

## Web Application Usability Testing Checklist

Here is checklist that we follow for usability of a web application for login module. Please note that the checklist items may vary to some degree based on web application.

### Make it clear where to login

When users come to a website or app, and already have an account it should be immediately clear where they go to login. Rather than providing a 'Login' or 'Sign in' link (the term 'Sign in' is probably more commonplace than 'Login', but users generally understand either) it's better to show the input fields, so that users can login directly from a page.

It's better to show login input fields on a page like Facebook, rather than just a link.

Sign In

Sign in with Twitter

Sign in with Facebook

Sign in with Google

Email

Password

Stay signed in

[Forgot password?](#)

Sign In

Cancel

Your credentials will be sent over a secure connection

or

[Click here to Register](#)

### Differentiate login from registration

On an increasing number of websites the login input fields (email and password) are very similar, if not exactly the same as the registration input fields. It's important therefore to clearly differentiate the two, and to minimise the chance of users accidentally attempting to login in via registration form. One way to do this is to get users only show one form at a time or just simply provide the registration link on which clicking, user will navigate to registration page to register him/herself.

The image shows a 'Sign In' form with a yellow border. On the left, there are three buttons: 'Sign in with Twitter', 'Sign in with Facebook', and 'Sign in with Google'. On the right, there are input fields for 'Email' and 'Password', a checkbox for 'Stay signed in', and a link for 'Forgot password?'. Below the input fields are 'Sign In' and 'Cancel' buttons. A small note at the bottom says 'Your credentials will be sent over a secure connection'. Below the form, there is a horizontal line with the word 'or' in a circle, and a rounded button that says 'Click here to Register'.

## Allow User to login with an external account (e.g. Facebook)

Why force users to have to remember another set of login details when it's now so easy to let them login via an external account, such as a Facebook, Google or LinkedIn account? Of course not everyone is likely to be happy doing this, but it's a great way to let users easily login to your website or app with an account that they use day in, day out.

This image is identical to the one above, showing a 'Sign In' form with social media options and a registration link. The 'Sign in with Facebook' button is highlighted with a yellow border.

## Use email address, rather than username

This is a particular usability frustration of mine – namely websites asking users to login with a username, rather than their email address. I have two main emails addresses (my personal and work addresses), but many, many different usernames for various websites. Since

usernames have to be unique, invariably a preferred username has been taken, so users end up registering with a new username that they are never going to remember. If your site or app does use usernames then like Twitter, at least allow users the option to login with their email address.

**Sign In**

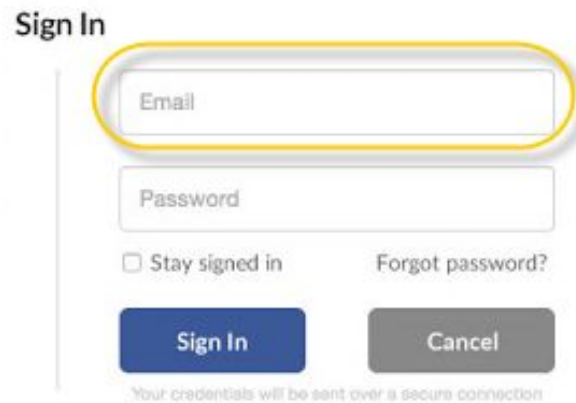
Email

Password

Stay signed in      [Forgot password?](#)

**Sign In**      **Cancel**

Your credentials will be sent over a secure connection

A sign-in form titled "Sign In" is shown. It features two input fields: "Email" and "Password". The "Email" field is highlighted with a yellow oval. Below the input fields, there is a checkbox labeled "Stay signed in" and a link labeled "Forgot password?". At the bottom, there are two buttons: "Sign In" (blue) and "Cancel" (grey). A small note at the very bottom states "Your credentials will be sent over a secure connection".

### **Let users see their password (if they want to)**

A common problem when users attempt to login is mis-typing their password. This is all too easy to do as the password field is of course masked. A useful feature is to allow users to see the password they have entered; (if they want to) by providing a show password checkbox. This checkbox should of course be unchecked by default (i.e. the password is always masked by default). This is especially useful for mobile login pages, as getting the wrong key is all too easy on a fiddly mobile keyboard.

**Sign In**

smith@mailinator.com

sm1k@123Klm

Stay signed in    [Forgot password?](#)

**Sign In**    Cancel

Your credentials will be sent over a secure connection

**Sign in with Twitter**

**Sign in with Facebook**

**Sign in with Google**

or

[Click here to Register](#)

## Tell users if Caps Lock is on

Another simple way to help users enter their password correctly is to warn them if their Caps Lock button is on.

**Sign In**

smith@mailinator.com

\*\*\*

Stay signed in    [Forgot password?](#)

**Sign In**    Cancel

Your credentials will be sent over a secure connection

Warning: CAPS Lock is on

It's a good idea to warn users when Caps Lock is on.

## Make it easy if users forget their password

Just as everyone sometimes forgets people's names, or their wallet, or their anniversary (I'd recommend you try to avoid this one!), users will forget their password. Let me repeat that for any security analysts out there. PEOPLE WILL FORGET THEIR PASSWORD.

It's therefore very important that if users do forget their password (and they will) that this is well handled by the login process. As a starter always have a clear 'Forgotten your password?' link for users to use. The best thing to do is to send a reset password link via the registered email address. Also, make sure the reset password email is as instantaneous as possible.

The diagram illustrates a user flow for password recovery. On the left, a login form contains fields for 'Email' and 'Password', a 'Stay signed in' checkbox, and a 'Forgot password?' link highlighted with a yellow circle. Below these are 'Sign In' and 'Cancel' buttons. A red arrow points from the 'Forgot password?' link to a secondary form on the right. This secondary form is titled 'Forgot password?' and features a red asterisk and the text '\* Required Fields' above an 'Email' input field. It also includes a 'Back to Login' link and a 'Submit' button. At the bottom of the login form, a small note reads 'Your credentials will be sent over a secure connection'.

## Warn users before locking their account

To prevent brute force attacks user accounts are often temporarily locked out after a number of failed login attempts. This is of course a necessary security measure, but be sure to warn users before their account is to be locked. For example, if it will be locked for 10 mins after 5 unsuccessful login attempts, warn the user after the third attempt that their account will be locked for 10 mins follow 2 more unsuccessful login attempts.

## Keep users logged in

Gone are the days of accessing the Internet using a public computer so rather than having a 'Keep me signed in' option it's better just to automatically keep users logged in to a website or app for a set period (unless of course security is a real issues, such as banking apps and websites). Of course sometimes a computer is shared (such as the family laptop), so it's still very important that users can easily login as a different user if they need to.

## Remember users when they return

When a user does have to re-login to their account, ensure that their details are retained where possible. Ensure that browsers are able to pre-populate fields (such as email address) and if possible remember their account details, so that users only have to enter their password.



**John Miller**  
**johnm88@mailinator.com**

Password

[Forgot password?](#)

**Sign In**