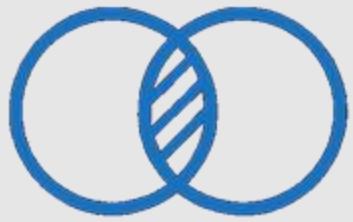
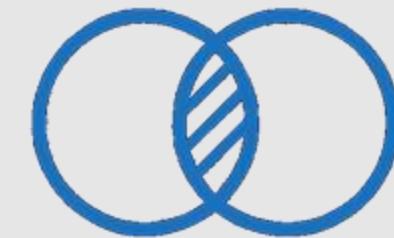




# Advanced GitHub Copilot



**AI in Production**



**AI in Production**

Presented by Brent Laster &  
Tech Skills Transformations LLC

© 2025 Brent C. Laster & Tech Skills Transformations LLC



All rights reserved



# Agenda

- Using Copilot for jumping into a project
- Copilot modes – Ask, Edit, Agent
- Using Agent mode to do changes autonomously
- Using Edit mode to do targeted refactoring and improvements
- Extending Copilot functionality with MCP servers
- Adding functionality to Copilot with extensions
- Bonus Tips



# Lab prep - repo is [github.com/skillrepos/copilot-adv](https://github.com/skillrepos/copilot-adv)

1. Go to <https://github.com/skillrepos/copilot-adv> (Chrome may work best for copy and paste actions.)
2. Follow instructions in **README.md**
3. Startup codespace with quickstart button in README
4. Open labs.md in browser or in preview mode in codespace

The screenshot shows the GitHub README.md page for the 'copilot-adv' repository. At the top, there's a commit history entry from 'techupskills' updating the README.md file 20 hours ago. Below the commit, there are tabs for 'Preview', 'Code', and 'Blame'. The content of the README is displayed, starting with a section titled 'Advanced Copilot'.

**Advanced Copilot**

Repository for Advanced Copilot Workshop

These instructions will guide you through configuring a GitHub Codespaces environment that you can use to run the code.

1. Click on the button below to start a new codespace from this repository.

Click here [Open in GitHub Codespaces](#)

The screenshot shows the GitHub labs.md page in preview mode for the 'copilot-adv' repository. At the top, there's a commit history entry from 'techupskills' updating the labs.md file 8 hours ago. Below the commit, there are tabs for 'Preview', 'Code', and 'Blame'. The content of the labs.md page is displayed, featuring sections like 'Advanced features in GitHub Copilot', 'Hands-on Workshop for AI in Production', 'Session labs', and 'Revision 1.4 - 07/01/25'. A note at the bottom states: 'Versions of dialogs, buttons, etc. shown in screenshots may differ from current version of Copilot'.

**Advanced features in GitHub Copilot**

**Hands-on Workshop for AI in Production**

**Session labs**

**Revision 1.4 - 07/01/25**

Versions of dialogs, buttons, etc. shown in screenshots may differ from current version of Copilot

Follow the startup instructions in the README.md file IF NOT ALREADY DONE!

**IMPORTANT NOTES:**

# About me

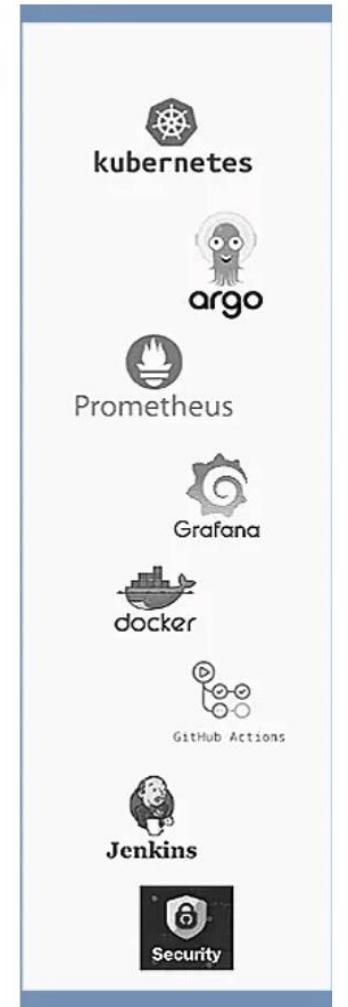


- Founder, Tech Skills Transformations LLC
- <https://getskillsnow.com>
- [info@getskillsnow.com](mailto:info@getskillsnow.com)
- Long career in corporate as dev, manager, and director in DevOps and other areas
- Author
  - O'Reilly "reports"
  - Books
    - Professional Git
    - Jenkins 2 – Up and Running
    - Learning GitHub Actions
    - AI-Enabled SDLC
    - Learning GitHub Copilot
- Speaker
- Social media
  - [LinkedIn: brentlaster](#)
  - [X: @BrentCLaster](#)
  - [Bluesky: brentclaster.bsky.social](#)
  - [GitHub: brentlaster](#)



LISTED BELOW ARE A FEW OF THE TECHNOLOGIES FOR WHICH WE OFFER HANDS-ON TRAINING. THESE CAN BE CUSTOMIZED FOR ANY SIZE TEAM FROM 1-100 AND FOR ANY LEVEL FROM BEGINNER TO ADVANCED.

- ArgoCD
- Containers
- Docker
- Gerrit
- Git
- GitHub Actions
- GitHub Codespaces
- GitHub Copilot
- GitHub Foundations
- GitHub Security
- GitLab
- GitOps
- Gradle
- Grafana
- Helm
- Jenkins
- Kubernetes
- Kustomize
- LLMs
- Prometheus
- Tekton
- VS Code





# How do we normally use AI assistants like Copilot?

- **Accelerators/Acceleration** - code completion - working in the flow and AI provides completion suggestions (usually in the editor)
- **Explorers/Exploration** - using the chat interface to ask how to code something? What is the API?
- **Directors/Direction** - giving the AI direct and specific instructions to create something via the chat interface or in the editor (via comments)
- All of these *can* lead to increased developer productivity – when working on known code

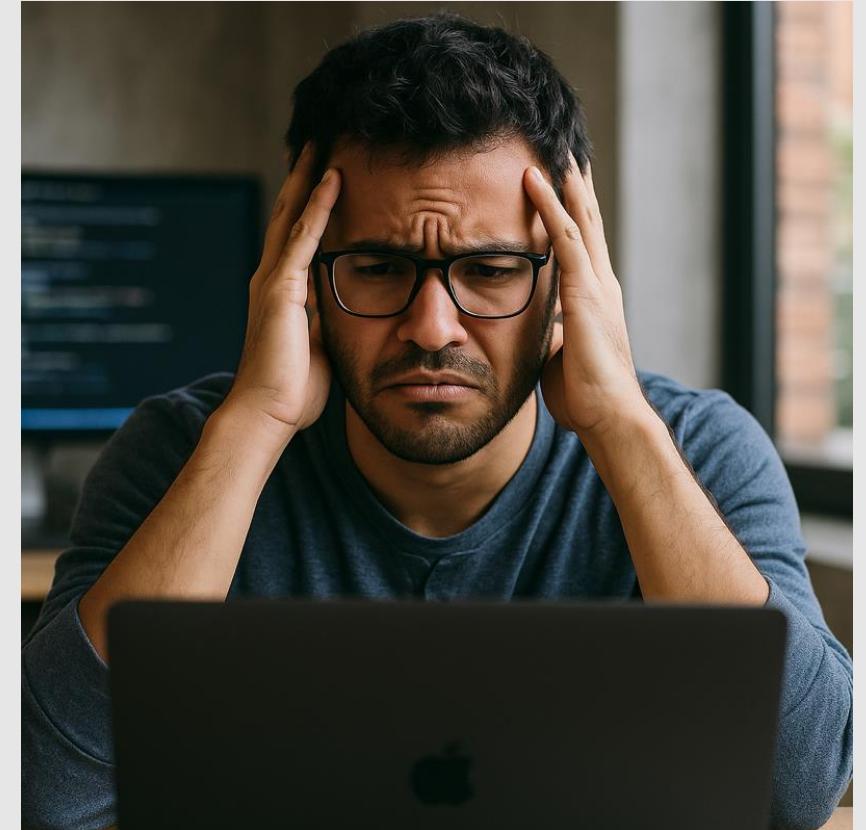


source: "<https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/unleashing-developer-productivity-with-generative-ai>



## But what about *those* cases when you have to deal with...

- a team member leaving or changing roles
- inheriting a legacy system or old project
- onboarding to/for a new team or organization
- supporting or maintaining open-source contributions
- acquiring a third-party or partner codebase
- participating in a code rotation
- working on cross-functional or cross-team initiatives
- taking over stalled or unfinished projects
- integrating multiple systems or merging repositories



**In short, when you have to deal with code you're not familiar with**



# Ways Copilot Can Help

- Understanding a New Codebase
  - Explain code
  - Identify key files, entry points, main modules
  - Summarize functions and classes
- Generating Onboarding Documentation
  - Draft onboarding guide to get started quickly
  - Write quick-start examples and usage instructions
  - Creating documentation for APIs, functional docs
- Getting Productive Quick
  - Explain how to run and test code
  - Suggest test cases and fixes
  - Reviewing code
- Filling in Gaps
  - Adding docstrings and inline comments
  - Generating tasks lists
  - Documenting architecture
  - Asking for edge cases for testing
- Managing repo support pieces
  - Summarizing issues, pull requests
  - Suggesting next steps on issues and pull requests

**Copilot includes multiple "modes" we can use for interacting with it to do these types of tasks...**

The screenshot shows a GitHub pull request titled "Fix code scanning alert - Database query built from user-controlled sources #3". The pull request has a green "Open" button and a "New issue" button. A tooltip on the right says "How do I add tests to my project?". Below it, a "prime.py:1-8" file is shown with code completion suggestions. A "Workspace" section indicates "Used /tests (rerun without)" and "Used 1 reference". A detailed explanation follows:

To add tests to your project, you can create a new file named `test_prime.py` in the same directory as `prime.py`. In this file, you can import the `is_prime` function and create a test class that inherits from `unittest.TestCase`. Write test methods to test various cases for the `is_prime` function. Use assertions to verify the expected outcomes.

Here is the code for `test_prime.py`:

```
test_prime.py
1+ import unittest
```



# GitHub Copilot Modes

## Ask Mode

- Quick, context-aware Q&A without touching code
- Highlight code or ask general programming questions
- Returns explanations or snippet suggestions instantly

## Edit Mode

- Describe your desired change in plain language
- Applies inline diffs across one or more files
- Shows you a review-ready diff before you commit
- Ideal for surgical refactors or consistent batch updates

## Agent Mode

- Hand over a high-level task and watch Copilot plan & execute
- Autonomously selects files, applies edits, and runs commands
- Streams changes continuously, surfacing only risky steps for review
- Best for multi-step workflows: scaffolding features, fixing bugs, end-to-end tasks
- Iterates until all good (validated) or it can go no further

CHAT

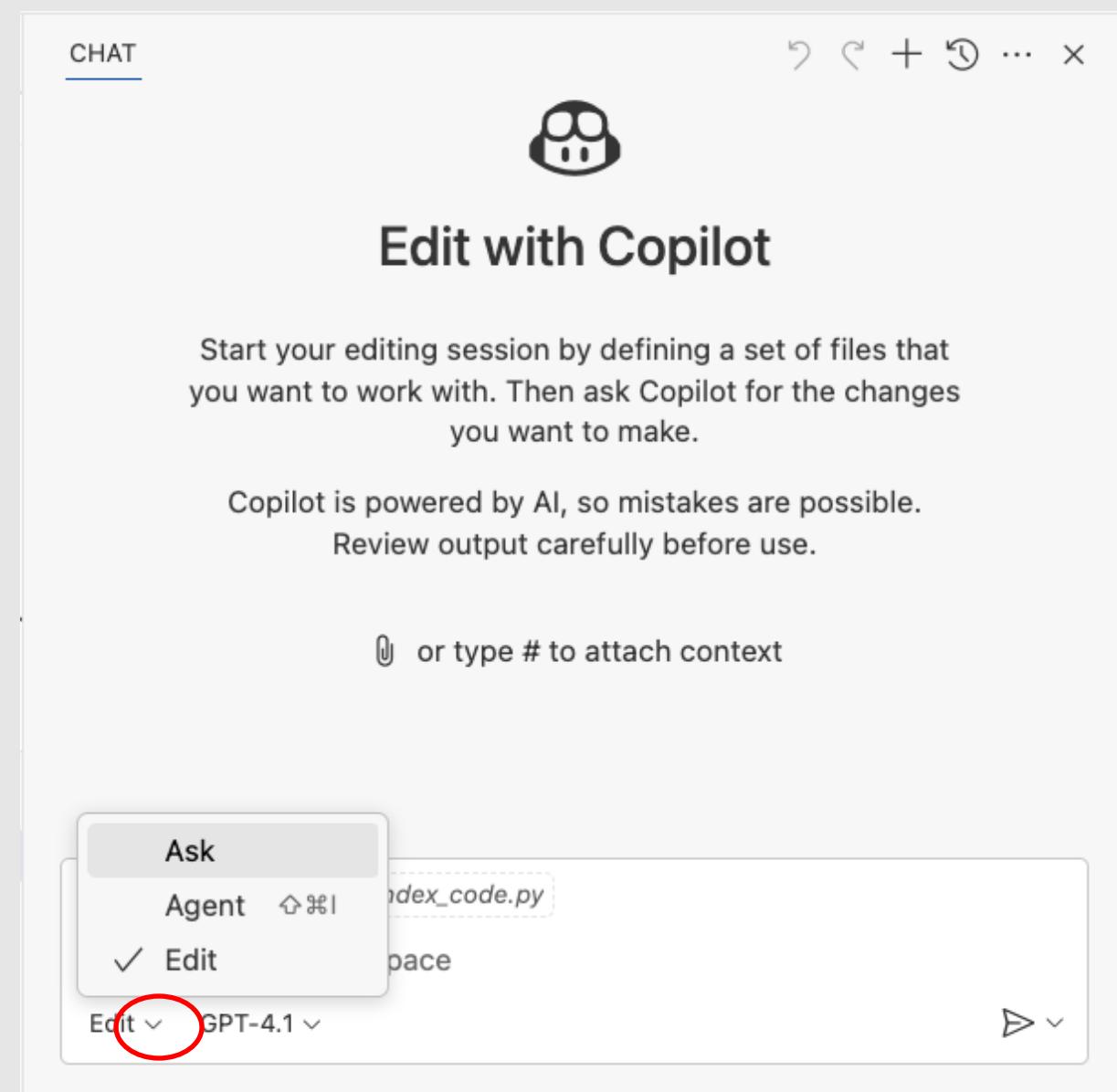
 Edit with Copilot

Start your editing session by defining a set of files that you want to work with. Then ask Copilot for the changes you want to make.

Copilot is powered by AI, so mistakes are possible.  
Review output carefully before use.

⌚ or type # to attach context

Ask  
Agent ⌂⌘I  
✓ Edit  
Edit ▾ GPT-4.1 ▾





# Use Cases for Ask Mode

- Getting quick answers about a specific code snippet
- Asking for a short explanation or usage example
- Clarifying syntax, library functions, or framework behavior
- Verifying a pattern or best practice
- Getting documentation-style responses without changing code

# Lab 1 – Using Copilot for Onboarding, Explaining, and Running Projects

**Purpose:** In this lab, we'll look at how to use Copilot to help quickly get up to speed on any project.



# Tip: Customizing Copilot Responses in VS Code

- You can create custom instructions to guide how Copilot writes code
- Customize how Copilot behaves for code, comments, reviews, and commits
- Use prompt files to create reusable tasks (like “Write unit tests”)
- Set rules globally, per workspace, or even per file type
- Tailor Copilot’s responses to match your project’s style and goals
- Use variables in prompts to adapt to selected code or file context
- Sync your settings across devices for consistent behavior

See:

<https://code.visualstudio.com/docs/copilot/copilot-customization>

```

--- general.instructions.md ---
applyTo: "**"

# Project general coding standards

## Naming Conventions
- Use PascalCase for component names, interfaces, and type aliases
- Use camelCase for variables, functions, and methods
- Prefix private class members with underscore (_)
- Use ALL_CAPS for constants

## Error Handling
- Use try/catch blocks for async operations
- Implement proper error boundaries in React components
- Always log errors with contextual information

--- settings.json ---
"github.copilot.chat.codeGeneration.instructions": [
  {
    "text": "Always add a comment: 'Generated by Copilot.'"
  },
  {
    "text": "In TypeScript always use underscore for private field names."
  },
  {
    "file": "general.instructions.md" // import instructions from file `general.instructions.md`
  },
  {
    "file": "db.instructions.md" // import instructions from file `db.instructions.md`
  }
],

```



# Agent Mode is Powerful...

- In Agent Mode, can provide prompt for what kind of changes you want Copilot to make
- Example: “Convert all Python files in project to Go”
  - Copilot makes changes to *Keep* or *Undo*
  - Goes further to look for any issues that might come up as a result of the changes
  - If any issues found, Copilot will iterate to try and resolve them
    - » Goes until it gets to a point where everything looks good or it can't proceed on its own

The Go implementation of the `prime.py` logic has been completed in `prime.go`, and the corresponding tests have been written in `prime_test.go`. Next, I'll run the Go tests to ensure the functionality matches the original Python implementation.

2 files changed

`prime_test.go`  
`prime.go`

Add Context... `prime.go` Current file

Edit files in your workspace in agent mode

Agent GPT-4o

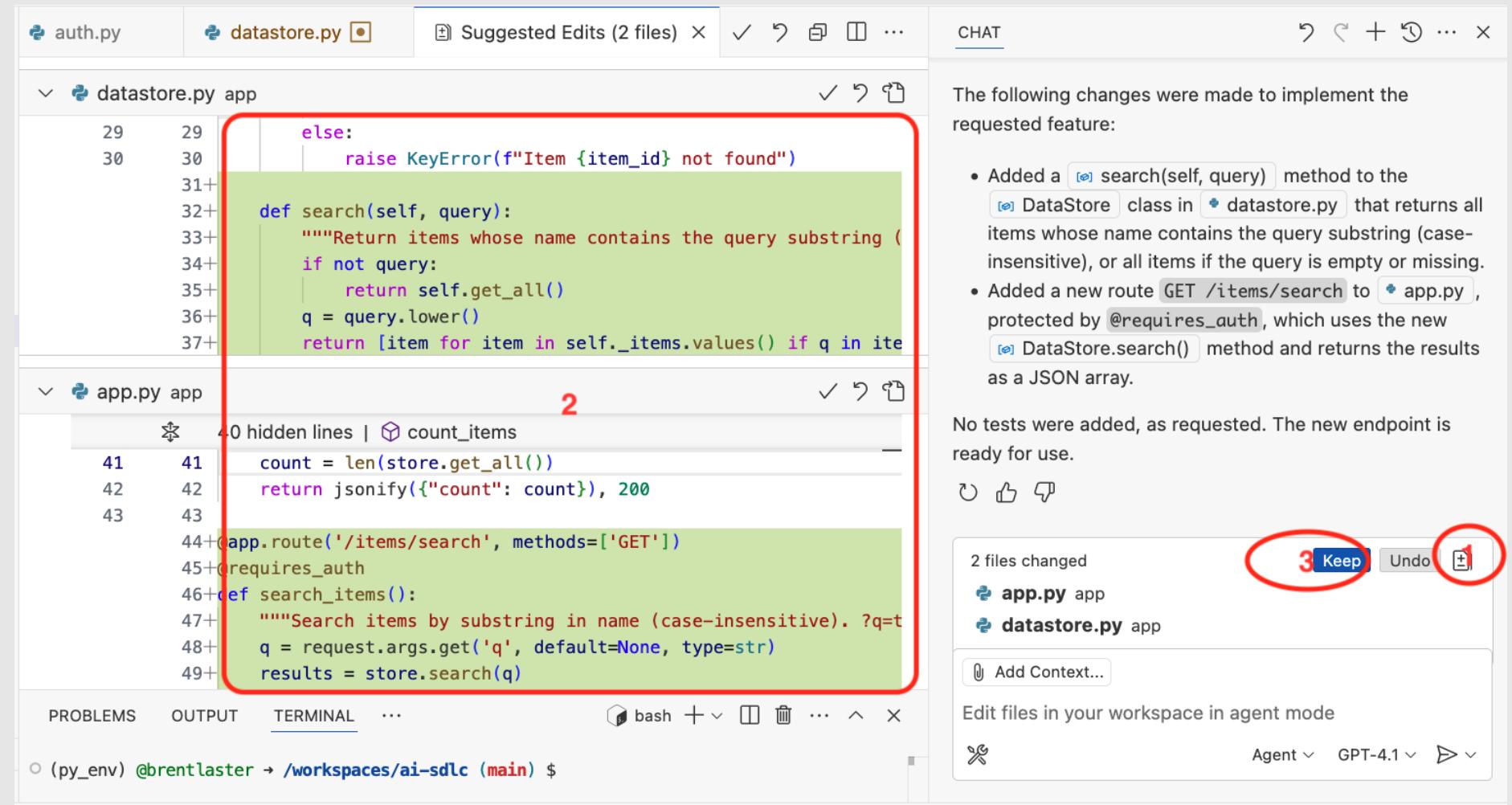


# Use Cases for Agent Mode

- Asking multi-step or open-ended questions
- Getting help reasoning through complex logic
- Creating or reviewing onboarding documentation
- Debugging issues with multiple contributing factors
- Generating code with toolchain or config dependencies
- Performing structured tasks like refactoring with explanations

# Reviewing Suggested Edits

- Comes into play when Copilot does an operation that results in suggested edits
- +/- doc icon applies suggested edits for review
- Suggested edits are shown as green/red blocks inline
- Individual Keep/Undo or Accept/Discard options per edit and "for all edits" in "files changed" area above chat text area



The screenshot shows a code editor with two tabs: `auth.py` and `datastore.py`. A third tab, `Suggested Edits (2 files) X`, is active. The `datastore.py` tab shows a green-highlighted code block containing new logic for a `search` method. The `app.py` tab shows another green-highlighted block containing a new `search_items` route. Red boxes highlight these changes in both files. In the bottom right corner of the editor, there's a panel titled "2 files changed" listing `app.py` and `datastore.py`. To the right of this panel is a "Keep" button circled in red, along with "Undo" and "Discard" buttons. The number "1" is shown near the "Keep" button.

```

datastore.py
else:
    raise KeyError(f"Item {item_id} not found")

def search(self, query):
    """Return items whose name contains the query substring (case-insensitive), or all items if the query is empty or missing.
    if not query:
        return self.get_all()
    q = query.lower()
    return [item for item in self._items.values() if q in item.name]

```

```

app.py
count = len(store.get_all())
return jsonify({"count": count}), 200

@app.route('/items/search', methods=['GET'])
@requires_auth
def search_items():
    """Search items by substring in name (case-insensitive). ?q=t
    q = request.args.get('q', default=None, type=str)
    results = store.search(q)

```

## Lab 2 – Using Copilot's Agent Functionality to Implement a New Feature

**Purpose:** In this lab, we'll see how to use Copilot to automatically implement a feature request to our codebase.



# Tip: Copilot Vision to analyze/debug code

Suppose we use this image...

- Attach screenshot and have Copilot analyze/debug

C# divide.cs

```
1  public int Divide(int a, int b)
2  {
3      return a / b;
4  }
5
```



# Tip: Copilot Vision to analyze/debug code

- Attach screenshot and have Copilot analyze/debug

Attach screenshot to chat prompt area and tell it to debug

The image shows a screenshot of a Copilot Vision interface. At the top, there is a yellow bar with the text "Attach screenshot to chat prompt area and tell it to debug". Below this, the main interface has a blue border. Inside, there is a "Add Context..." button with a paperclip icon, a thumbnail for a screenshot named "Screenshot 2025-04-28 at 12.37.05 PM.png", and a close button. Below the thumbnail, the text "Referencing the attached screenshot, debug any problems shown" is displayed. At the bottom left is an "@" icon, and at the bottom right are dropdown menus for "Ask", "GPT-4o", and a large arrow icon.



# Tip: Copilot Vision to analyze/debug code

Copilot analyzes code, finds potential problems, and suggests updated code

- Attach screenshot and have Copilot analyze/debug

The screenshot shows the GitHub Copilot interface. At the top, there's a GitHub Copilot icon and the text "GitHub Copilot" followed by "Used 1 reference". Below this, a message states: "The code in the screenshot has a potential issue: **division by zero**. If the parameter `b` is passed as `0`, the program will throw a `DivideByZeroException` at runtime." A note below says, "To fix this, you should add a check to ensure that `b` is not zero before performing the division. Here's the updated code:" followed by a code block. At the bottom, a note says, "This ensures that the method handles invalid input gracefully by throwing an appropriate exception."

```
public int Divide(int a, int b)
{
    if (b == 0)
    {
        throw new ArgumentException("Division by zero is not allowed.", nameof(b));
    }
    return a / b;
}
```



# Copilot Edits

- Feature in GitHub Copilot chat for code editing assistance
- Allows quick, AI-powered edits across one or multiple files
- Enhances productivity by automating repetitive coding tasks
- Works with natural language instructions for code modifications
- Available via drop-down in chat area

CHAT

The screenshot shows the GitHub Copilot Chat interface. At the top, there's a header with icons for search, refresh, and exit, followed by the number '20'. Below the header is a user profile icon. The main area has a title 'Edit with Copilot' with a sub-instruction: 'Start your editing session by defining a set of files that you want to work with. Then ask Copilot for the changes you want to make.' It also notes that 'Copilot is powered by AI, so mistakes are possible. Review output carefully before use.' Below this are three tips: '@ or type # to attach context', '@ to chat with extensions', and 'Type / to use commands'.

The screenshot shows the GitHub Copilot Chat interface with a dropdown menu open over the '@ to chat with extensions' tip. The menu items are 'Ask', 'Edit' (which is highlighted with a blue background and a checkmark), and 'Agent'. At the bottom of the menu are buttons for 'Edit ▾', 'GPT-4o ▾', and a right-pointing arrow. The main interface below the menu shows a text input field with 'Edit files in your workspace' placeholder text, and a status bar at the bottom with '0 0' and icons for a keyboard, a bell, 'Layout: U.S.', and a notification bell.



# Use Cases for Edit Mode

- Suggest edits based on written instructions or comments
- Add new functionality, like logging, to existing codebases
- Refactor code to improve readability or performance automatically
- Update multiple files in a project with consistent changes
- Create new files when required, like helper scripts or configs
- Enable iterative edits by refining suggestions interactively
- Save time by automating complex, multi-step coding processes
- Offers a preview of changes before applying them

CHAT



Start your editing session by defining a set of files that you want to work with. Then ask Copilot for the changes you want to make.

Copilot is powered by AI, so mistakes are possible.  
Review output carefully before use.

📎 or type # to attach context

@ to chat with extensions

Type / to use commands

📎 Add Context... ⊕ calculator.py X ⊕ test\_calculator.py X

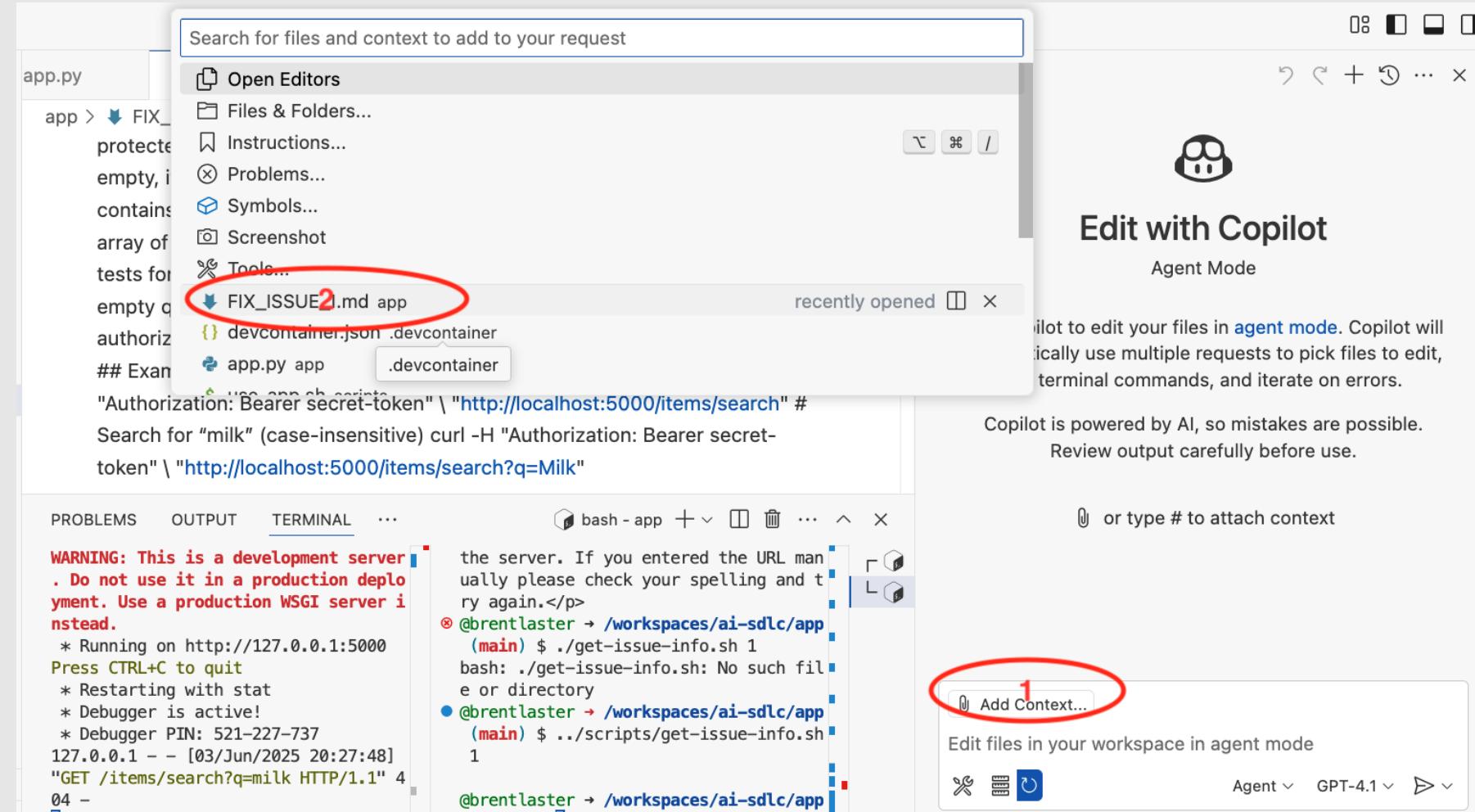
- Refactor the divide function to handle division by zero more gracefully.
- Add type hints to all functions in calculator.py.
- Write additional unit tests for edge cases in test\_calculator.py.

Edit ▾ GPT-4o ▾ ➔ ▾



# Setting Context

- In Chat, "context" more directly tells Copilot what to focus on when responding to a non-specific prompt
- Active file is default context
- Can drag and drop files into Chat dialog area
- Can use "Add Context" control to select files or other items as context
- Can have multiple items





# Supplying instructions for Copilot Edits

- Select files to edit
- Add prompts/directives in chat area

The screenshot shows the Microsoft Copilot interface. At the top, there's a toolbar with icons for file operations like New, Open, Save, and Close. Below that is a section titled "Edit with Copilot" with a "PREVIEW" button. A descriptive text box says: "Start your editing session by defining a set of files that you want to work with. Then ask Copilot for the changes you want to make." Another text box below it says: "Copilot is powered by AI, so mistakes are possible. Review output carefully before use." At the bottom, there's a "or type #" placeholder followed by a context attachment icon. The main workspace has a "Add Files..." button and two AI-generated prompts: "Update Kubernetes APIs to the latest version" and "Add logging functionality". At the bottom right, there are icons for microphone and speaker, and a dropdown menu labeled "GPT 4o" with a play button icon.



# Processing prompts

- Changes are processed across Working Set

The screenshot shows a GitHub Copilot interface with a user profile for 'brentlaster'. It displays three processing prompts:

- GitHub Copilot** (main.go)  
Update Kubernetes APIs to the latest version and add logging functionality.  
[main.go](#)
- suite\_test.go**  
Update Kubernetes APIs to the latest version and add logging functionality.  
[suite\\_test.go](#)
- roarapp\_controller.go**  
Update Kubernetes APIs to the latest version and add logging functionality.



# AI review - Having Copilot review code in VS Code

- AI-powered feedback on code as you write it
  - Review selection:
    - » select code in VS Code and ask for initial review
    - Review changes: ask Copilot for deeper review of all changes before pushing from “Source Control” tab
      - » Can also be done in pull requests on GitHub.com

117 if numAvailable > instance.Spec.Replicas {

Code Review Comment (1 of 5)

GitHub Copilot

The condition numAvailable > instance.Spec.Replicas could be simplified for better readability. Consider using a variable to store the difference.

Suggested change:

```
-     if numAvailable > instance.Spec.Replicas {  
-         log.Info("Scaling down pods", "Currently available", numAvailable, "Required replicas", instance.Spec.  
-             diff := numAvailable - instance.Spec.Replicas  
+     diff := numAvailable - instance.Spec.Replicas  
+     if diff > 0 {  
+         log.Info("Scaling down pods", "Currently available", numAvailable, "Required replicas", instance.Spec.
```

Apply and Go to Next ▾ Discard and Go to Next ▾

118 log.Info("Scaling down pods", "Currently available", numAvailable, "Required replicas", instance.Spec.Replicas)  
119 diff := numAvailable - instance.Spec.Replicas

## Lab 3 – Refactoring and Updating Code via Copilot Edits

**Purpose:** In this lab, we'll see how to use Copilot Edits functionality to refactor targeted sets of code, both for efficiency and improvements.



# Tip: Next Edit Suggestions (NES)

27

- Much coding activity is editing existing code
- NES allows coding suggestions to work on existing code
- Based on change in one area, suggests changes in other areas of code
- To enable, set  
github.copilot.nextEditSuggestions.enabled=true
- When you accept a change, if there's a related change that should be made, Copilot will display an arrow in the gutter
- Streamlines refactoring

Users > developer > point-class.py > ...

```
1 # Define a 3D Point class
2 class Point:
3     def __init__(self, x, y, z):
4         self.x = x
5         self.y = y
6         self.z = z
7
8     def __init__(self, x, y):
9         self.x = x
10        self.y = y
11        self.z = 0
12
13     def distance(self, other):
14         return ((self.x - other.x)**2 +
15                 (self.y - other.y)**2 +
16                 (self.z - other.z)**2)**0.5
17
18 # Create two points and calculate distance
19 p1 = Point(1, 2, 0)
20 p2 = Point(4, 6, 0)
21 print(p1.distance(p2))
22
```

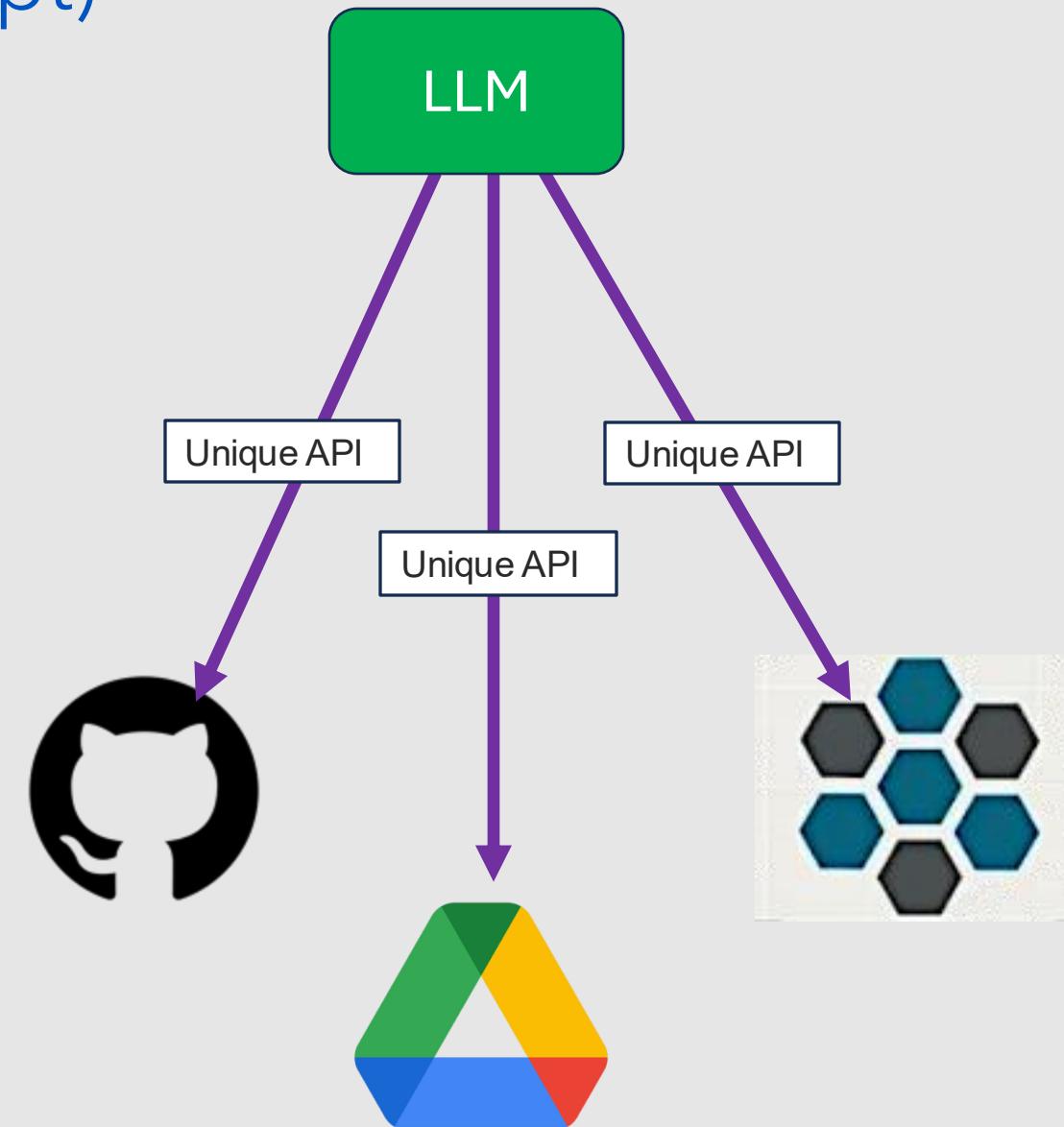
→ 19 p1 = Point(1, 2, 0)



# Model Context Protocol (Concept)

- MCP is an open protocol for standardizing how applications provide tools and resources to AI applications
- Like a universal connector for AI (sometimes called "usb-c" of AI)
- Manages
  - Tool discovery: Identifies right tool for request
  - Invocation: Executes the function call
  - Response handling : Returning results in a structured format

Without MCP

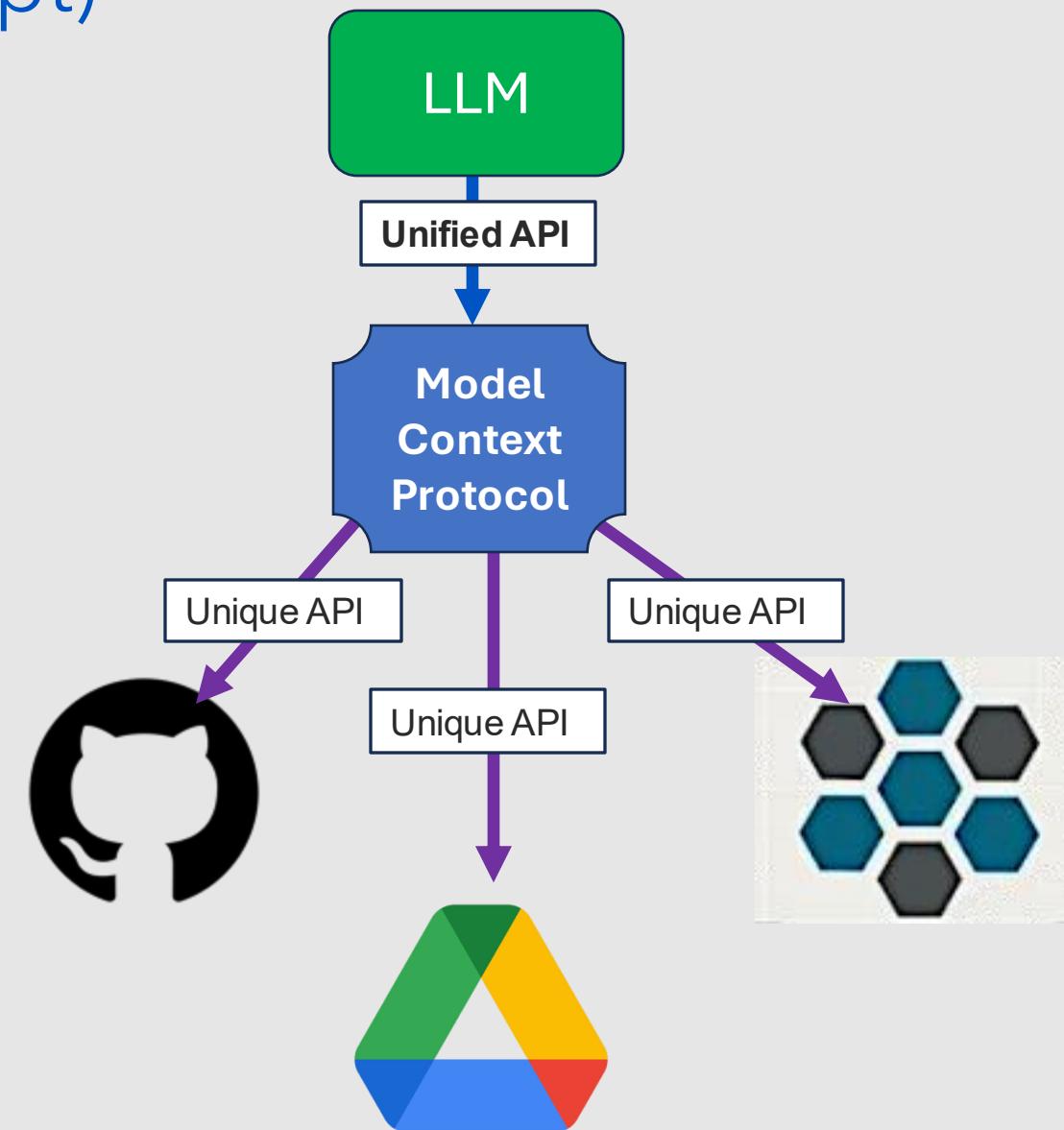




# Model Context Protocol (Concept)

- MCP is an open protocol for standardizing how applications provide tools and resources to AI applications
- Like a universal connector for AI (sometimes called "usb-c" of AI)
- Manages
  - Tool discovery: Identifies right tool for request
  - Invocation: Executes the function call
  - Response handling : Returning results in a structured format

With MCP

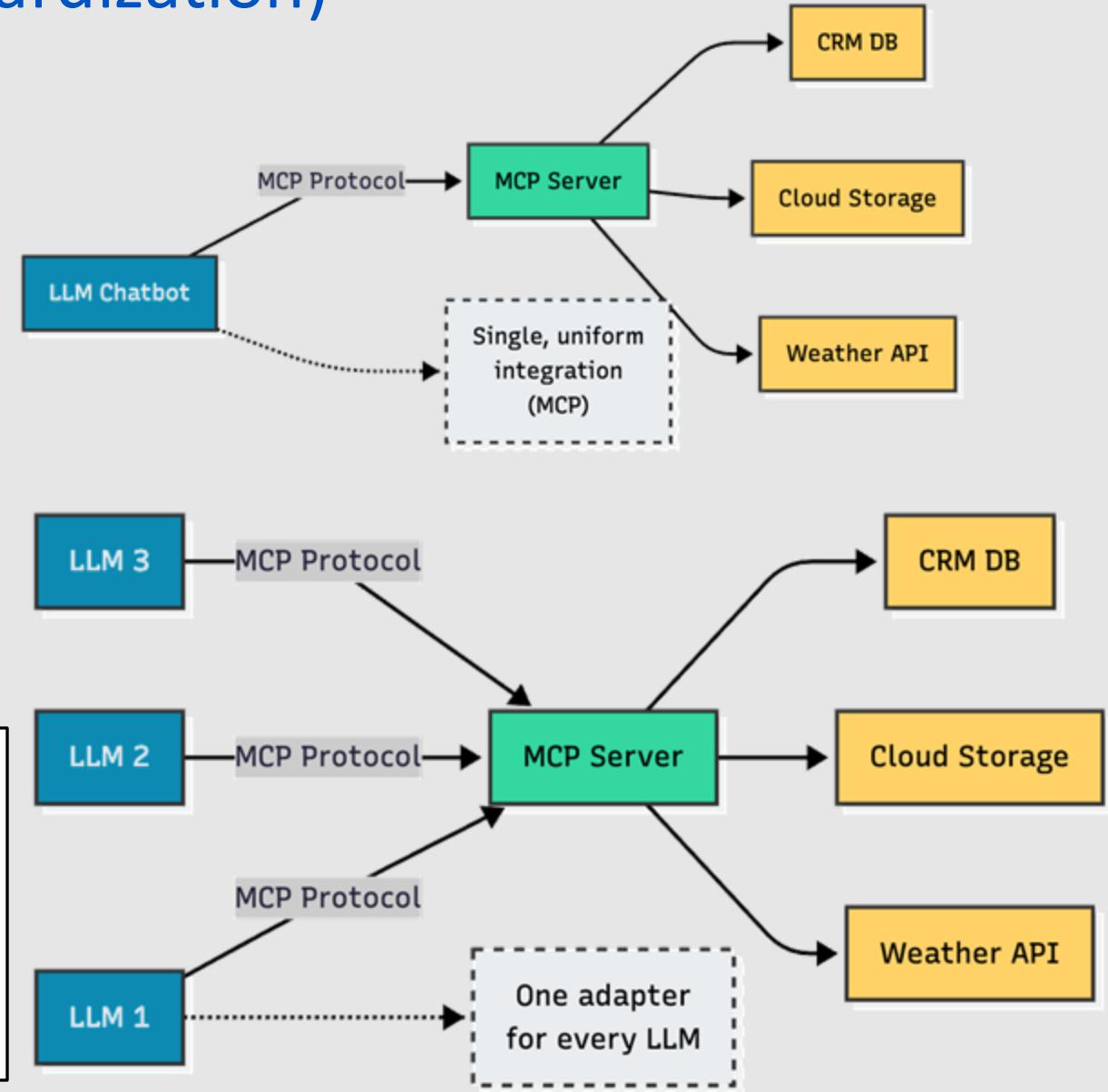


# How does MCP Help? (Standardization)

- MCP introduces one shared interface for everyone
- Each AI app implements the MCP client side once
- Each tool or data source implements the server side once
- AI apps can easily "plug in" new capabilities through MCP servers
- Integration count falls from  $M \times N$  to  $M + N$
- Outcome: faster lower cost, easier scaling

**Before MCP:** each agent must know each service's custom API.

**With MCP:** agents speak a single protocol; the MCP server handles all downstream specifics.





# How does MCP Help? (Discoverability)

- Makes tools for AI apps discoverable
- Standardization means any app using an MCP client can use the tool
- Discovery features of MCP means its easy to know how to use the tool
  - MCP providers can "advertise"
- Opens up much larger ecosystem for AI apps to leverage external tools

mcpservers.org

**MCP Servers** Home Remote Servers Resources ChatHub Submit

New: Remote MCP Servers

## Awesome MCP Servers

A collection of servers for the Model Context Protocol.

All Official Search Web Scraping Communication Productivity Development Database Cloud Service File System Cloud Storage Version Control Other

<b>Bright Data</b> <small>sponsor</small> Discover, extract, and interact with the web - one interface powering automated access across the public internet.  <a href="#">View Details</a>	<b>Everything</b> Reference / test server with prompts, resources, and tools  <a href="#">View Details</a>	<b>Fetch</b> Web content fetching and conversion for efficient LLM usage  <a href="#">View Details</a>	<b>Filesystem</b> Secure file operations with configurable access controls  <a href="#">View Details</a>
<b>Git</b> Tools to read, search, and manipulate Git repositories  <a href="#">View Details</a>	<b>Memory</b> Knowledge graph-based persistent memory system  <a href="#">View Details</a>	<b>Sequential Thinking</b> Dynamic and reflective problem-solving through thought sequences  <a href="#">View Details</a>	<b>Time</b> Time and timezone conversion capabilities  <a href="#">View Details</a>
<b>21st.dev Magic</b> <small>official</small> Create crafted UI components inspired by the best 21st.dev design engineers.  <a href="#">View Details</a>	<b>Adfin</b> <small>official</small> The only platform you need to get paid - all payments in one place, invoicing and accounting reconciliations with Adfin.  <a href="#">View Details</a>	<b>AgentQL</b> <small>official</small> Enable AI agents to get structured data from unstructured web with AgentQL.  <a href="#">View Details</a>	<b>AgentRPC</b> <small>official</small> Connect to any function, any language, across network boundaries using AgentRPC.  <a href="#">View Details</a>

# Overview of MCP Architecture

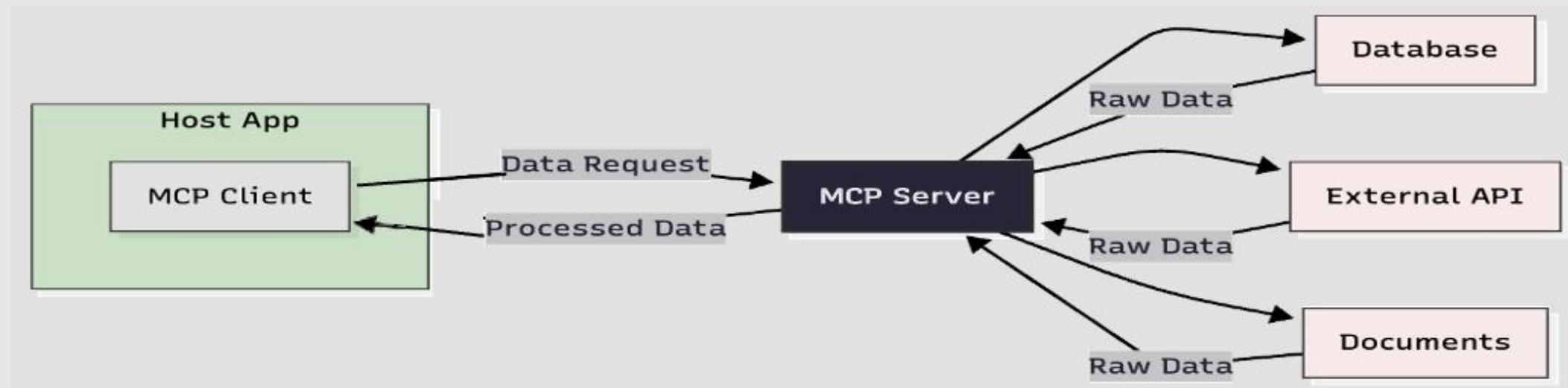
- MCP uses a **client-server architecture** for structured AI-tool communication.
- Enables systems to access external tools through a common protocol.

## Core Components

**MCP hosts** - apps (Claude Desktop, Windsurf, Cursor, VS Code, custom apps) that want to access data via MCP

**MCP clients** – implement client protocol and maintain 1:1 connections with MCP servers  
act as communications bridge

**MCP servers** – lightweight programs to expose specific capabilities (calling an API, reading data, etc.) via the server protocol





# GitHub's MCP Server

- Provides seamless integration with GitHub's APIs
- At [github.com/github/github-mcp-server](https://github.com/github/github-mcp-server)
- Enables advanced automation
- Simplifies interaction
- Useful for:
  - Automation
  - Getting data from repos
  - Building AI tools and apps
- Can connect to it from Copilot Agent

[README](#) [Code of conduct](#) [MIT license](#) [Security](#) [Edit](#) [☰](#)

---

## GitHub MCP Server

The GitHub MCP Server is a [Model Context Protocol \(MCP\)](#) server that provides seamless integration with GitHub APIs, enabling advanced automation and interaction capabilities for developers and tools.

### Use Cases

- Automating GitHub workflows and processes.
- Extracting and analyzing data from GitHub repositories.
- Building AI powered tools and applications that interact with GitHub's ecosystem.

---

### Remote GitHub MCP Server

[VS Code Install Server](#) [VS Code Insiders Install Server](#)

The remote GitHub MCP Server is hosted by GitHub and provides the easiest method for getting up and running. If your MCP host does not support remote MCP servers, don't worry! You can use the [local version of the GitHub MCP Server](#) instead.

### Prerequisites

1. An MCP host that supports the latest MCP specification and remote servers, such as [VS Code](#).

### Installation

### Usage with VS Code

## Lab 4 – Extending Copilot's Utility via MCP Servers

**Purpose:** In this lab, we'll see how to connect GitHub Copilot to the GitHub MCP Server.



# GitHub Copilot Extensions

- Integrate external tools and services into Copilot Chat
- Function as GitHub Apps (server-side) or VS Code chat participants (client-side)
- Enable Copilot to query APIs, run commands, fetch data – all within the chat
- Allows users to interact with external tools and services directly in chat
- Can be:
  - Marketplace-listed: Public with listing on GitHub Marketplace
  - Public: Shareable GitHub App for teams or open-source use
  - Private: Used within your org for internal APIs and tools

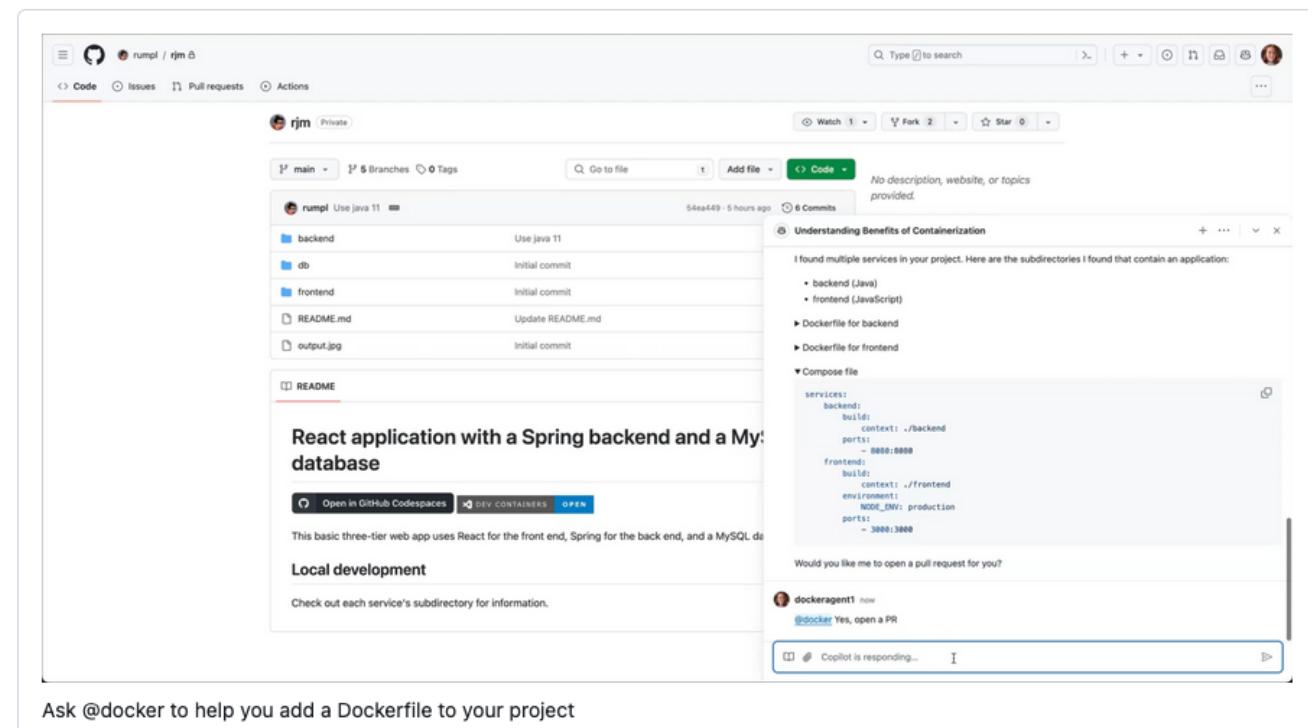
 Docker for GitHub Copilot 
Copilot

Learn about containerization, generate Docker assets and analyze project vulnerabilities in GitHub Copilot

## Docker in GitHub Copilot

- Generate the right Docker assets for your project: Get help containerizing your application and watch it generate the `Dockerfile`, `docker-compose.yml`, and `.dockerignore` files tailored to your project's languages and file structure: "`@docker` How would I use Docker to containerize this project?"
- Open a Pull Request with the assets to save you time: With your consent, the Docker extension can open a PR with these generated Docker assets on GitHub, allowing you to review and merge them at your convenience.
- Find project vulnerabilities with Docker Scout: The Docker extension also integrates with Docker Scout to surface a high-level summary of detected vulnerabilities and provide the next steps to continue using Scout in your terminal via CLI: "`@docker` can you help me find vulnerabilities in my project?"

Docker for GitHub Copilot in early access. Have feedback? Please [file an issue](#).



The screenshot shows a GitHub repository named "rumpl / rjm" with a private status. The repository has 1 watch, 2 forks, and 0 stars. It contains 6 commits from "rumpl" made 5 hours ago. The repository description is "No description, website, or topics provided." A sidebar on the right titled "Understanding Benefits of Containerization" lists "backend", "db", and "frontend" as subdirectories. It also shows a "Compose file" section with the following YAML code:

```

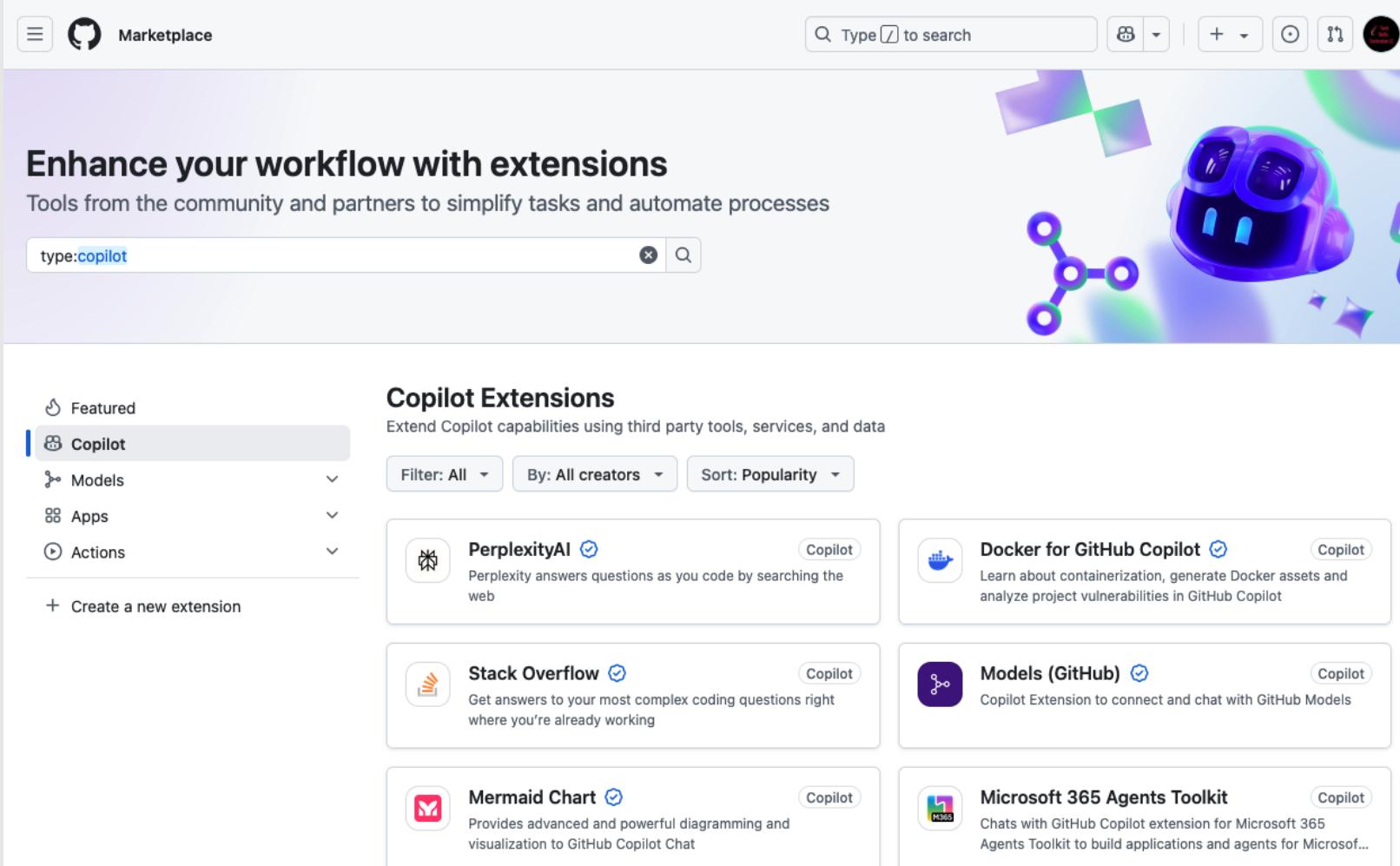
services:
  backend:
    build:
      context: ./Backend
      ports:
        - 8080:8080
  frontend:
    build:
      context: ./Frontend
      environment:
        NODE_ENV: production
    ports:
      - 3000:3000
  
```

A message at the bottom of the sidebar says "Would you like me to open a pull request for you?" Below the sidebar, a message from "dockeragent1" says "@docker Yes, open a PR". At the bottom of the page, a message from "Copilot" says "Copilot is responding..."



# Extensions Marketplace

- Browse first-party and partner extensions: **Docker, Azure, Sentry, Jira, MongoDB, and more**
- Marketplace listings include descriptions, permissions, and install flows
- Supports both individual and organization-level installations



The screenshot shows the GitHub Extensions Marketplace interface. At the top, there's a navigation bar with a search bar containing "Type / to search". Below the search bar, a banner says "Enhance your workflow with extensions" and "Tools from the community and partners to simplify tasks and automate processes". A search bar below the banner has "type:copilot" typed into it. To the right of the search bar is a decorative graphic featuring a purple and green abstract shape and a blue 3D-style extension icon.

The main content area is titled "Copilot Extensions" and describes how to "Extend Copilot capabilities using third party tools, services, and data". It includes filters for "All", "By: All creators", and "Sort: Popularity". Below the filters, there are six extension cards:

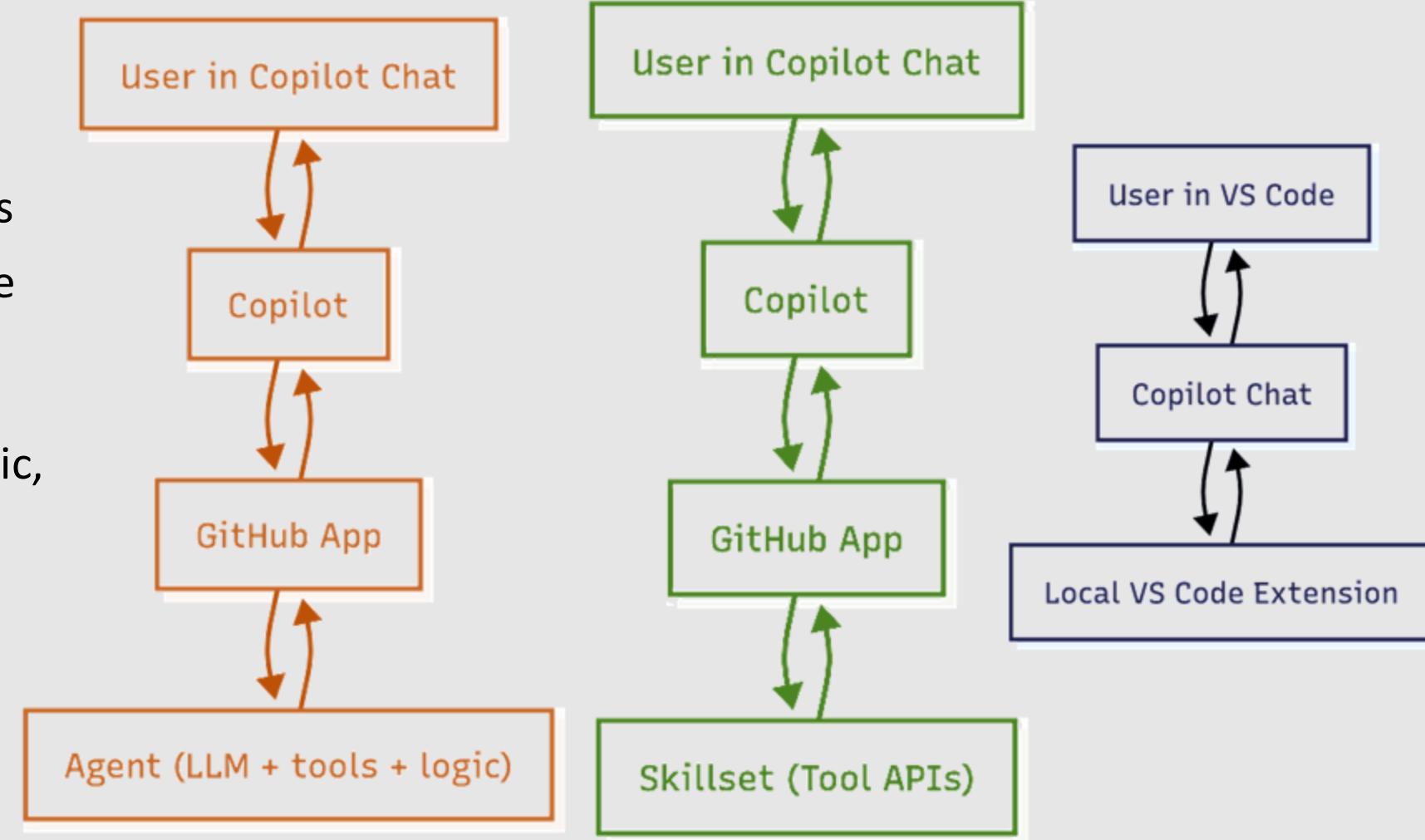
- PerplexityAI** Copilot: Answers questions as you code by searching the web.
- Docker for GitHub Copilot** Copilot: Learn about containerization, generate Docker assets and analyze project vulnerabilities in GitHub Copilot.
- Stack Overflow** Copilot: Get answers to your most complex coding questions right where you're already working.
- Models (GitHub)** Copilot: Copilot Extension to connect and chat with GitHub Models.
- Mermaid Chart** Copilot: Provides advanced and powerful diagramming and visualization to GitHub Copilot Chat.
- Microsoft 365 Agents Toolkit** Copilot: Chats with GitHub Copilot extension for Microsoft 365 Agents Toolkit to build applications and agents for Microsoft...

See: [https://github.com/marketplace?type=apps&copilot\\_app=true](https://github.com/marketplace?type=apps&copilot_app=true)



# Building Your Own Extensions

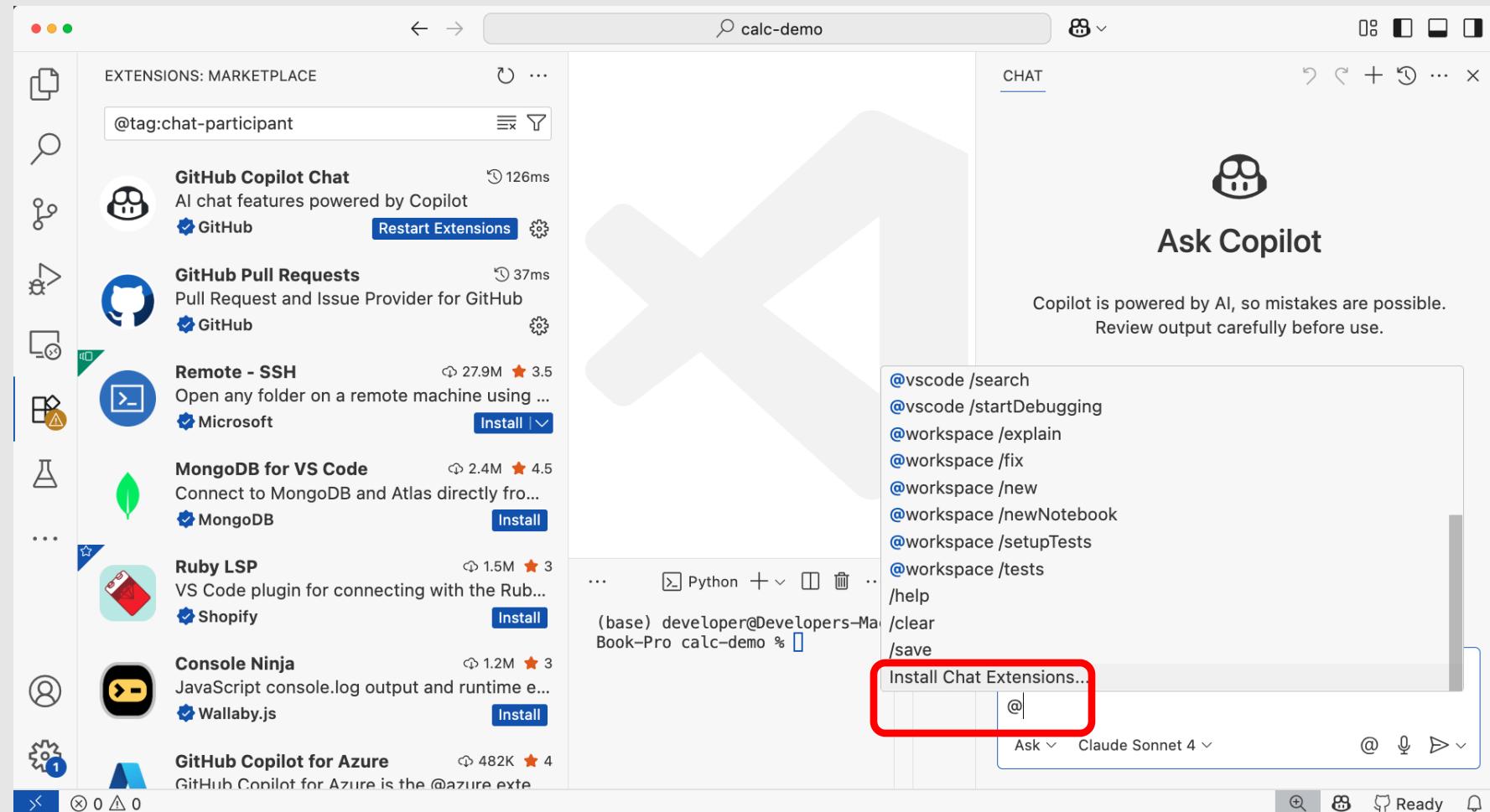
- Use GitHub App + “skillset” or “agent” deployed server-side
- Optionally build VS Code *chat participants* for local-first tasks
- Register tool metadata, handle user queries, and emit structured output
- Choose visibility (private, public, marketplace-listed)
- Use GitHub’s SDK, CLI tooling, API references & samples





# Option: Install Chat Participants through VS Code

- In Copilot Chat, type @
- Scroll down and find "Install Chat Extensions"
- List appears on left

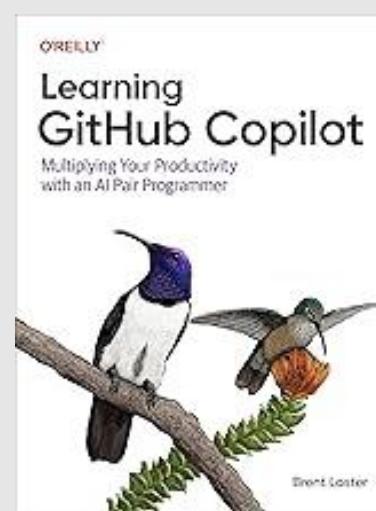
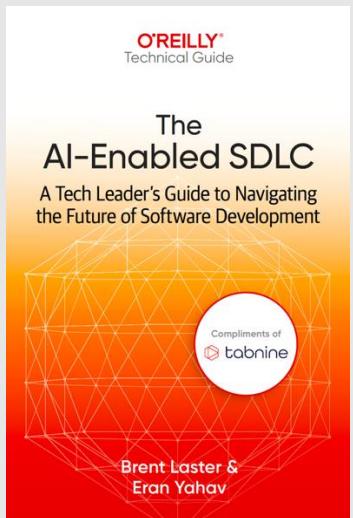
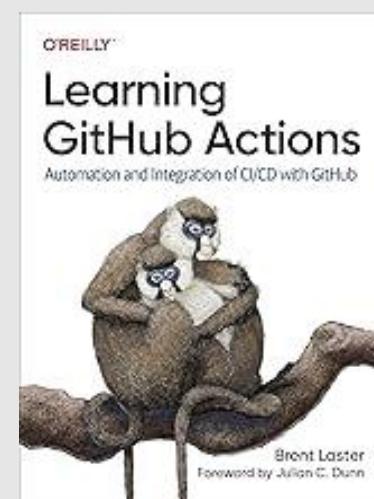
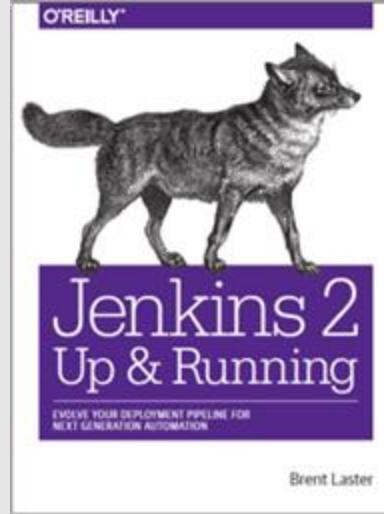
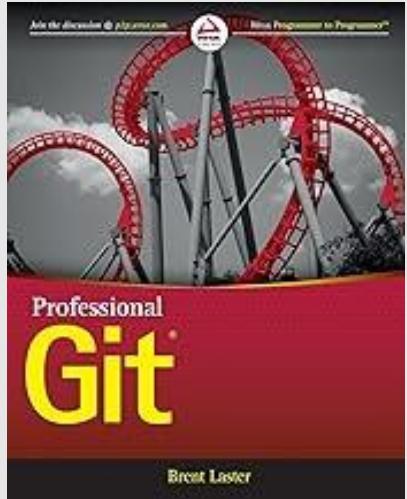




# That's all - thanks!

**Contact:** [training@getskillsnow.com](mailto:training@getskillsnow.com)

[techskillstransformations.com](http://techskillstransformations.com)  
[getskillsnow.com](http://getskillsnow.com)



[techskillstransformations.com](http://techskillstransformations.com)

Grow your marketability and career potential with in-demand skills!

**TECH SKILLS TRANSFORMATIONS LLC**

HOME ABOUT US CORPORATE TRAINING CUSTOM TRAINING TESTIMONIALS PUBLICATIONS MORE ▾

**THE CONTINUOUS UPSKILLING COMPANY!**

Get the instruction you need to upskill now! [Contact us](#) to see how we can help you understand & use the tech you need.

UPCOMING LIVE TRAINING WITH O'REILLY MEDIA!

**TRAINING TOPICS**