SkillBlend Intern Assignment: Accessibility Widget

Thank you again for your interest in the internship at **Skillblend**. We've reviewed your profile and are excited to see how you approach a real-world challenge.

This assignment is designed to evaluate your **problem-solving**, **technical capability**, and **design sense** — while keeping things practical and achievable. We're not looking for perfection — we're looking for **clarity, effort, and how you think**.

Note: These instructions are for reference only. If you have a better approach while meeting the requirements, it will be ok.

Part 1: Requirements

Objective:
Build a comprehensive floating accessibility widget with multiple customization options (as shown in the reference image at the end). The widget should help users personalize their browsing experience for better accessibility.

You are free to choose your **own UI/UX layout**, as long as it is:

- Clean, simple, and accessible
- Functional and clear for end users
- Matches a modern design style (like the reference)

Core Features

1. Floating Button
- Position: Bottom-right corner
- Expands into a menu when clicked

2. Accessibility Controls
A. Visual Adjustments

- • Contrast+: Increase color contrast
- • Highlight Links: Emphasize hyperlinks
- • Bigger Text: Adjust font size
- • Text Spacing: Modify line/letter spacing

B. Behavior Controls

- • Pause Animations: Stop CSS/JS animations
- • Hide Images: Toggle image visibility

- • Dyslexia-Friendly Font: Switch to OpenDyslexic

C. Interface Tools

- • Cursor: Enlarge pointer
- • Tooltips: Force visible alt-text
- • Page Structure: Show heading outlines

3. Technical Requirements
- Embeddable via single script tag
- Real-time preview of changes
- Persistent settings (localStorage or backend)

Part 2: Implementation Guide

Task Breakdown & Technical Approaches

1. Base Structure (Vue.js)

Command:
```bash
npm create vite@latest a11y-widget --template vue
```

Key Components:
- FloatingButton.vue (Trigger)
- MenuPanel.vue (All controls in accordion sections)

2. Feature Implementation

Feature	Implementation Approach	Libraries/Tools
Contrast+	Apply CSS filter: filter: contrast(120%)	Pure CSS
Highlight Links	Add border-bottom: 2px solid #FF0000 to <a> tags	https://github.com/kleinfreund/vue-accessible-color-picker
Bigger Text	Modify :root { --font-size: Xrem }	CSS Variables
Text Spacing	Adjust letter-spacing/line-height	https://www.npmjs.com/package/vue3-resize-text
Pause Animations	Inject prefers-reduced-motion styles	https://github.com/formkit/auto-animate

Hide Images	Apply img { visibility: hidden }	CSS/JS toggle
Dyslexia Font	Load OpenDyslexic via CDN	https://opendyslexic.org
Cursor	Enlarge with cursor: url(large.cur)	Custom cursor files
Tooltips	Force title/aria-label visibility	https://www.npmjs.com/package/%40programic/vue3-tooltip?activeTab=readme
Page Structure	Outline headings (h1-h6)	https://github.com/ffoodd/a11y.css

3. Advanced Integration

Dyslexia Toggle Example:

```javascript
function loadDyslexicFont() {
  const link = document.createElement("link");
  link.href = "https://cdn.jsdelivr.net/npm/opendyslexic@latest/opendyslexic.css";
  link.rel = "stylesheet";
  document.head.appendChild(link);
}
```

Animation Pausing Example:

```css
@media (prefers-reduced-motion: reduce) {
  *, ::before, ::after {
    animation-delay: -1ms !important;
    animation-duration: 1ms !important;
    animation-iteration-count: 1 !important;
  }
}
```

Submission Requirements

1. Code
- GitHub repo with /widget (Vue) + optional /api (Django)

2. Demo Video

3. Documentation
- README.md with:

- Setup instructions
- Implementation choices explained

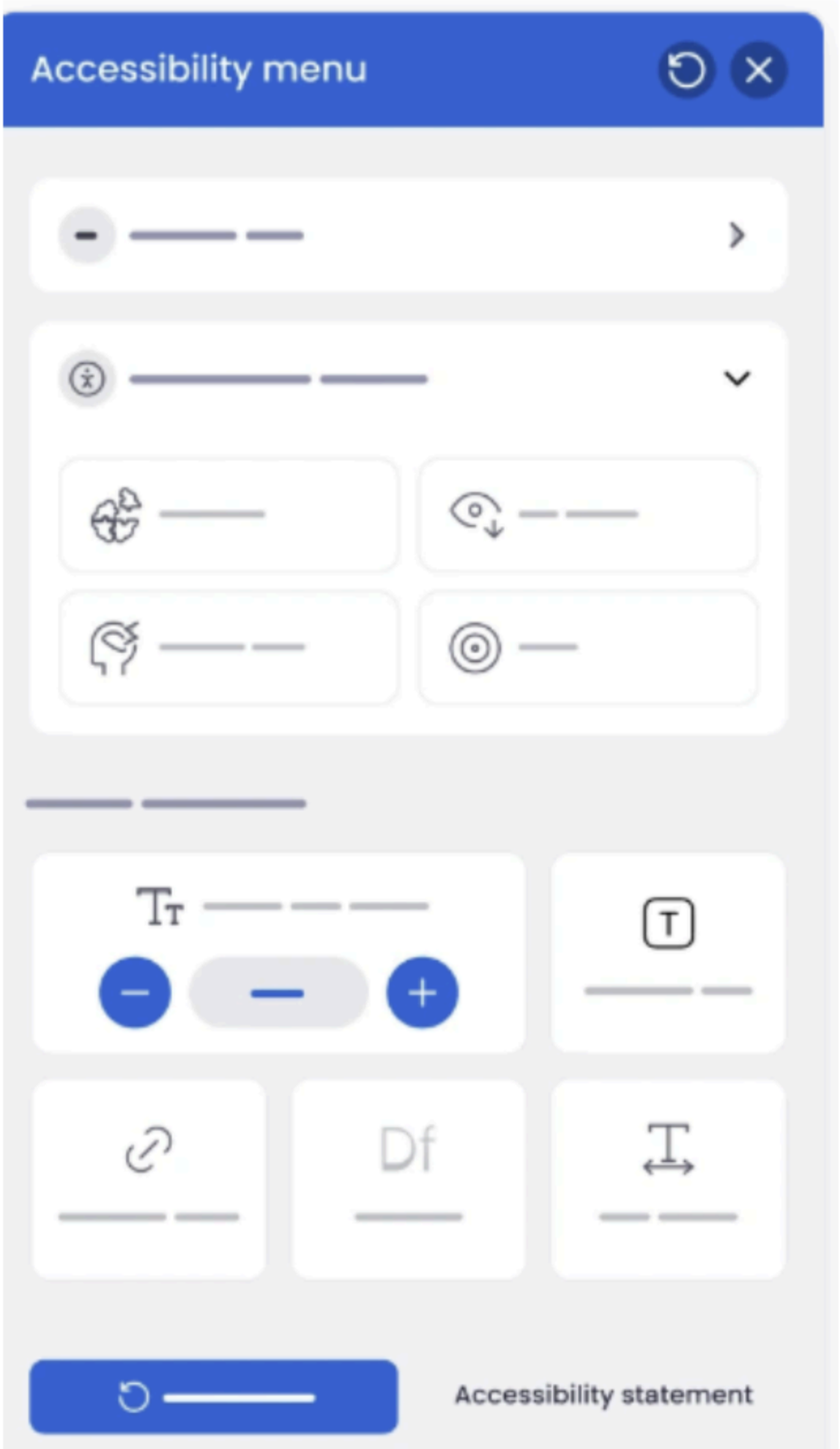
Evaluation Focus

- Self start and problem solving
- Feature completeness
- Code organization
- User experience

Deadline: 10 days

Tip: "Start with 3 core features (Contrast+/Text Size/Dyslexia), then expand. Quality > quantity!"

UX Reference


Accessibility menu
Df
Accessibility statement