Graphical user interface, text, application

AI-generated content may be incorrect.

# Why use AI

* Prototyping from text prompting to text prompting from an image or video
* To write complex code that takes a long time when a developer can write it themselves
* To write code, a developer does not have expertise with
* Creation of documentation
* Seamlessly integrate into all popular programming languages
* To learn what a segment of code does

## Tips

* Think of AI tools as paired programming, which can be enabled when needed and disabled when not needed.
* Visual Studio now supports prompt files in your repository to run or share your frequently used prompts.
  + The custom instructions feature enables you to automatically add pre-specified contextual details to your chat questions.
* GitHub Copilot and ChatGPT offer all of the popular models
* ChatGPT:
  + Projects provide a means to have a collective of similar chats/responses
  + Custom GPTs
* GitHub Copilot works in both Microsoft VS Code and Microsoft Visual Studio
  + Both support Agents and Model Context Protocol (MCP) - MCP is out of scope for this meeting

The plan is to use GitHub Copilot Enterprise edition, which safeguards all data and allows local LLM.

Currently, there has been approval for the ChatGPT API, but no purchase approval.

# Benefits

* Saves time on completing difficult tasks
* Optimize code
* Document code
* Debugger assistant: If something throws an exception in VS2022, a developer can ask Copilot for assistance.
* Ask questions about existing code
  + Using Copilot: inline with code and/or chat window
* Create new code
* Fix accessibility issues such as orphaned label/input
* Copilot Agent Mode: Use one prompt, have multiple file creations and/or edits
* Next Edit Suggestions (NES) which leverages existing code to anticipate what comes next.
* Implement with Copilot: Create a method with a meaningful name and get recommendations on coding it.
* Anticipate what comes next
  + Properties
  + Conditional statements