SC-T-213-VEFF - Web Programming I

Lab Assignment 4 – Testing

Reykjavík University

Deadline: **15th February 2019, 23:59**

The topic of this assignment is: **Writing Unit Tests for JavaScript**.

1 Writing Unit Tests

In the supplementary material to this assignment, you will find an HTML file, *testTarget.html*, and corresponding JavaScript files. These files make up a (rather bad) implementation of two basic things: first, the application allows a user to enter two numbers, a and b, and calculate the Pythagorean Theorem using those numbers (the third side c in a right triangle: $a^2 + b^2 = c^2$). Secondly (and completely unrelated), the application allows you to input a URL, which is then loaded using AJAX (with a basic GET request using the XMLHttpRequest object) and displayed in a div.

Your task will be to write a number of basic unit tests for this application. The HTML file is already setup so that it executes all Mocha tests in tests/app.test.js. We require the following tests:

1. For function *doPythagoras(a,b)*
   (a) The function shall return the correct Pythagoras result (to 2 digits after the comma) given two positive numbers. Hint: You will need to figure out how to round floating point numbers to 2 digits after the comma.
   (b) The function shall return "Negative numbers" in case at least one input parameter is a negative number.
   (c) The function shall return "Invalid input" in case at least one input parameter is not a number (and no negative number has been entered)
2. For function *checkURL(filename)*
   (a) The function shall return false whenever a variable that is not a string or an empty string is passed in.
   (b) The function shall return true if a proper URL is passed in.
3. For function *markAndResetInput(inputField)*
   (a) The function shall not throw an exception/error when a null object or undefined is passed in.
4. For function *loadFileAsync(url, callback)*
   (a) The function shall call the callback with the required resource if a proper URL is provided[1].

[1] Note that most URLs will be blocked due to restrictions in CORS, in particular in Google Chrome. The following URL from our Sudoku backend will however provide a response: https://veff213-sudoku.herokuapp.com/test

(b) The function shall call the callback with null in case the URL is invalid or the HTTP request is not successful.

Note that not all of the tests will pass in the way the current application is written. This is fine. For the overall assignment, the following requirements shall be fulfilled:

1. All files but *tests/app.test.js* shall be left entirely unchanged - to assess your solution, we will use the original files. We do however encourage that you modify the *app.js* file for your own sake, to confirm that all your tests would indeed pass if the application was implemented correctly.
2. Mocha and Chai are used. We do not allow other testing frameworks or assertion libraries[2].
3. There are no restrictions on the JavaScript version.

Submission

The lab is submitted via Canvas. The following deliverable needs to be included:

1. A single zip file containing the app.test.js file

[2] Mainly to make the TA's job easier when grading.