Senior Frontend Developer Assignment

Candidate Version

Introduction

Congratulations, you made it to the technical interview for the Senior Frontend Developer position @Eversports 🥳

Context

At Eversports, as a senior frontend developer, besides being an active contributor to the delivery efforts of a product team, you will be responsible for collaborating with others on contributing to the development of our design system components and principles.

This design system is used by different teams, both B2C and B2B and needs to potentially accommodate different needs according to product specifications.

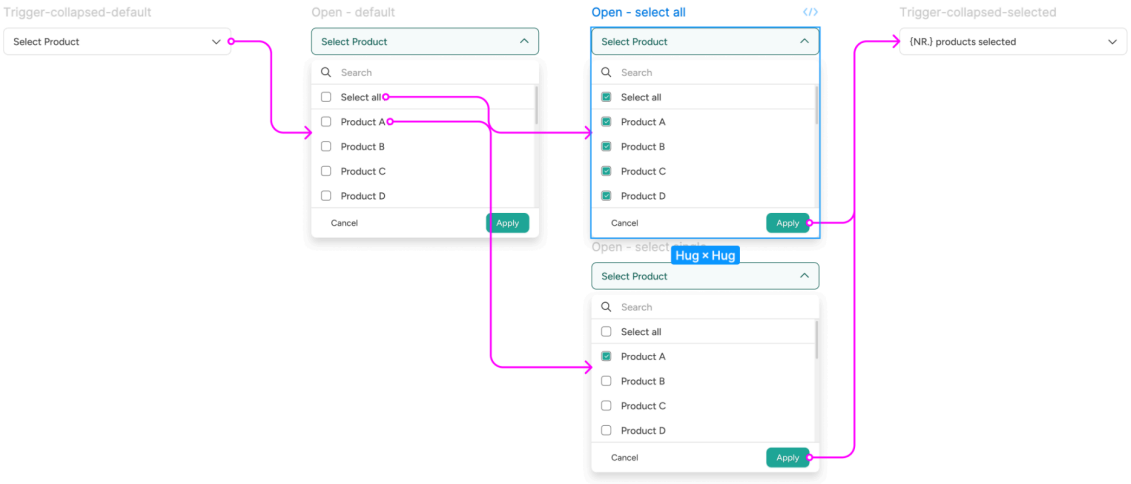
You are tasked with 2 challenges:

1. develop a re-usable multi-select component for the design system;
2. integrate the component within a page;

Assignment

Task 1 - Multi-Select Components

Develop a reusable and responsive multi-selection component based on the provided Figma design that can be used within different contexts (be sure to be logged in to figma so you can see all the design specifications of each element).



Note: Since there's no design system provided for you, feel free to either reach out to any component library that you would like to use to help you with the building blocks for the UI or to implement the components themselves from scratch. The design specs from figma should be respected and match as close as possible independently of which solution you choose.

Acceptance Criteria

- As a user I can search for specific items.
- As a user I can see the list of queried items.
- As a user I can select or deselect all items at once.
- As a user I can select/deselect each item individually.
- As a user I can cancel the selection process and revert any changes by clicking the cancel button or clicking outside of the dropdown.
- As a user I can apply my selection by clicking the apply button.
- As a user I can use the filter in different types of screen sizes.

Technical considerations

- Implement the component in a way that it can be easily reused in different domains (e.g: product selection, user selection, sport class selection, etc..).

Assumptions and Decisions

- You are expected to make valid assumptions and decisions to create the best possible user experience (UX) and component functionality - some of them are product decisions, some of them are technical. (p.ex: what if there's no items available for the search? What if..., etc...)
- Document any assumptions and decisions you think are worth mentioning that you decided on during the development process with co-located code comments or if longer within a README.md file.

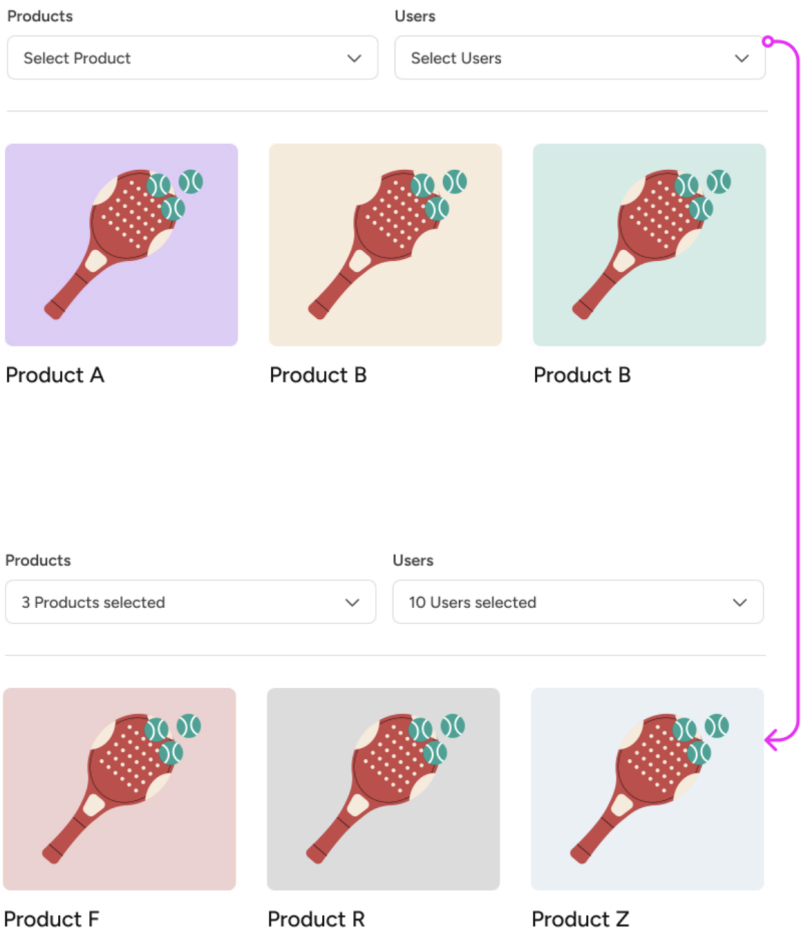
Task 2 - Purchased Product Page

Build a page that shows a list of purchased products by users leveraging the component from part 1 to allow for the filtering of both products and users. This list should show the purchases that are the result of applying those filters.

The UI below is just a representation of how we expect the page to look, but feel free to customize it however you feel best, the important part is to see the component you developed from before in action within the context of a general web application.

In the page the user can interact with both a product and a user filter and see the filtered data when the filters are applied.

The dataset available has 300 products, 100 users and 150 purchases, all of them randomly generated using https://fakerjs.dev . These lists can be paginated through a cursor based pagination.



Both the **products** and the **users** for this example can be fetched over a graphql api that allows both cursor based pagination and search (you can find it in the **graphql-server** within the boilerplate code). For the filtered results you can use the **purchases** query that returns you all purchases by default, or the filtered purchases according to the products/users selected.

Acceptance Criteria

- As a user I can see the available products, users and purchases.
- As a user I can filter purchases by specific products and/or users.
- As a user I can see the results of my selection.
- As a user I can clear the filter selection.
- As a user I can use the filters in different types of screen sizes.

Resources

Together with this document you should receive access to the following two documents

🎨Figma Design (Task 1)

Here is the Figma Design for task 1.

🗂️Boilerplate (Task 2)

Together with this document you should have received a **frontend-interview-main.zip** file containing the assignment boilerplate code where you can find both a mocked graphql-server as well as a client boilerplate setup with a graphql-integration that you can use to start your development. In it you can also find a README.md file that explains how to set it up and provides some helpful links for the development of the assignment.

Deliverables

Github Repository

The deadline for handing in your assessment is **the day before the technical interview** takes place.
Please create a dedicated repository on Github and provide the link to it via the Eversports Technical Homework Submission. The repository should include all documents that you want to hand-in.

README File

Please provide a README.md file in your Github repository, which **explains any decision or assumptions** you took when developing the component that you believe is worth mentioning, as well as a rough estimation of how much time you spent on the assignment.

Code Implementation

Implement the component and the page components using React and **choose a styling solution**, whatever you feel most comfortable and/or would fit the solution best (the provided client boilerplate is setup with tailwind, but feel free to use your own styling solution if you prefer). Ensure clean, readable, and well-documented code.

Use of AI Tools

At Eversports, we actively use AI as a supporting tool in our daily work, and we welcome candidates who can use it effectively and responsibly. You are free to use AI tools (such as ChatGPT, GitHub Copilot, Claude Code or others) while completing this assignment. To help us understand your individual contribution and reasoning, please **describe where and how you used AI assistance** in the AI-SUPPORT.md file in the Github repository.

Example AI-SUPPORT.md format:

- Used Claude Code to generate initial code structure.
- Asked GitHub Copilot for suggestions on unit test setup.
- Used ChatGPT to rephrase parts of this README for clarity.

It's important to note that there are **no penalties for using AI**. Our goal is simply to understand which parts reflect your own problem-solving and which were supported by tools.

If you have any questions or need further clarification, feel free to reach out.

Enjoy the assignment 🙂