

AllianceBook

Testing assignment for a frontend developer

jan.timoransky@brackets.sk www.brackets.sk



Introduction

This documents represents the testing brief of the imaginary project called *AllianceBook*. It will be used to assess the abilities of the candidate to verify theirs

overall skills and chosen approach in solving problems and challenges. It tries to simulate the real world scenario or an example of the problem that programmer at BRACKETS encounters, but in the very fun world of Star Wars.

Don't hesitate to ask any further questions, we'll try to be cooperative.

[2]

[3]

Brief

Situation

You are a highly tech skilled member of Alliance programming UI to help other members to see info about all the characters in the universe.

Luckily, you found an API: <https://swapi.py4e.com/> with all the data you need. But since the API doesn't provide any avatar or image attribute for said characters, you can use this endpoint to load avatar by character ID: <https://starwars-visualguide.com/assets/img/characters/{id}.jpg>

May the Force be with you.

Goals

Your goal is:

1. Use API to load all characters and show them in listing with pagination
2. Implement search and some filtering by the attributes of your choice
3. Make it all neat, pretty and responsive. Sky is the limit here but the UI should be clean, modern and user friendly
4. Code with a11y in mind so the UI is fully functional with keyboard only (some of the Alliance computers sadly don't have mouse attached)
5. Animations are more than welcome to see but don't overdo it, find the balance

Stack

We prefer to use React or Vue on frontend so you are free to use either of them. For the styles, we love to use Tailwind, but you are free to choose any other CSS framework or even a component library (we use ChakraUI on some projects as well).

Evaluation Criteria

We will consider multiple criteria during assessment. Focus rather on the approach and best practices then on the completeness of the assignment (rather less, but higher

quality code). Feel free to create a README file, where you can explain key concepts used, your approach or other aspect of your results, that should not be overlooked.

These are only examples that may guide you fulfilling this assignment, you don't need to cover everything and you can add more.

- Show us your work through your commit history
- **Best practices and approach chosen**
- **Maintainability** - is it written in a clean, maintainable way?
- Correctness - does the functionality act in sensible, thought-out ways? • Is it ready for cooperation with the rest of the team (code style, Prettier config, ...)
- Completeness of the system is not important (we don't expect it to be the final solution), but **highlight what's not yet complete when submitting**