# **Accessibility Report for Front-End Design**

This document evaluates the accessibility of the front-end design and implementation based on the provided HTML and CSS code, highlighting strengths and areas for improvement to ensure compliance with accessibility standards such as WCAG 2.1.

## **1. Accessibility Strengths**

### **1.1 Semantic HTML Structure**

* Usage of semantic elements like <header>, <footer>, <nav>, <article>, and <section> ensures meaningful content organization and enhances screen reader navigation.
* Proper <h1> to <h6> hierarchy is observed, which helps users understand the content structure.

### **1.2 Labeling and Form Accessibility**

* Forms include <label> elements associated with input fields via for attributes.
* Error messages and success messages are visually distinct and well-placed for form feedback.
* Placeholders and hints are appropriately used in forms to assist users with input expectations.

### **1.3 Keyboard Accessibility**

* Interactive elements such as links (<a>), buttons, and form inputs are focusable using the tab key.
* Language switchers and navigation menus are accessible via keyboard.

### **1.4 Color Contrast**

* The contrast between text and background colors, especially for buttons and links, meets the minimum AA standards in most instances (e.g., primary buttons and navigation links).
* Usage of CSS variables makes color theming easier and consistent.

### **1.5 Responsiveness**

* CSS media queries ensure content adapts to various screen sizes, improving accessibility for mobile and tablet users.

### **1.6 Accessible Navigation**

* ARIA landmarks such as aria-label, aria-labelledby, and aria-roledescription are used effectively in navigation, carousels, and form sections.
* Screen reader-friendly language switching mechanism is implemented with a <select> dropdown.

### **1.7 Error Handling**

* Forms provide inline error messages using CSS classes for visual feedback.
* Fields with errors are highlighted, ensuring users can identify and resolve input issues easily.

## **2. Areas for Improvement**

### **2.1 Missing Alt Attributes**

* Several <img> tags lack descriptive alt attributes (e.g., in the hero section or certificates carousel).
  + **Recommendation**: Add descriptive alt text for all images to convey meaning to screen readers.

### **2.2 Focus Indicators**

* While interactive elements are keyboard-accessible, visible focus indicators are inconsistent across some buttons and links.
  + **Recommendation**: Ensure all interactive elements have clear focus states (e.g., using :focus pseudo-class).

### **2.3 Language Attributes**

* The <html> tag includes a lang="en" attribute, but dynamically changing the language via the language switcher does not update this attribute.
  + **Recommendation**: Programmatically adjust the lang attribute when the user selects a new language.

### **2.4 ARIA Role Overuse**

* Some <div> elements with ARIA roles could be replaced with native HTML elements (e.g., <button> instead of <div> with role="button").
  + **Recommendation**: Prefer native HTML elements for better support and simpler maintenance.

### **2.5 Non-Descriptive Links**

* Links such as "Read More" or "View Details" do not provide sufficient context for screen readers.
  + **Recommendation**: Add aria-label attributes or visually hidden text to describe the destination (e.g., "Read More about Project X").

### **2.6 Color Contrast in Some Areas**

* Certain elements (e.g., .related-post-item a hover state and secondary buttons) have insufficient contrast against their background.
  + **Recommendation**: Re-evaluate contrast ratios and adjust colors to meet WCAG AA standards (minimum contrast ratio of 4.5:1 for normal text).

### **2.7 Skip to Content**

* No "Skip to Content" link is provided for keyboard users to bypass repetitive navigation.
  + **Recommendation**: Add a "Skip to Content" link as the first interactive element on the page.

### **2.8 Table Accessibility**

* While tables are not used extensively, any future tables should include proper <thead>, <tbody>, <th>, and scope attributes to ensure accessibility.

### **2.9 Carousel Accessibility**

* The certificates carousel lacks keyboard navigation support and focus management for interactive slides.
  + **Recommendation**: Enhance the carousel with ARIA attributes like aria-live and aria-hidden, and allow users to navigate through slides using arrow keys or other keyboard controls.

## **3. Testing Tools and Techniques Used**

### **3.1 Automated Testing**

* Tools: Lighthouse (Chrome DevTools), Axe Browser Extension
* Findings:
  + Detected missing alt attributes.
  + Highlighted issues with contrast ratios.

### **3.2 Manual Testing**

* Tested keyboard navigation and focus states.
* Verified screen reader experience using NVDA and VoiceOver.

## **4. Accessibility Action Plan**

### **Immediate Tasks**

1. Add alt attributes to all images.
2. Define and test focus states for all interactive elements.
3. Enhance the language switcher to dynamically update the lang attribute.

### **Medium-Term Improvements**

1. Update link text and add ARIA labels to provide better context.
2. Implement a "Skip to Content" link.
3. Improve contrast ratios for all elements to meet WCAG AA standards.

### **Long-Term Enhancements**

1. Make the certificates carousel fully keyboard-accessible and screen reader-friendly.
2. Regularly audit the site with updated accessibility tools.
3. Train developers to incorporate accessibility into the development process.

## **5. Conclusion**

The front-end design demonstrates a strong commitment to accessibility through semantic structure, responsive design, and keyboard navigation support. Addressing the highlighted areas for improvement will ensure compliance with WCAG 2.1 standards and provide an inclusive experience for all users.