

|  |
| --- |
| **Dokumentation zum Webengineering Projekt** |

des Studiengangs Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

**Jona Krumrein**

Klicken Sie hier, um ein Datum einzugeben.

|  |  |
| --- | --- |
| Abgabedatum | Klicken Sie hier, um ein Datum einzugeben.07.12.2020 |
| Matrikelnummern | 8366074 |
|  |  |
|  |  |
| Wählen Sie ein Element aus. | Klicken Sie hier, um Gutachter/in einzugeben. |

**Inhaltsverzeichnis:**

[1. Implementation der Wetter-Seite 2](#_Toc58168969)

[1.1 Design 2](#_Toc58168970)

[1.2 Umsetzung 2](#_Toc58168971)

[2. Implementation der Aktien-Seite 3](#_Toc58168972)

[2.1 Design 3](#_Toc58168973)

[2.2 Umsetzung 3](#_Toc58168974)

[3. Implementation der Wikipedia-Funktion 4](#_Toc58168975)

[3.1 Design 4](#_Toc58168976)

[3.2 Umsetzung 4](#_Toc58168977)

[3.3 Probleme 4](#_Toc58168978)

[4. Erfahrungsbericht 5](#_Toc58168979)

|  |  |
| --- | --- |
|  |  |
|  |  |

Einleitung

Im Folgenden werden die einzelnen Funktionen der im Rahmen des Webengineering Projekts erstellten Website erläutert und dokumentiert. Folgende Elemente wurden zur Erreichung der vorgegebenen Punktzahl implementiert:

1. Anzeigen von Aktien Verlaufskurven
2. Anzeige von Wetter Daten
3. Vorlesen von Wikipedia Beschreibung zu gewünschten Begriffen (Client-seitig)
4. Eigenes CSS

Zusätzlich sind ein Installationsguide und Flow Diagramme beigefügt, um die Übersicht und Verwendung des Projekts zu verdeutlichen.

# Implementation der Wetter-Seite

Die Spezifikation der Wetterseite war es, mittels einer API Wetterdaten auszulesen und diese ansprechend zu gestalten. Dabei sollten das jetzige Wetter und eine Wettervorhersage für die nächsten fünf Tage angezeigt werden. In meiner Implementation wurde die Vorhersage auf vier Tage gekürzt, da es aufgrund der Werte der API nicht möglich ist alle sechs Tage zu einer späten Uhrzeit anzuzeigen.

## Design

Als Design für die Wetterseite überlegte ich mir zwei schwarze Banner, welche sich beim Scrollen über den fixen Hintergrund bewegen. Im oberen Banner sollte die Stadt, das Land sowie das aktuelle Datum angezeigt werden. Außerdem darunter eine Kurzbeschreibung des Wetters, sowie die Temperatur in Celsius und das passende Icon um das Ganze zu verdeutlichen. Daneben sollte es mittels eines Eingabefelds ermöglicht werden die Stadt zu ändern, um andere Wetterdaten zu erhalten. Im unteren Banner sollte dann die Wettervorhersage stehen, welche den jeweiligen Tag, eine Beschreibung des Wetters, die Temperatur und ebenfalls das passende Icon anzeigt. Jeder Tag sollte in einer getrennten Box dargestellt werden um eine Abgrenzung von den anderen Tagen zu erzeugen. Zum Schluss sollte mit Hilfe von GIF Dateien das Wetter im Hintergrund durch zum Beispiel Regen, etc. verdeutlicht werden.

## Umsetzung

Umgesetzt wurde das ganz mit Hilfe der Openweathermap API, welche die oben genannten Daten als JSON Datei zur Verfügung stellt. Dabei wird beim Laden der Webseite das Wetter von Stuttgart bzw. bei der Eingabe einer bestimmten Stadt die Daten der API heruntergeladen. Danach werden die einzelnen Werte in der JSON Datei ausgelesen, gegebenenfalls bearbeitet, um bspw. den Wochentag oder das Icon zu bestimmen und in HTML-Elementen abgespeichert. Für den animierten Hintergrund werden Schlagworte aus der Beschreibung des Wetters ausgelesen und daraufhin die CSS Eigenschaften für das jeweilige Element überschrieben.

Die Eingabe funktioniert mit Hilfe eines EventListeners, welcher ausgeführt, wenn die Enter-Taste innerhalb des Textfelds betätigt wird. Dabei wird dann die Funktion zur Bestimmung der einzelnen Wetterdaten erneut aufgerufen, dieses Mal jedoch mit einer neuen Stadt.

# Implementation der Aktien-Seite

Ziel der Aktienseite war es den aktuellen Aktienkurs und den historischen Verlauf für unterschiedliche Aktien darzustellen.

## Design

Für das Design überlegte ich mir eine mittige Darstellung des Graphen, welcher rechts von aktuellen Werten und links von der Möglichkeit andere Aktienwerte anzuzeigen umrandet werden. Dabei sollten drei Aktien fest programmiert sein, welche über Buttons angezeigt werden können und ein Textfeld mit dem beliebige Aktienwerte gesucht werden können. Auf der rechten Seite sollten dann wichtige Daten wie Datum, Aktienname, Aktueller Wert und historischer Höchst- und Niedrigstwert der letzten 95 Tage angezeigt werden.

## Umsetzung

Umgesetzt wurde das Ganze mithilfe der Alpha Vantage API, welche Daten für beliebige Aktien zurückgeben kann, sowie chartjs zur Darstellung des historischen Verlaufs. Dabei werden hier ähnlich wie bei der Wetterseite die Daten von der Alpha Vantage Seite heruntergeladen, aufbereitet und an den Graphen weitergegeben. Der Graph wird dann noch so konfiguriert, dass das Ergebnis anschaulich ist und dann an das HTML Dokument weitergegeben. Die drei Buttons rufen eine vorgefertigte Funktion auf während die Suche gleich wie bei der Wetterseite erfolgt.

# Implementation der Wikipedia-Funktion

Ziel war es eine Eingabe zu erstellen, in welche Wikipedia Suchbegriffe eingegeben werden können und diese anschließend vorgelesen werden.

## Design

Diese Funktion sollte direkt auf der Frontpage, mittels einer Überschrift und einem Textfeld mit zugehörigem Button angezeigt werden. Zudem wurde das Aussehen an den Rest der Website angepasst.

## Umsetzung

Die Eingabe erfolgt hier durch ein Textfeld und einen Button. Wird dieser gedrückt so wird mithilfe der Wikipedia API die jeweilige Kurzbeschreibung des eingegebenen Wortes aufgerufen und heruntergeladen. Aus der JSON Datei wird dann die Kurzbeschreibung ausgewertet und an die bei Chrome, Firefox, etc. vorinstallierte Text-To-Speech API weitergegeben, welche diesen dann vorliest. Dabei wird der Button für die Eingabe deaktiviert um eine Mehrfacheingabe zu verhindern. Sobald der Text fertig vorgelesen wurde wird der Button wieder aktiviert.

## Probleme

Probleme ergaben sich hier bei der Wikipedia API. Zuerst musste das Cross-Origin Ressource Sharing Problem umgangen werden, welches das Abrufen der Daten. Dies erreichte ich durch die Verwendung eines Proxys, welche die Anfrage umleitet, nur um dann festzustellen, dass in der normalen Wikipedia API die Kurzbeschreibung nicht enthalten ist. Nach langen Recherchieren habe ich dann einen neuen Link gefunden welcher die Kurzbeschreibung ebenfalls abspeicherte. Außerdem gab es bei manchen Codestrukturen dass der Wikipedia nicht vorgelesen und kein Fehler ausgegeben wurde, was die Fehlersucher erheblich erschwerte.

# Erfahrungsbericht

Insgesamt hat mir das Webengineering sehr gut gefallen, da ich die Möglichkeit hatte meine eigene Webseite von Grund auf aufzubauen und nach meinem Belieben zu gestalten. Dabei fand ich es sehr gut, dass der Großteil der Implementierung durch Learning-By-Doing erfolgte. Ich habe mir zu Beginn überlegt wie die Webseite am Ende aussehen und was für Funktionen sie haben soll und dann einfach angefangen. Zwischendurch sind mir dann bessere Ideen oder Möglichkeiten eingefallen die dann direkt geändert werden konnten. Insgesamt konnte ich dann das theoretisch erlangten Wissen in den Themen HTML, CSS und JavaScript, aber auch in Themen wie APIs oder JSON Files durch die Praxis vertiefen und auch erweitern.

Probleme ergaben sich beim Debugging der Funktionen welches sich aufgrund von fehlenden und ungenauen Fehlermeldung oft sehr schwer gestalteten. Alles in allem konnte jedoch zu jedem Problem auch eine Lösung gefunden werden.

Zum Schluss möchte ich noch anmerken, dass das Programmieren einer Webseite von Grund auf für mich einen besseren Lernfaktor bietet als das reine Ausfüllen von Lücken in einem Template. Somit hätte ich es persönlich angenehmer gefunden hätte man die DHBW Beispielseite im Laufe der Vorlesung gebaut und nicht mit Templates in den Übungen.