# Design of “A Very Random Quiz” Website

## Introduction

“The Very Random Quiz” website is a trivia quiz. It consists of five questions varying in topic and level of difficulty. If the user answers at least four questions correctly, they are rewarded with a dad joke. If the user answers at least three incorrectly, they are punished with an inspirational quote.

## Goals

* Simply layout that is thematically consistent
* Adequately loads a random question every time a button is pressed
* Adequately displays a random inspirational quote if at least three answers are incorrect
* Adequately displays a random dad joke if at least four answers are correct
* Responsive on various devices

## Choice of APIs

The website uses three APIs: a trivia API, a dad joke API and an inspirational quote API. The purpose of the website is to function as a quiz game. The trivia API provides random questions and multiple answers. Implementing this API allows the use to have a varied quizzing experience. To make this quiz different, the site displays different content depending if the user passed or failed the quiz. The dad joke API and the inspirational quote API supply a random dad joke and a random inspirational quote respectively.

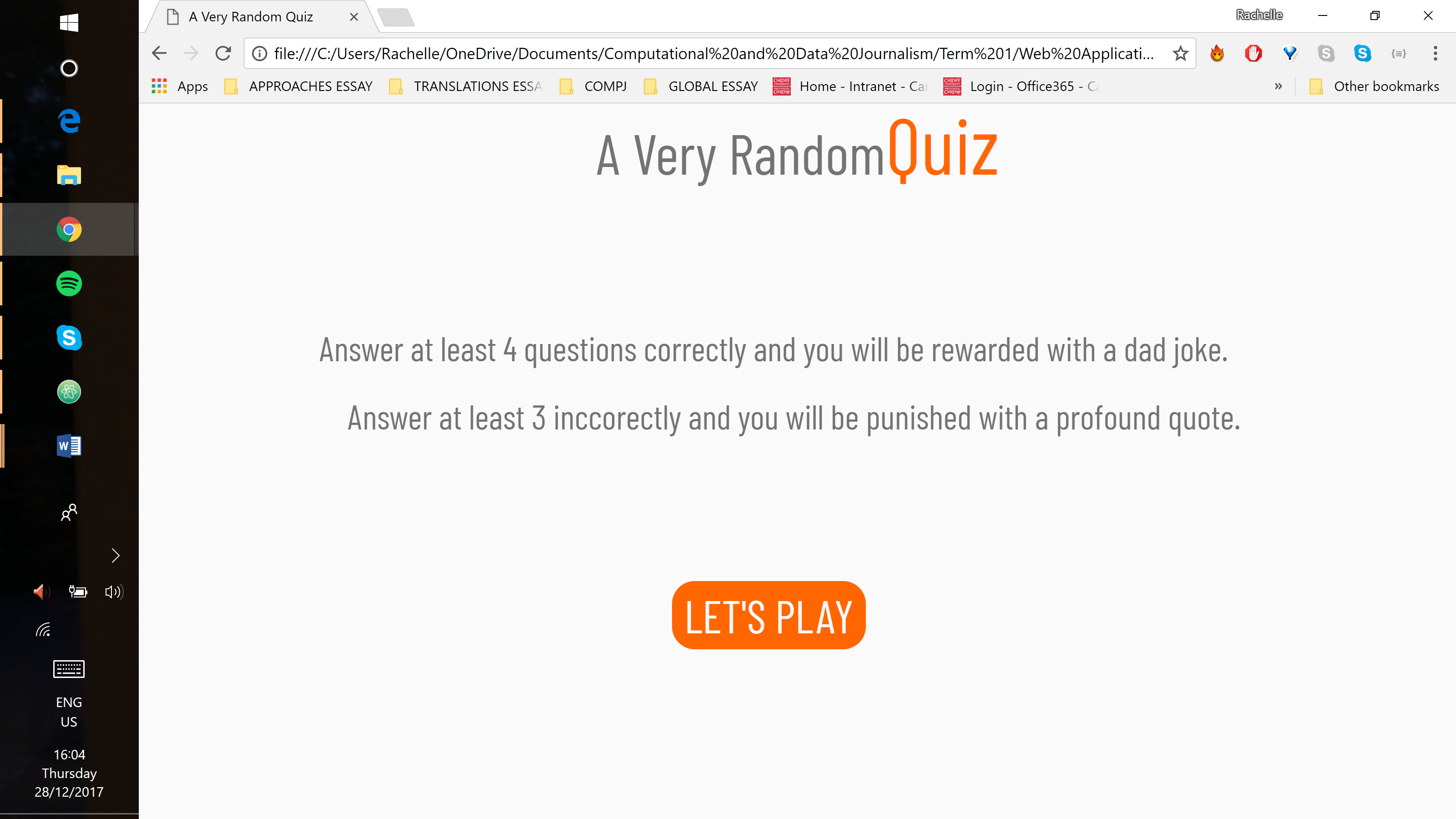
## Design Theme

All pages share the same colour scheme: a cream background, grey fonts and orange accents (psychologically proven to stimulate fun). All of the text was the same sans-serif font in two different sizes. The title of page “The Very Random Quiz” is styled in the same grey font. The word Quiz is a hyperlink that leads back to the ‘home display’. It is orange and slightly bigger than the rest of the words, for emphasis. The overall aesthetic of the website follows the same principles: simple to put all of the focus on the game.

The original design consisted of three HTML pages, but with a few alterations, the design was simplified to just one HTML page. This simplifies the code, but also eliminates the issue of remembering changing data (i.e. the score) on a newly loaded HTML page. The website has four displays: ‘Home Display’, ‘Quiz Display’, and ‘Results Display’. Structurally, the design of these displays is the same and any content changes were executed primarily through JavaScript. The basic design consists of one flexbox with four divs: the ‘title div’, the ‘answer div’ and the ‘button div’. The content of ‘title div’ is the only content that is consistent on all of the displays, so it is the only one written into the HTML. The rest of the divs are empty as the content changes with each display. If the content is not supplied by the APIs, it is implemented using JavaScript.

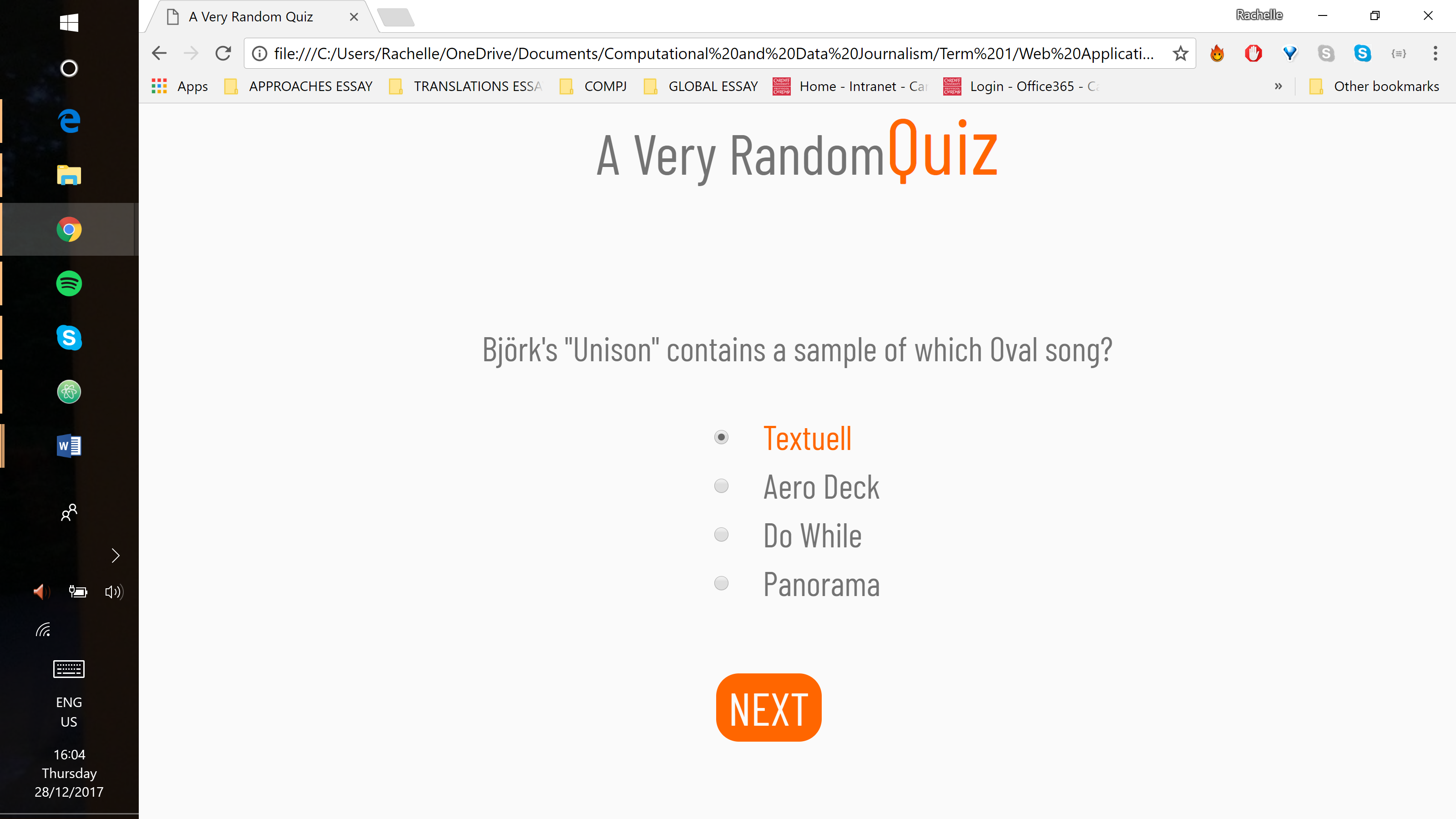
### Home Display

This display is the first the user sees. The title appears across the top, the instructions are in the ‘question div’. The answer div is empty and shrunk so as to avoid a scroll bar and to eliminate any empty space. The instructions are followed by a button that says, “Let’s Play”. As all of this content only appears on the ‘home display’, it was written into the JavaScript file. When the button is clicked, the trivia API will load and the page will switch to the ‘quiz display’.

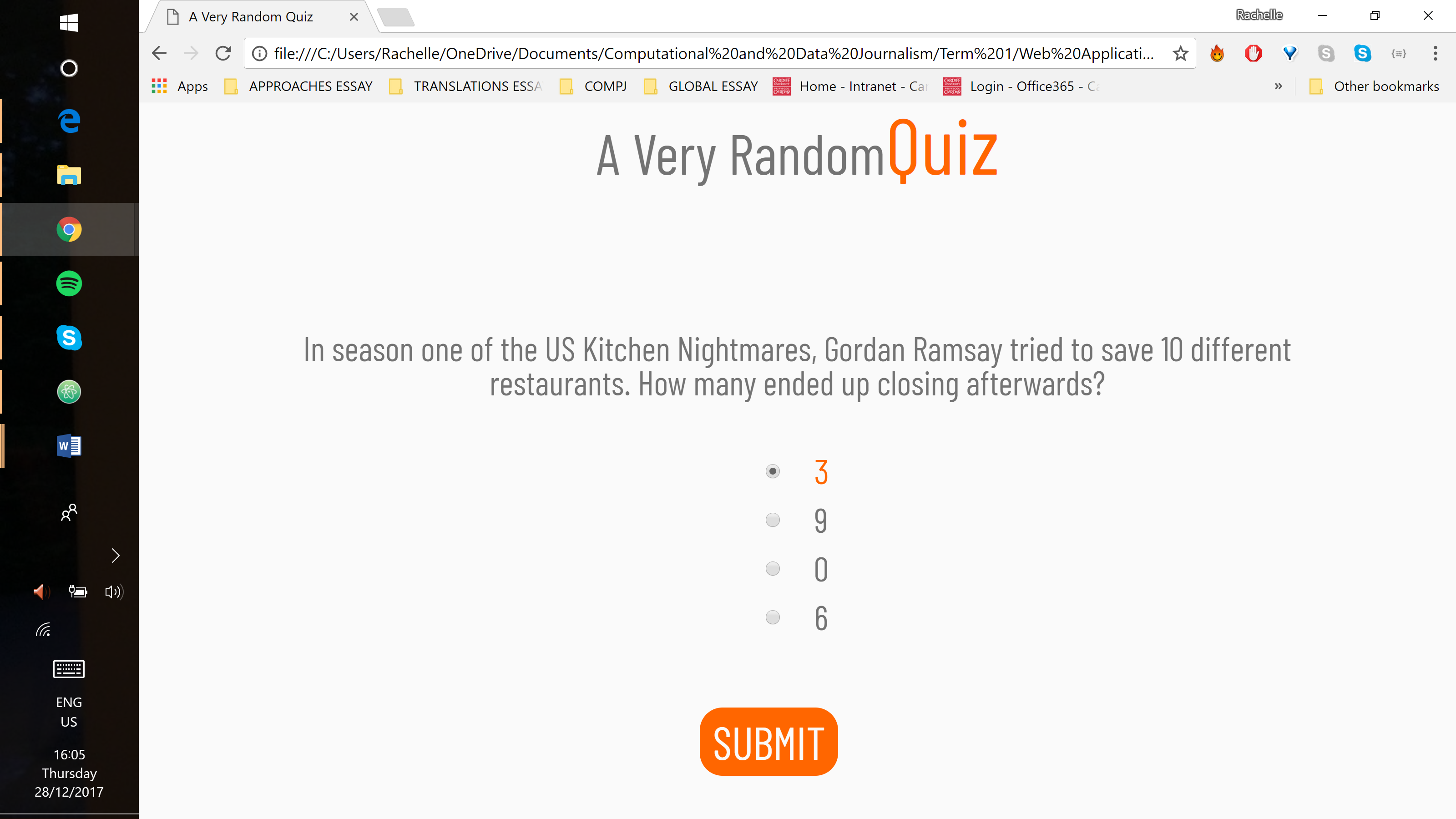


### Quiz Display

The title remains at the top of the page as in the ‘home display’. The instructions in the ‘question div’ are replaced with the question loaded from the trivia API. The API also supply the possible answers which fill the ‘answer div’. They are structured with radio buttons and the selected answer turns orange. At the bottom, the button now says ‘NEXT’. This display remains for four of the five questions.



On the last question, the button changes to ‘SUBMIT’. Depending on the user’s score, clicking the button will load the appropriate API and the page will switch to the ‘results display’.



### Results Display

The title remains at the top of the page. The question is replaced with wither an inspirational quote or a dad joke, depending on the user’s score. The button changes to either: “You suck. Play Again?” or “Good Job! Play Again?” It is moved up as the ‘answer div’ shrinks when empty. The button redirects the user to the ‘quiz display’ rather than the ‘home display’.

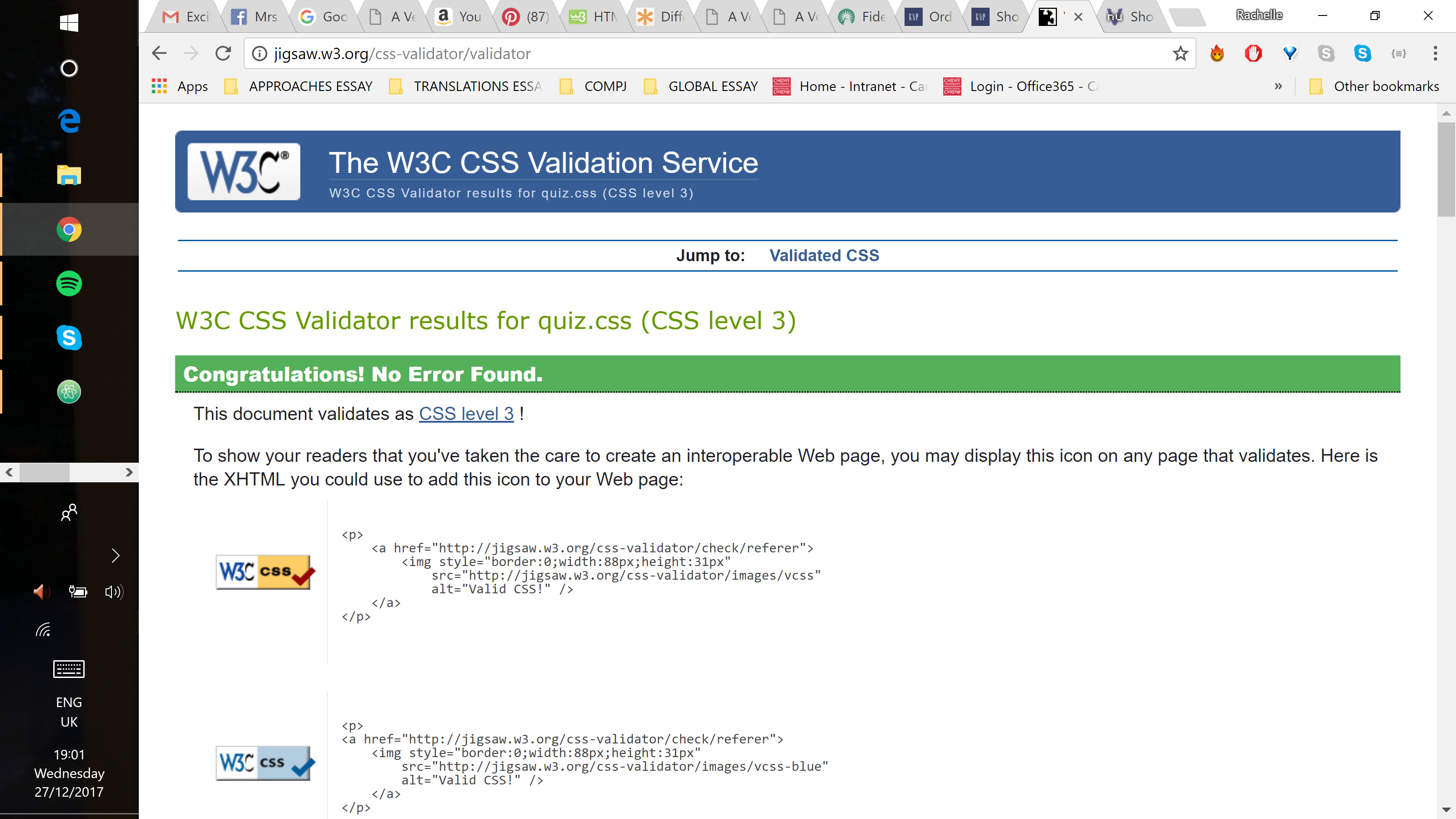
## Validation

### HTML Validation

There are no mistakes in the HTML code.

### CSS Validation

There are no mistakes in the CSS code.

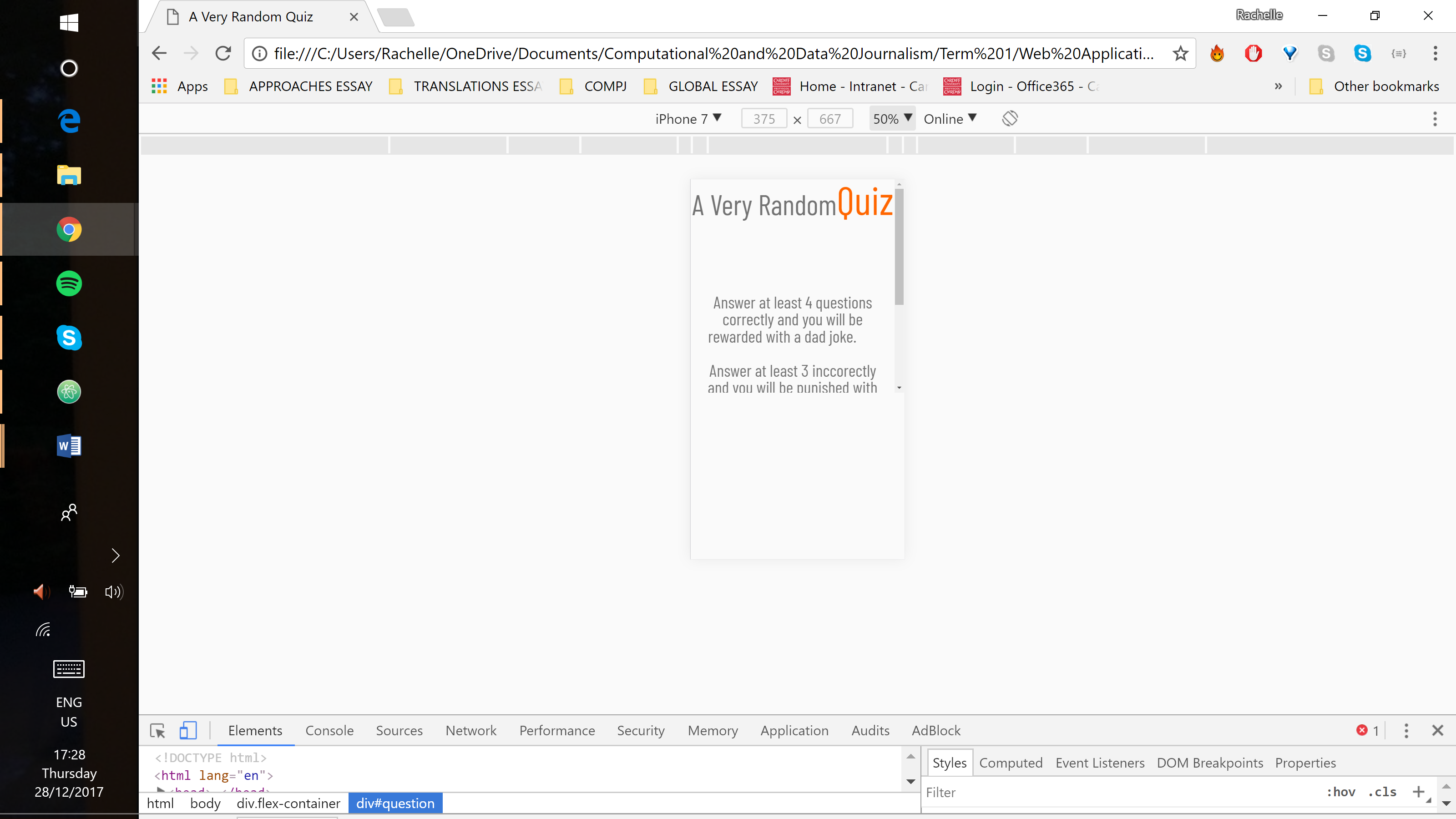
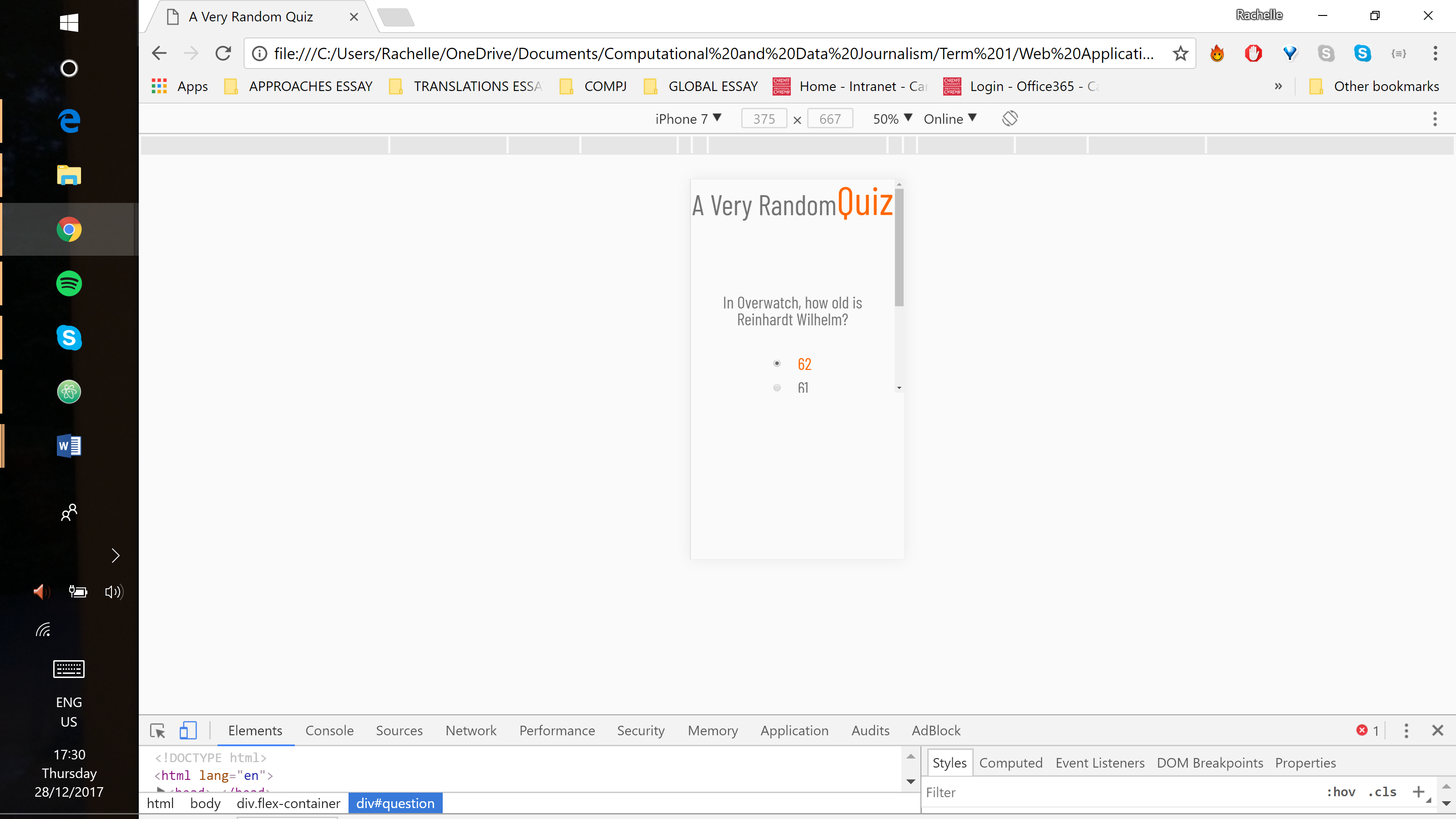


## Evaluation

The website consists of one HTML page styled and positioned with CSS elements. JavaScript was used to implement APIs and to add interactivity to the website. The website demonstrates the ability to access web APIs and to retrieve, manipulate and display data. Browser debugger tools were implemented to better understand the performance and the execution of code in the browser. The website also demonstrates the ability to assess the role of web frameworks in web application development.

The site displays and operates exactly as intended both on desktop and on mobile devices. Media queries were not used since the design is vertical and relatively simple. Any necessary alterations for mobile devices could be made in the main css without impacting the desktop display.

The only issue on mobile devices is that the mobile testing in the Chrome browser would only display half of the screen, so it creates a scroll bar. Assuming that on an actual mobile device the full height will be visible, this was assumed to be a browser issue. This issue is demonstrated in the images below.

Additionally, the title will occasionally display on two lines rather than in one line. As this happens very infrequently, with no change to the code, it is presumed to be a browser issue.

## Conclusion

This website demonstrates an improved knowledge in HTML and CSS and a growing knowledge of JavaScript. Finding and troubleshooting bugs in HTML and CSS is good. The ability to find and analyse bugs in JavaScript has improved, but finding a solution still poses some difficulty. Frequently, too much time was spent trying to fix a problem without full understanding of how to do it or what to look up to understand better. The time between understanding a problem and fixing it would be sped up with a better understanding of how JavaScript works.