STJ ACM
Student Chapter

AI, and YOU!

How to Effectively Use AI Tools as a Tech Major

Sign In Here!

EFFECTIVE AI STRATEGY

- Tool Overview
- Cursor Overview
- Basic Prompt Engineering
- Cursor Interactive Demo + Strategy
- Other Strategies

WHY AI?

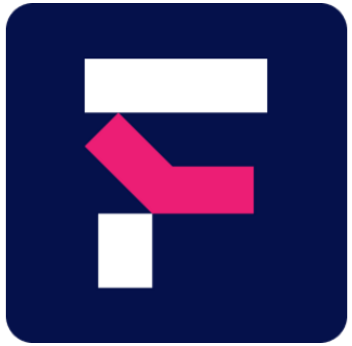
GitHub Copilot

Functionality:

- AI-powered code completion tool integrated into IDEs.
- Provides context-aware code suggestions and auto-completions.

Benefits:

- Increases coding efficiency and productivity.
- Supports multiple programming languages and IDEs like VS Code and JetBrains.



Figstack

Functionality:

- AI tool for code explanation and translation between languages.
- Optimizes code efficiency using Big O notation.

Benefits:

- Helps developers understand and improve code across multiple languages.
- Provides detailed docstring generation for better code documentation.

Adobe Sensei

Functionality:

- AI and machine learning framework integrated into Adobe Creative Cloud.
- Automates repetitive design tasks and enhances creative workflows.

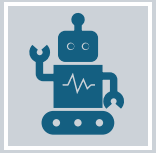
Benefits:

- Speeds up design processes with features like auto-tagging and smart cropping.
- Improves user experience by providing data-driven insights and design suggestions

But how can we use AI to...

Write code?

Find flags?



Automate my job?

What is Cursor?

- A fork of Visual Studio Code with integrated AI capabilities.
- Designed to enhance coding efficiency and productivity.

Key Features:

- **AI Chat with Codebase:**
  - Allows interaction with your codebase for questions and insights.
  - Provides context-aware responses by referencing files and documentation.
- **Code Generation and Refactoring:**
  - Generate new code or refactor existing code with simple commands.
  - Supports multi-file code generation.
- **Debugging and Lint Fixing:**
  - Automated debugging feature to identify and fix issues.
  - Simplifies fixing lint errors with AI assistance.

Demo Time!

Basic Prompt Engineering

- When talking to an AI language model, we need to be *deliberate* with the instructions we give.
- In other words, we need to carefully **prompt** our model to do a certain thing!
- We can break down a basic prompt into three separate parts - or roles – that gives the language model the necessary information to carry out your request.
  - System Role
  - User Role
  - Task Role

System Role

- **Definition:** This defines the role or capabilities of the language model (LLM). It sets the context for how the model should respond.
- **Importance:** By specifying the system, you guide the model's tone, focus, and depth of knowledge. This helps tailor responses to your specific needs.
- **Utilization:** Clearly state what kind of assistant the LLM should be. For example, "You are a language model specialized in educational content" informs the model to prioritize educational outputs.

User Role

- **Definition:** This describes who is asking the question or making the request. It helps the model understand the perspective and needs of the requester.
- **Importance:** Knowing the user's background and intentions helps the model provide more relevant and contextually appropriate responses.
- **Utilization:** Include details about yourself or the audience, such as, "I am a college instructor looking to engage tech majors." This helps the LLM customize its advice.

Task Role

- **Definition:** This specifies the actual request or action you want the model to perform. It clarifies what you need from the model.
- **Importance:** A well-defined task ensures the model delivers focused and useful outputs, minimizing ambiguity.
- **Utilization:** Clearly articulate what you want, such as “Help me create a lesson plan on AI tools.” This sets a clear objective for the model’s response.

Do I HAVE to use Cursor?

Questions?

Join our Discord! Let’s talk more about AI!