Multimedia Systems

CA2 – Password Game

April 21, 2018

|  |  |  |  |
| --- | --- | --- | --- |
|  | N00143888 |  | Aaron O’Hare |

Table of Contents

[Introduction – Aims 3](#_Toc512186992)

[Interface Design Issues 3](#_Toc512186993)

[Implementation 3](#_Toc512186994)

[Checking Passwords 3](#_Toc512186995)

[Regex 4](#_Toc512186996)

[Progressbar.js 5](#_Toc512186997)

[Generating Passwords 6](#_Toc512186998)

[Hosting 6](#_Toc512186999)

[Materialize CSS 6](#_Toc512187000)

[Summary 6](#_Toc512187001)

[Screenshots 7](#_Toc512187002)

[References 9](#_Toc512187003)

# Introduction – Aims

This project involved building a JavaScript based password game. The aim of this project was to provide a simple way for user’s to check the security of their passwords, provide them with the ability to improve their passwords and to allow them to generate new passwords through the application.

The application is a web-based application – due to the requirements of the continuous assessment. It is hosted online via getforge (<http://passwordcheck.getforge.io/>).

# Interface Design Issues

There are no outstanding design issues although the application is not 100% mobile optimized.

# Implementation

## Checking Passwords

The core function of this application is to verify the security of the users’ password. The first step in this process was to build the homepage.

The home page is made up of a number of elements: The navigation, input field, buttons, modal window and the ‘results’ div which is populated by the getPw() function within checkPassword.js. This function is called when the ‘check’ button is clicked.

<button *id=*"send" *onclick=*"getPw()" *class=*"btn waves-effect waves-light" *name=*"action">Check  
 <i *class=*"material-icons right">send</i>  
</button>

The getPw function is then executed. This function starts by setting the InnerHTML of the ‘results’ div to the required layout. This creates the card layout used to display the charts. There are three cards: Numbers, Symbols, Length and Total. Each one follows the same layout.

<div class="col l3">\n' +  
' <div class="card">\n' +  
' <div class="card-content">\n' +  
' <span id ="ctitle" class="card-title">Symbols</span>\n' +  
' <div class="chart" id="symcircle"></div>\n' +  
' </div>\n' +  
' </div>\n' +  
' </div>\n'

The users’ input is then retrieved – this is the password which will be analysed.

*let* pw = document.getElementById('pwInput').value;

### Regex

The password is then analysed using regular expressions. These expressions are designed specifically to extract required information from the user input.

\d will extract all numbers from the string.

*let* numsregex = pw.match(/\d/g);

The results of this regex are then counted by getting the length of the array created.

*if* (numsregex) {  
 nums = numsregex.length;  
}

This same pattern was followed to extract symbols, with the regex for non language characters being \W:

*let* symaregex = pw.match(/\W/g);

### Progressbar.js

Progress Bar is a JavaScript library used to create responsive progress bars for the web.

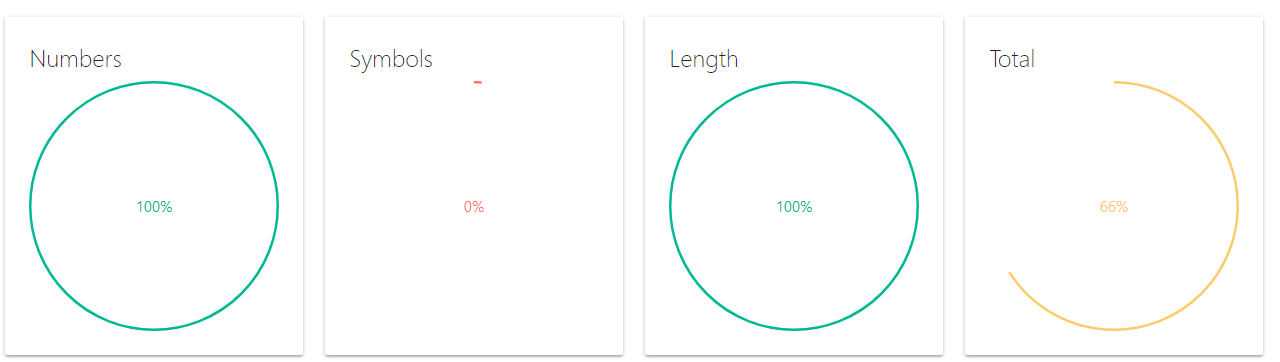
You need to specify the shape, html div id, colour, duration of animation and the method of animation (easing).

The color is determined based on the result of the regex – red indicating a low score, yellow medium and green high.

**var** totalchart = **new** ProgressBar.Circle('#tcircle', {  
 color: tcol,  
 duration: 2000,  
 easing: 'easeInOut'  
});

You can then animate the chart to a defined value (between 0 and 1), at this point I also added the value text to the chart.

totalchart.animate(percentagescore / 100);  
  
totalchart.setText(percentagescore + '%');



## Generating Passwords

## Hosting

The application is hosted using getforge. This service allows for simplified and fee hosting of HTML/CSS/JavaScript based projects. I simply upload the compressed project to their server via the web interface and my project is accessible at <http://passwordcheck.getforge.io/>.

## Materialize CSS

Materialize CSS is a CSS library. It provides a number of pre-styled elements that can be incorporated into existing projects. This simplifies the layout/design process and provides the application with a clean, consistent style.

Materialize also provides a grid system which allows the developer to split the webpage into columns. This aids the design process for multiple screen sizes.

# Summary

The final product allows the user to check the quality of their password. It provides visual feedback on the makeup of the password, including the inclusion of numbers, symbols and the length of the password. Each of these categories are scored and graphed and the user receives a total score. The user may also generate new passwords, choosing between a more secure password for use in a password manager or an easier to remember password.

The project was built using HTML, CSS and JavaScript. Progressbar.js was used to construct the graphs, random-words (<https://github.com/punkave/random-words>) was used to provide the array of words used in the simpler passwords, Materialize CSS was used to style and layout the project and getforge was used to host the project online.

# Screenshots

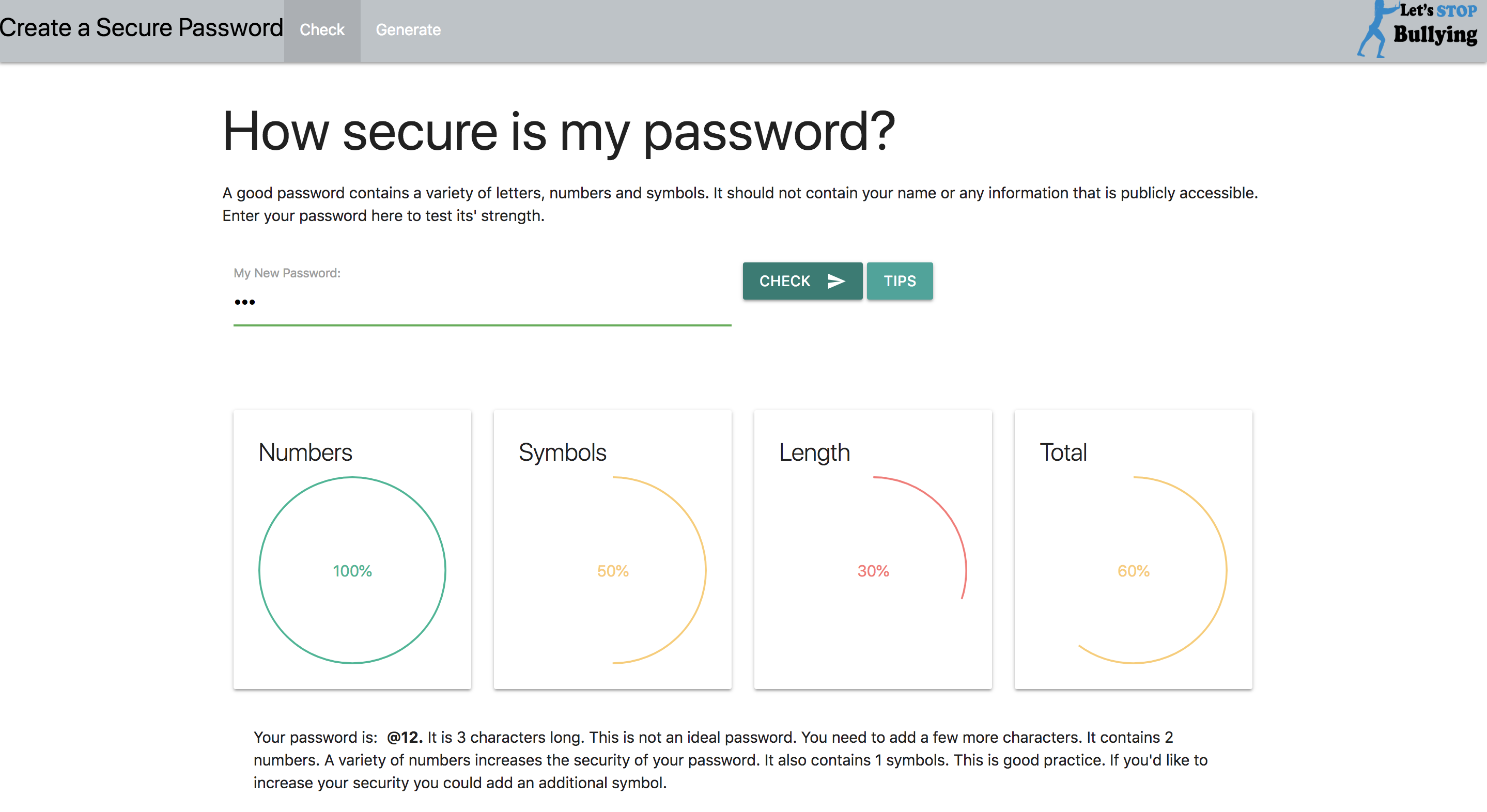


Figure 1 – Check Password

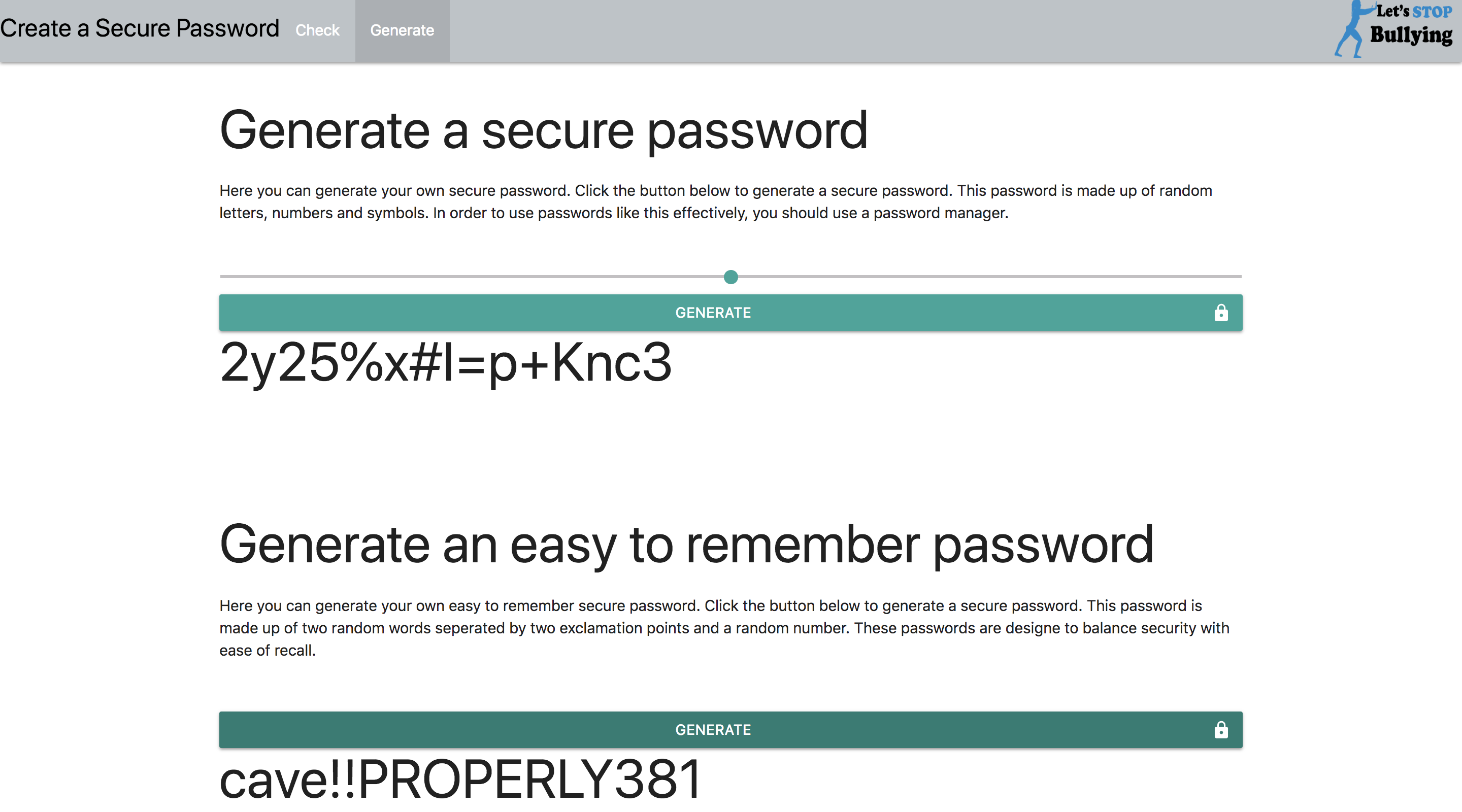


Figure 2 - Generate Password

# References