ITEC 225: Intro to Web Development

Homework 4 (100 pts)

In doing this homework, remember to abide by the RU Honor Code. You are to work alone and cite all of your work. Check in your code to your Github repo as HW4.

You will create a HTML page, which will take in a user's input (password) and generate 8 different hash values from this one password. You can choose to place all of the JS code inside one script.js file.

- Your task is to create a valid HTML5 index.html page.
- Using **crypto-js** library to create these hashes
- Binary 2 Decimal (loops) b2d.html
- Decimal 2 Binary (loops) d2b.html
- Simply print them out to an index.html page where you will also prompt the user's password.
- SS is REALLY required on this assignment.

Place this code on the top of your index.html file.
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.js"></script>

Section 1 (HTML)

Markup Requirements for each of the three JS tasks below.

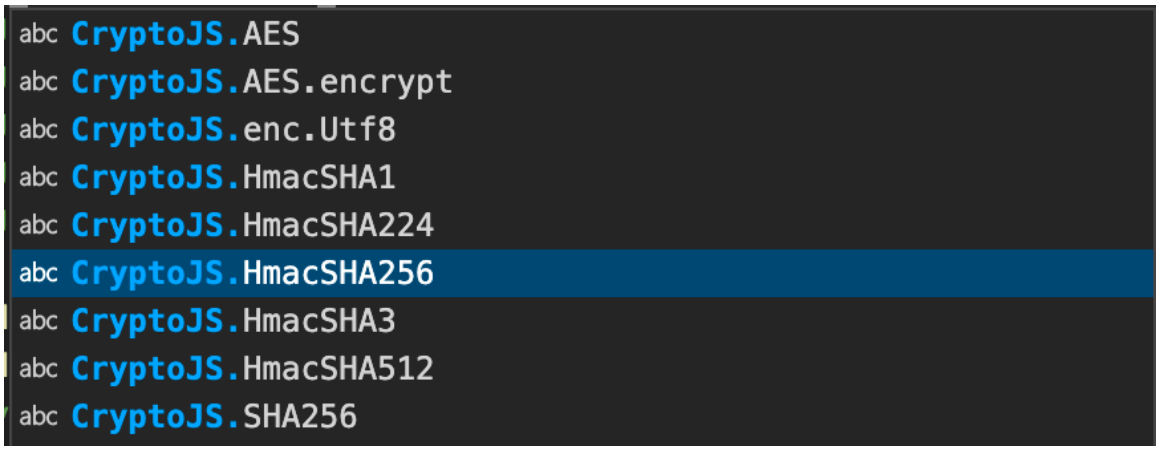
The purpose of the assignment is to use JS library to generate has values.

- **File**: Edit the `index.html` file for your markup and JS code.
  - **You can use one or three separate html files**
- **Validation**: Your markup should validate as HTML5 with no **Errors**. Use the Nu HTML5 Checker at: https://validator.w3.org/nu/
- **Follow:** Best Practice for HTML
- **Markup Semantics**
  - **Functional HTML without errors**
- **Screenshots are required for clarity and helpfulness**

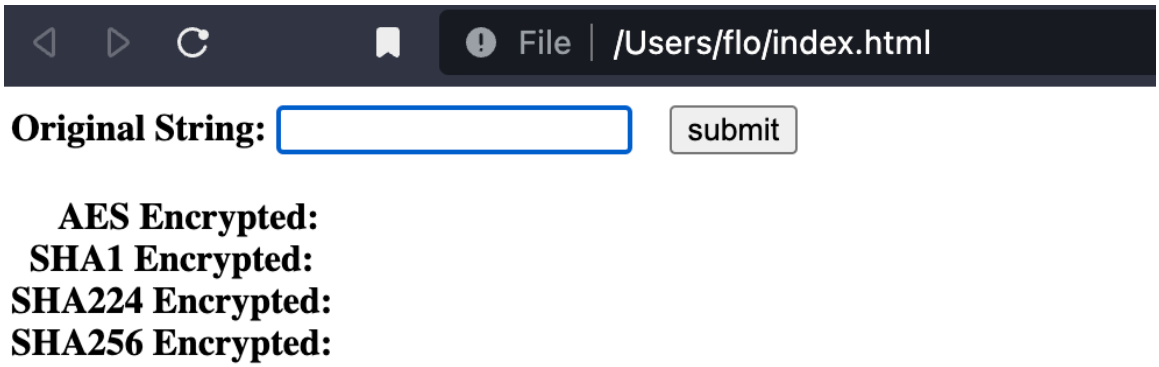
Section 2 (JS)

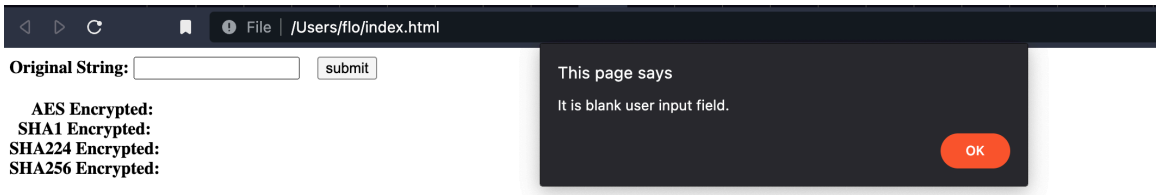
Programming Requirements

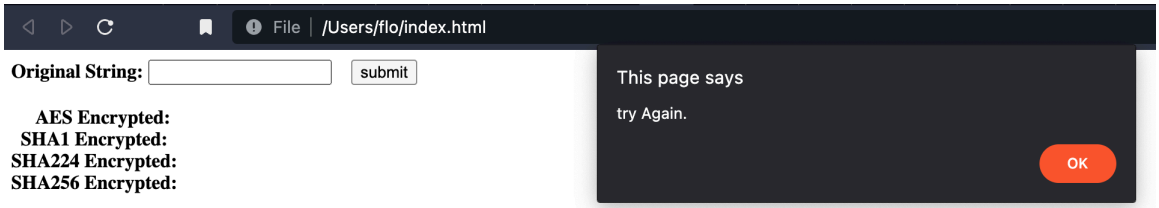
Integrate a JS library into your index.html file to then prompt the user using an input textbox to capture this User's string. Once done, you can start developing some sample code to generate 8 of the following hashes by choose 4 out of them from this list.

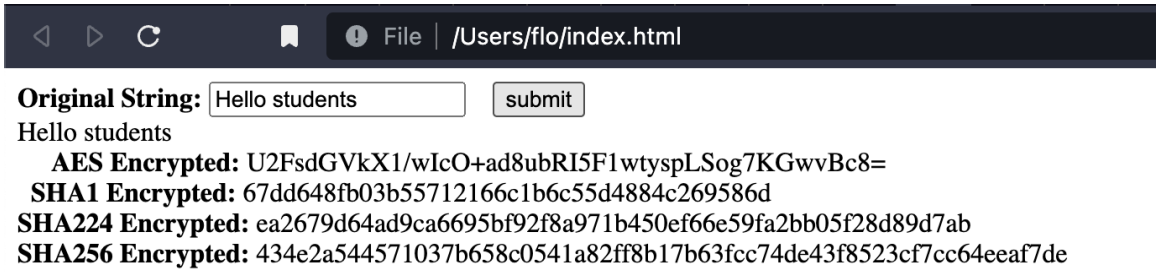


Example of GUI and output:




File | /Users/flo/index.html
Original String: submit
AES Encrypted:
SHA1 Encrypted:
SHA224 Encrypted:
SHA256 Encrypted:
This page says
It is blank user input field.
OK


File | /Users/flo/index.html
Original String: submit
AES Encrypted:
SHA1 Encrypted:
SHA224 Encrypted:
SHA256 Encrypted:
This page says
try Again.
OK


File | /Users/flo/index.html
Original String: Hello students submit
Hello students
AES Encrypted: U2FsdGVkX1/wIcO+ad8ubRI5F1wtyspLSog7KGwvBc8=
SHA1 Encrypted: 67dd648fb03b55712166c1b6c55d4884c269586d
SHA224 Encrypted: ea2679d64ad9ca6695bf92f8a971b450ef66e59fa2bb05f28d89d7ab
SHA256 Encrypted: 434e2a544571037b658c0541a82ff8b17b63fcc74de43f8523cf7cc64eeaf7de

Section 3 (JS coding)

Write a simple Binary to decimal converter using JS via HTML. Must prompt user for an input, check to see if user has entered in a number otherwise inform user to enter in a number. CANNOT use native function to solve this problem !

Section 4 (JS coding)

Write a simple Decimal to binary converter using JS via HTML. Must prompt user for an input, check to see if user has entered in a number otherwise inform user to enter in a number. CANNOT use native function to solve this problem !

Pledge requirements:

As a Highlander, I will conduct myself with honor and integrity at all times. I will not lie, cheat, or steal, nor will I accept the actions of those who do. This program is solely my work, or proper attribution has been given to code that I did not write. If I am found to violate this policy, I realize I will receive an F for this course with no exceptions.

Grading rubric

5pts – HTML file for the following FOUR Functional hashes. Algorithms after display with Inline coding doc, Must have Alert and error handling.
5 pts – AES hashes.
5 pts – SHA 1 hashes.
5 pts – SHA 224 hashes

The two items below, write your own JS functions to complete the following tasks:

25pts – Functional B -> D. Don't use native functions otherwise 0, with Documentations
25 pts – Functional D -> B. Don't use native functions otherwise 0, with Documentations

10 pts – HTML document and content structure. Appropriate use of header, footer, main, article, section (as needed) for B->D and D-> B
10 pts – Must check in your source code to Github with a link provided for Prof to check.
0 pts – Provide proper folder structures and file naming conventions.
10 pts – Provide Screen Shots (minimum 1 for each section) Save them into a word doc.