

Working with and learning HTML, CSS, and JavaScript for the front end of the web platform was almost trivial considering that I have spent time with some of these programming languages in the past. But still, the process to get re-familiarized with the languages takes time. For instance, when trying to create an appealing [login](#) form using JavaScript, similar to the one in the link, I ran into problems such as, the form would always appear on the right side of the screen instead of popping up onto the screen when the login button was clicked. At first we have thrown around the idea to use a HTML 5 themed template for our application to save time on the web's front end, however, due to some minor technical difficulties that we dealt with during our live demo presentation – where it was impossible to make the header's text readable – we had to scratch the original design and come up with a new solution. In came Bootstrap, a web framework used to create web projects; now instead of given all the aesthetic parts of the original website, most of that would have to be created from scratch. With a little magic of CSS our most current webpage looks just as appealing as our original design.

Credit Gael with suggesting the use of an image slider template for the website's main homepage as it creates a scroll-free webpage, which eliminates all the hassle of endless scrolling. This neat template was as bland as it comes, no offense to Bootstrap, but by adding pictures and content onto the homepage with the power of CSS we have created something out of nothing. It is known that reading text on top of images is difficult, especially white colored text with a light background image, same thing goes with black colored text on a dark background image. To preserve the entire image, while still being able to read the text, one solution was to create a transparent background layer on top of the background image in order for the text to become legible for the audience. This was done through HTML and CSS, by creating a class – carousel-caption – where each slide had illegible text, changing the opacity of the background helped preserve the image to a certain degree while still being able to read the text. On top of that, in order to make each slide's transparent background layer proportional, the amount of text on a slide determines the size of the transparent background layer, a lot of testing went on to determine the height of each background layer as well as the text's margins. The *Meet the Team* webpage was all done from scratch. Since the majority of the team members loved the original *Meet the Team* webpage using a HTML 5 template, I decided to somewhat emulate that design aspect of it and try to make it come to existence. Creating the green side panel was time consuming since obtaining the length of the panel to get it matched up to the content was as difficult as it comes. However, with a little help from Google and Stack Overflow, I managed to come up with a working solution. One of the characteristics of the original team page I liked was the shaping of the image and having a full view pop-up image once the picture was clicked. However, while dedicating time in attempting to make such a pop-up image for the current team page the final product wasn't an ideal fit, so we decided it was best to create a rounded border image frame instead. Creating a user-friendly interface was our plan going in to the web design to attract users in order to give our fitness application more oomph. However, creating the webpage was only a small piece of the entire puzzle.

In order to make a fitness application have all its functionalities working, the main part of the application comes from having an array of exercises for a user to choose from. In order for us to do this we picked out hundreds of workout exercises covering just about every muscle in the body from [www.bodybuilding.com](http://www.bodybuilding.com). Then we created a csv file which had categories ranging from exercise name and level of difficulty to targeted muscles and necessary equipment needed. By creating this log, we can provide users a plethora of exercises to choose from when they are at the gym. By creating a different set of keys – one for each exercise – we are able to easily identify the key to a corresponding exercise. The following is a full list of categories contained in our exercise data file:

- Key: A unique set of characters correlating to each exercise; used to gather information as quick as possible
- Exercise Name: Name of exercise
- Exercise Difficulty: Level of difficulty ranging from beginner, intermediate, to expert
- Exercise Type: Type of exercise (Stretches, Strength, Plyometric, Powerlifting, Olympic Lifting)
- Worked Muscle: Targeted muscle
- Exercise Equipment: Necessary equipment needed for exercise

However, we understand that there may be some exercises that are too difficult for some users to attempt, a leveling system was put in place to prevent injuries from occurring. See below for a brief explanation on how some of the attribute in our application works:

Leveling system – At first, the leveling system was implemented as a means to have fun while at the gym or at other various types of fitness centers. Then we realized that while there are some extremely difficult exercises to perform, the leveling system can offset this issue by having exercises locked until a user enters a specific level. In brief, the leveling system calculates your level of strength based on callisthenic exercises performed (i.e. push-ups, sit-ups, pull-ups etc.).

Injury Data – A idea suggested by Syed, we made this come to existence by researching the most common injuries related to all the muscle groups correlating to the exercises in our exercise database. Since exercising and injuries go hand-in-hand, this attribute in our application helps remove a particular exercise relating to an injury to a workout program. This was one of the more difficult tasks to complete in the project because the list of injuries to certain muscles is endless, which is why we chose the 1-2 most common injuries. The algorithm works as follows: first, input the injury you are suffering from. Next, we match the injury to the ones in our injury database. Then, if the muscle related to that injury is found, search for the targeted muscle in the exercise csv file and 'hide' those exercises from the user. If the injury is not found, then it is suggested that the user do not irritate the injured muscle any further by ending the exercise routine and if necessary, should consider visiting a medical professional.