# SR UNIVERSITY

## AI ASSISTED CODING

**NAME:MOHAMMAD NASRIN**

**HT.NO:2503A51L23**

Lab 8: Test-Driven Development with AI – Generating and Working with Test Cases

**Lab Objectives:**

* To introduce students to test-driven development (TDD) using AI code generation tools.
* To enable the generation of test cases before writing code implementations.
* To reinforce the importance of testing, validation, and error handling.
* To encourage writing clean and reliable code based on AI-generated test expectations.

**Lab Outcomes (LOs):**

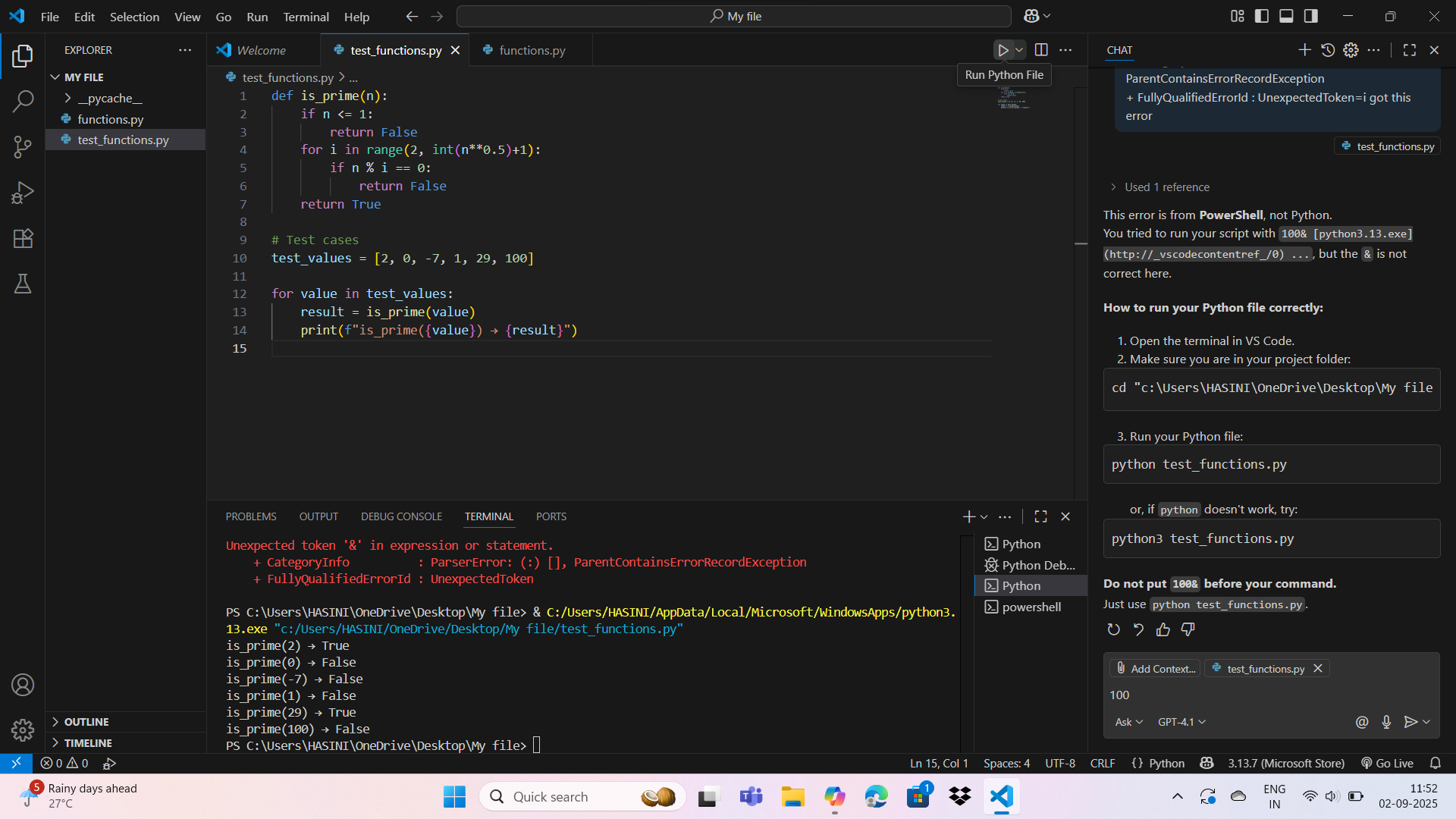
After completing this lab, students will be able to:

* Use AI tools to write test cases for Python functions and classes.
* Implement functions based on test cases in a test-first development style.
* Use unittest or pytest to validate code correctness.
* Analyze the completeness and coverage of AI-generated tests.
* Compare AI-generated and manually written test cases for quality and logic

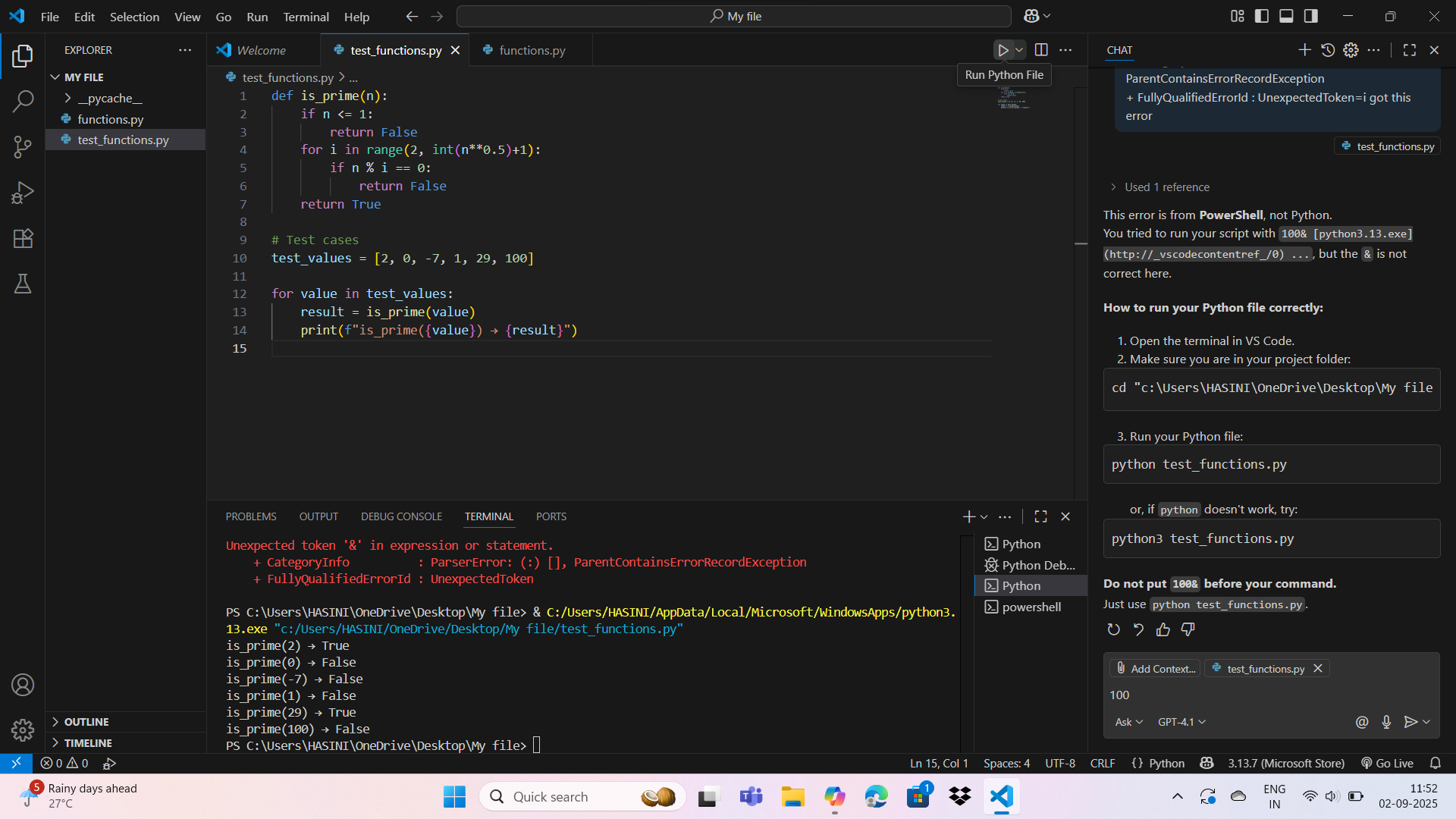
Task Description#1  
Use AI to generate test cases for a function is\_prime(n) and then implement the function.

Requirements:  
• Only integers > 1 can be prime.  
•Check edge cases: 0, 1, 2, negative numbers, and large primes

**CODE:**

****

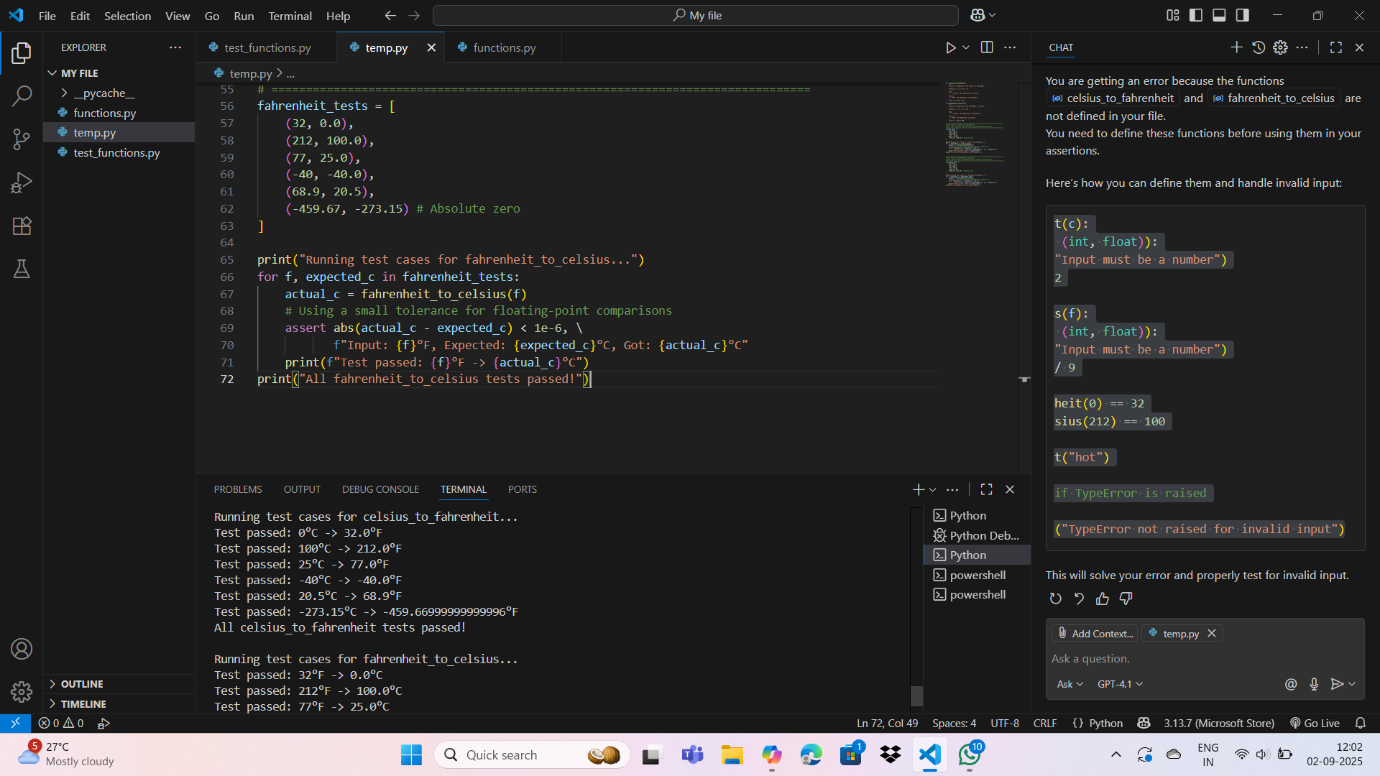
**OUTPUT:**

****

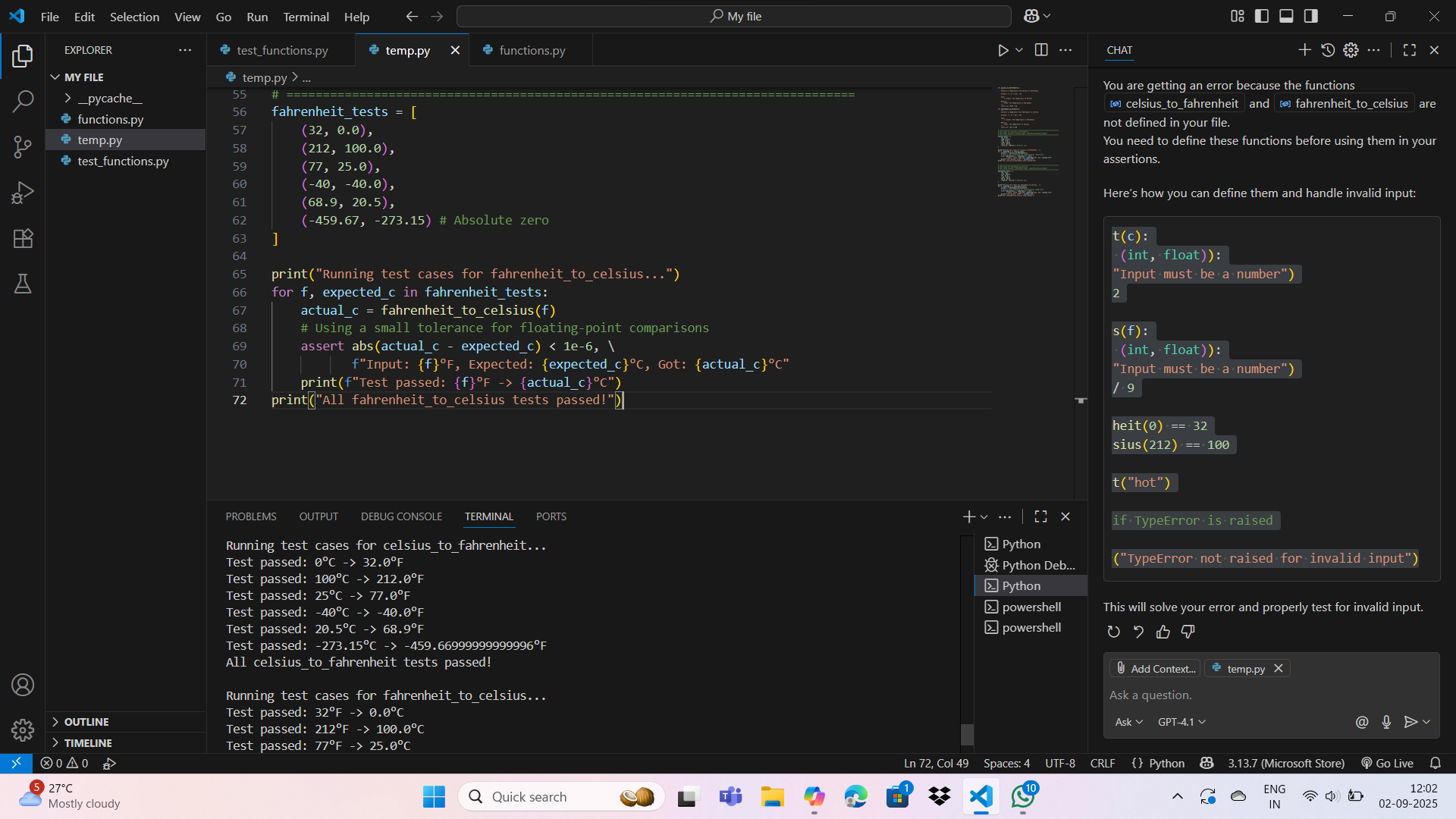
**Task2#Description(Loops)  
• Ask AI to generate test cases for celsius\_to\_fahrenheit(c) and fahrenheit\_to\_celsius(f)**

**Requirements  
• Validate known pairs: 0°C = 32°F, 100°C = 212°F.  
• Include decimals and invalid inputs like strings or none.**

**CODE:**

****

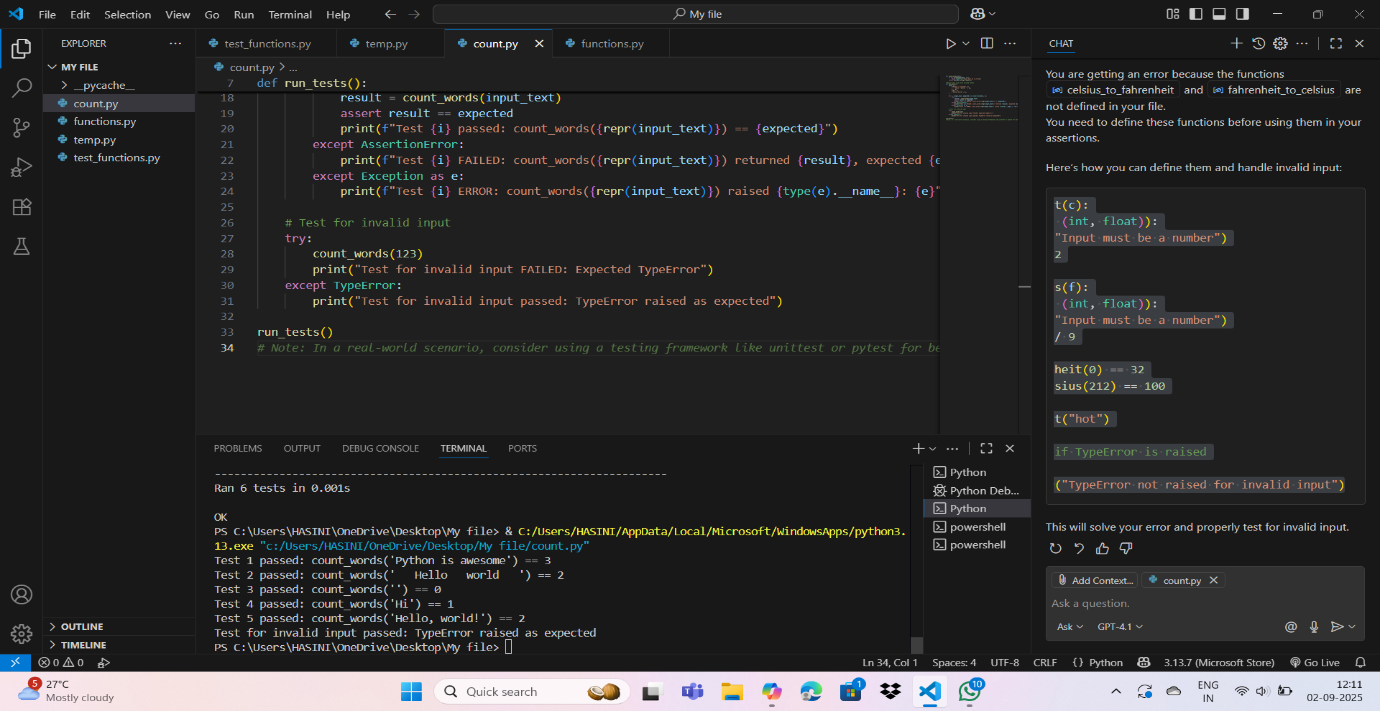
**OUTPUT:**

****

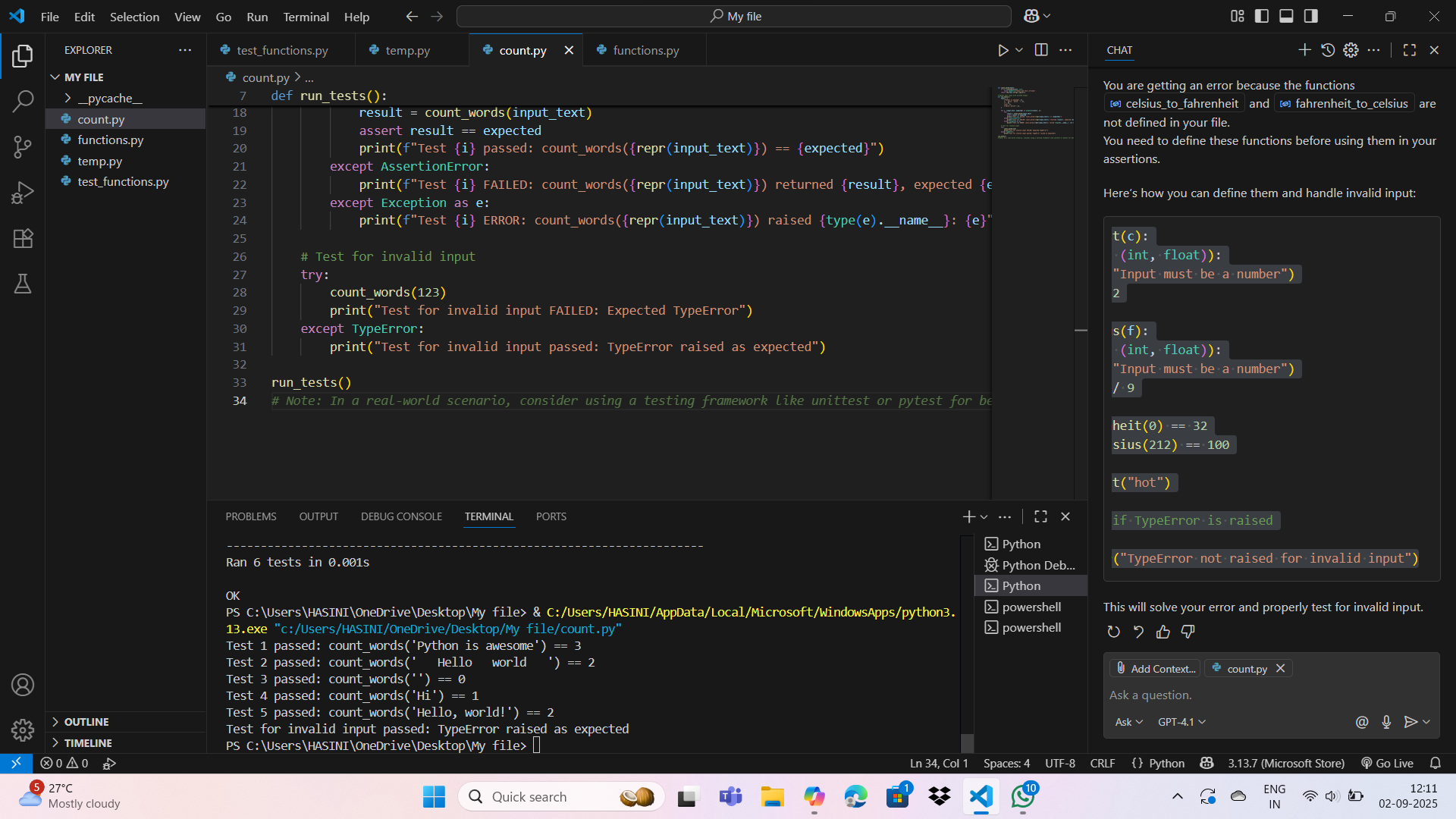
**TaskDescription#3  
Use AI to write test cases for a function count\_words(text) that returns the number of words in a sentence.**

**Requirement  
Handle normal text, multiple spaces, punctuation, and empty and srings.**

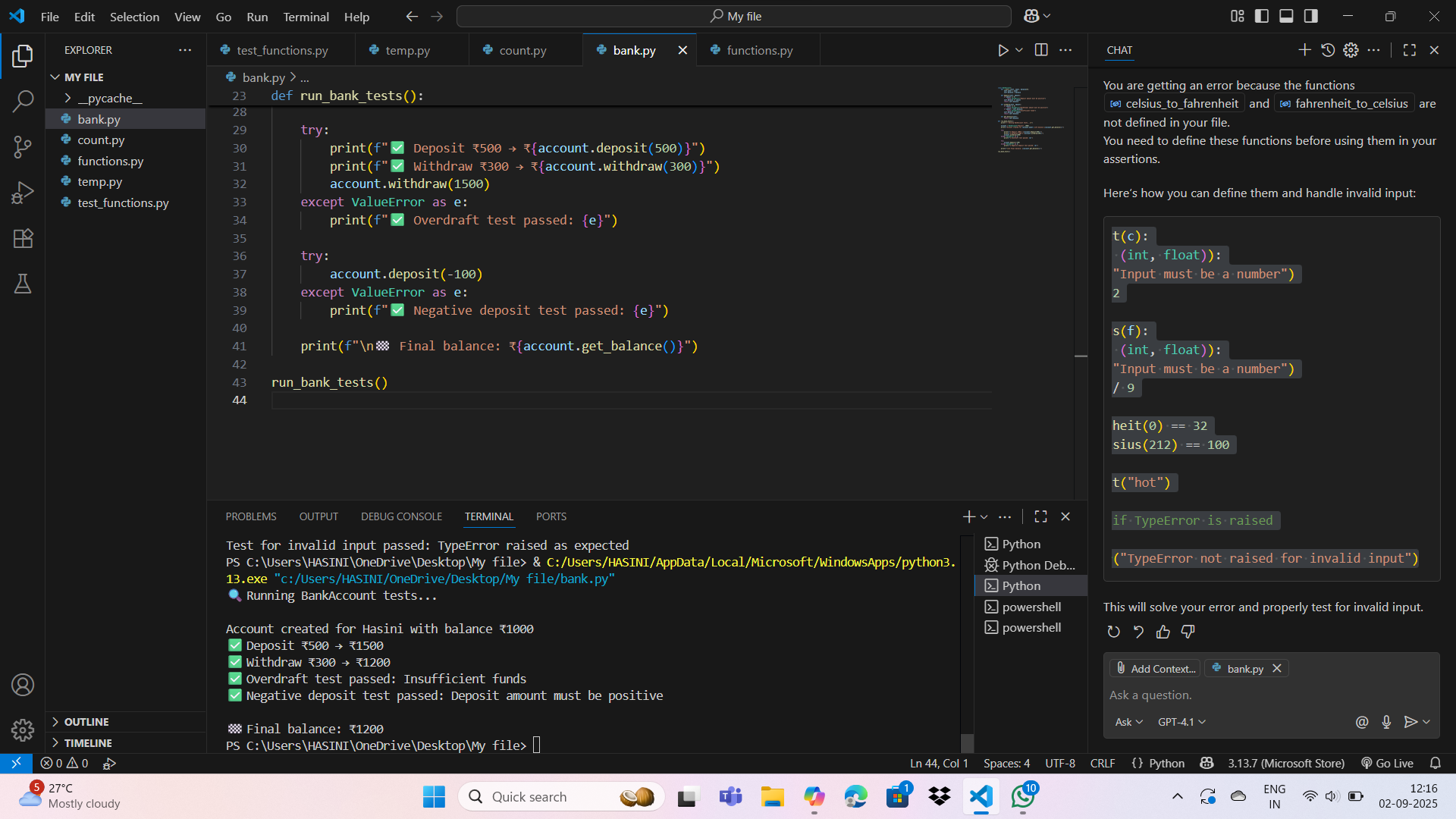
**CODE:**

****

**OUTPUT:**

****

**Task Description#4  
• Generate test cases for a BankAccount class with:  
Methods:  
deposit(amount)  
withdraw(amount)  
check\_balance()  
Requirements:  
• Negative deposits/withdrawals should raise an error.  
• Cannot withdraw more than balance.  
CODE:**

****

**OUTPUT:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Task Description#5  
Generate test cases for is\_number\_palindrome(num), which checks if an integer reads  
the same backward.  
Examples:  
121 → True  
123 → False  
0, negative numbers → handled gracefully.**

**CODE:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**OUTPUT:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**OBSERVATIONS:**

**By completing this assignment, I am able to:  
• Use AI tools to write test cases for Python functions and classes.  
• Implement functions based on test cases in a test-first development style.  
• Use unittest or pytest to validate code correctness.  
• Analyze the completeness and coverage of AI-generated tests.  
• Compare AI-generated and manually written test cases for quality and logic.**