# MIS2402 Assignment: Health Calculators

In this assignment you will work through a series of instructions that are similar to an MIS2402 exam. The assignment is meant to re-enforce multiple concepts, concluding with an emphasis on form/data validation using jQuery.

## Scenario

You are creating a simple application to help people calculate their body mass index (BMI) and their ideal weight.

Body Mass Index (BMI) is a number commonly used in industries related to medicine, health care and insurance. The BMI can be used to determine if a person is above or below their ideal weight.

## Instructions (together)

Open the zip file provided by your instructor. Copy the health folder in the zip file to your mis2402workspace folder and open up the folder in Visual Studio Code.

When you are done with the assignment, upload your work to a health directory under your wwwroot folder on misdemo.temple.edu.

You are to use the formulas provided in the instructions, and round all results as directed.

## Steps

1. Open index.html in Visual Studio Code and look around. There are functions already defined that you may find to be helpful later in the assignment.

Edit the contents of the footer row. Change the text that says “Created for MIS2402 by Sam Student”. Specify your own name instead of “Sam Student”.

1. There is no title in the head tag of this HTML document. Add a title tag. The content of the title tag should be: BMI and Ideal Weight Calculator
2. There is an external CSS file named bmi.css. Add a link to it so that the page changes appearance.

***Before***

******

***After***



1. Add the “use strict” declaration to the script tag.
2. In the HTML body there are three label tags that are to be used in conjunction with input tags of type text. Add the missing “for” attribute in each label, so that each label is correctly associated with each input tag of type text.
3. In the HTML body tag there are four input tags of type button. Add the missing “id” and “value” attribute in each input. Use the following table. Careful! Spelling and case matter.

|  |  |
| --- | --- |
| Button Id | Button Text |
| calculateBMI | Calculate BMI |
| calculateIdealM | Ideal Weight (men) |
| calculateIdealF | Ideal Weight (women) |
| clear | Clear Entries |

Here’s a picture of how things should look so far:



1. Try the application. Not much works yet. Type in some data. The Clear Entries button does not work. Make this button work by completing ALL of the following steps.
	1. Add a click event handler to the to the document ready() event. The click event should call the clearEntries function.
	2. Add logic to the clearEntries function. Specifically:
		1. Set the value the three input tags to “” (the empty string)
		2. Set the inner html of the message div to “” (the empty string)
2. The Calculate BMI button does not work. Following the logic of the code, you will notice that when the Calculate BMI button is clicked a function named buttonBMI is called. If there are no error conditions, the calculateBMI is then called to perform the calculation.

The code for the calculateBMI function is missing. You need to supply it.

The formula to calculate BMI is as follows:

$BMI=\frac{\left(weight in pounds\right)}{(height in inches)^{2}} ×703$

Make sure the calculateBMI function returns a result rounded to 2 decimal places.

1. The Calculate Ideal Weight buttons do not work. Following the logic of the code, you will notice that when either of the Calculate Ideal Weight buttons is clicked a function named buttonIdeal is called. If there are no error conditions, the calculateIdeal function is then called to perform the calculation.

Note that that the person’s gender and height are to be passed to the calculateIdeal function as parameters.

**CONTINUED**

The calculateIdeal function works, but there is logic missing in buttonIdeal. Supply the missing code so that the ideal weight buttons work. Specifically, you will need to:

* 1. Calculate the person’s height in inches
	2. Calculate the ideal weight for the person’s gender using the calculateIdeal function.
	3. Report the answer to the user by putting text in the message div, with “Ideal Weight = “

***Example***:



Click

* 1. If the results from the **calculateIdeal** function are less than or equal to 0, report “Ideal Weight Unavailable” in the message div.
1. Upload your work. Confirm that you have uploaded all of your work to your wwwroot folder on misdemo.temple.edu.
2. Inthe box below, write the **complete** URL, including protocol, where your work can be viewed:

|  |
| --- |
|  |

## CONTINUED…

## Instructions (on your own)

Now that you have part one done, make a copy of index.html. Name it index2.html.

### Instructions for index2.html

There are three input tags on the Health Calculators form. A person filling out that form may make mistakes or omissions in one or more of those inputs. So, instead of reporting one error at a time in the message div, you can report multiple errors at once in an unordered list. Create a <ul> tag with an id of “errors”. Use jQuery’s append method to populate it.

If you need help imagining what this should look like, see this animated gif:

<https://misdemo.temple.edu/gifs/health2.gif>

Please note, you’ll need to make several adjustments your code to produce the effect in that animation. Most notably, you will need to restructure the “if else if” blocks of code so that each error condition is detected.

### Instructions for index3.html

Once you have index2.html working, make a copy of it and create index3.html. Instead of using an unordered list to report errors, place an HTML <span> tag next to each input box. Use the jQuery next() method to put the error messages into the span next to each problematic input box.

If you need help imagining what this should look like, see this animated gif:

<https://misdemo.temple.edu/gifs/health3.gif>

Please note, in the sample animation the error message text was abbreviated and the CSS width of the label tag was adjusted to 40%. You do not need to exactly reproduce the appearance of the gif. The important thing is that the error messages appear **next to** the relevant input tag.

### Instructions for index4.html

Once you have index3.html working, make a copy of it and create index4.html. Create a function called renderBMITable that will use jQuery append to put table rows into a table with an id of “output”. The table can show the users current weight in bold, as well as the BMI scores that range from 10 lbs above to 10 pounds below the user’s current weight. Use the categorizations described in a previous assignment to indicate what category a BMI score is in (i.e. Underweight, Healthy, Overweight, Obese).

If you need help imagining what this should look like, see this animated gif:

<https://misdemo.temple.edu/gifs/health4.gif>

Here’s the html for the table.

                <table id="output" class="table-bordered">

                </table>

Your script should add in a first row <tr> with <th> cells, followed by the desired number of <tr> tags with normal data cells <td>

**CONTINUED…**

**Upload all your files to the class server in a folder named “health” before the due date / time. This assignment will be graded.**

### How will this assignment be graded?

|  |  |
| --- | --- |
| Item | Point Value |
| index.html – All buttons work. (-5 for minor bugs … no credit for code that does not work.) | 25 |
| Index2.html – form validation with unordered list of messages. All works. (-5 for minor bugs … no credit for code that does not work.) | 25 |
| Index3.html – form validation with messages “next to” input tags. All works. | 25 |
| Index4.html – table generated correctly including the highlighted value. All works.(-5 for minor bugs … no credit for code that does not work.) | 25 |