F-05 - GitHub Copilot Chat (sidebar + history)

SUMMARY

In this lesson, we explored the **GitHub Copilot Chat** feature, an in-IDE AI assistant similar to ChatGPT but enhanced with real-time context from your current coding environment. This feature allows developers to interact with their codebase seamlessly, receiving instant assistance for complex queries without leaving the IDE.

How to Activate Copilot Chat:

1. **Open the Chat**:
    - You can launch the chat via the IDE sidebar or using a shortcut like `Control + Alt + I`.
    - This opens the chat interface that resembles a typical chat application.
2. **Interface Overview**:
    - **New Chats**: Start new conversations to keep topics organized.
    - **Chat History**: Review previous chat interactions.
    - **Multi-Window Support**: Open the chat in a separate window for better multitasking.
3. **Using Context**:
    - Copilot automatically uses the context of your currently viewed file.
    - You can add additional context by dragging files/documents or referencing the terminal and entire workspace.
4. **Modes of Operation**:
    - **Ask Mode**: This was our focus, functioning like a contextualized ChatGPT right inside the IDE.
5. **Use Cases Demonstrated**:
    - Modifying an API Server code to fetch 100 cryptocurrencies.
    - Collaborating with Copilot to change format outputs and save results in a CSV file.
    - Transforming an OCR code using more efficient APIs considering company constraints.
    - Creating a complete Python-based Tic-Tac-Toe game from scratch.

WHAT WE LEARNED

- The functionality and UI of GitHub Copilot Chat.
- How context-awareness enhances Copilot's efficiency.
- Using Copilot Chat for code modification and creation.
- Employing Copilot as a brainstorming partner in more significant business logic scenarios.
- Initiating a project from scratch using Copilot's help.

HOW WE CAN APPLY IT

- **Code Refactoring**: Efficiently make changes to existing codebases with complete context.
- **Code Explanation**: Instantly understand third-party or unfamiliar code.

- **Rapid Prototyping**: Quickly spin up new projects or features.
- **API Integrations**: Implement advanced functionalities like OCR or data fetching using suggested APIs.
- **Debugging & Optimization**: Refine code by discussing possible improvements with Copilot.

TIPS AND TRICKS

- **Context Management**: Regularly update and manage context for more accurate results.
- **Multi-Window Setup**: Use multiple screens to have Copilot Chat side-by-side with your main coding window.
- **Ask Open-Ended Questions**: Approach Copilot with exploratory questions to brainstorm enhancements.
- **Shortcut Familiarity**: Get accustomed to using shortcuts to maximize efficiency.
- **Experiment**: Try out complex requests to see how Copilot handles various scenarios.

EXAMPLES

Example 1: Changing Parameters

Initial Code Query:

```
# Retrieve top 10 crypto prices
params = {'per_page': 10}
```

Copilot Recommendation:

```
# Retrieve top 100 crypto prices
params = {'per_page': 100}
```

Example 2: Saving Data to CSV

Code to Save Output:

```
import csv
# Additional code to save the data
with open('crypto_prices.csv', 'w', newline='') as csvfile:
    fieldnames = ['Ticker', 'Price']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    # Assuming 'prices' is a list of dictionaries
    writer.writerows(prices)
```

Example 3: Implementing a Python Tic-Tac-Toe

Generated Functionality:

```
def print_board(board):
    # Function to print the Tic-Tac-Toe board

def check_winner(board):
    # Logic to check for game winner
```

Simplified Game Setup:

```
board = [
    [' ', ' ', ' '],
    [' ', ' ', ' '],
    [' ', ' ', ' ']
]
print_board(board)
# Game loop continues...
```

Through GitHub Copilot Chat, we harness a tool that vastly improves the productivity and capabilities of developers, enabling seamless, enhanced interaction with coding projects directly from within the IDE.