

Prototyping Projektdokumentation

Name: Albert Gstöhl

E-Mail: gstoealb@students.zhaw.ch

URL der deployten Anwendung: <https://ags-portfolio.netlify.app>

Einleitung

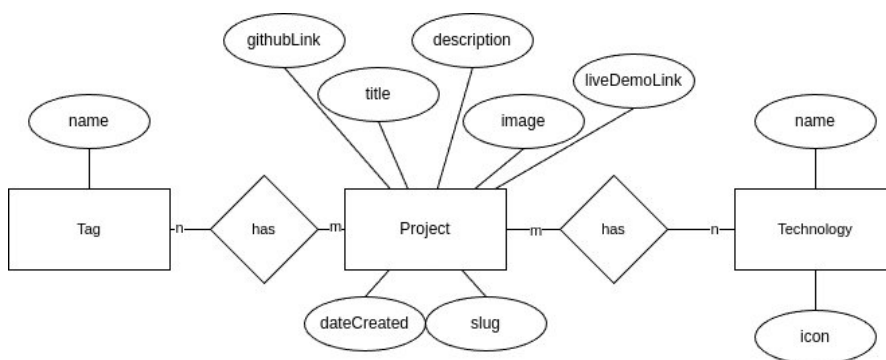
Die Anwendung "Portfolio" ist eine Webanwendung, die als zentrale Plattform zur Darstellung von Projekten dient. Sie wurde entwickelt, um persönliche Projekte für Bewerbungen und zur Veröffentlichung von Open-Source-Ideen übersichtlich darzustellen. Für die visuelle Darstellung wurde [Tailwind CSS](#) und [shadcn-svelte](#) verwendet.

Grundfunktionen:

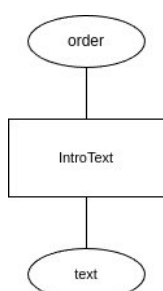
1. **Intro-Seite mit Typewriter-Effekt:** Die Startseite zeigt Texte aus einer MongoDB-Collection in einem Typewriter-Stil an.
2. **WebGL-Rendering:** Ein interaktiv gerendertes 3D-Laptop-Modell visualisiert dynamisch die Icons der in den Projekten verwendeten Technologien. Ein Klick auf ein Icon leitet zur gefilterten Projektübersicht weiter. Die Icons werden von devicon.dev verwendet.
3. **Projektübersicht und Filter:** Detaillierte Ansicht der Projekte mit Filteroptionen basierend auf Technologie.
4. **Lebenslauf-Seite:** Ermöglicht den monatlich aktualisierten Download des Lebenslaufs. Das PDF wird durch einen Proxy-Endpunkt aus einem separaten Repository bezogen, welches per Latex den Lebenslauf generiert und als Release des Repos ablegt.
5. **Kontaktformular:** Implementiert mit Netlify, um Nachrichten zu senden.
6. **Admin:** Authentifizierung per GitHub OAuth, wodurch ein Administrator neue Projekte erstellen, bearbeiten und löschen kann.

Datenmodell

Collection Projects:



Collection Intro (Für den Typewriter-Text für die Intro Seite):



Beschreibung der Anwendung

Intro Seite

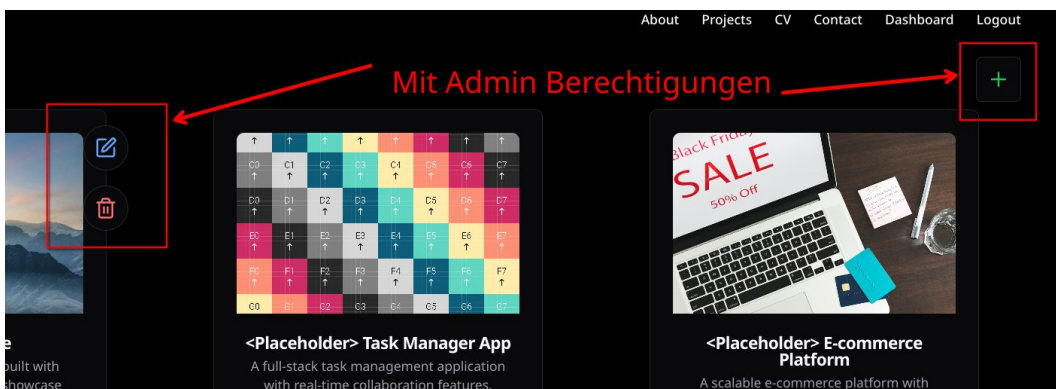
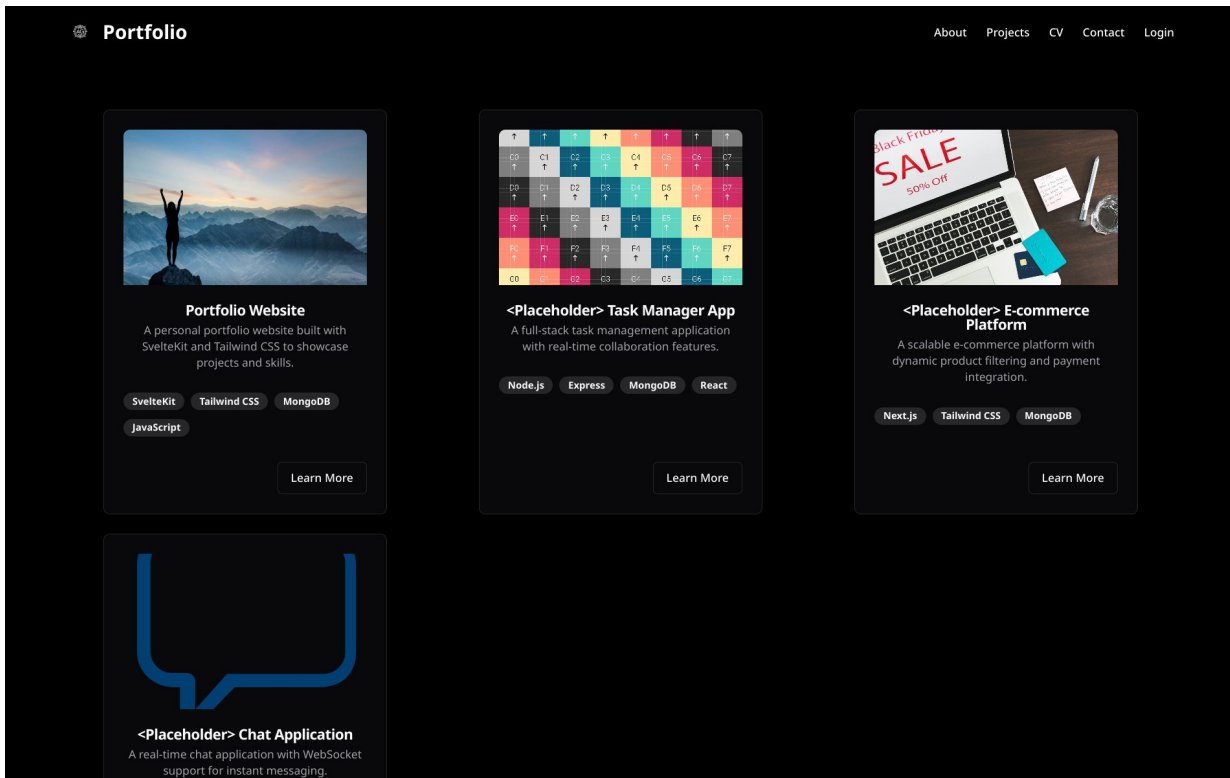
Route: /



Auf der Intro-Seite wird ein interaktiver Laptop dargestellt, der dynamisch Icons der in den Projekten verwendeten Technologien anzeigt. Diese Icons werden aus der projects Collection geladen. Die Seite enthält zudem einen Typewriter-Effekt, der Text aus einer MongoDB-Collection auflistet. Durch Klick auf ein Icon wird der Benutzer direkt zur gefilterten Projektübersicht weitergeleitet, die Projekte filtert, die mit der entsprechenden Technologie erstellt wurden. In einem Store wird gespeichert, ob das Intro schon angeschaut wurde, falls ja wird es nicht mehr gestartet wenn die Seite geladen wird.

Dateien:

- routes/+page.svelte
- routes/+page.server.js
- stores/introStore.js
- lib/SceneOne.svelte
- components/Laptop.svelte



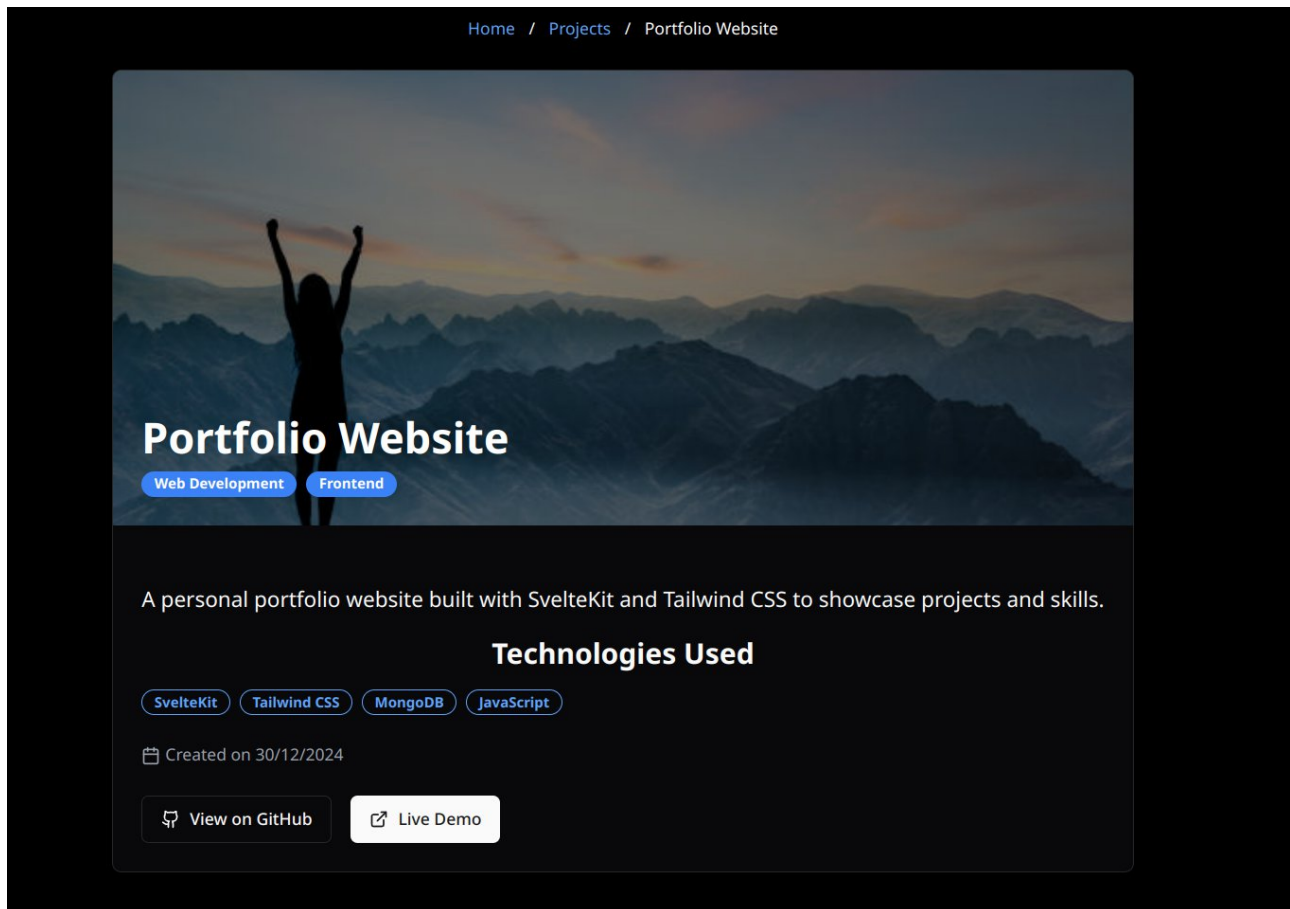
Auf der Projects-Seite wird eine Übersicht der Projekte aus der Projects-Collection dargestellt. Die Projekte werden in Form von Project Cards angezeigt, die den Titel, eine Kurzbeschreibung und die verwendeten Technologien enthalten. Durch einen Klick auf die aufgeführten Technologien kann die Projektübersicht dynamisch gefiltert werden. Über Learn More-Buttons gelangt der Benutzer zur Detailseite des jeweiligen Projekts. Ist der Admin authentifiziert, stehen zusätzliche Funktionen zur Verfügung, um Projekte zu bearbeiten, neue Projekte zu erstellen oder bestehende Projekte zu löschen (Löschfunktion triggert von dieser Seite aus den Endpunkt `/admin/[slug]?/delete`).

Dateien:

- routes/projects/+page.svelte
- routes/projects/+page.server.js
- components/ProjectCard.svelte

Detallierte Projektseite

Route: /projects/[slug]



Auf der Detailseite eines Projekts wird ein einzelnes Projekt aus der Projects-Collection detailliert dargestellt. Die Seite zeigt den Titel, eine ausführliche Beschreibung, das Erstellungsdatum sowie die verwendeten Technologien des Projekts an. Technologien werden als klickbare Badges angezeigt, mit denen die Projektübersicht gefiltert werden kann. Ausserdem gibt es Buttons, um direkt zur GitHub-Seite oder zur Live-Demo des Projekts zu gelangen. Ein Slug ist dabei ein URL-freundlicher, eindeutiger Bezeichner, der verwendet wird, um das Projekt in der Datenbank und der URL zu identifizieren.

Dateien:

- routes/projects/+page.svelte
- routes/projects/+page.server.js

Projekt erstellen

Route: /admin/create

Create New Project

Title
Test

Description
TESTEST

Image
Picture.png

GitHub Link
https://imgs.search.brave.com/pHUqOICO_uZNY6NKBcumMGeu1gdHsIm

Live Demo Link
https://imgs.search.brave.com/pHUqOICO_uZNY6NKBcumMGeu1gdHsIm

Technologies
Name: asdfasdf Icon class: +

Tags
Add a tag: asdfasdf +

Create Project

Die Create-Seite ermöglicht Administratoren nach der Authentifizierung, ein neues Projekt zu erstellen. Dabei werden alle relevanten Daten des Projekts in einem Formular erfasst und an den Server übermittelt (Endpunkt /admin/create?/add). Es können Bilder per Upload hinzugefügt werden (in “projects” Collection base64 encoded abgelegt).

Dateien:

- routes/admin/create/+page.svelte
- routes/admin/create/+page.server.js
- components/ProjectForm.svelte

Projekt bearbeiten

Route: /admin/[slug]/edit

Edit Project

Title
<Placeholder> Task Manager App

Description
A full-stack task management application with real-time collaboration features.

Image
No file chosen

Current Image:
A 10x10 grid of colored squares with letters A-J and numbers 1-10.

GitHub Link
https://github.com/albertgstoahl/task-manager

Live Demo Link
https://task-manager-demo.com

Technologies
Name: Node.js, Express, MongoDB, React +

Tags
Add a tag: Full Stack, Productivity +

Save Changes

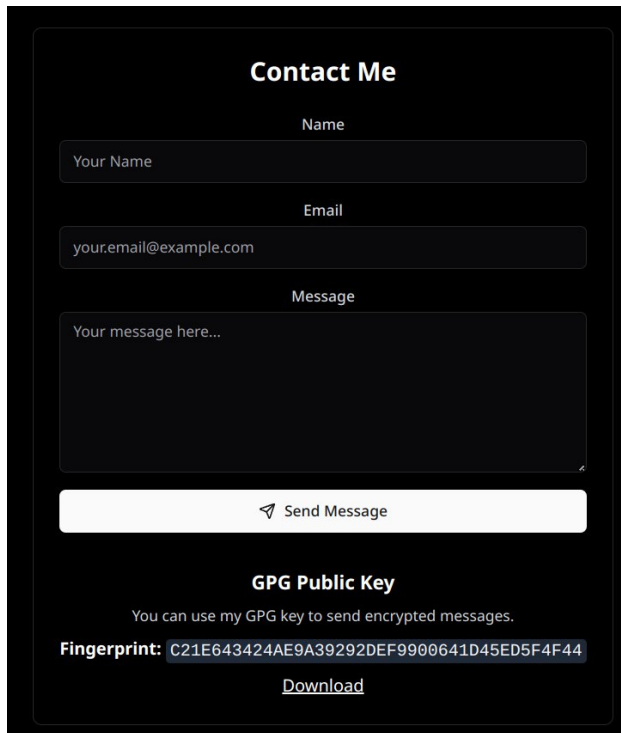
Die Edit-Seite ermöglicht Administratoren nach der Authentifizierung, ein bestehendes Projekt zu bearbeiten. Alle relevanten Projektdaten werden in einem Formular vorab ausgefüllt und können angepasst werden. Nach dem Speichern werden die Änderungen an den Server übermittelt (Endpunkt /admin/[slug]/edit?/save).

Dateien:

- routes/admin/[slug]/edit/+page.svelte
- routes/admin/[slug]/edit/+page.server.js
- components/ProjectForm.svelte

Kontakt-Seite

Route: /contact



The screenshot shows a dark-themed web form titled "Contact Me". It contains three input fields: "Name" with placeholder text "Your Name", "Email" with placeholder text "your.email@example.com", and "Message" with placeholder text "Your message here...". Below these fields is a white button with a paper plane icon and the text "Send Message". At the bottom, there is a section titled "GPG Public Key" with the text "You can use my GPG key to send encrypted messages." and a "Fingerprint: C21E643424AE9A39292DEF9900641D45ED5F4F44". A "Download" link is provided below the fingerprint.

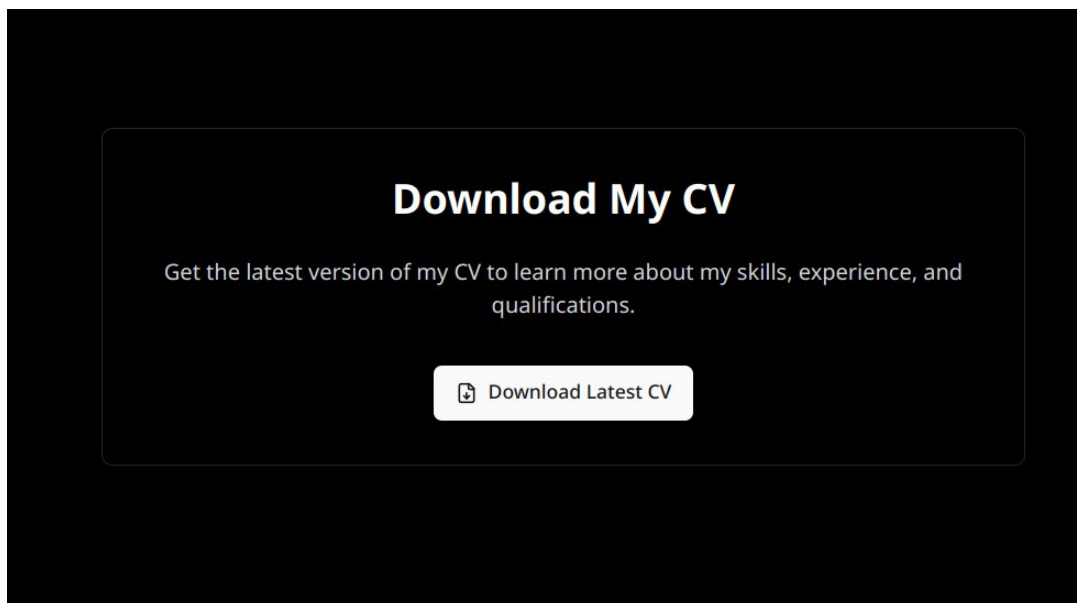
Auf der Seite /contact können Besucher eine Nachricht über ein Formular versenden. Die Eingabe von Name, E-Mail und Nachricht ist erforderlich. Nach dem Absenden wird dem Benutzer eine Bestätigung angezeigt, dass die Nachricht erfolgreich übermittelt wurde. Das Formular verwendet Netlify für die Verarbeitung der Formulardaten.

Dateien:

- routes/contact/+page.svelte

CV-Seite

Route: /cv



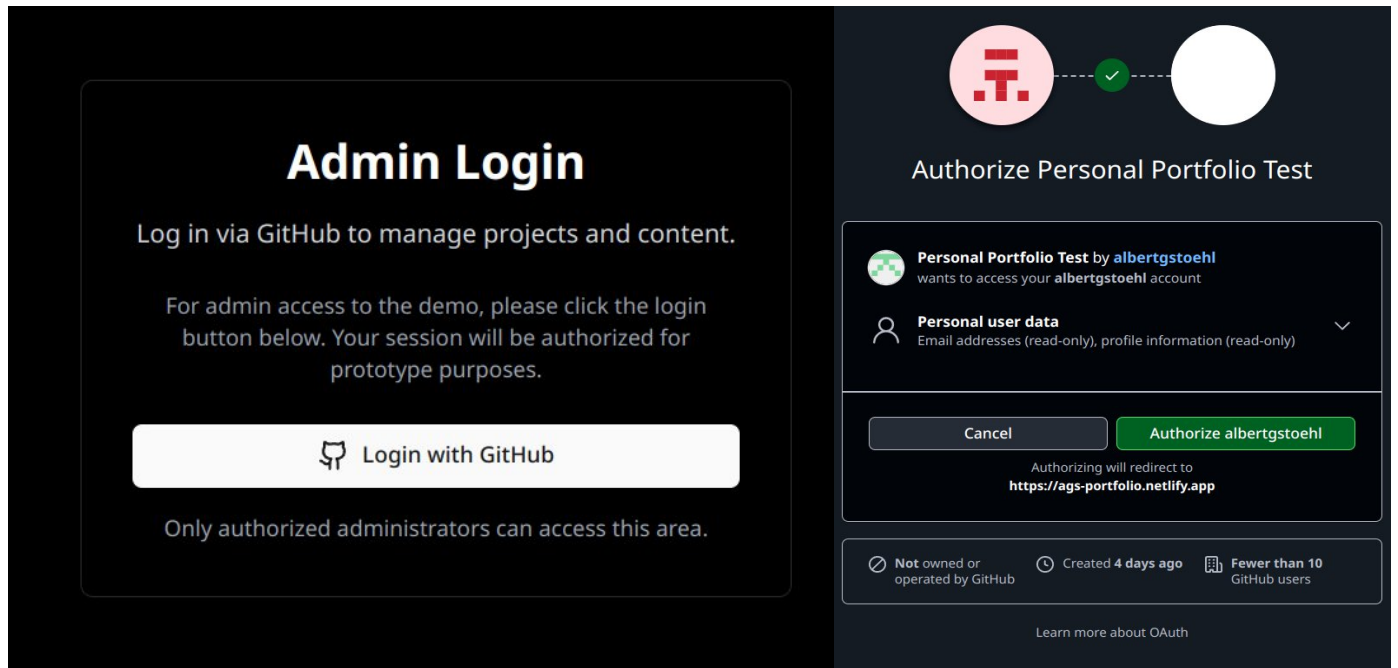
Auf der Seite /cv kann der neueste aus Datenschutzgründen zensierte Lebenslauf als PDF aus einem separaten Repository heruntergeladen werden. Dies wurde über einen serverseitigen Proxy-Endpoint realisiert, der den Link zum aktuellen PDF bereitstellt.

Datei(en):

- routes/cv/+page.svelte
- routes/cv/+page.server.js
- routes/api/download-cv/+server.js

Login-Seite mit GitHub-OAuth

Route: /signin



Auf der Seite /signin können sich Benutzer via GitHub-OAuth authentifizieren. Für Demo-Zwecke erlaubt die Seite momentan jede Anmeldung. Dies wurde durch die Nutzung der GitHub-OAuth-API und serverseitiger Authentifizierungslogik umgesetzt. Hierfür wurde [auth.js](#) verwendet. Auf der /signout Seite, wird der Benutzer ausgeloggt.

Datei(en):

- routes/signin/+page.server.js
- routes/signin/+page.svelte
- routes/signout/+page.server.js
- routes/+layout.server.js
- src/auth.js

Erweiterungen

Laptop mit dynamischen Icon-Meshes auf der Intro-Seite

Auf der Intro-Seite wird ein interaktiver Laptop als 3D-Objekt dargestellt, dessen Bildschirm dynamisch die Icons der in den Projekten verwendeten Technologien anzeigt. Diese Icons werden aus der Projektdatenbank geladen. Durch einen Klick auf ein Icon wird der Benutzer direkt zur gefilterten Projektübersicht weitergeleitet, die nur Projekte mit der ausgewählten Technologie anzeigt. Dies wurde mit einer Svelte Library für [three.js](#) mit dem Namen [threlte](#) umgesetzt

Datei(en):

- lib/SceneOne.svelte
- components/Laptop.svelte

Projektverwaltung:

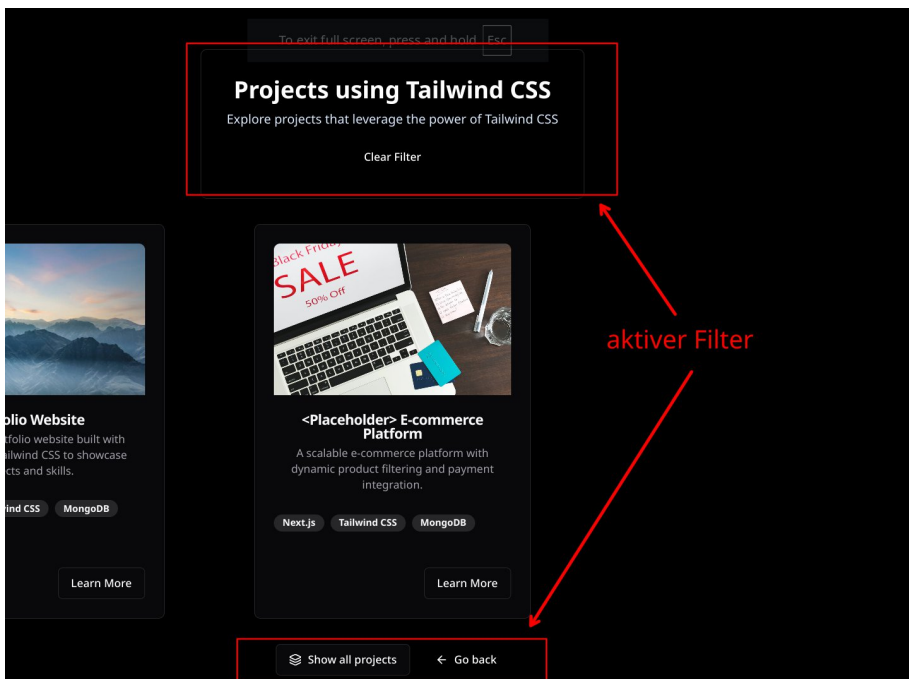
Auf der Seite /projects können neue Projekte hinzugefügt (entspricht der Grundanforderung), bestehende Projekte bearbeitet oder gelöscht werden (Erweiterung). Diese Funktionen stehen nur authentifizierten Admins zur Verfügung. Dies wurde mit Formularen, einer MongoDB-Integration und serverseitigen Actions realisiert. Die zugehörigen Endpoints für die Actions sind:

- /admin/create?/add (Hinzufügen eines neuen Projekts)
- /admin/[slug]/edit?/save (Speichern der Änderungen an einem Projekt)
- /admin/[slug]?/delete (Löschen eines Projekts)

Datei(en):

- routes/admin/[slug]/edit/+page.svelte
- routes/admin/[slug]/edit/+page.server.js
- routes/admin/create/+page.svelte
- routes/admin/create/+page.server.js
- routes/admin/[slug]/+page.server.js

Filter

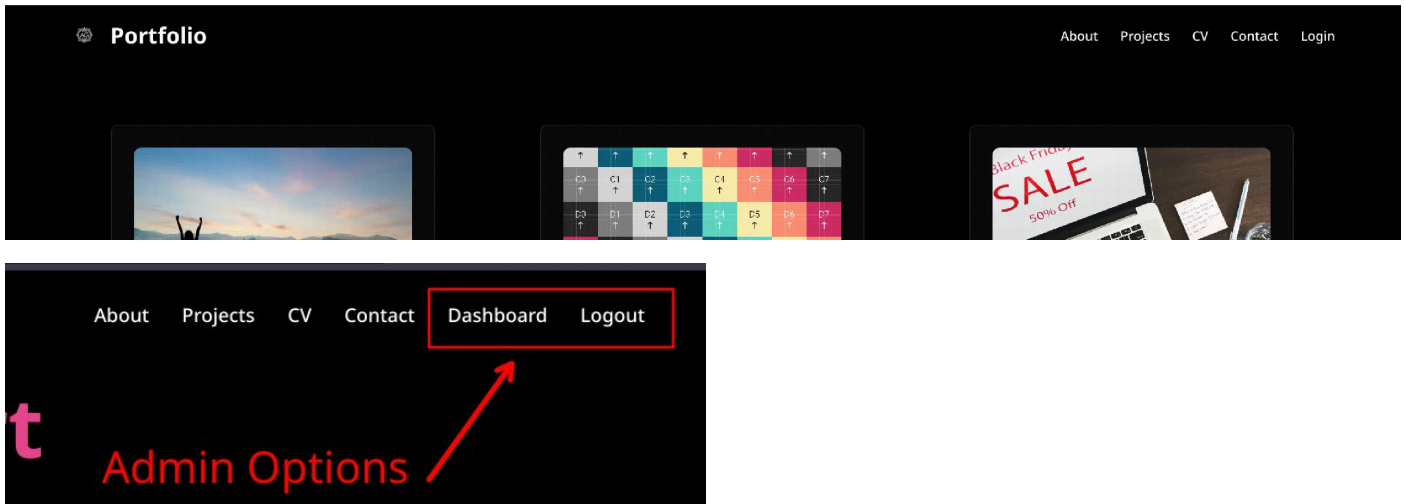


Die Filterfunktion ermöglicht es, Projekte nach Technologien zu filtern. Technologien werden als klickbare Badges dargestellt, die per URL-Parameter (?technology=<name>) die Projektübersicht dynamisch anpassen.

Datei(en):

- routes/projects/+page.svelte
- routes/projects/+page.server.js

Navbar mit Scroll-Interaktion und dynamischer Anpassung

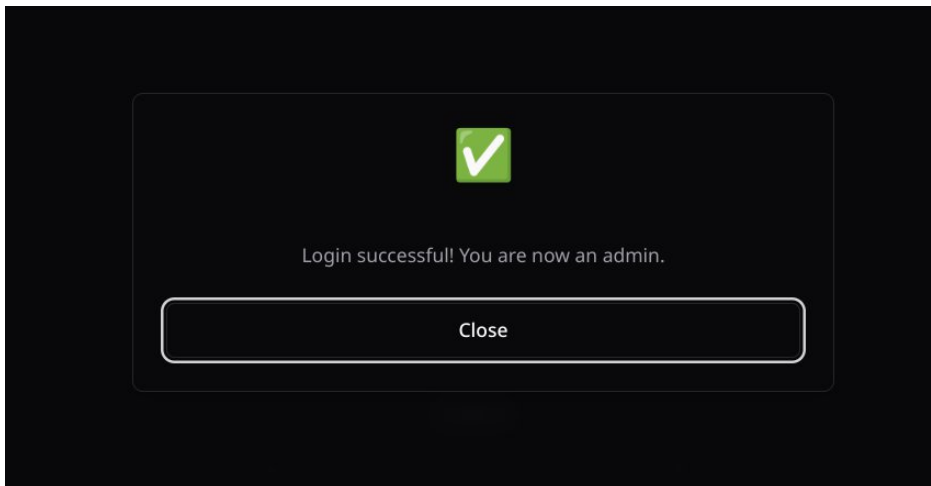


Die Navbar verschwindet beim Scrollen nach unten und erscheint wieder oben. Zudem passt sich die Navbar dynamisch an, wenn ein Admin eingeloggt ist, indem zusätzliche Links und eine Logout-Option angezeigt werden.

Datei(en):

- lib/components/Navbar.svelte
- lib/menuStore.js
- routes/+layout.svelte

Lokale Popups mit reaktiven Stores



In der Anwendung wurden lokale Popups implementiert, die mithilfe eines reaktiven Stores gesteuert werden. Diese Popups können dynamisch Nachrichten anzeigen (z. B. Erfolg, Fehler, Informationen) und schliessen sich nach einer vordefinierten Zeit automatisch. Sie werden verwendet, um dem Benutzer Feedback zu Aktionen zu geben, z. B. beim Erstellen, Bearbeiten von Projekten und beim Login.

Datei(en):

- stores/modalStore.js
- components/GlobalModal.svelte

Login mit GitHub-OAuth

Die Login-Seite ermöglicht die Authentifizierung von Benutzern über GitHub-OAuth. Für Demo-Zwecke erlaubt die Seite momentan jede Anmeldung. Nach erfolgreicher Anmeldung wird ein Admin-Bereich freigeschaltet, in dem Projekte bearbeitet, erstellt und gelöscht werden können. Eine Logout-Seite sorgt für die Abmeldung des Benutzers.

Datei(en):

- routes/signin/+page.server.js
- routes/signin/+page.svelte
- routes/signout/+page.server.js
- routes/+layout.server.js
- src/auth.js

Kontaktformular

Das Kontaktformular ermöglicht es Besuchern, Nachrichten direkt über die Webseite zu senden. Nach dem Absenden wird eine Erfolgsmeldung angezeigt, und die Formulardaten werden mithilfe von [Netlify](#) verarbeitet. Diese Funktion erweitert die Benutzerfreundlichkeit und bietet eine direkte Kommunikationsmöglichkeit.

Datei(en):

- routes/contact/+page.svelte

CV-Seite

Auf der CV-Seite kann der neueste Lebenslauf des Entwicklers als PDF heruntergeladen werden. Die Datei wird aus einem separaten Repository bezogen, das den Lebenslauf automatisch in LaTeX generiert und regelmässig aktualisiert. Ein serverseitiger Proxy-Endpunkt sorgt für die sichere Bereitstellung der Datei.

Datei(en):

- routes/cv/+page.svelte
- routes/cv/+page.server.js
- routes/api/download-cv/+server.js