

Prototyping Projektdokumentation

Name: Arijan Gashi

E-Mail: gashiari@students.zhaw.ch

URL der deployten Anwendung: <https://reiseplanergasari.netlify.app/>

Einleitung

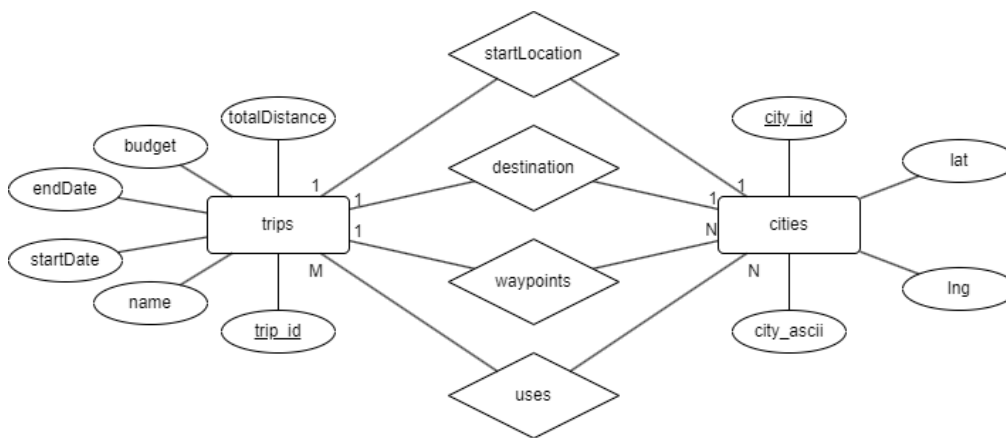
Der Reiseplaner ist eine webbasierte Anwendung, die es den Nutzern ermöglicht, Reisen zu planen, zu verwalten und übersichtlich darzustellen. Die Hauptidee besteht darin, eine intuitive Plattform bereitzustellen, die sowohl für die Planung einfacher Reisen als auch für komplexere Reiserouten mit Zwischenzielen geeignet ist.

Grundfunktionen:

- Erstellung neuer Reisen mit Start- und Zielorten, einem Budget und einem Zeitrahmen.
- Hinzufügen von Zwischenzielen mit spezifischen Daten.
- Visualisierung der Route auf einer interaktiven Karte.
- Bearbeitungs- und Löschfunktion für bestehende Reisen.

Die App bietet eine benutzerfreundliche Oberfläche und flexible Features, die den Anforderungen von Vielreisenden gerecht werden. Die Entscheidung, einen Reiseplaner zu entwickeln, basiert auf der Beobachtung, dass viele bestehende Tools entweder zu simpel oder zu kompliziert für den Alltag sind.

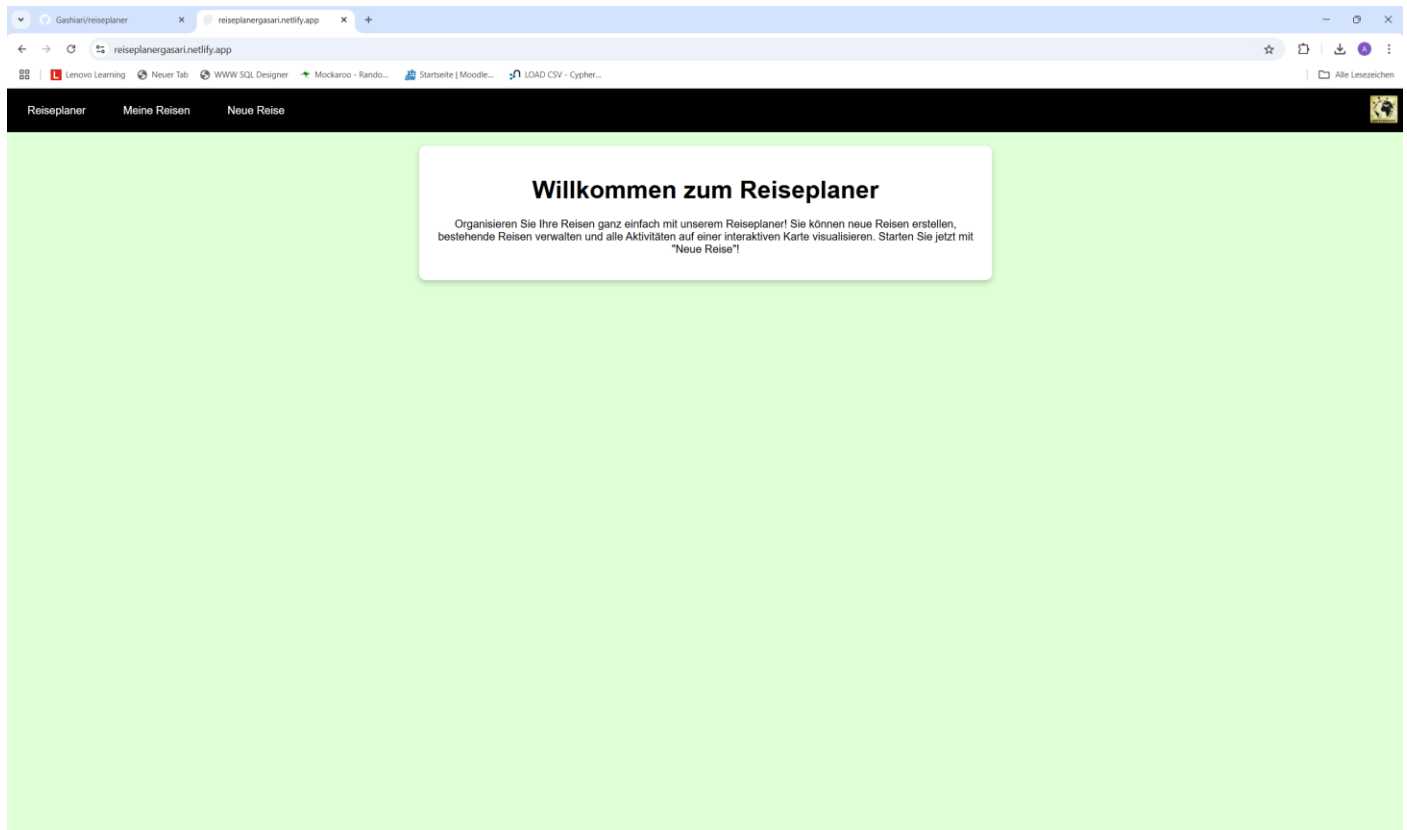
Datenmodell



Beschreibung der Anwendung

Startseite und Navigation

Route: /



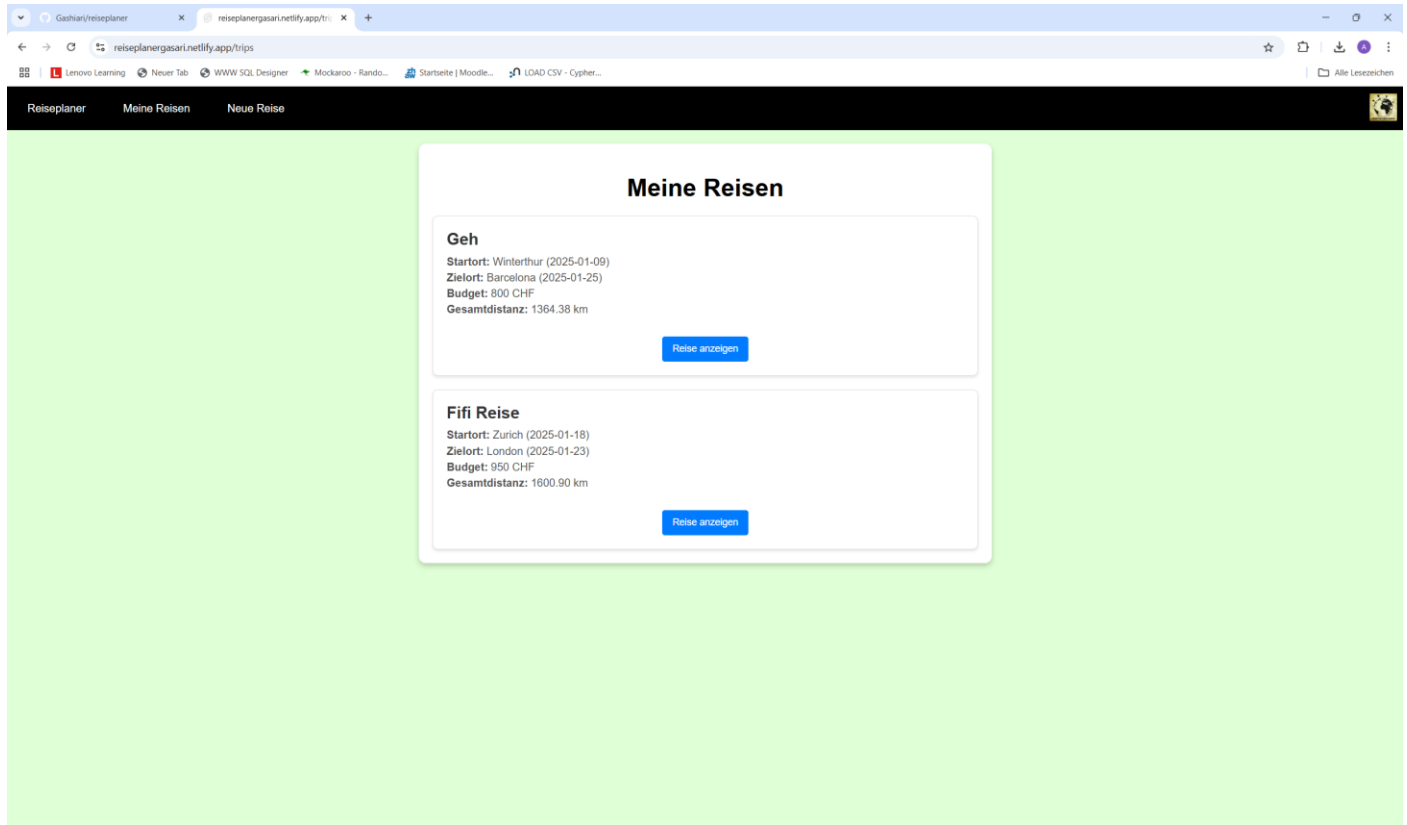
Die Startseite bietet einen Überblick und ermöglicht den Einstieg in die zentralen Funktionen: Reiseübersicht, neue Reise erstellen und Details einzelner Reisen. Eine einheitliche Navigation und globale Styles sorgen für ein konsistentes Erscheinungsbild.

Dateien:

- `src/routes/+layout.svelte`
- `src/routes/+page.svelte`
- `static/global.css`

Meine Reisen

Route: /trips



Diese Seite zeigt alle angelegten Reisen in einer Kartenansicht. Über die Karten gelangt der Nutzer direkt zu den Details einzelner Reisen. Die Daten werden dynamisch aus der MongoDB geladen.

Dateien:

- `src/routes/+layout.svelte`
- `src/routes/trips/+page.svelte`
- `src/routes/api/trips/+server.js`
- `static/global.css`

Neue Reise

Route: /trips/new

Neue Reise erstellen

Name der Reise:

Startort:

Zielort:

Startdatum:

Enddatum:

Budget (CHF):

Zwischenziele:

Zwischenziel hinzufügen

Speichern

Gesamtdistanz: 0.00 km

Map of Zurich showing the current route and distance.

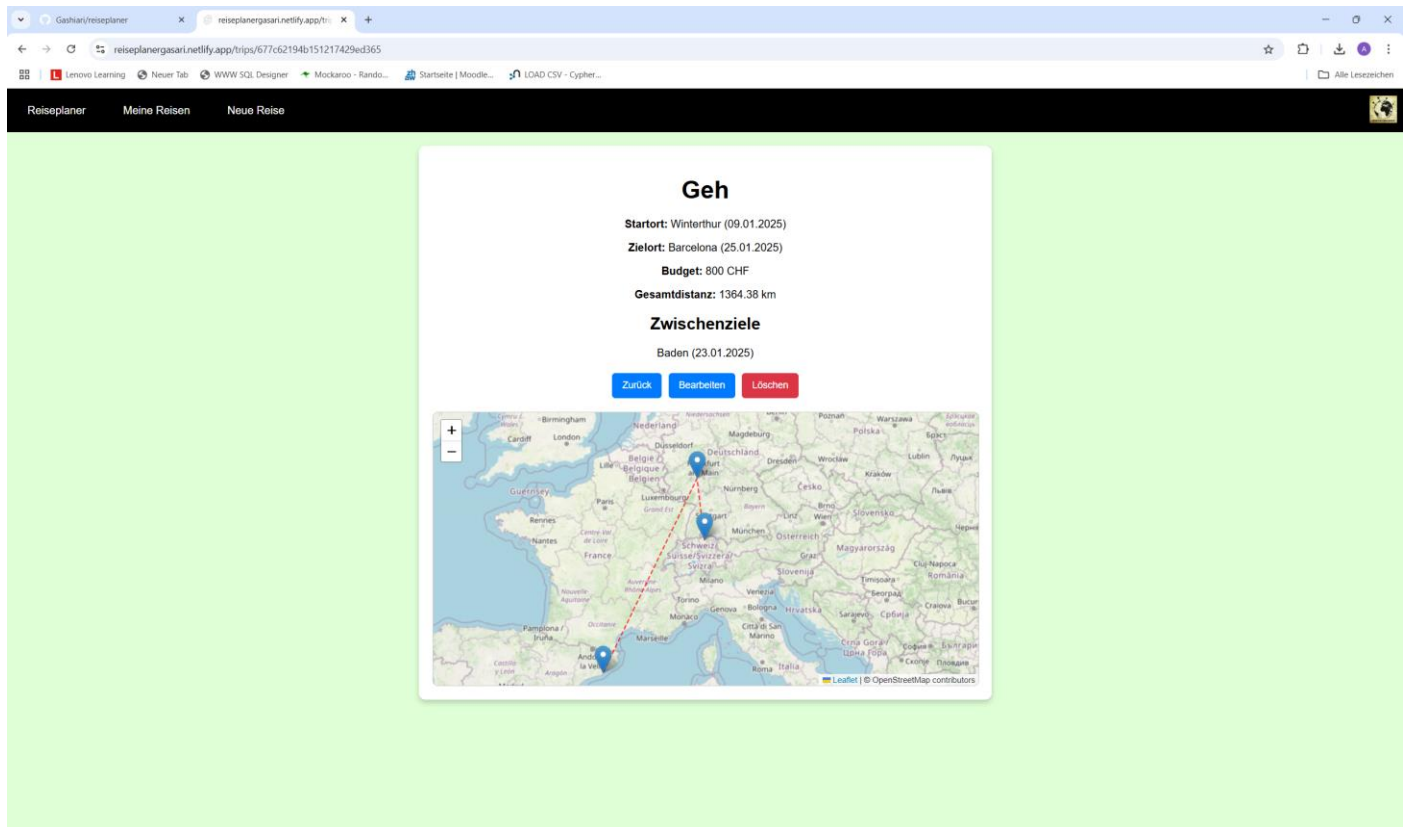
Auf der Seite „Neue Reise“ können Reisen erstellt werden. Ein Formular mit interaktiver Karte ermöglicht die Eingabe von Namen, Orten und Daten. Validierungen verhindern fehlerhafte Eingaben, und die Route wird in Echtzeit angezeigt.

Dateien:

- `src/routes/+layout.svelte`
- `src/routes/trips/new/+page.svelte`
- `src/routes/api/trips/+server.js`
- `src/lib/db.js`
- `static/global.css`

Detailansicht einer Reise

Route: /trips/[id]



Die Detailseite zeigt Reiseinformationen wie Namen, Budget und Route. Funktionen wie Bearbeiten und Löschen sind integriert. Falls keine Zwischenziele existieren, wird der Abschnitt ausgeblendet.

Dateien:

- src/routes/+layout.svelte
- src/routes/trips/[id]/+page.svelte
- src/routes/api/trips/[id]/+server.js
- src/lib/db.js
- static/global.css

Erweiterungen

Bearbeitungsmodus

Ermöglicht das Bearbeiten von Budget, Start- und Enddatum sowie Zwischenzieldaten in der Detailansicht.

Dateien:

- src/routes/trips/[id]/+page.svelte
- src/routes/api/trips/[id]/+server.js

Interaktive Karte mit Polylinien

Visualisiert die Reise auf einer interaktiven Karte mit Markern und einer Polyline zwischen Start-, Ziel- und Zwischenzielen.

Dateien:

- src/routes/trips/+page.svelte
- src/routes/trips/[id]/+page.svelte

Löschen mit Bestätigung

Fügt einen Bestätigungsdialog hinzu, bevor eine Reise gelöscht werden kann.

Datei:

- `src/routes/trips/[id]/+page.svelte`

Validierung der Daten

Eingaben wie Start- und Enddatum werden überprüft, um fehlerhafte Eingaben zu vermeiden. Solche Validierungsregeln bieten eine benutzerfreundliche und sichere Umgebung.

Dateien:

- `src/routes/trips/+page.svelte`
- `src/routes/trips/[id]/+page.svelte`

Responsive Design

Die App ist für unterschiedliche Geräte optimiert. Die Navigation und Inhalte passen sich automatisch an die Bildschirmgröße an.

Datei:

- `src/routes/+layout.svelte`

API-Integration

Die API dient als Schnittstelle zwischen Frontend und Backend und ermöglicht eine zentrale Datenverwaltung. Sie unterstützt grundlegende Funktionen wie das Abrufen (GET), Hinzufügen (POST), Aktualisieren (PUT) und Löschen (DELETE) von Daten. Dadurch wird die Anwendung modular, flexibel und zukunftssicher.

Dateien:

- `src/lib/db.js`
- `src/routes/api/places/server.js`
- `src/routes/api/trips/+server.js`
- `src/routes/api/trips/[id]/+server.js`