Frontend Assessment - Word Scrambler

The goal of this assessment is to replicate the frontend application below (as close as possible). You are allowed to use any frontend framework (React.js, Vue.js, etc.) or use plain Javascript, HTML, and CSS. We recommend using a frontend framework as the later parts of the assessment will be very difficult without it.

If you notice something is not working (like the API, or any of the links in this document), please contact hello@hatchways.io.

This assessment will be evaluated based on the following criteria:

- **Correctness**: Is your solution complete and does it pass different test cases?
- **Code Organization, Readability, & Maintainability**: Is your code easy to read and well organized?
- **Code Performance**: Is your code efficient? Did you use appropriate data structures?
- **Best Practices**: Did you utilize good programming practices (write unit tests, avoid anti-patterns)? Did you show a good grasp of your language/framework of choice?
- **Completion speed**: A fast completion time comparable to the completeness of your solution. This is the least important criteria.

We use the following rubric to evaluate your submission. **Please note that if your submission does not attempt to complete all of the requirements, we will be unable to provide feedback on it.**

This is what the completed assessment should look like. Below, we will outline the steps to take to reach this level of completion. You only need to submit the final, completed assessment.

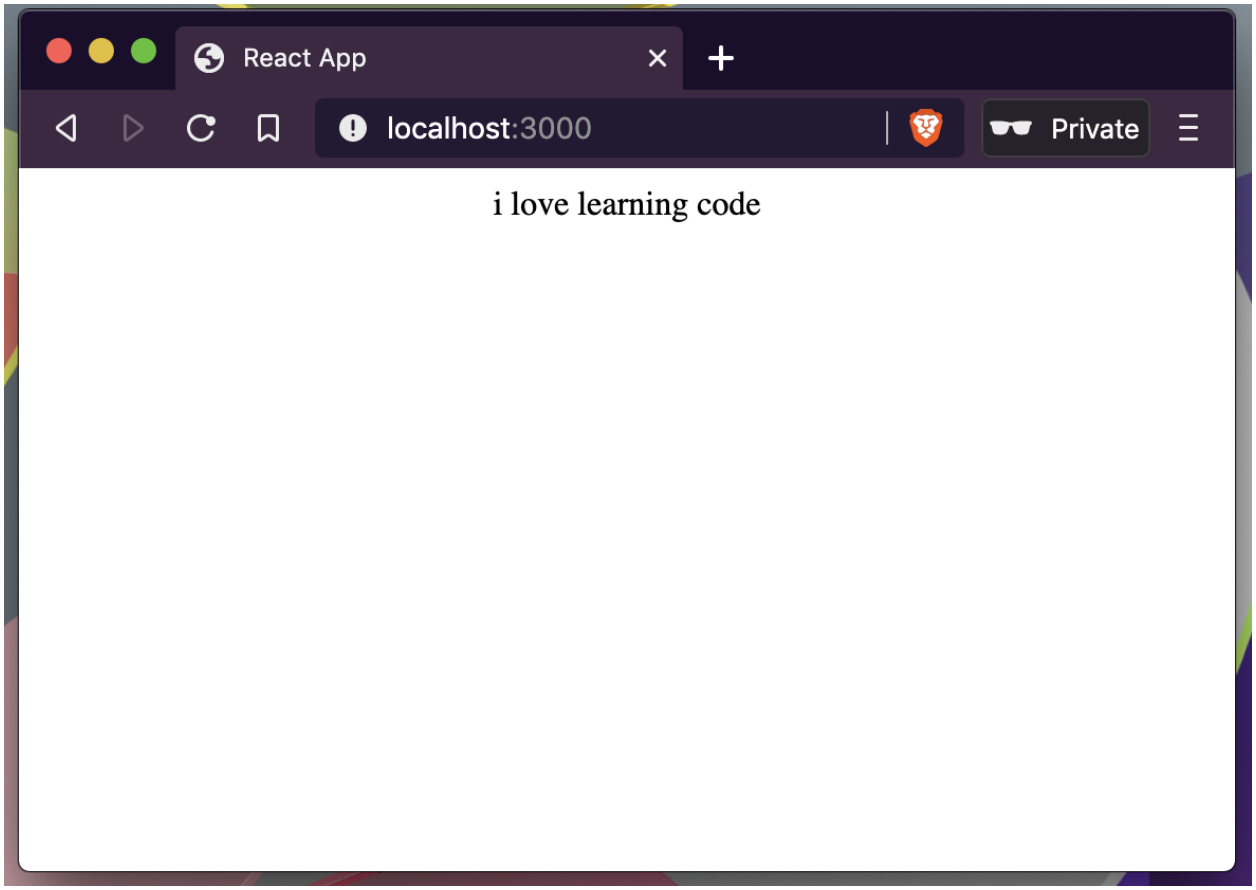
Part 1: Fetch Data

The first step of the assignment is to fetch data from this public JSON API, and present the information on the screen. The route to fetch the data is:

> method: **GET**
> url: https://api.hatchways.io/assessment/sentences/:counter

Replace :counter with 1 (https://api.hatchways.io/assessment/sentences/1) to get the first sentence to begin with.

You do not need any credentials to access the URL above. The goal of this part is to present the data (sentence) on the screen like below. Wrap this text in an element with id “scrambled-word”



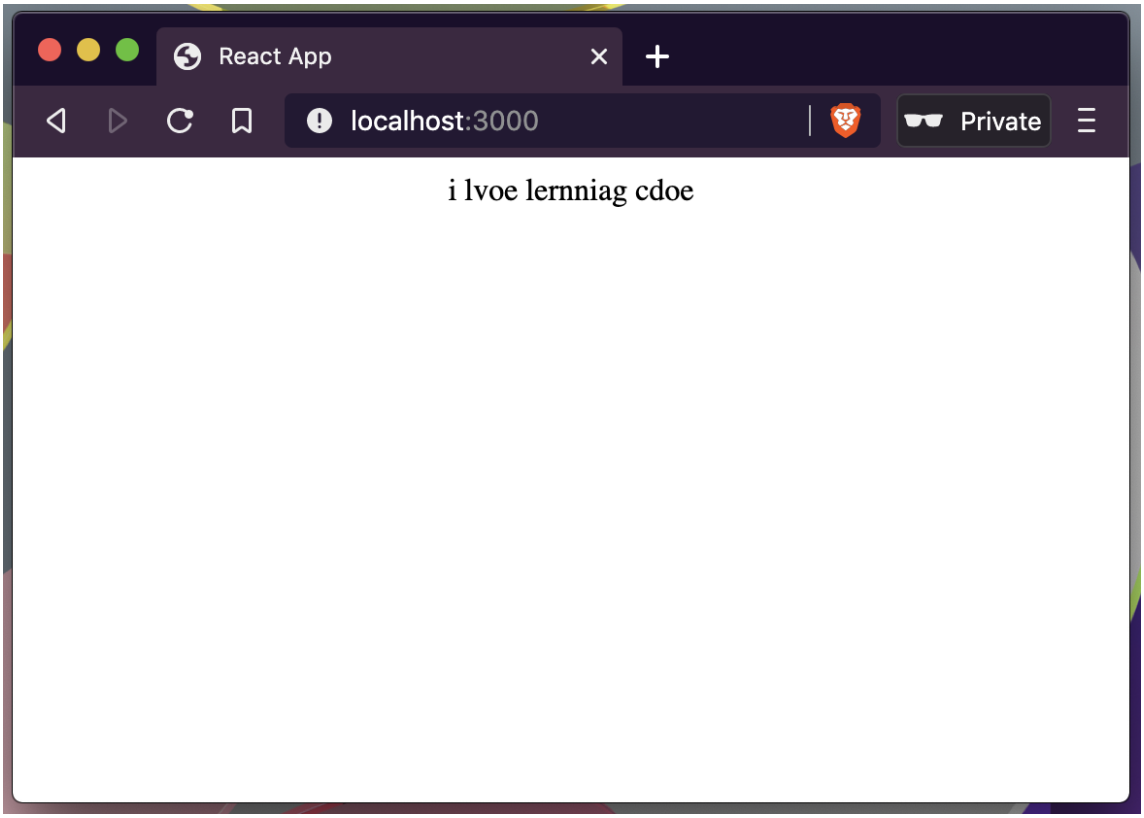
Displaying the sentence

Part 2: Scramble

The second step of the assignment is to “scramble” the sentence and display the results.

- We only want to “scramble” each word of the sentence, so the order of the words remain the same
- Each word will retain the position of the first and last letter, and the middle letters will be scrambled. For example, if the original word is “animal”, it should randomize the word to something like “aainml”
- If the word is only one or two characters, it remains unchanged
- At most, each sentence will have 5 words
- At most, each word will have 15 characters
- There will never be more than one space in between each word

Below is a picture of what it may look like for the original sentence “i love learning code”:



Scrambled sentence

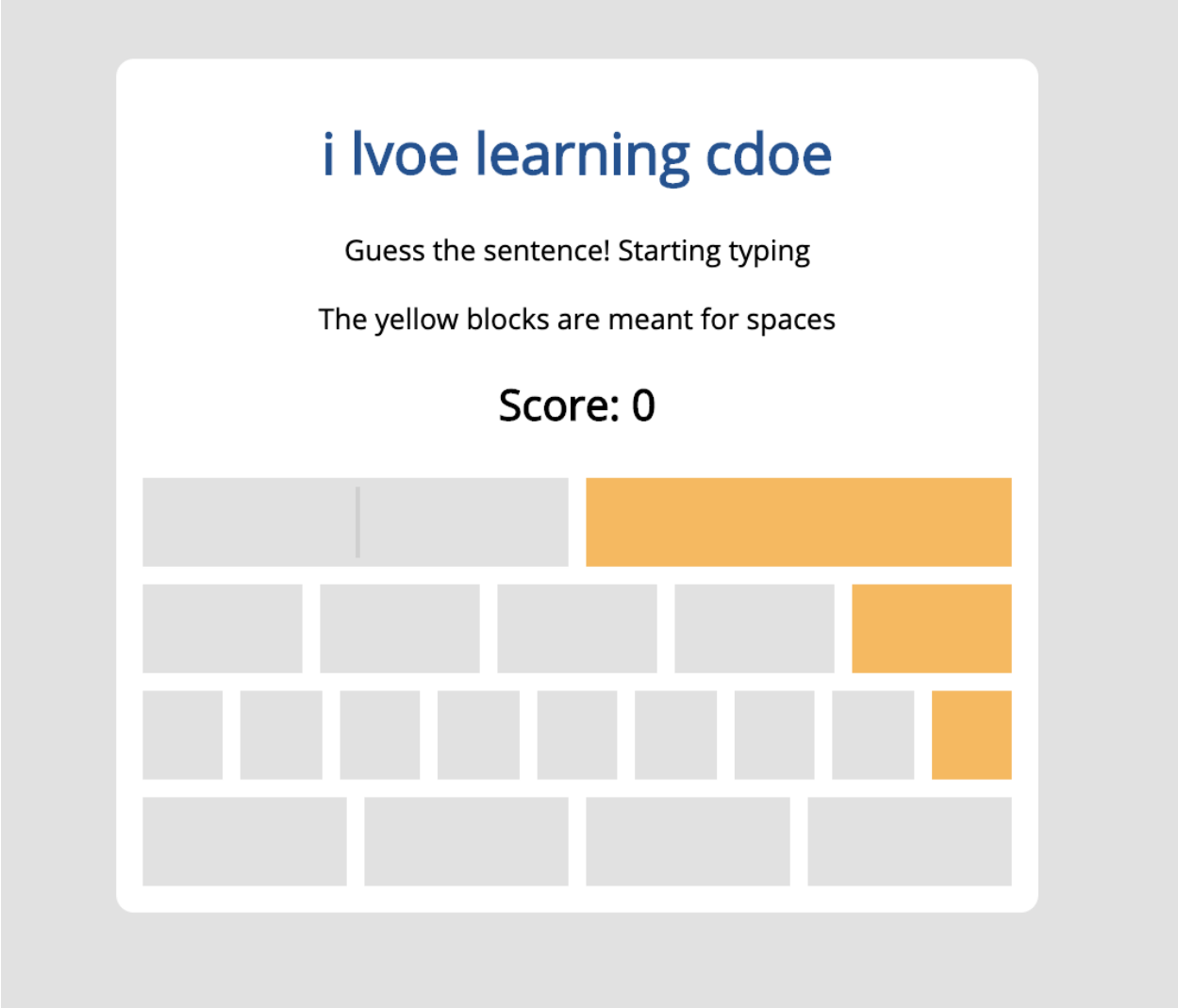
Part 3: Styling

Now it's time to start styling this assignment. Replicate the style of the image below as best as you can. There are 5 main parts:

1. The white container that holds the main content
2. The "scrambled" sentence
3. Explanatory text "Guess the sentence! Start typing. The yellow blocks are meant for spaces"
4. Score text: "Score: 0"
5. The number of guessing blocks depending on the length of the given sentence. Each word should be on its own row.

Here are some style guides:

- Font family: Open Sans (https://fonts.googleapis.com/css2?family=Open+Sans)
- Background grey color: #e1e1e1
- Correct Green color: #4caf50
- Next button green color: #388e3c
- Space yellow color: #ffb74d



- The original sentence was “i love learning code”
- Each guessing row is one word (there are 4 words in this example)
- Yellow blocks are designated for spaces

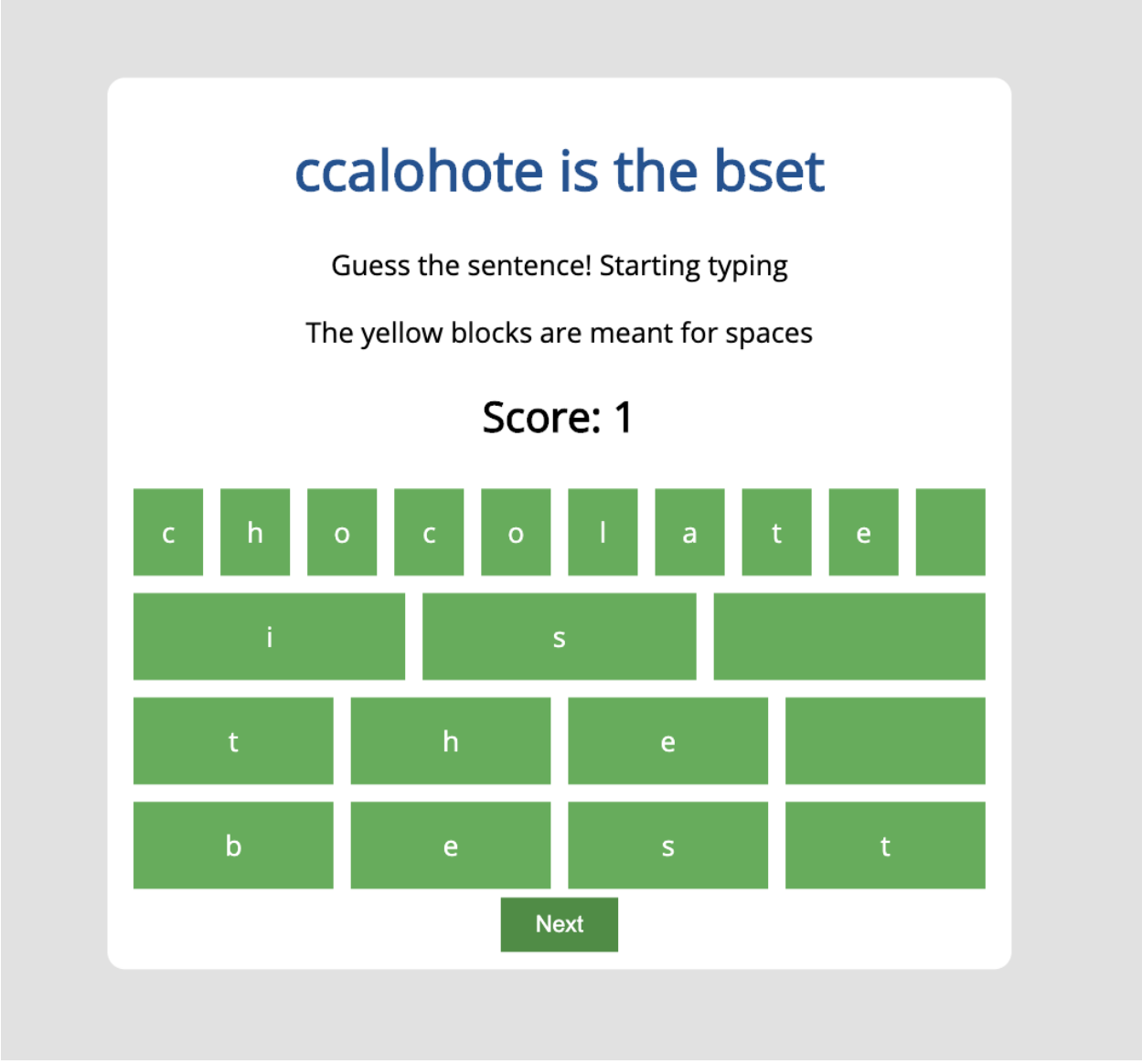
Part 4: Guess the sentence

In this part, you are going to add functionality for “guessing the sentence”. Take in keyboard input to type out the unscrambled word (user shouldn't have to use the mouse at all). The order of guessing must start from the beginning and you can only guess the next character. Guessed letters should be displayed in the letter blocks. For every character that was guessed correctly in the right position, the block turns green, otherwise remains as is.

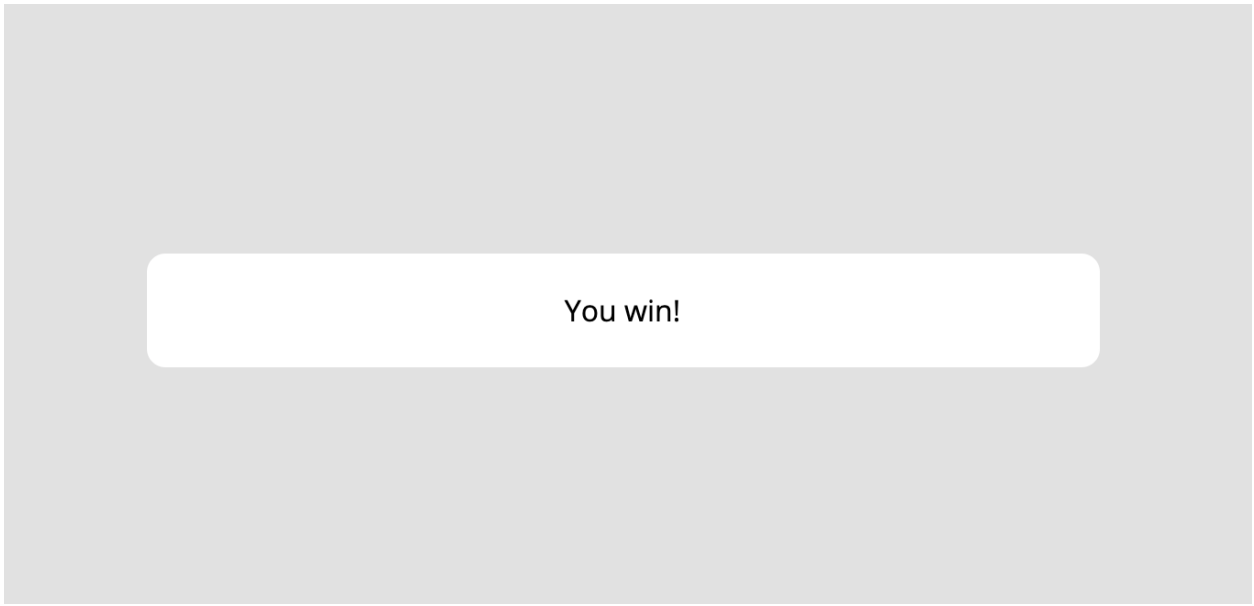
Part 5: Guess next

If the whole sentence is correctly guessed, there should be a next **button (<button>)** that shows at the bottom. If you click it or hit enter on your keyboard, it would try to fetch the next sentence from the same API, increasing the counter (for example the next API call would be https://api.hatchways.io/assessment/sentences/2). This would start the next guess.

Keep track of the score (each sentence +1 score)!



The maximum number of sentences is 10. After 10 correct sentences, the game will end! You can simply just display a message “You win!” like the following:



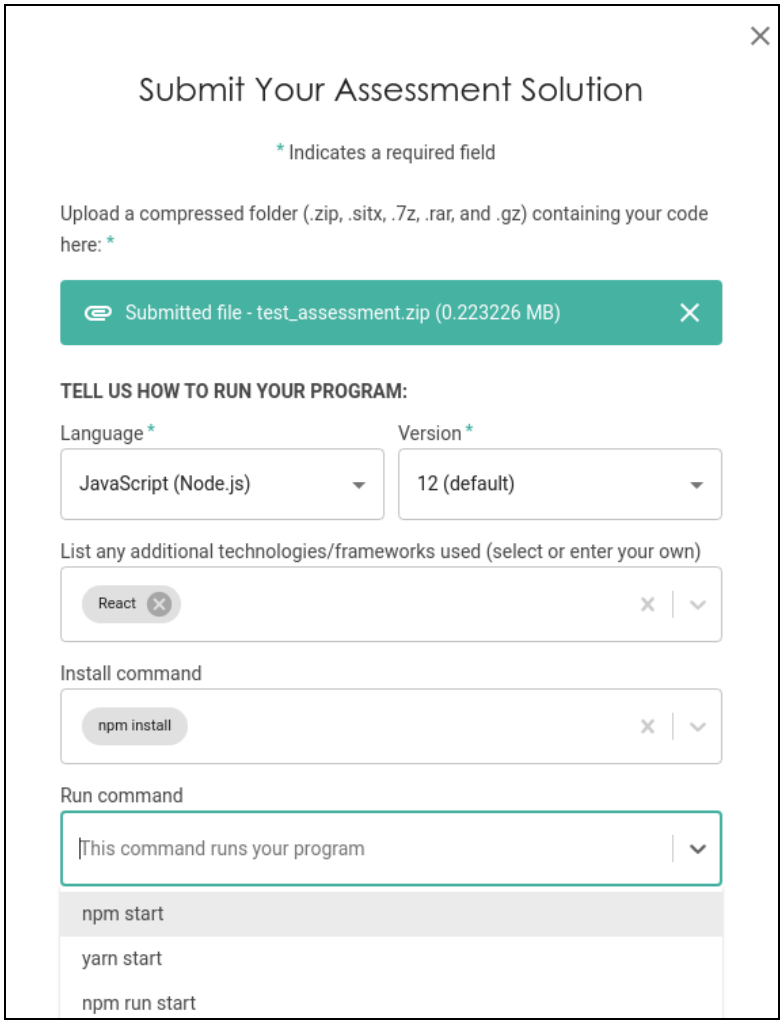
Here is a video of the entire app in action

Submission Details

Please submit your code in a compressed folder on the Hatchways platform. The max submission size is 5MB.

Upon clicking the submission button, you will see a form as pictured below. We need this information to be able to test your application.

1. Choose which language and technologies you used to develop your solution
2. Provide us with the **install command**, the **run command**, and the **port** that you used to run your application.
3. If you cannot find your commands in our suggestions, simply type your own and select “Use command”.



Do not submit any built folders, since the compressed folder will be too large. **Do not submit your external dependencies (like the node_modules folder), since the compressed folder will be too large. We will be installing your dependencies before we run your code.**

If your submission is too big and you can't figure out how to compress, you are welcome to email your solution to hello@hatchways.io.

Please include your name, and use the email you signed up with on the Hatchways platform. Use the subject line “Front-end Assessment Submission”.

Public Repositories

Do not post your solution to a public repository. We understand that you may want to share projects you have worked on, but many hours go into developing our tools so we can provide a fair skills evaluation.