion.so/product/ai

# Demo

https://github.com/up1/workshop-ai-with-technical-team

# Using GitHub Copilot Chat correlates with better code quality

85% of developers felt more confident in their code quality when authoring code with GitHub Copilot and GitHub Copilot Chat

**85%**

Code reviews were more actionable and completed 15% faster than without GitHub Copilot Chat

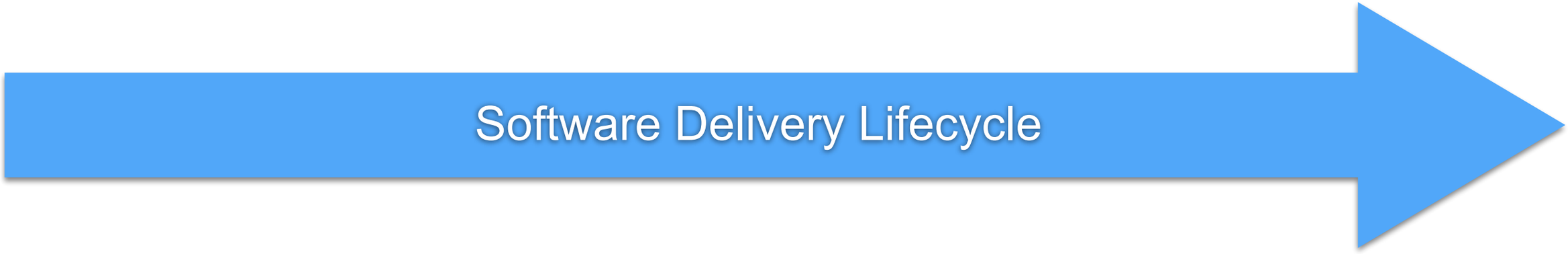
**15%**

88% of developers reported maintaining flow state with GitHub Copilot Chat

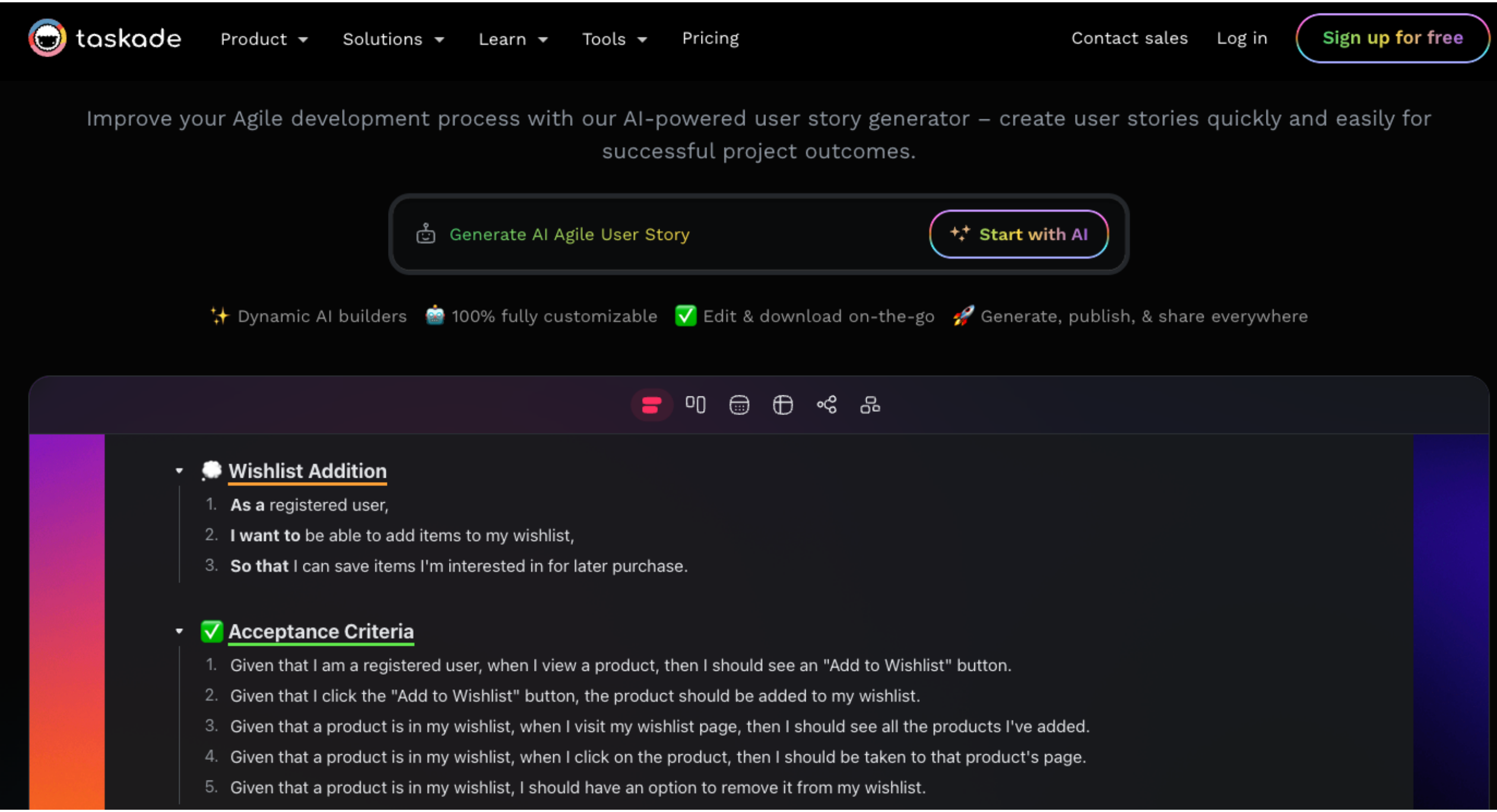
**88%**

# Assist tasks

Requirement | Design | Develop | Testing | Deploy



# User Story Generator

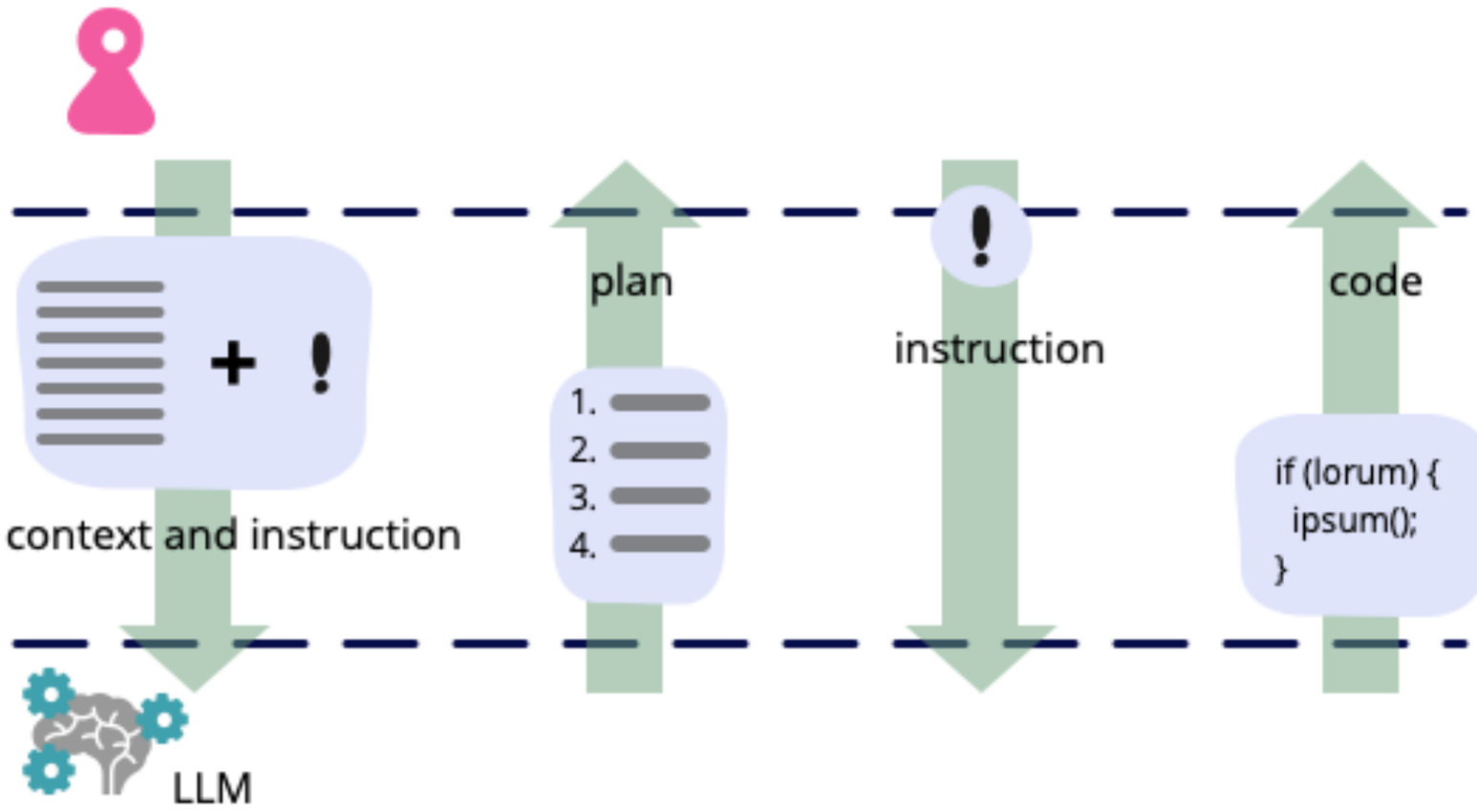



https://www.taskade.com/generate/project-management/agile-user-story

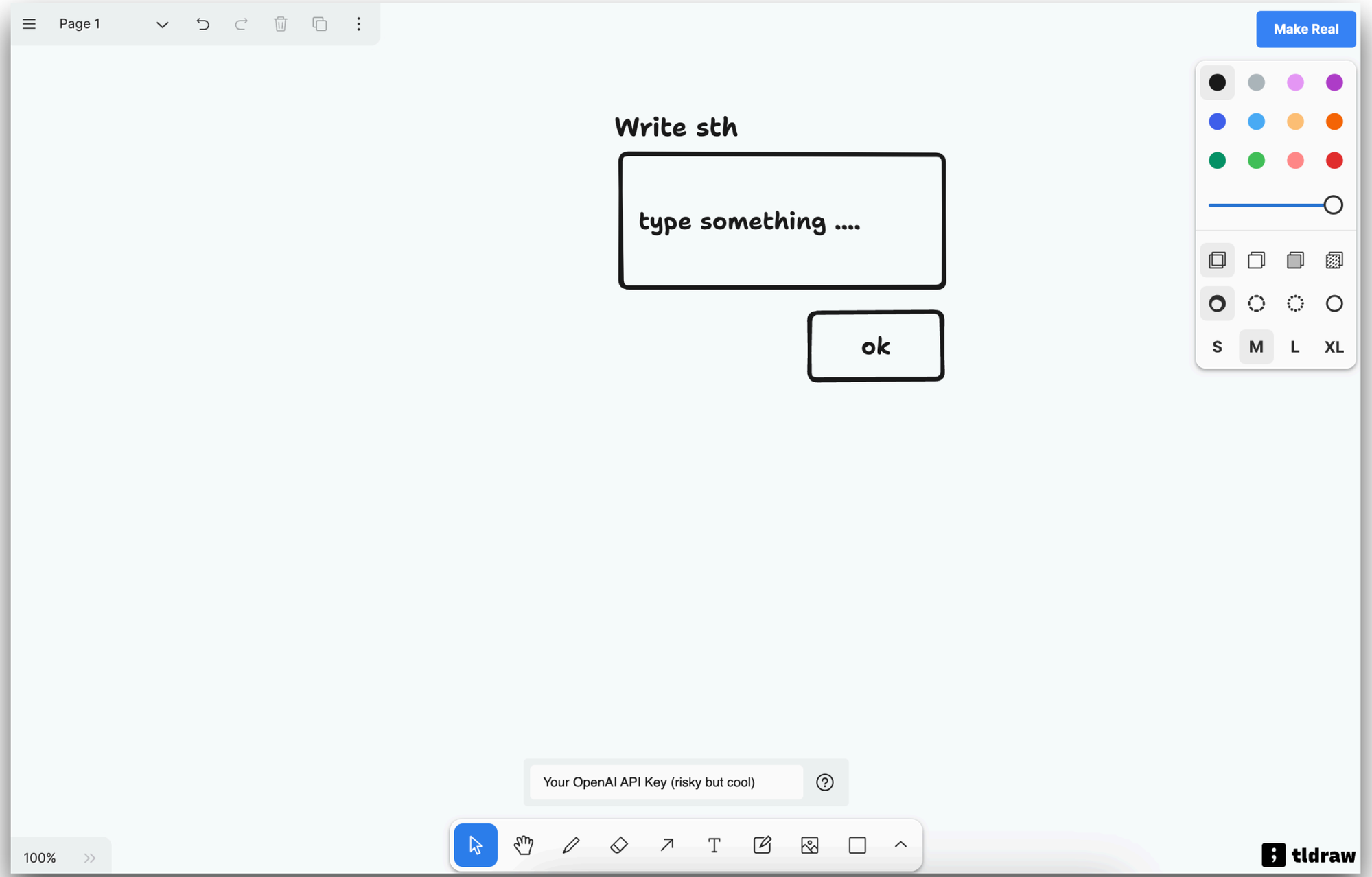
# User Story Generator


https://userstorygenerator.ai/

# Development

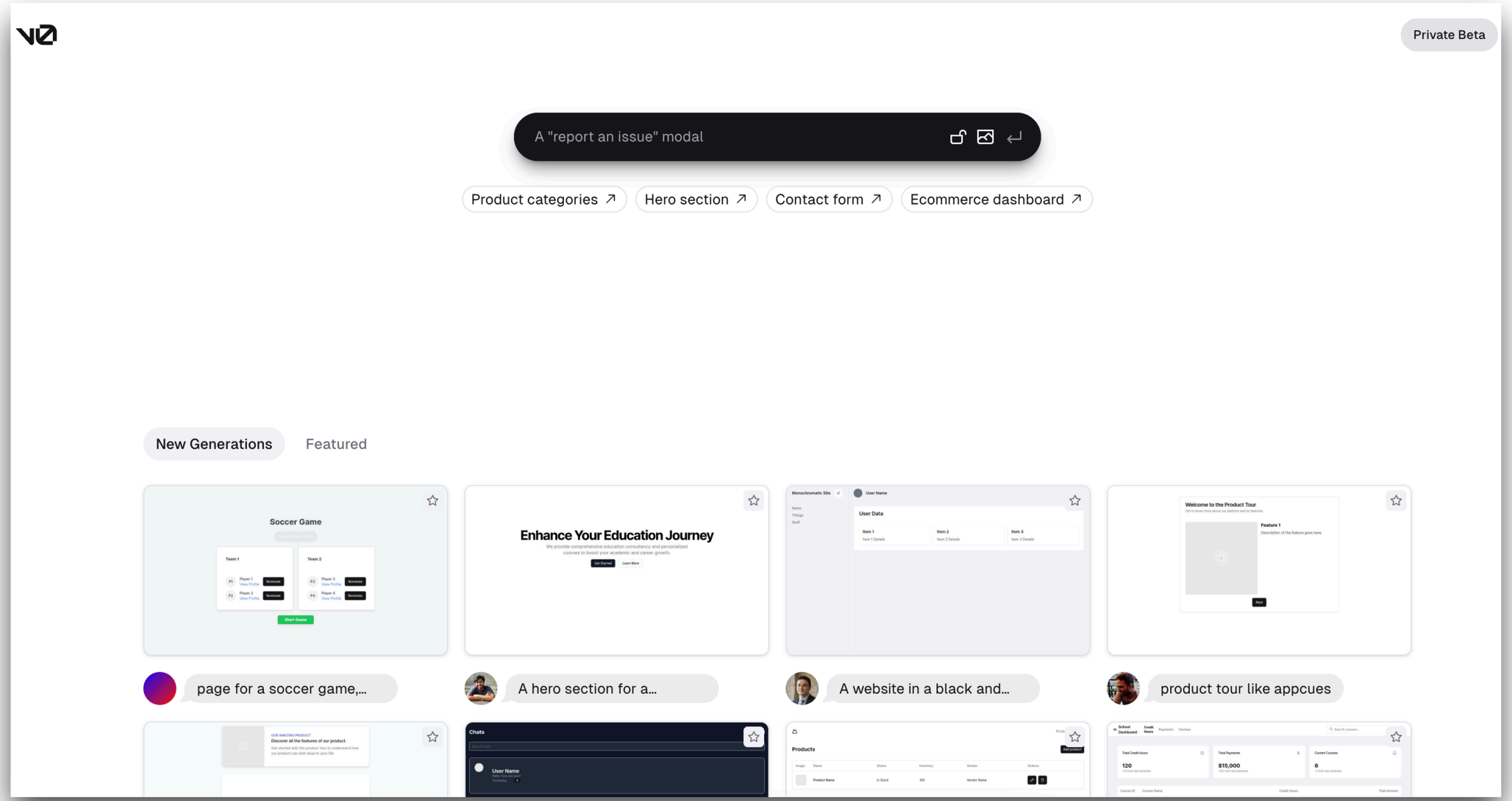


# Make Real




https://github.com/tldraw/make-real

# V0.dev

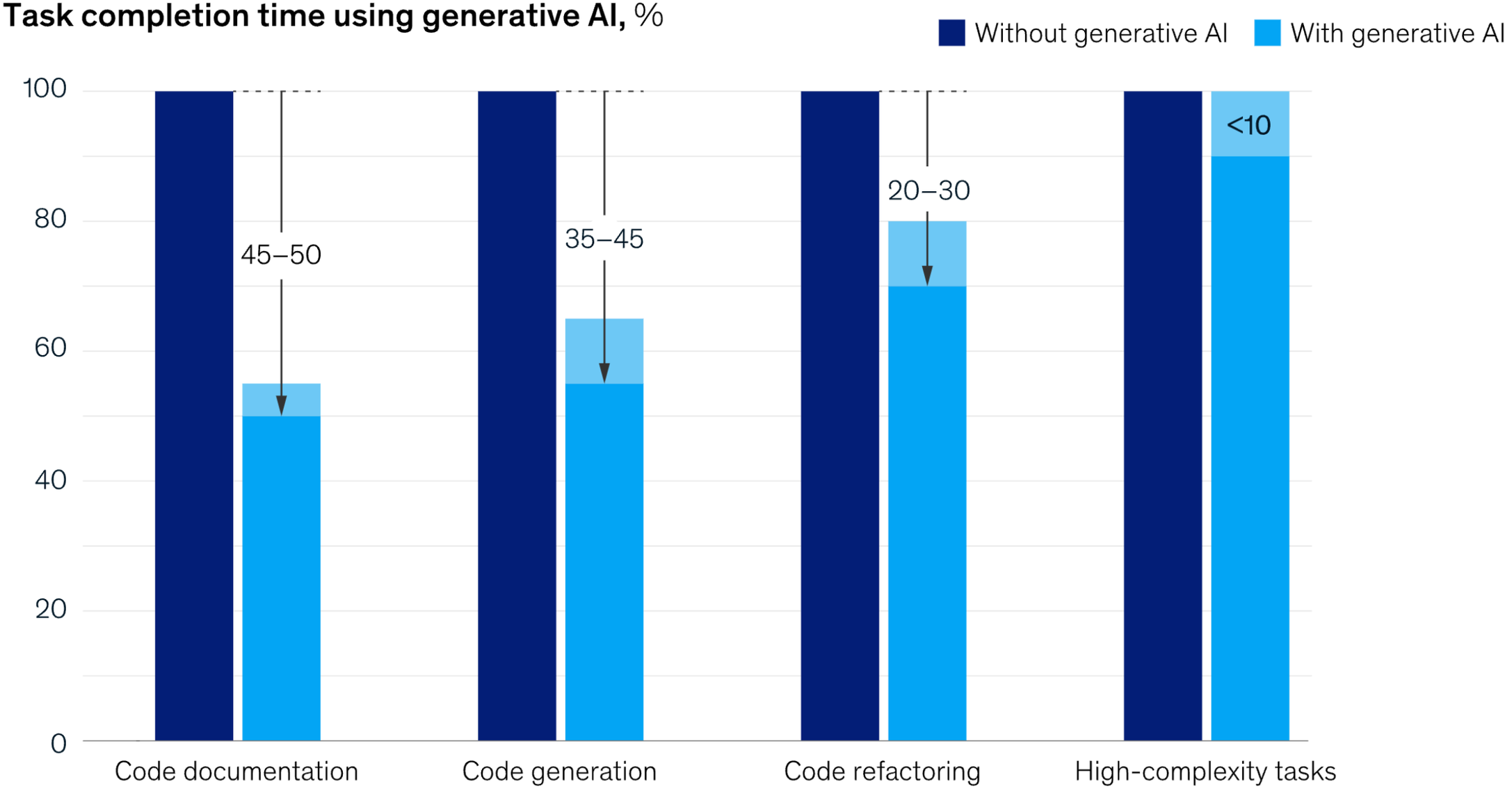



https://v0.dev/

# Development

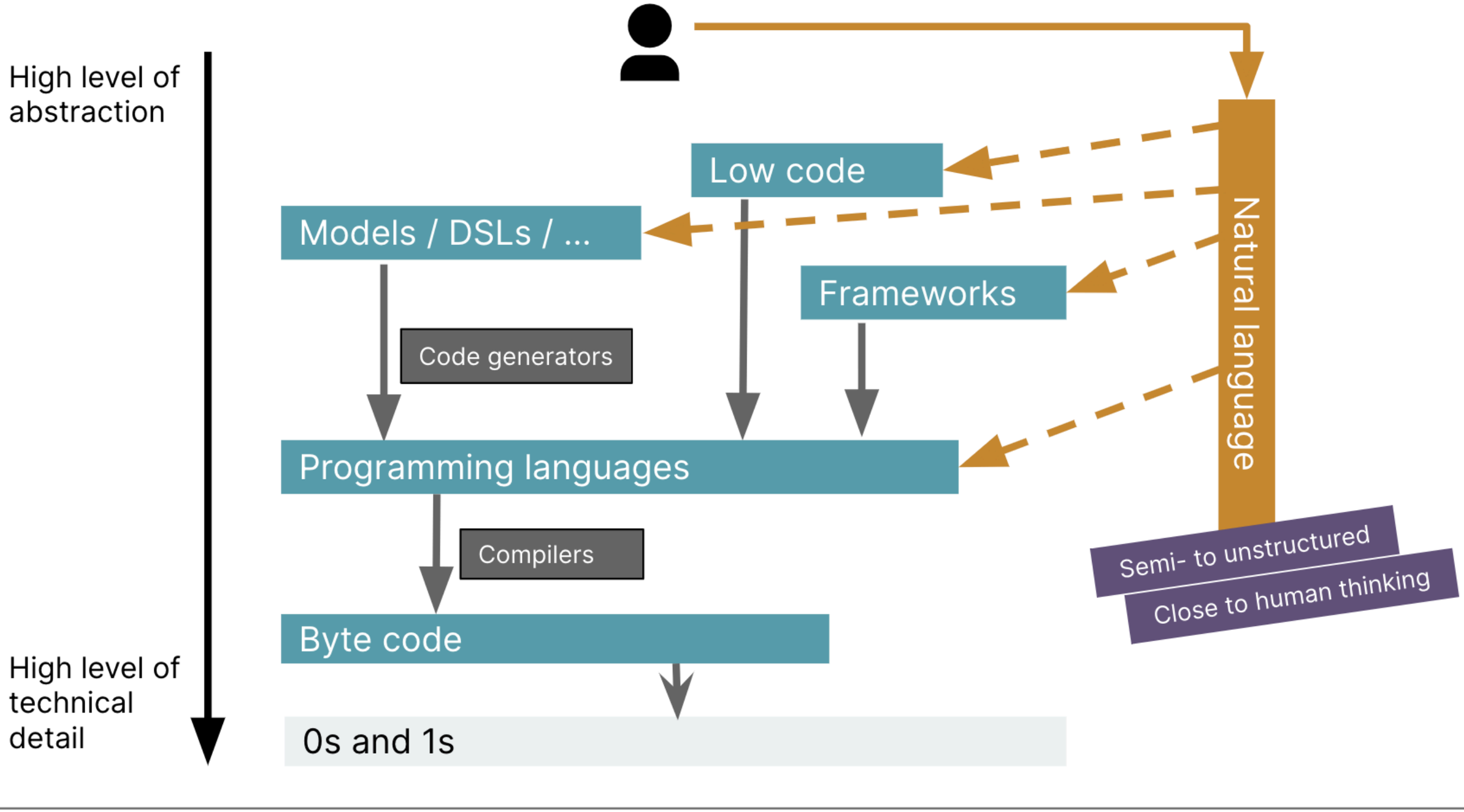
## Generative AI can increase developer speed, but less so for complex tasks.

**Task completion time using generative AI,** %

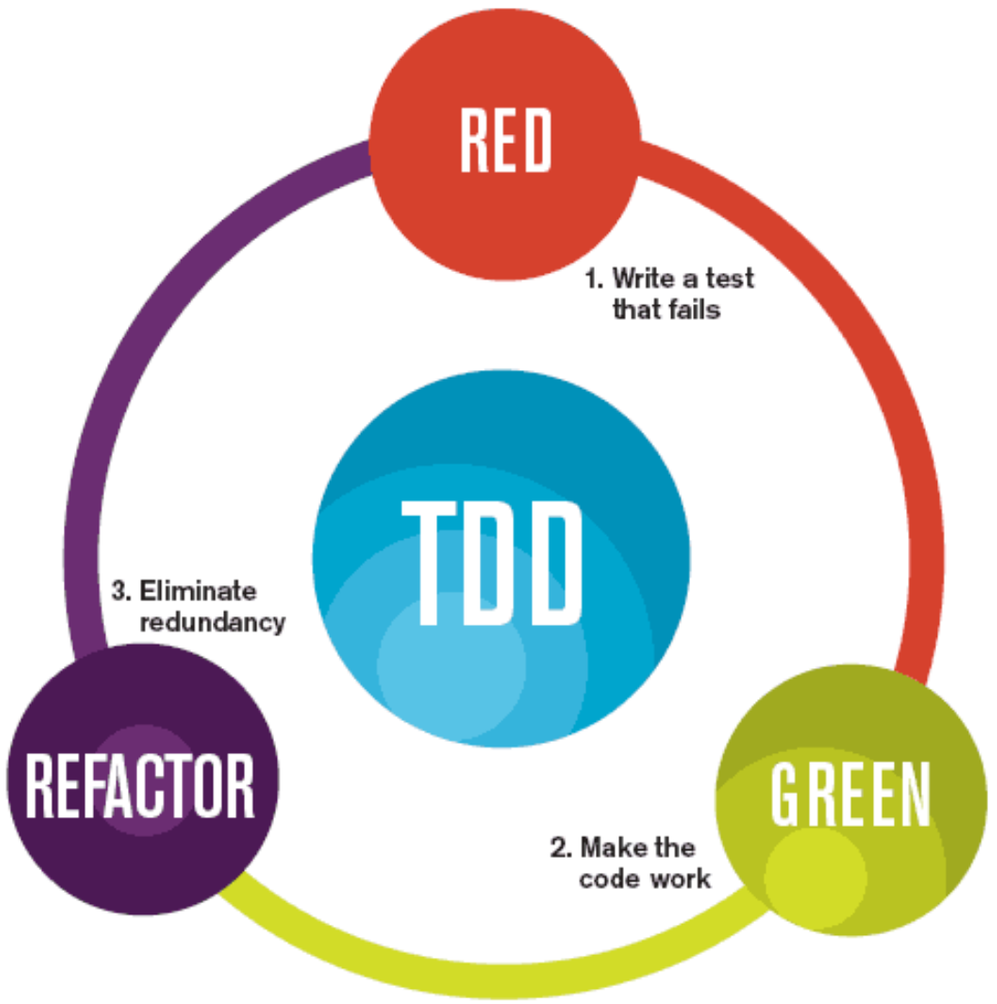



https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai

# Development



# TDD




The mantra of Test-Driven Development (TDD) is "red, green, refactor."

# Tips

Understand what you want

Modular approach

Clear and Precise inputs

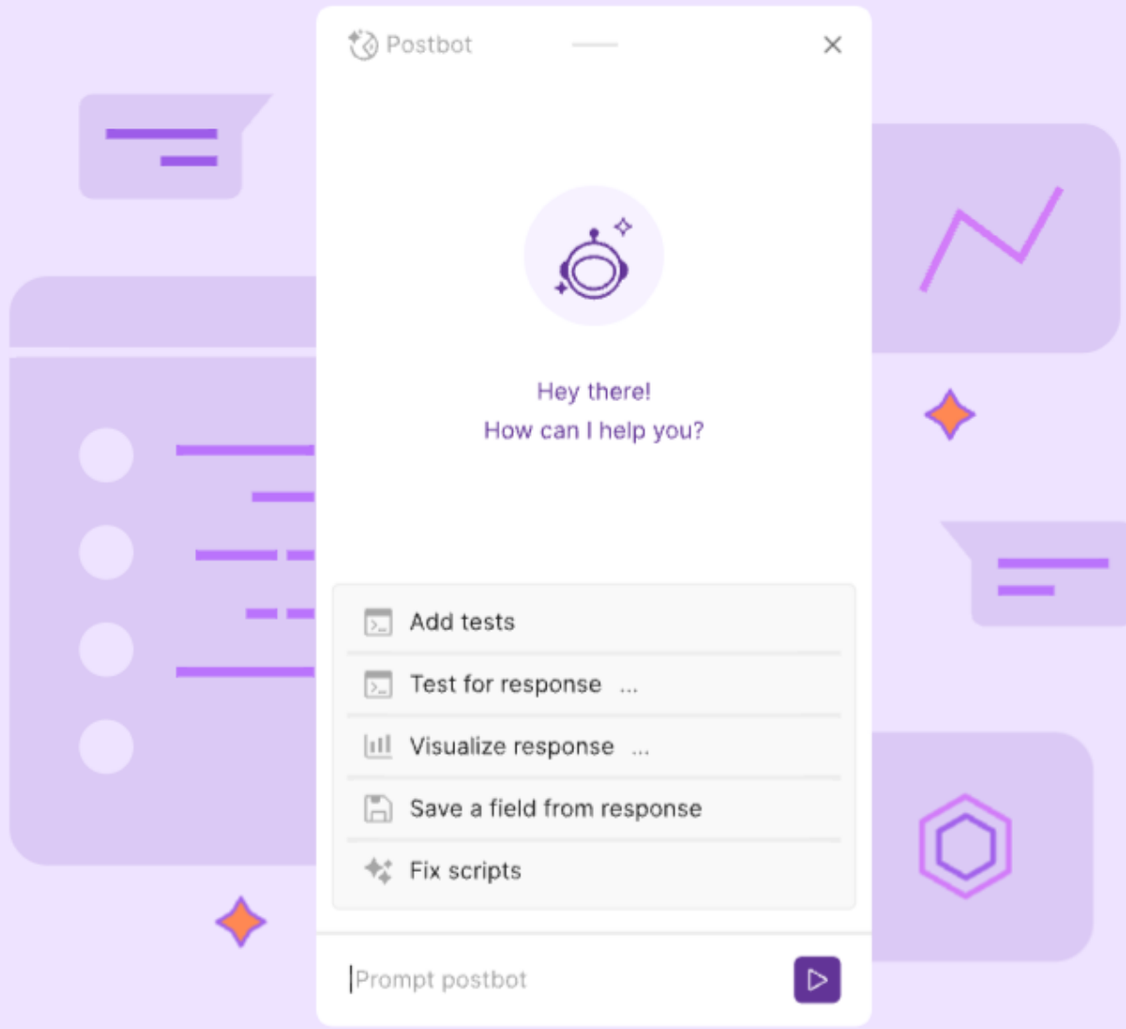
Make sure you understand the code

# Testing with Postman

**Postbot**, our AI-powered assistant, will supercharge your API development.

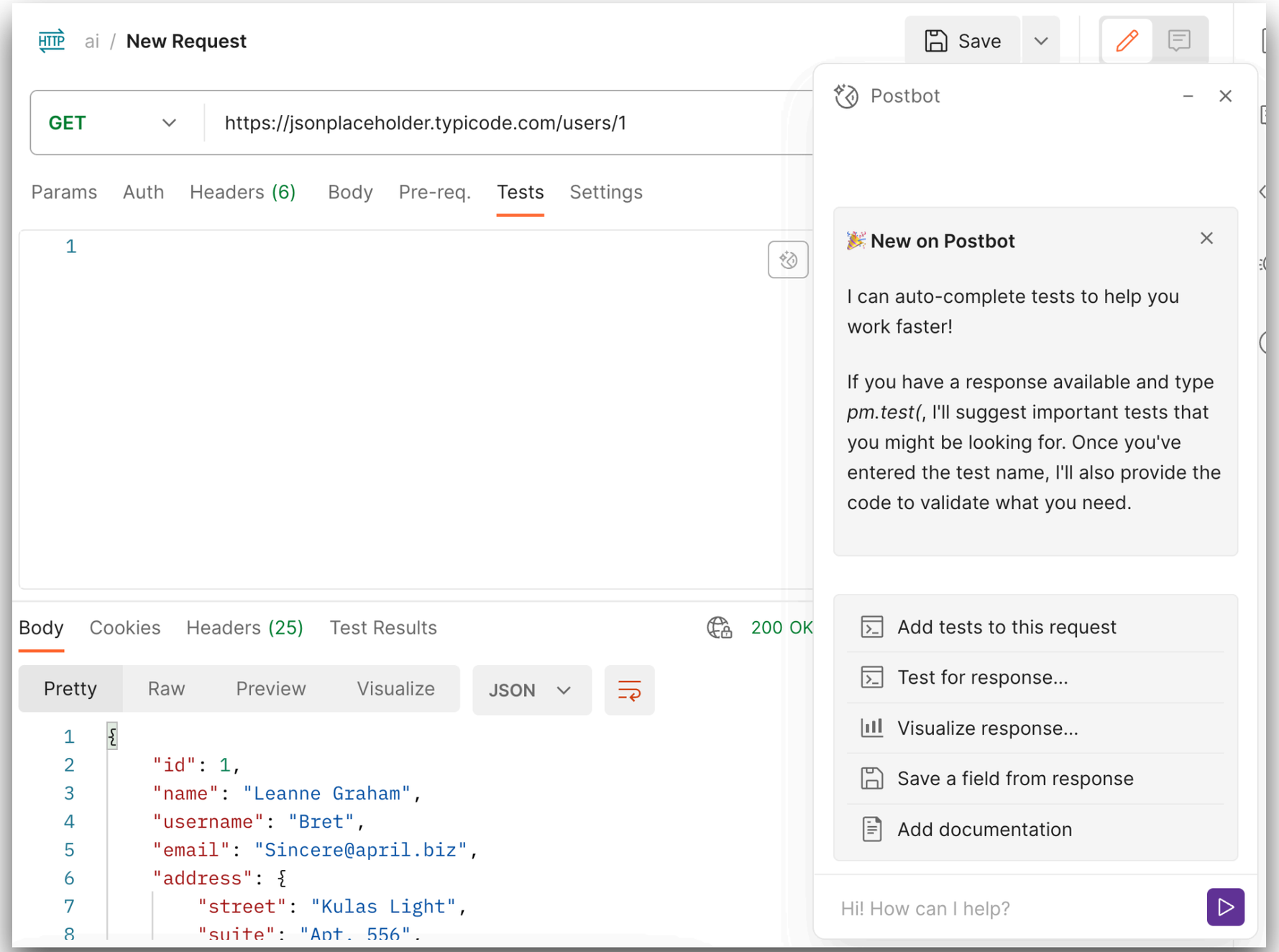
Speed up your most common API development workflows with natural-language input, conversational interactions, and contextual suggestions.

Get Started




https://www.postman.com/product/postbot/

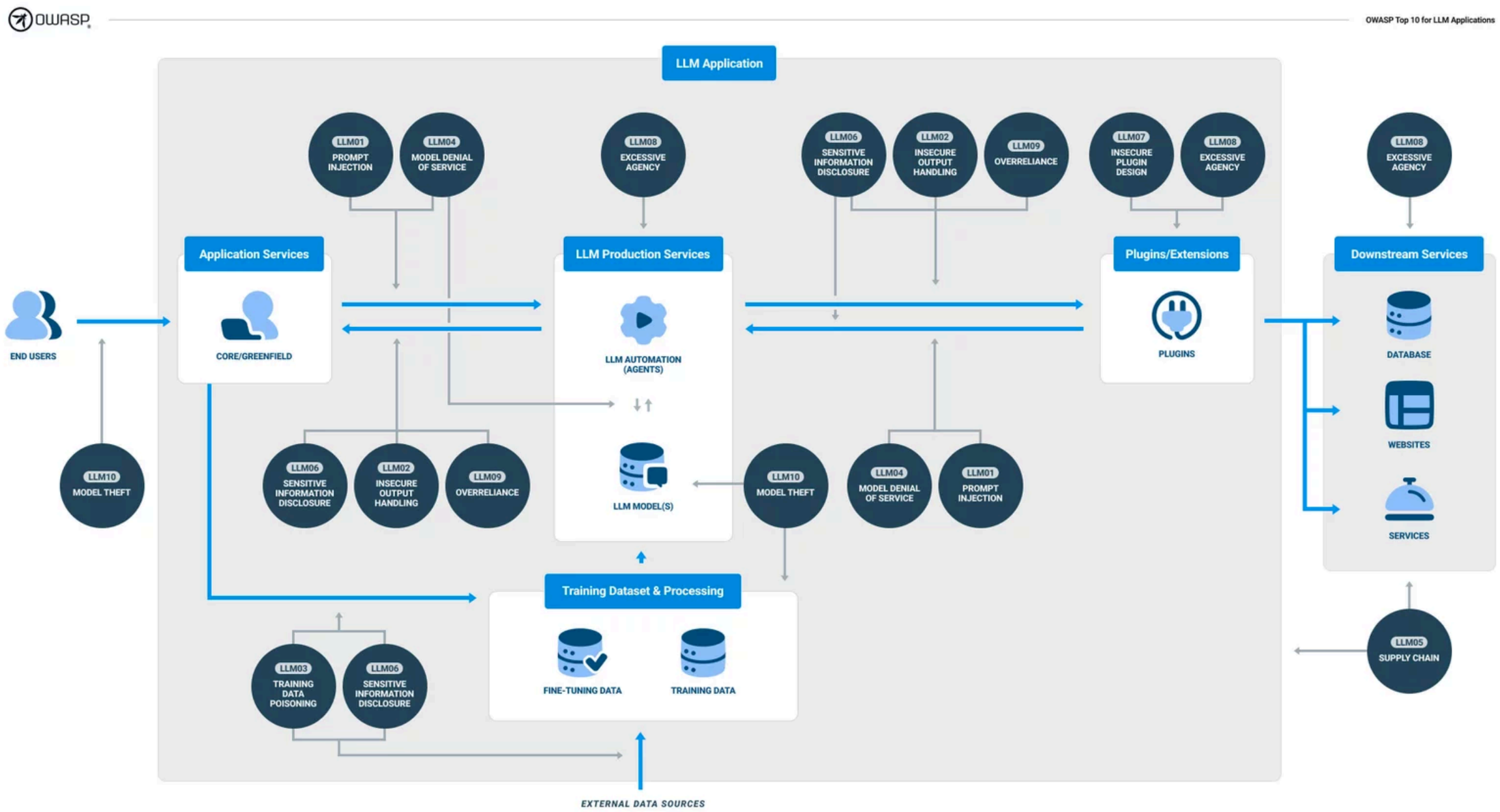
# PostBot




https://www.postman.com/product/postbot/

# Security !!

# OWASP Top 10 for LLM app




https://llmtop10.com/

# OWASP Top 10 for LLM app

LLM01

## Prompt Injection

Crafty inputs can manipulate a Large Language Model, causing unintended actions. Direct injections overwrite system prompts, while indirect ones manipulate inputs from external sources.

LLM02

## Insecure Output Handling

This vulnerability occurs when an LLM output is accepted without scrutiny, exposing backend systems. Misuse may lead to severe consequences like XSS, CSRF, SSRF, privilege escalation, or remote code execution.

LLM03

## Training Data Poisoning

This occurs when LLM training data is tampered, introducing vulnerabilities or biases that compromise security, effectiveness, or ethical behavior. Sources include Common Crawl, WebText, OpenWebText, & books.

LLM04

## Model Denial of Service

Attackers cause resource-heavy operations on Large Language Models leading to service degradation or high costs. The vulnerability is magnified due to the resource-intensive nature of LLMs and unpredictability of user inputs.

LLM05

## Supply Chain Vulnerabilities

LLM application lifecycle can be compromised by vulnerable components or services, leading to security attacks. Using third-party datasets, pre- trained models, and plugins can add vulnerabilities.

LLM06

## Sensitive Information Disclosure

LLM's may reveal confidential data in its responses, leading to unauthorized data access, privacy violations, and security breaches. It's crucial to implement data sanitization and strict user policies to mitigate this.

https://llmtop10.com/

# OWASP Top 10 for LLM app

LLM07

### Insecure Plugin Design

LLM plugins can have insecure inputs and insufficient access control. This lack of application control makes them easier to exploit and can result in consequences like remote code execution.

LLM08

### Excessive Agency

LLM-based systems may undertake actions leading to unintended consequences. The issue arises from excessive functionality, permissions, or autonomy granted to the LLM-based systems.

LLM09

### Overreliance

Systems or people overly depending on LLMs without oversight may face misinformation, miscommunication, legal issues, and security vulnerabilities due to incorrect or inappropriate content generated by LLMs.

LLM10

### Model Theft

This involves unauthorized access, copying, or exfiltration of proprietary LLM models. The impact includes economic losses, compromised competitive advantage, and potential access to sensitive information.

https://llmtop10.com/

# Q/A