**level---1(20 points each)**

1. **Name**

For this project you will create two text input elements:

* First Name
* Last Name

When the user clicks the "Get Name" button, a paragraph will appear that reads: Your pig latin name is *first name last name*.

Note: The first name and last name must have a first letter that is capitalized and the rest lowercase. (See the example below.)

In case you do not know how pig latin works, here is the construction:

1. Take the first letter and put it at the end.
2. Add "ay" to the end.

**Example Result:**

* Input: Sydney Hubbard
* Result: Your pig latin name is Ydneysay Ubbardhay.

1. **Cost Calculator**

One of your first assignments was to do a cost calculator. This calculator will figure in a discount code. For this project you will need to create two input elements:

* Price
* Discount Code

**Discount Code:** You will create three possible discount codes, each with different values. For example:

* SAVE20: 20%
* JOLLYDAYS: 15%
* HOLIDAY10: 10%

When the user clicks the "New Price" button, a paragraph will display the new price, factoring in the discount AND a 8% tax. You will also display the amount saved.

**Example Result**

* Input: 24.99 and JOLLYDAYS
* Result: Your Total: $22.80  
  You saved $3.74!

1. **Find Your Generation**

an input that asks a user for their age and a button that will submit the answer. You will print out the year the user was born. Then use a switch statement to get the following results:

* If birthyear is 1946-1964, print *You are a member of the Baby Boomer Generation!*
* If birthyear is 1965-1980, print *You are part of Generation X!*
* If birthyear is 1981-1996, print *You belong in the Y Generation!*
* If birthyear is 1997-2012, print *Generation Z is your generation!*
* If birthyear is 2013-2025, print *You are part of Generation Alpha.*
* Everything else: print *Sorry - your generation is not listed.*

**Example Result**

* Input: 32
* Result: You were born in 1991.*You belong in the Y Generation!*

1. **Mini Calculator**

For this assignment, you will need the following:

* First Number input element
* Second Number input element
* Dropdown box with the following choices:
  + Addition
  + Subtraction
  + Multiplication
  + Division

After the user selects the "Calculate Button," a paragraph will appear with the correct *equation* along with the correct answer.

**Example Result**

* Input: 4, 5 and +
* Result: 4+5=9

1. **Number Guessing Game**

Have your user select a number between 1-100. Your user will have 5 chances. Display their number chances on the screen. For each guess, decrease the number chance and let them know if the number is too low or too high. Display their wins and losses. After they win or lose, use a confirm window to play the game again if the user selects okay with a new random target number.

NOTE: Here is information on a [Confirm window](https://www.w3schools.com/jsref/met_win_confirm.asp)

1. **Regular Expression Replacement**

Using regular expressions, replace all the hollys with jollys, indpendent of case. When a user clicks the "Replace" button, the text will be changed.

* hollylights.html
* jollyOrnaments.html
* ChristmasHollyVillage.html
* jollyHollyWreath.html

**Level-2(30 each)**

1. **Rock, Paper, Scissors Game**

For this game, you will have a button labeled "Play Game." When the user, clicks the button two images will appear: rock, paper, or scissors - one for each player. These will be chosen at random.

After the images appear, a paragraph will read one of the following:

* Player 1 wins! Play again.
* Player 2 wins! Play again.
* It's a tie! Play again.

Create a table that will list the winner by game for each game played.

It case you do not know the rules of rock, paper, scissors:

* Scissors beats paper
* Paper beats rock
* Rocks beats scissors

**Example Result**

|  |  |
| --- | --- |
| **Games** | **Winner** |
| 1 | Player 1 |
| 2 | Player 2 |
| 3 | Player 1 |
| 4 | Player 2 |
| 5 | Player 2 |

1. **Square Foot Grow Box**

For this project, you will have two input fields:

* Width
* Height

When the user clicks the "Calculate" button, a paragraph will appear that reads the answer. (ex. 25 square feet). You will also then have a box appear that illustrates the box.

1 foot = 10 pixels.

So, if the user puts in 6 for height and 8 for width, it would be 60 pixels high and 80 pixels wide.

Each time the user clicks the button, the old box will disappear and the new box will appear.

**Example Result**

* Input: 12 width 7 height
* Result: 84 square feet

1. **Change Calculator**

For this project, you will have one input field that will let users type in the amount of change they have *in coins*. (READ: must be a whole number.) When the user clicks the "Count Change" button, you will decide how much change the user can get with the least amount of coins. You will print out how many dollars, quarters, dimes, nickels, and pennies it would take to equal the amount they entered.

1. **Mortgage Calculator**

For this assignment, you are going to create a simple mortgage calculator that will estimate payments based on the following inputs:

* Loan Amount
* Interest Rate
* Loan Terms (in years)

You will then use a [mortgage payment formula](https://www.mtgprofessor.com/formulas.htm) to find out the monthly payments. A paragraph will appear with the monthly payment listed, along with the other variables.

**Example Result**

* Loan Amount: $250,000
* Loan Term: 30 years
* Interest Rate: 9.3%

Your monthly payment will be $2,066.00

**Level-3(40 each)**

### Coin Flip

For this project, you will have a button that will do a coin flip. You will have an image that illustrates whether the result is heads or tails. You will also display the current ratio of heads to tails.

You will keep track of all the flips in a table:

|  |  |
| --- | --- |
| **Flips** | **Result** |
| 1 | Heads |
| 2 | Heads |
| 3 | Tails |

### Circle Generator with Event Target

For this project, you will have an input field where the user will type in a number between 1-100. Once the user clicks the "Generate" button, the value entered will be the number of circles that appear on the page. The user can click on each circle to change the color and then click back on the same circle to change the color back to the original color.

### Login Page

This page will consist of two sections:

* Login registration
* User Login

## Login Registration

You will create three input fields:

1. username
2. Password
3. Second Password field
4. Register Button

*NOTE: the password fields should be of password type.*

The user will enter their user name and their password twice. The password must be between 8-10 characters. The passwords must be of the correct length and match in order to register the user. If the passwords do not meet the length requirement, a message should appear that reads "Must be between 8-10 characters." If the passwords do not match, a message should appear that reads "The passwords must match." If the user get registered, a message should appear that reads "Thank you, *username* for registering," where username is the username they entered.

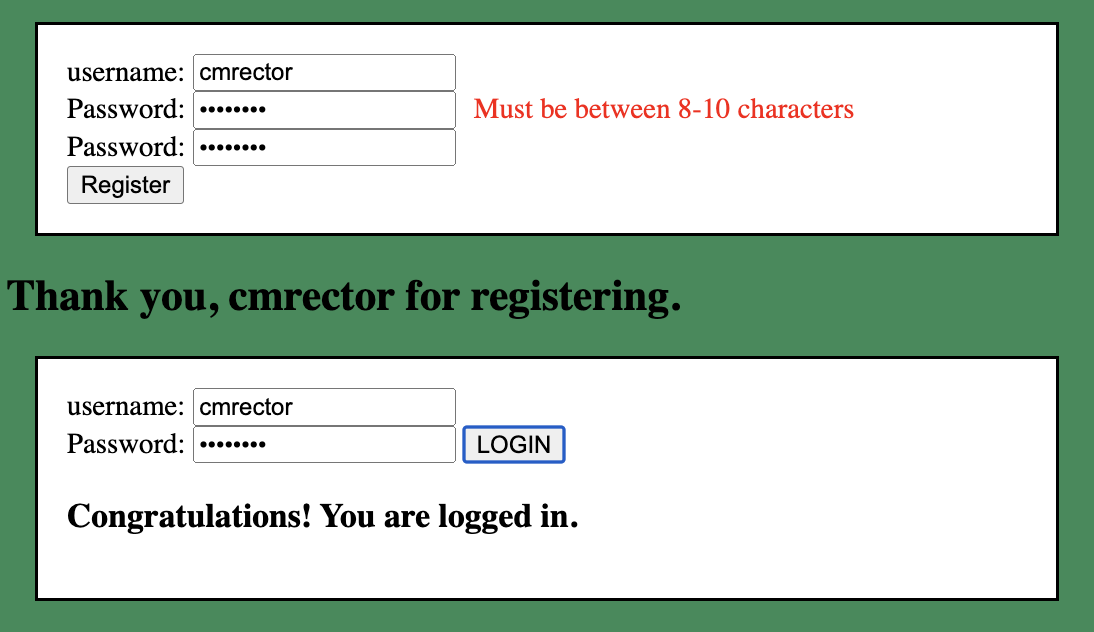
## User Login

You will create three input fields:

1. username
2. Password
3. Login Button

The user will enter their username and password. If it matches what is on file, a sentence should read: "Congratulations! You are logged in." If the information does not match what is on file, a sentence should appear that reads: "Your login information does not match those on file. Please try again."

# Example Result



**Level 4 (50 points each)**

* 1. **Advanced Slide Show**

For this project, you will create a self-running slideshow that has the following functionality:

* Previous button to look at the previous picture
* Next button to look at the next picture
* Pause/Play button that will alter images depending on if it is in play/pause mode.
* A drop down that has different timing options for the self-running slideshow.
* It must have at least 10 pictures.

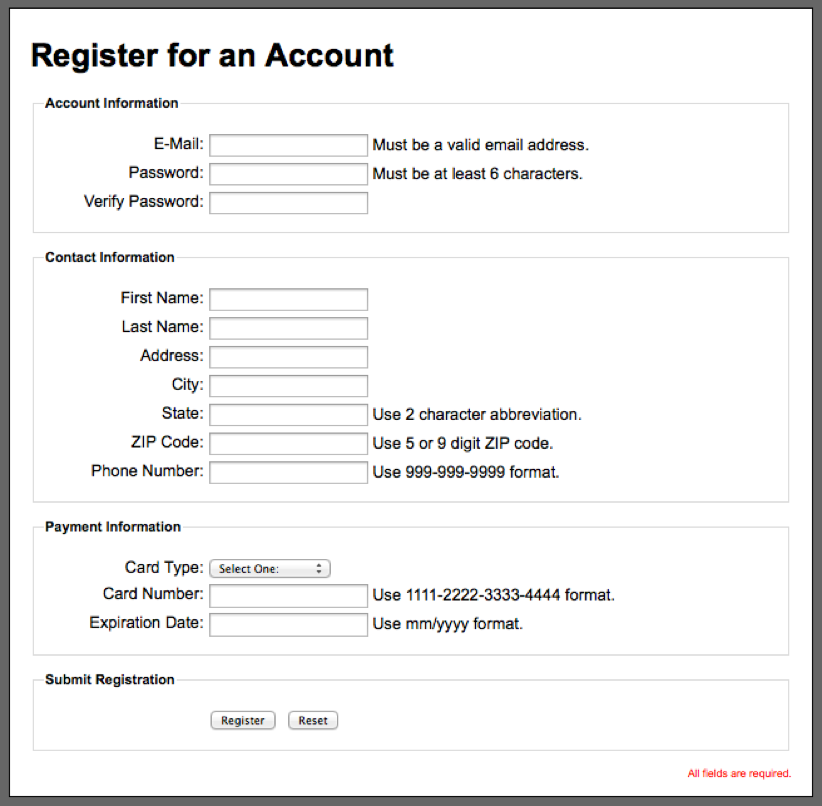
NOTE: This project uses timing functions which we have not talked about in class. You will have to research this on your own to complete this project.

* 1. **Form Validation**

You will create a robust registration form with the following validations:

1. Email must contain @ and a period followed by AT LEAST two characters
2. Password must be between 8 and 10 characters
3. Passwords must match
4. State must be two characters long and must match state abbreviations  
   NOTE: a quick google search for *HTML states dropdown* can result in code for this looong dropdown.
5. Zip code must be 5 numbers long
6. Phone number must include 10 numbers in the following format xxx-xxx-xxxx
7. Credit card must consist of 12 numbers
8. After the user clicks the submit button, all the fields are validated. If there are any errors, the errors will appear next to the respective input field.

The following image is what the form should look like:



* 1. **Grade Finder**

Using the array below, generate a table through javascript so that the table appears like the one below.

let gradingScale = ["A", 90, 100, "B", 80, 89, "C", 70, 79, "D", 60, 69, "F", 59, "lower"];

Create an input where the user enters their point total. (Note: use a placeholder that indicates the max points possible is 500.)

Create a button that reads *Your Grade*. Once the user clicks on that button, two things will happen:

* The user's grade percentage will appear on the page.
* The letter grade line of their percentage will be highlighted in a color.
* When the user enters a new value, the colored line will return to normal and the new grade line will be highlighted.

**Example Result**

* Input: 432
* Result: Grade Average: 86.40%

|  |  |
| --- | --- |
| **Grading Scale** | |
| A | 90-100 |
| B | 80-89 |
| C | 70-79 |
| D | 60-69 |
| F | 59-lower |