AI-powered software engineer to AI Coding Agents team

Role description

You will work closely with coding agents and use them to develop tools for coding agents. We are looking for a developer who is passionate about AI coding and who is willing to adopt new technologies quickly.

Requirements:

- Python and/or Javascript experience
- Experience with chat-bots (ChatGPT, Claude, Gemini)
- Experience with one or more of these AI coding agents: Cursor, Claude Code, Codex, Gemini
- Basic knowledge of how coding agents and MCPs work
- Basic experience with Git/GitHub
- Strong English proficiency

What to do:

- Use coding agents a lot
- Develop tools for AI coding

Test task — implement your own local MCP server

MCPs are a great way to enrich a coding agents with additional tools. You can learn more about MCPs here. You can find a lot of existing MCP servers here.

In this task, your goal is to develop a **local** MCP server using Javascript or Python. So it should work as a process one can run on their laptop.

You can implement any idea you can think of. Examples you might implement:

- MCP that has a tool to search documentation for predefined library
- MCP to return list of available components in Javascript components library
- MCP to generate images using Nebius Token Factory or any other API
- MCP to return youtube video transcription by video URL
- MCP that is connected to any third-party API or service of your choice

Implementing MCP

Good starting point for both Python and JS is https://modelcontextprotocol.io/docs/develop/build-server. You can follow the tutorial to implement weather MCP and then change it to something else to complete the task.

You can use AI to implement it, but you need to understand how it works and why is it implemented in that way.

Testing your MCP

Feel free to use this free tool to test your MCP: https://github.com/modelcontextprotocol/inspector

This MCP should work when installed for coding agents (at least one: Cursor/Claude Code/Codex/Gemini etc.).