Interface Design (Web Forms)

**7 Goals of UI design:**

1. Path to completion
   * Primary goal
   * Every input requires consideration and action
   * Remove any unnecessary inputs
   * Provide a clear path (better to have aligned vertical)
   * Enable smart defaults
   * Enable flexible inputs (different phone number formats)
   * For long forms, save progress and show (tabs on the left side showing progress)
2. Label Alignment
   * Three choices. The correct type depends on the system.
     1. Top Aligned
        + When data asked for is common (familiar).
        + Minimize time to completion (A)
        + Requires more vertical space (D)
        + Spacing or contrast is vital for the best scanning by user
     2. Left Aligned
        + When data required is unfamiliar
        + Makes it easy to scan labels (A)
        + Reduced vertical space (A)
        + Less clear association between labels and fields (D)
        + Tougher to design layout (D)
     3. Right Aligned
        + Clear association between label and field
        + Requires less vertical space (A)
        + Reduced readability (D)
        + Fast completion times
   * Eye tracking data
     1. Left-Aligned:
        + Easily associated labels with proper input fields
        + Excessive distances can make it hard to read
     2. Right-Aligned:
        + Reduced overall number of fixations by half
        + Form completion is much faster
     3. Top-Aligned:
        + Can capture both label and field with single eye movement
        + Fastest completion time
   * Best Practices
     1. For reduced times & familiar data: Top-Aligned
     2. When vertical space is constrained: Right-Aligned
     3. For unfamiliar data entries: Left-Aligned
3. Proper Validation
   * Provide real-time feedback
   * Suggest valid inputs
   * Help users stay within their limits
   * Indicate required fields
     1. Indicating optional fields is useful when there are few optional fields
     2. Indicating required fields is useful when there are few required but lots of fields
     3. Neither is useful when they are all required.
   * Best Practices
     1. Try to avoid optional fields
     2. If most are required, indicate optional.
     3. If most are optional, indicate required.
     4. Text is best, but \* works for required.
     5. Associate errors with labels.
   * **Field Lengths**
     1. They indicate how much data can be entered.
     2. Appropriate lengths provide enough space for inputs
     3. Random field lengths may add visual noise to a form
     4. Best Practices
        + Provide length appropriate for input or just enough for data that can be entered.
4. Help & Tips
   * Useful when:
     1. Asking for unfamiliar data
     2. To know why info is being requested
     3. There are recommended ways of entering data
     4. Certain data requests are optional
   * Carful not to overwhelm a form with them
   * To avoid overwhelming the client
     1. Provide user activated help so its there if they want it
   * Best Practices:
     1. Minimize number of help & tips
     2. Make the help visible and adjacent to a data request
     3. When lots of unfamiliar data, use dynamic help.
5. Form Organization
   * Grouping content logically provides structure to a form
   * Allows to scan information required at high level
   * Gives a sense of how information is related
   * Best Practices:
     1. Use relevant content grouping to organize forms
     2. Use minimum number of visual elements to group content (horizontal rules)
6. Actions
   * Equal Visual weight
   * Visual Distinctions
   * Primary action must stand out more than secondary
   * Best Practices:
     1. Avoid secondary actions if possible
     2. Ensure clear visual distinctions between secondary and primary
     3. Align primary action with input fields for a clear path to completion
     4. Label the actions in a natural language
     5. Almost never use a reset button
7. Gradual Engagement
   * Progressive Disclosure:
     1. Not all users require all information all at once
     2. Progressive disclosure provides additional options when needed.
     3. Best Practices:
        + Map progressive disclosure to prioritize user needs
        + Most effective when user initiated
        + Maintain consistent approach
   * Selection Dependent Input
     1. If a user clicks an option it may need more input which will be prompted on the event.
     2. Exposing Dependent Inputs:
        + Page level (Radio buttons)
          1. Requires additional steps
        + Finger Tabs
          1. Follow path to completion (Scan the line)
        + Horizontal Tabs
        + Group selector
        + Expose below or within
          1. Can be confusing
        + Inactive until selected
          1. Association between primary selection is impaired
        + **Maintain clear relationship between initial selection options**
        + **Clearly associate additional inputs with their trigger**
        + **Avoid “jumping” which disassociates initial selection options.**

**Design Principles**

* Minimize the pain
* Illuminate the path to completion
* Consider the context (familiar vs foreign)
* Ensure consistent communication