[Web] Error Handling and Form Validation

Context

Any typical feature with a form; e.g. the user is presented with a few inputs and a CTA (submit, next, etc.).

Current UX

1. The user fills out the form input by input; the submit button is disabled on load.
2. All inputs are required unless otherwise noted as "optional".
3. When the user moves focus out of an input, for example by going to the next question, the input is considered "dirty" and validation messages are shown if the value is invalid, whether the user entered anything or not.
4. Validation starts right away on all fields, so an error will display as soon as the user starts typing on any field with a character minimum requirement (e.g. A user will get an error as soon as they start typing into the zip code field because they have not entered the required 5 numeric characters).
5. When the user finishes entering values for all required inputs and all pass validation the submit button is enabled.

Problem

This approach is not the standard for web, has some a11y concerns, and can be improved for all users.

Proposed UX

See the design POC demonstrating this proposed UX here.

1. The user fills out the form input by input. The submit button is always enabled.
2. All inputs are required unless otherwise noted as "optional". Form instructions at the top of the form are provided to let users know whether all field are required, or that all fields are required unless marked “optional” - see more about this below in the sections called **Co ntent** and **Other A11y Considerations.**
3. When the user moves focus out of an input *where they have entered a value*, for example by going to the next question, the input is considered "dirty" and validation messages are shown if the value is invalid.
    - e.g. If a user only entered 2 characters for a zip code and then moves focus, an in-line error will display.
    - If a user moved focus from a required field where they have *not* input a value, an in-line error will *not* display.
4. Validation will not always start right away on a field, and will depend on the requirements for that field. We will move to more *timely* valida tion that allows a user to meet the minimum character requirement before displaying an error.
    - See more about timely validation in the section below called **Timely Validation.**
5. When the submit button is clicked:
    - The form is considered dirty, so all inputs show validation messages, including required fields that have been left blank.
    - An error banner is displayed at the top of the form. The banner will contain: general language that there are problems on the form, and an anchor list (<li>) listing of the inputs that are invalid.
    - The page 'pops' to the top of the form to show the error banner (no animation) and the banner is given focus.
    - All inputs are validated, and if all pass, then the form is submitted.
    - If any input validation fails, then the form is not submitted.
    - See content guidance for this in the section labeled **Content.**

6. When a user clicks on one of the anchor links, they are ‘popped’ to the invalid input (with no animation). The field is given focus and is activated

Content

Form Overview

Set expectations about the task with an *optional* description. See the first sentence below for an example. Some other ideas: list the numbers of form fields or summarize sections or steps of the form. Consult with CS and a11y to determine whether this is necessary, based on context.

Always display "All fields are required unless marked optional."

e.g. "Choose the amount you'd like to pay and your preferred payment method. All fields are required unless marked optional."

In-Line Errors

Define simple, straightforward errors or instructions with four copy "patterns".

1. "{Label} is required."
    - Use when field is left empty.
    - Example: *“First name is required.“*
2. "Invalid {label}"
    - Use when entered information includes invalid characters or formatting.
    - Example: *“Invalid email address”*
    - Error should not include period as it's not a full sentence.
3. "{Label}s must match."
    - Use in confirmation / re-entry scenarios when entries don't match.
    - Example: *“Passwords must match."*
4. "{Label} must be at least {minimum} {characters, digits}."
    - Use when entry does not meet the minimum character or digit requirement.
    - Example: *“Full name must be at least 2 characters.“*
    - Common fields (e.g. ZIP, phone, credit card) should be locked at the maximum character or digit count, eliminating the need for a "cannot be more than...." variation.
    - Fields with unexpected / uncommon limits are exceptions and should display error text based on the specific limitation (see "weight (lbs)" example in Timely Validation section below).

Error Overview

Surface multiple errors at top of form when the CTA is clicked to lend visibility into all required actions. On web, this will be done using an alert container. In the list of errors, use the field name rather than the in-line error itself to avoid repetition and confusion. The screenreader should read out the in-line error text when the user focuses on the invalid field.

e.g. Headline: Review the following fields for missing or invalid information:

Body: Go to {field name}

Screen reader experience considerations

In the error overview banner, we want a screen reader user to understand the extent of the problems to be fixed so they can navigate appropriately. If the list of errors is coded as an <li>, the screen reader will readout “list of __ items” before starting to read the links, giving the user an idea of the task ahead to fix their errors. As the user fixes errors in the form, the error overview banner should update dynamically to only show the remaining errors on the form.

Once focus is moved to the invalid field, we want the screen reader user’s experience to feel natural and for them to be informed that the field contains an error and what the error is. In addition, we want the screen reader user to be able to navigate to invalid fields easily in the case that there is more than one invalid field that they must fix. We should use aria-describedby to point to the error message. This will make it possible for users to hear the field label and then they should hear the error message immediately. Users who came from the error message will realize they arrived where they wanted to be and those who are navigating down the page may need to be reminded of the error.

Other A11y Considerations

Form Overview

See content guidance above.

When moving the user to a different part of the screen (showing the error banner at the top after hitting the CTA, or focusing them on an invalid field after clicking a link in the error banner), we want to ‘pop’ them there with no animation instead of scrolling them there with animation. This should work similarly to how a standard anchor link works on web.

Timely validation

Different form fields will have different rules that must be met for a valid response. This could come in the form of restrictions (e.g. only alphanumeric characters allowed), requirements (e.g. must contain @____.com), character minimums (e.g. at least 5 characters required), or character maximums (e.g. no more than 70 characters accepted). Each one of these form field rules requires different timing of validation for the best user experience.

We should validate for restrictions, requirements, and character maximums while the user is still entering information (that is, as soon as they start typing). We should validate for character minimums on blur (that is, when they tap out of the field or submit their entry). If the user taps out of the field before meeting the minimum character requirement, we can initiate validation and display an error.

The following examples describe sample form field prompts, their rules (requirements, restrictions, minimums and maximums), and when the entry should be validated.

- Example 1: Form field prompt with no minimum character requirement: "Weight (lbs)"
  - *Form field restrictions:* Entry must be a number. Number must be between 1-9999. No character minimum. Character maximum of 4.
  - *When should the entry be validated?* We can start validating the user's entry as soon as they start typing. If the member enters a letter or enters more than 4 numeric characters, an error should be displayed.
- Example 2: Form field prompt with a minimum character requirement: "Phone number)“
  - *Form field restrictions:* Entry must be a number. Character minimum and maximum of 10 digits.
  - *When should the entry be validated?* The field can be validated for restrictions (i.e. if the user enters a letter instead of a number) right away, but should only be validated for the minimum once the user has entered 10 digits. If the user taps out of the field before entering the required 10 characters, an error should be displayed.