Prototyping Projektdokumentation

**Name**: Delia Troncato

**E-Mail:** troncdel@students.zhaw.ch

**URL der deployten Anwendung:** [myeventplace2.netlify.app](https://myeventplace2.netlify.app/)

# Einleitung

"My Event Place" wurde mit dem Ziel entwickelt, eine zentrale und benutzerfreundliche Plattform zu schaffen, auf der Nutzer:innen eine Vielzahl unterschiedlicher Veranstaltungen entdecken, erleben und verwalten können. Egal ob Musikfestivals, Open-Air-Konzerte, kulinarische Erlebnisse wie Wine Tastings oder kulturelle Events – die Webapplikation bietet eine übersichtliche Darstellung kommender Veranstaltungen.

Im Zentrum der Anwendung stehen drei Hauptfunktionen:

1. Veranstaltungen entdecken und anmelden:

Nutzer:innen erhalten einen direkten Überblick über bevorstehende Events. Zu jedem Event werden alle relevanten Informationen wie Titel, Beschreibung, Datum, Uhrzeit, Ort und Dauer bereitgestellt. Mit nur wenigen Klicks ist eine verbindliche Anmeldung über ein eingebundenes Formular möglich.

1. Lieblings-Venues verwalten:

Neben der Event-Registrierung bietet die Plattform eine Funktion zum Speichern persönlicher Lieblingslocations. Nutzer:innen können neue Veranstaltungsorte hinzufügen oder bestehende Favoriten jederzeit wieder entfernen. Dies erleichtert die spätere Suche nach Events an vertrauten Orten und ermöglicht eine individuell angepasste Veranstaltungsübersicht.

1. Direkter Kontakt zu Veranstaltern:

Eine zusätzliche Detailseite für jeden Veranstalter erlaubt es, mehr über die Organisatoren zu erfahren. Ausserdem steht ein Kontaktformular zur Verfügung, mit dem Nutzer:innen direkt Nachrichten an die Veranstalter senden können.

# Datenmodell

*Ein Bild, das Diagramm, Reihe, Text enthält.

KI-generierte Inhalte können fehlerhaft sein.*

# Beschreibung der Anwendung

## Startseite

Route: /

Ein Bild, das Text, Elektronik, Computer, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Beim Aufrufen der Webanwendung erscheint als Erstes diese Seite. Diese Seite soll die Startseite darstellen und bietet die Möglichkeit, sich einen direkten Überblick über die Funktionalitäten der Webanwendung zu schaffen.

Zu Beginn sind drei Bilder mit drei weiterführenden Buttons ersichtlich:

* Der Button "Zu den Veranstaltungen" leitet auf die Seite /veranstaltungen weiter
* Der Button "Zu meinen Lieblings-Venues" leitet auf die Seite /lieblingsvenues weiter
* Der Button "Zu den Veranstaltern" leitet auf die Seite /veranstalter weiter

Weiter unten findet man den Terminkalender. Für die Darstellung wird die Komponente Terminkalender.svelte verwendet. Der Terminkalender zeigt alle anstehenden Events auf, die chronologisch sortiert sind.

Zusätzlich befindet sich in der Kopfzeile das Logo. Rechts in der Kopfzeile wird die Navigationsleiste dargestellt, die es ermöglicht, sich in die verschiedenen Seiten zu begeben. Die Kopfzeile ist auf allen Seiten der Webanwendung sichtbar und wird durch die Datei +layout.svelte eingebunden.

Dateien:

* components/Terminkalender.svelte
* lib/db.js
* routes/+layout.svelte
* routes/+page.server.js
* routes/+page.svelte

## Veranstaltungen

Route: /veranstaltungen

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite wird eine Übersicht von allen anstehenden Events abgebildet. Bevor alle Events aufgelistet werden, hat man zusätzlich die Möglichkeit, verschiedene Filter zu setzen. Über die Suchleiste kann man auch gezielt nach Veranstaltungen suchen. Im unteren Bereich der Seite sind alle Veranstaltungen mit deren Details wie Datum, Uhrzeit, Ort und Dauer aufgelistet. Diese Daten werden direkt aus der MongoDB Collection "events" geladen. Jeder Event hat einen dazugehörigen "Details & Anmeldung"-Button, mit dem man auf der Detailseite der jeweiligen Veranstaltung landet.

Dateien:

* lib/db.js
* routes/veranstaltungen/+page.server.js
* routes/veranstaltungen/+page.svelte

## Veranstaltungen-Details

Route: /veranstaltungen/[id]

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf der Detailseite der jeweiligen Veranstaltung sind im oberen Teil wieder die Informationen zur Veranstaltung ersichtlich, mit dem Unterschied, dass auf der Detailseite auch die Beschreibung der Veranstaltung ersichtlich ist. Im unteren Teil wird direkt das Anmeldeformular angezeigt, welches mit der Komponente RegistrationsFormular.svelte eingebunden wurde. Das Anmeldeformular besteht aus vier Eingabezeilen und einer Checkbox. Diejenigen Elemente, die mit einem Stern markiert sind, sind auch Pflichtfelder. Nach dem Absenden des Formulars werden die Daten serverseitig verarbeitet und in der MongoDB Collection "registrations" gespeichert. Bei erfolgreicher Anmeldung wird eine entsprechende Erfolgsmeldung.

Dateien:

* lib/db.js
* components/RegistrationsFormular.svelte
* routes/veranstaltungen/[veranstaltungen\_id]/+page.server.js
* routes/veranstaltungen/[veranstaltungen\_id]/+page.svelte

## Meine Lieblings-Venues

Route: /lieblingsvenues

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite sieht man alle favorisierten Veranstaltungsorte, mit dem Ziel, individuelle Präferenzen zu verwalten, um in Zukunft gezieltere Veranstaltungsangebote zu erhalten. Im oberen Teil hat man die Möglichkeit, mittels Eingabe des Namens und der Adresse weitere Veranstaltungsorte in der Liste zu speichern. Mit dem Button "Hinzufügen" wird die Eingabe automatisch zuunterst zur Liste hinzugefügt und in der MongoDB Collection "venues" erfasst. Jeder Venue-Eintrag enthält einen "Entfernen"-Button, mit dem die Nutzer:innen den Ort wieder aus seiner Liste löschen können. Mit dem Entfernen des Ortes wird der Eintrag aus der MongoDB Collections auch wieder gelöscht.

Dateien:

* lib/db.js
* routes/lieblingsvenues/+page.server.js
* routes/lieblingsvenues/+page.svelte

## Veranstalter

Route:/veranstalter

Ein Bild, das Menschliches Gesicht, Text, Person, Frau enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite werden alle Veranstalter in einer übersichtlichen Kartenansicht dargestellt. Die Daten werden direkt aus der MongoDB Collection "organizers" geladen. Jeder Veranstalter enthält ein Portraitbild, den vollständigen Namen, eine Kurzbeschreibung mit Rolle, Ort und Erfahrung und die dazugehörige E-Mail-Adresse. Unter jedem Eintrag befindet sich der Button "Details anzeigen", der zu einer separaten Detailseite des jeweiligen Veranstalters führt. Zusätzlich befindet sich im oberen Bereich der Seite ein Ortsfilter. Hier können Nutzer gezielt Veranstalter aus einer bestimmten Stadt auswählen und so die Liste filtern. Direkt darunter unterstützt eine Suchleiste die gezielte Suche nach Veranstaltern anhand ihres Namens, des Orts oder Begriffen aus der Kurzbeschreibung.

Dateien:

* lib/db.js
* routes/veranstalter/+page.server.js
* routes/veranstalter/+page.svelte

## Veranstalter-Detailseite

Route:/veranstalter/[id]

Ein Bild, das Text, Menschliches Gesicht, Screenshot, Lächeln enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite werden die Informationen zu einem einzelnen Veranstalter übersichtlich dargestellt. Im oberen Bereich befinden sich das Portrait, der vollständige Name, die Biografie sowie die E-Mail-Adresse der jeweiligen Person. Die E-Mail-Adresse ist als mailto: -Link eingebunden. Ein Klick auf die E-Mail-Adresse mit gedrückter Ctrl-Taste öffnet automatisch das E-Mail-Programm mit einer neuen Nachricht an den Veranstalter. Im unteren Seitenbereich ist ein Kontaktformular eingebunden. Dieses wurde über die wiederverwendbare Komponente KontaktFormular.svelte integriert. Nutzerinnen können über dieses Formular eine persönliche Nachricht an den Veranstalter senden. Die übermittelten Daten werden dabei serverseitig verarbeitet und automatisch in der MongoDB-Collection "messages" gespeichert. Sobald die Daten über den Button "Nachricht senden" abgesendet werden, erscheint eine Erfolgsmeldung.

Dateien:

* lib/db.js
* components/ KontaktFormular.svelte
* routes/veranstalter/[veranstalter\_id]/+page.server.js
* routes/veranstalter/[veranstalter\_id]/+page.svelte

# Erweiterungen

## Startseite

Auf der Startseite ist eine Übersicht über alle Funktionalitäten der Webanwendung ersichtlich und mittels Button zu den entsprechenden Seiten verlinkt:

* Verbindung zu /veranstaltungen
* Verbindung zu /lieblingsvenues
* Verbindung zu /veranstalter

Dateien:

* routes/+page.server.js
* routes/+page.svelte

## Veranstaltungen-Filter

Auf der Seite /veranstaltungen wurde zur besseren Übersicht und Benutzerfreundlichkeit eine Filterfunktion nach Veranstaltungsort implementiert. Oberhalb der Eventliste befindet sich eine Leiste mit Buttons, die die verfügbaren Veranstaltungsorte als visuelle Filter anzeigen. Mittels Klick auf den Button wird die Liste der Veranstaltungen clientseitig gefiltert und zeigt nur die Veranstaltungen an, die am entsprechenden Ort stattfinden. Mit dem Button "Alle" lässt sich der Filter wieder zurücksetzen, sodass alle Veranstaltungen gezeigt werden. Die Filterfunktion wurde direkt im +page.svelte implementiert.

Datei:

* routes/veranstaltungen/+page.svelte

## Veranstaltungen-Suche

Ergänzend zur Filterfunktion wurde auf der Seite /veranstaltungen auch ein Suchfeld integriert, welches eine freie Texteingabe ermöglicht. So können Nutzer:innen gezielt nach den bestimmten Begriffen wie Titel der Veranstaltung, Ort und Datum suchen. Die Suche wird clientseitig durchgeführt und funktioniert in Echtzeit. Das bedeutet, dass schon bei der Eingabe des Begriffs die Ergebnisliste dynamisch aktualisiert wird.

Datei:

* routes/veranstaltungen/+page.svelte

## Veranstalter-Filter

Oberhalb der Ergebnisliste der Seite /veranstalter befindet sich eine Auswahl dynamisch generierter Ort-Buttons. Diese werden aus den Biografie-Texten der Veranstalter automatisch extrahiert. Mit einem Klick auf einen dieser Buttons werden nur diejenigen Veranstalter angezeigt, deren Biografie den jeweiligen Ort enthält. Die aktive Auswahl ist optisch hervorgehoben, sodass man schnell erkennen kann, welcher Filter aktiv ist. Mit dem Button "Alle" lässt sich der Filter wieder zurücksetzen, sodass alle Veranstalter wieder angezeigt werden. Die Filterfunktion wurde auch hier direkt im +page.svelte implementiert.

Datei:

* routes/veranstalter/+page.svelte

## Veranstalter-Suche

Ergänzend zum Ortsfilter erlaubt ein eingebundenes Suchfeld auf der Seite /veranstalter die freie Texteingabe. Die Suche erfolgt nach dem Namen, Ort oder Biografie. Die Liste aktualisiert sich automatisch während der Eingabe des Suchbegriffs, da die Suche in Echtzeit durchgeführt wird.

Datei:

* routes/veranstalter/page.svelte

## Zusätzliche Detailpage – /veranstalter/[id]

Zusätzlich wurde im Rahmen einer funktionalen Erweiterung eine weitere Detailseite erstellt, welche die Veranstalter im Detail abbildet. Auf dieser Seite befindet sich wie bereits im Kapitel 3.6 beschrieben alle Informationen zum Veranstalter wie Name, Biografie, Bild und E-Mail-Adresse. Die weiteren zusätzlichen Funktionalitäten dieser Seite werden in den nachfolgenden Kapitel beschrieben.

Datei:

* routes/veranstalter/[veranstalter\_id]/+page.svelte

## Mailto: Verlinkung (Zusätzliche Detailpage – /veranstalter/[id])

Auf der Detailseite eines Veranstalters wird die E-Mail-Adresse nicht nur als Text angezeigt, sondern als interaktiver Link umgesetzt. Diese Verlinkung wurde mithilfe des HTML-Schemas mailto: realisiert. Dadurch wird beim Klicken mit der Ctrl-Taste auf die E-Mail-Adresse automatisch das lokal installierte E-Mail-Programm des Nutzers geöffnet und eine neue Nachricht an den jeweiligen Veranstalter vorbereitet.

Datei:

* routes/veranstalter/[veranstalter\_id]/+page.svelte

## Veranstalter Kontaktformular (Zusätzliche Detailpage – /veranstalter/[id])

Zusätzlich wurde auf der Detailseite eines Veranstalters ein Formular eingebunden, das den Nutzenden ermöglicht, dem Veranstalter direkt über ein Formular eine Nachricht zu senden. Die übermittelten Daten werden anschliessend automatisch in der Collection "messages" gespeichert. Das Kontaktformular wurde über den wiederverwendbaren Komponenten KontaktFormular.svelte in die +page.svelte eingebunden.

Dateien:

* components/ KontaktFormular.svelte
* routes/veranstalter/[veranstalter\_id]/+page.svelte