

The Web Content Accessibility Guidelines have 4 principles: Perceivable, Operable, Understandable, and Robust. The CAL test code I have sent in follows these principles in the following ways:

Summary:

Perceivable:

WCAG 1.1.1: Screen reader support via labels tied to input fields using AssociatedControlID.

WCAG 1.3.1: Grouping related items using fieldset and legend elements, and using aria-live="assertive" to notify users of validation errors.

WCAG 1.3.4: The content is viewable on different screen orientations/resolutions, ensuring accessibility across devices.

Operable:

WCAG 2.1.1: Uses standard controls (e.g., TextBox, RadioButtonList) that are fully operable by keyboard.

WCAG 2.3.2: No flashing content, ensuring no risk of seizures.

WCAG 2.4.3: The form focus is managed to ensure meaningful navigation order, focusing on validation errors when they occur.

WCAG 2.4.7: Keyboard focus is always visible for user interface components.

WCAG 2.4.11: No components are hidden due to author-created content.

Understandable:

WCAG 3.3.1: Clear error messages for required fields, guiding users through form completion. Javascript automatically focuses validation errors for the screen reader.

WCAG 3.3.3: Validators like RequiredFieldValidator, RegularExpressionValidator, and CompareValidator provide real-time input validation feedback.

WCAG 3.3.6: The user has the opportunity to correct uncaught errors after submission, errors are detected.

Robust:

WCAG 4.1: Standard HTML and ASP.NET controls ensure compatibility with assistive technologies, future-proofing the site.

Aria attributes such as aria-labelledby, aria-label, and aria-live ensure the content is accessible by present and future assistive technologies.

Long version:

Perceivable - Content must be presentable to users in different ways, e.g. using colorblind specific palettes, allowing alt text for screen readers.

- **WCAG 1.1.1:** *All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below.*

E.g. AssociatedControlID allows screen readers to read out form labels along with their input fields.

```
<asp:Label AssociatedControlID="txtEmail" Text="Email (Required): "
runat="server" />
```

- **WCAG 1.3.1:** *Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.*

E.g. using the aria-live="assertive" property to notify users of validation errors in the registration page. I also added a javascript section to scroll up to the errors section and automatically read it aloud.

```
<asp:ValidationSummary ID="ValidationSummary1" aria-live="assertive" />
```

Also E.g. using Fieldsets and Legends to group related items together.

```
<fieldset aria-labelledby="lblLevelOfStudy">
<legend id="lblLevelOfStudy">Level of Study (Required):</legend>
```

- **WCAG 1.3.4:** *Orientation. Content does not restrict its view and operation to a single display orientation, such as portrait or landscape, unless a specific display orientation is essential.*
Website is viewable and operable at multiple resolutions and on multiple devices.

Operable - User interface components and navigation must be operable (e.g., making sure functionality is available from a keyboard).

- **WCAG 2.1.1:** *All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.*

I used standard controls, e.g. TextBox, DropDownList, RadioButtonList, CheckBoxList, so the website is functionally operable by keyboard.

- **WCAG 2.3.2:** *Web pages do not contain anything that flashes more than three times in any one second period.*

There are no intentional flashes on the web page. Dark mode browser add ons may affect this.

- **WCAG 2.4.3:** *If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability.*

When validation fails, users are moved to the ValidationSummary area and it is focused so that screen readers will automatically read the errors.

- **WCAG 2.4.7:** *Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.*

If Tab is pressed it cycles through important information and arrow keys can be used to navigate whatever the tab button does not select.

- **WCAG 2.4.11:** *When a user interface component receives keyboard focus, the component is not entirely hidden due to author-created content.*
No user interface components are hidden by author created content.

Understandable - Information and the operation of the user interface must be understandable (e.g., making text content readable and predictable).

- **WCAG 3.3.1:** *If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text.*
Error messages are clear in their explanation of what happened. E.g.
`ErrorMessage="Email is required", ErrorMessage="Please select your international student status."`
- **WCAG 3.3.3:** *If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.*

The use of RequiredFieldValidator, RegularExpressionValidator, and CompareValidator assists users by providing feedback on input in real-time

```
<asp:RegularExpressionValidator ID="revEmail" runat="server"
ControlToValidate="txtEmail"
ValidationExpression="^[\\w-\\.]+@([\\w-]+\\.){2,4}$"
ErrorMessage="Invalid email format" CssClass="text-danger" />
```

- **WCAG 3.3.6:** *For Web pages that require the user to submit information, at least one of the following is true:*
 - *Reversible*
 - *Submissions are reversible.*
 - *Checked*
 - *Data entered by the user is checked for input errors and the user is provided an opportunity to correct them.*
 - *Confirmed*
 - *A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.*

Submissions are reversible via navigating to the record in the List of submissions and clicking view. From there they can edit any information they wish to. The data is checked for some input errors such as email pattern validation and Consent name matching their first and last names. It does not check for other problems, such as numbers in the name (Which could actually be an indigenous word like *K̓alhwa7acw* (Hello in Squamish)).

Robust - Content must be robust enough to work with current and future technologies (e.g., maximizing compatibility with current and future user agents and assistive technologies).

- **WCAG 4.1:** *Maximize compatibility with current and future user agents, including assistive technologies.*
I used native HTML elements and standard ASP.NET server controls like TextBox, RadioButtonList, and CheckBoxList. It works well across multiple browsers.
I also used aria-labelledby, aria-label, and aria-live attributes to make my content more

accessible to present and future assistive technologies. E.g.

```
<fieldset aria-labelledby="lblLevelOfStudy">
```

```
<asp:ValidationSummary aria-live="assertive" />
```