

# Prototyping Einzelarbeit (Leistungsnachweis 2)

In diesem Dokument wird die Einzelarbeit des Moduls Prototyping beschrieben. Dazu gehört die Aufgabenstellung, die erwarteten Ergebnisse, die Bewertungsgrundlagen und Informationen zum Ablauf der Diskussion der Ergebnisse.

## Aufgabenstellung

In der Einzelarbeit werden die im Modul Prototyping eingeführten Technologien (HTML, CSS, JavaScript, Svelte(Kit), MongoDB) und Tools angewendet, um eine Webanwendung zu implementieren. Die im Unterricht und in den Kleinklassen erarbeiteten Beispielanwendungen (z.B. *Music DB* oder *Movie Watchlist*) dürfen als Ausgangslage verwendet werden.

Die Projektidee ist grundsätzlich frei wählbar, solange die unten beschriebenen Anforderungen erfüllt werden. Die in der Gruppenarbeit (Leistungsnachweis 1) erarbeiteten Artefakte können als Grundlage genutzt werden. Es ist aber auch erlaubt, eine neue Idee umzusetzen. Die Bewertung erfolgt in jedem Fall unabhängig von der Leistung in der Gruppenarbeit. Bei Unklarheiten und Fragen zu Umfang und Inhalt wenden Sie sich bitte an die Dozenten.

**Für das Projekt muss zwingend Svelte 5 im Runes-Mode verwendet werden** (siehe <https://svelte.dev/docs/svelte/what-are-runes>), wie in der Vorlesung behandelt. Wenn die Legacy API (siehe <https://svelte.dev/docs/svelte/legacy-overview>) verwendet wird, müssen Sie mit Abzügen rechnen.

## Empfohlenes Vorgehen zu Beginn des Projekts

Überlegen Sie sich, welche Entitätstypen Sie darstellen bzw. verwalten möchten (z.B. Projekte, Mitarbeitende, ...)

Erstellen Sie eine Skizze der Anwendung. Starten Sie mit einer einfachen Variante. Sie können Ihre Anwendung bei Bedarf später weiter ausbauen.

Starten Sie mit einer der im Unterricht behandelten Beispielanwendungen (*Music DB* oder *Movie Watchlist*) und passen Sie diese an:

- Erstellen Sie weitere Pages für die Svelte-Anwendung, bzw. passen Sie die vorhandenen Pages an.
- Erstellen Sie eine Datenbank (MongoDB) mit Demo-Daten für die Anwendung.
- Erweitern Sie das Svelte-Projekt, damit Sie auf die Daten in der Datenbank zugreifen können.
- Erweitern Sie die Anwendung gemäss den folgenden Anforderungen und Ihren eigenen Ideen.

## Diskussion der Ergebnisse

Die Ergebnisse werden vor Ort in Winterthur während 10 Minuten diskutiert. Das Ziel der Diskussion ist, die Umsetzungsqualität zu bewerten, sowie das Verständnis und die Eigenständigkeit bei der Umsetzung zu überprüfen. **Dabei ist der eigene Laptop mit dem funktionsfähigen Projekt mitzubringen**, um den Code gemeinsam zu analysieren und zu diskutieren.

Die Diskussionen werden im Zeitraum vom 27. Mai bis 6. Juni stattfinden. Die Terminwahl erfolgt via Moodle. Die Freischaltung der Terminwahl wird per E-Mail bekanntgegeben.

In Ausnahmefällen sind frühere Termine nach Absprache mit Max Meisterhans möglich.

## Abgabe des Projekts

Folgende Elemente sind Teil der Abgabe:

- **Sourcecode** des Projekts auf GitHub.
- **Dokumentation** des Projekts auf GitHub **inklusive URL der deployten Anwendung**.  
Verwenden Sie für die Projektdokumentation die zur Verfügung gestellte Vorlage.

Detaillierte Informationen zu GitHub, Dokumentation und Deployment folgen in den letzten beiden Semesterwochen in der Vorlesung und den Kleinklassen.

Die Abgabe erfolgt auf Moodle (URL des GitHub-Repositories).

**Deadline für die Abgabe: 24 Stunden vor dem Termin der Diskussion. Der Code und die Dokumentation dürfen nach der Deadline nicht mehr verändert werden!**

## Bewertung

Die Einzelarbeit wird mit einer Note bewertet (80% der Modulendnote).

Die Punkte werden wie folgt für jede Anforderung (ausser «Deployment», «Dokumentation» und «Erweiterungen») vergeben:

- **0 Punkte**, falls einer oder mehrere der folgenden Punkte zutrifft:
  - Die Anforderung ist nicht erfüllt.
  - Fragen zur Umsetzung können nicht beantwortet werden.
  - Der im Unterricht zur Verfügung gestellte Beispielcode wurde praktisch unverändert verwendet.
- **1 Punkt**, falls einer oder mehrere der folgenden Punkte zutrifft:
  - Die Anforderung ist teilweise erfüllt.
  - Fragen zur Umsetzung können nur teilweise beantwortet werden.
  - Der im Unterricht zur Verfügung gestellte Beispielcode wurde verwendet und nur minimal angepasst (z.B. nur Variablennamen und einzelne Properties geändert)
- **2 Punkte**, falls die Anforderung erfüllt ist und die Fragen zur Umsetzung beantwortet werden können.
- **3 Punkte**, falls die Erwartungen übertroffen wurden.

*Hinweis:* Es werden sowohl **Quantität** als auch **Qualität** (z.B. clean Code, individuelle Lösung, umfangreicher Einsatz der im Unterricht behandelten Technologien, usw.) bewertet.

Zusätzlich werden Punkte vergeben für:

- die **Implementierung von eigenen Erweiterungen, die nicht explizit Teil der Aufgabenstellung sind** (siehe auch Hinweise auf der nächsten Seite)

## Bewertungsskala

Note	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6
Punkte	0	4	8	12	16	20	24	28	32	36	40

Thema	Anforderungen	max.
<b>Dokumentation</b> max. 4 Punkte	<p>Für die Dokumentation wird ein Template zur Verfügung gestellt.</p> <p>Inhalt:</p> <ul style="list-style-type: none"> <li>Kurzbeschreibung der Anwendung, <b>inklusive URL zur deployten Anwendung.</b></li> <li>Datenmodell (ER-Diagramm)</li> <li>Beschreibung der einzelnen Pages und Funktionen inklusive Screenshots.</li> <li>Liste der Erweiterungen, die über die Grundanforderungen hinausgehen, siehe «Erweiterungen». <b>Achten Sie auf Vollständigkeit. Es werden nur die hier beschriebenen Erweiterungen bewertet.</b></li> </ul> <p>Hier wird nur die Form und Vollständigkeit der Dokumentation bewertet. Die Bewertung des Codes erfolgt in den nachfolgenden Punkten.</p>	<b>4</b>
<b>HTML</b> max. 3 Punkte	Erstellen Sie die Struktur und den Inhalt der Webanwendung mit <b>HTML</b> . Integrieren Sie mindestens ein Bild und/oder andere Medien (z.B. ein Video).	<b>3</b>
<b>CSS/Bootstrap</b> max. 6 Punkte	<b>Layout:</b> Verwenden Sie <b>CSS und/oder Bootstrap</b> für das <b>Layout</b> (z.B. Grid, Box Model)	<b>3</b>
	<b>Styling:</b> Verwenden Sie <b>Bootstrap-Komponenten</b> , um die Webanwendung zu bauen <b>und/oder</b> verwenden Sie <b>CSS</b> , um die Elemente (Buttons, Inputs, Tables, ...) zu individualisieren.	<b>3</b>
<b>Webanwendung</b> max. 12 Punkte	<b>Deployment:</b> Die Anwendung muss lauffähig sein, deployed werden und via URL extern erreichbar sein.	<b>3</b>
	<p><b>Inhalt:</b> Verwenden Sie Svelte bzw. SvelteKit, um die Pages zu erstellen:</p> <ul style="list-style-type: none"> <li>Erstellen Sie Pages für mindestens 2 verschiedene Entitätstypen.</li> <li>Mindestens eine Page, auf der die Details einer einzelnen Entität angezeigt werden.</li> <li>Mindestens eine Page mit einem Formular zur Dateneingabe.</li> </ul> <p>Am Beispiel einer Projektdatenbank: Projekte auflisten, Projektdetails anzeigen, neue Projekte erstellen, an einem Projekt beteiligte Mitarbeitende auflisten, ...</p>	<b>3</b>
	<b>Navigation:</b> <ul style="list-style-type: none"> <li>Funktionierende Navigation (z.B. Navigation Bar)</li> <li>Verwenden Sie das in der Vorlesung vorgestellte Routing-System</li> </ul>	<b>3</b>
	<b>Components:</b> <ul style="list-style-type: none"> <li>Erstellen Sie mindestens 5 unterschiedliche Svelte-Komponenten (<i>.svelte</i>-Dateien), z.B. für die einzelnen Pages oder für wiederverwendbare Teil der Anwendung. <b>Eine Page (+page.svelte) zählt als Komponente.</b></li> </ul>	<b>3</b>
<b>Datenbank, Datenverwaltung</b> max. 9 Punkte	<b>Datenbank:</b> Verwenden Sie eine MongoDB mit mindestens 2 Collections, um Ihre Daten zu verwalten. Implementieren Sie mind. 4 verschiedene Funktionen, die auf die Datenbank zugreifen. Mindestens eine dieser Funktion erstellt neue Daten (z.B. insertOne). Sie dürfen die im Unterricht verwendeten Codebeispiele (Datei <i>db.js</i> ) verwenden und anpassen.	<b>3</b>
	<b>Daten laden mit der load-Funktion:</b> Verwenden Sie Datenbankfunktionen (z.B. in lib/db.js) und die load-Funktion (z.B. in +page.server.js) um Daten zu laden und darzustellen (z.B. Liste aller alle Projekte, Detailseite eines einzelnen Projekts)	<b>3</b>
	<b>Daten verwalten mit Form Actions:</b> Erstellen Sie mindestens eine Page, auf der Daten eingegeben und in der Datenbank gespeichert werden können. (Beispiel: Formular, um ein neues Projekt zu erfassen) Verwenden Sie Form Actions und die entsprechenden Datenbankfunktionen, um Daten zu erstellen oder zu verändern.	<b>3</b>
<b>Erweiterungen</b> max. 9 Punkte	<p>Es werden Punkte für die Implementierung von Erweiterungen vergeben, die über die Grundanforderungen hinausgehen.</p> <p><i>Beispiele:</i> Zusätzliche Pages für komplexe Datenverwaltung, zusätzliche interaktive Inhalte, Verwendung zusätzlicher Libraries, Anbindung externer APIs, usw.</p> <p><b>Hinweis:</b> dies ist keine abschliessende Aufzählung. Eigene Ideen sind willkommen und werden in der Bewertung berücksichtigt.) Die Anzahl der Punkte hängt vom Umfang, der Komplexität der Features und der Qualität der Implementierung ab.</p>	<b>9</b>

**Total: 43 Punkte**