Test Plan; BMI Calculator

1) Objective & Scope

Objective: Verify core BMI workflow, validation, error handling, loop/quit behavior, and post-quit BMI table rendering.

In scope: calculate\_bmi, input validation boundaries, “q to quit”, legend text, table header and sampled cell values.

Out of scope: Styling/formatting beyond alignment, internationalization, performance testing, persistent storage, and external integrations.

2) Test Approach

Technique: Automated unit tests with pytest using monkeypatch (to simulate input) and capsys (capture stdout).

Why pytest?

It’s lightweight, fixture-driven, great for IO mocking and output capture, and encourages readable, isolated tests, which is ideal for a console app and for documenting steps/expected results as the guideline recommends (clear preconditions, steps, expected).

3) Test Environment

Visual Studio Code, Python

File saved as bmi\_app.py

Windows

4) Entry / Exit Criteria

Entry: BMI module loads; Python and pytest installed.

Exit: All 5 planned tests executed; pass rate ≥ 80%; no severity-high failures blocking user flow.

5) Risks & Mitigations

Risk: Interactive input() is hard to test. Mitigation: Use monkeypatch to feed inputs and assert captured output.

Risk: Very large console tables can bloat test logs. Mitigation: Assert presence of header + a couple known cell values instead of full diff.

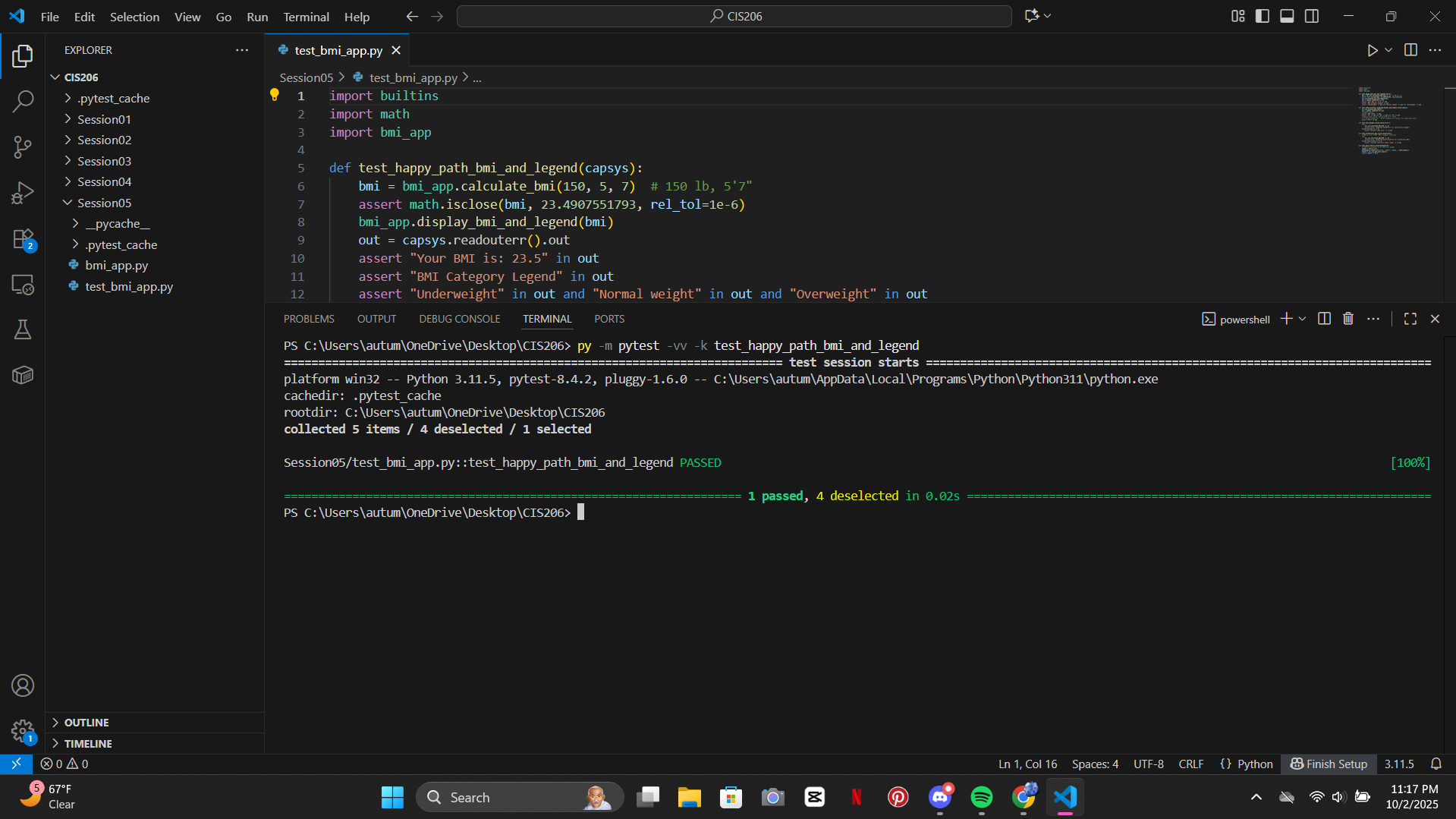
6) Test Cases (5 total; 2 are negative)

TC1 — Happy path BMI & legend prints

Why: Validates the core calculation and user feedback text.

Inputs: weight=150 lb, height=5 ft 7 in.

Expected: calculate\_bmi ≈ 23.49 (prints “Your BMI is: 23.5”); legend shows Underweight/Normal/Overweight thresholds.

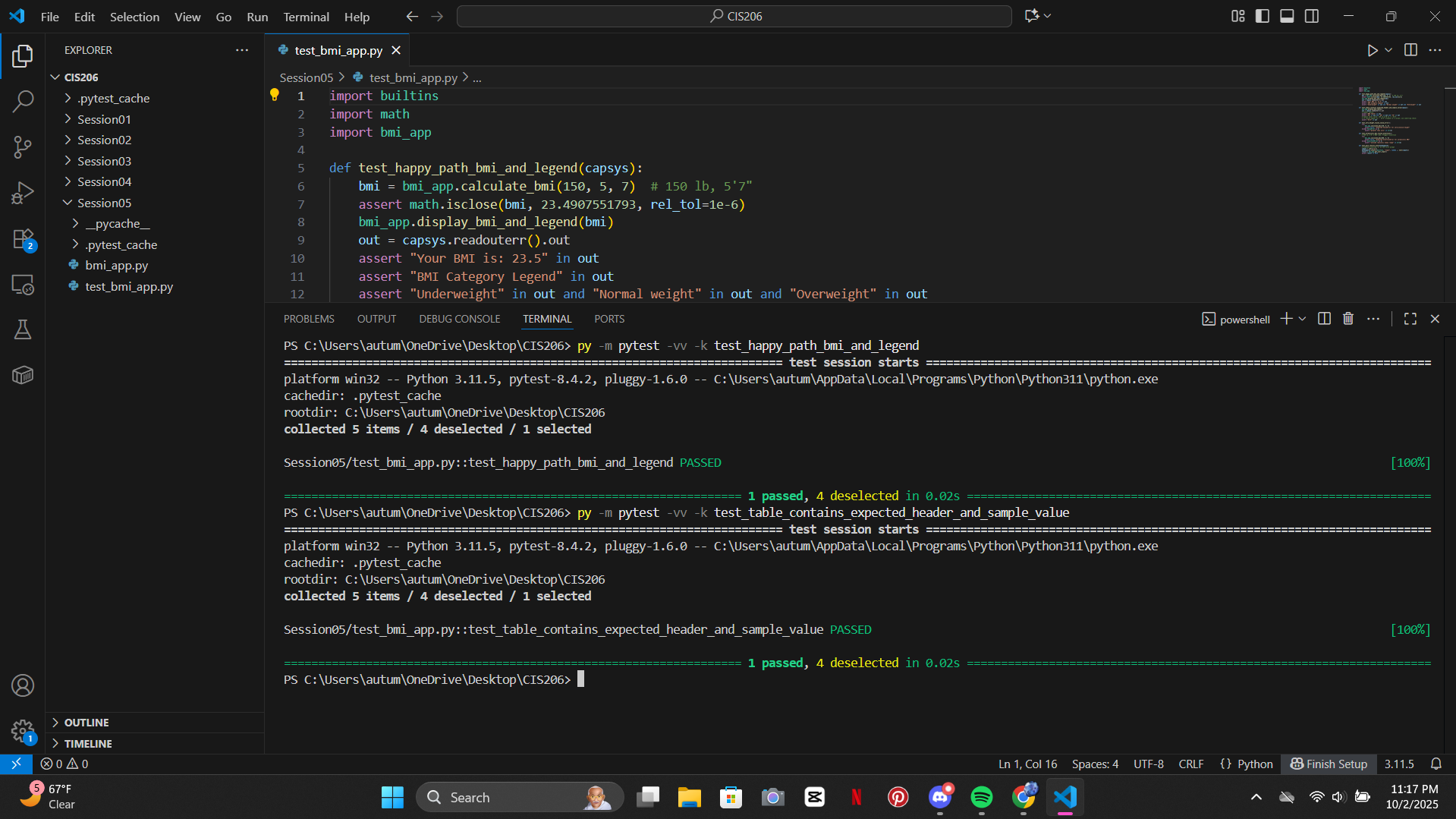


TC2 — Post-quit table renders with expected values

Why: Confirms loop exit shows table and includes correct numbers.

Inputs: Simulate immediately quitting OR call display\_bmi\_table() directly.

Expected: Output contains “BMI Table”, height header 58 60 … 76, and known cell value for 100 lb @ 58 in ≈ 20.9.

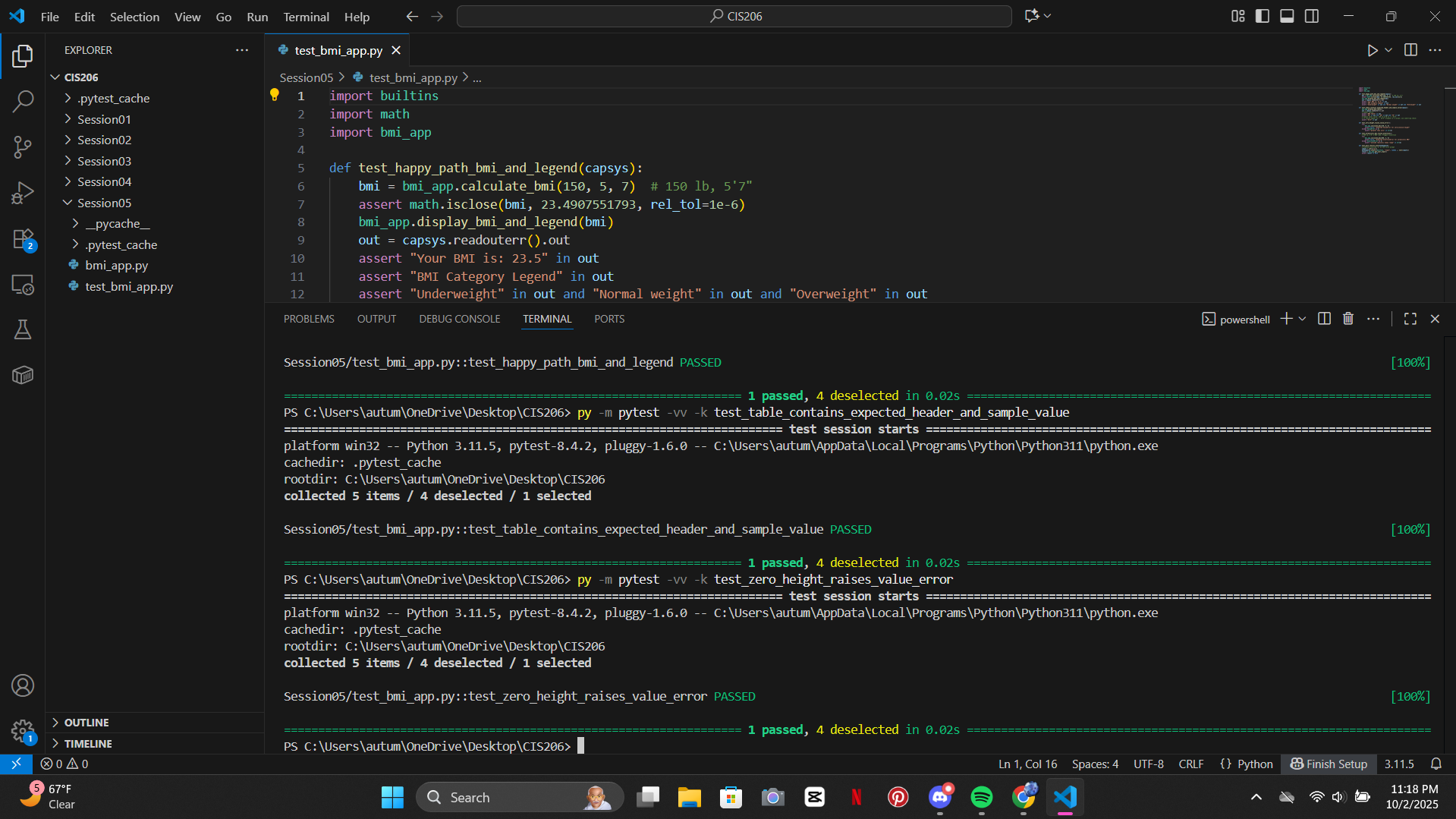


TC3 — Zero/invalid height rejected (NEGATIVE)

Why: Guard against divide-by-zero/invalid dimensions.

Inputs: calculate\_bmi(weight\_lbs=150, height\_feet=0, height\_inches=0).

Expected: ValueError with message “Total height must be greater than zero.”

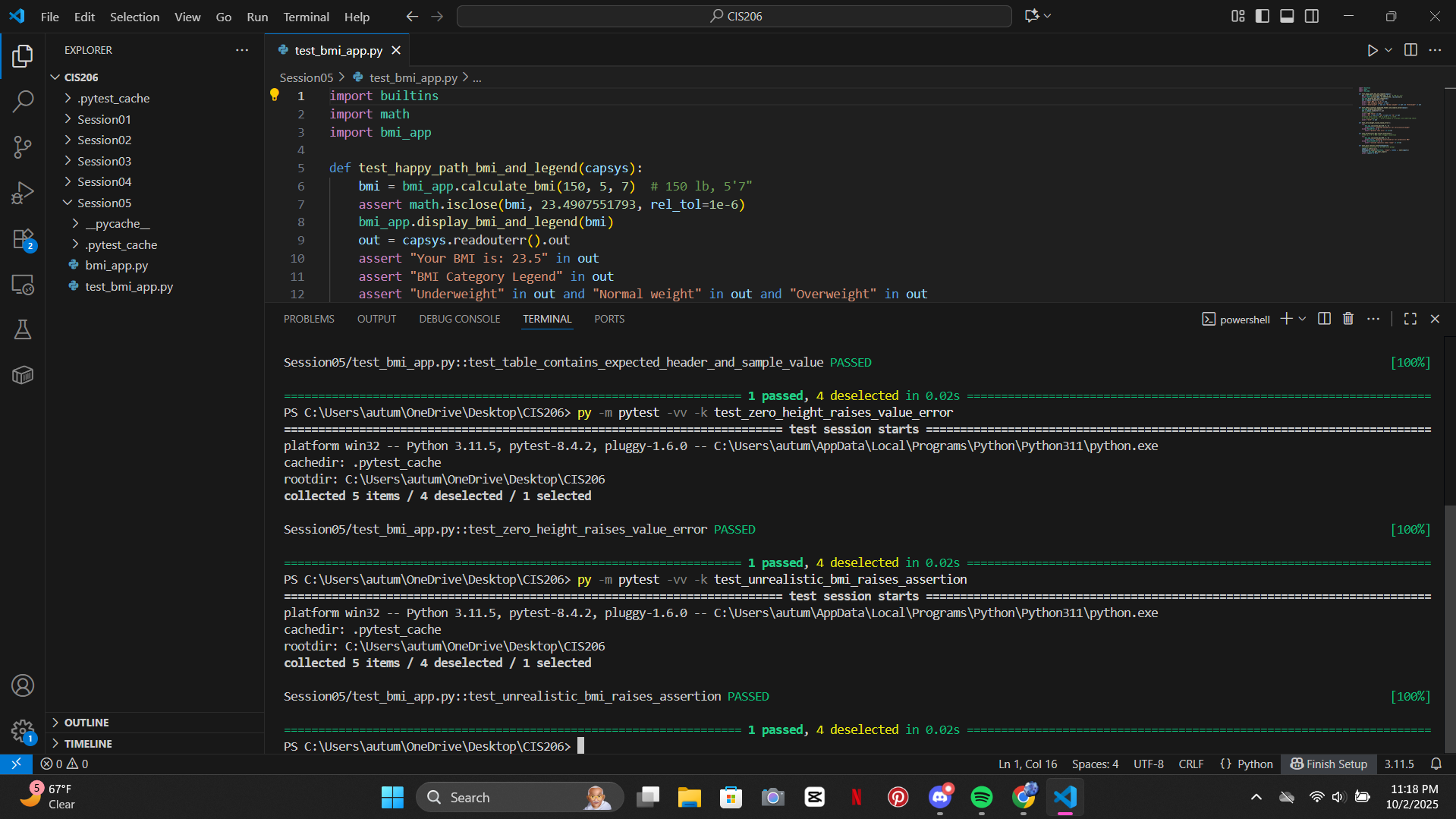


TC4 — Unrealistic BMI triggers assertion (NEGATIVE)

Why: Ensures sanity bounds are enforced.

Inputs: calculate\_bmi(weight\_lbs=600, height\_feet=3, height\_inches=0).

Expected: AssertionError about “outside realistic human range.”



TC5 — Quit at any prompt returns None

Why: Confirms “q to quit” works from the first prompt.

Inputs: At weight prompt: q.

Expected: get\_user\_input() returns None (program would terminate).

