**Folder Structure**

src/

├── components/

│ └── AccessibilityWidget/

│ ├── AccessibilityWidget.jsx

│ ├── ThemeToggle.jsx

│ ├── FontSizeControl.jsx

│ ├── LineHeightControl.jsx

│ └── AccessibilityWidget.module.css

├── context/

│ └── AccessibilityContext.jsx

├── App.jsx

└── index.js

**1. 📦 Accessibility Context (AccessibilityContext.jsx)**

import React, { createContext, useState, useContext } from 'react';

const AccessibilityContext = createContext();

export const AccessibilityProvider = ({ children }) => {

const [theme, setTheme] = useState('default');

const [fontSize, setFontSize] = useState('default');

const [lineHeight, setLineHeight] = useState('default');

const resetSettings = () => {

setTheme('default');

setFontSize('default');

setLineHeight('default');

};

return (

<AccessibilityContext.Provider value={{

theme, setTheme,

fontSize, setFontSize,

lineHeight, setLineHeight,

resetSettings

}}>

{children}

</AccessibilityContext.Provider>

);

};

export const useAccessibility = () => useContext(AccessibilityContext);

**2. 🧩 Common Components**

**AccessibilityWidget.jsx**

import React from 'react';

import styles from './AccessibilityWidget.module.css';

import ThemeToggle from './ThemeToggle';

import FontSizeControl from './FontSizeControl';

import LineHeightControl from './LineHeightControl';

import { useAccessibility } from '../../context/AccessibilityContext';

const AccessibilityWidget = () => {

const { resetSettings } = useAccessibility();

return (

<div className={styles.widget}>

<div className={styles.header}>

<h2>Accessibility Adjustments</h2>

<button onClick={resetSettings}>Reset settings</button>

</div>

<ThemeToggle />

<FontSizeControl />

<LineHeightControl />

</div>

);

};

export default AccessibilityWidget;

**ThemeToggle.jsx**

import React from 'react';

import { useAccessibility } from '../../context/AccessibilityContext';

import styles from './AccessibilityWidget.module.css';

const ThemeToggle = () => {

const { theme, setTheme } = useAccessibility();

return (

<div className={styles.control}>

<label>Theme</label>

<div>

<button onClick={() => setTheme('default')} className={theme === 'default' ? styles.active : ''}>Default</button>

<button onClick={() => setTheme('dark')} className={theme === 'dark' ? styles.active : ''}>Dark</button>

</div>

</div>

);

};

export default ThemeToggle;

**FontSizeControl.jsx**

import React from 'react';

import { useAccessibility } from '../../context/AccessibilityContext';

import styles from './AccessibilityWidget.module.css';

const FontSizeControl = () => {

const { fontSize, setFontSize } = useAccessibility();

const increaseFont = () => {

setFontSize(prev => prev === 'large' ? 'large' : 'large');

};

const decreaseFont = () => {

setFontSize(prev => prev === 'small' ? 'small' : 'small');

};

return (

<div className={styles.control}>

<label>Font size</label>

<div>

<button onClick={decreaseFont}>A−</button>

<span>Default</span>

<button onClick={increaseFont}>A+</button>

</div>

</div>

);

};

export default FontSizeControl;

**LineHeightControl.jsx**

import React from 'react';

import { useAccessibility } from '../../context/AccessibilityContext';

import styles from './AccessibilityWidget.module.css';

const LineHeightControl = () => {

const { lineHeight, setLineHeight } = useAccessibility();

const increaseLineHeight = () => {

setLineHeight('high');

};

const decreaseLineHeight = () => {

setLineHeight('low');

};

return (

<div className={styles.control}>

<label>Line height</label>

<div>

<button onClick={decreaseLineHeight}>↓</button>

<span>Default</span>

<button onClick={increaseLineHeight}>↑</button>

</div>

</div>

);

};

export default LineHeightControl;

**3. 🧩 CSS Module Example (AccessibilityWidget.module.css)**

.widget {

background-color: #1e1e1e;

color: white;

padding: 1rem;

border-radius: 8px;

width: 300px;

font-family: sans-serif;

}

.header {

display: flex;

justify-content: space-between;

align-items: center;

}

.control {

margin-top: 1rem;

}

button {

margin: 0 0.5rem;

padding: 0.4rem 0.8rem;

border: none;

border-radius: 4px;

cursor: pointer;

}

.active {

background-color: #0d6efd;

color: white;

}

**4. ✅ Apply Global Styles Conditionally**

You can conditionally add classes to your <body> in App.jsx or use useEffect to apply font size and theme globally:

import React, { useEffect } from 'react';

import { useAccessibility } from './context/AccessibilityContext';

import AccessibilityWidget from './components/AccessibilityWidget/AccessibilityWidget';

function App() {

const { theme, fontSize, lineHeight } = useAccessibility();

useEffect(() => {

document.body.className = `${theme} ${fontSize} ${lineHeight}`;

}, [theme, fontSize, lineHeight]);

return (

<div>

<AccessibilityWidget />

<main>

<p>Your content here</p>

</main>

</div>

);

}

export default App;

**5. 🌍 Add Global CSS (Optional)**

body.default {

background: white;

color: black;

}

body.dark {

background: black;

color: white;

}

body.large {

font-size: 18px;

}

body.small {

font-size: 14px;

}

body.high {

line-height: 2;

}

body.low {

line-height: 1;

}

**Here's a Breakdown of What We'll Build**

**Accessibility Features:**

* Theme Switcher: Default / Dark
* Font Size Adjustment: Increase / Decrease / Reset
* Line Height Adjustment: Increase / Decrease / Reset
* Reset All Settings
* Hide Widget

**🧱 Architecture Overview (React Best Practices)**

**1. Components:**

* AccessibilityWidget: Main panel
* ThemeToggle
* FontSizeAdjuster
* LineHeightAdjuster
* ResetControls
* WidgetToggleButton: Floating icon to open/close the panel

**2. Context:** Use AccessibilityContext to manage settings globally.

**3. Storage:** Use localStorage to persist user preferences.

**4. Styling:** Use a global CSS module or Tailwind CSS for consistent and scalable styles.