**Form Design Best Practices (UX)**

**Layout:**   
  
Group form elements logically. If, for example, you are asking for user information, do not ask about banking information between first name and last name.   
  
If an order that is not strictly logical is engrained in real life, follow the real life example. For example, in the US, street address to apartment number is getting more specific, while apartment number to city to street is getting less specific, but putting fields in apartment -> street -> city would be very confusing.   
  
Inside the larger logical groups, sub-group the fields by keyboard or mouse input. Do not switch from mouse to keyboard and back unless required by logic rules above.

**Accessibility:**

Keep accessibility in mind at all times. Keep the form simple and if the form includes conditional fields, consider breaking the form into multiple sections based on those conditions.

Proper contrast at all times.

Do not rely on color to communicate status.  
  
**Labels:**

Use simple labels. Show the labels as close to the respective fields as possible, within design constraints.   
  
Ideally labels will align left with the fields they belong to, label above the field. If that isn’t possible, the label should be placed to the left of the field. Unusual label alignment (centered, right) should be avoided.   
  
 The title of the form is also a label. Make sure it is descriptive.

If you are asking the user to invent a password, include a description of all pertinent password rules as a note near the password field(s).

Any form element that requires specific format, include those rules as a hint below the field. Consider eliminating all format rules if the user data can be reformatted by the server before saving.

Any placeholder that shows the field format ideal should be reiterated below the field as a hint.

**Form element choice:**

Use the appropriate field for the information you are expecting.

Us long space text fields sparingly. If you are expecting specific answers, use a select element rather than a long form text field.

**Privacy, and user hesitance**:

Edit your form with an eye toward user privacy. Only require information that is needed for the operation. Consider removing personal identifying fields if they are unneeded.

Avoid long forms which may appear intimidating. If the form is longer than ~10 fields, consider breaking the fields into steps. Save information between steps and include a progress bar.

If your form asks for something that the user must retrieve (for example, a driver’s license), include instructions for the user at the beginning, listing all such items that are needed to complete the form.

Allow for auto-fill from user’s browser, but always allow auto-filled text to be edited.

If you use defaults, make sure the user can change them easily.

Password characters and other sensitive information should be replaced by bullets after enough time for the user to confirm the character is correct.

If the user may hesitate in giving information (social security number, for example) include an explanation as to why you need this information. This usually takes the form of a ‘Why do you need this?’ question and a tooltip explanation.

**Saving work and validation:**

Use inline validation when possible. If not possible, break the process into steps and validate info at the server while asking the user for further info to maximize the user’s time.

**Errors:**

Errors for the form in general should be near the form title.

Errors in specific fields should be shown near the field in error.

When writing error messages, write to assist the user in fixing the error, not to just point out they made an error. Avoid negative words.

**Buttons:**

Use a standard “Save” or “Submit” button, whatever is the common phrase for the website/app.

Do NOT include a cancel or reset button. If it must be included, make it visually different than the submit button and display it as far away as Submit as possible.