Take Home Front End Engineering Challenge: Searchable Dropdown with Auto-Complete

Time Limit: 30 min

Objective:

Implement a custom built **Searchable Dropdown Component** that allows users to dynamically filter and select options based on their input.

Example here: https://ui.shadcn.com/docs/components/combobox

Instructions:

You are required to create a **TypeScript (you pick the framework)** component that meets the following requirements:

Core Functionality

- Users can type into an input field.
- The dropdown filters the list of available options based on user input.
- Clicking on an option selects it and updates the input field.
- If the input is empty, all available options should be displayed.
- Clicking outside the dropdown should close it.

Bonus Points:

- **Debounced Input Handling**: Implement debouncing to optimize filtering performance.
- **Dropdown Animations**: Smooth fade-in/out animations when opening and closing.

Starter Data

Use the following sample data for the dropdown options:

```
const options = [
  { id: 1, label: "Apple" },
  { id: 2, label: "Banana" },
  { id: 3, label: "Cherry" },
  { id: 4, label: "Date" },
  { id: 5, label: "Elderberry" }
 ];
```

Step 2: Feedback Interview

Once you have completed the take-home challenge, the next step will be a **feedback interview**. This will consist of two parts:

Part 1: Code Walkthrough - 5 minutes.

You will be asked to:

- Explain your thought process behind your implementation.
- Justify architectural and performance decisions.
- Highlight any trade-offs you made.

Part 2: Live Pair Programming - 10 minutes.

You will be given a small feature to implement live. This will include:

- Integrating an API response into your dropdown component.
- Transforming API data into the required dropdown format.
- Handling cases where the API request fails or returns an empty list.

Final Step: Pair Programming & Optimization - 10 minutes

During the live session, we will:

1. **Refactor the component** to improve readability and efficiency.
2. **Optimize performance** to reduce unnecessary renders and improve responsiveness.
3. **Discuss trade-offs and best practices** in real-world applications.

Wrap-Up

This challenge is designed to evaluate both your technical proficiency and problem-solving approach in a real-world scenario.

By the end of this process, we aim to assess how you:

- Implement clean, efficient, and scalable code.
- Handle dynamic user interactions with smooth UX.
- Think critically through live problem-solving and optimizations.
- Communicate and collaborate effectively in a pair programming setting.

We look forward to seeing your approach and discussing your thought process in the feedback interview! 🚀