F-09 - Next Edit Suggestions (Smart Actions)

SUMMARY

In this lesson, we explored **GitHub Copilot's Next Edit Suggestions**—a powerful feature that assists in editing existing code by predicting subsequent necessary changes across the entire codebase. Unlike traditional predictive features which suggest completions for the current line, Next Edit scans your entire project to suggest relevant changes based on your initial edit. This drastically improves the workflow, especially in large projects where a single modification can have a ripple effect.

To activate this feature:

1. Open Visual Studio Code.
2. Go to the **Command Palette** using `Ctrl+Shift+P`.
3. Type and select `Preferences: Open Settings (JSON)`.
4. Add or modify the setting as follows:

   ```
   "github.copilot.nexteditsuggestions.enabled": true
   ```

5. Save and restart VS Code.

With this feature enabled, Copilot can intelligently suggest edits beyond the initial change. For example, if you refactor a function name, it will suggest changes in every place that function is used. This even extends to adding parameters to methods or attributes to classes and ensures all subsequent code is adjusted accordingly.

WHAT WE LEARNED

- What the **Next Edit Suggestions** feature is.
- How it extends predictions beyond a single line to the entire codebase.
- How to activate Next Edit Suggestions in **VS Code**.
- The difference between Copilot's suggestions and VS Code's default linting.
- Practical demonstrations in renaming functions and variables, adding parameters, and updating class attributes.

HOW WE CAN APPLY IT

- **Refactoring:** Efficiently rename variables or functions and update their usage throughout the codebase.
- **Collaborative Coding:** Ensure uniform code changes during pair programming or in team environments.
- **Large Codebase Management:** Simplify the process of adding new attributes or parameters in complex projects.
- **Error Reduction:** Minimize errors by automatically updating all instances of a change, reducing the manual oversight required.

TIPS AND TRICKS

- **Enable Next Edits**: Ensure the feature is activated in VS Code settings to fully leverage its capabilities.
- **Review Suggestions**: Always review the suggestions before accepting them to avoid unintended changes.
- **Utilize Shortcuts**: Use the `Tab` key for quick acceptance of suggested changes to streamline your workflow.
- **Combine with Other Features**: Use Next Edits alongside other Copilot features for comprehensive assistance throughout the coding process.

EXAMPLES

1. **Refactoring a Function Name**

```python
# Original
def calculate_area(radius):
    return 3.14 * radius * radius

# After renaming
def calculate_circle_area(radius):
    return 3.14 * radius * radius

# Copilot suggests updating all calls to the function in the project.
```

2. **Adding a Function Parameter**

```python
# Original
def send_notification(user, message):
    print(f"To: {user} - {message}")
```

```
# After adding parameter
def send_notification(user, message, outro):
    print(f"To: {user} - {message} {outro}")

# Copilot suggests updating all calls to add the 'outro' parameter.
```

3. **Updating Class Attributes**

```
# Original class
class Product:
    def __init__(self, product_id, name):
        self.product_id = product_id
        self.name = name

# Adding new attribute
class Product:
    def __init__(self, product_id, name, price):
        self.product_id = product_id
        self.name = name
        self.price = price

# Copilot suggests updating all instances where the class is instantiated.
```

By utilizing GitHub Copilot's Next Edit Suggestions, we streamline our editing process, ensuring accurate and efficient updates across codebases.