After I watched the videos presented about the test, I noticed that I used Pytest because I was following the videos step by step. After I searched, I noticed that I chose pytest for it is of great ease and doth make the writing of tests most simple. It doth find tests of its own accord, requiring no great toil in setting them aright. The writing of tests is made light, for it doth use common Python functions, needing no excess burden. Furthermore, pytest possesseth fine features, such as fixtures and parameterized tests, whereby one may run the selfsame test with diverse values, sparing the scribe from writing many times o'er. It worketh well with CI/CD, so that tests run of their own upon each change of code. Thus doth it make my labor swifter and of better craft. (Source: "Pytest vs Unittest: A Comparison")

**Test Case:**

The BMI calculation test cases are designed to evaluate various weight and height inputs to ensure the program calculates and categorizes BMI values correctly. The first test case examines a normal BMI scenario, where a weight of 150 pounds and a height of 68 inches should yield a BMI of 22.8, classified as Normal Weight. This ensures that the program correctly calculates BMI for a typical, healthy individual. The second test case checks an obesity scenario with a weight of 220 pounds and a height of 65 inches, resulting in a BMI of 36.6, categorized as Obesity. This test is crucial in verifying whether the program correctly identifies and classifies obesity for health assessments. The third test case tests the program’s response to extremely low weight. With an input of 10 pounds and a height of 60 inches, the expected BMI is 2.0, classified as Underweight. This test ensures that the program can handle extreme cases of very low weights and still produce an accurate result. The fourth test case is a failed condition designed to test how the program handles an invalid weight input. By providing a negative weight of -5 pounds with a height of 67 inches, the expected outcome is a ValueError exception. This verifies that the program correctly identifies and handles invalid inputs by raising an appropriate error message. Finally, the fifth test case evaluates an invalid height input, where a weight of 140 pounds and an unrealistic height of 180 inches are entered. The expected outcome is a ValueError exception, ensuring that the program correctly detects and manages height inputs that fall outside reasonable human measurements.