

Prototyping Projektdokumentation

Name: Robin Hinder

E-Mail: hindoero1@students.zhaw.ch

URL der deployten Anwendung: [https://myevents1.netlify.app/]

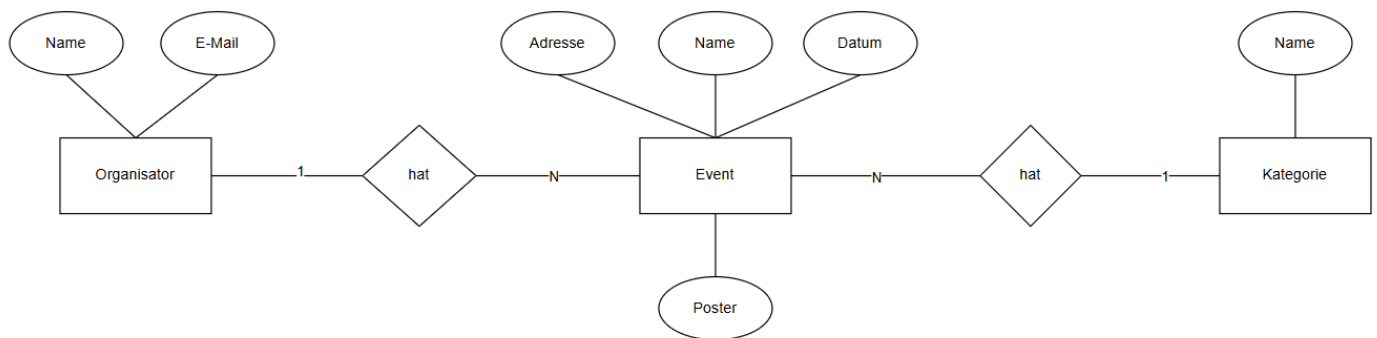
Einleitung

MyEvents ist eine Event-Plattform, die es Nutzern ermöglicht, Veranstaltungen einfach zu entdecken und zu verwalten. Die Grundidee der App ist es, sowohl Veranstalter als auch Teilnehmer miteinander zu vernetzen und den Prozess der Eventorganisation zu erleichtern.

Die Hauptfunktionen der App umfassen die Anzeige von Veranstaltungen, das Erstellen von Events, das Verwalten von Events auf der Eventslist und der individuellen Suche von Events über Organisatoren und Kategorien. Ausserdem können Nutzer auf den Detailpages weitere Informationen wie den Veranstaltungsort, die Organisatoren, die Kategorie und das Datum einsehen.

Das Ziel der App ist es, eine einfache und benutzerfreundliche Lösung für die Suche nach passenden Events zu bieten. Gleichzeitig soll sie Organisatoren helfen, ihre Events effektiv zu präsentieren.

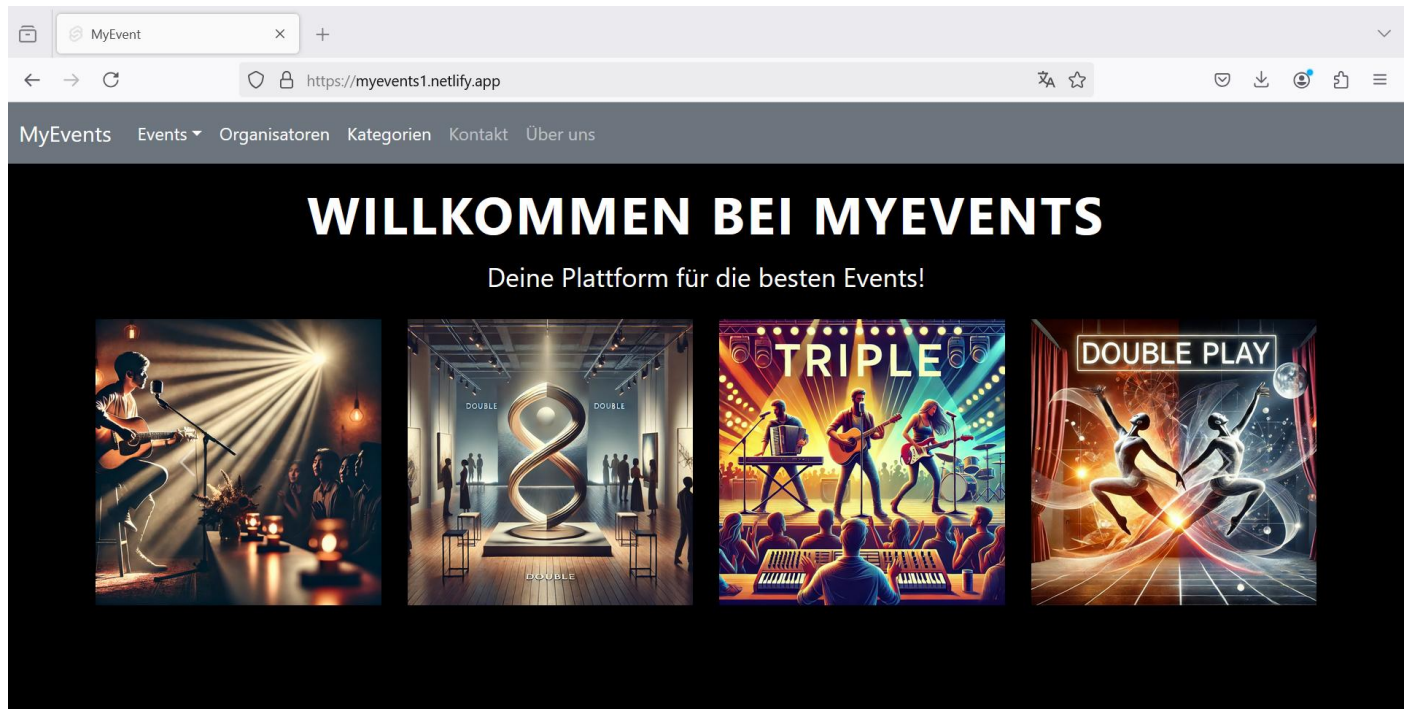
Datenmodell



Beschreibung der Anwendung

Hauptseite MyEvents

Route: /



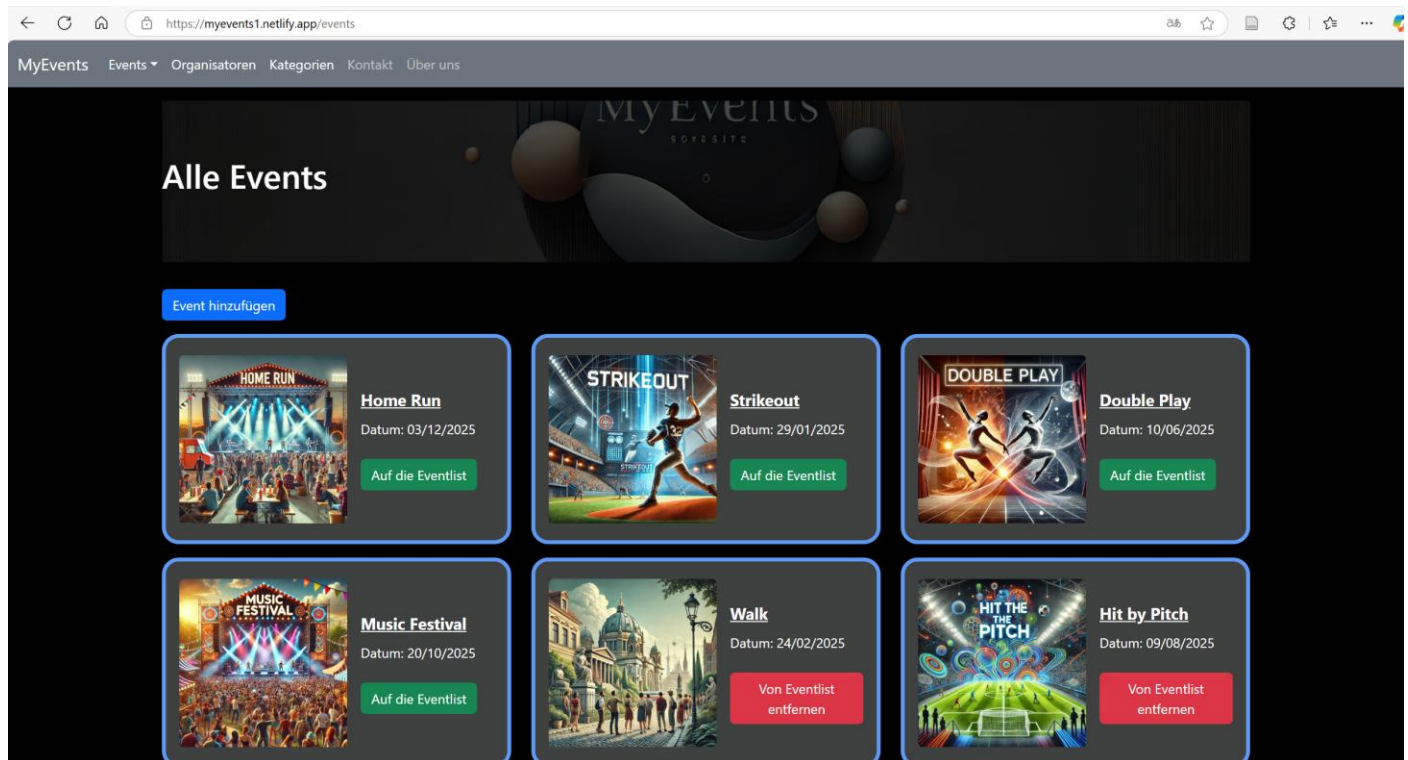
Auf dieser Startseite wird der Nutzer von MyEvents begrüßt. Sie dient als zentraler Einstiegspunkt zur Plattform und bietet eine Übersicht über den Zweck der Anwendung. Neben dem Titel und den Bildern, welche mit der Carousel-Funktion angezeigt werden, können Nutzer die Navigationsleiste nutzen, um zu den verschiedenen Bereichen der Plattform zu gelangen. Durch einen Klick auf das Dropdown-Menü Events kann man zwischen verschiedenen Ansichten wählen. Ein Klick auf Alle Events führt zur Seite /events, auf der alle verfügbaren Events angezeigt werden. Wählt man Eventlist, gelangt man zur Seite events/eventlist, auf der ausschliesslich Events angezeigt werden, die zur persönlichen Eventlist hinzugefügt wurden. Darüber hinaus gibt es die Möglichkeit, Events über die jeweiligen Organisatoren auf der Seite /organisatoren oder über die Kategorien auf der Seite /kategorien anzuschauen.

Dateien:

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

Eventübersicht

Route: /events



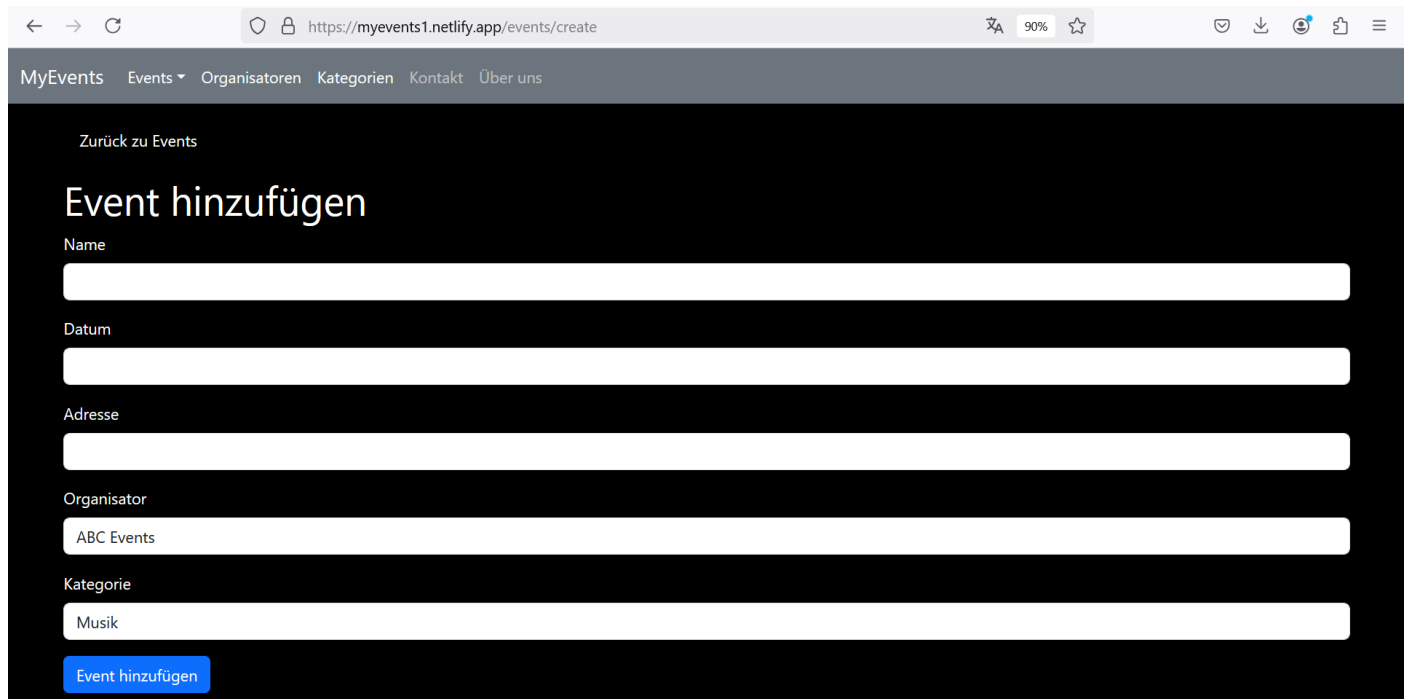
Auf dieser Seite werden alle verfügbaren Events durch die `getEvents()` Funktion dargestellt. Jedes Event wird mithilfe der Komponente `EventCard.svelte` angezeigt, welche die wichtigen Informationen wie den Namen, das Datum und ein Bild des Events enthält. Nutzer können Events durch einen Klick auf den Button „Auf die Eventlist“ zu ihrer Eventliste hinzufügen. Ausserdem ist es möglich, durch einen Klick auf den Namen eines Events zu `/events/[event_id]` zu gelangen und weitere Details aufzurufen. Die Seite dient als zentraler Ort zur Verwaltung und Entdeckung von Veranstaltungen. Über den Button „Event hinzufügen“ gelangt man zu `/events/create` und kann einen neuen Event hinzufügen.

Dateien:

- `lib/components/EventCard.svelte`
- `routes/events/+page.svelte`
- `routes/events/+page.server.js`
- `routes/styles.css`
- `lib/db.js`

Event hinzufügen

Route: /events/create



The screenshot shows a web browser window with the URL `https://myevents1.netlify.app/events/create`. The browser's address bar and navigation icons are visible at the top. Below the browser window, a dark-themed navigation bar contains the text 'MyEvents' followed by a dropdown menu 'Events' and links for 'Organisatoren', 'Kategorien', 'Kontakt', and 'Über uns'. The main content area has a dark background. At the top left of this area is a link 'Zurück zu Events'. The title 'Event hinzufügen' is centered. Below the title are five form fields, each with a label to its left: 'Name', 'Datum', 'Adresse', 'Organisator', and 'Kategorie'. The 'Organisator' field contains the text 'ABC Events' and the 'Kategorie' field contains 'Musik'. At the bottom left of the form is a blue button with the text 'Event hinzufügen'.

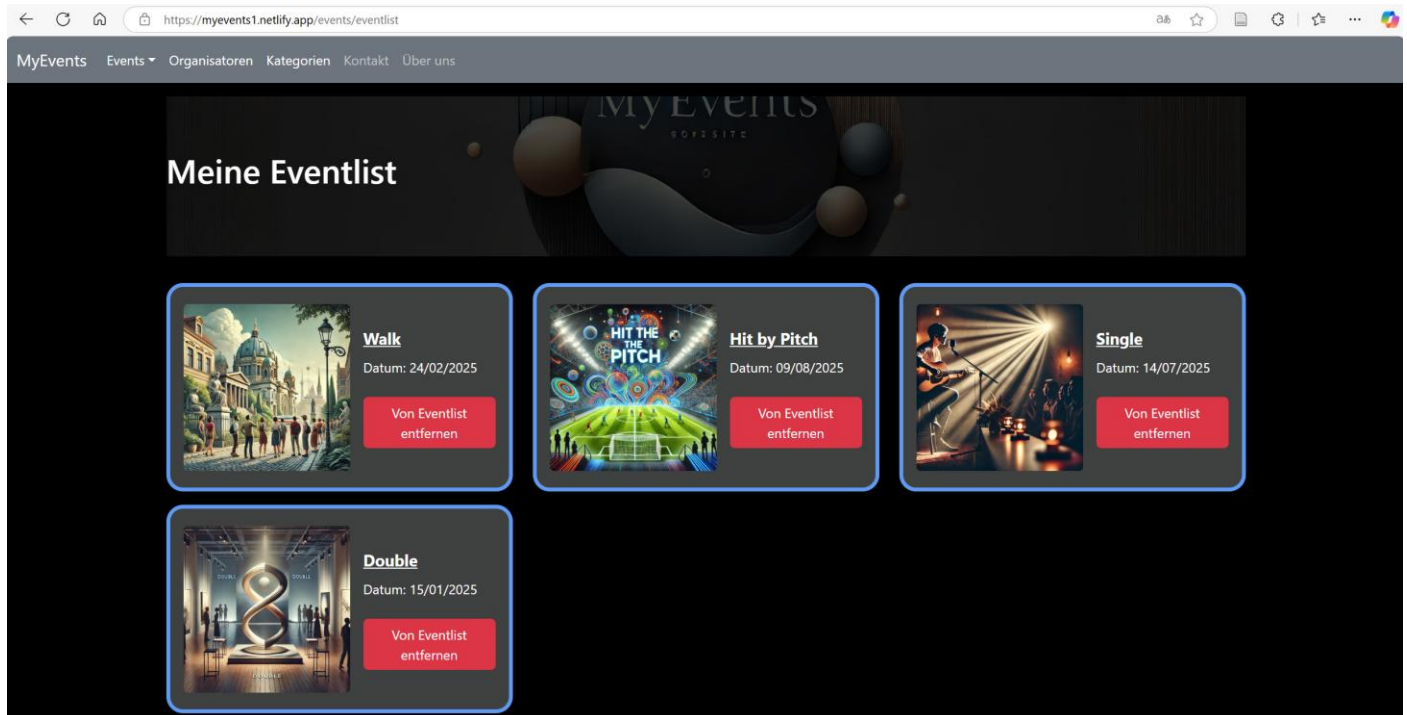
Auf dieser Seite können Nutzer neue Events erstellen. Informationen wie den Namen des Events, das Datum und die Adresse müssen eingegeben werden. Organisator und Kategorie werden durch die `load()` Funktion in Dropdown-Menüs angezeigt (Siehe Erweiterungen). Über den Button „Event hinzufügen“ und die Funktion `createEvent()` wird der Event in der Datenbank gespeichert und in der Eventübersicht angezeigt. Im Anschluss wird erscheint unterhalb des Buttons als Bestätigung, die Meldung «Event wurde erfolgreich hinzugefügt». Ein „Zurück zu Events“-Link führt die Nutzer zurück zur Eventübersicht, falls sie das Formular verlassen möchten.

Dateien:

- `routes/events/create/+page.svelte`
- `routes/events/create/+page.server.js`
- `lib/db.js`

Eventlistübersicht

Route: events/eventlist



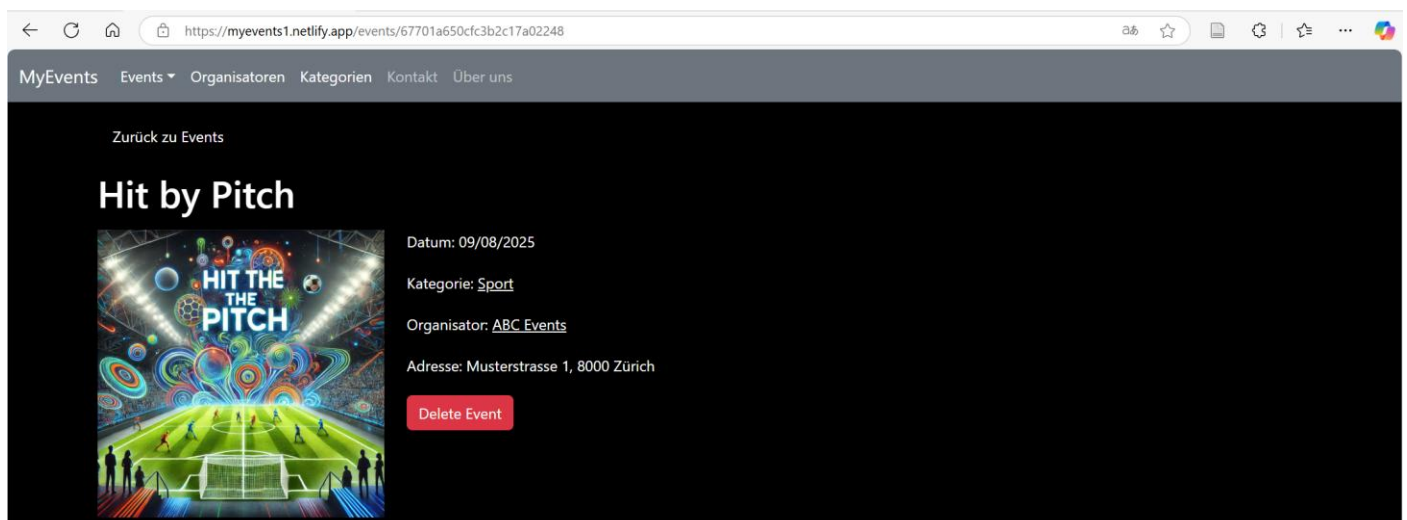
Auf dieser Seite wird die persönliche Eventliste angezeigt. Durch die Funktion `getEvents()` in `db.js` werden die Events bereitgestellt. Dabei wird `events.filter(event => event.eventlist)` genutzt, um nur die relevanten Events anzuzeigen. Anschliessend werden alle Events aufgeführt, die der Nutzer zuvor zu seiner Liste hinzugefügt hat. Jedes Event wird mithilfe der Komponente `EventCard.svelte` angezeigt. Über den Button „Von Eventlist entfernen“ kann ein Event wieder aus der Liste entfernt werden. Durch einen Klick auf den Namen des Events gelangt man zu `/events/[event_id]` und der Nutzer kann sich weitere Details zum Event anzeigen lassen.

Dateien:

- `lib/components/EventCard.svelte`
- `routes/events/eventlist/+page.svelte`
- `routes/events/eventlist/+page.server.js`
- `routes/styles.css`
- `lib/db.js`

Event Detailseite

Route: `/events/[event_id]`



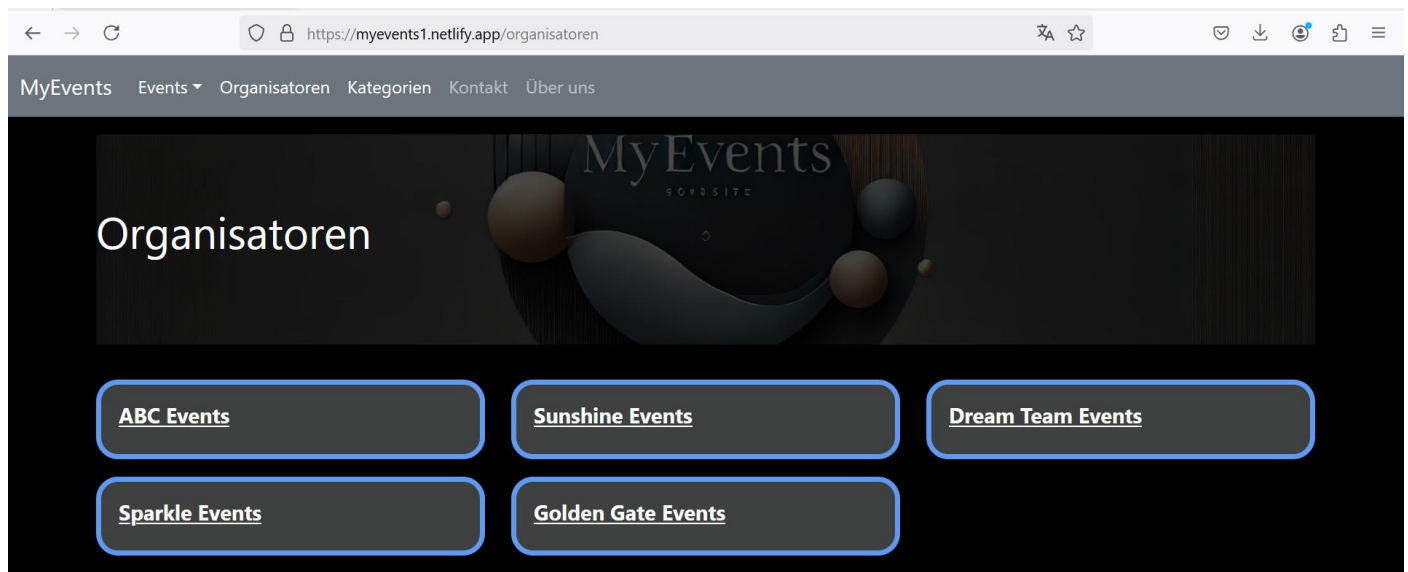
Auf der Detailseite eines Events werden alle relevanten Informationen zu einer Veranstaltung durch die Funktion `getEvent(id)` dargestellt. Neben dem Event-Namen und Datum werden auch die Kategorie, der Organisator sowie die Adresse angezeigt. Der Nutzer hat die Möglichkeit, auf die Kategorie oder den Organisator zu klicken, um zur Seite `/kategorien/[kategorie_id]` oder `/organisatoren/[organisatoren_id]` zu gelangen (Siehe Erweiterungen). Über den Button „Delete Event“ kann das Event dauerhaft gelöscht werden. Ein „Zurück zu Events“-Link ermöglicht es, zur vorherigen Eventübersicht zurückzukehren.

Dateien:

- `routes/events/[event_id]/+page.svelte`
- `routes/events/[event_id]/+page.server.js`
- `lib/db.js`

Organisatorenübersicht

Route: `/organisatoren`



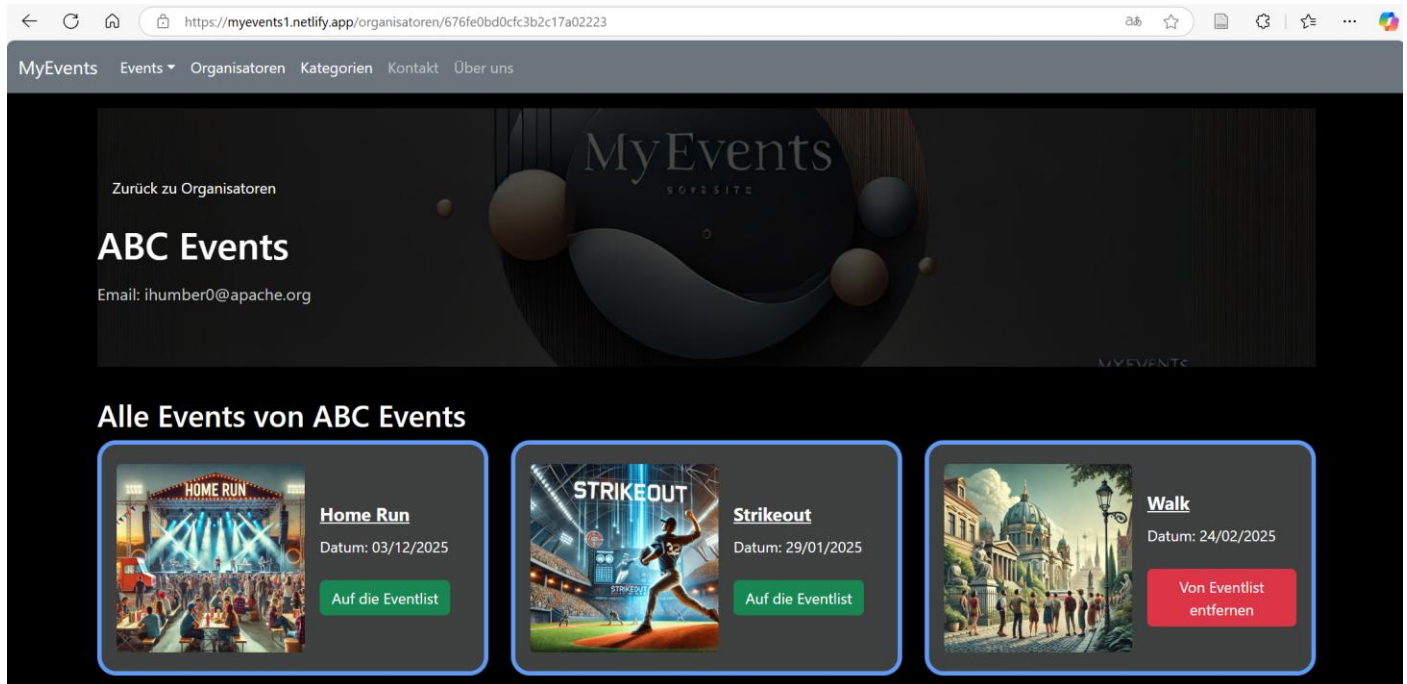
Auf dieser Seite werden alle Organisatoren durch die Funktion `getOrganisatoren()` angezeigt. Jeder Organisator wird mit der Komponente `OrganisatorCard.svelte` dargestellt. Durch einen Klick auf einen Organisator gelangt der Nutzer zu `/organisatoren/[organisatoren_id]`, wo er sich Infos und eine Übersicht von allen Events, die von diesem Organisator veranstaltet werden, ansehen kann. Die Seite bietet eine einfache Möglichkeit, Veranstaltungen nach Organisator zu durchsuchen und gezielt Informationen zu einem bestimmten Veranstalter zu erhalten.

Dateien:

- `lib/components/OrganisatorCard.svelte`
- `routes/organisatoren/+page.svelte`
- `routes/organisatoren/+page.server.js`
- `routes/styles.css`
- `lib/db.js`

Organisator Detailseite

Route: /organisatoren/[organisatoren_id]



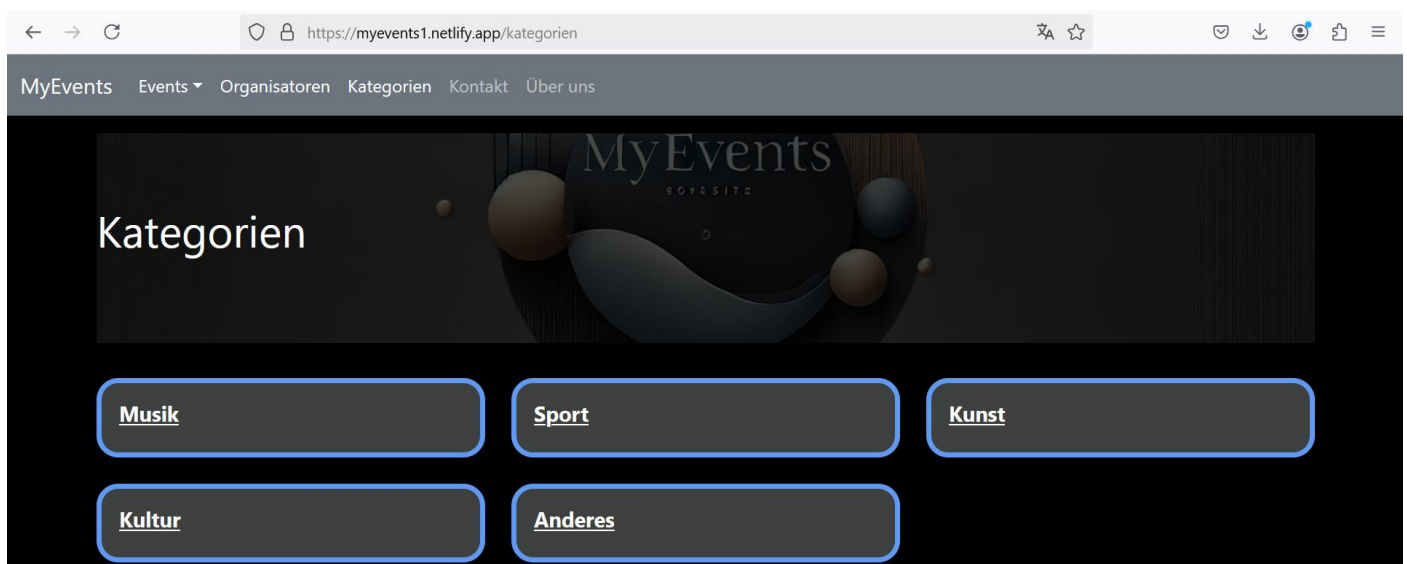
Auf dieser Seite werden die Details eines einzelnen Organisators durch die `getOrganisator(id)` Funktion angezeigt, inklusive des Namens und der E-Mail-Adresse. Darunter befindet sich eine Übersicht aller Events dieses Organisators, welche mit der `getOrganisatorWithEvents(organisator_id)` Funktion angezeigt werden (Siehe Erweiterungen). Die Events werden erneut mithilfe der `EventCard.svelte`-Komponente dargestellt. Nutzer können die Events zur persönlichen Eventliste hinzufügen, durch einen Klick auf den Namen eines Events weitere Details einsehen oder über den „Zurück zu Organisatoren“-Link zur Übersicht aller Organisatoren zurückkehren.

Dateien:

- `lib/components/EventCard.svelte`
- `routes/organisatoren/[organisatoren_id]/+page.svelte`
- `routes/organisatoren/[organisatoren_id]/+page.server.js`
- `routes/styles.css`
- `lib/db.js`

Kategorien

Route: /kategorien



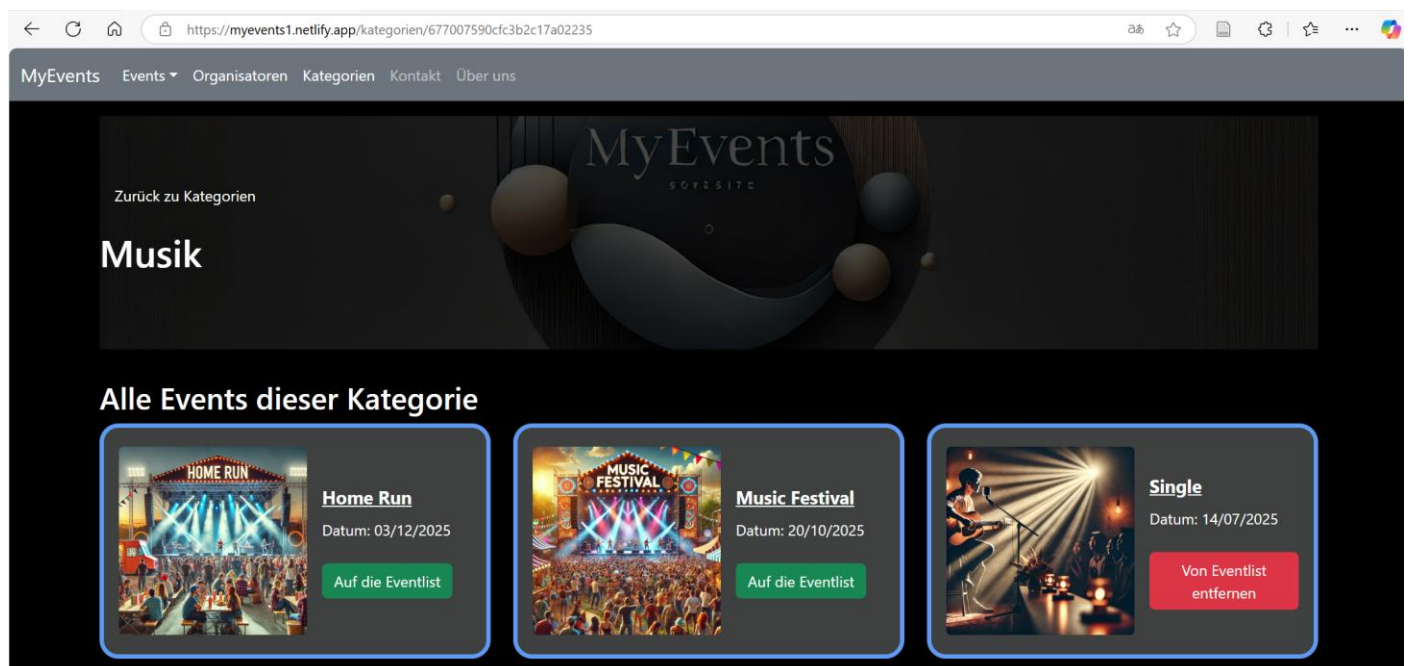
Auf dieser Seite werden die verschiedenen Kategorien durch die Funktion `getKategorien()` angezeigt. Die wurden alle mit der Komponente `KategorieCard.svelte` erstellt. Jede Kategorie ist als klickbarer Button dargestellt. Durch einen Klick auf eine Kategorie gelangt der Nutzer zu `/kategorien/[kategorie_id]`, wo er eine Übersicht aller Events, die dieser Kategorie zugeordnet sind, ansehen kann. Die Seite ermöglicht eine thematische Navigation und hilft Nutzern, schnell passende Veranstaltungen nach ihren Interessen zu finden.

Dateien:

- `lib/components/KategorieCard.svelte`
- `routes/kategorien/+page.svelte`
- `routes/kategorien/+page.server.js`
- `routes/styles.css`
- `lib/db.js`

Kategorie Detailseite

Route: `/kategorien/[kategorie_id]`



Auf dieser Seite werden alle Events mit der Funktion `getKategorieWithEvents(kategorie_id)` angezeigt, die einer bestimmten Kategorie zugeordnet sind (Siehe Erweiterungen). Die durch die Funktion `getKategorie(id)` wird der Name der Kategorie im Titel hervorgehoben, und darunter befinden sich die entsprechenden Events, die mithilfe der `EventCard.svelte`-Komponente dargestellt werden. Nutzer können die Events zur Eventliste hinzufügen, aus der Liste entfernen oder durch einen Klick auf ein Event weitere Details einsehen. Ein „Zurück zu Kategorien“-Link ermöglicht es, schnell zur Übersicht aller Kategorien zurückzukehren.

Dateien:

- `lib/components/EventCard.svelte`
- `routes/kategorien/[kategorie_id]/+page.svelte`
- `routes/kategorien/[kategorie_id]/+page.server.js`
- `routes/styles.css`
- `lib/db.js`

Erweiterungen

Dropdown Menu



Im Navbar wurde ein Dropdown-Menü mit Bootstrap Komponenten hinzugefügt. Es ermöglicht den Nutzern zwischen der Ansicht aller Events und der Eventlist zu wechseln. Über die Dropdown-Optionen wird der Nutzer entweder auf die Seite /events (alle Events) oder /events/eventlist (Nur Events auf Eventlist) weitergeleitet.

Datei:

- routes/+layout.svelte

Carousel Komponente

Auf der Startseite wurde ein Carousel implementiert, das mithilfe von Bootstrap-Komponenten realisiert wurde.

Datei:

- routes/+page.svelte

Hero Komponente

Für die Seiten Events, Eventlist, Kategorien und Organisatoren wurde eine wiederverwendbare Hero-Komponente integriert, um eine einheitliche und visuell ansprechende Überschrift zu schaffen. Der Hero-Bereich wurde über CSS in der Datei styles.css gestaltet und individuell in den jeweiligen +page.svelte-Dateien genutzt.

Dateien:

- routes/styles.css
- routes/events/+page.svelte
- routes/events/eventlist/+page.svelte
- routes/organisatoren/+page.svelte
- routes/organisatoren/[organisator_id]/+page.svelte
- routes/kategorien/+page.svelte
- routes/kategorien/[kategorie_id]/+page.svelte

Create Dropdown

Auf der Create Event-Seite /events/create wurde die Funktionalität erweitert, indem zwei Dropdown-Menüs hinzugefügt wurden. Diese ermöglichen es Nutzern, Kategorien und Organisatoren direkt aus der Datenbank auszuwählen. Dadurch wird sichergestellt, dass nur gültige Werte gespeichert werden können. Die Funktionen getKategorien() und getOrganisatoren() dienen zur Abfrage der Datenbank.

Dateien:

- routes/events/create/+page.svelte
- routes/events/create/+page.server.js
- lib/db.js

Event hinzufügen

Name

Datum



Füllen Sie dieses Feld aus.

In den Standardeingabefeldern des Formulars wurde die Erweiterung implementiert, dass diese zwingend ausgefüllt werden müssen. Dies wurde mithilfe des required-Attributs sichergestellt.

Dateien:

- routes/events/create/+page.svelte

Navigation zu weiteren Events und Organisatoren

Auf der Event-Detailseite /events/ [event_id] wurde eine Erweiterung implementiert, die es Nutzern ermöglicht, durch Klick auf den Organisator oder die Kategorie, weitere Events des jeweiligen Organisations oder der jeweiligen Kategorie anzusehen. Diese Funktionalität basiert auf den Datenbankabfragen `getEventsWithOrganisator(event)` und `getEventsWithKategorie(event)`, die in der Datei `db.js` definiert sind.

Dateien:

- routes/events/[event_id]/+page.svelte
- routes/events/[event_id]/+page.server.js
- lib/db.js

Events von bestimmten Organisatoren und Kategorien anzeigen

Auf den Detailseiten von Organisatoren und Kategorien wurde eine Erweiterung implementiert. Sie ermöglicht es Nutzern, durch Klick auf den Organisator oder Kategorie, weitere Events des jeweiligen Organisations oder der Kategorie anzusehen. Diese Funktionalität basiert auf den Datenbankabfragen `getEventsWithOrganisator(event)` und `getEventsWithKategorie(event)`, die in der Datei `db.js` definiert sind.

Dateien:

- routes/organisatoren/[organisator_id]/+page.svelte
- routes/organisatoren/[organisator_id]/+page.server.js
- routes/kategorien/[kategorie_id]/+page.svelte
- routes/kategorien/[kategorie_id]/+page.server.js
- lib/db.js

Anpassung der Zurück Buttons

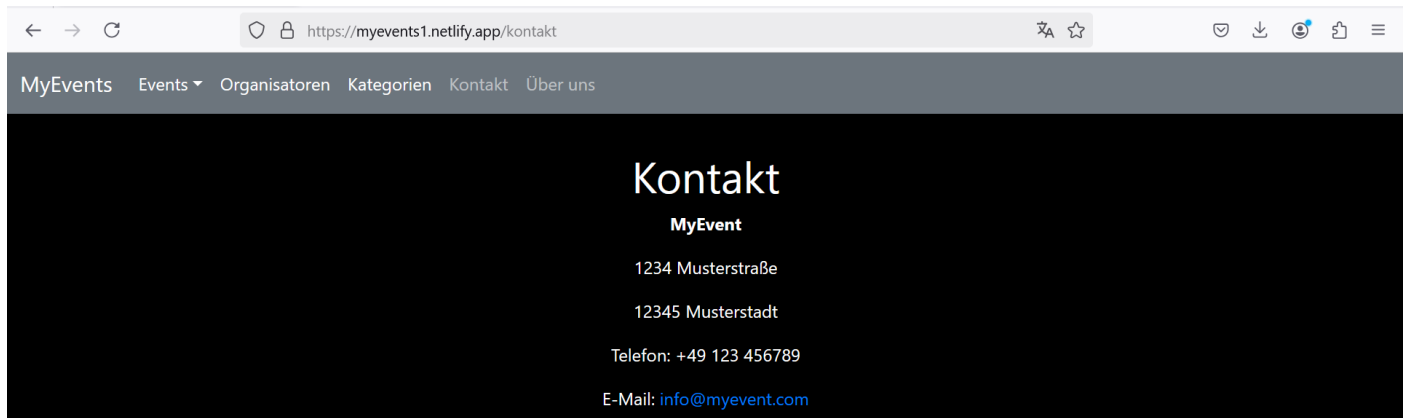
Die Back-Buttons wurden im `styles.css` angepasst, damit sie einheitlich auf den jeweiligen Detailseiten angezeigt werden. Zusätzlich wurde ein Hover-Effekt hinzugefügt, der bei Interaktion eine visuelle Rückmeldung gibt.

Dateien:

- routes/styles.css
- routes/events/[event_id]/+page.svelte
- routes/organisatoren/[organisator_id]/+page.svelte
- routes/kategorien/[kategorie_id]/+page.svelte

Zusätzliche Seite Kontakt

Route: /kontakt



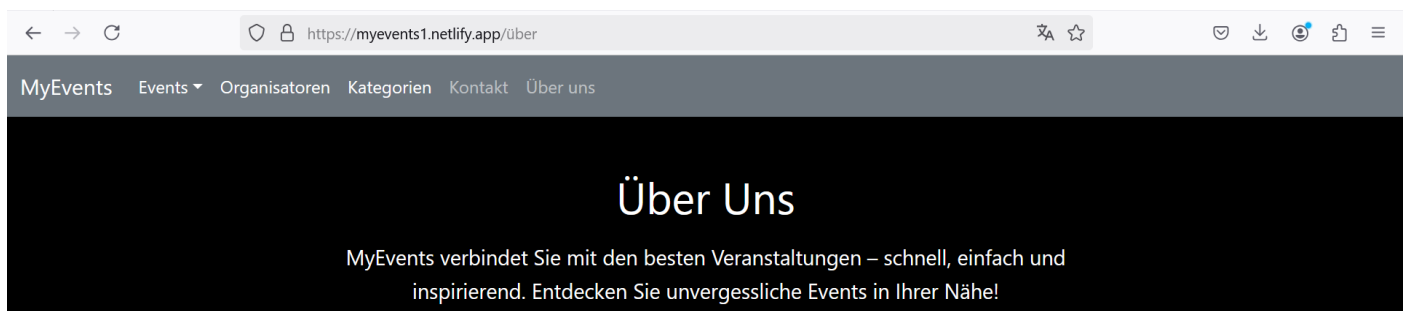
Auf der Kontaktseite werden die wichtigsten Kontaktdaten von MyEvents angezeigt. Hier finden Nutzer die Adresse, Telefonnummer und E-Mail-Adresse des Unternehmens, um bei Bedarf Kontakt aufzunehmen. Diese Seite dient ausschliesslich zur Information und enthält keine interaktiven Funktionen.

Dateien:

- routes/kontakt/+page.svelte

Zusätzliche Seite Über uns

Route: /über



Auf der „Über Uns“-Seite wird die Mission von MyEvents kurz und prägnant dargestellt. Diese Seite dient lediglich der Information und bietet keinen interaktiven Inhalt.

Dateien:

- routes/über/+page.svelte