GitHub Copilot

(AI Pair Programming)

GitHub Copilot

Contents

- AI Pair Programming
- GitHub
  - GitHub Pro
  - GitHub Enterprise
- GitHub Copliot
  - Copilot Individual
  - Copilot Enterprise

AI Pair Programming

- GitHub Copilot
- Gemini Code Assist
- CodiumAI
- Gemini in Android Studio (Studio Bot)

- https://www.codium.ai/codiumai-vs-github-copilot/
- https://www.kajetandomagala.com/codeium-vs-copilot/
- https://developer.android.com/studio/preview/gemini

GitHub (Basic)

- https://medium.com/@naruapon/การใช้งาน-git-8a563531d1f1

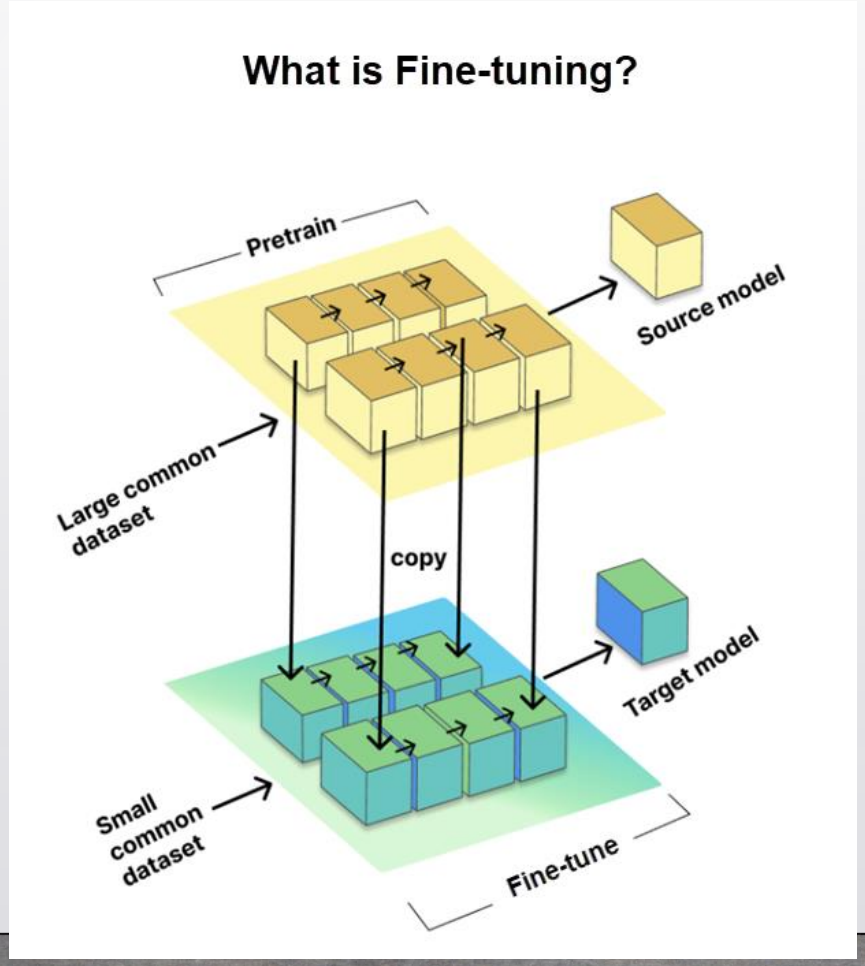
GitHub Pro

- GitHub Support via email
- 3,000 GitHub Actions minutes per month
- 2 GB GitHub Packages storage
- 180 GitHub Codespaces core hours per month
- 20 GB GitHub Codespaces storage per month
- Advanced tools and insights in private repositories

How GitHub Copilot Works

- GitHub Copilot is powered by OpenAI Codex. The auto-generated suggestions come from the context within the file, like function names, code comments, docstrings, file names, cursor position, and more. Based on this information, Copilot suggests code snippets that developers can accept by simply pressing the Tab key on their keyboards.
- The AI tool understands TypeScript, Python, JavaScript, Ruby, and dozens of other common languages.

GitHub Copilot Large Language Models (LLMs)



How GitHub Copilot Works

Copilot Individual

- Who's it for: Individual developers, freelancers, students, and educators who want to code faster.
- Cost: $10 per month or $100 per year.
- Free for verified students, teachers, and maintainers of popular open-source projects.

What’s included:

- Chat
  - Unlimited messages and interactions
  - Context-aware coding support and explanations
  - Debugging and security remediation assistance
- Code completion
  - Real-time code suggestions
  - Comments to code

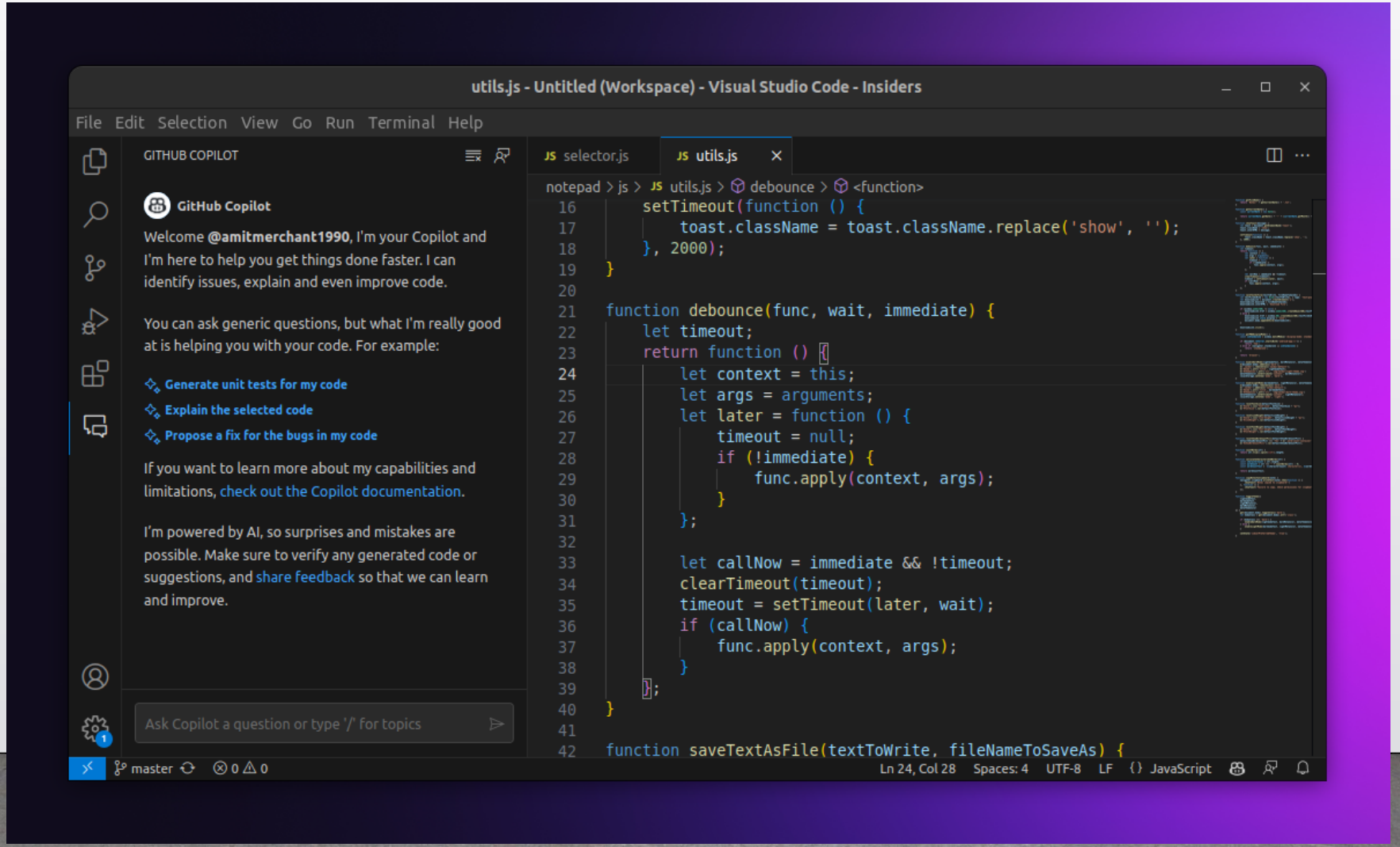
What’s included:

- Smart actions
  - Inline chat and prompt suggestions
  - Slash commands and context variables
  - Commit message generation
- Supported environments
  - IDE, CLI, and GitHub Mobile
- Management and policies
  - Public code filter

Copilot Enterprise

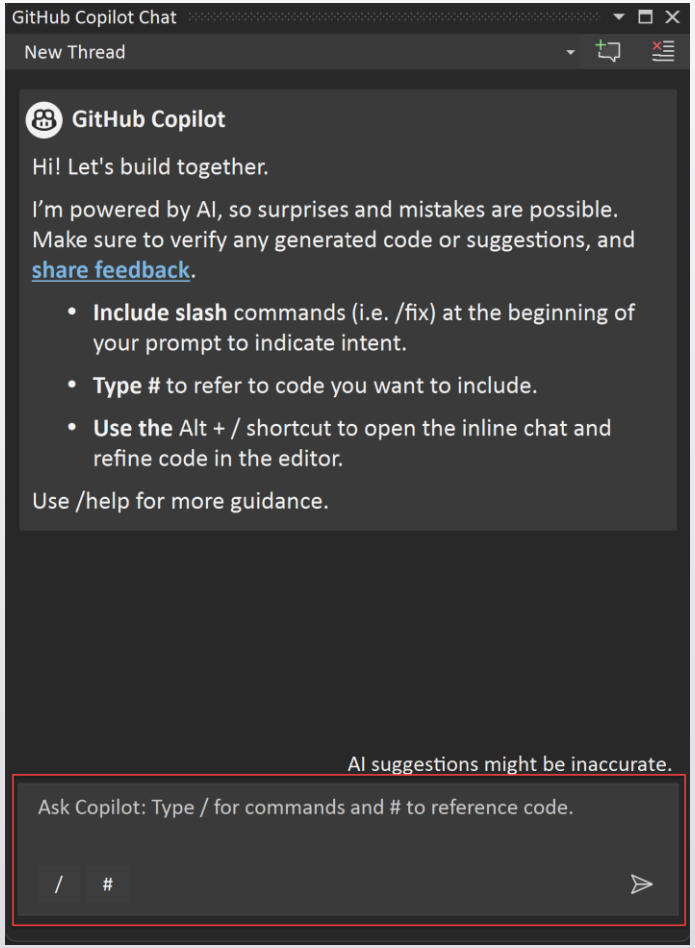
- **Who’s it for:** Companies that want to customize GitHub Copilot to their organization and infuse AI across the developer workflow.
- **Cost:** $39 per user per month.

GitHub Copilot Chat (Visual Studio Code)



Ask questions in the chat window

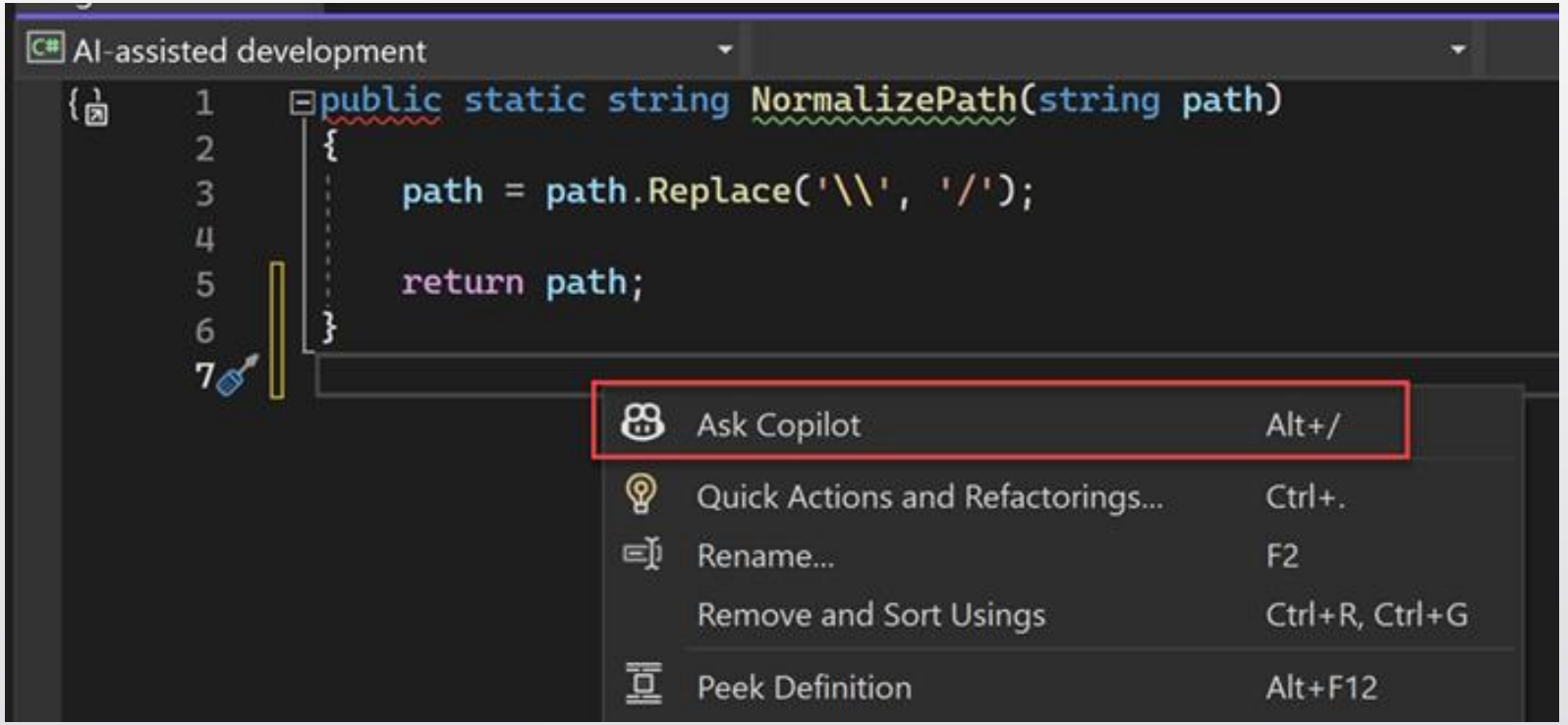
- In Visual Studio, select View > GitHub Copilot Chat.



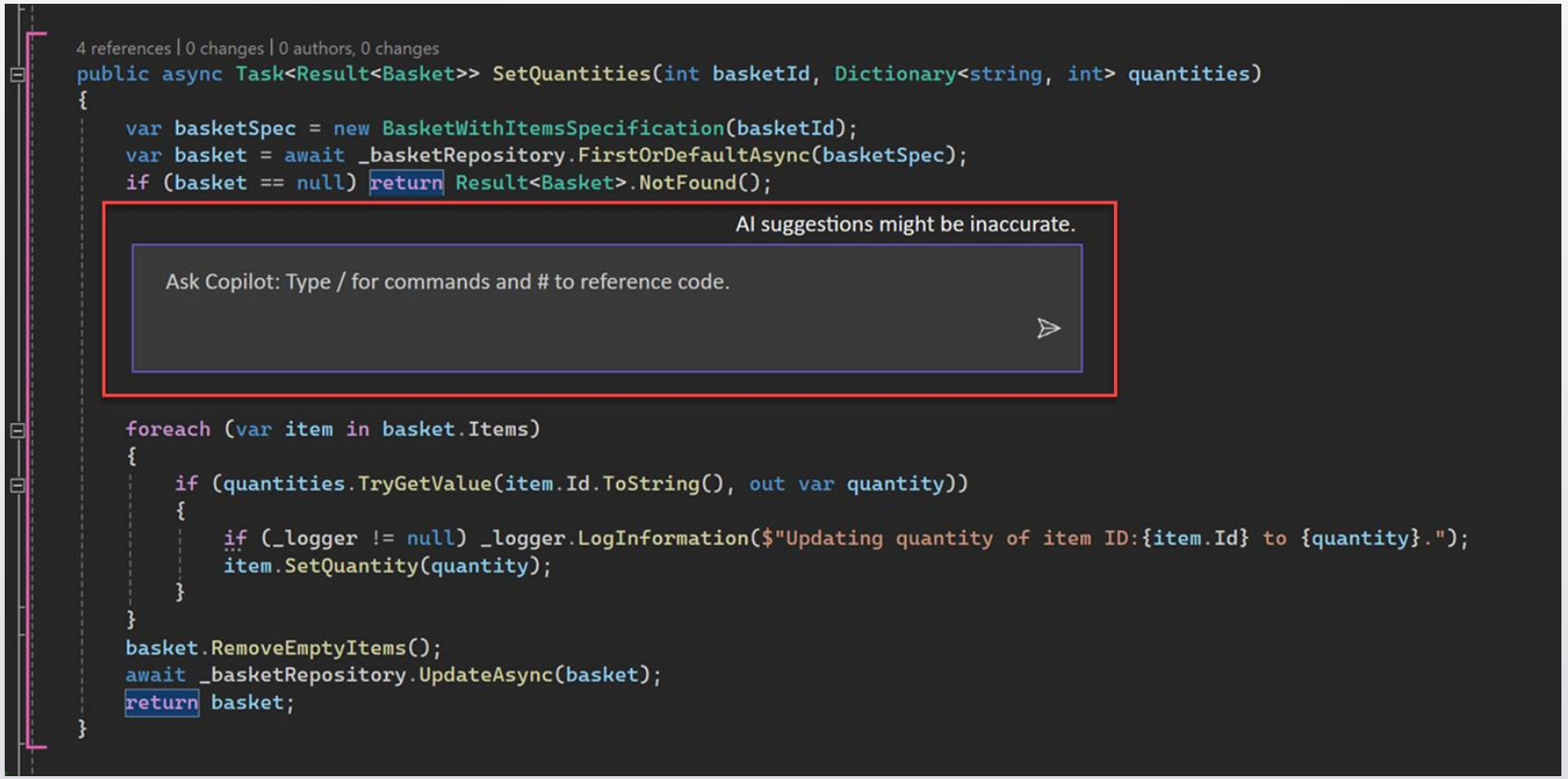
GitHub Copilot in Visual Studio

- Inline Chat View
  - Ctrl+I (Visual Studio Code)
  - Alt + / (Visual Studio)

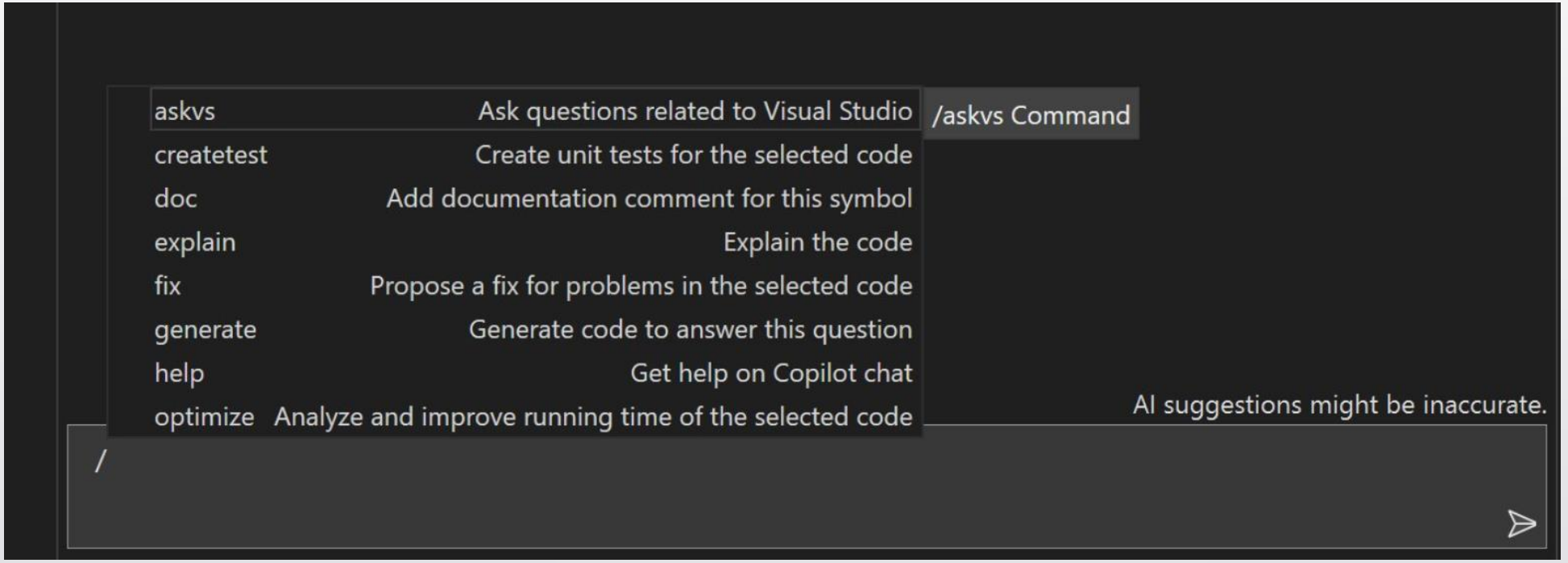
Inline Chat View



Inline Chat View



Slash Commands to State Intent

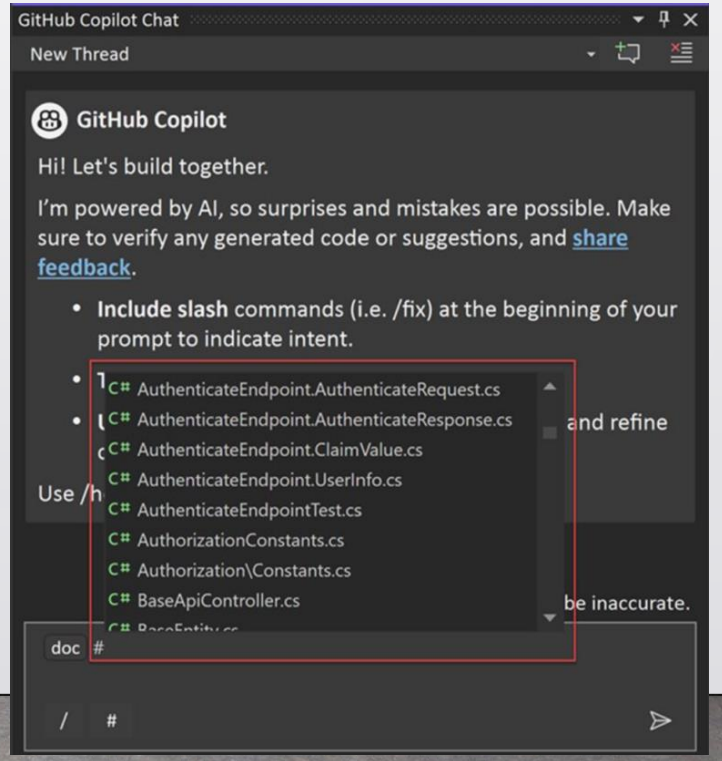


Chat Window and Inline Chat

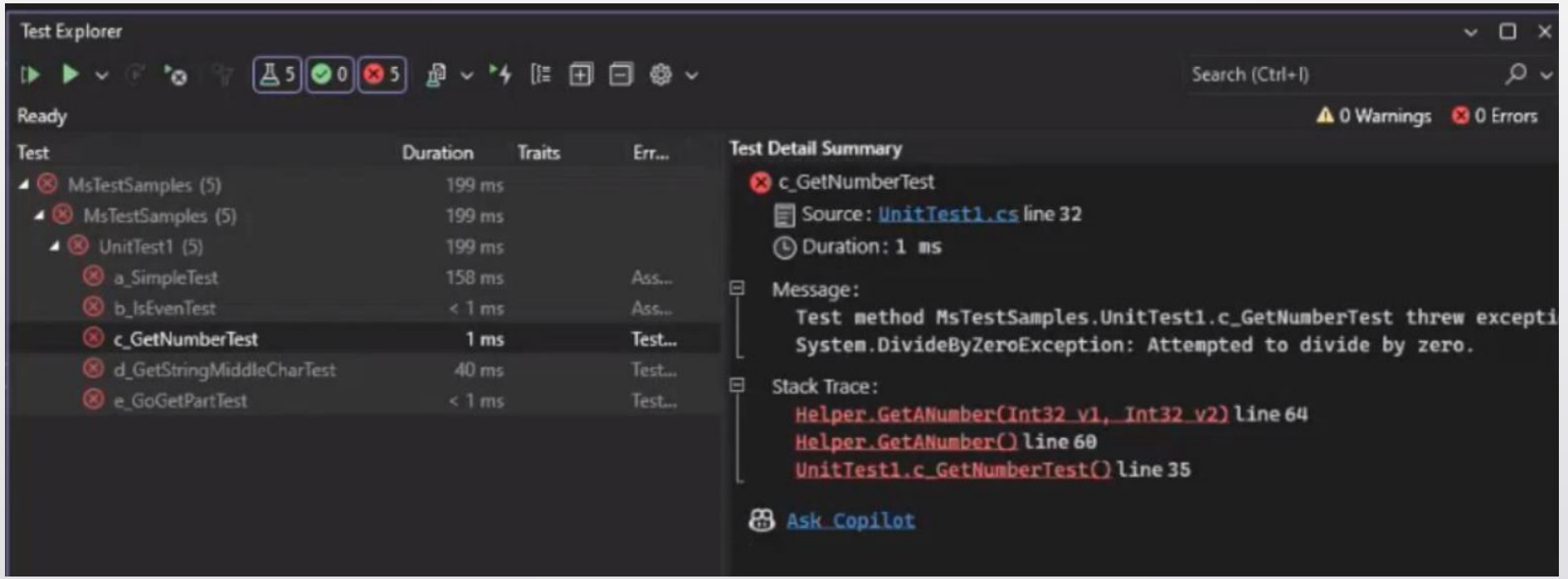
Command	Usage	Chat window	Inline chat
/doc	Add comments for specified or selected code. Examples: - `/doc DeleteBasketAsync method in BasketService.cs` - select desired code and enter `/doc`	Yes	Yes
/explain	Get code explanations. Examples: - `/explain the AddItemToBasket method in BasketService.cs` - select desired code and enter `/explain`	Yes	Yes
/fix	Propose a fix for problems in the selected code. Examples: - `/fix the SetQuantities method in BasketService.cs` - select desired code and enter `/fix`	Yes	Yes
/generate	Generate code to answer specified question. Example: `/generate code to add two numbers in Calculator.cs`	Yes	Yes

Context Variables to Refine Your Scope

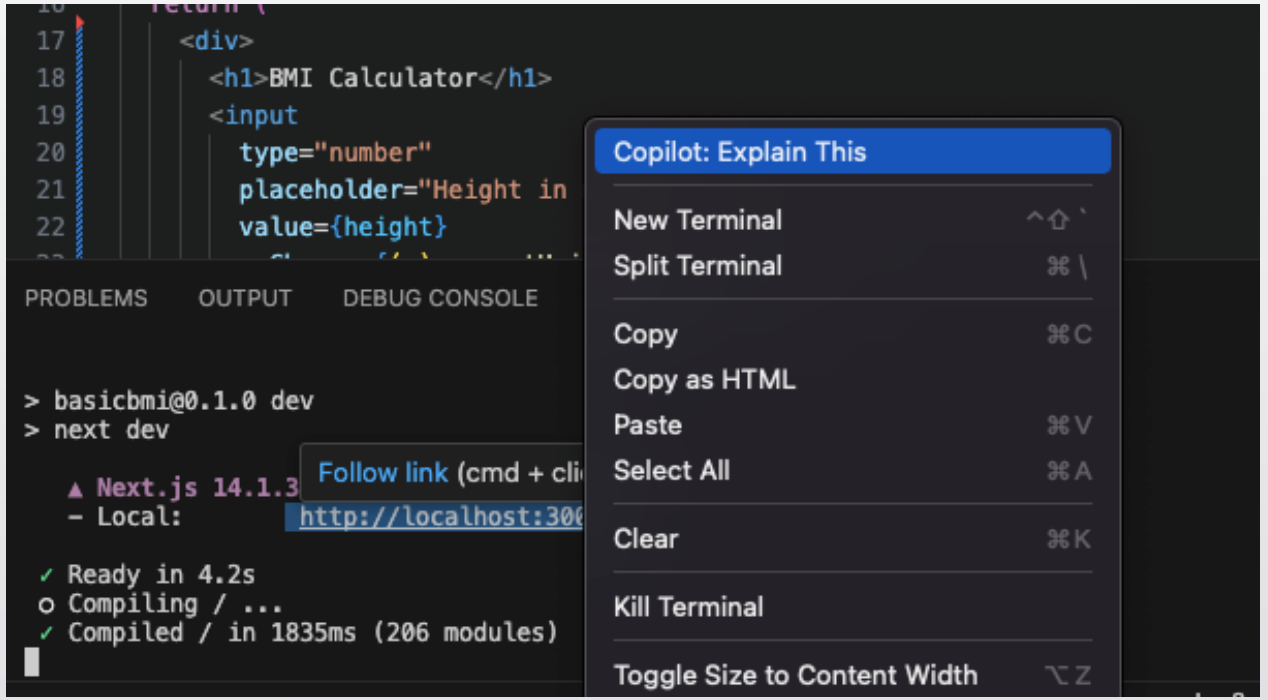
- The context variables feature allows you to specify files from your solution in your questions by using the # symbol.



Analyze and Fix Test Window Failures



Explain terminal



Github Copilot @workspace

- Github Copilot @workspace — New Development Experience
- https://medium.com/@yar.dobroskok/github-copilot-workspace-new-development-experience-d69857fbd067

yardobr

how do I start the current project?

GitHub Copilot

To start a project in Visual Studio Code, you typically need to follow these steps:

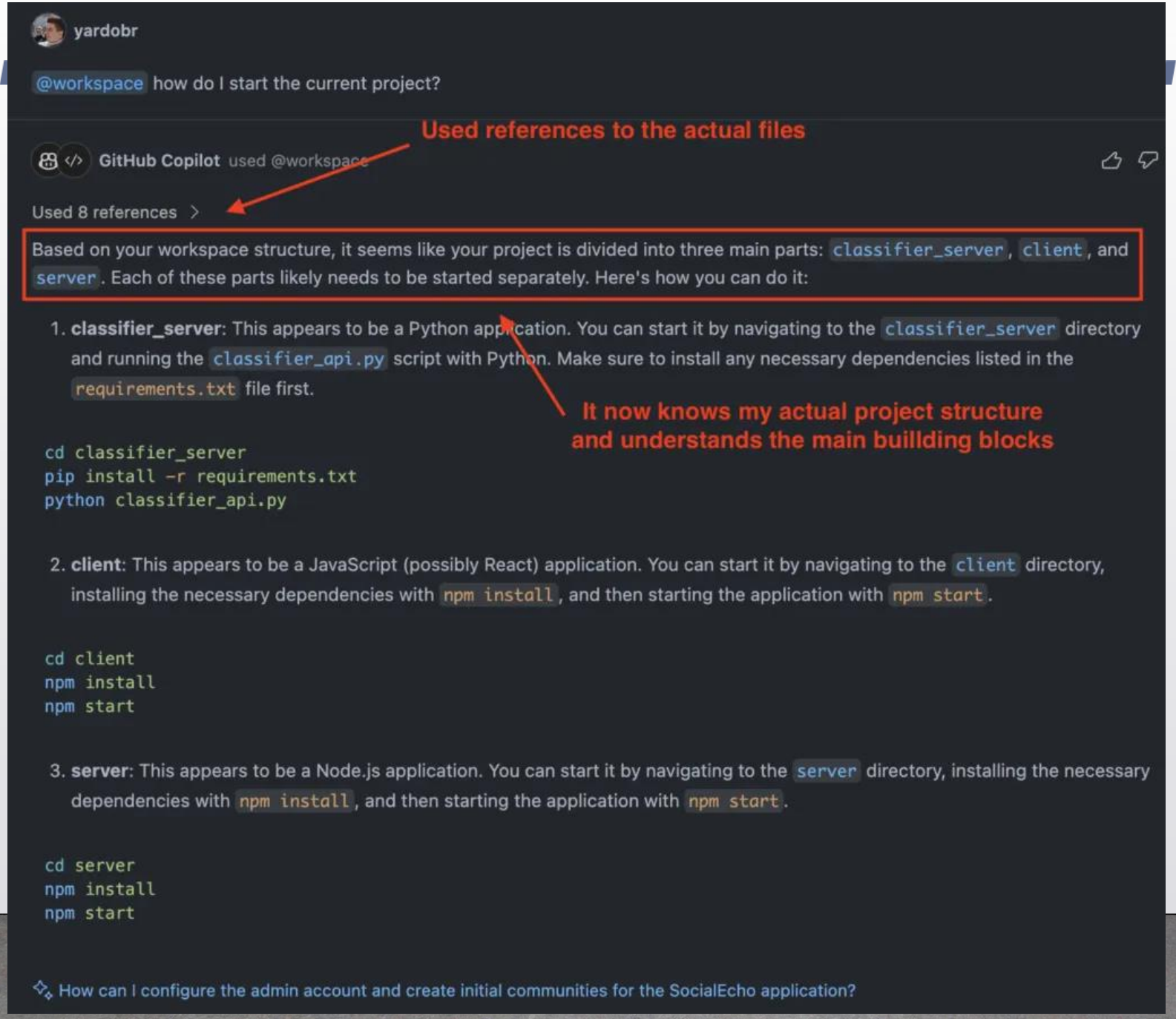
1. Open the terminal in Visual Studio Code. You can do this by going to the top menu and selecting `View -> Terminal`, or by using the shortcut `Ctrl+` backtick (`).

2. Navigate to the project's directory. You can do this by typing `cd path/to/your/project` in the terminal, replacing `path/to/your/project` with the actual path to your project.

3. Depending on the type of your project, you will need to run a specific command to start it. Here are some common examples:

    - For a Node.js project, you might run `npm start` or `node app.js` (replace `app.js` with the entry point to your application).
    - For a Python project, you might run `python main.py` (replace `main.py` with the main Python file in your project).
    - For a Java project, you might run `javac Main.java` followed by `java Main` (replace `Main` with the main class in your project).

Please replace the commands with the ones specific to your project. If you're not sure what command to run, check the documentation of the technology you're using or the README file of your project.

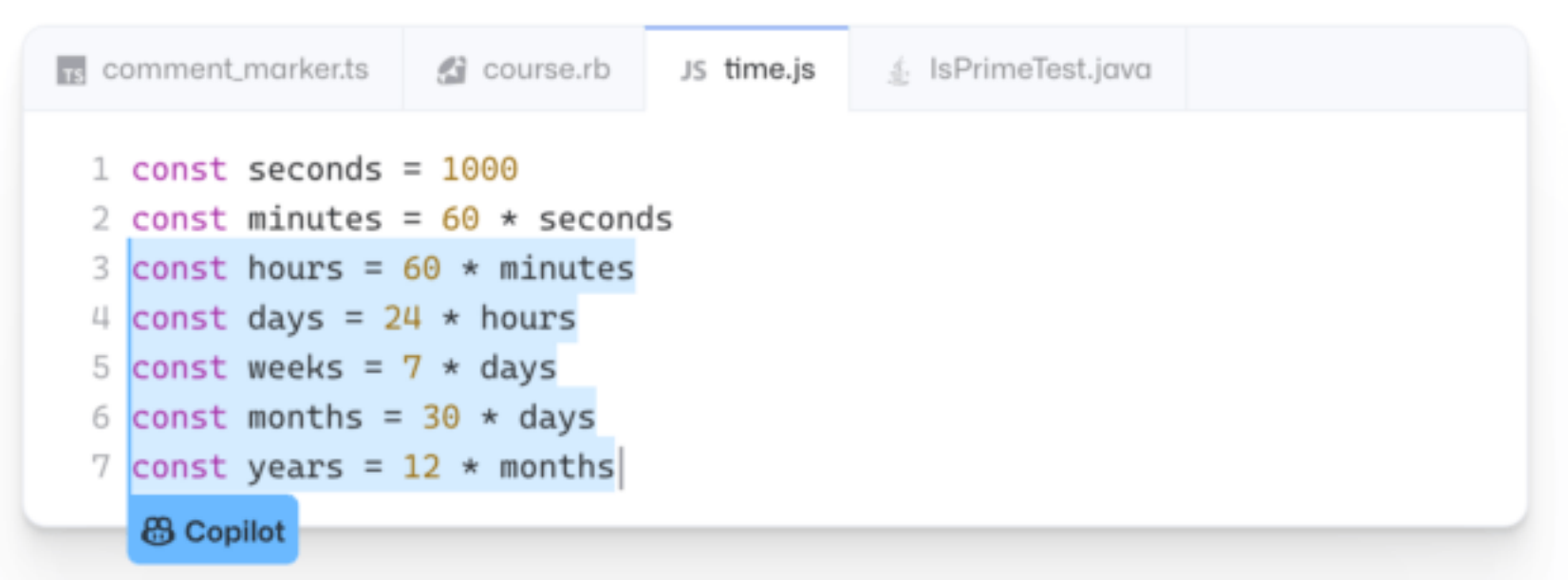

yardobr
@workspace how do I start the current project?
Used references to the actual files
GitHub Copilot used @workspace
Used 8 references
Based on your workspace structure, it seems like your project is divided into three main parts: classifier_server, client, and server. Each of these parts likely needs to be started separately. Here's how you can do it:
1. classifier_server: This appears to be a Python application. You can start it by navigating to the classifier_server directory and running the classifier_api.py script with Python. Make sure to install any necessary dependencies listed in the requirements.txt file first.
It now knows my actual project structure and understands the main buillding blocks
cd classifier_server
pip install -r requirements.txt
python classifier_api.py
2. client: This appears to be a JavaScript (possibly React) application. You can start it by navigating to the client directory, installing the necessary dependencies with npm install, and then starting the application with npm start.
cd client
npm install
npm start
3. server: This appears to be a Node.js application. You can start it by navigating to the server directory, installing the necessary dependencies with npm install, and then starting the application with npm start.
cd server
npm install
npm start
How can I configure the admin account and create initial communities for the SocialEcho application?

The Ultimate Manual to GitHub Copilot

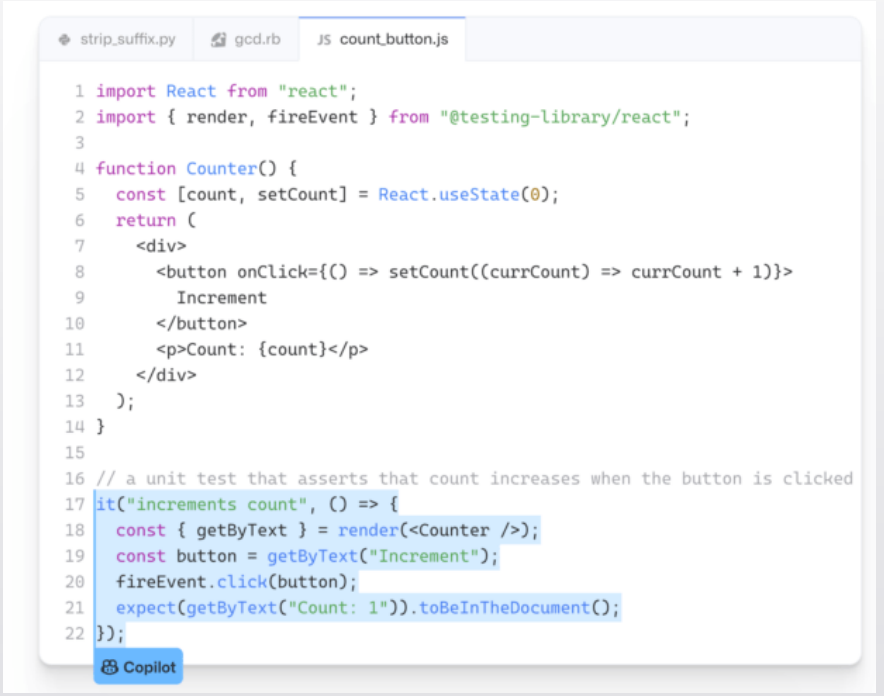
- Example #1: Convert Comments to Code
- Example #2: Autofill Repetitive Code
- Example #3: Run Tests
- Example #4: Navigating Unfamiliar Territory
- Example #5: Creating an Application Entirely With Copilot

Example #1: Convert Comments to Code

Example #2: Autofill Repetitive Code



Example #3: Run Tests



Example #4: Navigating Unfamiliar Territory

How to Get Started With GitHub Copilot

- Step 1: Narrow Down Your Use Case and Goal
- Step 2: Install the GitHub Copilot Extension
- Step 3: Learn the GitHub Copilot Keyboard Shortcuts
- Step 4: Start Writing Your Code and Review the Suggestions
- Step 5: Make Edits and Test Your Code

Step 2: Install the GitHub Copilot Extension

- GitHub Copilot — Visual Studio Code Marketplace
- GitHub Copilot — JetBrains Marketplace
- Neovim Plugin for GitHub Copilot

GitHub Copilot — Visual Studio Code Marketplace

- https://marketplace.visualstudio.com/items?itemName=GitHub.copilot

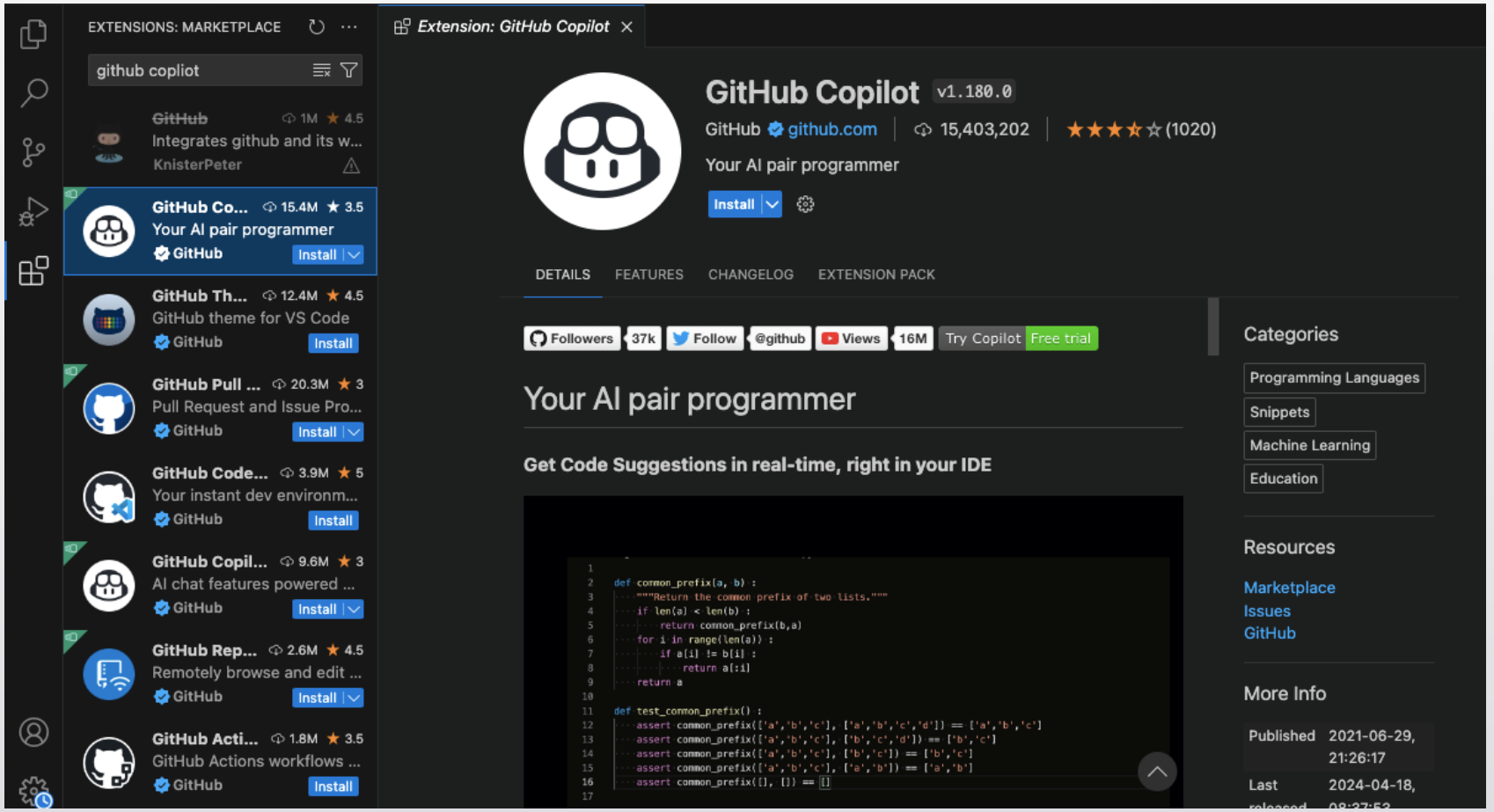
How to Install GitHub Copilot in Visual Studio

- A GitHub account. You can sign up for free at GitHub.com.
- An active GitHub Copilot subscription (Individual, Business, or Enterprise).
- Visual Studio 2022 17.6 or later installed.

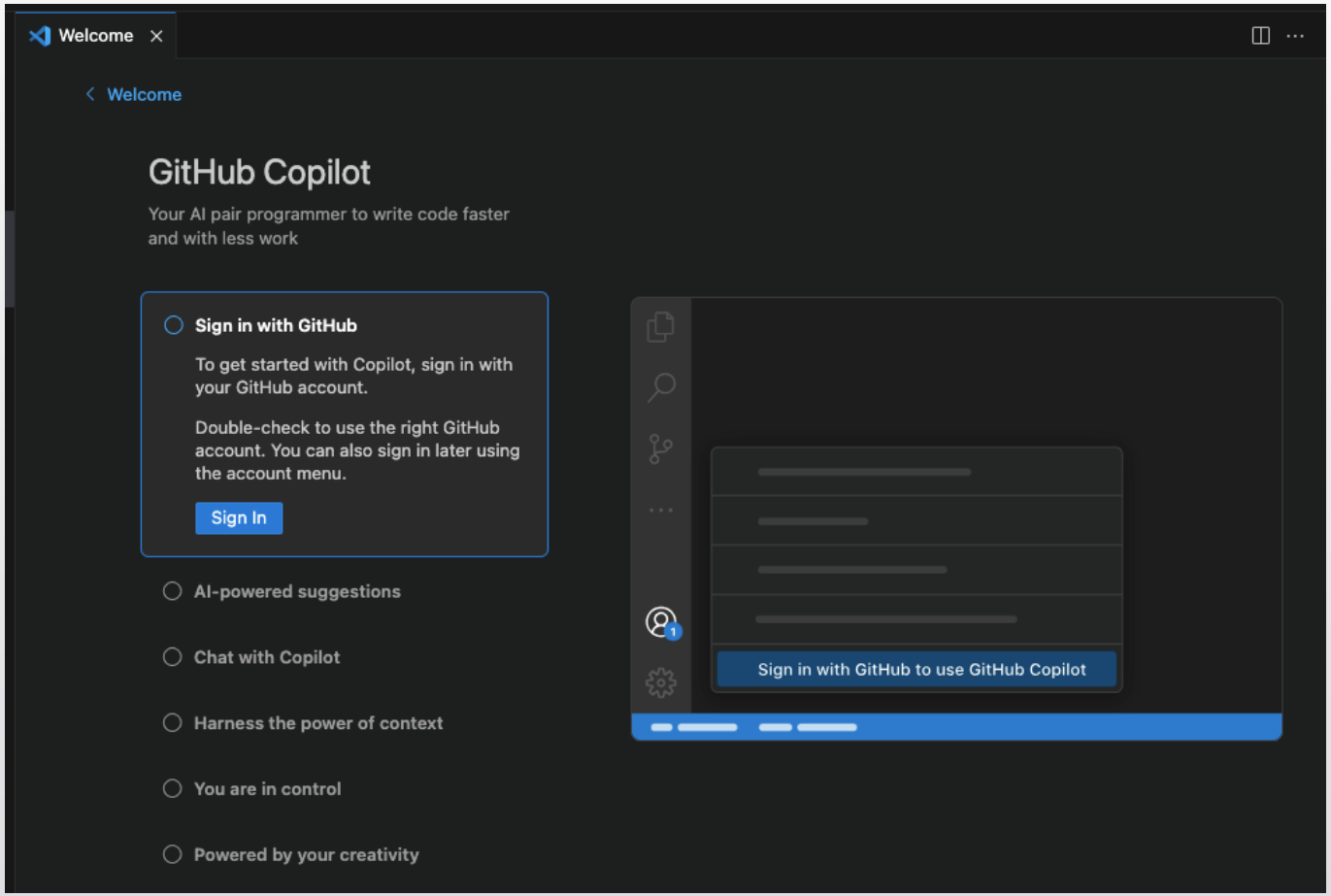
How to Install GitHub Copilot in Visual Studio Code

- In the Visual Studio Code Marketplace, go to the GitHub Copilot extension page and click Install.
- A popup will appear, asking to open Visual Studio Code. ...
- In the "Extension: GitHub Copilot" tab in Visual Studio Code, click Install.

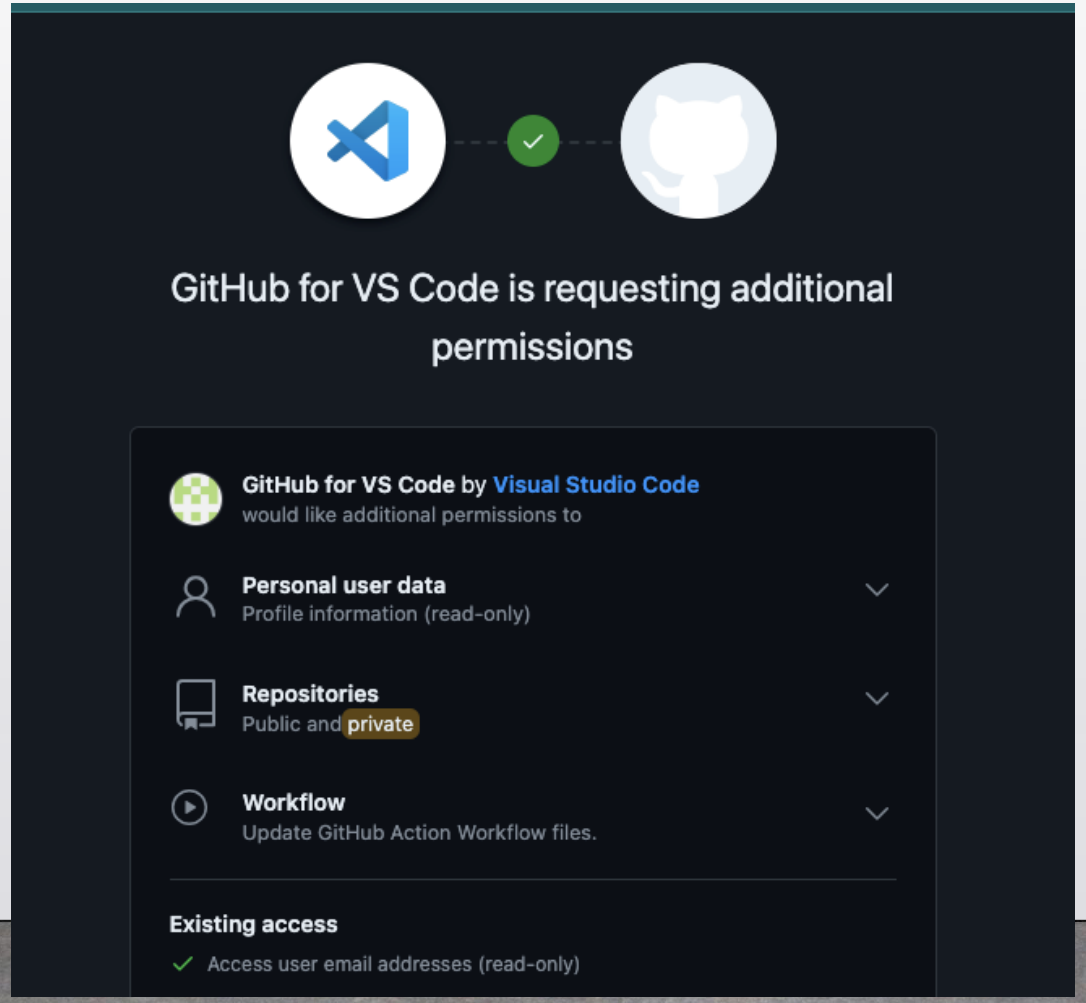
Install GitHub Copilot in Visual Studio Code



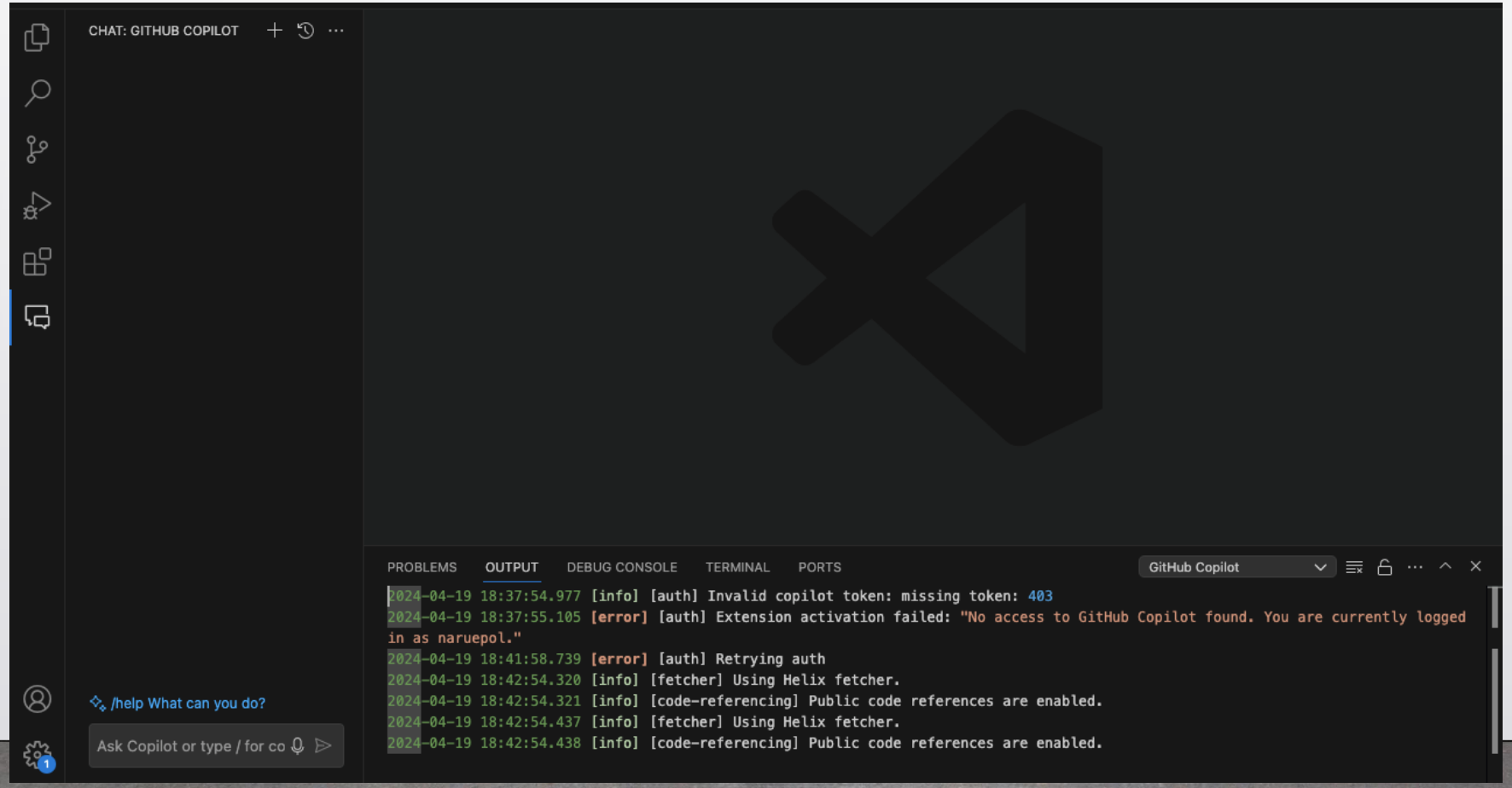
Install GitHub Copilot in Visual Studio Code



Install GitHub Copilot in Visual Studio Code



Chat GitHub Copliot

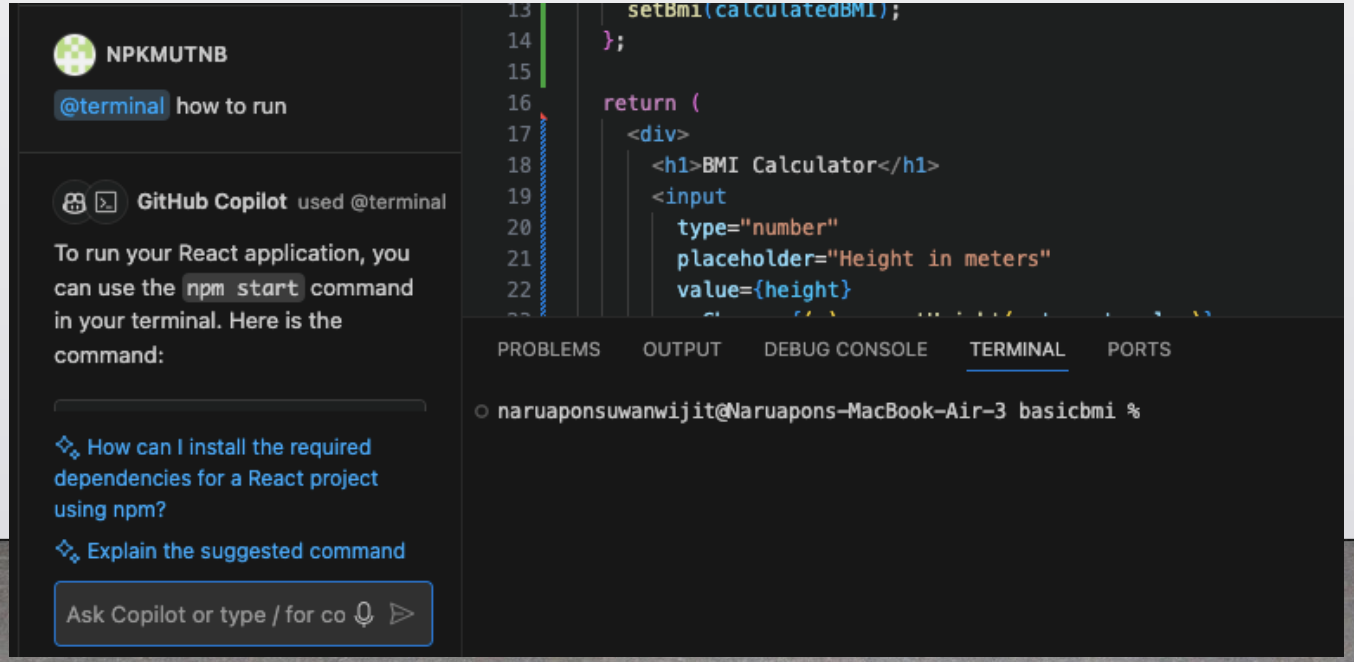


Step 3: Learn the GitHub Copilot Keyboard Shortcuts

- Accept inline code suggestion — **Tab**
- Dismiss inline code suggestion — **Esc**
- Show next suggestion — **Alt +] or Option (⌥) +]**
- Show previous suggestion — **Alt + [or Option (⌥) + [**
- Trigger suggestion — **Alt + \ or Option (⌥) + **
- Open 10 suggestions in a separate pane — **Ctrl + Enter**

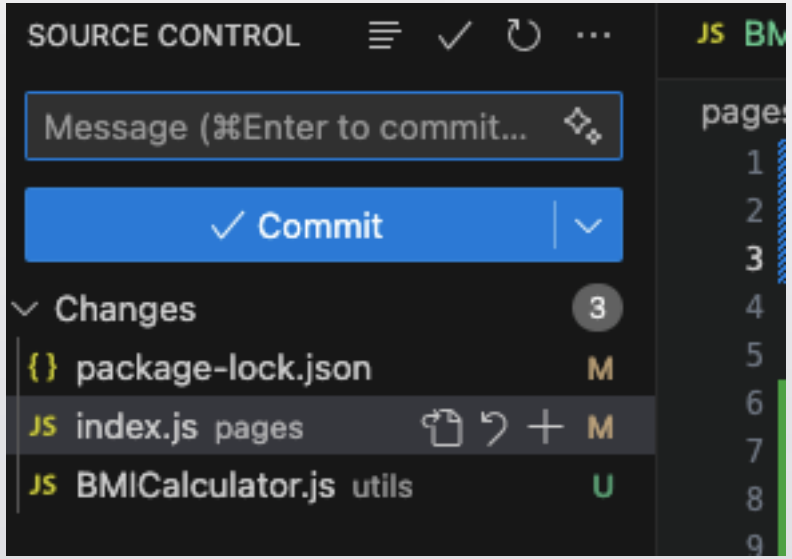
Run terminal commands from GitHub Copilot Chat

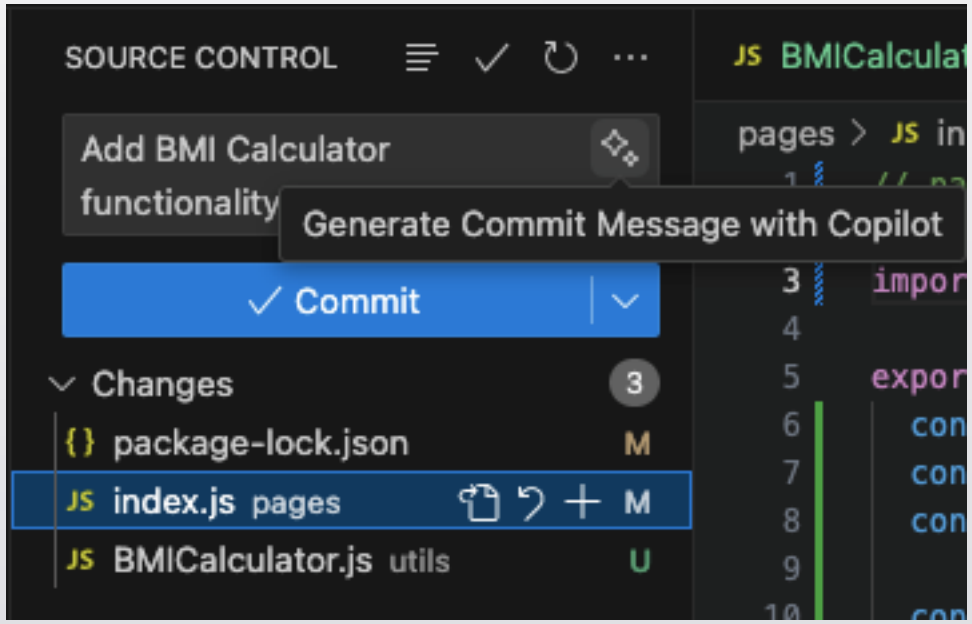
- If you ever forget how to run a particular command when you're working in your VS Code, GitHub Copilot Chat is here to help! With the new @terminal agent in VS Code, you can ask GitHub Copilot how to run a particular command.



Generate commit messages

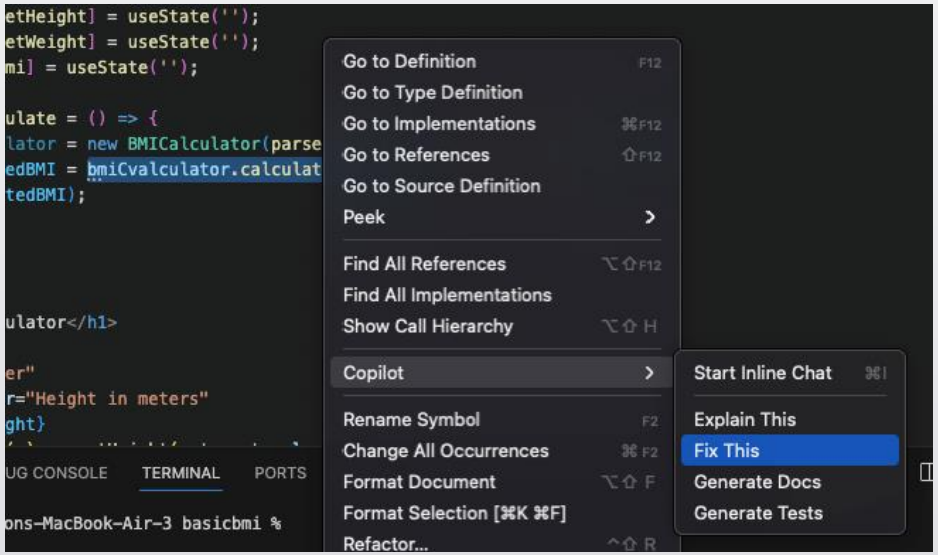
- GitHub Copilot can help you generate commit messages right in your IDE.

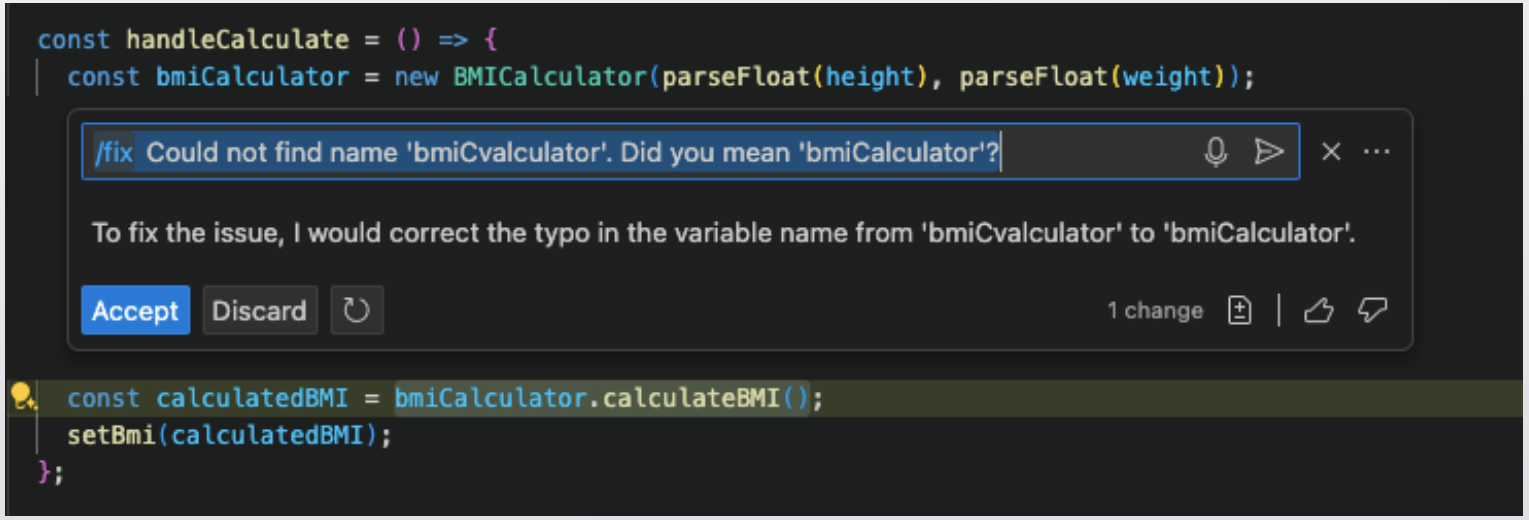




Fix code inline

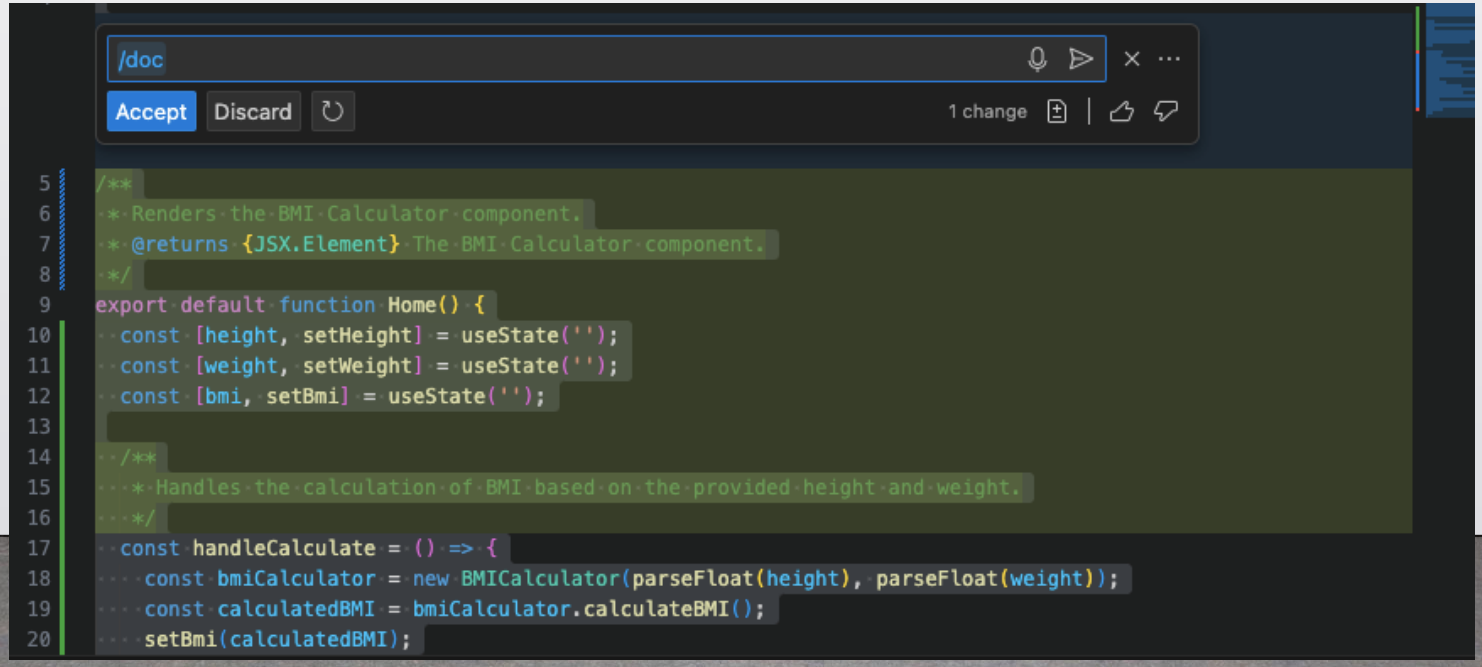
- highlight the code you want to fix, right click, and select “Fix using Copilot.”





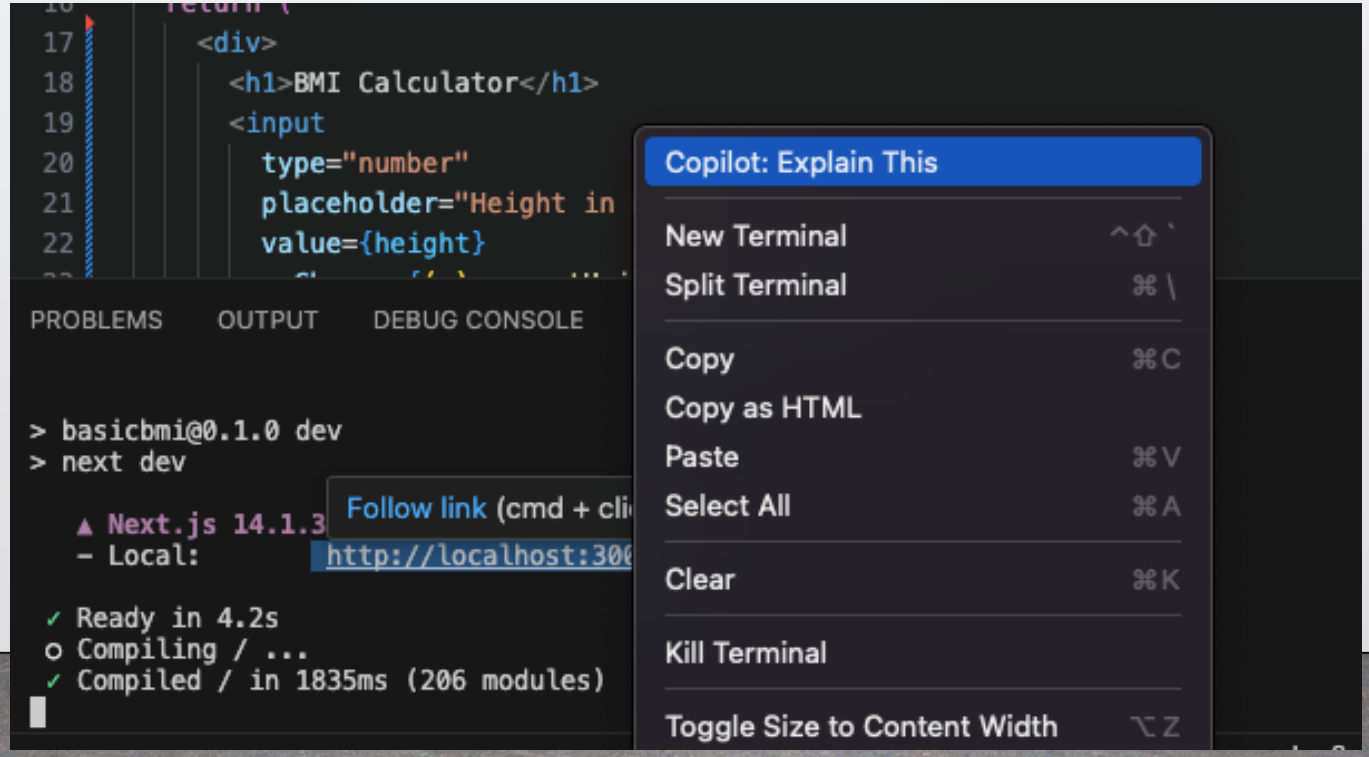
Generate documentation for your code

- /doc
- quickly generate documentation following language specific formats—Docstring for Python, JSDoc for Javascript or Javadoc for Java.



Get help with error messages in your terminal

- highlight the error message, right click, and select “Explain with Copilot.”



Prompt engineering foundations and best practices

- What is prompt engineering?
- Foundations of prompt engineering
- Best practices in prompt engineering
- How Copilot learns from your prompts

What is prompt engineering?

- Prompt engineering is the process of crafting clear instructions to guide AI systems, like GitHub Copilot, to generate context-appropriate code tailored to your project's specific needs.
- This ensures the code is syntactically, functionally, and contextually correct. Think of it like giving precise directions to a driver.
- Without them, the journey might be inefficient. But with clear guidance, the route becomes direct and efficient, saving time and energy.

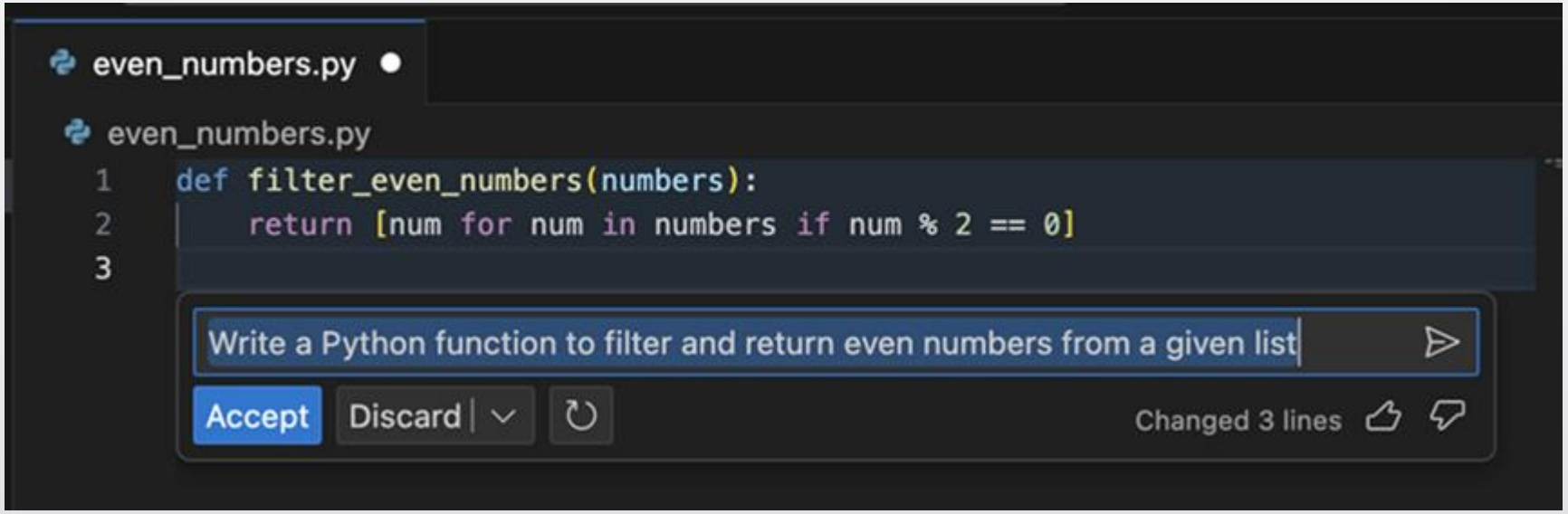
Principles of prompt engineering (4 S's)

- Single: Always focus your prompt on a single, well-defined task or question. This clarity is crucial for eliciting accurate and useful responses from Copilot.
- Specific: Ensure that your instructions are explicit and detailed. Specificity leads to more applicable and precise code suggestions.
- Short: While being specific, keep prompts concise and to the point. This balance ensures clarity without overloading Copilot or complicating the interaction.
- Surround: Utilize descriptive filenames and keep related files open. This provides Copilot with rich context, leading to more tailored code suggestions.

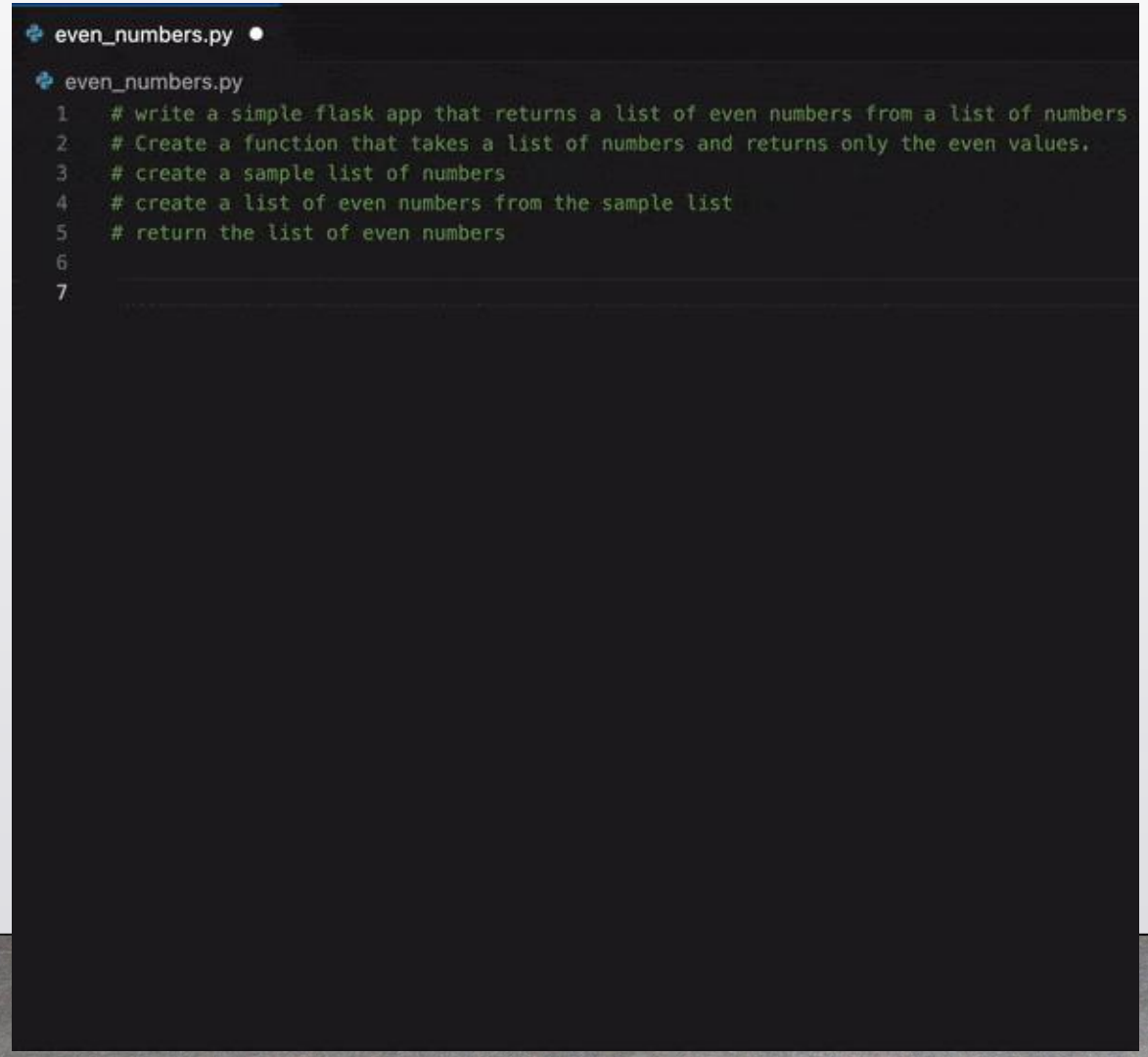
Best practices in prompt engineering

- Provide enough clarity
- Provide enough context with details
- Provide examples for learning

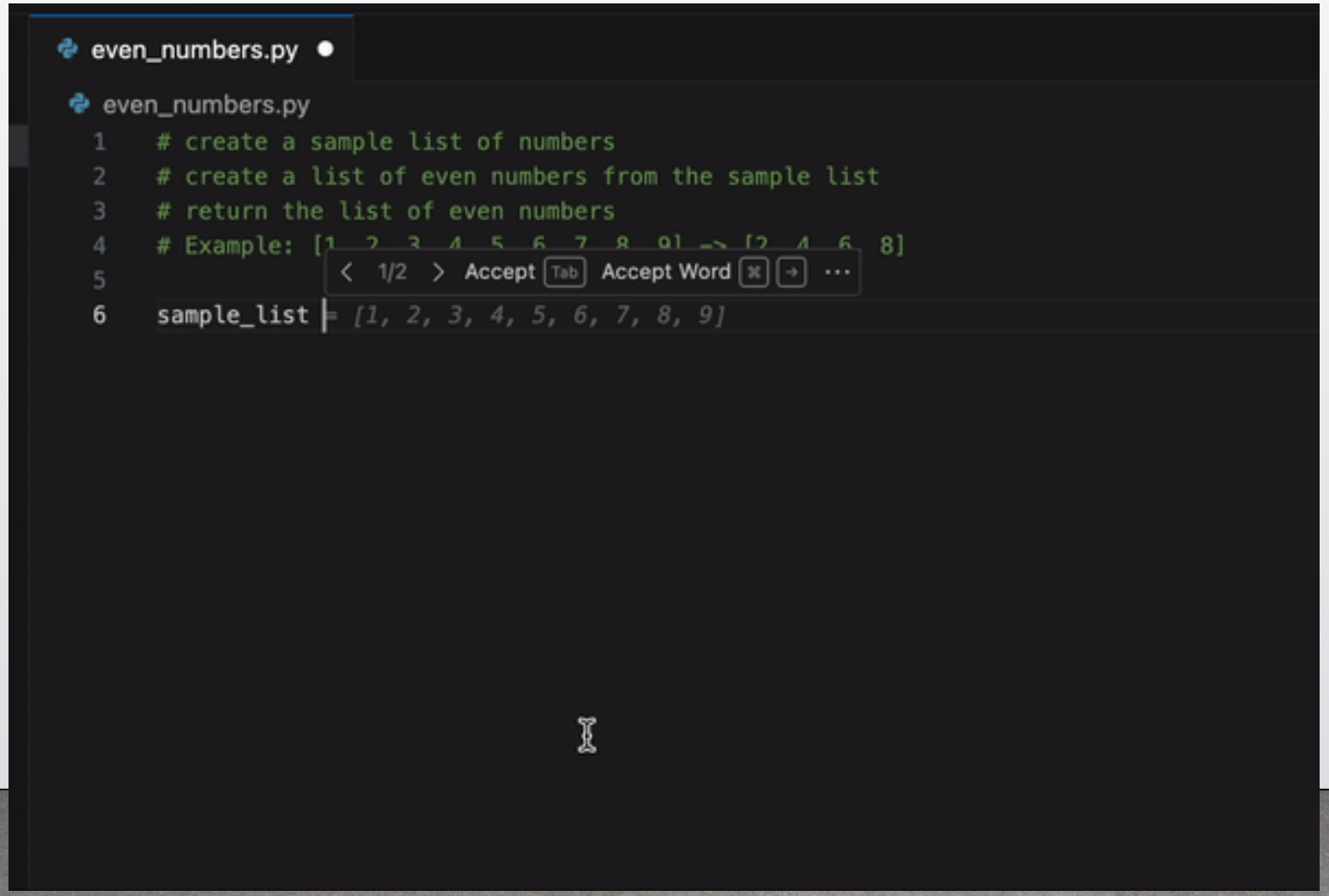
Provide enough clarity



Provide enough context with details



Provide examples for learning



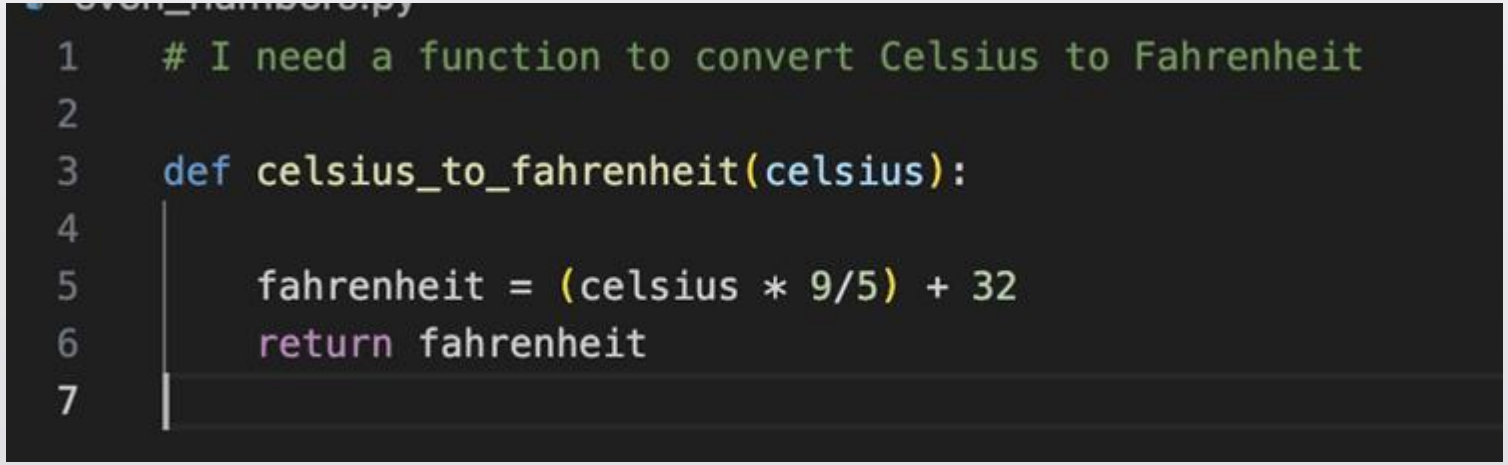
How Copilot learns from your prompts

GitHub Copilot operates based on AI models trained on vast amounts of data. To enhance its understanding of specific code contexts, engineers often provide it with examples. This practice, commonly found in machine learning, led to different training approaches such as:

- Zero-shot learning
- One-shot learning
- Few-shot learning

Zero-shot learning

- GitHub Copilot generates code without any specific examples, relying solely on its foundational training



One-shot learning

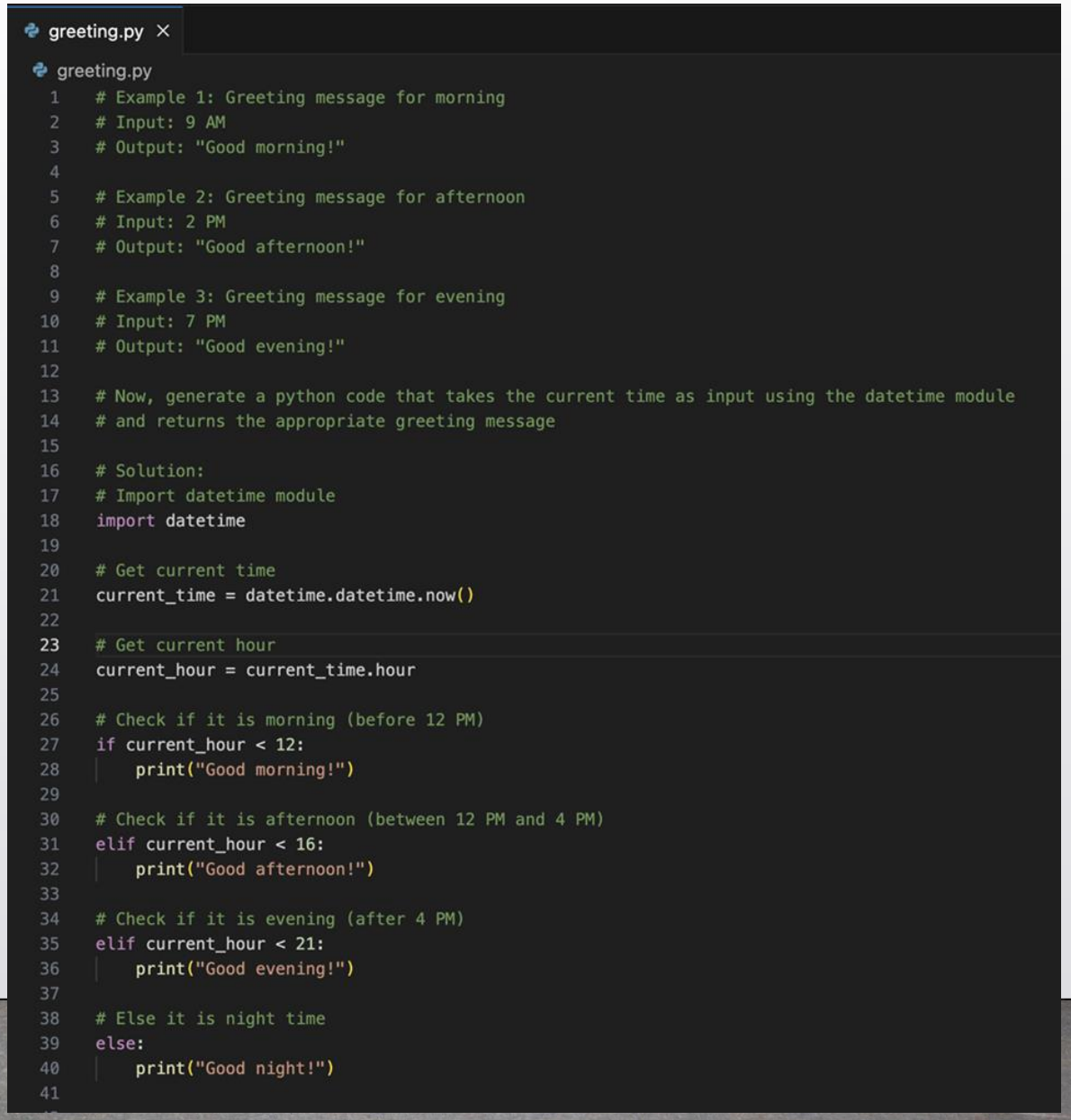
- Building upon the previous zero-shot example

```
# Example: Function to convert Fahrenheit to Celsius
def fahrenheit_to_celsius(fahrenheit):
    celsius = (fahrenheit - 32) * 5/9
    return celsius

# Now I need a function to convert Celsius to Fahrenheit

def celsius_to_fahrenheit(celsius):
    fahrenheit = celsius * 9/5 + 32
    return fahrenheit
```

Few-shot learning

- In this method, Copilot is presented with several examples, which strike a balance between zero-shot unpredictability and the precision of fine-tuning.

Few-shot learning



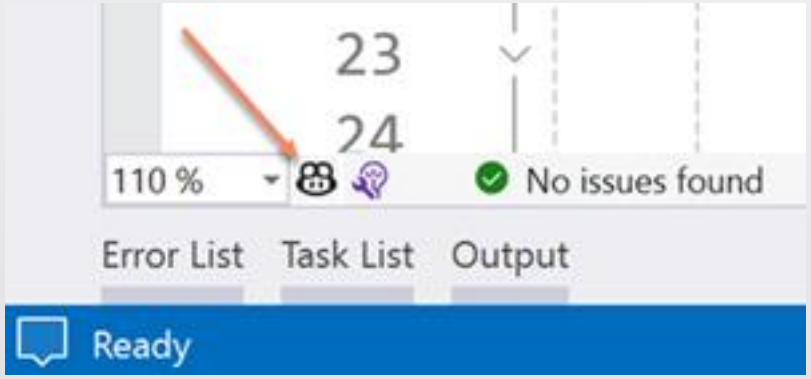
Validating the output of GitHub Copilot

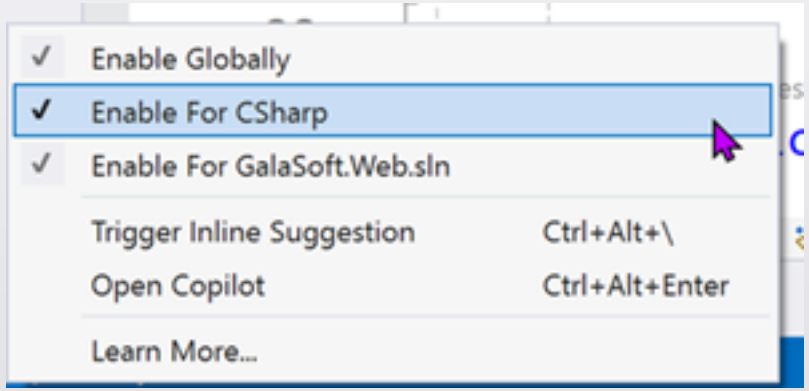
- The key principle to remember when working with AI assistants is that you should always verify the output. You have the chance to review the Copilot output either before accepting, when the code appears greyed out inline or after you have accepted the suggestion with the Tab key.

GitHub Copilot is not a compiler!

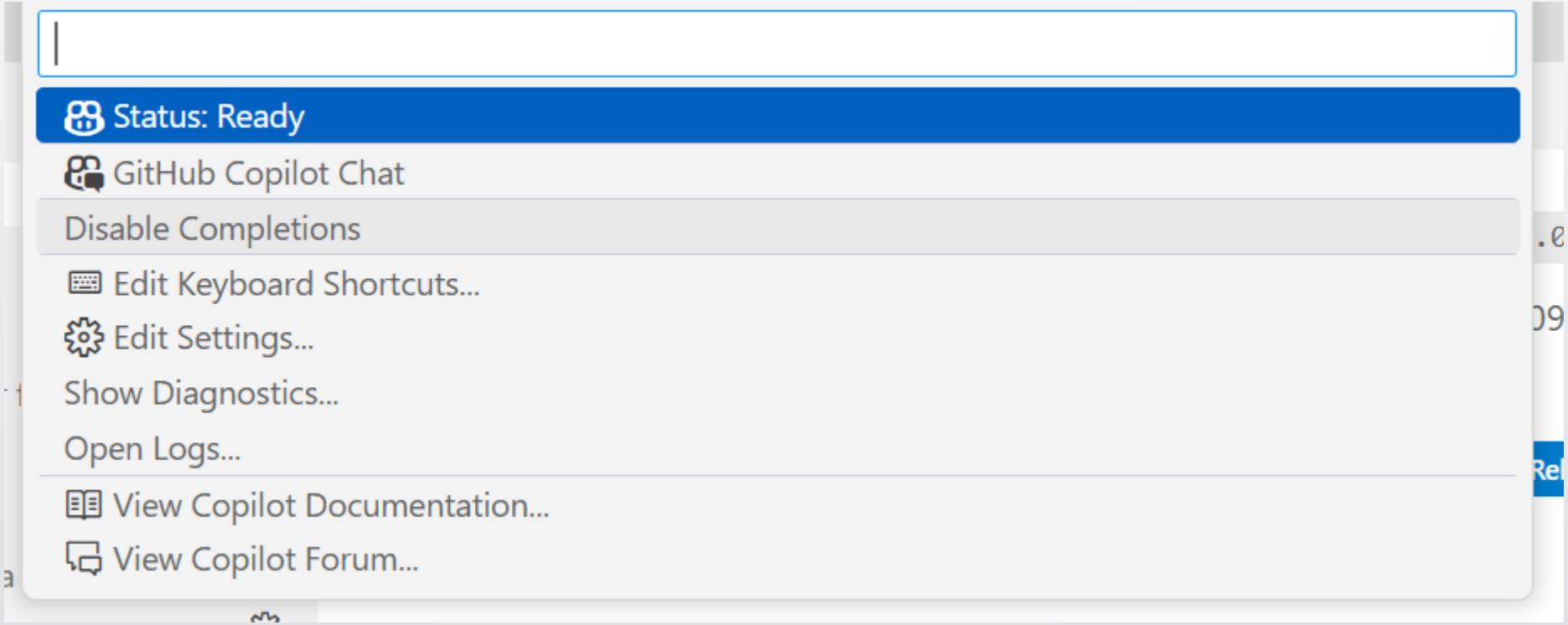
- This means that the code produced by GitHub Copilot can indeed fail to compile and/or be incorrect. It is your responsibility to review, resolve, and improve the suggested code, as necessary.

Disabling GitHub Copilot completions (VS Studio)



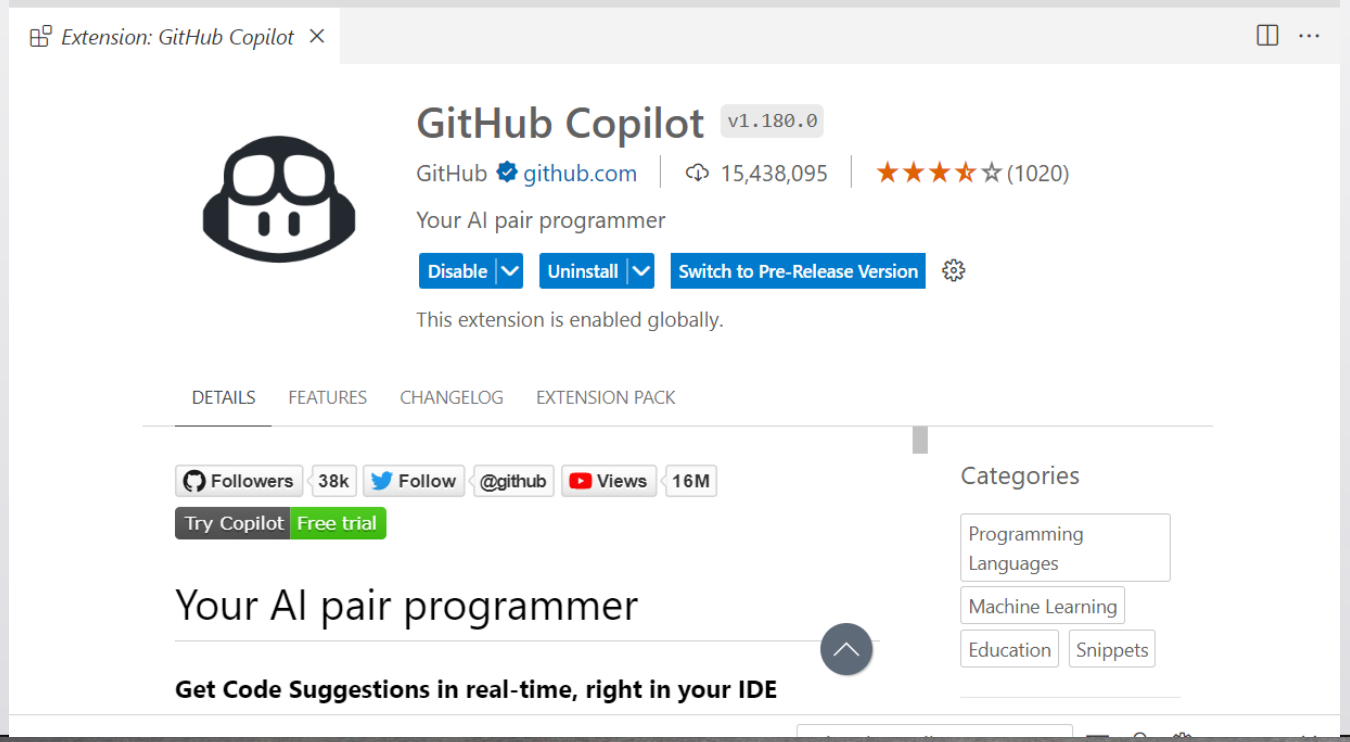


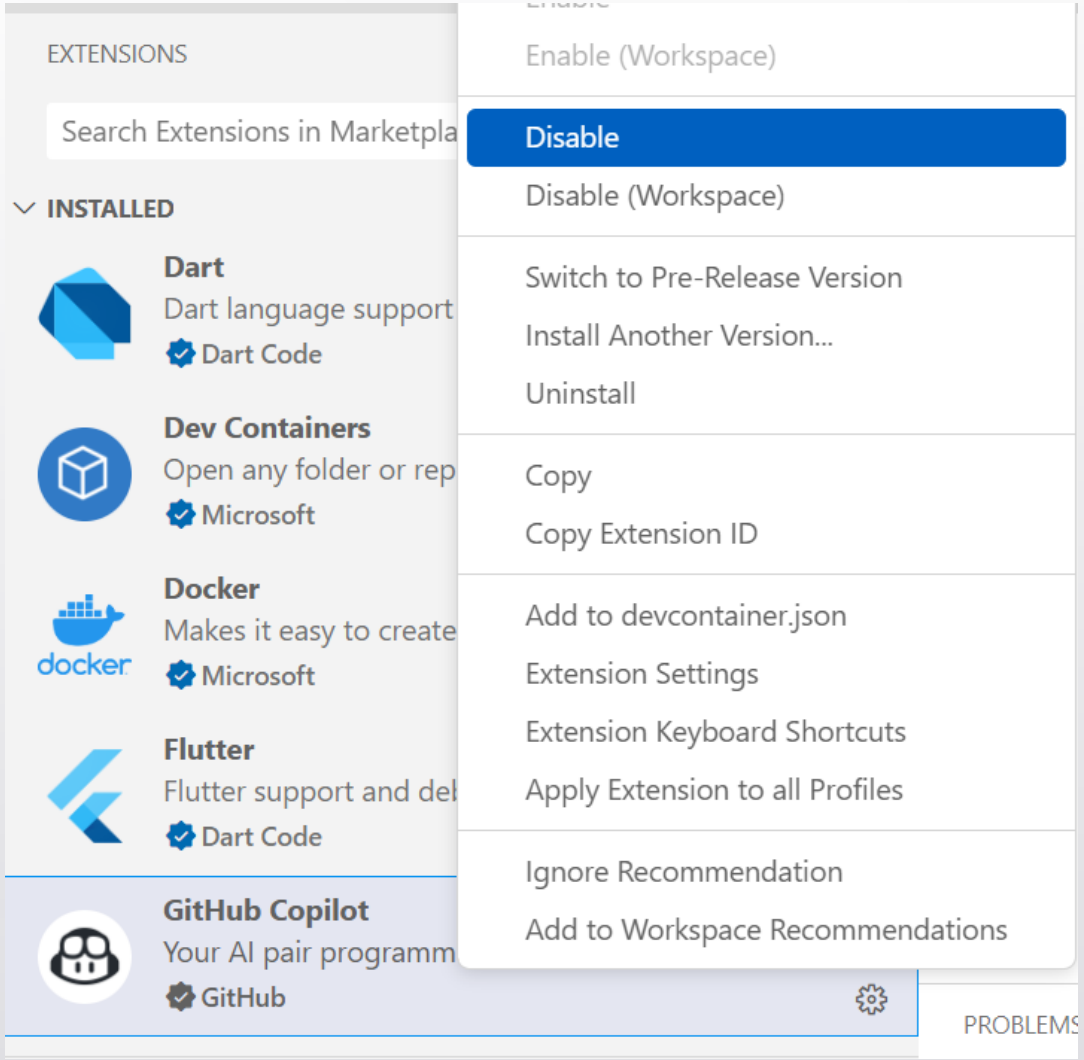
Disabling GitHub Copilot completions (VS Code)



Disabling GitHub Copilot

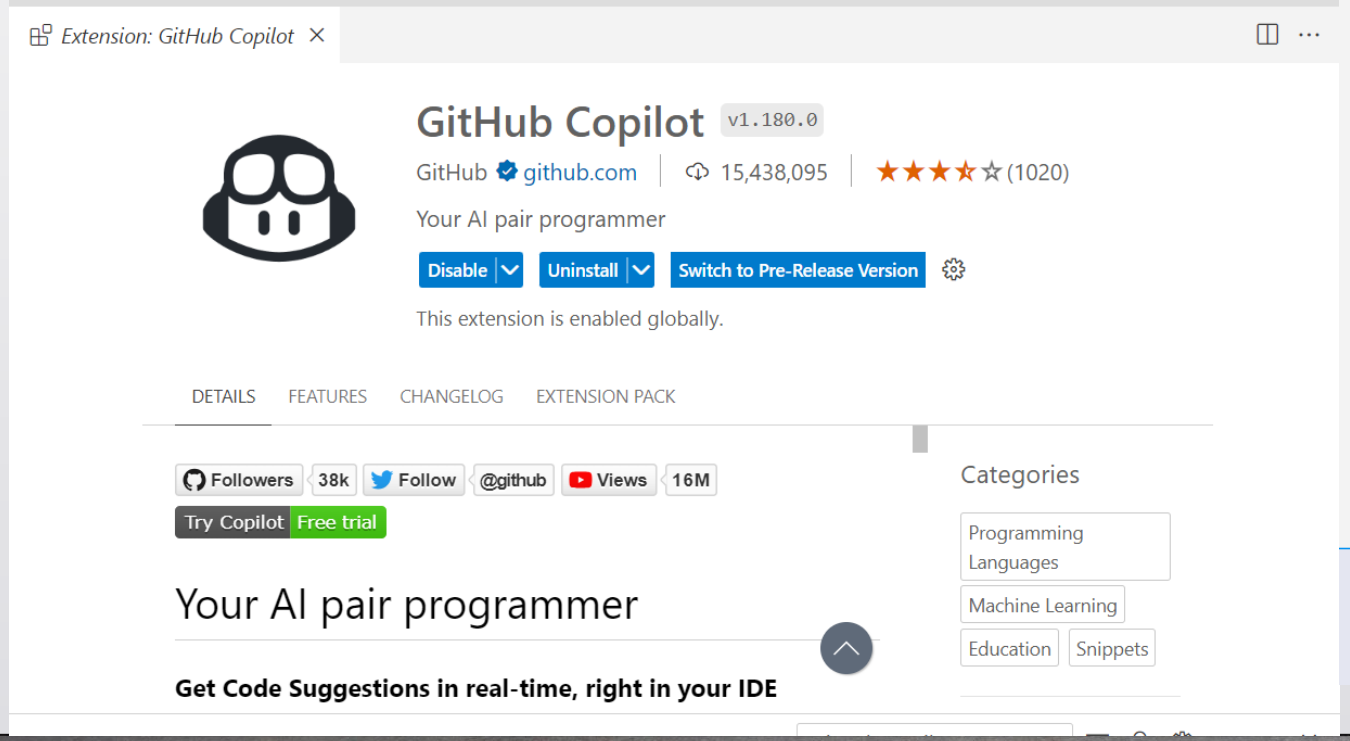
- ctrl-shift-x : right click copilot - disable

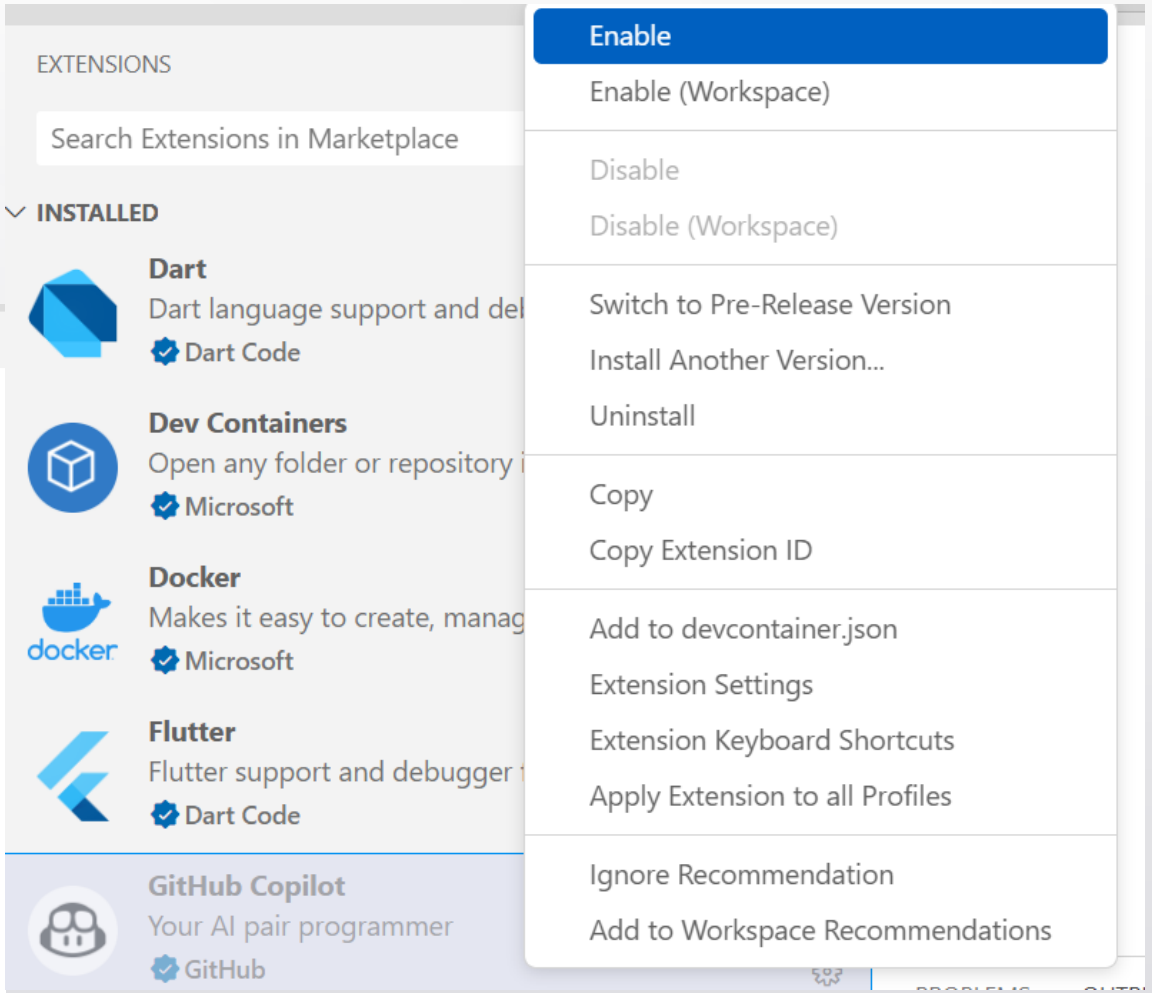




Enabling GitHub Copilot

- ctrl-shift-x : right click copilot - Enable



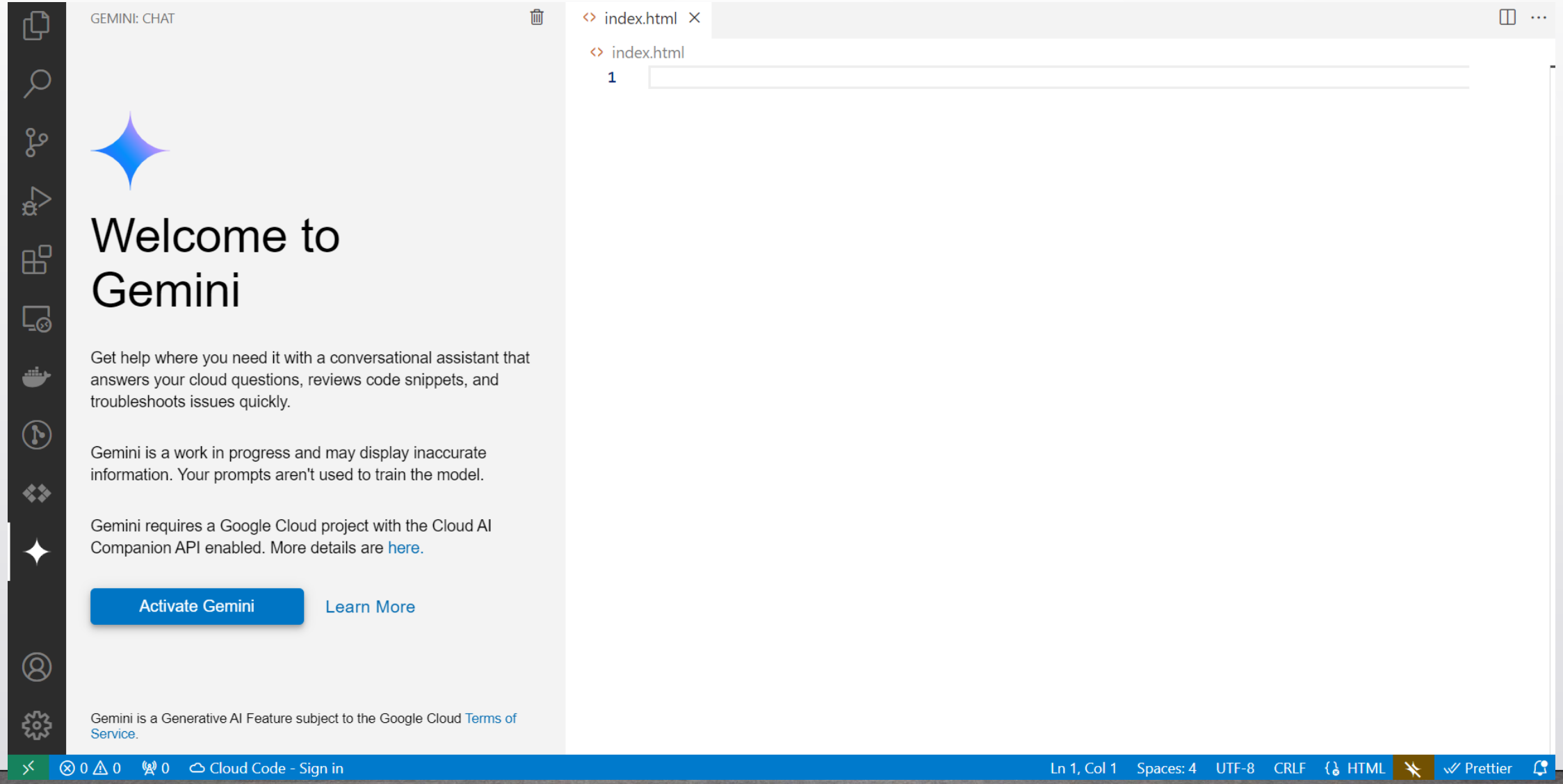


Build auto suggest engine with Copilot

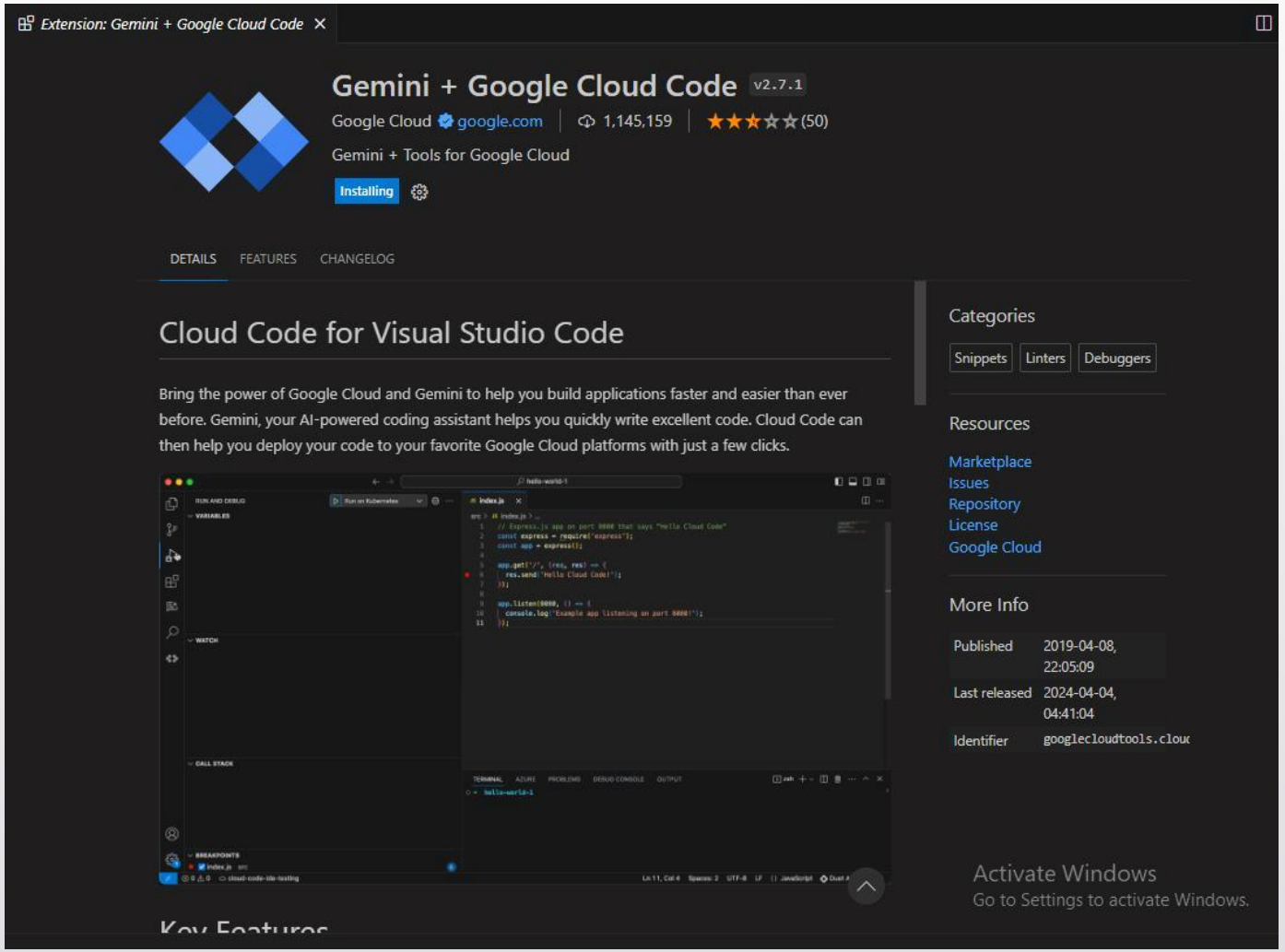
- Exercise - Use Copilot to interpret code
- Exercise - Use Copilot to autocomplete code
- Exercise - Prompt Copilot to generate code
- Exercise - Debug code with Copilot Chat
- Exercise - Test code with Copilot
- https://learn.microsoft.com/en-us/training/modules/build-auto-suggest-engine-copilot
- https://github.com/MicrosoftLearning/Guided-project-Build-an-Autosuggest-Engine-with-Copilot.git

Gemini Code Assist

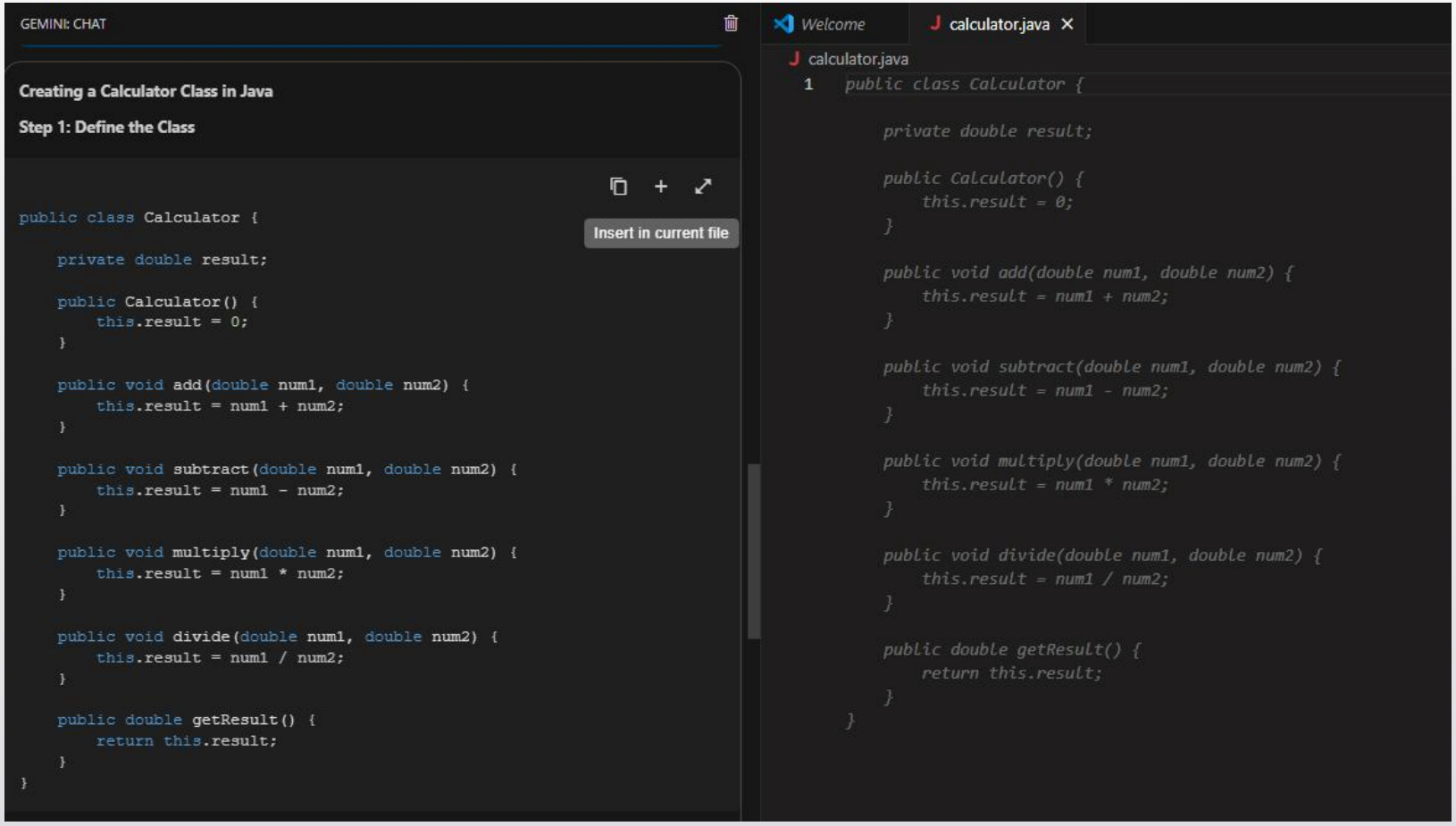
- https://cloud.google.com/gemini/docs/codeassist/use-in-ide
- https://www.youtube.com/watch?v=WsXVGr0Q3C4


GEMINI: CHAT
index.html
index.html
1
Welcome to Gemini
Get help where you need it with a conversational assistant that answers your cloud questions, reviews code snippets, and troubleshoots issues quickly.
Gemini is a work in progress and may display inaccurate information. Your prompts aren't used to train the model.
Gemini requires a Google Cloud project with the Cloud AI Companion API enabled. More details are here.
Activate Gemini
Learn More
Gemini is a Generative AI Feature subject to the Google Cloud Terms of Service.
0 0 0
Cloud Code - Sign in
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
HTML
Prettier

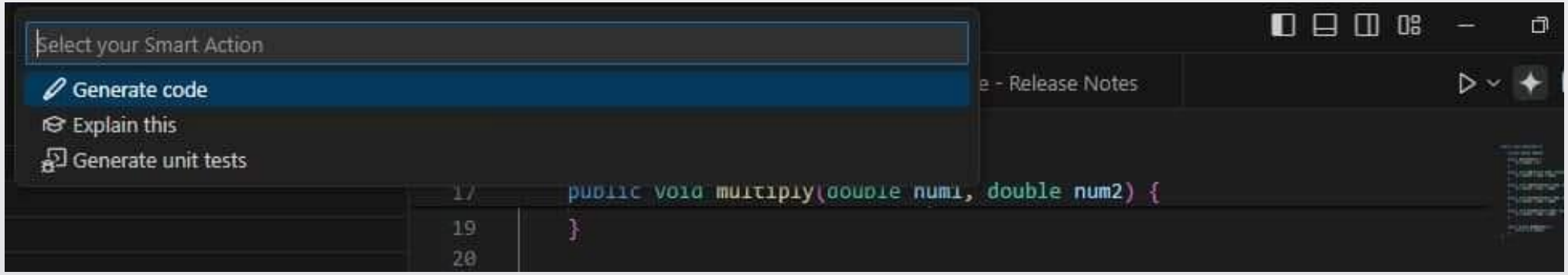
Extension


Extension: Gemini + Google Cloud Code
Gemini + Google Cloud Code
v2.7.1
Google Cloud google.com
1,145,159
(50)
Gemini + Tools for Google Cloud
Installing
DETAILS
FEATURES
CHANGELOG
Cloud Code for Visual Studio Code
Bring the power of Google Cloud and Gemini to help you build applications faster and easier than ever before. Gemini, your AI-powered coding assistant helps you quickly write excellent code. Cloud Code can then help you deploy your code to your favorite Google Cloud platforms with just a few clicks.
Categories
Snippets
Linters
Debuggers
Resources
Marketplace
Issues
Repository
License
Google Cloud
More Info
Published 2019-04-08, 22:05:09
Last released 2024-04-04, 04:41:04
Identifier
Activate Windows
Go to Settings to activate Windows.

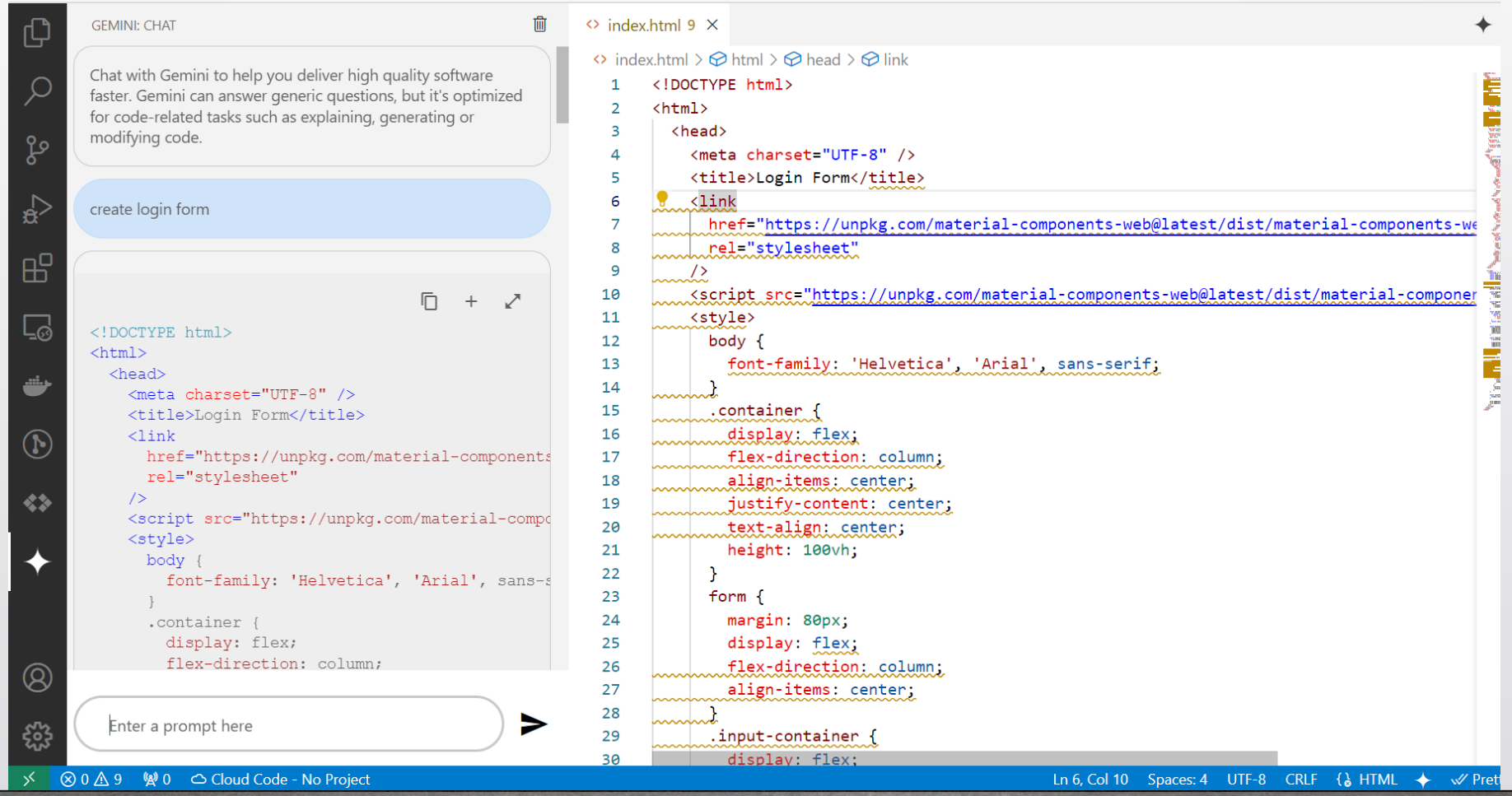
Generate Code (Ghost Text)



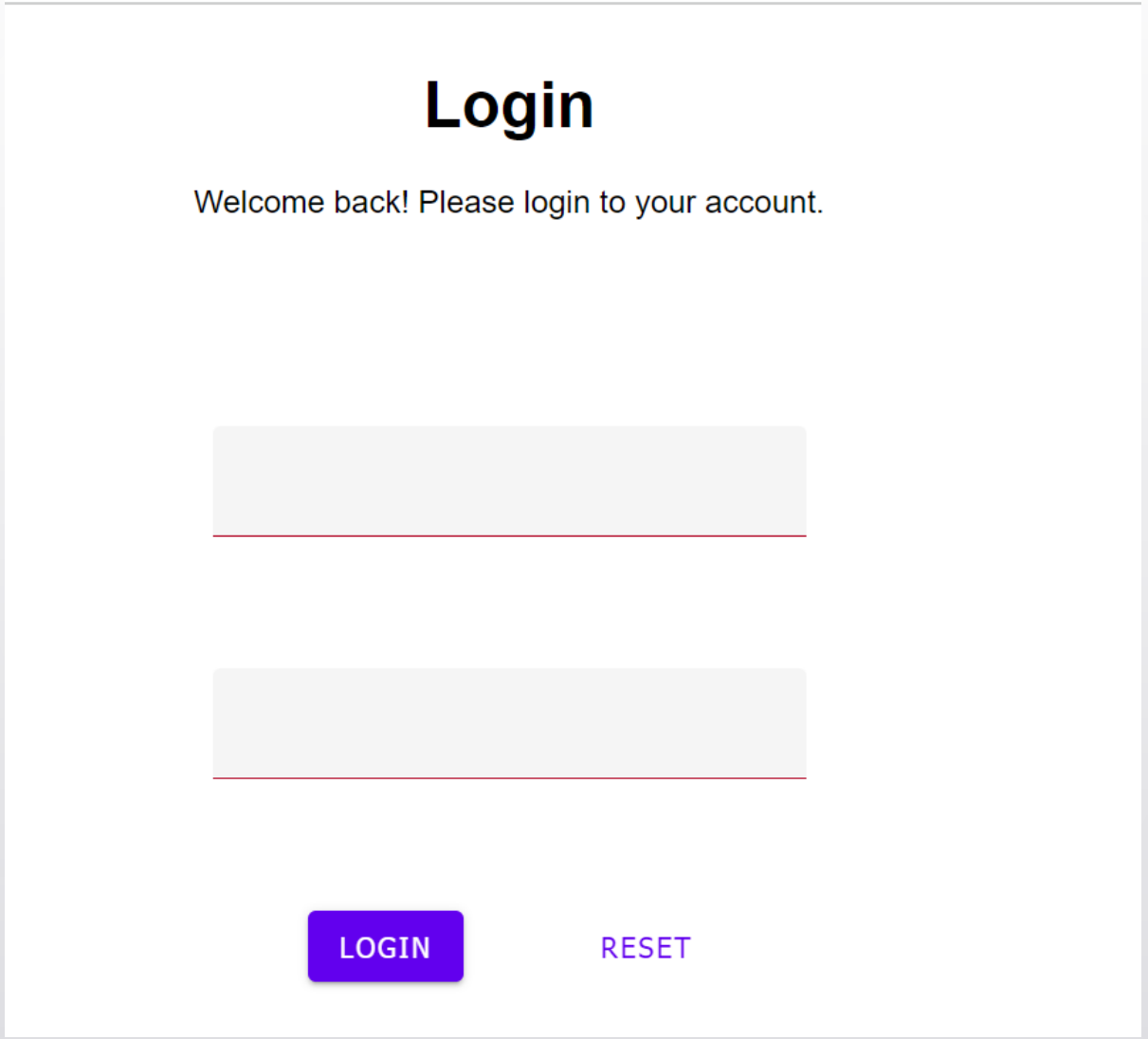
Smart Action



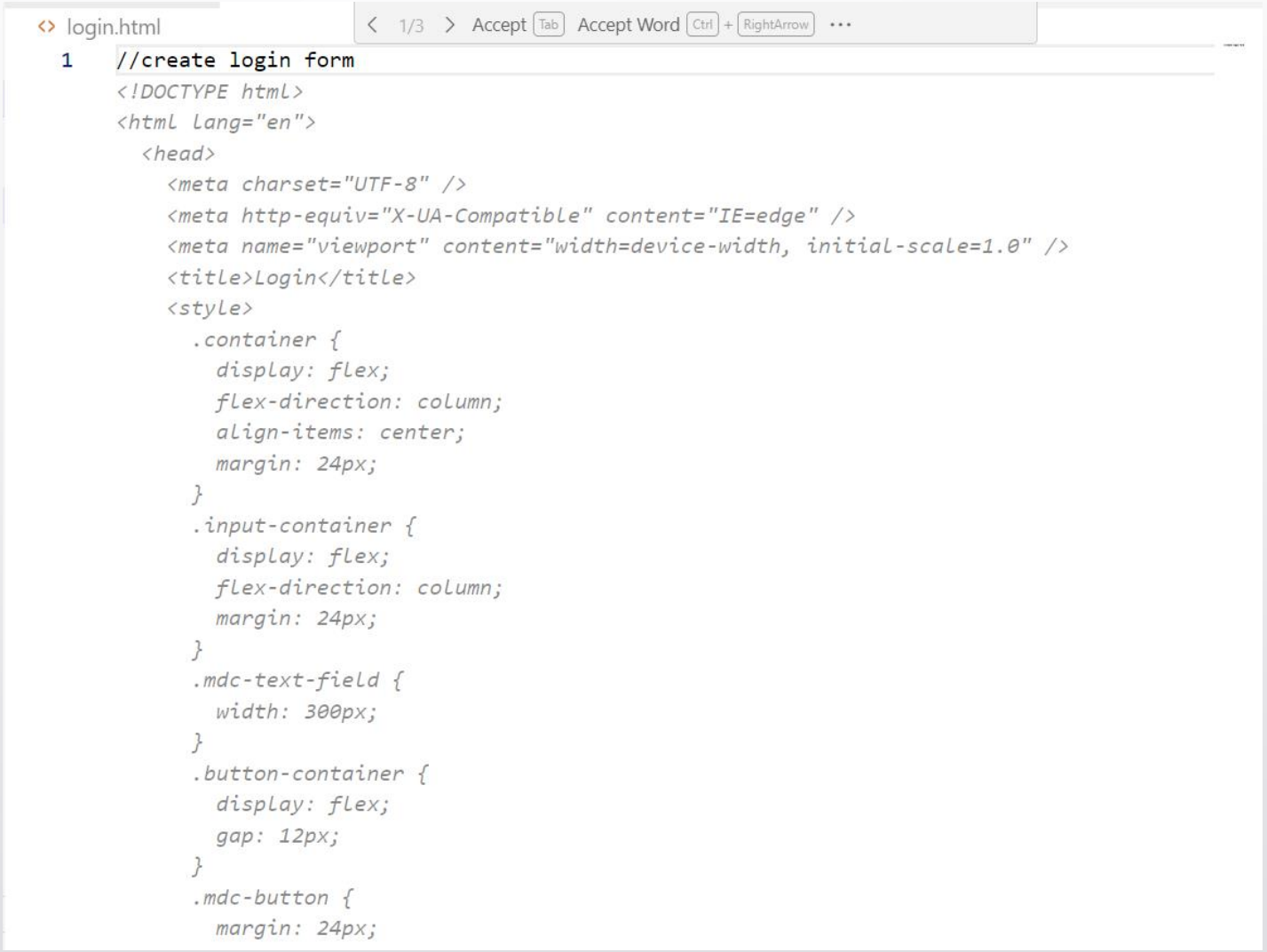
GEMINI CHAT


GEMINI: CHAT
Chat with Gemini to help you deliver high quality software faster. Gemini can answer generic questions, but it's optimized for code-related tasks such as explaining, generating or modifying code.
create login form
Enter a prompt here
index.html 9
index.html > html > head > link
Cloud Code - No Project
Ln 6, Col 10
Spaces: 4
UTF-8
CRLF
HTML

Result


Login
Welcome back! Please login to your account.
LOGIN
RESET

Ctrl+Enter


login.html
1/3
Accept Tab
Accept Word Ctrl + RightArrow
//create login form
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Login</title>
<style>
.container {
display: flex;
flex-direction: column;
align-items: center;
margin: 24px;
}
.input-container {
display: flex;
flex-direction: column;
margin: 24px;
}
.mdc-text-field {
width: 300px;
}
.button-container {
display: flex;
gap: 12px;
}
.mdc-button {
margin: 24px;

DayDiff

```php
//create daydiff function
function daydiff($date1, $date2) {
    //convert dates to timestamps
    $timestamp1 = strtotime($date1);
    $timestamp2 = strtotime($date2);
    //calculate the difference in days
    $days = ($timestamp2 - $timestamp1) / (60 * 60 * 24);
    //return the difference in days
    return $days;
}
//call daydiff function
$days = daydiff("2023-03-08", "2023-03-15");
echo "The difference in days between 2023-03-08 and 2023-03-15 is: $days";
```

How to use a Comment to Prompt GitHub Copilot

- // function to get the year of birth
- // function to get the year of birth *from the age*

PHP

- DayDiff
- Login Form
- Calculator Class
- PHPUnit Test
  - https://medium.com/@naruapon/การก้าวสู่นักพัฒนามืออาชีพ-กับการทดสอบแบบหน่วยย่อย-unittest-ด้วย-php-unit-9-0-1-ในเชิงปฏิบัติ-44692908b764

Refference

- https://github.com/features/copilot/getting-started
- https://nira.com/github-copilot/
- https://devblogs.microsoft.com/visualstudio/how-to-install-github-copilot-in-visual-studio/
- https://github.blog/2024-01-22-10-unexpected-ways-to-use-github-copilot
- https://learn.microsoft.com/en-us/training/modules/introduction-prompt-engineering-with-github-copilot/
- https://devblogs.microsoft.com/visualstudio/how-to-use-comments-to-prompt-github-copilot-visual-studio/

Refference

- https://devblogs.microsoft.com/visualstudio/github-copilot-in-visual-studio-a-recap-of-2023/
- https://learn.microsoft.com/en-us/training/modules/build-auto-suggest-engine-copilot
- Gemini for Application Developers
- https://www.youtube.com/watch?v=WsXVGr0Q3C4
- https://cloud.google.com/gemini/docs/codeassist/use-in-ide