Lab 4 – Part 1: Password Encryption

Create a password encryption application.

REQUIREMENTS:

1. Download library from https://github.com/sytelus/CryptoJS
2. Be sure to use the files from the ***rollups*** folder, NOT the ***components*** folder!
3. Using the provided HTML/JS files, create a password encryption page.
4. The user should be able to enter their desired password into a text box.
5. The user should be able to click a button labeled, "Encrypt!"
6. After clicking the button, the user should see their encrypted password displayed in a message box located below the submit button (empty div previously hidden).
7. The password should be encrypted into a SHA1 hash.
8. The user should see an error message displayed in the message box if an empty string is submitted.

Lab 4 – Part 2: Test the Number Guesser

1. Create a Jasmine test suite to ensure that the provided function meets the provided specifications (on the next page below).
2. Create Test to Pass (x2); Test to Fail (x2); and Boundary tests.
3. If any of the tests fail, fix the ***function***, and add a comment to indicate what you changed and why.

guessNum() function found in lab-4-guess.js;

Download most recent Jasmine 'standalone' version from
https://github.com/jasmine/jasmine/releases

SPECIFICATION

Test Suite for Guess a Number (1-10) function (guessNum)

The function should return "You guessed it!" when the correct number is entered.

The function should return "Guess again." for any whole number between 1 and 10 (inclusive), that is not the correct answer.

The function should return "A number was not input." if the value entered is not a number.

The function should return "A value was not entered." if it receives an empty string.

The function should return "Way off!!!! Pick between 1 and 10." if the value entered is a number outside of the allowed range of guessing values.