mavens°

Hello and welcome to the mavens° coding assignment,

Goal

Implement a POC (proof-of-concept) of a web app game.
The game tests and provides feedback to the user on his reaction time and quick decision making.

Since we are working in cycles and using an MVP approach, your goal is to ship a functioning minimal product quickly.

General

In the game, a shape will be shown to the user in a random location on the screen (left/right), the user will have 1 second to react and enter on which side of the screen the shape was displayed.
By the user's reaction, feedback will be displayed.

Specification

The app starts with a page for inserting the user name and a ‘START’ button. The moment the user clicks ‘START’ the game loads in a waiting state.

During the waiting state, the user is not supposed to do anything. This state lasts randomly between 2-5 seconds.

After the waiting state, show the user an indicator on the left or right side of the screen.
It does not matter what the indicator looks like, it can be a simple black circle or anything else as long as it is easy to see that it appeared and on what side of the screen.
The indicator appears on the screen for 1 second.

During the indicator is visible, the user is expected to tap the 'a' or 'l' key on the keyboard:

- 'a' if the indicator is on the **left** side
- 'l' if the indicator is on the **right** side.

States:

User reaction	State type	Message
User taps any key in the waiting state	Mistake	‘Too Soon’
User taps an incorrect key in the indicator showing state	Mistake	‘Wrong Key’
The indicator disappears before the user types any key	Mistake	‘Too Late’
Correct key is tapped while the indicator is displayed	Success	

When there is a success or mistake, restart the game by going to the waiting state and at the same time show feedback for the user at the bottom of the page until the new indicator is displayed.

The feedback should be a text (the error type) in red for mistakes and green for success.

Backend

The backend should be simple. You will support a capability to show a leaderboard of results and work with API to enrich the data of the user.
For that, you are required to store the user name and the number of steps he completed successfully.

In addition to storing the data from the game, enrich each user's data using two APIs:

- For each user name, we want to guess the gender using this API: Genderize.io| Determine the gender of a name. Only accept probability > 0.95. Otherwise, the gender is “undetermined”. Add this data to each row in the DB

- Once we have the genders, fetch some mock data (based on gender) for each user using Random User Generator API. Enrich each row with this data.

In addition to storing the data, create an endpoint returning the list of users sorted by the number of steps they reached. No need to create the UI of the leaderboard.
The data can be stored in the backend in any way you find fit.

What we care about

- Accurate and correct code that works according to the specs above.
- Fast Implementation
- Readability of code
- Folder structure, abstraction of code, components hierarchy

What we care less about

- Pretty design
- Use of packages (use anything you want or nothing)
- Performance
- Security

After you submit a working code, we will probably talk about some of these subjects, but they do not affect how we score the task.

General Guidelines

- Must use React + Typescript
- You can use 'create-react-app' or similar for a template.
- After you finish the implementation, send us your code along with brief instructions for us to be able to test it.
- You can send the code in an e-mail or share a git repository

If you have any questions, feel free to contact us