

Login, Registrierung, Passwort

Projektleitung

Geschäftsführung

IT

Recht

So sicherst du den Zugang zu deinen Kernfunktionen für alle

Weil zentrale Funktionen wie Login, Registrierung und Passwort-Reset oft die größten Frustmomente für Nutzer:innen mit Einschränkungen sind.

Ohne einen barrierefreien Zugang zu diesen Kernprozessen verlieren Menschen ihre digitale Selbstständigkeit – und du verlierst Kund:innen, Leads oder Mitglieder.

Wann musst du das machen?

Immer dann, wenn du steuern willst, wie Nutzer:innen mit deiner Plattform interagieren, sich anmelden oder zurückkehren. Konkret bei:

- Allen geschützten Bereichen wie Kundenkonten, Mitgliederportalen oder internen Dashboards.
- Shops, SaaS-Angeboten, Bewerbungsportalen und Buchungssystemen.

Welches Gesetz verlangt das?

- EAA / BFGS: Fordern die barrierefreie Gestaltung von digitalen Dienstleistungen.
- WCAG 2.1: Insbesondere die Kriterien 1.3.1 (Info and Relationships), 1.3.5 (Identify Input Purpose), 2.4.6 (Headings and Labels), 3.3.1 (Error Identification), 3.3.2 (Labels or Instructions) und 4.1.3 (Status Messages).
- DSGVO Art. 25 (Privacy by Design): Verlangt Datensparsamkeit und Sicherheit auch in der Benutzeroberfläche.

Der Prozess im Detail

1 Phase 1: Setze Felder und Beschriftungen sauber um

Ein barrierefreies Formular beginnt mit einer sauberen, semantischen Struktur und klaren Anweisungen. Du kennst das ja schon aus SOP 19. Und für Newsletter-Anmeldungen kommen noch einmal weitere technische Anforderungen hinzu:

- Achte darauf, dass jedes Feld sein eigenes Label hat: Jedes Eingabefeld muss ein permanent sichtbares <label> haben (z. B. „E-Mail-Adresse“, „Passwort“).
- Ermögliche Autofill: Unterstütze Browser beim automatischen Ausfüllen durch die Verwendung des autocomplete-Attributs (z. B. autocomplete="email").
- Biete Eingabehilfen an: Mache Anforderungen, wie die an ein Passwort, direkt sichtbar (z. B. „Mind. 8 Zeichen, 1 Zahl“).
- Sorge für eine logische Reihenfolge: Ordne die Felder logisch an, z. B. zuerst die E-Mail-Adresse, dann das Passwort.

2

Phase 2: Stelle die Tastatursteuerung und den Fokusindikator sicher

Die Bedienung ohne Maus muss einwandfrei funktionieren. Da gibt es kein Wenn und Aber. Das bedeutet:

- Vollständige Tab-Erreichbarkeit: Alle Felder und Buttons müssen per Tab-Taste in einer logischen Reihenfolge erreichbar sein, ohne dass der Fokus unerwartet springt.
- Sichtbarer Fokusindikator: Das jeweils aktive Element muss immer einen klaren, sichtbaren Rahmen haben. Ein outline: none im CSS ist tabu.
- Verwende Echte Buttons: Ein Login-Button muss als `<button type="submit">` implementiert sein, damit er mit der Enter-Taste ausgelöst werden kann.
- Biete ein Button zum Anzeigen des Passworts an: Biete optional einen Button zum Anzeigen des Passworts an, der mit „aria-pressed“ und Tastatursteuerung korrekt umgesetzt ist.

3

Phase 3: Gestalte deine Fehler- und Statusmeldungen klar

Nutzer:innen müssen sofort und verständlich erfahren, ob eine Aktion erfolgreich war oder was schiefgelaufen ist. Die Fehlermeldungen müssen sichtbar, verständlich und per Screenreader erfassbar sein.

- Schreibe spezifische Fehlermeldungen: Statt einer allgemeinen Fehlermeldung, gib konkrete Hinweise wie „Bitte gib deine E-Mail-Adresse ein“ oder „Das Passwort stimmt nicht – bitte erneut versuchen“.
- Prüfe Passwort-Vorgaben: Melde klar, wenn die Passwort-Anforderungen nicht erfüllt sind.
- Kommuniziere den Erfolg deutlich : Eine erfolgreiche Anmeldung sollte durch eine Weiterleitung und eine für Screenreader wahrnehmbare Statusmeldung (aria-live="polite") bestätigt werden.

4

Phase 4: Gestalte den %22Passwort vergessen%22-Prozess barrierefrei

Dieser Prozess ist oft eine Fehlerquelle und muss genauso zugänglich sein wie der Login selbst.

- Link klar beschriften: Der „Passwort vergessen“-Link muss klar als solcher benannt und per Tab erreichbar sein.
- Verständliche E-Mail: Die E-Mail zur Wiederherstellung muss einen klar beschrifteten Button enthalten (z. B. „Neues Passwort setzen“) und einen Textlink als Backup anbieten. (Beachte dazu auch SOP 17)
- Klarer Prozess: Der gesamte Ablauf zum Setzen eines neuen Passworts muss verständliche Vorgaben und eine saubere Fokussteuerung haben.
- Vermeide Pop-ups oder Overlays , die vom Screenreader nicht erkannt werden können.

5

Phase 5: Sicherheit, Datenschutz und Selbstbestimmung gewährleisten

Barrierefreiheit, Datenschutz und Datensicherheit gehen Hand in Hand. In der Praxis heißt das:

- Verlinke deine Datenschutzerklärung: Die DSGVO-konforme Datenschutzerklärung muss vor dem Login einfach erreichbar sein.
- Datensparsamkeit: Fordere bei der Registrierung nur die absolut notwendigen Felder (E-Mail & Passwort) als Pflichtfelder.
- Mache 2FA zugänglich: Wenn eine Zwei-Faktor-Authentisierung angeboten wird, muss diese barrierefrei sein (z. B. eine Alternative zur reinen App-Nutzung per E-Mail anbieten).
- Kommuniziere Timeouts: Informiere Nutzer:innen, wenn sie aufgrund von Inaktivität abgemeldet wurden.

Ergebnis

Am Ende dieser SOP hast du:

- Ein Login- & Registrierungsformular, das mit Maus, Tastatur und Screenreader funktioniert.
- Hilfreiche Hinweise, verständliche Fehlermeldungen und klare Erfolgsmeldungen.
- Eine barrierefreie Passwort-Wiederherstellung.
- Sichergestellt, dass keine Nutzer:innen ausgeschlossen werden – auch nicht im sensibelsten Bereich deiner Anwendung.

Um Login, Registrierung & Passwort-Reset blitzschnell barrierefrei zu gestalten, stell dir bei jedem Schritt die zentrale Frage:

„Kann das JEDE:R mit Tastatur und Screenreader problemlos nutzen?“

Wenn die Antwort NICHT SOFORT „Ja“ ist, dann liegt hier dein Optimierungspotenzial.

Prüfe konkret:

- Labels & Hinweise: Sind alle Felder klar und dauerhaft beschriftet? Erklären sich Passwortheingaben von selbst?
- Fokus & Bedienung: Kann man alles mit der Tab-Taste erreichen? Sieht man immer, wo man gerade ist? Lässt sich der „Login“-Button mit Enter auslösen?
- Fehler & Feedback: Sind Fehlermeldungen klar und spezifisch? Bekommt man sofort und deutlich mit, ob etwas geklappt hat oder nicht?
- Passwort vergessen: Ist dieser Weg genauso einfach zu finden und zu durchlaufen wie der Login selbst?

Fragen & Antworten

Wie testen wir am besten, ob unsere Login- und Registrierungsformulare wirklich barrierefrei sind?

Am effektivsten ist eine Kombination aus manuellen und automatisierten Tests.

- Manuelle Tests: Führe die Prozesse komplett ohne Maus durch – nur mit der Tastatur (Tab-Taste, Enter, Leertaste) . Prüfe, ob der Fokus sichtbar ist und logisch springt. Nutze einen Screenreader (z.B. NVDA für Windows, VoiceOver für macOS/iOS, TalkBack für Android) und versuche, dich damit anzumelden oder zu registrieren. Achte darauf, ob alle Anweisungen und Fehlermeldungen korrekt vorgelesen werden. Lass auch Testpersonen mit tatsächlichen Einschränkungen die Prozesse durchlaufen.
- Automatisierte Tests: Nutze Browser-Erweiterungen wie Lighthouse (im Chrome DevTools integriert), AXE DevTools oder Wave, um erste Hinweise auf technische Barrieren zu erhalten. Diese Tools erkennen oft fehlende Labels, unzureichende Kontraste oder Probleme mit ARIA-Attributen.

Was ist mit Captchas oder reCAPTCHA? Wie gestalten wir die barrierefrei?

Captchas sind oft eine große Barriere – wir behandeln sie ausführlich in der nächsten SOP 29. Aber hier schon einmal eine Kurzfassung: Wenn du sie einsetzen musst, wähle immer die zugänglichste Variante und biete Alternativen an:

- ReCAPTCHA v3 (unsichtbar): Bevorzuge unsichtbare Captchas, da sie für Nutzer:innen oft gar nicht bemerkbar sind. Allerdings kann es hier zu Fehlklassifizierungen kommen.
- Audio-Captchas: Biete immer eine Audio-Option an, wenn visuelle Captchas (Text oder Bilder) genutzt werden.
- Alternativen: Ziehe Alternativen in Betracht, die weniger störend sind, wie z.B. Honeypot-Felder (versteckte Felder, die nur von Bots ausgefüllt werden) oder zeitbasierte Felder (Prüfung, ob das Formular zu schnell ausgefüllt wurde). Im Idealfall sollte ein menschliches Nutzerverhalten ausreichen, um Bots zu erkennen.

Unsere Designer:innen haben eine spezielle Idee für den Fokusindikator, die von den Standardeinstellungen abweicht. Ist das erlaubt?

Ja, absolut! Der Fokusindikator muss nicht der Browser-Standard sein. Wichtig ist, dass er immer klar und deutlich sichtbar ist und sich vom Hintergrund und dem Element selbst abhebt .

- Mindestanforderungen: Es muss ein klarer Umriss oder eine deutliche Farbänderung sein, die nicht nur auf subtilen Änderungen im Kontrast beruht. Ein outline: none; ist weiterhin ein No-Go.
- Kreativität erwünscht: Ein dickerer Rahmen, ein leuchtender Schein (box-shadow), eine inverse Farbe des Elements oder eine Kombination sind oft sogar besser als der Standard, solange die Sichtbarkeit gewährleistet ist. Wichtig ist, dass der Fokus den Nutzer:innen jederzeit verrät, wo sie sich gerade im Formular befinden.

Wir haben ein dynamisches Anmeldeformular, das sich je nach Nutzertyp ändert (z.B. Privatkunde/ Geschäftskunde). Wie stellen wir hier Barrierefreiheit sicher?

Bei dynamischen Formularen ist die Konsistenz der Schlüssel.

- ARIA Live Regions: Wenn Inhalte oder Abschnitte dynamisch nachgeladen oder verändert werden, informiere Screenreader-Nutzer:innen darüber mit aria-live="polite" oder aria-live="assertive" für kritische Änderungen.
- Fokus-Management: Nach einer dynamischen Änderung sollte der Fokus idealerweise auf das erste interaktive Element im neu geladenen oder relevanten Bereich gesetzt werden, damit Nutzer:innen sofort weiterarbeiten können, ohne sich neu orientieren zu müssen.
- Logische Reihenfolge beibehalten: Auch wenn Felder dynamisch erscheinen oder verschwinden, muss die Tab-Reihenfolge logisch bleiben und darf nicht springen.

Wie gehen wir mit Fehlermeldungen um, wenn ein Feld sowohl visuell als auch per Screenreader hervorgehoben werden soll?

Verbinde die Fehlermeldung direkt mit dem fehlerhaften Feld.

- aria-describedby & aria-invalid : Setze aria-invalid="true" auf das fehlerhafte Eingabefeld. Verknüpfe dann die konkrete Fehlermeldung visuell und per aria-describedby mit dem Feld.
- Beispiel: `<input type="email" id="email" aria-invalid="true" aria-describedby="email-error">` und `Bitte geben Sie eine gültige E-Mail-Adresse ein.` . Das role="alert" sorgt dafür, dass die Fehlermeldung sofort vorgelesen wird.
- Fokus: Setze den Fokus optional auf das erste fehlerhafte Feld, nachdem das Formular abgesendet wurde, damit Nutzer:innen sofort wissen, wo sie korrigieren müssen.