Front-end Test (Angular 2+) - Anota AI

This test is presented to the Front End development candidates to evaluate technical knowledge.

Prerequisites

- Make sure that the Angular 2+ framework is already installed along with its dependencies.

Notes

- You can use CSS pre-processors and we will be happy if you organize your code using a know CSS architecture/methodology.
- You can **NOT** use a complementary framework/library (bootstrap, angular material, etc.) for handling the style of the page and listing the data.
- Don't be alarmed, we don't expect everyone to be able to complete all the tasks, so do what you can without worrying.
- Instructions for submitting the test are located at the end of this document.

The Challenge

Your goal is to create a simple page replicating the provided **image** and **video** using Angular 2+ and CSS. While precise measurements aren't explicitly provided, we encourage candidates to pay close attention to the details and dimensions of each element.

A key aspect of front-end development is the ability to translate visual designs into pixel-perfect user interfaces. Despite not expecting everyone to match the image precisely, we highly value candidates who demonstrate a keen eye for design elements.

Files

https://githubanotaai.github.io/frontend-interview-mock-data/teste/Test-front-anota-ai.jpg

https://githubanotaai.github.io/frontend-interview-mock-data/teste/Test-front-anota-ai.webm

Auxiliary files

https://githubanotaai.github.io/frontend-interview-mock-data/assets/favicon.png

https://githubanotaai.github.io/frontend-interview-mock-data/assets/close.svg

https://githubanotaai.github.io/frontend-interview-mock-data/assets/128x128.png

Tasks

1 - Create a workspace and an **initial application** with the Angular 2+ framework, preferably Angular 17+.

2 - Create the **Header** of the page.

3 - Create a **list of cards**. For, this you'll need to send a HTTP GET request to the URI below to receive the JSON list:

```
https://githubanotaai.github.io/frontend-interview-mock-data/cardlist.json
```

3.1 - Use the following reference for the type names:

```
Type 1: Paisagem

Type 2: Flor

Type 3: Pizza
```

4 - Insert an option to **delete card** (the card must be removed from the list on the front-end)

5 - Insert a "search field" and implement a **product filtering** feature where the user can filter the cards by title or description.

Bonus Tasks

1 - If you're comfortable writing unit tests, we would greatly appreciate it.

1.1 - While we do prefer Jest, if you prefer to use Karma/Jasmine, that's perfectly fine too.

Submission Instructions

1 - Make sure your project is fully functional and ready for evaluation.

2 - Remove the "node_modules" folder to reduce file size.

3 - Compress all project files into a .zip file, naming it after yourself.

4 - Send us the .zip file with your name for evaluation.