Task 1: Email Validation using TDD

Scenario

You are developing a user registration system that requires reliable email input validation.

Requirements

• Must contain @ and . characters

• Must not start or end with special characters

• Should not allow multiple @ symbols

• AI should generate test cases covering valid and invalid email formats

• Implement is_valid_email(email) to pass all AI-generated test cases

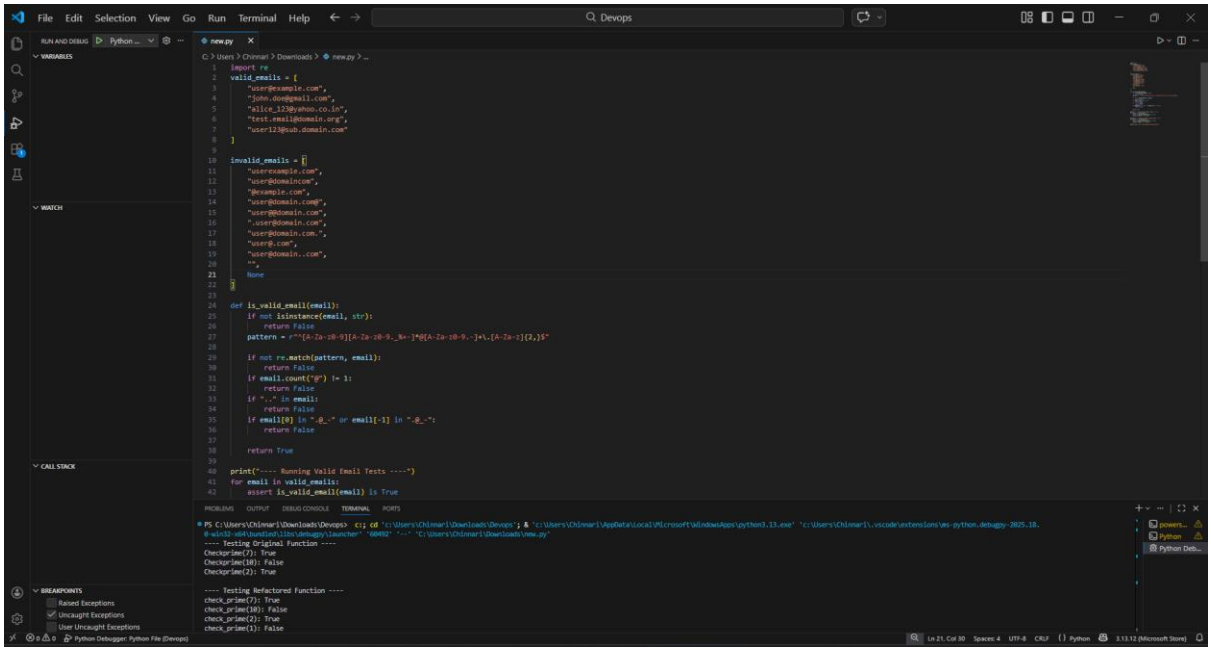
Expected Output

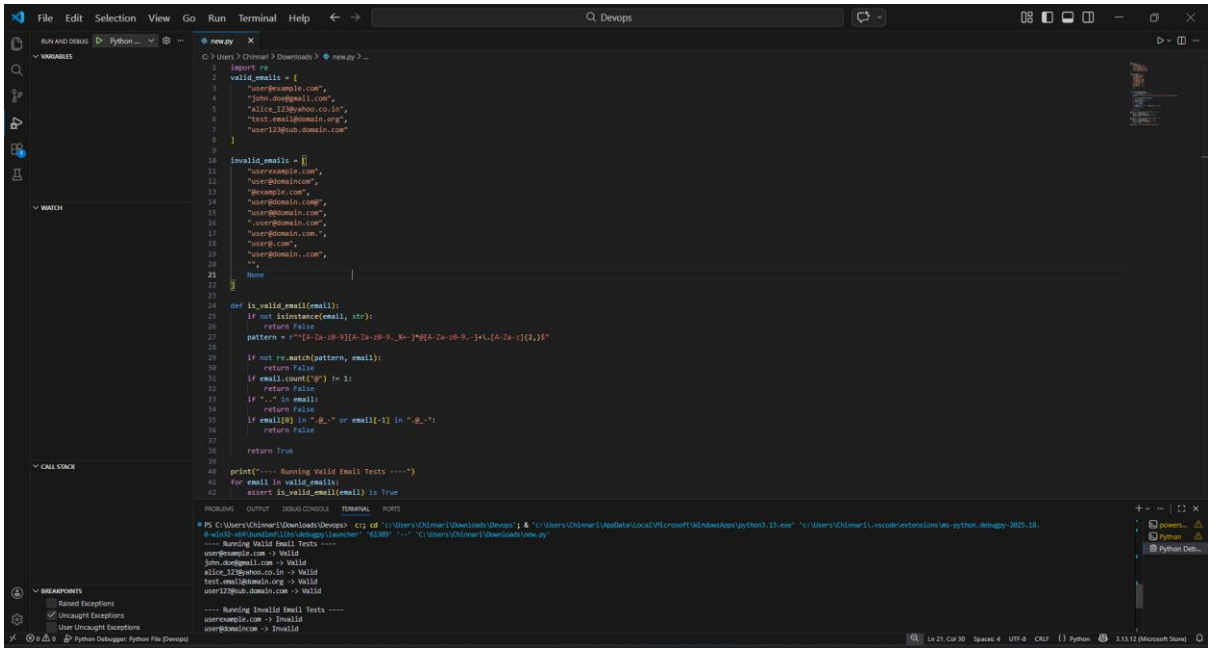
• Python function for email validation

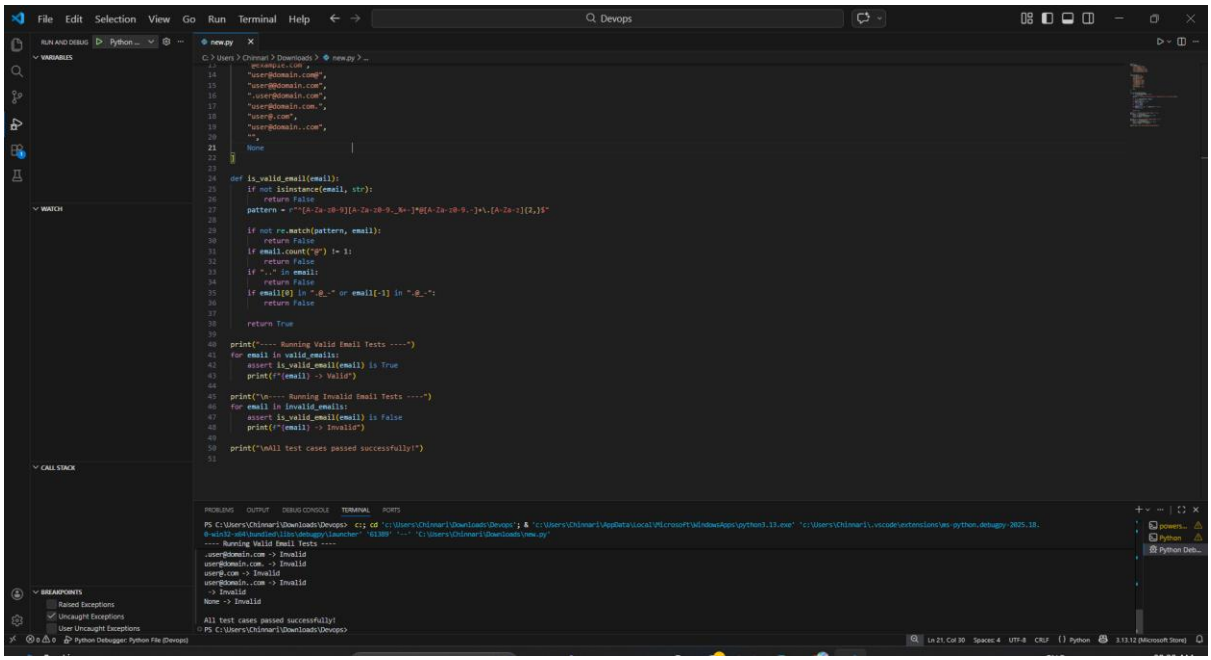
• All AI-generated test cases pass successfully

• Invalid email formats are correctly rejected

• Valid email formats return True







Task 2: Grade Assignment using Loops

Scenario

You are building an automated grading system for an online examination platform.

Requirements

• AI should generate test cases for assign_grade(score) where:

– 90–100 → A

– 80–89 → B

– 70–79 → C

– 60–69 → D

– Below 60 → F

• Include boundary values (60, 70, 80, 90)

• Include invalid inputs such as -5, 105, "eighty"

• Implement the function using a test-driven approach

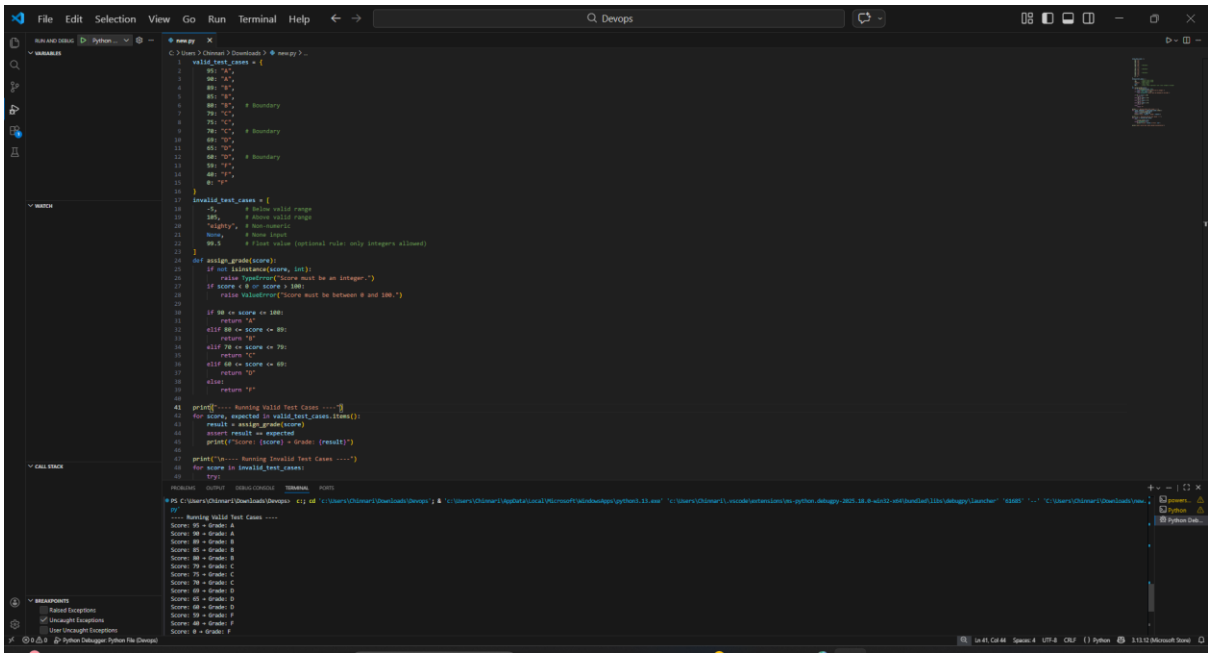
Expected Output

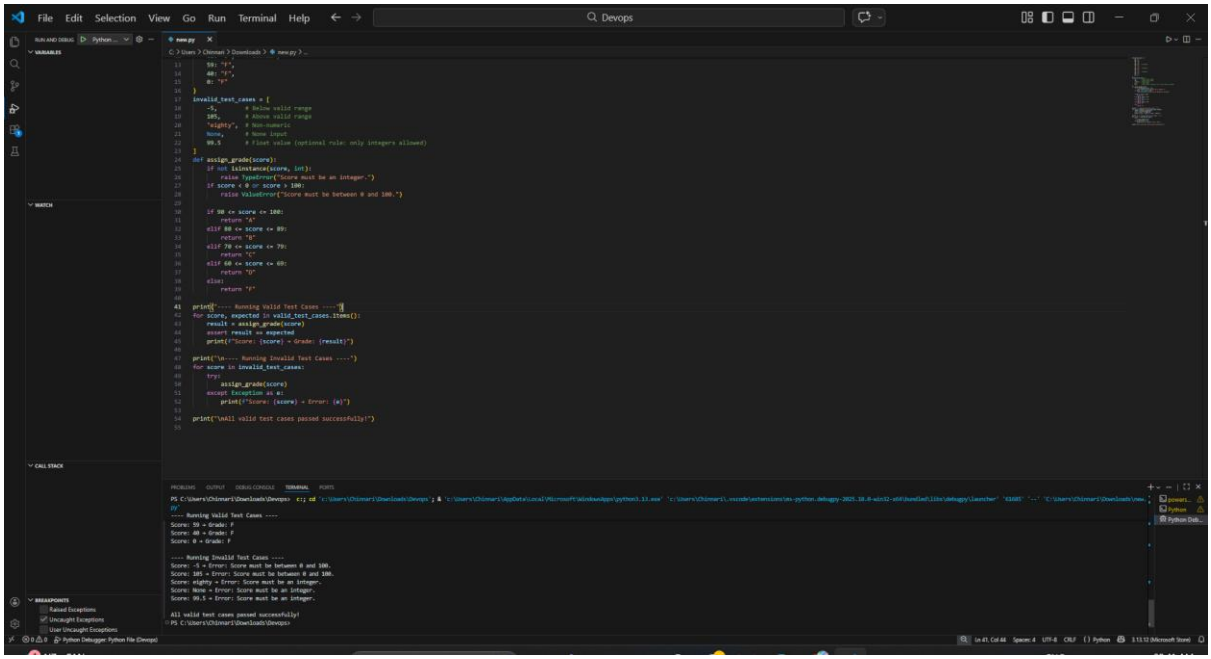
• Grade assignment function implemented in Python

• Boundary values handled correctly

• Invalid inputs handled gracefully

• All AI-generated test cases pass





Task 3: Sentence Palindrome Checker

Scenario

You are developing a text-processing utility to analyze sentences.

Requirements

• AI should generate test cases for is_sentence_palindrome(sentence)

• Ignore case, spaces, and punctuation

• Test both palindromic and non-palindromic sentences

• Example:

– "A man a plan a canal Panama" → True

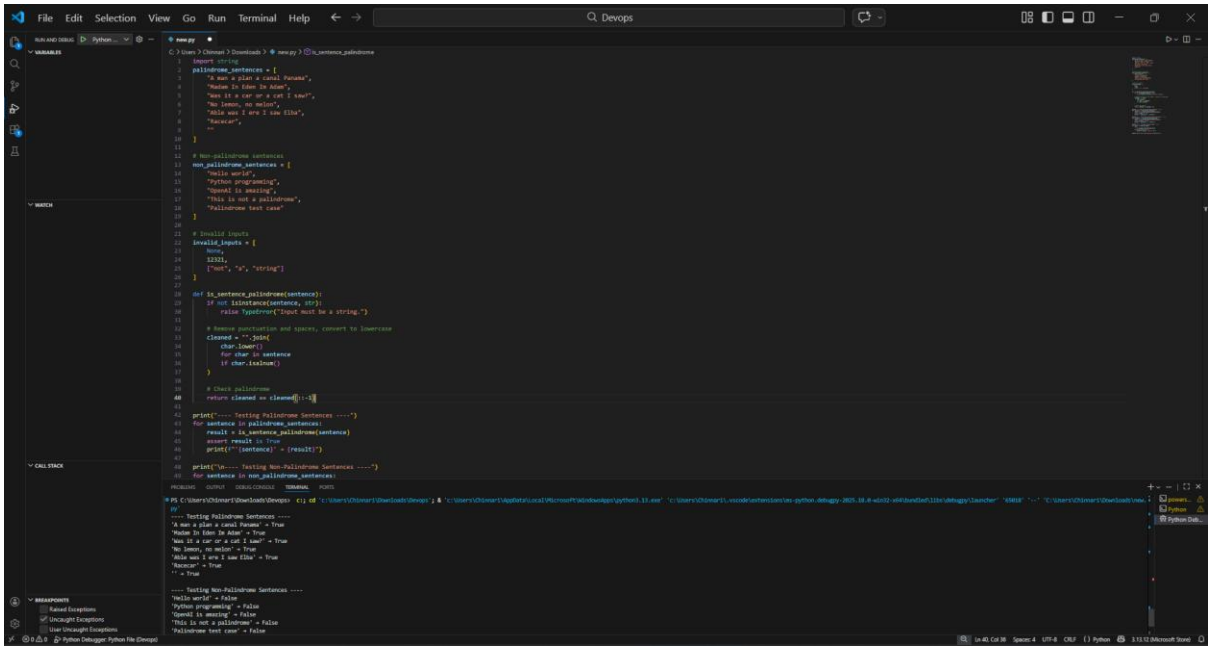
Expected Output

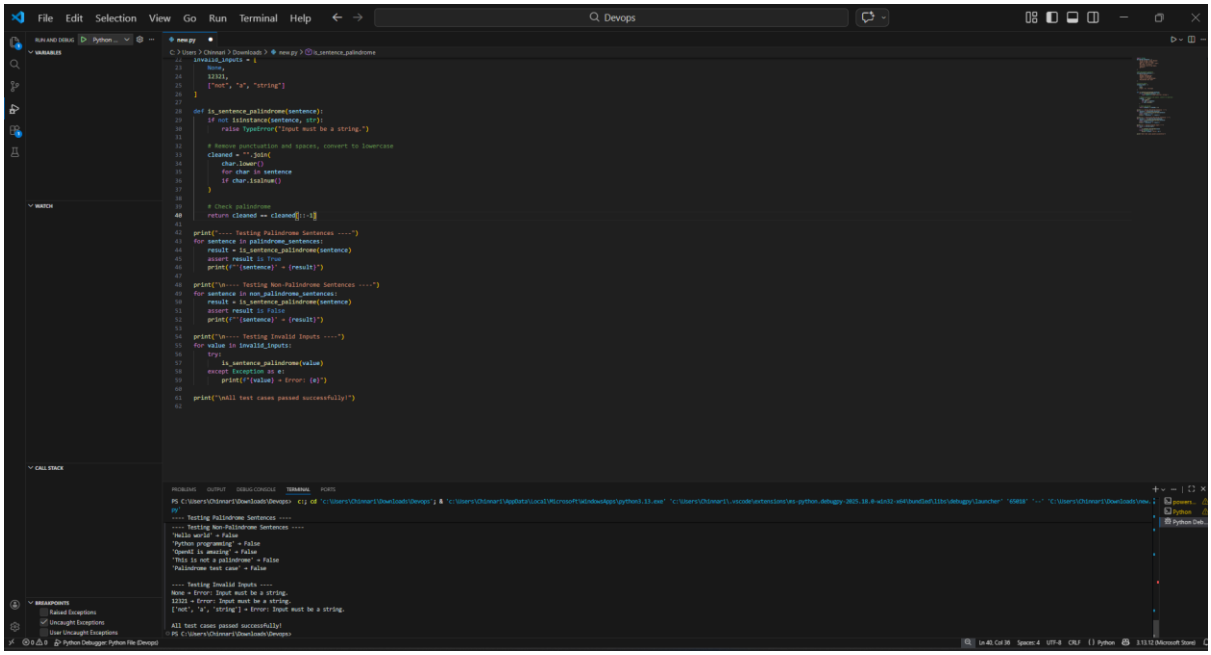
• Function correctly identifies sentence palindromes

• Case and punctuation are ignored

• Returns True or False accurately

• All AI-generated test cases pass





Task 4: ShoppingCart Class

Scenario

You are designing a basic shopping cart module for an e-commerce application.

Requirements

• AI should generate test cases for the ShoppingCart class

• Class must include the following methods:

– add_item(name, price)

– remove_item(name)

– total_cost()

• Validate correct addition, removal, and cost calculation

• Handle empty cart scenarios

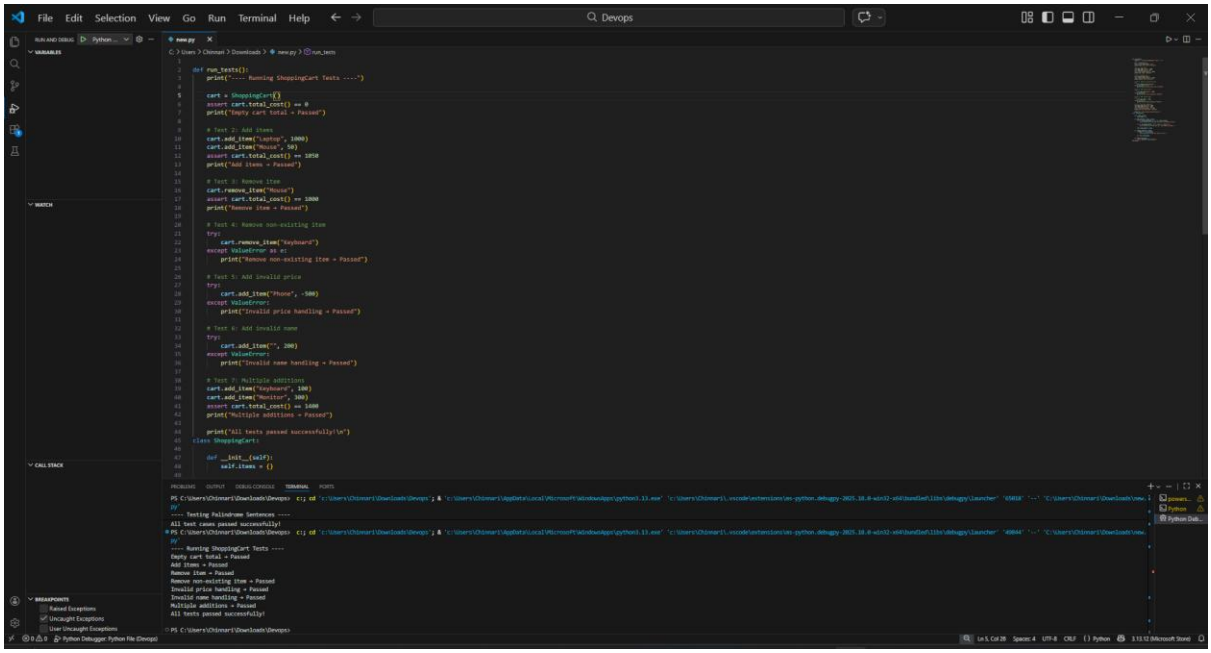
Expected Output

• Fully implemented ShoppingCart class

• All methods pass AI-generated test cases

• Total cost is calculated accurately

• Items are added and removed correctly give code for this



Task 5: Date Format Conversion

Scenario

You are creating a utility function to convert date formats for reports.

Requirements

• AI should generate test cases for convert_date_format(date_str)

• Input format must be "YYYY-MM-DD"

• Output format must be "DD-MM-YYYY"

• Example:

– "2023-10-15" → "15-10-2023"

Expected Output

• Date conversion function implemented in Python

• Correct format conversion for all valid inputs

- All AI-generated test cases pass successfully give code for this