

# Challenge 19 - JavaScript

For this challenge, you are going to make a quiz.

Along with this document, you should have a video of how the application should work from a user point of view, so before you proceed, please take a look at it.

## General guidelines:

Make an index.html and a main.js file, and put all your logic in the main.js file. For styling, you are encouraged to use bootstrap. As you have probably seen in the video, there are some smooth effects for showing the content on the screen, so to achieve that effect you can use the following library:

```
<link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.5.2/animate.min.css">
```

More info about the library [here](https://animate.style/).

Or feel free to make you own custom effects with css.

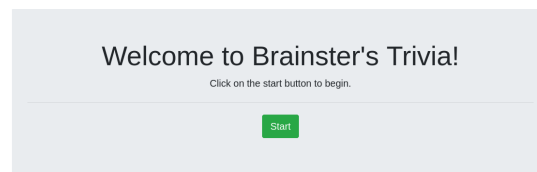
Another thing you will need is to fetch the questions for the quiz from the following api: <https://opentdb.com/api.php?amount=20>

Note: This api will give you 20 questions, the quiz should end after the 20th question.

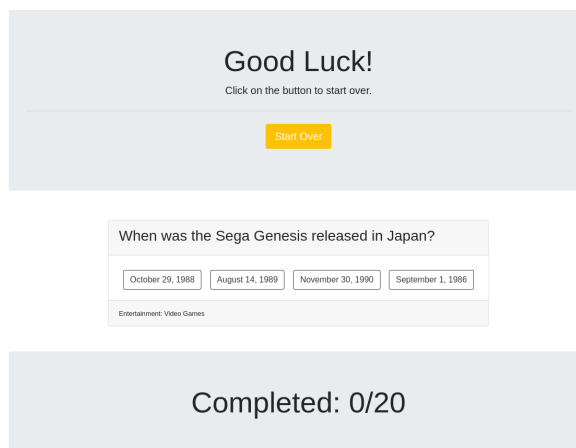
More info about the api [here](https://opentdb.com/).

## Challenge Objectives:

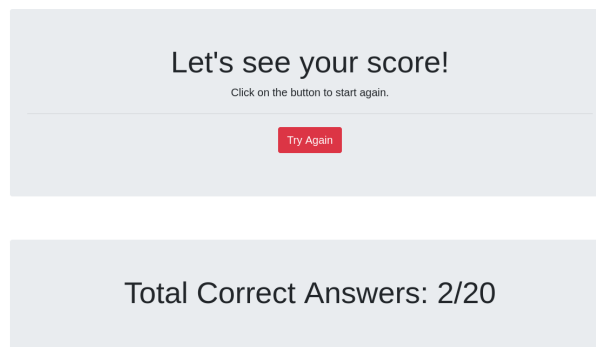
1. Make the application's skeleton with html and css/bootstrap. First show the loading screen and hide everything else
2. Make an api call to the provided address to fetch the questions, once the api returns a response, hide the loading screen and show the start screen



3. When the user clicks on the start button, append to the url a string with a hash "#", for example: index.html#question-1
4. Using the hashchange event, change the content of the screen once again and show the user the first question you received from the api, along with the progress bar below the question



5. When the user clicks on any of the suggested answers for the question, change the string after the hash to correlate to the next question, for example #question-2, and thus display the second question, and so on...
6. You also need to keep track of the number of correct answers the user has selected. For this purpose, you need to create some kind of a counter that you will increment every time the user answers a question correct and keep that information in local storage
7. Once the user answers all questions, change the content of the screen and show them the amount of correct answers they got.



8. If the user clicks on Start Over or Try Again, reload the page and clear the local storage

Deadline:

**Two** weeks after its presentation, 23:59h end of the day.