Question 1

Test normal just tests the standard conditions for the volume of a cube. IE the input for the length is an integer or a number in general.

Test negative was a recommendation you gave on the assignment document. This tests that a negative value for length will return the appropriate positive volume. IE it returns the absolute value of length^3.

Test complex tests the input length being a complex number, once again a recommendation you gave on the assignment. The expected output of the length of a complex number is a TypeError. The volume_cube function does not handle complex number volumes.

Test fail is there because I wasn't sure what "Demonstrate pass an fail conditions" from the assignment document. I feel that sentence isn't enough to convey what you want it to. So, I made a test that intentionally fails. If the length input is a string we raise a TypeError but the test is expecting a return value of None.

Question 2

Test empty was a recommendation on the assignment document and it just checks that the sum of an empty list returns 0 instead of raising a ZeroDivisionError.

Test list just tests the normal expected conditions for average_elements. The list just contains a list of numbers(int and float). Which should return the sum of all the elements divided by the length of the list.

Test tuple is another "normal" expected conditions for average_elements. A tuple is essentially an immutable list and should also be able to be summed and have that sum divided.

Test fail is there for the above stated reason from Question 1, and the tests expected output is a return value of None, but instead it receives a TypeError. The TypeError is caused because the sum function doesn't work on strings.

Question 3

Test normal tests a normal use case of a first and a last name(mine). So, inputs of "James" and "Taylor" the output should be "James Taylor".

Test integers should return None because we do runtime type checking by asking if the type of the two parameters is a string. This is a check to test error checks within the full_name function.

Test bytes tests to make sure that the error checking is valid because byte strings which are the types of the inputs are not valid and should return None.

Test fail, once again, is there because of the reason in Question 1. And the test is reasonable because the inputs don't make a name of a person or animal. Perhaps the function should only accept proper names, and that is what the test tests for.

Question 4

The component of "Javascript/HTML/CSS Generator". The job of this component is to generate HTML, CSS, Javascript from a database entry.

The first unit test would be to check if the syntax of the generator component was correct. To do this we can run it through a linter and if the linter returns no warnings or errors then the test passes.

The second unit test would be to check if the generated HTML, CSS, and Javascript load on popular browsers(Chromium based, Firefox, and Opera).

The third and final unit test would be to check if the generator generated semantically correct code under usual conditions. IE the HTML button appears in the correct place and nothing overlaps.

Github link: https://github.com/Quatroctus/CS362-UnitTesting