



# PROJEKT ARBEIT

Modul 293

Technische Berufsschule  
07.07.2025

Ruchti Lynn  
[Lynn.ruchti@edu.tbz.ch](mailto:Lynn.ruchti@edu.tbz.ch)

## Contents

1. Projektübersicht .....	2
2. Technische Umsetzung .....	3
3. Einsatz von Künstlicher Intelligenz.....	5
4. Herausforderungen & Problemlösungen .....	7
5. Reflexion und persönliche Erkenntnisse .....	8
6. Fazit und Ausblick.....	9
7. Anhang: Tools & Ressourcen .....	10

# 1. Projektübersicht

**Projekttitle:** Funlog (Fun + Blog)

**Blog-Generator mit KI-basierter Unterstützung zur Inhaltserstellung und Codeoptimierung**

**Projektziel:**

Das Ziel dieses Projekts war die Entwicklung einer dynamischen Webanwendung, die auf Basis strukturierter Daten (JSON) automatisch Blogbeiträge erstellt und visuell darstellt. Besonderer Fokus lag auf dem Einsatz von Künstlicher Intelligenz (KI) zur Generierung von Inhalten, zur Codeentwicklung, zur Fehlerbehebung und zur Effizienzsteigerung während des gesamten Entwicklungsprozesses.

## 2. Technische Umsetzung

### 2.1. Struktur

Das Projekt besteht aus einer Kombination von:

- **HTML-Template** zur Darstellung von Blogkarten
- **CSS** für responsives und modernes Styling
- **JavaScript** zur dynamischen Befüllung der Inhalte über DOM-Manipulation
- **JSON-Datenstruktur** zur Trennung von Daten und Logik

### 2.2. Hauptfunktionen

- Dynamische Generierung von Blogbeiträgen aus einem Array von Objekten
- Automatische Zuordnung von Autoreninformationen
- Darstellung von Profilbildern, Hashtags, Interaktionen (Likes, Kommentare)
- Erweiterbarkeit für weitere Features (z. B. Filter, Likesystem, etc.)

### 2.3. Beispielstruktur eines Blogeintrags

```
{
  "id": 1,
  "Title": "Regenwald Tiere",
  "Image": "https://example.com/image.jpg",
  "Article_Title": "Geheime Allianz im Regenwald entdeckt",
  "Article": "Lorem ipsum...",
  "creator_id": 2,
  "Hashtags": ["#Regenwald", "#Tiere"],
  "Likes": 842,
  "Comment_Count": 57,
  "Creation_Date": "2025-05-03"
}
```

## 2.4. Rendering-Logik (gekürzt)

```
const user = creator.find(c => c.id === data.creator_id);
const username = user?.username ?? 'Unbekannt';
...
const html = template
  .replace('{{username}}', username)
  ...
```

## 3. Einsatz von Künstlicher Intelligenz

### 3.1. Motivation für KI-Einsatz

Die Softwareentwicklung, insbesondere bei kleineren Einzelprojekten oder im Bildungskontext, ist oft durch Zeitmangel, fehlende UX-Erfahrung oder mangelnde Testressourcen geprägt. Künstliche Intelligenz bietet eine effiziente Möglichkeit, diese Herausforderungen durch intelligente Vorschläge, automatisches Debugging oder sogar Generierung ganzer Codeblöcke zu lösen.

### 3.2. Verwendete KI-Tools

#### ▪ ChatGPT (OpenAI)

##### Einsatzbereiche:

- Generierung der ersten Datenstruktur (Blogobjekte, Creator-Objekte)
- Erstellung von realitätsnahen, humorvollen Beispieltexten (z. B. über „El Mono Verde“)
- Unterstützung bei der Formatierung, Fehlerbehebung und bei der strukturellen Optimierung von JavaScript
- Konvertierung von Logik in HTML-Templates mit `replace()`

##### Beispiel:

ChatGPT wurde verwendet, um eine Hashtag-Zeile automatisch zeilenumbruchfähig zu machen:

Lösung: `style="white-space: pre-wrap;"`

#### ▪ GitHub Copilot (OpenAI + Microsoft)

##### Einsatzbereiche:

- Autovervollständigung von DOM-Selektoren und -Manipulationen
- Erkennung und Vorschlag von notwendigen If-Abfragen (z. B. bei `user?.username`)
- Optimierung von `forEach/Map`-Schleifen zur Einbindung von JSON-Arrays

**Vorteil:**

Copilot erkennt den Kontext (z. B. HTML-ID vs. class) und schlägt relevante DOM-Operationen vor.

**▪ Cursor (Code-Editor mit integrierter KI)****Einsatzbereiche:**

- Inline-Erklärung komplexer oder fehleranfälliger Zeilen
- Automatisches Refactoring (z. B. Vorschlag zur Funktionalisierung von Template-Injection)
- Visuelle Darstellung von Objektzuweisungen und Debugginghilfe

**3.3. Vorteile des KI-Einsatzes im Projekt**

Kategorie	Vorteile durch KI
Content-Erstellung	Realitätsnahe, humorvolle Artikel-Texte
Codegenerierung	Schnelle Erstellung von Templates & Logik
Fehlerdiagnose	Sofortige Erkennung von undefinierten-Fehlern
Styling	Optimierung von css-Klassen
Produktivität	Deutlich verkürzte Entwicklungszeit

## 4. Herausforderungen & Problemlösungen

### 4.1. Typische Fehlerquellen

- **Case-Sensitivity bei JSON-Feldern:**  
z. B. Username statt username → undefined
- **Fehlender Fallback bei optionalen Daten:**  
→ Einsatz von user?.username ?? 'Unbekannt'

### 4.2. Dynamische Hashtag-Zeilen umbrechen

**Problem:**

Hashtags in einer Zeile ohne Zeilenumbruch

**Lösung (mit ChatGPT):**

```
<p style="white-space: pre-wrap;">${entry.Hashtags.join(' ')}</p>
```

### 4.3. Dynamisches Template-Injection

Die HTML-Templates wurden mit Platzhaltern ({{title}}) definiert, welche über replace() mit den Daten befüllt werden.



## 5. Reflexion und persönliche Erkenntnisse

### 5.1. Was wurde gelernt?

- **Datenstrukturierung mit JSON:** Trennung von Inhalt und Darstellung
- **Dynamisches DOM-Rendering:** Aufbau einer Mini-„Templating-Engine“
- **Modularisierung mit Funktionen:** Wiederverwendbare Logik mit `renderBlogCard`
- **Einsatz von KIs als Assistenz, nicht Ersatz:** KIs helfen beim *Verstehen*, nicht nur beim Erstellen

### 5.2. Rolle von KI im Entwicklungsprozess

Dieses Projekt zeigt beispielhaft, wie KI-Werkzeuge wie ChatGPT oder Copilot die Kreativität fördern, Routineaufgaben automatisieren und gleichzeitig den Lernprozess beschleunigen können. Besonders im Bereich der Webentwicklung ermöglichen KI-basierte Assistenten eine niedrigere Einstiegshürde.

## 6. Fazit und Ausblick

Das Projekt beweist, wie mächtig die Kombination aus klassischer Webentwicklung und modernen KI-Werkzeugen sein kann. Besonders im Bildungsbereich oder bei Prototyping-Projekten lässt sich mit ChatGPT, Copilot und Cursor nicht nur Zeit sparen, sondern auch qualitativ hochwertiger Code erstellen – auch ohne jahrelange Erfahrung.

Die Zukunft der Softwareentwicklung wird nicht ausschliesslich von KIs übernommen werden – aber sie wird sich mehr und mehr in Richtung „**Co-Entwicklung mit KI**“ bewegen. Dieses Projekt war ein praktisches Beispiel dafür.

## 7. Anhang: Tools & Ressourcen

<b>Tool</b>	<b>Verwendungszweck</b>
<b>ChatGPT</b>	Inhaltserstellung, Debugging, Refactoring
<b>GitHub Copilot</b>	Autovervollständigung, strukturelle Hinweise
<b>Cursor Editor</b>	Inline-Feedback, KI-Vorschläge im Code-Editor
<b>HTML-Template</b>	Visuelle Darstellung von Artikeln

## 8. Designs

