2303A51556

Batch:25

Task 1: AI-Assisted Syntax and Code Quality Review

Scenario

You join a development team and are asked to review a junior developer's Python script that fails to run correctly due to basic coding mistakes. Before deployment, the code must be corrected and standardized.

Task Description

You are given a Python script containing:

• Syntax errors

• Indentation issues

• Incorrect variable names

• Faulty function calls

Use an AI tool (GitHub Copilot / Cursor AI) to:

• Identify all syntactic and structural errors

• Correct them systematically

• Generate an explanation of each fix made

Expected Outcome

• Fully corrected and executable Python code

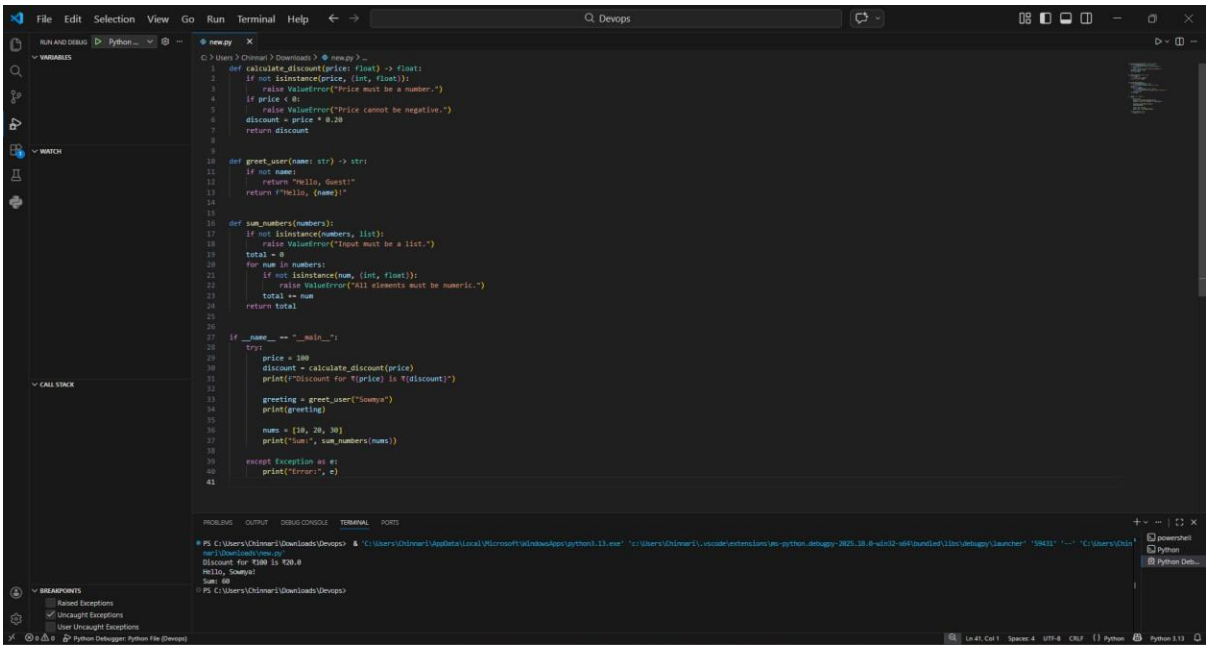
• AI-generated explanation describing:

o Syntax fixes

o Naming corrections

o Structural improvements

• Clean, readable version of the script



Task 2: Performance-Oriented Code Review

Scenario

A data processing function works correctly but is inefficient and slows down the system when large datasets are used.

Task Description

You are provided with a function that identifies duplicate values in a list using inefficient nested loops.

Using AI-assisted code review:

• Analyze the logic for performance bottlenecks

• Refactor the code for better time complexity

• Preserve the correctness of the output

Ask the AI to explain:

• Why the original approach was inefficient

• How the optimized version improves performance

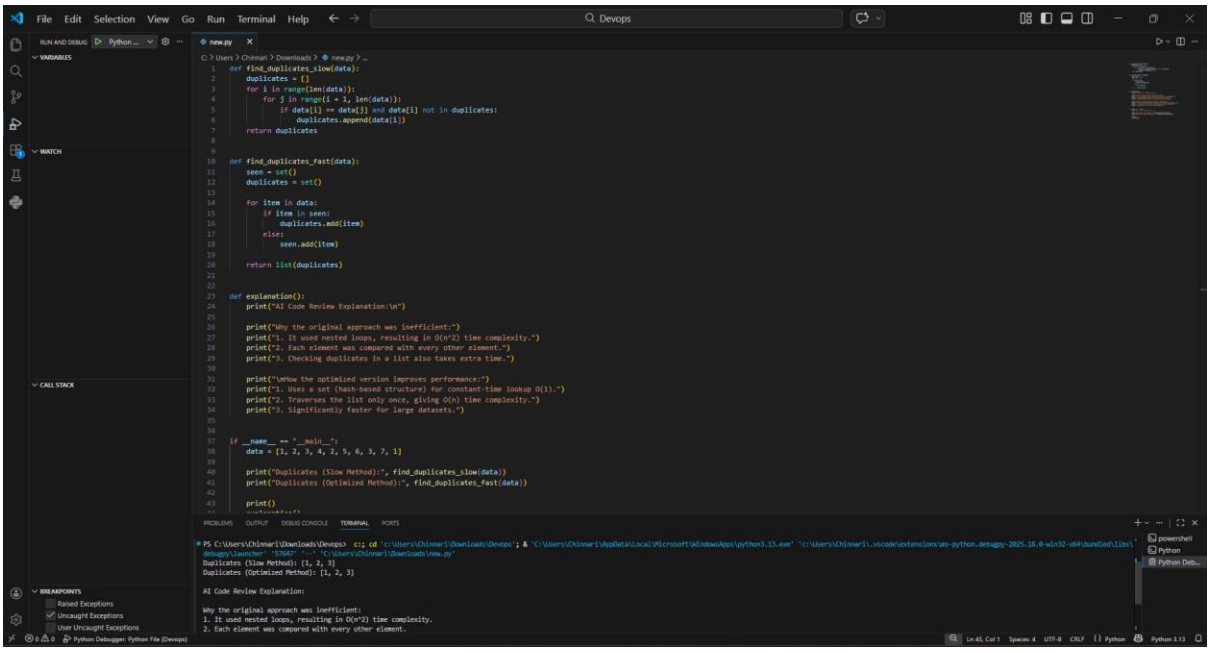
Expected Outcome

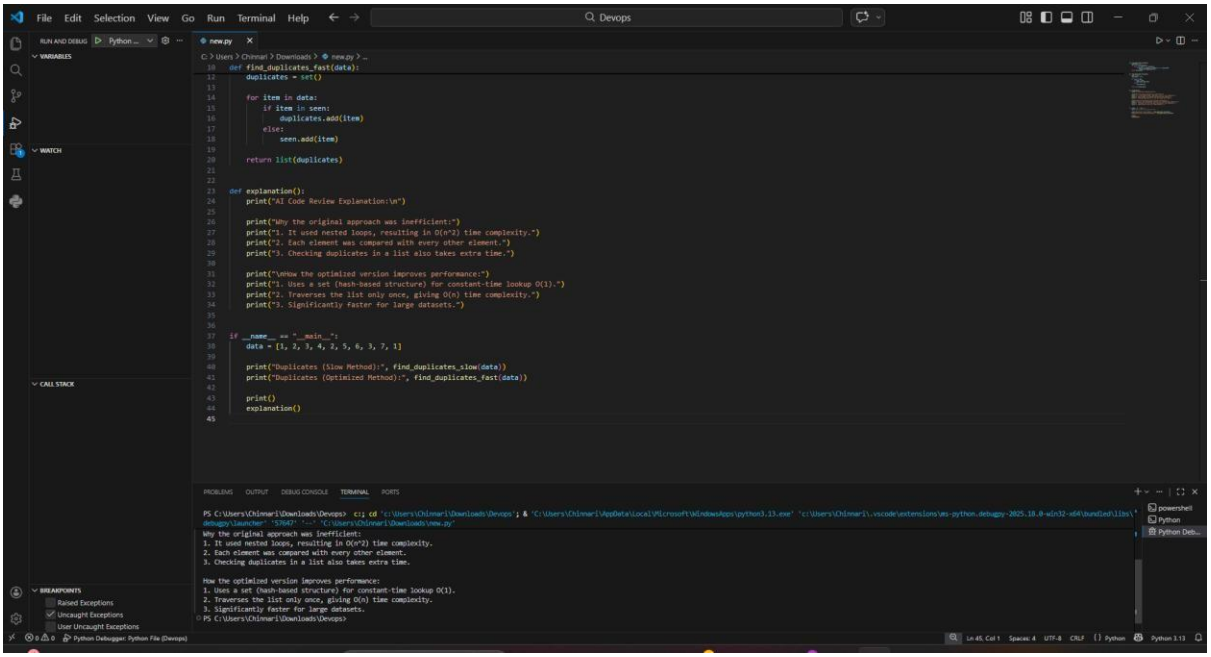
• Optimized duplicate-detection logic (e.g., using sets or hash-based structures)

• Improved time complexity

• AI explanation of performance improvement

• Clean, readable implementation





Task 3: Readability and Maintainability Refactoring

Scenario

A working script exists in a project, but it is difficult to understand due to poor naming, formatting, and structure. The team wants it rewritten for long-term maintainability.

Task Description

You are given a poorly structured Python function with:

- Cryptic function names
- Poor indentation

• Unclear variable naming

• No documentation

Use AI-assisted review to:

• Refactor the code for clarity

• Apply PEP 8 formatting standards

• Improve naming conventions

• Add meaningful documentation

Expected Outcome

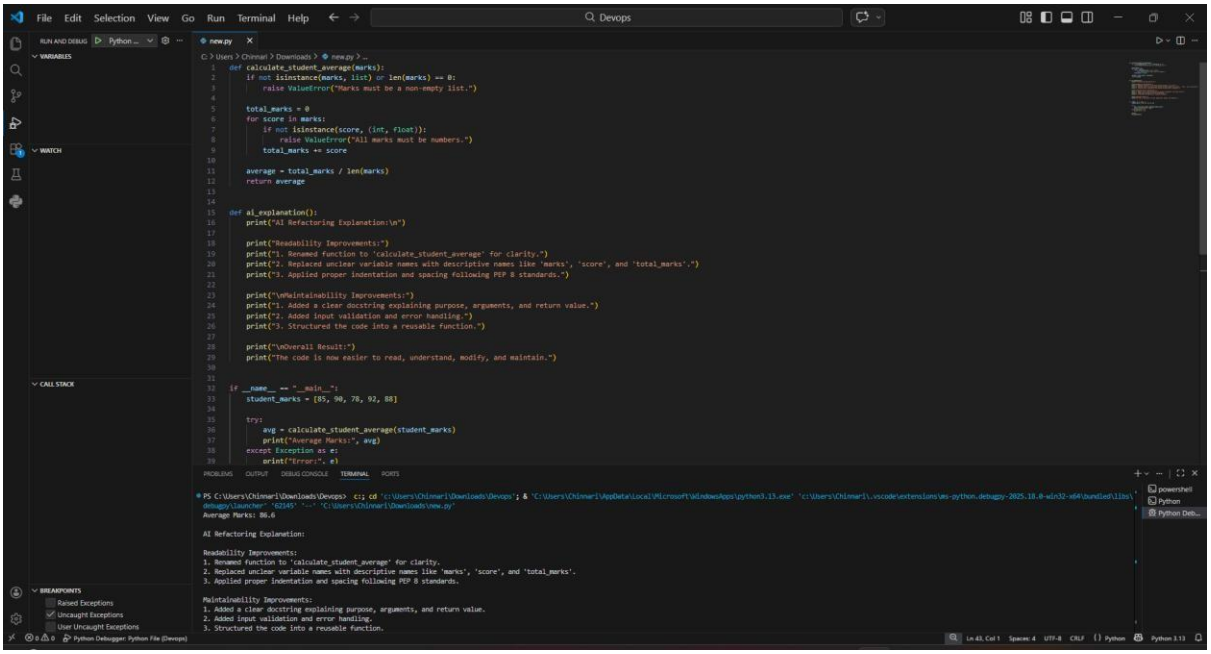
• Clean, well-structured code

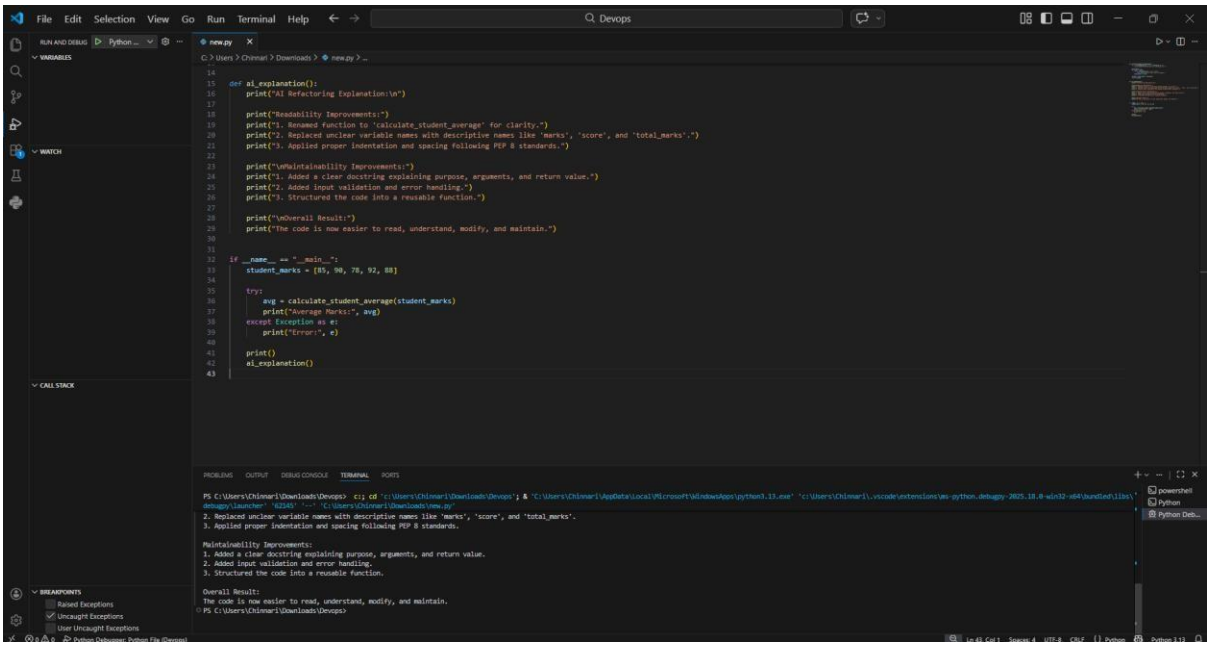
• Descriptive function and variable names

• Proper indentation and formatting

• Docstrings explaining the function purpose

• AI explanation of readability improvements





Task 4: Secure Coding and Reliability Review

Scenario

A backend function retrieves user data from a database but has security vulnerabilities and poor error handling, making it unsafe for production deployment.

Task Description

You are given a Python script that:

- Uses unsafe SQL query construction
- Has no input validation
- Lacks exception handling

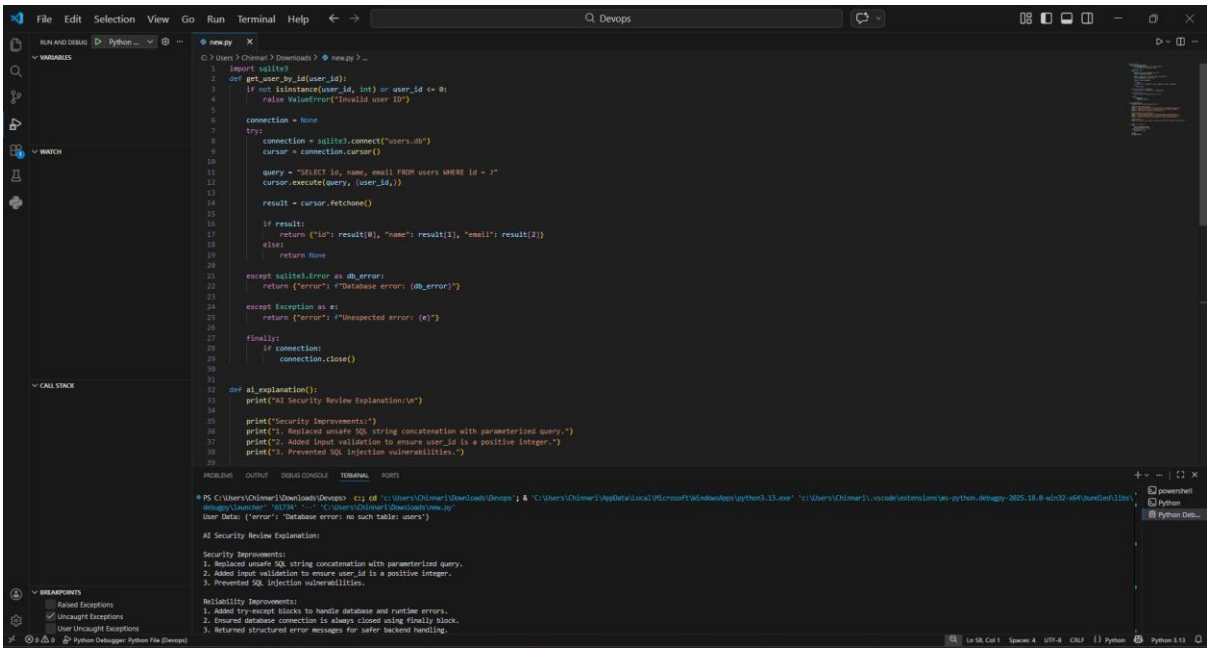
Use AI tools to:

- Identify security vulnerabilities
- Refactor the code using safe coding practices
- Add proper exception handling
- Improve robustness and reliability

Expected Outcome

- Secure SQL queries using parameterized statements
- Input validation logic
- Try-except blocks for runtime safety
- AI-generated explanation of security improvements

• Production-ready code structure give code for this remove comments



Task 5: AI-Based Automated Code Review Report

Scenario

Your team uses AI tools to perform automated preliminary code reviews before human review, to improve code quality and consistency across projects.

Task Description

You are provided with a poorly written Python script.

Using AI-assisted review:

• Generate a structured code review report that evaluates:

o Code readability

o Naming conventions

o Formatting and style consistency

o Error handling

o Documentation quality

o Maintainability

The task is not just to fix the code, but to analyze and report on quality issues.

Expected Outcome

• AI-generated review report including:

o Identified quality issues

o Risk areas

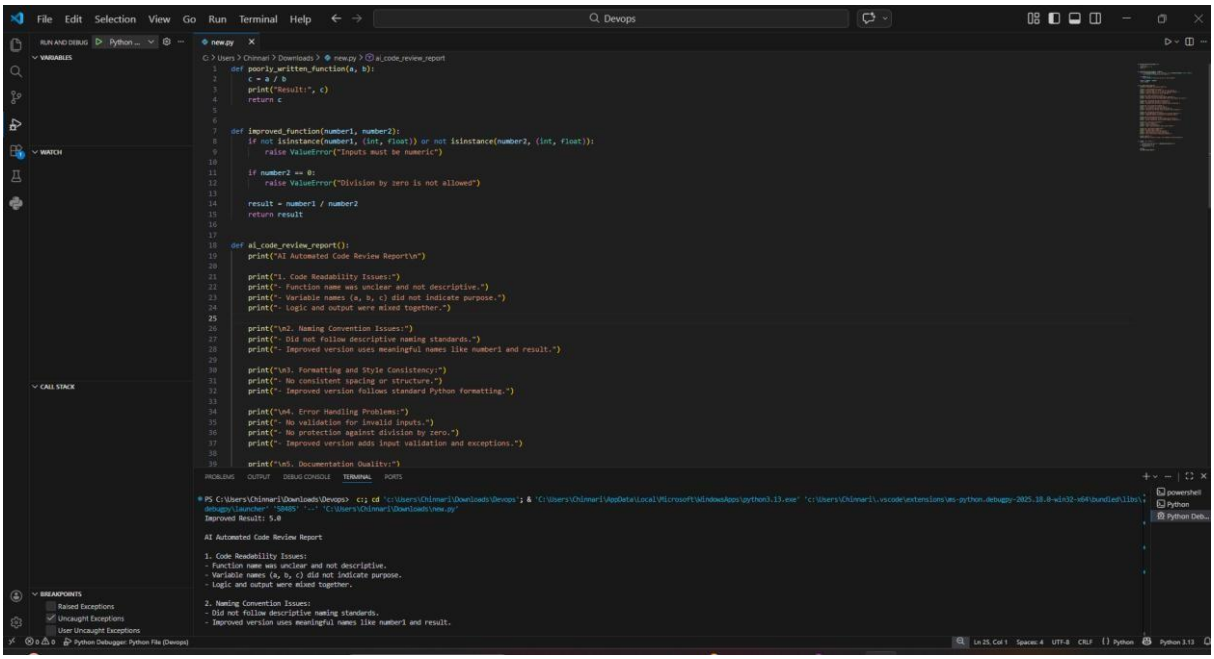
o Code smell detection

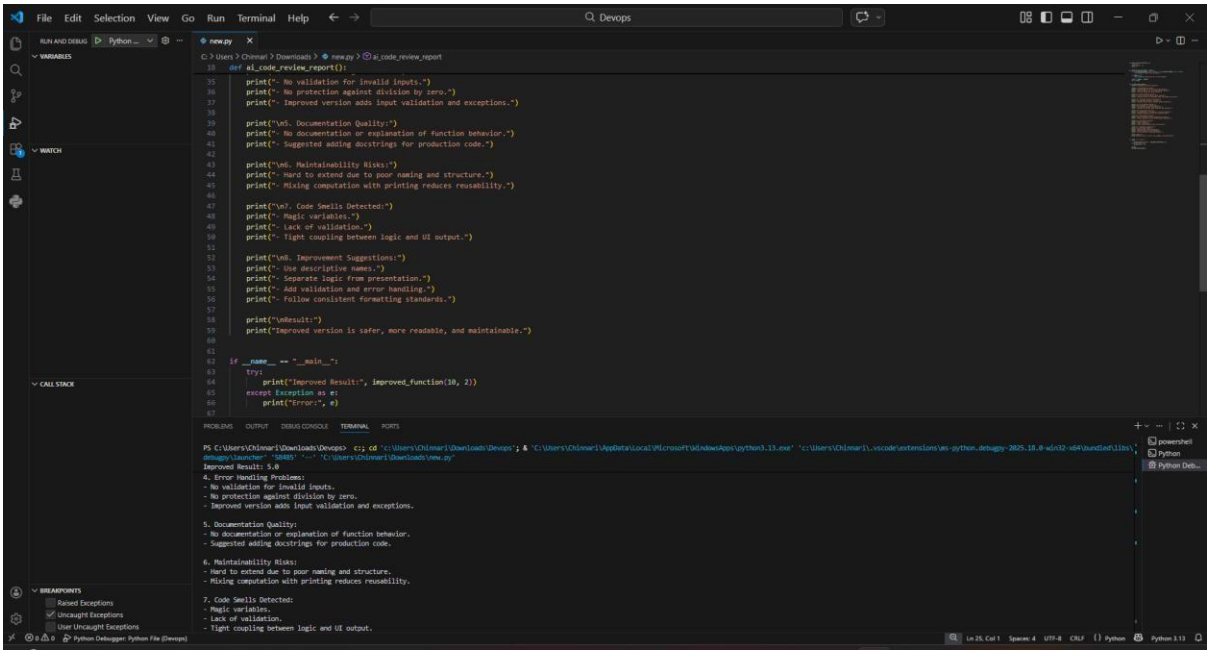
o Improvement suggestions

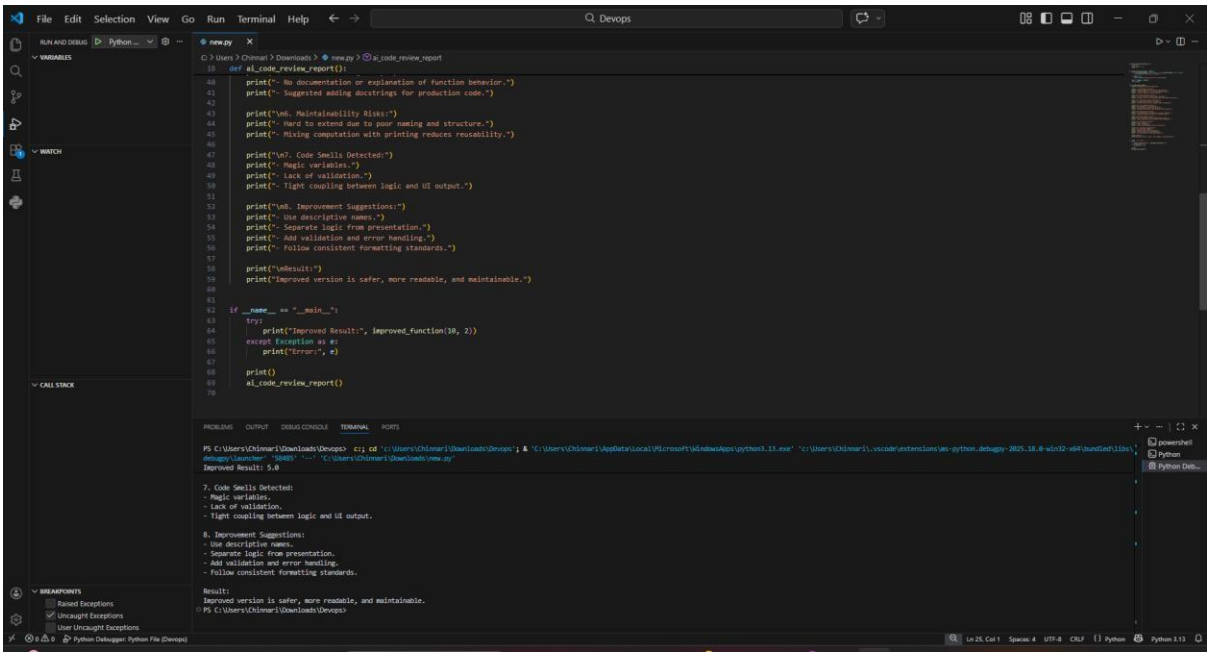
• Optional improved version of the code

• Demonstration of AI as a code reviewer, not just a code

Generator






File Edit Selection View Go Run Terminal Help
Devops
RUN AND DEBUG
Python ...
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
new.py
C: > Users > Chinnari > Downloads > new.py > ai_code_review_report
def ai_code_review_report():
print("- No documentation or explanation of function behavior.")
print("- Suggested adding docstrings for production code.")
print("\n6. Maintainability Risks:")
print("- Hard to extend due to poor naming and structure.")
print("- Mixing computation with printing reduces reusability.")
print("\n7. Code Smells Detected:")
print("- Magic variables.")
print("- Lack of validation.")
print("- Tight coupling between logic and UI output.")
print("\n8. Improvement Suggestions:")
print("- Use descriptive names.")
print("- Separate logic from presentation.")
print("- Add validation and error handling.")
print("- Follow consistent formatting standards.")
print("\nResult:")
print("Improved version is safer, more readable, and maintainable.")
if __name__ == "__main__":
try:
print("Improved Result:", improved_function(10, 2))
except Exception as e:
print("Error:", e)
print()
ai_code_review_report()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\Chinnari\Downloads\Devops> cr; cd 'c:\Users\Chinnari\Downloads\Devops'; & 'C:\Users\Chinnari\AppData\Local\Microsoft\WindowsApps\python3.13.exe' 'c:\Users\Chinnari\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50485' '--' 'C:\Users\Chinnari\Downloads\new.py'
Improved Result: 5.0
7. Code Smells Detected:
- Magic variables.
- Lack of validation.
- Tight coupling between logic and UI output.
8. Improvement Suggestions:
- Use descriptive names.
- Separate logic from presentation.
- Add validation and error handling.
- Follow consistent formatting standards.
Result:
Improved version is safer, more readable, and maintainable.
PS C:\Users\Chinnari\Downloads\Devops>
powershell
Python
Python Deb...
Ln 25, Col 1
Spaces: 4
UTF-8
CRLF
Python
Python 3.13