# Post

**AI Chat Choices: One or Many?** 🤔

With ChatGPT revolutionizing our tasks, a slew of specialized alternatives have emerged. This poses a dilemma: Stick to one AI chat or embrace multiple for diverse insights?

In my daily grind, I leverage ChatGPT, Phind, Bard, and Perplexity, seeking varied perspectives. While this approach means extra effort in posing the same query to different AIs, I've found a workaround!

I've crafted a Browser Web Extension that lets you prompt multiple AI chats simultaneously. Curious about how to streamline your AI chat experience? Stay tuned for details on this game-changer!

# AI Multi Prompt extension

## Use multiple AI Chat at same time.

After the launch of ChatGPT, which is quickly changing the way we handle tasks we need, we are witnessing the birth of numerous alternatives. Each of these specializes in a particular task and tries to stand out from the others.

This presents us with the following dilemma: Do we choose one AI chat and always use that, or do we use more than one, selecting the best response each time?

My choice has been to use more than one. Specifically, in my day-to-day work, I use: ChatGPT, Phind, Bard, and Perplexity. These guarantee a wide range of responses that I can assess each time.

Of course, my choice means that I must submit the same prompt to multiple AI chats, which can sometimes be tedious task.

For this reason, I decided to develop a Browser Web Extension that could help me submit the same prompt to multiple AI chats simultaneously.

In this article, I will briefly explain how it works and how to get and use it if, like me, you use more than one.

## Develop a Safari Web Extension

Since I use mostly Safari Browser my first implementation has been focused on Safari Web Extension also if I'd take in account to port such implementation on Chrome and Edge if requested.

### The requirements

Idea was to create an extension that detect if there are (supported) AI Chat opened in browser tabs gather them and open a popup that allow to input a prompt that will be submitted simultaneously to each AI Chat identified.

### The challenges

The only development’s challenge here has been **to simulate of what a user does to submit a prompt**, to achieve this has been needed to analyze each AI Chat web page to identify the HTML component that play the role of prompt (ie. textarea), the submit button (ie. button) and write a tailored javascript code for each of them. Below a code snippet extracted by implementation provided for ChatGPT

const chatgptSubmit = ( prompt ) => {

const promptElem = document.querySelector("form #prompt-textarea");

promptElem.value = prompt;

promptElem.dispatchEvent(new Event('input', { 'bubbles': true }));

const parentForm = promptElem.closest("form")

const buttons = parentForm.querySelectorAll("button");

if( buttons && buttons.length > 0 ) {

// get last button in the form

const submitButton = buttons[ buttons.length - 1 ];

submitButton.click()

}

}

### The Result

After a couple of hours of development and bit more ones of tests finally I’ve released the first working release and the result is shown below

A screenshot of a computer

Description automatically generated

It seems work as expected and I’m using it intensively using the approach “Eat your own dog food” (AKA dogfooding) to assure quality and improvements as much as I can**.**

## Conclusion

I published the project on github,

### V. Upcoming Features and Developments

* **Cross-browser Compatibility**: Plans to adapt the extension for Chrome and Edge.
  + Dive into the importance of expanding the extension’s compatibility to cater to a wider audience.
* **Ability to Save Prompts Locally**:
  + Elaborate on how this would benefit users in terms of revisiting previous prompts and refining them for better results.

### VI. How to Use the Extension

* **Step-by-step guide**: Visual aids (like screenshots) would be beneficial here.
  1. Installation process on Safari.
  2. Setting up for the first use.
  3. Submitting a prompt using the extension.
  4. Viewing and comparing results from each platform.

### VII. Conclusion

* Reiterate the value proposition of the ‘AI Multi Prompt extension’ for users.
* An open invitation for community feedback, as it helps in refining and improving the extension.
* Encourage readers to share their comparative findings and experiences using the extension on various AI platforms.

### VIII. Call to Action (CTA)

* Direct users to where they can download and install the extension.
* Encourage sharing the blog post to spread the word and increase its adoption among the AI chat platform user community.