GitHub Expert Services

GitHub Copilot Tips, Tricks and Best practices

Agenda

- Beyond code completion
- Context, Context, Context
- Inline Chat
- Copilot CLI
- Workspace Agent
- Threads
- Slash Commands
- Q&A

GitHub Copilot

Beyond code completion

It’s no longer just a code completion tool in your editor—it now includes a chat interface that you can use in your IDE, a command line tool via a GitHub CLI extension, a summary tool in your pull requests, a helper tool in your terminals, and much, much more.

- Copilot

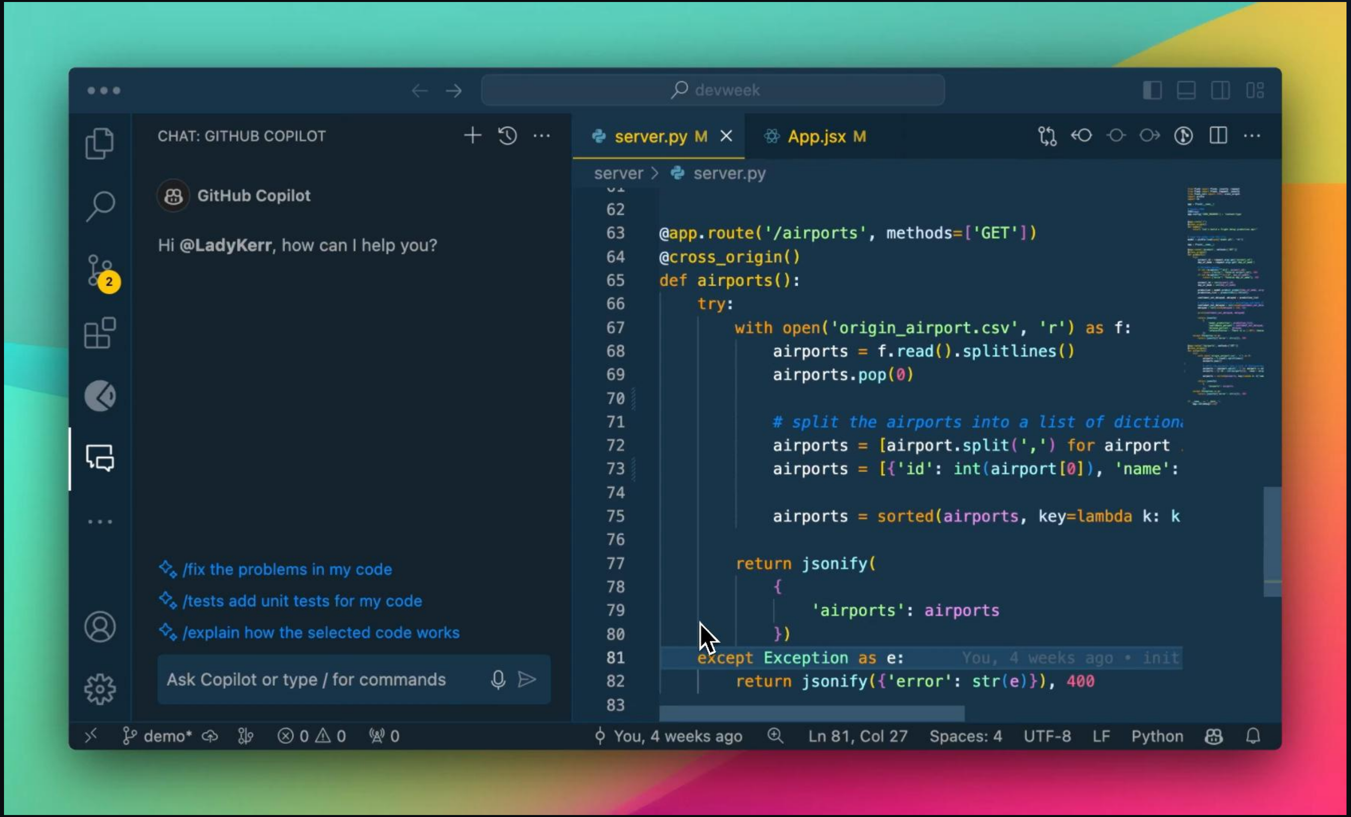
Context, Context, Context

- LLMs are designed to make prediction based on the context provided
- Provide ample context is the key
- Copilot is able to infer from the code in IDE editor
- How about adding more context?

Context, Context, Context

- Open Relevant files
- Provide Top level comment
- Set Includes and references

Context, Context, Context



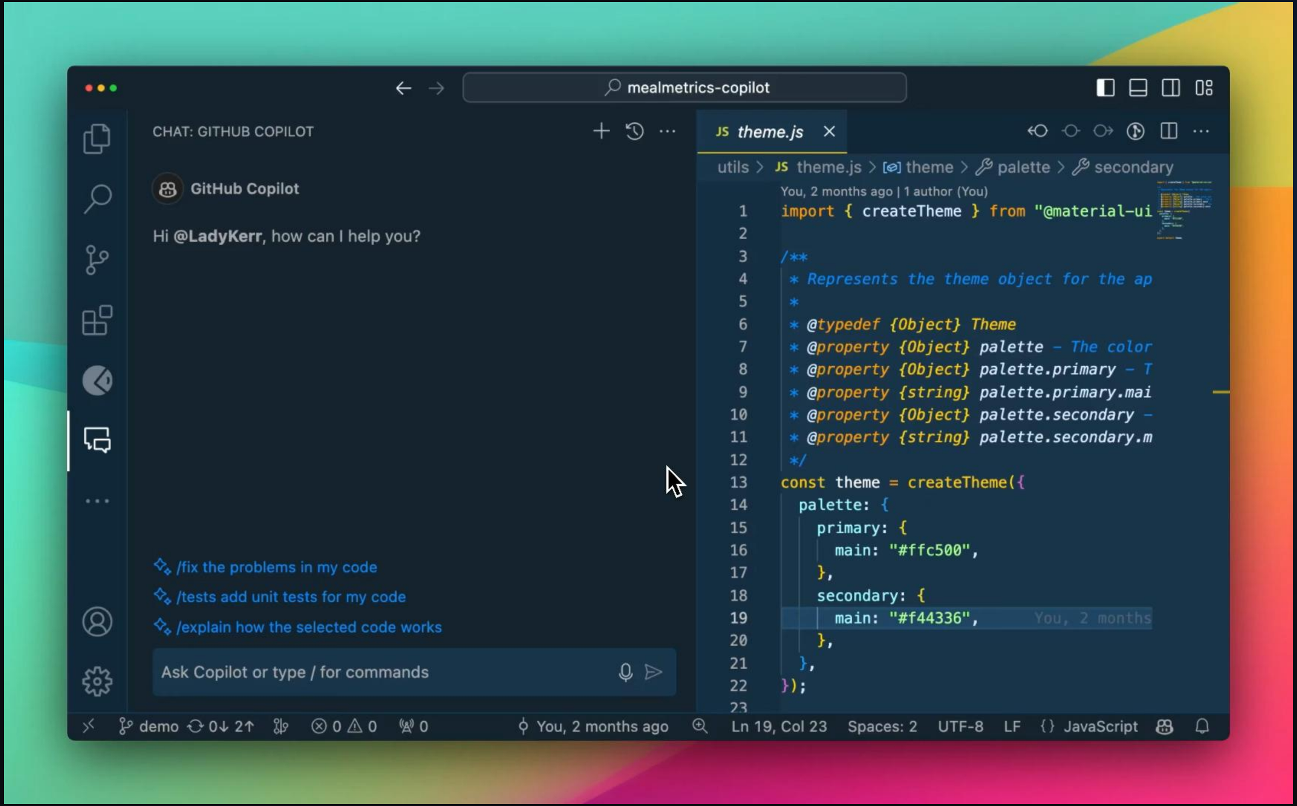
Hi @LadyKerr, how can I help you?

/fix the problems in my code
/tests add unit tests for my code
/explain how the selected code works

Ask Copilot or type / for commands

```
@app.route('/airports', methods=['GET'])
@cross_origin()
def airports():
    try:
        with open('origin_airport.csv', 'r') as f:
            airports = f.read().splitlines()
            airports.pop(0)

            # split the airports into a list of diction
            airports = [airport.split(',') for airport
            airports = [{'id': int(airport[0]), 'name':

            airports = sorted(airports, key=lambda k: k

        return jsonify(
            {
                'airports': airports
            })
    except Exception as e:      You, 4 weeks ago • init
        return jsonify({'error': str(e)}), 400
```

Context, Context, Context



GitHub Copilot

Hi @LadyKerr, how can I help you?

/fix the problems in my code

/tests add unit tests for my code

/explain how the selected code works

Ask Copilot or type / for commands

You, 2 months ago | 1 author (You)

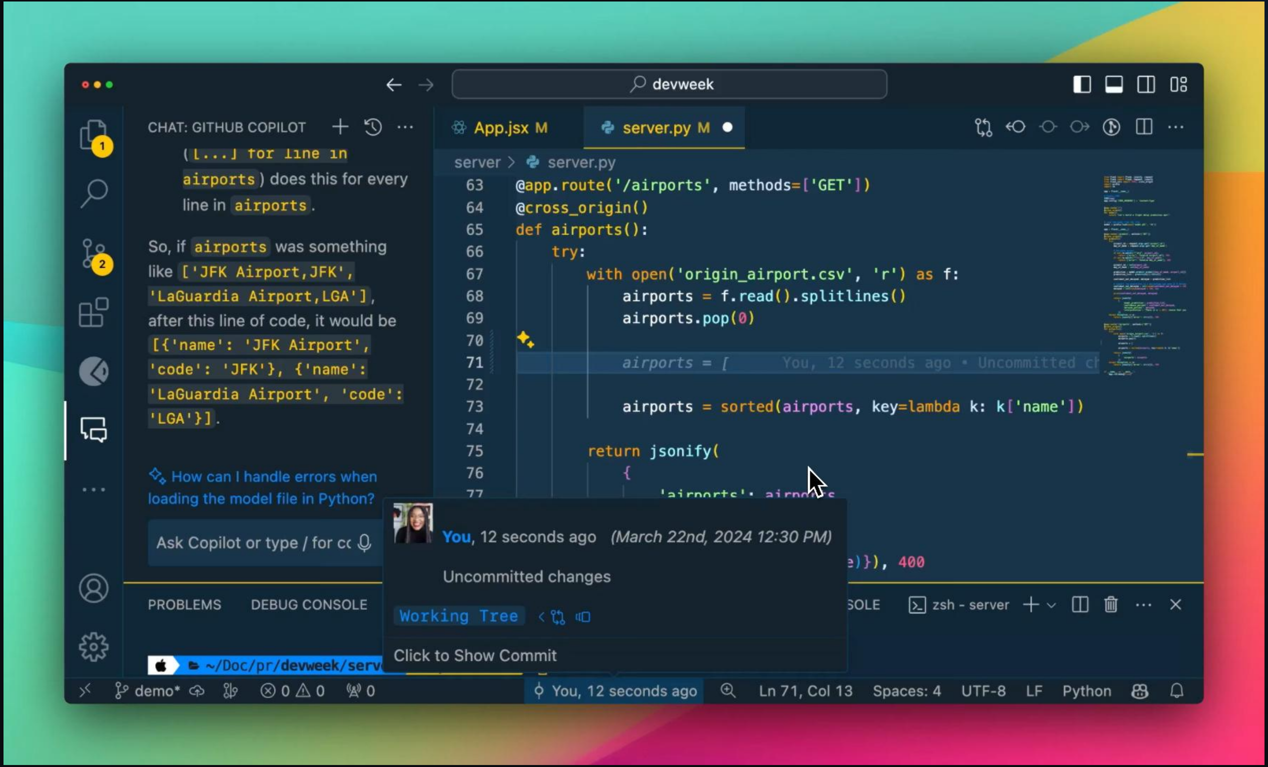
```
import { createTheme } from "@material-ui

/**
 * Represents the theme object for the ap
 *
 * @typedef {Object} Theme
 * @property {Object} palette - The color
 * @property {Object} palette.primary - T
 * @property {string} palette.primary.mai
 * @property {Object} palette.secondary -
 * @property {string} palette.secondary.m
 */
const theme = createTheme({
  palette: {
    primary: {
      main: "#ffc500",
    },
    secondary: {
      main: "#f44336",      You, 2 months
    },
  },
});
```

demo 0↓ 2↑ ⊗ 0 ⚠ 0 0 You, 2 months ago Ln 19, Col 23 Spaces: 2 UTF-8 LF {} JavaScript

Context, Context, Context

- Meaningful names
- Specific function comments
- Provide sample code
- Let’s give them a try

Context, Context, Context



([...] for line in airports) does this for every line in airports.

So, if airports was something like ['JFK Airport,JFK', 'LaGuardia Airport,LGA'], after this line of code, it would be [{'name': 'JFK Airport', 'code': 'JFK'}, {'name': 'LaGuardia Airport', 'code': 'LGA'}].

How can I handle errors when loading the model file in Python?

Ask Copilot or type / for cc

```
@app.route('/airports', methods=['GET'])
@cross_origin()
def airports():
    try:
        with open('origin_airport.csv', 'r') as f:
            airports = f.read().splitlines()
            airports.pop(0)

            airports = [      You, 12 seconds ago • Uncommitted ch

            airports = sorted(airports, key=lambda k: k['name'])

        return jsonify(
            {
                'airports': airpo
```

```
e)}), 400
```

You, 12 seconds ago *(March 22nd, 2024 12:30 PM)*

Uncommitted changes

Working Tree

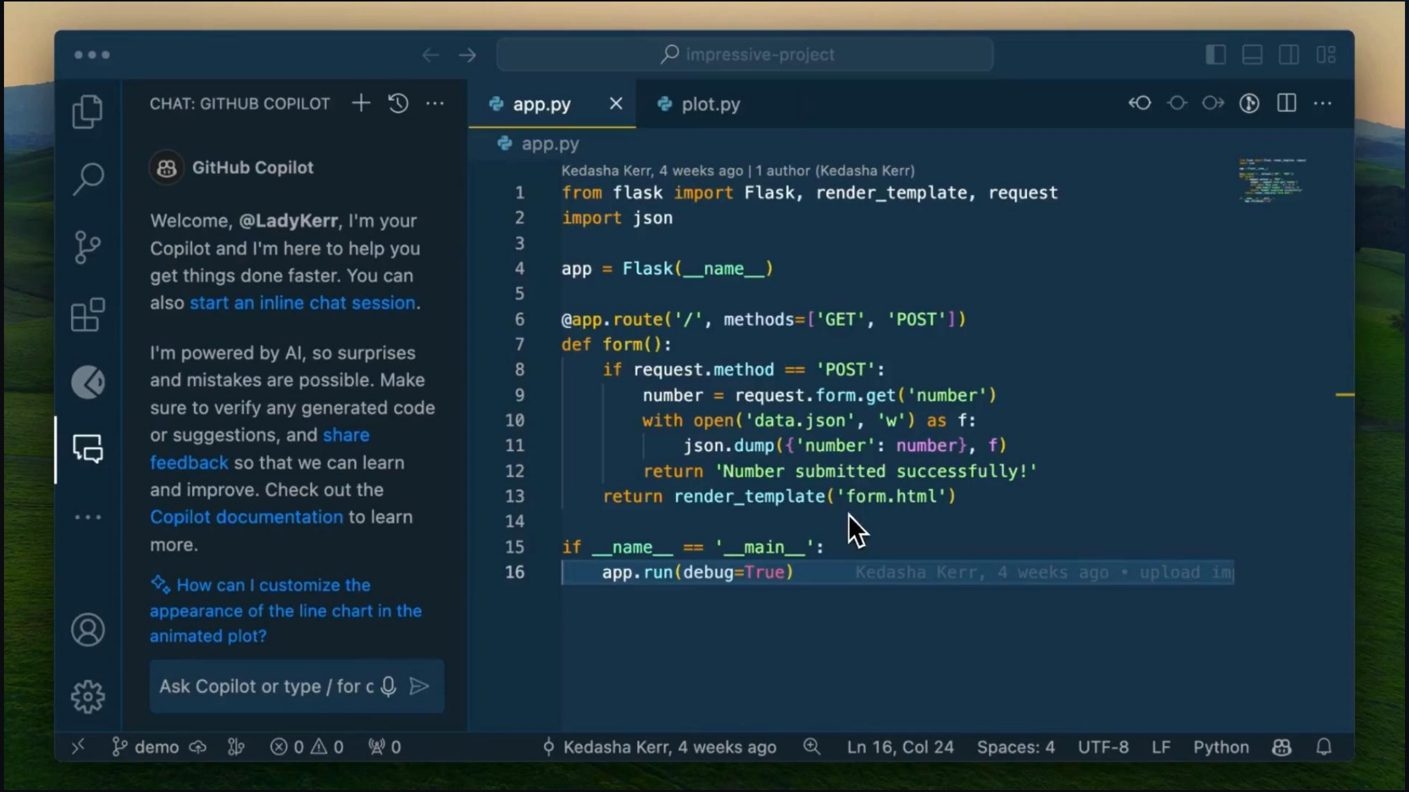
Click to Show Commit

You, 12 seconds ago

Inline Chat with GitHub Copilot

- CTRL + I on Windows
- Ask specific questions
- Inline code diffs
- Slash command support
- Let’s give it a go

Inline Chat with GitHub Copilot



Welcome, **@LadyKerr**, I'm your Copilot and I'm here to help you get things done faster. You can also start an inline chat session.

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve. Check out the Copilot documentation to learn more.

How can I customize the appearance of the line chart in the animated plot?

Ask Copilot or type / for c

Kedasha Kerr, 4 weeks ago | 1 author (Kedasha Kerr)

```python
from flask import Flask, render_template, request
import json

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def form():
    if request.method == 'POST':
        number = request.form.get('number')
        with open('data.json', 'w') as f:
            json.dump({'number': number}, f)
        return 'Number submitted successfully!'
    return render_template('form.html')

if __name__ == '__main__':
    app.run(debug=True)     Kedasha Kerr, 4 weeks ago • upload im
```

GitHub Copilot CLI

- **You can use Copilot with the GitHub CLI to get suggestions and explanations for the command line.**

- **To ask Copilot in the CLI to explain a command, run** gh copilot explain **followed by the command that you want explained.**

- **To ask Copilot in the CLI to suggest a command, run** gh copilot suggest **followed by the command that you want**

Copilot Chat

GitHub Copilot Chat provides an experience in your editor where you can have a conversation with the AI assistant. You can improve this experience by using built-in features to make the most out of it.

- Copilot

Copilot Chat

- Remove irrelevant requests
- Navigate through your conversation
- Use the @workspace agent
- Highlight relevant code
- Organize your conversations with threads

Copilot Edits

Copilot Edits start an AI-powered code editing session and iterate quickly on code changes across multiple files by using natural language. Copilot Edits applies the edits directly in the editor, where you can review them in-place, with the full context of the surrounding code.

- Copilot

Copilot Edits

- Select which files to edit
- Provide the relevant context and prompt
- Review the suggested edits
- Accept or discard the suggested edits
- Iterate on the code changes

Copilot Edits (Agent Mode)

In agent mode, Copilot Edits operates in a more autonomous and dynamic manner to achieve the desired outcome. Copilot agent mode determines the relevant context, offers both code changes and terminal commands, and iterates to remediate issues.

- Copilot

How is Copilot Edits different from Copilot Chat?

- Copilot Edits puts you in the context of code editing. It can generate and apply code changes directly across multiple files
- The Chat view gives you a more general-purpose chat interface for asking questions about your code or technology topics in general.

Copilot Edits (Agent Mode)

- Copilot Edits Agent Mode uses a set of tools to accomplish the individual tasks to complete a request. E.g., listing the files in a directory, editing a file in your workspace, running a terminal command, getting the output from the terminal, and more.

- Still in Preview

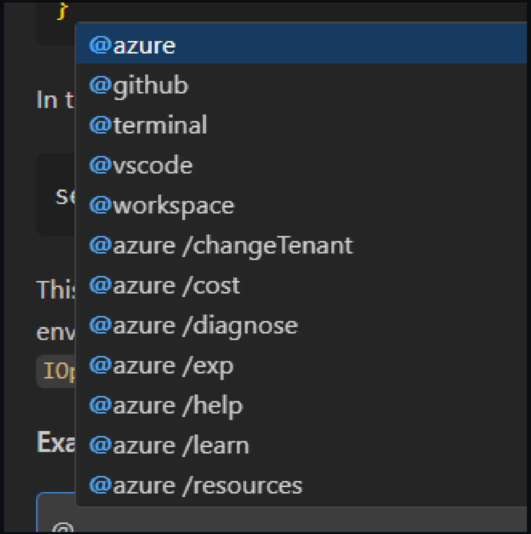
Slash Commands

Command	Description	Usage
/explain	Get code explanations	Open file with code or highlight code you want explained and type: /explain what is the fetchPrediction method?
/fix	Receive a proposed fix for the problems in the selected code	Highlight problematic code and type: /fix propose a fix for the problems in fetchAirports route
/tests	Generate unit tests for selected code	Open file with code or highlight code you want tests for and type: /tests
/help	Get help on using Copilot Chat	Type: /help what can you do?
/clear	Clear current conversation	Type: /clear
/doc	Add a documentation comment	Highlight code and type: /doc You can also press CMD+I in your editor and type /doc/ inline

Slash Commands

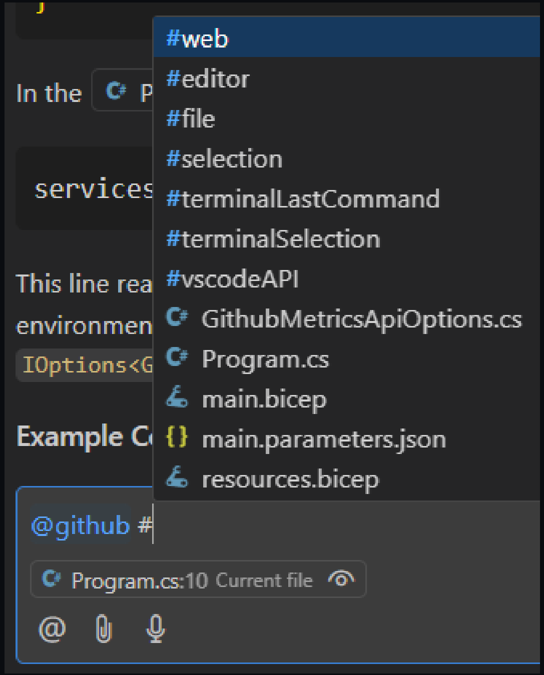
/generate	Generate code to answer your question	Type: /generate code that validates a phone number
/optimize	Analyze and improve running time of the selected code	Highlight code and type: /optimize fetchPrediction method
/clear	Clear current chat	Type: /clear
/new	Scaffold code for a new workspace	Type: /new create a new django app
/simplify	Simplify the selected code	Highlight code and type: /simplify
/feedback	Provide feedback to the team	Type: /feedback

Workspace Agent



- Doesn’t response straight away
- Analyzes the current workspace files and directories
- Creates a context from everything it was able to collect.

Chat Variables


In the P

services

This line rea
environmen
IOptions<G

Example C

#web
#editor
#file
#selection
#terminalLastCommand
#terminalSelection
#vscodeAPI
GithubMetricsApiOptions.cs
Program.cs
main.bicep
main.parameters.json
resources.bicep

@github #

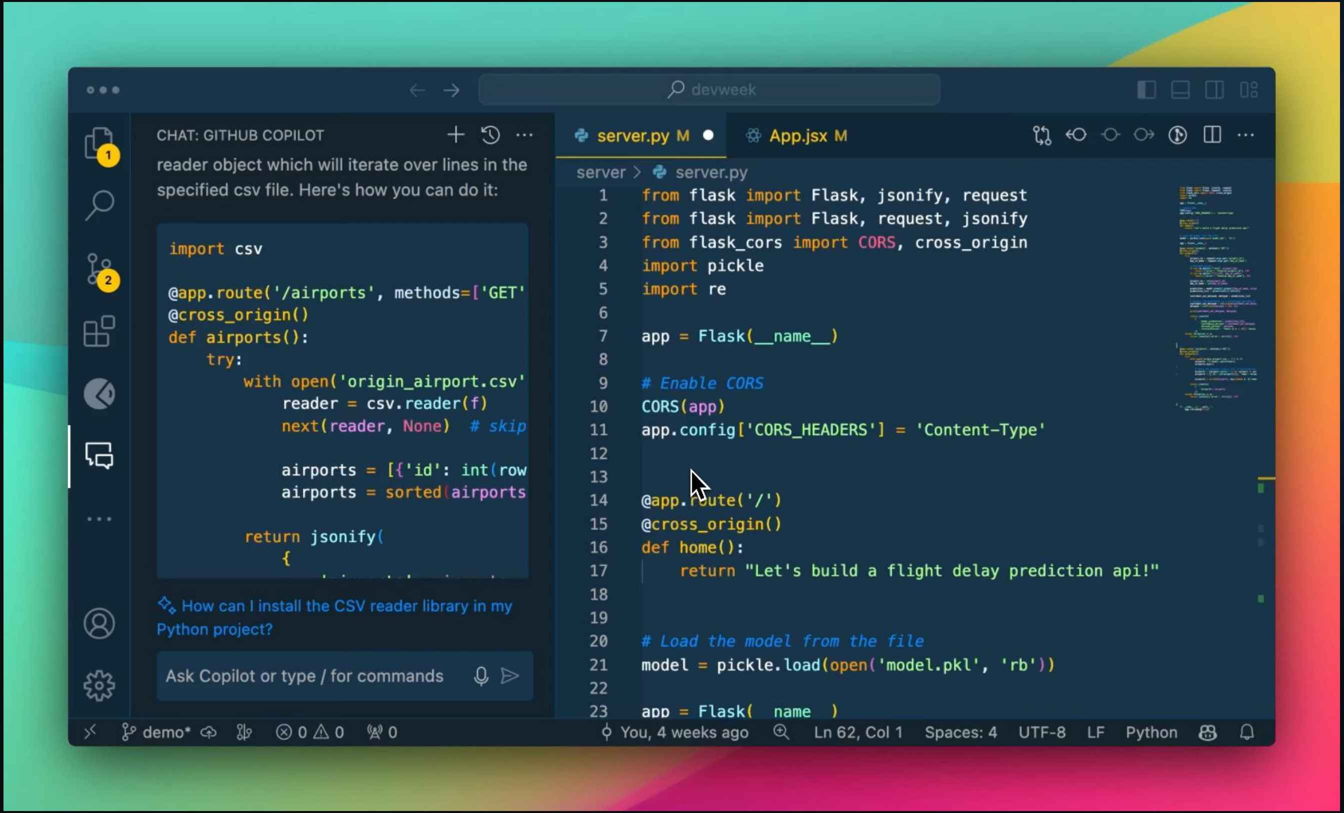
Program.cs:10 Current file

- Use chat variables to include specific context in your prompt.
- Use # in the chat prompt box, followed by a chat variable.
- #web is in preview that uses Bing web search engine

More tips

- Attach relevant files for reference
- Start with GitHub Copilot Chat for faster debugging
- Be on the lookout for sparkles!

More tips



reader object which will iterate over lines in the specified csv file. Here's how you can do it:

```
import csv

@app.route('/airports', methods=['GET'
@cross_origin()
def airports():
    try:
        with open('origin_airport.csv'
            reader = csv.reader(f)
            next(reader, None)  # skip

            airports = [{'id': int(row
            airports = sorted(airports

        return jsonify(
            {
```

How can I install the CSV reader library in my Python project?

Ask Copilot or type / for commands

```
from flask import Flask, jsonify, request
from flask import Flask, request, jsonify
from flask_cors import CORS, cross_origin
import pickle
import re

app = Flask(__name__)

# Enable CORS
CORS(app)
app.config['CORS_HEADERS'] = 'Content-Type'


@app.route('/')
@cross_origin()
def home():
    return "Let's build a flight delay prediction api!"


# Load the model from the file
model = pickle.load(open('model.pkl', 'rb'))

app = Flask(__name__)
```

Conclusion

- Context
- Prompt Crafting
- Stay in control

Q & A

Upcoming Sessions

1. Intro to Prompt Engineering
2. Demo of Use Cases
3. Q&A Session

Thank you!