

Hochschule Bremerhaven University of Applied Sciences

Software Engineering I

Durchgeführt auf der:

Hochschule Bremerhaven – University of Applied Sciences

Fachbereich: 2

Studiengang: Informatik/Wirtschaftsinformatik

Team 01

Wetterdaten auslesen

Teammitglieder:

Bayram Selimi – 40604

Fabian Tober – 40607

Ben Schlickowey – 40869

Julian Slavik – 41194

Inhaltsverzeichnis

Einleitung	2
Um was geht es?	2
Wer hat das Projekt bearbeitet	2
Unser Projekt	3-5
Projektbericht	3
Zielsetzung.....	3
Methoden und Technologien	3-4
Umsetzung	4-5
Ergebnis	5
Fazit.....	5
UML - Diagramme	5-9
Erläuterung typischer Eigenschaften.....	5
Anwendungsfalldiagramm.....	6
Aktivitätsdiagramm	6
Klassendiagramm.....	7
Prototyp/Wireframe	7-8
Codeschnipsel Beispiel.....	9
Zusammenfassung	9-10
Quellen	10

Einleitung

Um was geht es? - Projekt: Ausgabe von Wetterdaten

In einer Welt, die immer stärker von Daten und ihrer Analyse abhängig ist, spielt das Verständnis und die Nutzung von Wetterdaten eine entscheidende Rolle. Unser Projekt konzentriert sich auf genau diesen Aspekt - die Ausgabe von Wetterdaten.

Das Hauptziel dieses Projekts ist es, eine effiziente und benutzerfreundliche Plattform zu entwickeln, die in der Lage ist, die relevantesten Wetterdaten darzustellen. Diese Daten umfassen Parameter, wie Temperatur, Luftfeuchtigkeit und Wetterlage wie Nieselregen. Wichtig dabei ist das diese Datenausgabe auch eine sieben Tage Vorhersage einschließt.

Die Bedeutung dieses Projekts liegt in seiner Anwendbarkeit. Von der Landwirtschaft über die Stadtplanung und Eventplanung bis hin zur eigenen Geburtstagsfeier - genaue und zeitnahe Wetterdaten können in fast jedem Bereich einen erheblichen Einfluss haben.

Wer hat das Projekt bearbeitet? - Teammitglieder:

Unser Projektteam besteht aus vier Mitgliedern, die in drei Positionen arbeiten: Frontend, Backend und Organisation.

Im Bereich Frontend ist Ben Schlickowey. Sie sind verantwortlich für die Gestaltung der Benutzeroberfläche und die Gewährleistung einer reibungslosen Benutzererfahrung.

Im Backend arbeiten Bayram Selimi und Julian Slavik. Die beiden kümmern sich hauptsächlich um den Code, beispielsweise die Skripte. Ihre Arbeit stellt sicher, dass die Wetterdaten korrekt an das Frontend übermittelt werden.

Die Organisation des Projekts liegt in den Händen von Fabian Tober. Er koordiniert das Team und kümmert sich um organisatorische Dinge wie die UML-Diagramme.

Alle Positionen sind **nicht** fest, sodass jeder in jedem Bereich sich auskennt. Jedoch finden wir das es wichtig ist das sich immer irgendjemand ganz besonders auskennt wie zum Beispiel im Backend Bereich wo es schnell schwierig werden kann den Code zu überblicken, wenn man den Code nicht selbst geschrieben hat. Darum haben wir eine grundlegende Teameinteilung gemacht.

Unser Projekt

Projektbericht: Entwicklung einer leistungsarmen und benutzerfreundlichen Wetterdaten-Webseite.

Das Ziel dieses Projektes war die Entwicklung einer benutzerfreundlichen Webseite zur Anzeige von Wetterdaten, die auf einer frei verfügbaren API basiert. Die Webseite sollte leistungsarm sein, sich auf wesentliche Informationen konzentrieren und eine intuitive Benutzererfahrung bieten.

Zielsetzung:

Unser Hauptziel bestand darin, eine übersichtliche und leicht zugängliche Plattform zu schaffen, die es den Benutzern ermöglicht, aktuelle Wetterdaten abzurufen und nach ihren Bedürfnissen zu filtern. Dabei lag ein besonderer Fokus auf der Leistungsoptimierung, um die Ressourcennutzung zu minimieren und die Ladezeiten der Seite zu verbessern durch zum Beispiel weglassen von Videos und Bildern, die zu viele Daten verbrauchen. „Über 300 Mio. Tonnen CO2 jährlich fallen für die Nutzung von Videoinhalten an.“ (The Shift Project 2019).

Verwendete Methoden und Technologien:

Im Rahmen der Entwicklung unserer eigenen Webseite zur Abrufung von Wetterdaten haben wir verschiedene Methoden und Technologien eingesetzt, um ein optimales Nutzererlebnis zu gewährleisten.

Zunächst haben wir eine frei verfügbare API integriert, die es uns ermöglicht, jederzeit auf aktuelle Wetterdaten zuzugreifen und sie auf unserer Plattform anzuzeigen. Dieser Schritt war entscheidend, um unseren Nutzern stets zuverlässige und zeitnahe Informationen zur Verfügung zu stellen.

Für die Entwicklung der Benutzeroberfläche haben wir auf bewährte Webtechnologien wie HTML und CSS gesetzt. Durch diese Wahl konnten wir eine benutzerfreundliche Webseite gestalten, die es unseren Besuchern ermöglicht, die Wetterdaten dynamisch und übersichtlich zu erkunden.

Ein wesentlicher Schwerpunkt lag auf der Leistungsoptimierung unserer Webseite. Hierbei haben wir den Code sorgfältig optimiert, um die Ladezeiten zu minimieren und die Gesamtleistung zu verbessern. Dies umfasste die Reduzierung von unnötigen Abfragen sowie die effiziente Nutzung von Ressourcen, um ein reibungsloses Nutzererlebnis sicherzustellen. „Über die letzten fünf Jahre ist die Größe einer Website um 40 % gestiegen. Dabei ließen sich viele digitale Produkte schlanker gestalten, wenn beispielsweise Assets wie Bilder passend skaliert, Videos niemals per Autostart beginnen und Schriftarten einfach gehalten würden.“ (Winkler et al. 8)

Besonderes Augenmerk haben wir zudem auf die Benutzerfreundlichkeit gelegt. Durch eine klare Darstellung der Wetterdaten und die Implementierung einfacher Filtermöglichkeiten haben wir eine intuitive Navigation auf der Webseite ermöglicht. Dies trägt dazu bei, dass unsere Nutzer die Informationen schnell finden und leicht zugänglich sind.

Insgesamt war die Entwicklung unserer Wetterdaten-Webseite von einem ganzheitlichen Ansatz geprägt, der darauf abzielte, sowohl technische Leistung als auch Benutzerfreundlichkeit zu maximieren. Durch die sorgfältige Integration verschiedener Elemente konnten wir eine hochwertige Plattform schaffen, die den Anforderungen unserer Nutzer gerecht wird.

Umsetzung:

Die Umsetzung unserer Wetterdaten-Webseite erfolgte in mehreren klar definierten Phasen:

1. Anforderungsanalyse:

Zu Beginn des Projekts führten wir eine umfassende Anforderungsanalyse durch. Dabei identifizierten wir die Kernfunktionen, die unsere Webseite bieten sollte, und legten fest, welche Anforderungen wir an die Darstellung und Aktualisierung der Wetterdaten hatten. Diese Phase war entscheidend, um eine klare Vision für das Projekt zu entwickeln und die Grundlage für die weitere Entwicklung zu legen.

2. API-Integration:

Im Anschluss an die Anforderungsanalyse integrierten wir die ausgewählte API in unsere Webseite. Dabei konfigurierten wir sie entsprechend unseren spezifischen Anforderungen, um einen reibungslosen Abruf und eine zuverlässige Anzeige der aktuellen Wetterdaten sicherzustellen. Die korrekte Integration der API bildete das Rückgrat unserer Webseite und ermöglichte es uns, kontinuierlich aktualisierte Informationen bereitzustellen.

3. Webdesign und Implementierung:

Die Gestaltung und Implementierung der Webseite erfolgte parallel zur API-Integration. Wir entwickelten ein ansprechendes und benutzerfreundliches Design, das es unseren Besuchern ermöglichte, die Wetterdaten intuitiv zu erfassen und zu nutzen. Dabei wurden die Funktionen zur Anzeige der Wetterdaten sowie die Implementierung von Filteroptionen sorgfältig umgesetzt, um eine nahtlose Nutzererfahrung zu gewährleisten.

Durch die enge Verknüpfung von Design und Funktionalität schufen wir eine Webseite, die nicht nur ästhetisch ansprechend ist, sondern auch die Bedürfnisse unserer Nutzer effektiv erfüllt.

In jeder Phase der Umsetzung legten wir großen Wert auf eine gründliche Planung und sorgfältige Ausführung, um sicherzustellen, dass unsere Wetterdaten-Webseite die höchsten Standards in Bezug auf Leistung, Benutzerfreundlichkeit und Funktionalität erfüllt.

Ergebnis:

Das Ergebnis ist eine leistungsarme und benutzerfreundliche Webseite zur Anzeige von Wetterdaten. Die Seite bietet eine klare Darstellung der Informationen und ermöglicht es den Benutzern, die Daten nach Temperatur, Luftfeuchtigkeit und Wetterbeschreibung zu filtern. Durch die optimierte Leistung lädt die Seite schnell und bietet eine reibungslose Nutzererfahrung. „Clean Code bei der Website-Erstellung reduziert ebenfalls den Energiebedarf – je einfacher der Code geschrieben ist, desto weniger Rechenleistung wird benötigt. Unnötige Skripts sollten entfernt werden.“ (Winkler et al. 8)

Fazit:

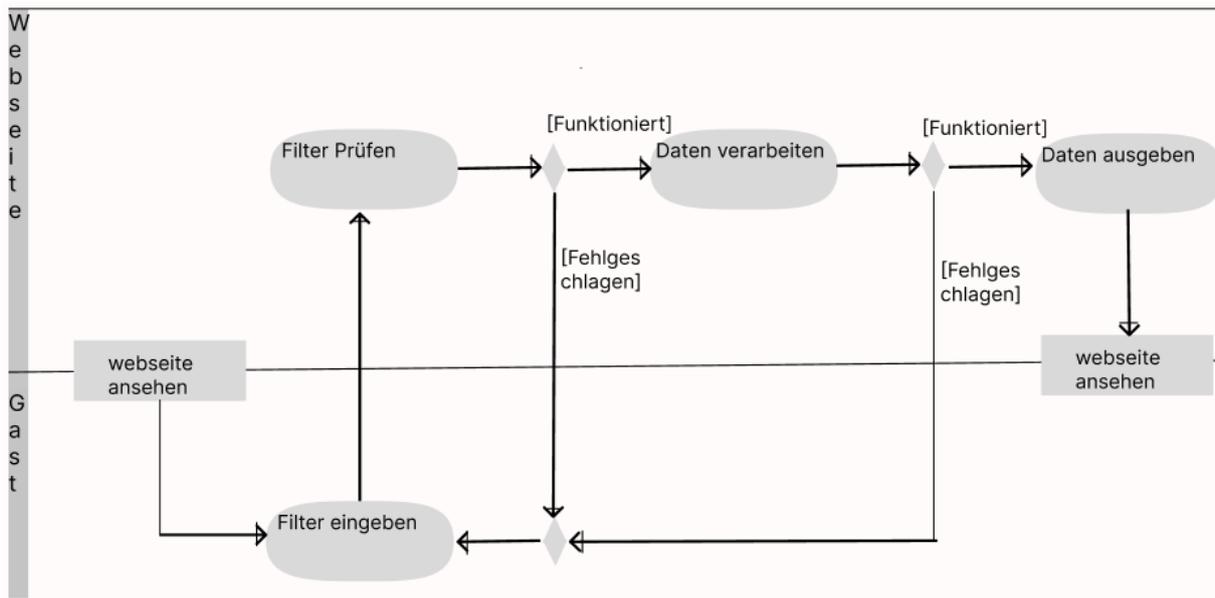
Die Entwicklung dieser Webseite war erfolgreich und hat unsere Ziele erreicht. Die Kombination aus Leistungsarmut und Benutzerfreundlichkeit macht die Webseite zu einem nützlichen Werkzeug für diejenigen, die schnell und einfach auf aktuelle Wetterdaten zugreifen möchten. „Digitalisierung benötigt Energie und verursacht somit auch schädliche Emissionen. Höhere Rechenleistungen und Internetgeschwindigkeiten verleiten Software-Entwickler und Webdesigner sogar dazu, immer strom-hungrigere Websites zu erstellen.“ (vgl. Gröger et al., 2018).

UML-Diagramme

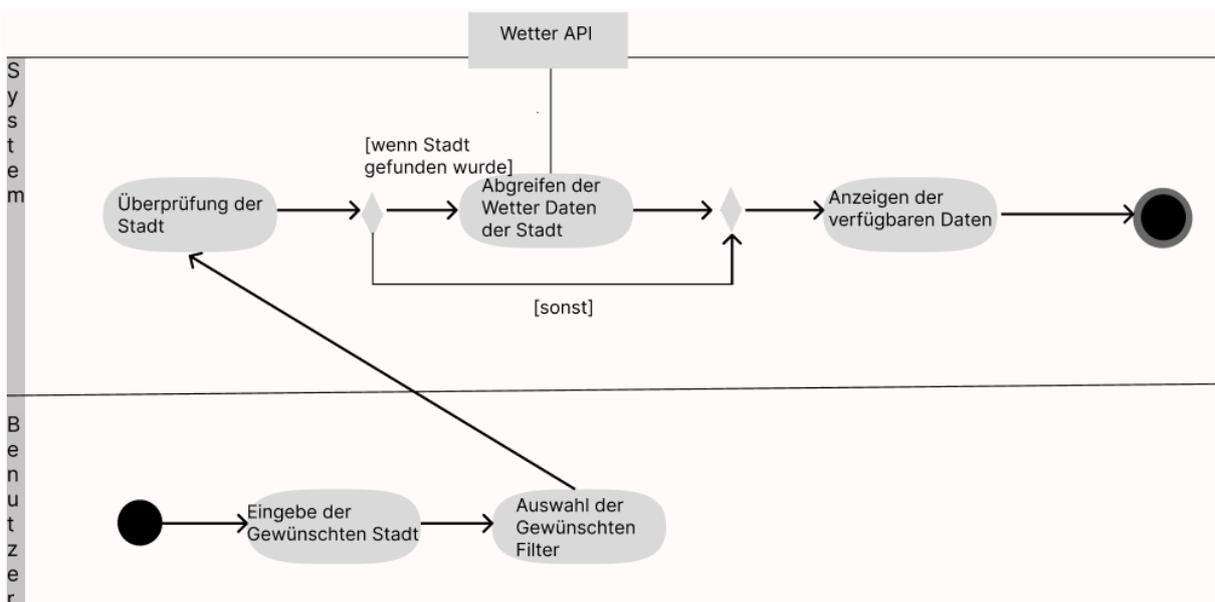
Erläuterung typischer Eigenschaften:

Das beschriebene Wetterdatenprojekt basiert auf einer benutzerfreundlichen Webanwendung, die über eine API-Wetterdaten abrufen und anzeigen. Die Anwendung ermöglicht es den Benutzern, eine Stadt auszuwählen und drei Filter ("Temperatur", "Luftfeuchtigkeit", "Wetterbeschreibung") anzuwenden, um die gewünschten Wetterdaten zu erhalten. Diese Daten werden für den aktuellen Tag sowie für die nächsten 7 Tage im Voraus angezeigt. Die Anwendung ist umweltfreundlich konzipiert und vermeidet große Dateien oder Bilder, um den CO₂-Ausstoß zu minimieren. Es wird ein responsives Design verwendet, um auf verschiedenen Geräten gut lesbar und nutzbar zu sein, und es wird die Datenübertragung optimiert, um den Energieverbrauch während der Nutzung zu minimieren.

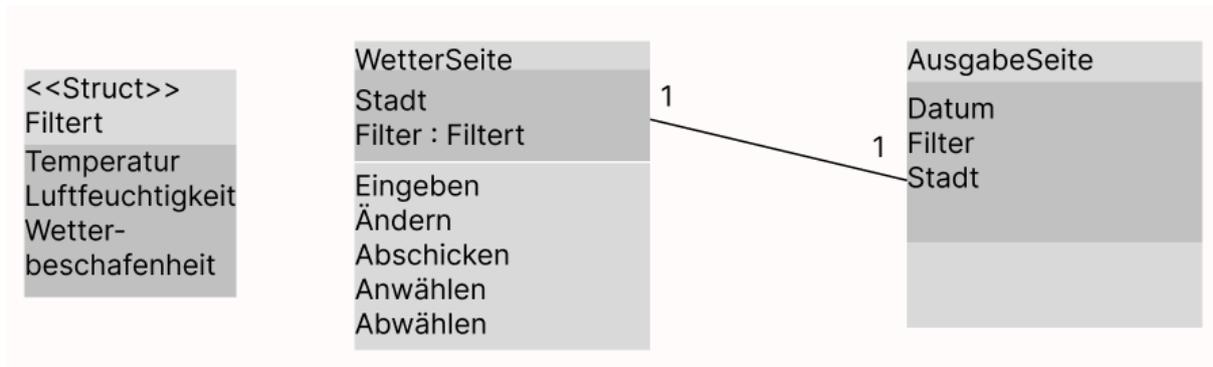
Anwendungsfalldiagramm:



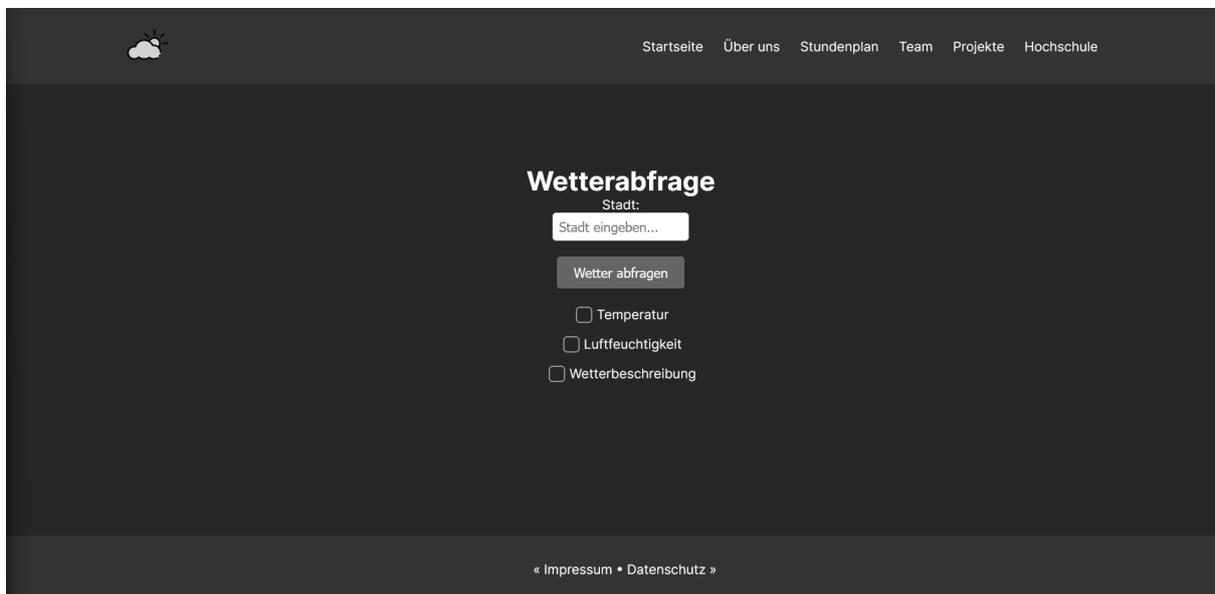
Aktivitätsdiagramm:



Klassendiagramm:



Prototyp/Wireframe:



Der Nutzer öffnet die Wetterdaten-anwendung und wird von einem übersichtlichen Layout begrüßt. Oben auf der Seite befindet sich ein Menü, das zusätzliche nützliche Funktionen bietet, aber für die unmittelbare Wetterabfrage nicht relevant ist. Durch einen Klick auf das Menü kann der Benutzer auf diese zusätzlichen Informationen wie beispielsweise Informationen über unser Team zugreifen. In der Mitte des Bildschirms befindet sich ein deutlich sichtbares Eingabefeld, in das der Benutzer den Namen seiner Stadt eingeben kann. Darunter werden die drei Filter "Temperatur", "Luftfeuchtigkeit" und "Wetterbeschreibung" angezeigt. Nachdem der Benutzer seine Stadt eingegeben und auf die Schaltfläche "Wetter abfragen" geklickt hat, lädt die Webseite die Daten über die API und fragt die Wetterdaten ab.

Aktuelle Wetterdaten in Bremerhaven

Temperatur: 6.1 Grad Celsius
Luftfeuchtigkeit: 86%
Wetterbeschreibung: Nebel

7-Tage Vorhersage für Bremerhaven

Wednesday 28.02.2024 - Temperatur: 4.5 Grad Celsius, Luftfeuchtigkeit: 86%, Wetterbeschreibung: Bewölkt
Thursday 29.02.2024 - Temperatur: 7.4 Grad Celsius, Luftfeuchtigkeit: 75%, Wetterbeschreibung: Leichter Schauerregen
Friday 01.03.2024 - Temperatur: 8 Grad Celsius, Luftfeuchtigkeit: 74%, Wetterbeschreibung: Bewölkt
Saturday 02.03.2024 - Temperatur: 8.7 Grad Celsius, Luftfeuchtigkeit: 70%, Wetterbeschreibung: Aufgelockert Bewölkt
Sunday 03.03.2024 - Temperatur: 8.7 Grad Celsius, Luftfeuchtigkeit: 75%, Wetterbeschreibung: Aufgelockert Bewölkt
Monday 04.03.2024 - Temperatur: 7.8 Grad Celsius, Luftfeuchtigkeit: 82%, Wetterbeschreibung: Bewölkt
Tuesday 05.03.2024 - Temperatur: 7.2 Grad Celsius, Luftfeuchtigkeit: 82%, Wetterbeschreibung: Bewölkt

-- Zurück zur Stadt/Filter Auswahl ==

Anschließend werden die Daten auf dem Bildschirm angezeigt. Wenn der Benutzer alle spezifischen Filter ausgewählt hat, dann werden alle verfügbaren Daten für die eingegebene Stadt angezeigt. Die angezeigten Daten beginnen mit den Informationen für den aktuellen Tag und werden dann für die nächsten sieben Tage untereinander aufgelistet. Dies ermöglicht dem Benutzer einen schnellen Überblick über die Wettervorhersage für die kommende Woche. Außerdem ist mittig unten ein Zurück Button mit dem Text „- Zurück zur Stadt/Filter Auswahl =“.

Aktuelle Wetterdaten in Bremerhaven

Temperatur: 6.1 Grad Celsius

7-Tage Vorhersage für Bremerhaven

Wednesday 28.02.2024 - Temperatur: 4.5 Grad Celsius,
Thursday 29.02.2024 - Temperatur: 7.4 Grad Celsius,
Friday 01.03.2024 - Temperatur: 8 Grad Celsius,
Saturday 02.03.2024 - Temperatur: 8.7 Grad Celsius,
Sunday 03.03.2024 - Temperatur: 8.7 Grad Celsius,
Monday 04.03.2024 - Temperatur: 7.8 Grad Celsius,
Tuesday 05.03.2024 - Temperatur: 7.2 Grad Celsius,

-- Zurück zur Stadt/Filter Auswahl ==

Der Benutzer kann bei Bedarf durch Klicken auf die Filteroptionen die angezeigten Daten weiter eingrenzen, um spezifische Details wie Temperatur, Luftfeuchtigkeit und/oder Wetterbeschreibung für die ausgewählte Stadt zu erhalten.

Codeschnipsel Beispiel:

```
1 # Die folgende Zeile zeigt den Titel "Aktuelle Wetterdaten in [Stadt]" an, wobei der Stadtwert aus der Variablen $city extrahiert wird.
2 echo "<h2>Aktuelle Wetterdaten in $(echo "$city" | sed 's/city=/' | sed 's/&.*//')</h2>"
3
4 # Anzeigen der ausgewählten Wetterinformationen
5
6 # Ueberpruefung, ob die Variable $show_temp auf 1 gesetzt ist (wenn ja, dann wird die Temperatur angezeigt).
7 if [ $show_temp -eq 1 ]; then
8     # Extrahieren der aktuellen Temperatur aus der JSON-Antwort und Speichern in der Variable $temperature_current_celsius
9     temperature_current_celsius=$(echo "$WEATHERBIT_RESPONSE_CURRENT" | jq -r '.data[0].temp')
10    # Anzeigen der Temperatur in Grad Celsius
11    echo "<p>Temperatur: $temperature_current_celsius Grad Celsius</p>"
12 fi
13
14 # Ueberpruefung, ob die Variable $show_humidity auf 1 gesetzt ist (wenn ja, dann wird die Luftfeuchtigkeit angezeigt).
15 if [ $show_humidity -eq 1 ]; then
16    # Extrahieren der aktuellen Luftfeuchtigkeit aus der JSON-Antwort und Speichern in der Variable $humidity_current
17    humidity_current=$(echo "$WEATHERBIT_RESPONSE_CURRENT" | jq -r '.data[0].rh')
18    # Anzeigen der Luftfeuchtigkeit in Prozent
19    echo "<p>Luftfeuchtigkeit: $humidity_current%</p>"
20 fi
21
22 # Ueberpruefung, ob die Variable $show_description auf 1 gesetzt ist (wenn ja, dann wird die Wetterbeschreibung angezeigt).
23 if [ $show_description -eq 1 ]; then
24    # Extrahieren der aktuellen Wetterbeschreibung aus der JSON-Antwort und Speichern in der Variable $description_current
25    description_current=$(echo "$WEATHERBIT_RESPONSE_CURRENT" | jq -r '.data[0].weather.description')
26    # Anzeigen der Wetterbeschreibung
27    echo "<p>Wetterbeschreibung: $description_current</p>"
28 fi
```

Kommentare im relevanten Codeschnipsel zur Erklärung der einzelnen Abschnitte. Codeschnipsel aus „weather_script.sh“.

Der gegebene Codeausschnitt extrahiert aktuelle Wetterdaten für eine bestimmte Stadt aus einer JSON-Antwort und zeigt sie auf einer Webseite an. Die Wetterdaten umfassen Temperatur, Luftfeuchtigkeit und Wetterbeschreibung. Die Stadt wird aus einer Variablen namens \$city extrahiert. Je nach den Einstellungen der Variablen \$show_temp, \$show_humidity und \$show_description werden die entsprechenden Wetterinformationen angezeigt.

Zusammenfassung

Das Projekt konzentrierte sich auf die Entwicklung einer benutzerfreundlichen Webseite zur Ausgabe von Wetterdaten. In einer Welt, die zunehmend von Daten abhängig ist, spielen genaue Wetterdaten eine entscheidende Rolle in verschiedenen Bereichen wie Landwirtschaft, Stadtplanung und Eventplanung. Das Ziel war es, eine Plattform zu schaffen, die es den Benutzern ermöglicht, relevante Wetterdaten wie Temperatur, Luftfeuchtigkeit und Wetterlage schnell und einfach abzurufen.

Das Team bestand aus sechs Mitgliedern, die in den Bereichen Frontend, Backend und Organisation tätig waren. Die Entwicklung erfolgte in mehreren Phasen, von der Anforderungsanalyse bis zur Implementierung der Webseite. Dabei wurde HTML, CSS und eine freie API verwendet, um eine benutzerfreundliche Benutzeroberfläche zu gestalten und die Wetterdaten dynamisch zu präsentieren.

Besonderes Augenmerk lag auf der Integration einer frei verfügbaren API sowie auf der Leistungsoptimierung der Webseite, um eine schnelle Ladezeit und eine reibungslose Benutzererfahrung zu gewährleisten. Das Ergebnis war eine leistungsarme und benutzerfreundliche Webseite, die es den Benutzern ermöglichte, die aktuellen Wetterdaten schnell und einfach abzurufen und nach ihren Bedürfnissen zu filtern.

Insgesamt war das Projekt erfolgreich in der Entwicklung einer Plattform zur Ausgabe von Wetterdaten. Die Kombination aus Leistungsarmut und Benutzerfreundlichkeit macht die Webseite zu einem nützlichen Werkzeug in verschiedenen Bereichen. Die Erkenntnisse aus diesem Projekt können in zukünftigen Projekten zur Entwicklung ähnlicher Anwendungen genutzt werden.

Quellen

The Shift Project (2019) "Climate crisis: the unsustainable use of Online video": our new REPORT on the environmental IMPACT of ICT.

<https://theshiftproject.org/en/article/unsustainable-use-online-video/>

Zugegriffen: 23. Febr. 2024

Winkler, Stefan, Jochen Günther, and Roland Pfennig. "Nachhaltige Digitalisierung oder Nachhaltigkeit durch Digitalisierung?." *HMD Praxis der Wirtschaftsinformatik* (2023): 1-22.

Nachhaltige Digitalisierung oder Nachhaltigkeit durch Digitalisierung? 835 Gröger J, Köhler A, Naumann S, Filler A, Guldner A, Kern E, Hilty L, Maksimov Y (2018) Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik. Umweltbundesamt. https://www.umweltbundesamt.de/sites/default/files/medien/1410/publikationen/2018-12-12_texte_105-2018_ressourceneffiziente-software_0.pdf

Zugegriffen: 23. Febr. 2024

Aufteilung:

- Unser Projekt -> 1. Anforderungsanalyse = Bayram & Julian
- 2. API-Integration -> Erläuterung typischer Eigenschaften + Codeschnipsel Beispiel = Ben & Fabian
- Prototyp/Wireframe = Alle (Aufteilung zwischen UML-Diagrammen und Text.)