

# Prototyping Projektdokumentation

Name: Milos Peric

E-Mail: pericmi1@students.zhaw.ch

URL der deployten Anwendung: <https://statify-me.netlify.app>

## 1. Einleitung

StatifyMe ist eine datengetriebene Webanwendung zur persönlichen Selbstvermessung. Die Plattform ermöglicht es Nutzerinnen und Nutzern, individuelle Metriken wie zum Beispiel tägliche Wasserzufuhr, Trainingszeiten, Lerneinheiten oder Schlafdauer selbst zu definieren und in Form sogenannter „Datapoints“ regelmässig zu erfassen. Ziel ist es, Gewohnheiten sichtbar zu machen und über Zeitverläufe ein besseres Verständnis für das eigene Verhalten und dessen Entwicklung zu gewinnen.

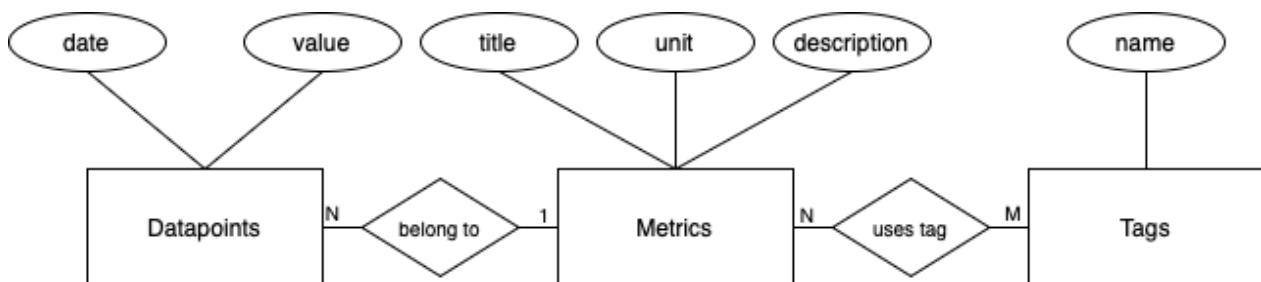
Das moderne und minimalistische User Interface wurde mit einem klaren Fokus auf Nutzerfreundlichkeit gestaltet. Die Eingabeoberflächen sind intuitiv bedienbar, das Layout reagiert flexibel auf verschiedene Bildschirmgrößen, sodass die Anwendung sowohl auf dem Smartphone als auch am Desktop optimal genutzt werden kann.

StatifyMe bietet nicht nur reine Datenerfassung, sondern auch eine automatische Auswertung der eingegebenen Werte. Für jede Metrik wird der Durchschnitt, das Maximum sowie das Minimum berechnet und übersichtlich dargestellt. Zusätzlich lassen sich die Daten über dynamische Liniendiagramme visualisieren, wodurch Trends auf einen Blick erkennbar werden.

Technisch basiert die Anwendung auf dem modernen JavaScript-Framework SvelteKit, das eine besonders performante und reaktive Nutzererfahrung ermöglicht. Die Daten werden in einer MongoDB-Datenbank gespeichert und über serverseitige Funktionen verwaltet. Zur grafischen Darstellung kommt die Bibliothek Chart.js zum Einsatz. Das Projekt wird über Netlify bereitgestellt, was eine einfache und performante Auslieferung der Anwendung im Web sicherstellt.

StatifyMe richtet sich an alle, die ihre Routinen analysieren, persönliche Ziele verfolgen oder einfach einen besseren Überblick über ihren Alltag gewinnen möchten – egal ob für Gesundheit, Produktivität oder Wohlbefinden.

## 2. Datenmodell



Das Datenmodell von StatifyMe basiert auf drei zentralen Entitäten: Metrics, Datapoints und Tags. Es wurde so gestaltet, dass es eine flexible, übersichtliche und leicht erweiterbare Struktur für die Erfassung und Auswertung persönlicher Metriken bietet.

Die Entität Metrics repräsentiert eine messbare Kategorie, beispielsweise „Gelernte Minuten“, „TikTok-Zeit“ oder „Getrunzene Tassen Kaffee“. Jede Metrik besitzt Attribute wie einen Titel (title), eine Beschreibung (description) und eine Einheit (unit), etwa Minuten, Stunden oder Tassen. Diese Metriken bilden die Grundlage für die Datenpunkte.

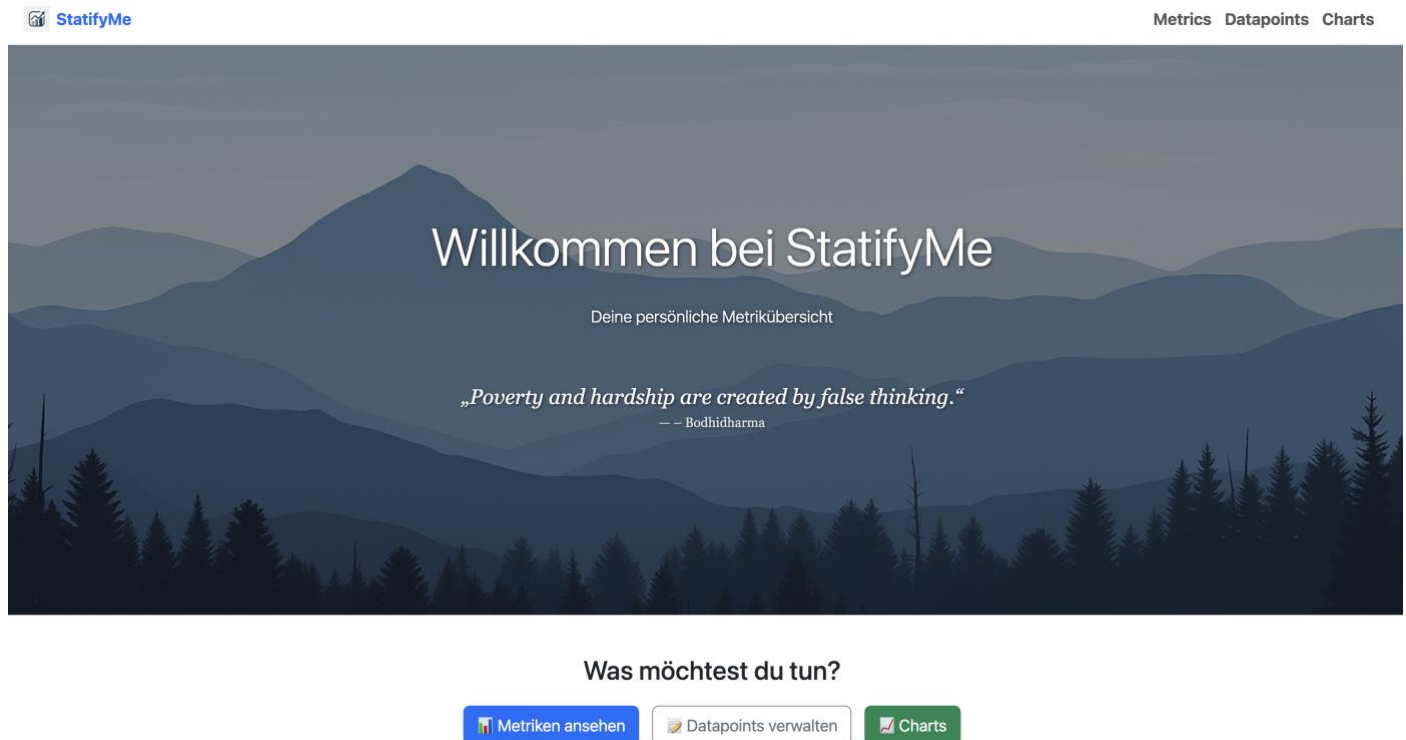
Datapoints sind konkrete Messwerte, die einem bestimmten Tag zugeordnet sind. Sie enthalten ein Datum (date) und einen numerischen Wert (value). Jeder Datapoint ist einer bestimmten Metrik zugeordnet, was einer 1:N-Beziehung entspricht – eine Metrik kann viele Datapoints besitzen, ein Datapoint gehört aber genau zu einer Metrik.

Die dritte Entität im Modell sind die Tags. Tags sind frei wählbare Kategorien wie „Gesundheit“, „Routine“ oder „Produktivität“, die zur besseren Strukturierung von Metriken verwendet werden. Eine Metrik kann mehrere Tags haben, und ein Tag kann gleichzeitig mehreren Metriken zugewiesen werden. Diese Beziehung ist vom Typ N:M (viele-zu-viele) und wird in der Datenbank durch ein Array von Tag-IDs in jedem Metric-Dokument abgebildet.

### 3. Beschreibung der Anwendung

#### 3.1. Startseite

Route: /



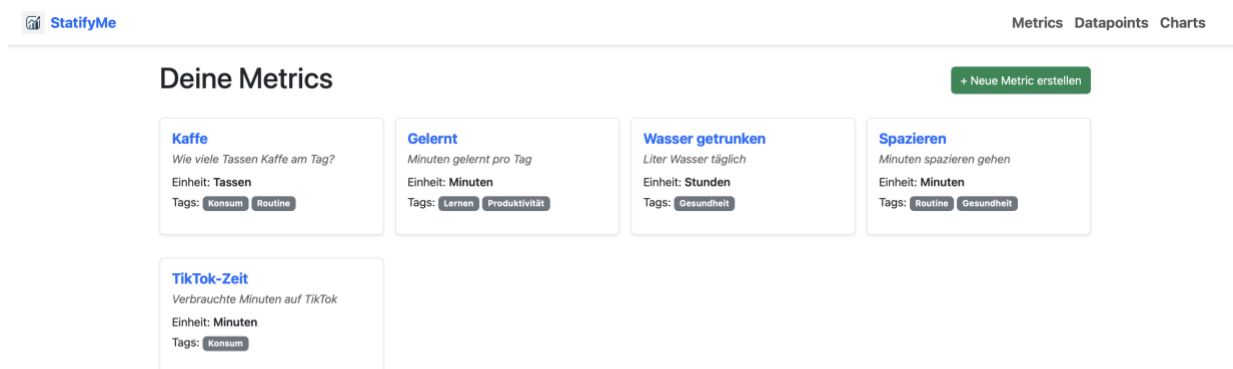
Die Startseite von StatifyMe dient als Einstiegspunkt und vermittelt sofort den Zweck der Anwendung. Ein Titel, ein motivierender Untertitel und das Logo schaffen einen professionellen Eindruck. Grosse Buttons führen direkt zu den Kernfunktionen: /metrics, /datapoints und /charts. Das Design ist modern, übersichtlich und mobiloptimiert, um Nutzer:innen eine intuitive und schnelle Orientierung zu ermöglichen.

Dateien:

- routes/+page.svelte

#### 3.2. Metricoverview

Route: /movies



Auf dieser Seite werden alle vorhandenen Metriken in Form von Karten dargestellt. Für die Darstellung der einzelnen Karten wird die Komponente MetricCard.svelte verwendet. Jede Metrik enthält einen Titel, eine Beschreibung, eine

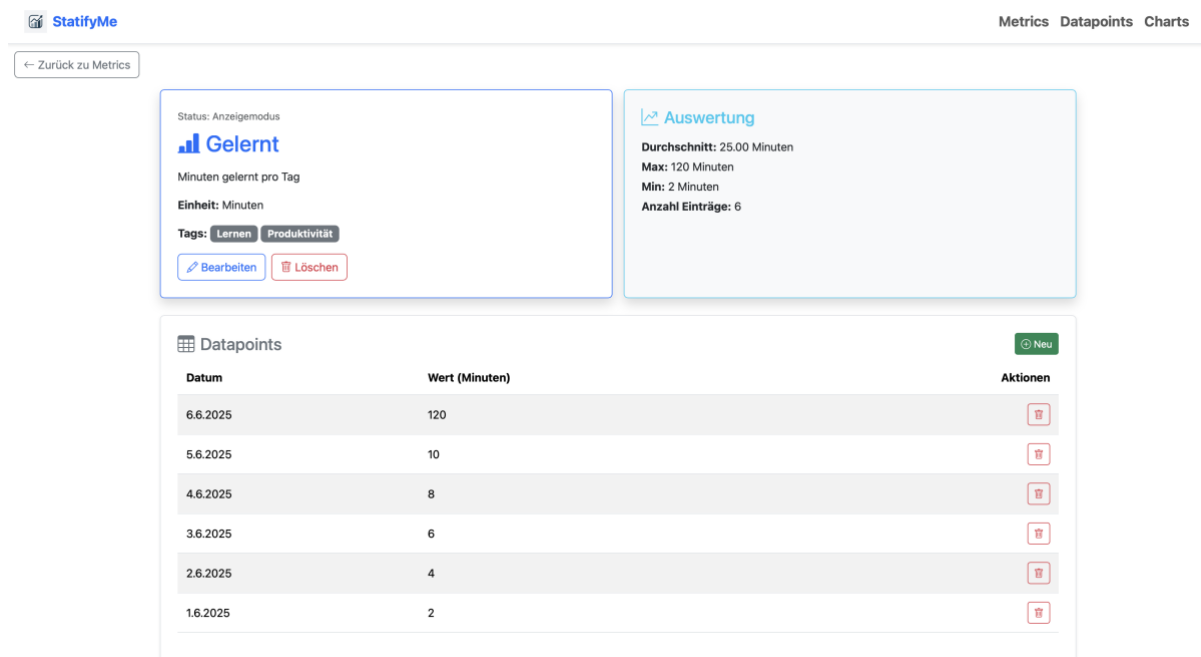
Einheit sowie zugehörige Tags. Über den Button „+ Neue Metric erstellen“ gelangt man zur Seite `/metrics/create`, auf der eine neue Metrik hinzugefügt werden kann. Beim Klick auf eine Metrikkarte wird man zur Detailansicht der jeweiligen Metrik weitergeleitet (`/metrics/[id]`).

Dateien:

- `lib/components/MetricCard.svelte`
- `routes/metrics/+page.svelte`
- `routes/metrics/+page.server.js`

### 3.3. Metric Detailpage

**Route:** `/metrics/[id]`



Auf dieser Seite wird eine einzelne Metrik mit all ihren Eigenschaften im Detail angezeigt. Dazu gehören Titel, Beschreibung, Einheit sowie die zugeordneten Tags.

Im unteren Bereich befindet sich eine Tabelle mit allen erfassten Datapoints zur ausgewählten Metrik. Jeder Datapoint besteht aus einem Datum und einem Wert (z. B. „2025-06-05 / 20 Minuten“). Die Datapoints ermöglichen es, den zeitlichen Verlauf einer Metrik nachzuvollziehen.

Zusätzlich stehen pro Metrik Statistiken zur Verfügung: Durchschnittswert, Minimum, Maximum und die Anzahl der Einträge. Neue Datapoints können direkt über ein eingblendetes Eingabeformular hinzugefügt werden, bestehende lassen sich über ein Papierkorb-Symbol gezielt löschen.


Die Seite bietet ausserdem die Möglichkeit, die Metrik selbst zu bearbeiten (Titel & Beschreibung) oder vollständig zu löschen.

Dateien:

- `routes/metrics/[id]/+page.svelte`
- `routes/metrics/[id]/+page.server.js`

### 3.4. Metric erstellen

**Route:** /metrics/create

 Metrics Datapoints Charts

## Neue Metric erstellen

**Titel**

**Beschreibung**

**Einheit**

**Tags**

Konsum Routine Lernen Gesundheit Produktivität

Erstellen

Auf dieser Seite kann der Benutzer eine neue Metrik anlegen. Dabei werden folgende Eingabefelder bereitgestellt:

- Titel: Bezeichnung der Metrik (z. B. „Gelernte Minuten“)
- Beschreibung: Optionales Textfeld zur genaueren Erläuterung der Metrik
- Einheit: Masseinheit der erfassten Werte (z. B. Minuten, Tassen, Schritte)
- Tags: Auswahl einer oder mehrerer Kategorien (z. B. „Gesundheit“, „Produktivität“) zur späteren Filterung

Die verfügbaren Tags werden beim Laden der Seite automatisch aus der Datenbank geladen und im Formular als Mehrfachauswahl angezeigt. Pflichtfelder sind „Titel“ und „Einheit“. Nach dem Absenden wird die neue Metrik gespeichert und der Benutzer zur Metrik-Übersichtsseite weitergeleitet.

Dateien:

- routes/metrics/create/+page.svelte
- → routes/metrics/create/+page.server.js

### 3.5. Datapoints

**Route:** /datapoints

#### Datapoints

Metric	Tag	Datum		
Alle	Alle	Alle		
Datum	Wert	Metric	Tags	Aktionen
6.6.2025	3 Stunden	Wasser getrunken	Gesundheit	
6.6.2025	19 Minuten	Spazieren	Gesundheit Routine	
6.6.2025	120 Minuten	Gelernt	Lernen Produktivität	
6.6.2025	210 Minuten	TikTok-Zeit	Konsum	
5.6.2025	30 Minuten	Pilates	Routine Gesundheit Produktivität	
5.6.2025	20 Minuten	Spazieren	Gesundheit Routine	
5.6.2025	15 Stunden	Wasser getrunken	Gesundheit	
5.6.2025	0 Tassen	Kaffe	Konsum Routine	
5.6.2025	10 Minuten	Gelernt	Lernen Produktivität	
4.6.2025	8 Minuten	Gelernt	Lernen Produktivität	
4.6.2025	12 Stunden	Wasser getrunken	Gesundheit	
4.6.2025	4 Tassen	Kaffe	Konsum Routine	
4.6.2025	20 Minuten	TikTok-Zeit	Konsum	

Auf dieser Seite werden alle vorhandenen Datapoints tabellarisch angezeigt. Ein Datapoint ist ein einzelner, zeitbezogener Messwert, der einer bestimmten Metrik zugeordnet ist (z. B. 15 Minuten „Gelernte Zeit“ am 05.06.2025).

Folgende Funktionen stehen zur Verfügung:

- Filterfunktion:
- Nach Metrik filtern: Dropdown-Auswahl einer Metrik zur Anzeige der zugehörigen Einträge.
- Nach Tag filtern: Ermöglicht die Anzeige aller Metriken, die mit einem bestimmten Tag (z. B. „Routine“) verbunden sind.
- Nach Datum filtern: Auswahl eines spezifischen Datums (via Datepicker), um alle Einträge dieses Tages anzuzeigen.
- Tabelle:
- Zeigt: Datum · Wert + Einheit · Metrik · zugehörige Tags
- Die Tags werden visuell als Badges dargestellt.
- Jeder Eintrag kann mit einem Button gelöscht werden.

Das Interface ist kompakt und nutzerfreundlich, die Anzeige aktualisiert sich dynamisch basierend auf den gewählten Filtern.

Dateien:

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

## Charts

Metric auswählen



Auf dieser Seite können Nutzer:innen ein Liniendiagramm generieren, das die Entwicklung einer bestimmten Metrik über die Zeit visualisiert. Die Charts basieren auf den zuvor eingegebenen Datapoints.

Funktionalität:

- Dropdown-Auswahl einer Metrik:

Die Benutzer:innen wählen eine Metrik (z. B. „Gewicht“ oder „Gelernte Stunden“) aus einer Liste aus.

- Diagramm-Erstellung:

Nach Auswahl wird automatisch ein interaktives Liniendiagramm mit Chart.js generiert:

- X-Achse: Datum der Einträge
- Y-Achse: Werte der Einträge inkl. Einheiten (z. B. „kg“)
- Glatte Linie mit dezentem Farbverlauf
- Responsive Design: Das Diagramm passt sich dynamisch an verschiedene Bildschirmgrößen an.

Technisches:

- Die Datapoints werden clientseitig gefiltert und sortiert.
- Es wird automatisch überprüft, ob ein vorheriges Diagramm existiert, das ggf. zerstört werden muss, bevor ein neues erzeugt wird.

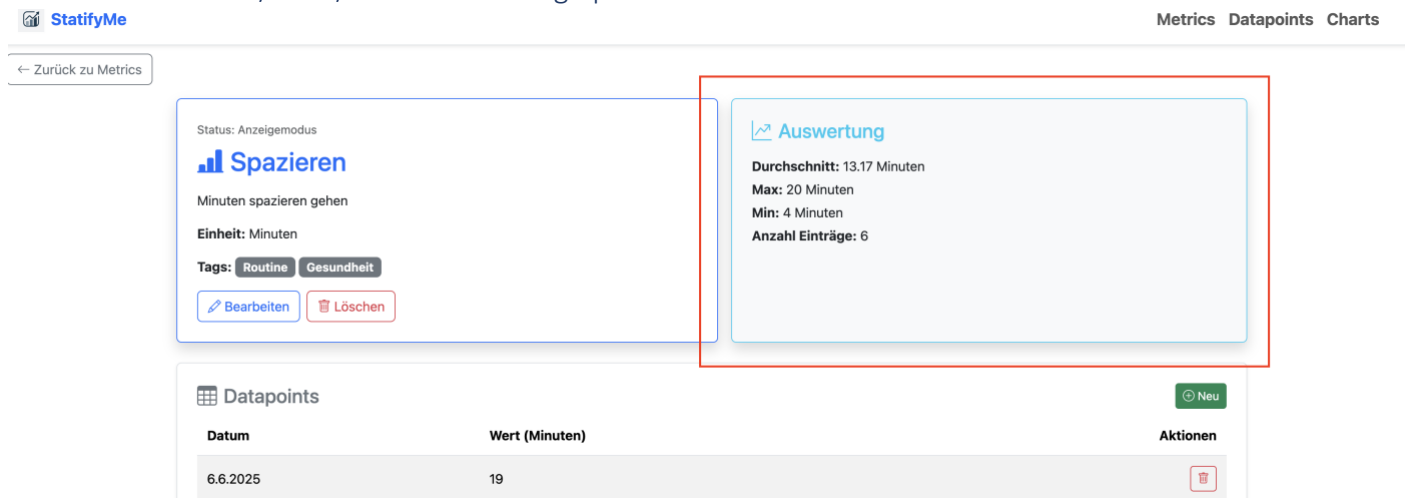
Dateien:

- routes/charts/+page.svelte

## 4. Erweiterungen

Im Folgenden werden Erweiterungen dokumentiert, die über die Grundanforderungen hinausgehen. Jede Erweiterung wird mit einem kurzen Überblick, dem Ort der Implementierung (Dateien, Funktionen) sowie ggf. mit einem Screenshot beschrieben.

### 4.1. Durchschnitts-/Max-/Min-Wert-Anzeige pro Metric



Auf der Detailseite einer einzelnen Metric (/metrics/[id]) wird eine automatische Auswertung der zugehörigen Datenpunkte vorgenommen. Dabei werden der Durchschnitt, das Maximum und das Minimum der Werte berechnet und in einer separaten Box angezeigt.

Diese Berechnung erfolgt dynamisch über das Frontend auf Basis der geladenen data.datapoints. Die Resultate werden jeweils als gerundete Zahlen in der Box "Auswertung" dargestellt.

Die Erweiterung unterstützt Nutzer:innen dabei, Trends schneller zu erkennen, ohne selbst Berechnungen vornehmen zu müssen.

Realisiert in:

- routes/metrics/[id]/+page.svelte

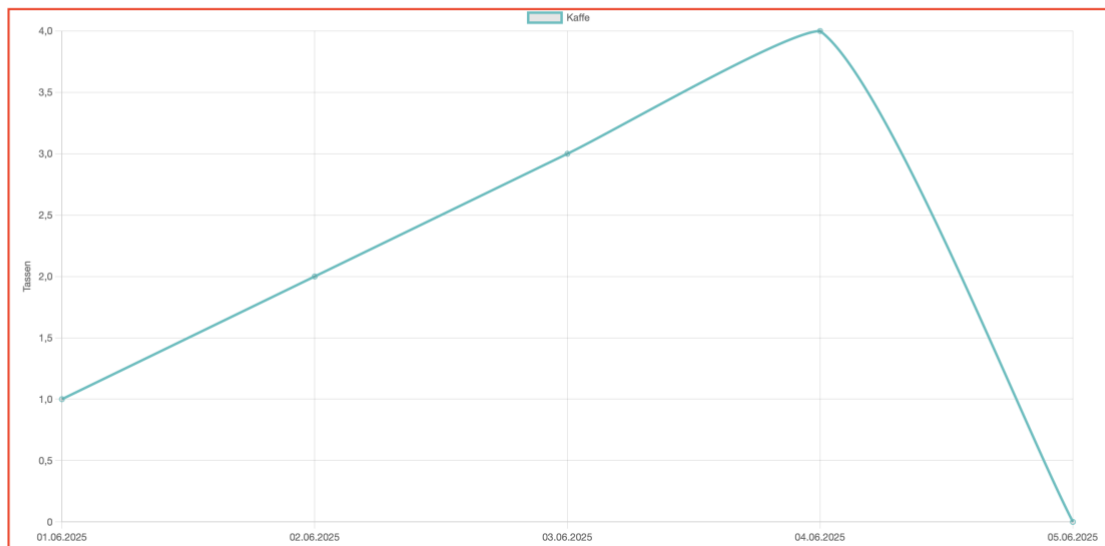
(Berechnung via \$:-Reactive-Statement und Darstellung in der Box mit Klasse .analysis-box)

## 4.2. Dynamische Diagramme für Metriken

### Charts

Metric auswählen

Kaffe



Unter der Route `/charts` bietet StatifyMe eine visuelle Auswertung der erfassten Metriken in Form eines interaktiven Liniendiagramms. Nutzer:innen wählen aus einem Dropdown-Menü eine Metrik aus. Daraufhin wird automatisch ein Diagramm generiert, das alle zugehörigen Datenpunkte chronologisch darstellt.

Die Darstellung erfolgt mithilfe der JavaScript-Bibliothek `Chart.js`, die in der Datei `+page.svelte` integriert ist. Sobald eine Metrik ausgewählt wird, filtert die Anwendung im Frontend alle zugehörigen Datenpunkte, sortiert sie nach Datum und übergibt die Werte an die Chart-Komponente. Bereits bestehende Diagramme werden vor dem Neuzeichnen gelöscht, um Speicherlecks und Überlagerungen zu vermeiden. Die X-Achse zeigt formatiert das Datum im Schweizer Format (`dd.mm.yyyy`), die Y-Achse trägt die zur Metrik gehörende Einheit.

Die notwendigen Daten (alle Metriken und Datenpunkte) werden serverseitig durch die `load()`-Funktion in `+page.server.js` geladen. Dadurch steht die gesamte Datenbasis direkt im Frontend zur Verfügung, was ein schnelles und interaktives Nutzererlebnis ermöglicht.

Dateien:

- `routes/charts/+page.svelte`
- `routes/charts/+page.server.js`
- `lib/server/db.js` (Datenbereitstellung durch `getAllMetrics()` und `getAllDatapoints()`)

Verwendete Bibliothek:

- `chart.js/auto`



### 4.3. Erweiterung: Tag-basierter Filter mittels MongoDB

#### Datapoints

Metric

Alle

Tag

Alle

Datum

Alle

Datum	Wert	Metric	Tags	Aktionen
6.6.2025	3 Stunden	Wasser getrunken	Gesundheit	
6.6.2025	19 Minuten	Spazieren	Gesundheit Routine	
6.6.2025	120 Minuten	Gelernt	Lernen Produktivität	
6.6.2025	210 Minuten	TikTok-Zeit	Konsum	
5.6.2025	30 Minuten	Pilates	Routine Gesundheit Produktivität	
5.6.2025	20 Minuten	Spazieren	Gesundheit Routine	
5.6.2025	15 Stunden	Wasser getrunken	Gesundheit	

Die Anwendung StatifyMe wurde um eine erweiterte Filterfunktion ergänzt, die es Nutzer:innen erlaubt, Datenpunkte basierend auf einem bestimmten Tag zu filtern. Diese Funktion wurde vollständig serverseitig realisiert und greift auf die Tag-Zuweisungen der Metriken zurück, um nur relevante Inhalte effizient darzustellen.

Funktionsweise:

Bei Auswahl eines Tags im Dropdown-Menü wird auf der Seite /datapoints ein Filter aktiviert. Dieser gleicht die gewählten Tags mit den tagIds der gespeicherten Metriken ab. Nur jene Datenpunkte werden angezeigt, deren zugeordnete Metrik den ausgewählten Tag beinhaltet. Die Filterung geschieht dynamisch auf dem Client, jedoch auf Basis vorab angereicherter Datenstrukturen, die beim Laden vom Server zur Verfügung gestellt werden.

Die Methode `getAllMetrics()` wurde dafür erweitert: Sie reichert jede Metrik direkt mit den zugehörigen Tag-Namen an (`tagNames`). Dadurch können die Tags bei der Anzeige direkt als Labels genutzt und gleichzeitig für die Filterlogik ausgewertet werden.

Ziel der Erweiterung:

- Verbesserung der Benutzerfreundlichkeit bei umfangreichen Datensätzen
- Gezielte Datenanalyse durch Tag-basierte Einschränkung
- Effizienzsteigerung durch serverseitige Datenvorbereitung

Betroffene Dateien:

- `routes/datapoints/+page.svelte` → Dropdown & Filterlogik
- `routes/datapoints/+page.server.js` → Anreicherung der metrics mit `tagNames`
- `lib/server/db.js` → `getAllMetrics()` nutzt `getAllTags()` für Namensauflösung

Diese Erweiterung bietet einen spürbaren Mehrwert bei der Analyse und Verwaltung von Metriken und zeigt, wie flexibel und erweiterbar die Anwendung mit MongoDB als Backend strukturiert wurde.

#### 4.4. Motivationszitat über externe API



Zur Aufwertung der Startseite wurde eine zusätzliche Funktionalität implementiert, die ein zufälliges, motivierendes Zitat von einer öffentlichen Schnittstelle einbindet. Hierzu wird die API von Quotable.io verwendet, die ein JSON-Objekt mit einem Zitat und dem entsprechenden Autor liefert.

Beim Laden der Startseite wird im Server-Load-Handler in der Datei `src/routes/+page.server.js` ein GET-Request an die externe API gestellt. Das zurückgelieferte Zitat wird anschliessend als `data.quote` und `data.author` an die Seite übergeben und im Frontend (`+page.svelte`) eingebunden. Dort erscheint es prominent in einem stilisierten Blockzitat unter dem Titel „Willkommen bei StatifyMe“.

Dadurch wird die Begrüssung dynamischer, persönlicher und bietet Nutzer:innen einen positiven Einstieg in die Anwendung.

Die Integration erfolgt vollautomatisch beim Laden der Seite. Sollte die API nicht erreichbar sein, wird ein Fallback definiert, sodass die Anwendung weiterhin funktionsfähig bleibt.