Frontend Skill Assessment

Welcome to the assignment for candidates applying for a frontend position at Retm.

Overview

As part of the hiring process we are evaluating the candidate's technical capabilities through a simple skill assessment. The assessment is divided into several steps:

1. **Home assignment**: After being handed the skill assessment you will spend approximately 5-10 hours at home preparing a solution following the brief described further below.

   See section: **Home assignment** to learn more about this step.

2. **Submission**: Once you're done with the solution; you submit it. Your submission will be evaluated by two or more members of the frontend team at Retm. Afterwards you will typically hear back from us within a week unless communicated otherwise.

   See sections: **How to submit** and **How we evaluate** to learn more about the submission step.

3. **Workshop**: Depending on the outcome of the evaluation you'll be invited for an on-site (or remote depending on your location and availability) workshop where we'll actively work with your solution, solidifying and extending it with features.

   See section: Workshop to learn more about this step.

4. **Final evaluation**: After the workshop we'll make a final evaluation on the process as a whole. This will also be the final step of the hiring process. Unless communicated otherwise, we expect to provide feedback within a work week.

Home assignment

As per step 1 you're tasked with creating a frontend prototype of a simple signup application. At Retm we use **Angular** as the main platform to build applications. The important thing is that you use **Angular** for solving the core tasks. UI kit and library choices certain requirements must be fulfilled, as these will be considered in the evaluation. The requirements are listed below in the form of business and technical requirements.

Business requirements

The result of your submission should cover the following use cases, A user should be able to:

1. Validate the token provided by the URL.
2. Show error message if the token is missing from the URL or the token is not valid.
3. Create store ID and validate is availability.
4. Choose user name with only (a-z) letters.
5. Choose password with the ability to reveal it.
6. Confirm the chosen password.
7. Post the data to the API Layer.

Technical requirements

The result of your submission should cover the following technical aspects:

1. You display modern best practices and good code hygiene.
2. You follow Git practices and your commit history tells the story of how you iterated on the task.
3. The application meets accessibility standards.
4. The application is scalable, maintainable and pragmatic in its structure. Both on code and file level.
5. The application is built for testability, demonstrated by actual tests.
6. Complicated or non-obvious choices in the code must be supported by code-level comments.
7. You use the provided API layer to develop the solution.
8. You use the UI design provided at this page: https://retmsa.webflow.io/sign-up-cloud/page
9. Your solution is available in the /client folder including a README.md file.
10. The README.md must contain a guide on how to start your application as well as brief (a few lines) explanation of how you solved the task.

API Layer

To support the development of the signup page we've provided a REST API layer implementing the actions for the resource.

This API reference can be used as follow:

- **Version:** 5.3
- **Host:** signup.retm.sa/join/
- **Protocols:** https
- **Accepts:** application/json
- **Responds With:** application/json

GET /check.php?get={{token}}

This is used to validate the token.

Responses 200

```
{
    data:  boolean   // Validation if the key exist
    name:  string    // the user first name
}
```

Sample success token: Ym5oMmMyMTRVeXNyYlhkemVVSnFiekpHTUZaeVp6MDk

Sample error token: bxpiwX7zfxF0j9ikSrTAQiip5hWREar1

GET /check.php?serviceCode={{store ID}}

This is used to validate the store ID.

Responses 200

```
{
    avilable:  boolean   // validate the availability of the store code
}
```

Sample error store id :

'hello', 'itsme', 'welcome', 'logout', 'login', 'items', 'shop', ‘blog', 'news','food', 'about', 'location', 'press', 'today', 'pizza', 'help', ‘faqs', 'admin', 'home', 'suppot', 'join', ‘payments'

Workshop

Depending on the outcome of the evaluation you'll be invited for an on-site (or remote depending on your location and availability) workshop where we'll actively work with your solution. More specifically, you will, in collaboration with two members of our Frontend team work on extending your solution with new features.

The workshop will approximately take 3 hours and consists of initial briefing, hands-on development and conclusion.

At the workshop we will develop your solution with the following features (in order).

During the workshop the following use cases should be covered, A user should be able to:

1. Develop landing page for switching from different version.
2. Validate ownership of the selected Store ID.
3. The ability to keep the same Store ID or create new one.

⚠

IMPORTANT: The feature development is a key part of the collaborative sessions during the workshop. It's therefore important that you do not implement these use cases in your initial home assignment submission or otherwise provide a final solution. You're of course welcome to consider how you would implement them as well as bringing notes, ideas, questions etc. in preparation for the workshop.

More details about the format, use cases and content will be given out at the initial briefing.

The goal of the workshop is not to work against the clock and test how much can be produced in 2 hours. The goal is to get a feel of how you work in a collaborative setting, how you approach and prioritise feature development, how well you can leverage your knowledge and how you handle impediments and seek sparring.

How to submit

You submit your result via a repo on GitHub. Clone this repo and point the remotes to your own public repo and you should be good to go.

Make sure that you have a Git history with meaningful commits that tells the story of how you solved the task. A single init commit (or similar) containing the entire solution is highly discouraged and will prevent us from doing proper evaluation.

Make sure to provide guidance on how to run your solution in a README.md. Also use this file to highlight if there are any particular aspects of your solution you want us to consider.

Make sure to fix the versions of any dependencies. This will ensure that the runtime environments will match.

```
# AVOID
"some-package": "^0.3.1"

# GOOD
"some-package": "0.3.1"
```

⚠

IMPORTANT: A static Zip/RAR file containing the solution is not considered a valid submission and will be disregarded.

Once you're ready to submit, send link to the remote GitHub repo to assignment@retm.com.sa

Unless agreed otherwise, we expect the submission to be handed in within 4 days. The net amount of time expected on the assignment is ~5-10 hours. If you, for any reason, estimate that you cannot finish the assignment within the time frame please be communicative about an ETA.

How we evaluate

When we evaluate your home assignment, before the on-site workshop, these are the main areas that we prioritise:

Does your Git history reveal your workflow?
When taking a first look at your Git repo we assess how you've iterated on the task. We do this by glancing through your Git commits. This will give us an idea of how you've prioritised the work and how you've iterated on the development. E.g. how are business critical vs. non-critical implementations balanced? Therefore make sure that your commit messages provide specific context, e.g.:

```
# AVOID
some fixes

# GOOD
fix(list): fix issues with the tasks list
```

Does your solution work out-of-the-box?
Next, we read through your README.md and follow your instructions. Did you add custom steps? Do they work? Or, does it requires manual debugging to get up and running?

To what degree are the requirements met?
Once everything is up and running, dev server, API etc., we open your solution in a browser and click around. Does it work? Are the requirements met? If not, we ask "why" and seek for answers in README.md or code-level comments.

How does it work?
Provided that all of the acceptance criteria are met, we start looking into how you implemented the solution.

- How did you decide to structure the application? How is it architectures?
- How do you manage state in the application?
- How do you divide the application into smaller components?
- How scalable is your application?
- How well does it perform? Which coding patterns and constructs do you utilise?
- Where and how do you handle edge cases? What about error handling? Is it accessible?

How's the sense of quality?
While assessing the inner workings we also take a step back and look at the overall solution. How is it crafted? Does it feel solid, yet simple, in its execution? Is it built using common sense and

pragmatic thinking? Is it well-tested? Is a good understanding of UX and application patterns applied? Is there a coherent idea emerging from the code and structure? What's the level of code smells? Are listing rules strictly followed?

Equally there are areas that will NOT weight heavily in our evaluation:
Visual design quality
We do not evaluate the overall visual look and feel. We expect that your main focus is spent on developing and fine-tuning business and component logic. We do, however, expect that you show an understanding of common UX patterns and best practices, and can make good use of these when making UI/UX decisions as well as when utilising/developing lower level UI components.

Timeframe
If nothing else is agreed, we expect the submission to be handed in within 4 days. However, we do not evaluate you on the time you spent on the assignment. I.e. we do not count the days from when you received the assignment till you submit, neither do we consider time intervals between your Git commits.

Kind regards,
The frontend team @ Retm