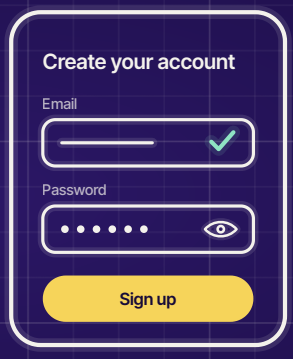
FREE CHECKLIST

The sign-up form checklist

12 checks



MobiStack

mobistack.co/free/signup-form-checklist

Every account your product gets passes through one form. Here are twelve checks to run on yours tonight, each from a standard or a platform's own guidelines.

Every source opened on 30 October 2026. The page with every link: mobistack.co/free/signup-form-checklist

01 Let them in first

If your app works without an account, let people use it before they sign up.

Source: Apple, App Review Guidelines 5.1.1(v): "If your app doesn't include significant account-based features, let people use it without a login." web.dev, Sign-up form best practices: "Avoid sign-in if you can."

02 Ask only for what you need now

Every extra field is a reason to stop. Ask for the rest later, when it is needed.

Source: Apple, App Review Guidelines 5.1.1(v): "Apps may not require users to enter personal information to function, except when directly relevant to the core functionality of the app or required by law."

03 Labels above the field, never only inside it

A hint inside the field disappears the moment people start typing.

Source: Nielsen Norman Group, Placeholders in form fields are harmful: "Disappearing placeholder text strains users' short-term memory." web.dev, Sign-in form best practices: "Don't use placeholders as input labels." and "It's probably best to put your labels above your inputs."

04 The right keyboard for every field

An email field opens the email keyboard; a phone field opens the number pad.

Source: web.dev, Sign-up form best practices: "use the appropriate `type` attribute to provide the right keyboard on mobile". Android Developers, Specify the input method type: "Always declare the input method for your text fields by adding the `android:inputType` attribute".

05 Let the phone and the browser fill it in

Tell the form what each field is, so saved details fill in by themselves.

Source: web.dev, Sign-up form best practices: "For sign-up forms use `autocomplete="new-password"` for new passwords", with `autocomplete="email"` for the email field and `autocomplete="username"` for usernames. W3C, WCAG 2.2 success criterion 1.3.5 (level AA), so that "browsers and other user agents can suggest and 'autofill' the right content".

06 Rules up front, not after

If a field has rules, show them before people type, not in the error after they submit.

Source: W3C, WCAG 2.2 success criterion 3.3.2 (level A): "Labels or instructions are provided when content requires user input.", most of all "if there are specific rules for correct input". Baymard Institute, guideline 676, names the cost: "Users may have to attempt multiple times to create a password that the site will accept, wasting valuable time and effort".

07 Length, not symbols

Ask for a long password. Don't force capitals, numbers or symbols, and don't force changes on a schedule.

Source: NIST SP 800-63B-4, section 3.1.1.2: passwords used alone need "a minimum of 15 characters in length" (eight when they are one step of a multi-factor sign-in); verifiers "SHOULD permit a maximum password length of at least 64 characters", "SHALL NOT impose other composition rules", "SHALL compare the prospective secret against a blocklist", and "SHALL NOT require subscribers to change passwords periodically".

08 Allow paste and password managers

Never block paste in a password field, and let password managers fill it.

Source: NIST SP 800-63B-4: "SHALL allow the use of password managers and autofill functionality" and "SHOULD permit claimants to use the 'paste' function when entering a password". W3C, WCAG 2.2 success criterion 3.3.8 names "support for password entry by password managers to reduce memory need" and "copy and paste to reduce the cognitive burden of re-typing". web.dev, Sign-up form best practices: "Allow password pasting".

09 Let them see the password

A show password control lets people check what they typed before they submit.

Source: NIST SP 800-63B-4: "SHOULD offer an option to display the password". web.dev, Sign-in form best practices: "You should add a Show password toggle to enable users to check the text they've entered."

10 Ask once

No "confirm your email" field. People who can see what they typed don't need to type it twice.

Source: web.dev, Sign-up form best practices: "Don't double up your inputs just to make sure users get their contact details right."

11 Errors next to the field, with the fix

Say what went wrong and how to fix it, right where it happened.

Source: Nielsen Norman Group, An error messages scoring rubric: "The error should be displayed next to the associated interface element(s)." and "The error message should describe a solution that is sufficient for the user to fix the problem." web.dev, Sign-up form best practices: "Show problems inline and explain how to fix them".

12 One tap, one account

The button presses in, says what it is doing, and ignores a second tap. Your server ignores a duplicate too.

Source: Apple, Human Interface Guidelines, Buttons: "Always include a press state for a custom button. Without a press state, a button can feel unresponsive, making people wonder if it's accepting their input." Baymard Institute, People still double-click online: "disable the submit button immediately after it has been clicked" and "code your back-end to ignore identical orders". "Do both if you want to be on the safe side."

Two more for iOS apps

- If people can create an account in your iOS app, they must be able to delete it in the app. Apple 5.1.1(v): "If your app supports account creation, you must also offer account deletion within the app."
- If your iOS app offers a third-party or social login, Apple 4.8 asks for an equivalent option that "limits data collection to the user's name and email address" and "allows users to keep their email address private".

Save it for your next release, and run it on your own sign-up tonight.

Sources, opened 30 October 2026

1. Apple, App Review Guidelines
   developer.apple.com/app-store/review/guidelines/
2. Apple, Human Interface Guidelines: Buttons
   developer.apple.com/design/human-interface-guidelines/buttons
3. web.dev, Sign-up form best practices
   web.dev/articles/sign-up-form-best-practices
4. web.dev, Sign-in form best practices
   web.dev/articles/sign-in-form-best-practices
5. Nielsen Norman Group, Placeholders in form fields are harmful
   www.nngroup.com/articles/form-design-placeholders/
6. Nielsen Norman Group, An error messages scoring rubric
   www.nngroup.com/articles/error-messages-scoring-rubric/
7. Android Developers, Specify the input method type
   developer.android.com/develop/ui/views/touch-and-input/keyboard-input/style
8. W3C, Understanding WCAG 2.2: Identify input purpose (1.3.5)
   www.w3.org/WAI/WCAG22/Understanding/identify-input-purpose
9. W3C, Understanding WCAG 2.2: Labels or instructions (3.3.2)
   www.w3.org/WAI/WCAG22/Understanding/labels-or-instructions
10. W3C, Understanding WCAG 2.2: Accessible authentication, minimum (3.3.8)
    www.w3.org/WAI/WCAG22/Understanding/accessible-authentication-minimum
11. NIST SP 800-63B-4, section 3.1.1.2: Password verifiers
    pages.nist.gov/800-63-4/sp800-63b/authenticators/
12. Baymard Institute, guideline 676: Informing users about password creation requirements
    baymard.com/guidelines/676-informing-users-about-password-creation-requirements
13. Baymard Institute, People still double-click online (4 February 2010)
    baymard.com/research-articles/people-still-double-click-online

Follow us for more like this on Instagram, Facebook and LinkedIn.

Made by MobiStack (mobistack.co). We take over, fix and build apps and web platforms.