

# Love Sender

**Leistungsnachweis 2 im Modul Prototyping**  
im Studiengang Wirtschaftsinformatik, Business Information Systems

Vorgelegt von:  
Bastian Steingruber, [steinba6@students.zhaw.ch](mailto:steinba6@students.zhaw.ch)

am  
08.01.2025

## Inhaltsverzeichnis

<b>Kurzbeschrieb .....</b>	<b>2</b>
<b>ER-Diagramm.....</b>	<b>3</b>
<b>Beschrieb der einzelnen Pages .....</b>	<b>4</b>
Authentifizierung .....	4
Startseite.....	5
Get Love Seite .....	6
Memories Seite.....	7
Memories Detail Seite.....	8
Neue Memory erstellen Seite.....	9
Kategorien Seite .....	10
Kategorien-Memories Seite .....	11
Gegen Langeweile Dropdown .....	12
Video-Page.....	12
Foto-Page .....	13
<b>Erweiterungen .....</b>	<b>14</b>

## Kurzbeschrieb

Diese Webapplikation wurde mit Svelte entwickelt. Sie ist passwortgeschützt und nutzt eine Cookie-basierte Session, die eine Stunde gültig ist, um die Sicherheit zu gewährleisten. Dies war mir wichtig, da persönliche Daten auf der Webseite zugänglich gemacht werden. Ursprünglich als Abschlussprojekt für die ZHAW konzipiert, wurde die Applikation zusätzlich so gestaltet, dass sie als Geschenk und persönliche Plattform für meine Freundin dient.

Die Webseite dient als zentraler Ort, um bedeutungsvolle Erinnerungen zu bewahren und sie auf vielfältige Weise erlebbar zu machen. Nach einem langen Tag könnte jemand beispielsweise die "Memories"-Seite nutzen, um schöne gemeinsame Momente anzusehen. Verschiedene Filterfunktionen, wie zum Beispiel das Filtern nach dem Jahr oder Kategorien-Seite mit den Kategorien "Ferien", "Zuhause" oder "Andere", können genutzt werden, um gezielter zu stöbern. Mit der "Get Love"-Funktion können schnell kleine Gesten der Wertschätzung initiiert werden. Für entspannte Unterhaltung bietet das "Gegen Langeweile"-Dropdown Zugang zu persönlichen Videos oder einer Slideshow mit Hintergrundmusik. Für die Präsentation der Website wurde ein weiteres Passwort für die Dozenten generiert, mit diesem der Zugang gewährleistet wird.

URL: <https://lovesender.netlify.app/>

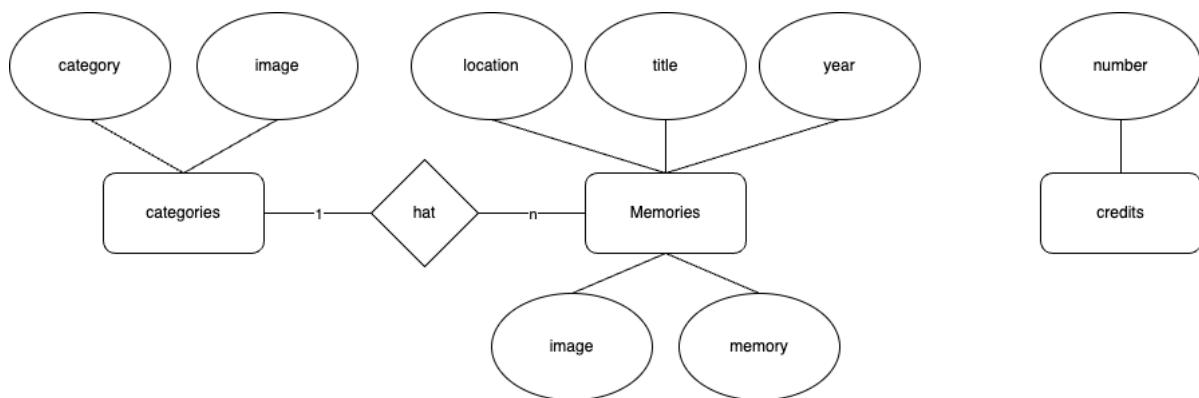
Verschlüsseltes Passwort für Dozenten: Anweisungen stehen im README File in /static/encryptedPassword

## ER-Diagramm

Das ER-Modell zeigt die Struktur der MongoDB Datenbank, die für die Webapplikation verwendet wurde. Neben der zentralen Kollektion "Memories", die den Kern der Anwendung bildet, wurden zwei zusätzliche Kollektionen hinzugefügt: "Categories" und "Credits". Diese erweitern die Funktionalität der Applikation über die Grundanforderungen hinaus.

- Memories: Enthält Memories, die mit Titel, Ort, Jahr, dem Status ob sie schon gemacht wurde oder noch nicht und einer zugeordneten Kategorie gespeichert werden. Jede Memory kann zusätzlich mit einem Bild versehen werden, um die visuelle Darstellung zu ergänzen.
- Categories: Enthält die verschiedenen Kategorien, die mit den Memories verknüpft werden können. Auch hier wird für das Cover der Category Komponenten ein Pfad zu einem Bild mitgegeben.
- Credits: Dient der Verwaltung des Credits-Systems für das Versenden der E-Mails.

Obwohl in NoSQL-Datenbanken grundsätzlich keine Foreign Keys verwendet werden, wurde in diesem Projekt bewusst eine Beziehung zwischen "Categories" und "Memories" geschaffen. Dies war notwendig, um den Anforderungen der Aufgabenstellung gerecht zu werden und eine klare Verbindung zwischen den Daten sicherzustellen.



## Beschrieb der einzelnen Pages

### Authentifizierung

Route: /

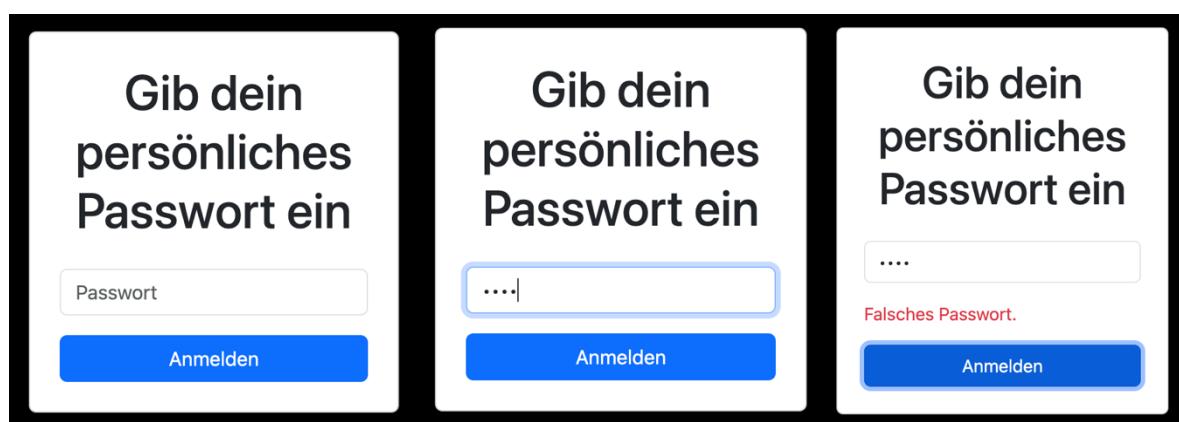
Die gesamte Applikation ist passwortgeschützt. Die Authentifizierung erfolgt durch eine Cookie-basierte Session, die für eine Stunde gültig ist. So ist die Seite auch gegen Bypassing via der Seiten-URL geschützt.

#### Technische Details:

- **Session-Management:** Implementiert über Cookies.
- **Sicherheitsmassnahmen:** Kein direkter URL-Zugriff ohne gültige Session, Passwort wurde in einer Umgebungsvariable gespeichert (nicht im Code).

#### Dateien:

- /routes/+page.svelte
- /routes/check-password/+server.js
- /routes/styles.css



## Startseite

*Route: /menue*

Die Startseite zeigt ein zufällig ausgewähltes Bild aus einem lokalen Verzeichnis. Das Bild ändert sich bei jedem neuen Seitenaufruf und sorgt so für Abwechslung.

### Technische Details:

- Bilder werden aus einem definierten Verzeichnis geladen.
- Zufallsauswahl erfolgt serverseitig und wird bei jedem Zugriff neu berechnet.

### Dateien:

- /routes/menue/+page.svelte
- /routes/menue/+layout.svelte
- /routes/menue/+layout.server.js
- /routes/api/images/+server.js
- /components/db.js
- /routes/styles.css



## Get Love Seite

Route: /menue/getLove

Diese Seite ermöglicht es, eine E-Mail an den Partner zu senden, um an einen liebevollen Text zu erinnern. Ein Credits-System stellt sicher, dass maximal 25 Nachrichten gesendet werden können, bevor eine Gegenleistung des anderen Partners gemacht werden muss.

### Funktionen:

- Versenden von E-Mails über **Nodemailer** mit **SMTP**.
- Credits (25) werden in einer separaten MongoDB-Collection gespeichert.
- Wenn die Credits aufgebraucht sind, wird der Senden-Button deaktiviert und eine Nachricht angezeigt.
- Hinweis: Das Senden der E-Mail ist aktuell nicht möglich, da die verwendete E-Mail-Adresse aufgrund zu vieler Tests von Google gesperrt wurde.



### Dateien:

- /routes/menue/getLove/+page.svelte
- /routes/menue/getLove/+page.server.js
- /functions/emailSender.js
- /routes/api/send-email/+server.js
- /routes/styles.css

The image contains two side-by-side screenshots of a web application. Both screenshots have a header 'Hallo love, brauchst du einen Text mit Liebe?'.  
Left screenshot: Below the header is the text 'Drücke auf den Knopf unten!'. A blue button labeled 'Liebestext bekommen' is centered. At the bottom, it says 'Du hast noch 18 Credits übrig'.  
Right screenshot: Below the header is the text 'Drücke auf den Knopf unten!'. A blue button labeled 'Liebestext bekommen' is centered. At the bottom, it says 'Du hast noch 17 Credits übrig'. A light blue bar at the bottom indicates 'Deine Anfrage wurde erfolgreich versendet!'.  
A vertical line separates the two screenshots.

## Memories Seite

Route: /menue/memories

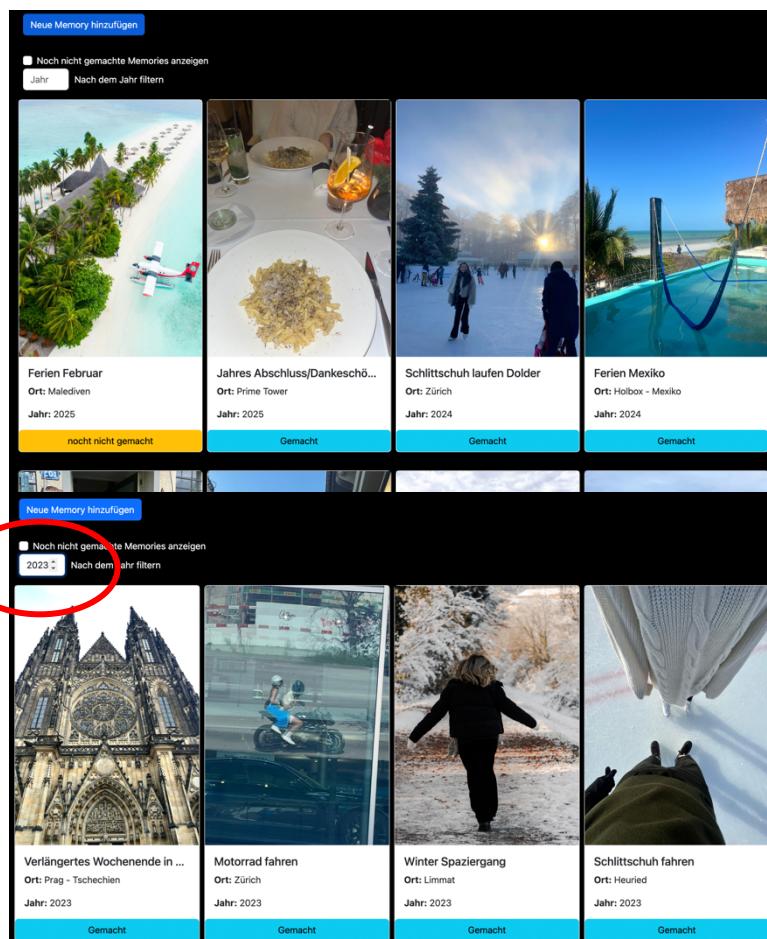
Die Memories-Seite zeigt Erinnerungen, die aus einer MongoDB-Kollektion geladen werden. Die Erinnerungen können nach ihrem Status (gemacht/noch nicht gemacht) oder einem bestimmten Jahr gefiltert werden. Der Status kann über den Klick auf den Status-Button bei den jeweiligen Memories geändert werden.

### Technische Details:

- **Datenhaltung:** Alle Memories werden in MongoDB gespeichert.
- **Filteroptionen:** Erinnerungen können nach "gemacht/nicht gemacht" oder nach Jahr gefiltert werden.
- **Verwendung einer Komponente:** Die einzelnen Memories werden über eine Komponente präsentiert.
- **Präsentation:** Die Memories werden verkehrt herum angezeigt, so werden die neusten Memories ganz oben präsentiert.

### Dateien:

- /routes/menue/memories/+page.svelte
- /routes/menue/memories/+page.server.js
- /components/MemoryCard.svelte
- /components/db.js
- /routes/styles.css



## Memories Detail Seite

**Route:** /menue/memories/[id]

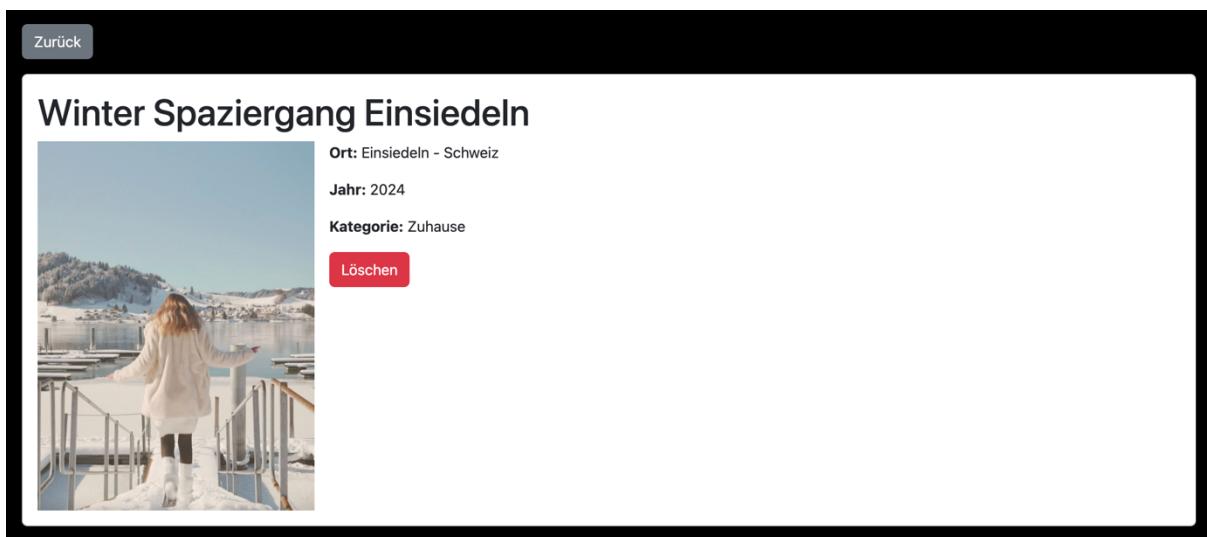
Die Detailseite zeigt die vollständigen Informationen einer Memory, darunter Titel, Ort, Jahr, Kategorie und das zugehörige Bild. Über einen Button kann die Memory gelöscht werden. Ein Zurück-Button ermöglicht die Navigation zurück zur Memories-Seite.

### Technische Details:

- **Auslesen einer Memory:** Die Daten werden basierend auf der Memory-ID aus der MongoDB geladen.
- **Löschen der Memory:** Die Löschfunktion entfernt die Memory aus der Datenbank und leitet den Nutzer zur Memories-Seite zurück.
- **Verwendung einer Komponente:** Die Detail-Ansicht wird über eine Komponente präsentiert.

### Dateien:

- /routes/menue/memories/[memory\_id]/+page.svelte
- /routes/menue/memories/[memory\_id]/+page.server.js
- /components/MemoryDetails.js
- /components/db.js
- /routes/styles.css



## Neue Memory erstellen Seite

Route: /menue/memories/create

Die Seite „Neue Memory erstellen“ ermöglicht es, neue Erinnerungen in die MongoDB-Kollektion einzufügen. Nutzer können wichtige Informationen wie Name, Ort, Jahr, Kategorie und ein Bild angeben, um eine Memory vollständig zu definieren.

### Technische Details:

- **Validierung:** Eingaben wie Jahr werden überprüft, um sicherzustellen, dass nur gültige Daten gespeichert werden.
- **Verwendung einer Komponente:** Das Formular wird über eine Komponente präsentiert.
- **Datenhaltung:** Alle eingegebenen Informationen werden direkt in der MongoDB-Kollektion gespeichert, auch das Bild.
- **Kategorie-Integration:** Kategorien werden dynamisch aus der MongoDB geladen und im Dropdown-Menü angezeigt.
- **Routing:** Dynamische Navigation führt den Nutzer nach dem Speichern automatisch zur Memories-Seite.

### Dateien:

- /routes/menue/memories/create/+page.svelte
- /routes/menue/memories/create/+page.server.js
- /components/AddMemory.js
- /components/db.js
- /routes/styles.css

The screenshot shows a user interface for creating a new memory. On the left, there's a main form with input fields for Name, Ort, Jahr, and Kategorie. The Kategorie field is a dropdown menu with three options: Ferien, Zuhause, and Andere. The 'Ferien' option is currently selected. There's also a checkbox for 'Memory schon gemacht'. Below these is a file input for Memory Cover. At the bottom of the form is a blue 'Memory hinzufügen' button. On the right, a modal window displays the expanded Kategorie dropdown with the same three options.

## Kategorien Seite

Route: /menue/categories

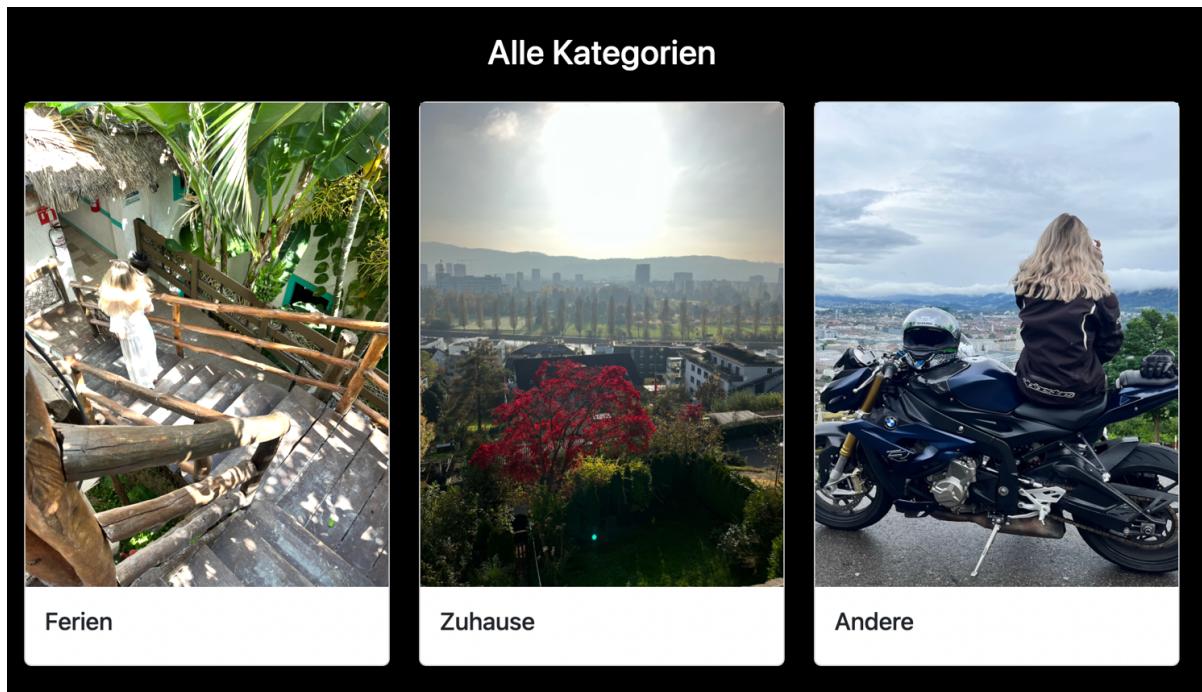
Diese Seite zeigt eine Übersicht aller Kategorien, die in einer MongoDB-Kollektion gespeichert sind.

### Technische Details:

- **Verwendung einer Komponente:** Die einzelnen Kategorien werden über eine Komponente präsentiert.
- **Auslesung aus der DB:** Die Daten werden aus der MongoDB ausgelesen.

### Dateien:

- /routes/menue/categories/+page.svelte
- /routes/menue/categories/+page.server.js
- /components/CategoryCard.js
- /components/db.js
- /routes/styles.css



## Kategorien-Memories Seite

Route: /menue/categories/[category\_id]

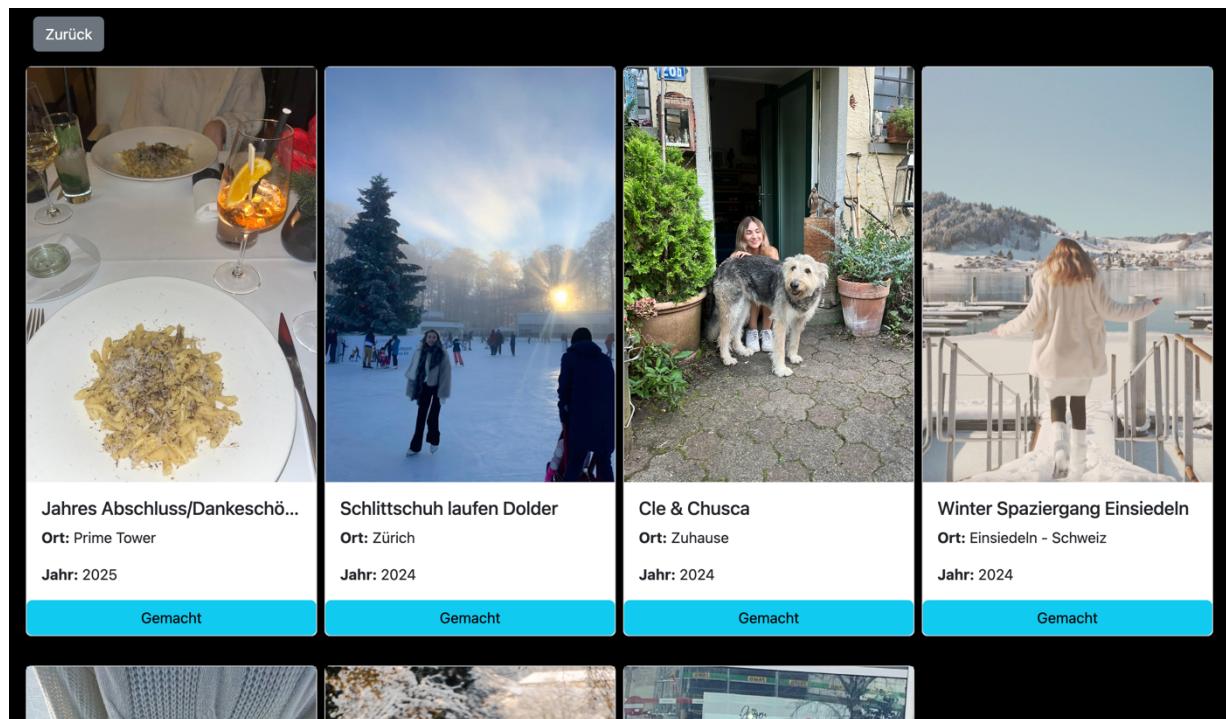
Diese Seite zeigt alle in der Datenbank gespeicherten Memories von einer bestimmten Kategorie.

### Technische Details:

- **Memories einer Kategorie:** Es werden nur die Memories von der gewählten Kategorie angezeigt.
- **Komponenten-Wiederverwendung:** Die gleiche Komponente wie auf der Memories-Seite wird hier eingesetzt, um Redundanz zu vermeiden.
- **Navigation:** Zurück-Button leitet den Nutzer zur Kategorienübersicht.
- **Filteroptionen:** Erinnerungen können nach "gemacht/nicht gemacht" gefiltert werden.
- **Präsentation:** Die Memories werden verkehrt herum angezeigt, so werden die neusten Memories ganz oben präsentiert.
- **Detail-Ansicht der Memory:** Auch hier wird mit einem Klick auf eine Memory, auf die Detailansicht verwiesen, wo man die Memory anschliessen auch löschen kann.

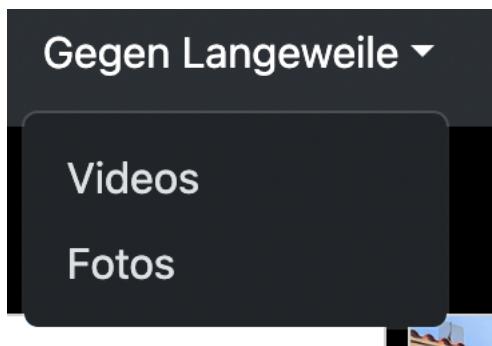
### Dateien:

- /routes/menue/categories/[category\_id]+page.svelte
- /routes/menue/categories/[category\_id]+page.server.js
- /components/MemoryCard.js
- /components/MemoryDetails.js
- /components/db.js
- /routes/styles.css



## Gegen Langeweile Dropdown

Das Dropdown bietet zwei Optionen: **Videos** und **Fotos**.



## Video-Page

Route: /menue/noMoreBoredVideos

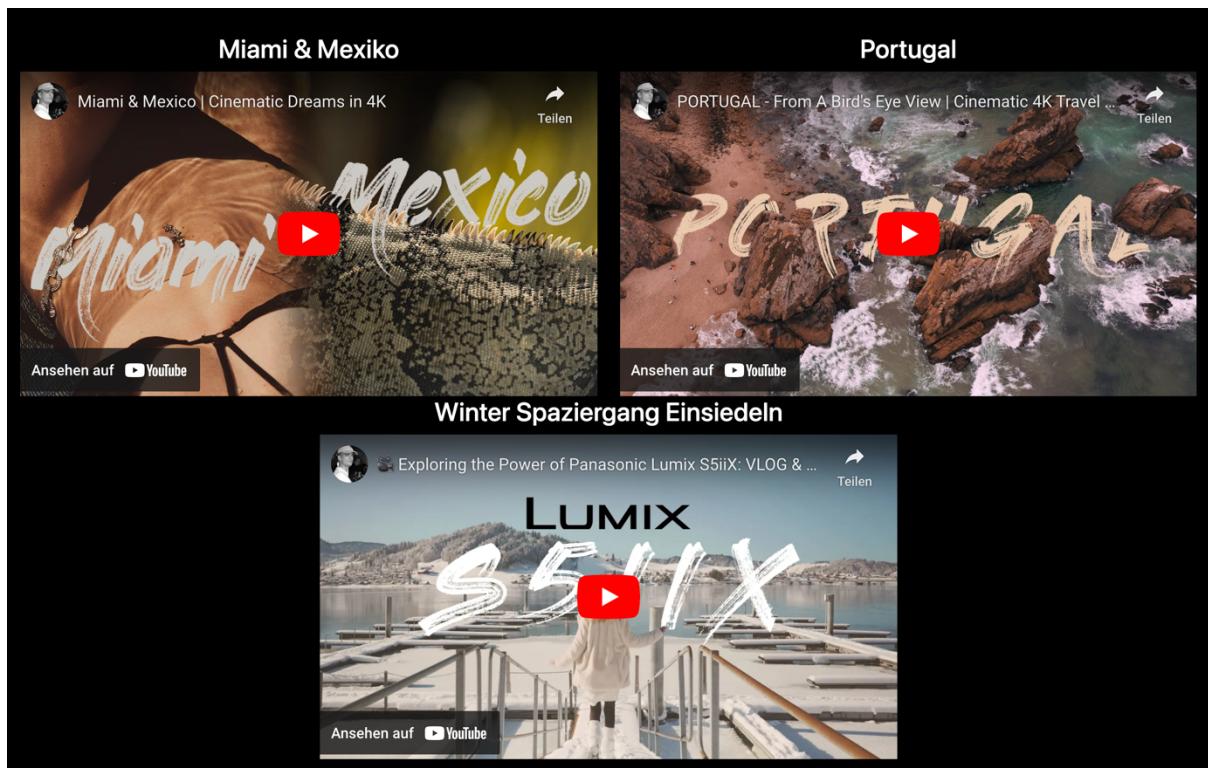
Diese Seite Präsentiert drei eingebettete YouTube Videos, die ebenfalls Erinnerungen hervorheben. Durch das Einbetten hat man vollen Zugriff auf das Video, genau wie, wenn man das Video auf YouTube anschaut.

### Technische Details:

- **YouTube-Embedding:** Videos werden mithilfe der YouTube-API eingebettet.

### Dateien:

- /routes/menue/noMoreBoredVideos/+page.svelte
- /routes/styles.css



## Foto-Page

Route: /menue/noMoreBoredPhotos

Eine Slideshow zeigt Bilder aus einem lokalen Verzeichnis. Die Bilder wechseln im 5-Sekunden-Intervall und können von einer Hintergrundmusik begleitet werden.

### Technische Details:

- **Bilder:** Zufällige Reihenfolge aus einem definierten Verzeichnis.
- **Musiksteuerung:** Zwei Buttons zum Starten und Stoppen der Hintergrundmusik.

### Dateien:

- /routes/menue/noMoreBoredPhotos/+page.svelte
- /routes/styles.css

Lass dich fallen, in die Welt der Geschichten



Mit Musik

Musik stoppen

# Erweiterungen

## 1. Authentifizierung

*Route:* /

Implementierung eines passwortgeschützten Systems mit einer Cookie-basierten Session. Das zur Validierung verwendete Passwort wurde in einer Umgebungsvariablen gespeichert, anstatt direkt im Code hinterlegt zu sein.

## 2. Startseite

*Route:* /menue

Implementierung einer Bilder Collage, die zufällig ein Bild von einem Ordner mit Bildern auswählt.

## 3. Get Love Page (Mail versenden)

*Route:* /menue/getLove

Implementierung einer neuen Seite für ein E-Mail-Versandsystem, das pro E-Mail einen Credit verbraucht. Die Credits werden in einer separaten Datenbank-Collection verwaltet. Sobald alle Credits aufgebraucht sind, wird die Versandfunktion deaktiviert. Der Versand wurde mit der Nodemailer Library gemacht.

## 4. Nach Jahr filtern

*Route:* /menue/memories

Hinzufügen einer Filterfunktion, um nur Memories von einem bestimmten Jahr anzeigen zu können.

## 5. Nach noch nicht gemachten Memories filtern

*Route:* /menue/memories

Hinzufügen einer Filterfunktion, um nur Memories, die noch nicht gemacht wurden, anzeigen zu können.

## 6. Kategorieauswahl über Dropdown

*Route:* /menue/memories/create

Integration eines Dropdown-Menüs bei der Erstellung einer neuen Memory, das alle Kategorien aus der MongoDB-Kollektion «categories» lädt. Der Nutzer kann so gezielt Kategorien auswählen.

## 7. Memory Status beim Erstellen angeben

*Route:* /menue/memories/create

Wenn eine neue Memory erstellt wird, kann direkt im Fenster angegeben werden, ob die Memory schon gemacht wurde oder noch nicht.

## **8. Bild in DB speichern**

*Route: /menue/memories/create*

Neu kann auch optional ein Bild bei dem Erstellen einer neuen Memory mitgegeben werden. Dieses Bild wird in der Datenbank gespeichert. Wenn kein Bild mitgegeben wird, wird ein Template-Bild gewählt.

## **9. Kategorien anzeigen**

*Route: /menue/categories*

Implementierung einer neuen Seite, die mithilfe einer neuen Komponente alle Kategorien ansprechend darstellt.

## **10. Memories einer Kategorie anzeigen**

*Route: /menue/categories/[category\_id]*

Durch Anklicken einer Kategorie werden alle Memories der gewählten Kategorie übersichtlich auf einer neu implementierten Seite angezeigt.

## **11. Einbettung von YouTube-Videos**

*Route: /menue/noMoreBoredVideos*

Integration von drei YouTube-Videos, die direkt innerhalb der Applikation abgespielt werden können.

## **12. Dropdown Menü**

*Route: /menue/*

Implementierung eines Dropdown Menues in der Navigation-Leiste.

## **13. Bild-Karussell**

*Route: /menue/noMoreBoredPhotos*

Implementierung eines automatischen Karussells, das Bilder in zufälliger Reihenfolge alle 5 Sekunden wechselt.

## **14. Einbettung von Musik**

*Route: /menue/noMoreBoredPhotos*

Option, Hintergrundmusik der Foto Seite aktivieren oder zu deaktivieren.

## **15. Reihenfolge der Memories geändert**

*Route 1: /menue/memories*

*Route 2: /menue/categories/[category\_id]*

Die Reihenfolge der Auflistung der Memories auf der Memories Seite, sowie auf der Kategorien Seite, wurde gekehrt. So werden die neusten Memories ganz oben präsentiert.