

Studienleistung - Weathertop

Abgabe bis Dienstag, den 28.06.2022, 23:00 Uhr über *GitHub Classroom*

Webtechnologien - Prof. Dr. Markus Heckner

Zum Bearbeiten Ihrer Aufgaben verwenden Sie *GitHub classroom*. Mit diesem Service erhält jeder von Ihnen ein eigenes *Repository*, auf dem Sie Ihren Code verwalten können. Beachten Sie dazu die Hinweise auf der Kursplattform zu *GitHub classroom*.

Unter dem folgenden Link erhalten Sie ein Repository, sowie den Startercode für die Studienleistung: <https://classroom.github.com/a/GT8ysrFu>

Tipp zum Zeitmanagement: Fangen Sie so früh wie möglich an. Meist braucht man doch länger zur Lösung eines Programmierproblems als ursprünglich gedacht. Orientieren Sie sich dazu an den Releases am Ende des Dokuments.

Sie können die Studienleistung in Ihrem eigenen Tempo bearbeiten, d.h. es steht Ihnen frei, ob Sie die Studienleistung Woche für Woche erweitern, oder ob Sie alles schneller bearbeiten.

Beachten Sie auch, dass wöchentliche Besprechungen des Projektfortschritts mit dem Dozenten während des Bearbeitungszeitraums verpflichtender Teil der Studienleistung ist. Die Besprechungen finden während der regulären Kurszeiten statt. Wählen Sie dazu einen Zeitslot in Grips, um eine gleichmäßige Verteilung zwischen beiden Kursterminen sicherzustellen. Sobald Sie mit ihrem Projekt fertig sind, müssen Sie es natürlich nicht mehr mit dem Dozenten besprechen.

Achtung, überprüfen Sie Ihre Dateien: Testen Sie vor Ablauf der Deadline, ob Sie alle Dateien korrekt in Ihr Repository übertragen haben. Rufen Sie dazu im Browser die Startseite Ihres Repositories auf und klicken auf *Clone or download* und dann auf *Download ZIP*. In dieser heruntergeladenen Datei müssen alle Dateien Ihres Projekts vorhanden sein. Wenn nicht, können wir Ihre Abgabe auch nicht bewerten!

Die Studienleistung ist ein Projekt, bei dem Sie sich selbstständig in Webtechnologien einarbeiten sollen. Alle Abgaben werden automatisiert untereinander verglichen. Bearbeiten Sie die Studienleistung alleine. Liegt ein Plagiat vor, können Sie den Kurs dieses Semester nicht bestehen.

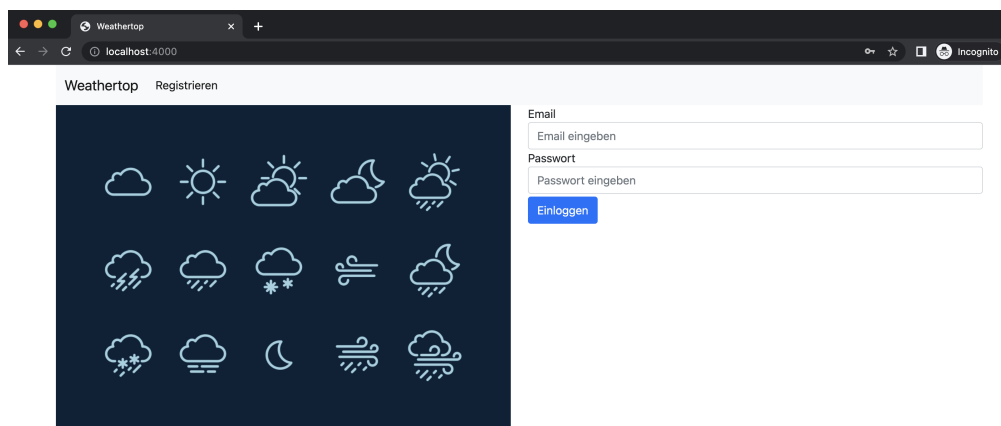
Beachten Sie auch die Hinweise zur Abgabe am Ende dieses Dokuments.

Weathertop

In dieser Aufgabe implementieren Sie *Weathertop*, eine Website, mit der Nutzer eine Liste von Wetterstationen verwalten können und Wetter-Messdaten anzeigen und eingeben können. Alle Nutzer von Weathertop sind begeisterte Hobby-Meteorologen und erfassen und analysieren gerne Wetterdaten.

Lesen Sie zuerst dieses Dokument vollständig durch, bevor Sie mit der Implementierung beginnen!

Die folgende Abbildung zeigt den Startscreen von Weathertop.



Entwicklung in Releases

In der agilen Softwareentwicklung entwickelt man eine Applikation häufig nicht in einem großen Release, der alle Features enthält sondern in kürzeren Zyklen (Sprints) an deren Ende jeweils eine lauffähige Version der Applikation (*Product Increment*) steht.

Dadurch lassen sich technische Risiken reduzieren, da die Software schon während der Entwicklung lauffähig ist, und nicht am Ende viele Komponenten integriert werden müssen. Andererseits erlauben die häufigen Releases Feedback von Nutzern einzuholen, um das Konzept einer App gemäß des Feedbacks anzupassen.

Sie entwickeln Weathertop in den folgenden Releases:

Release	Station – Angezeigte Daten	Reading – Angezeigte Daten	User (verwendete Daten)	Features
0	Name, aktueller Wettercode, aktuelle Temperatur, aktuelle Windgeschwindigkeit, aktueller Luftdruck (alle statisch!)	Wettercode, Temperatur, Windgeschwindigkeit, Luftdruck (alle statisch!)		Anzeige aller Wetterstationen mit Namen der Station und dem aktuelle Wetter auf dem Dashboard. Link auf Detailseite der Station. Detailseite zeigt aktuelles Wetter und alle dazugehörigen Readings. Alle Werte statische Dummywerte, bis jetzt keine Datenbank.
1	Name, aktueller Wettercode, aktuelle Temperatur, aktuelle Windgeschwindigkeit, aktueller Luftdruck	Wettercode, Temperatur, Windgeschwindigkeit, Luftdruck	-	Anzeige aller Wetterstationen mit Namen der Station und dem aktuelle Wetter auf dem Dashboard. Link auf Detailseite der Station. Detailseite zeigt aktuelles Wetter und alle dazugehörigen Readings.
2	+ Längen- und Breitengrad	+ Zeitpunkt der Messung	-	Formulare für das Hinzufügen neuer Stationen und Readings. Löschen von Stationen und Readings. Ergänzen erster Icons.
3			Vorname, Nachname, E-Mail, Passwort	User können sich registrieren, einloggen und ausloggen. Zuordnung der Wetterstationen zu Usern.
4	+ Windrichtung als Himmelsrichtung, + Wettercode als Text, + Maximal und Minimalwerte (für Temperatur, Windgeschwindigkeit und Luftdruck), + Trends für Temperatur, Wind, Luftdruck			Dashboard und Detailseite zeigen jetzt das aktuelle Wetter mit passenden Icons an. Verbesserte textuelle Anzeige. Anzeige von Wettertrends.
5				Stretch goal – EINES der folgenden Features: <ul style="list-style-type: none"> • Reading auf Detailseite der Station automatisch von OpenWeatherMap einlesen • Trendgraph auf Detailseite anzeigen • Wetterstationen auf einer Map anzeigen

Die einzelnen Releases werden in späteren Abschnitten dieses Dokuments noch genauer beschrieben und mit Screenshots der User Interfaces verdeutlicht.

Dabei entspricht jeder Release einem Notenspektrum (vgl. die folgende Abbildung):

Release	Bewertung
0	5,0
1	5,0
2	3,0 – 4,0
3	2,0 – 2,7
4	1,3 – 1,7
5	1,0

Die Tabelle besagt beispielsweise, dass Sie, wenn Sie alle Features der Releases 0 bis 3 umgesetzt haben eine Note zwischen 2,0 und 2,7 erhalten.

Bitte entwickeln Sie Weathertop anhand der vorgegebenen Releases. Dies hilft Ihnen die App strukturiert zu entwickeln und vermindert die Gefahr, dass Sie sich aufgrund einer zu hohen Anzahl offener Baustellen verzetteln. Das Projekt ist umfangreich, d.h. Sie müssen

nicht alle Releases implementieren, um den Kurs zu bestehen. Bitte schätzen Sie sich selbst realistisch ein und orientieren sich bei der Bearbeitung an der obigen Tabelle.

Vorbereitungen

PostgreSQL

Sie benötigen eine PostgreSQL Datenbank, die Sie sich kostenlos bei dem Online-Webhosting-Service Heroku erstellen können, falls Sie das nicht bereits getan haben. Gehen Sie dazu wie folgt vor:

1. Öffnen Sie die Seite <https://www.heroku.com> und erstellen Sie sich einen Account, falls Sie noch keinen besitzen.
2. Klicken Sie, nach dem Login, auf dem Heroku Dashboard auf *New* und wählen Sie *Create new app*.
3. Vergeben Sie einen Namen für Ihre Anwendung und klicken Sie *Create app*
4. Auf der *Resources* PostgreSQL in das Suchfeld eingeben und *Heroku Postgres* auswählen.
5. Den kostenfreien Plan *Hobby Dev - Free* auswählen und auf *Provision* klicken.
6. Auf den Link *Heroku Postgres :: Database* klicken.
7. Klick auf *Settings*, dann *View Credentials*. Hier finden Sie die Parameter für den Zugriff auf die Datenbank.

Mit den obigen Verbindungsparametern können Sie jetzt auf Ihre Datenbank über die Website Adminer <https://adminer.cs50.net/> zugreifen.

Auf der Heroku *Settings*-Seite erhalten Sie eine URL, mit der Sie die Verbindung zur Datenbank aufbauen können. Diese ist nach dem folgenden Format aufgebaut:

`postgres://user:pass@host:PORT/dbname`. Diese benötigen Sie in einer `.env`-Datei, damit Sie mit Ihrer Web-App auf Ihre Datenbank zugreifen können (weitere Informationen dazu im Abschnitt *Umgebungsvariablen*).

Startercode aus GitHub laden und sich mit dem Code vertraut machen

Laden Sie sich den Startercode aus GitHub Classroom (siehe Anfang dieses Dokuments) und machen Sie sich mit dem Code vertraut. Installieren Sie alle in `package.json` enthaltenen Pakete mit `npm install`.

Umgebungsvariablen

Lagern Sie die Umgebungsvariablen in eine Datei **.env** aus. Der Inhalt dieser Datei sieht wie folgt aus, wobei Sie die Werte für die Variablen durch Ihre Eigenen ersetzen müssen:

```
1 PORT=3000
2 DB_CON_STRING=YOUR_HEROKU_CONNECTION_STRING
```

Um auf diese Variablen in Node zuzugreifen, benötigen Sie das npm-Paket **dotenv**, das Sie installieren müssen. Das Paket ist bereits in **app.js** importiert.

```
1 const dotenv = require('dotenv');
```

Die Variablen werden mit dem folgenden Befehl aus der Datei **.env** geladen.

```
1 dotenv.config();
```

Anschließend können diese in der Webanwendung genutzt werden. Achtung: In einem echten Webprojekt würden Sie diese Datei nicht auf Github hochladen, da sonst Ihre Zugangsdaten für Dritte einsehbar sind. Für diese Aufgabe ist es aufgrund der Korrektur aber in Ordnung, da die App sonst nicht testbar ist.

Implementierung von Weathertop

Verwenden Sie die folgenden Technologien und Prinzipien:

- Node.js und Express
- Template Engine Handlebars
- PostgreSQL
- Bootstrap
- *Model View Controller*

Sie können sich bei der visuellen Gestaltung an den Screenshots dieser Aufgabenstellung orientieren oder ein eigenes ansprechendes Design entwickeln.

Verwenden Sie die Entwicklungsumgebung Webstorm. Verwenden Sie nicht repl.it, da Sie Ihr Projekt dort nicht debuggen können, zusätzlich ist ihr Code dann öffentlich einsehbar (vgl. Hinweise zu Plagiat).

Release 0 - Statische Daten

Beginnen Sie mit der Erstellung einer statischen Webseite mit Node, Express und Handlebars.

Die Startseite (Route `/`) zeigt ein Menü und eine Graphik mit Wettericons an (Sie können gerne eigene Bilder verwenden!). Implementieren Sie die Route `/dashboard` und zeigen dort eine Übersicht aller Wetterstationen an. Diese Liste zeigt eine Reihe von Wetterstationen und deren Namen an. In der Liste tauchen auch die jeweils aktuellen Werte für das Wetter auf (d.h. Wetter, Temperatur, Windgeschwindigkeit und Luftdruck). Das aktuelle Wetter wird dabei durch einen Wettercode repräsentiert. Die folgende URL zeigt eine Liste dieser Codes: <https://openweathermap.org/weather-conditions> - Es reicht wenn Sie die vollen Hunderter verwenden, d.h. beispielsweise 200 für *Thunderstorm* oder übersetzt Gewitter bzw. Unwetter.

Jede Wetterstation verfügt über einen Link auf eine Seite mit Details der Wetterstation. Dort finden Sie das aktuelle Wetter, sowie eine Übersicht aller bisherigen **readings** (d.h. Messungen des Wetters zu einem bestimmten Zeitpunkt).

Verwenden Sie für dieses Release statische Daten, d.h. tragen Sie feste Werte in die Templates ein. Die dynamische Anzeige der Daten aus einer Datenbank folgt erst ab dem nächsten Release.

Die folgenden Screenshots zeigen ein mögliches User Interface für das Release 0:

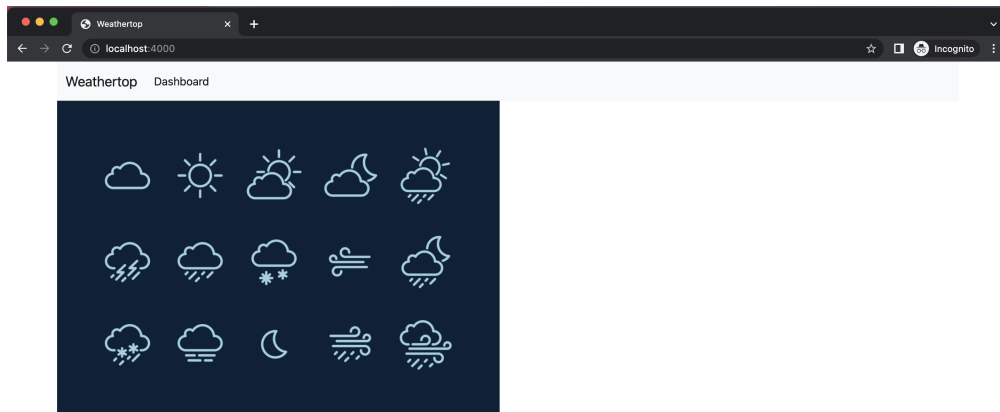


Abbildung 1: Indexseite Release 0

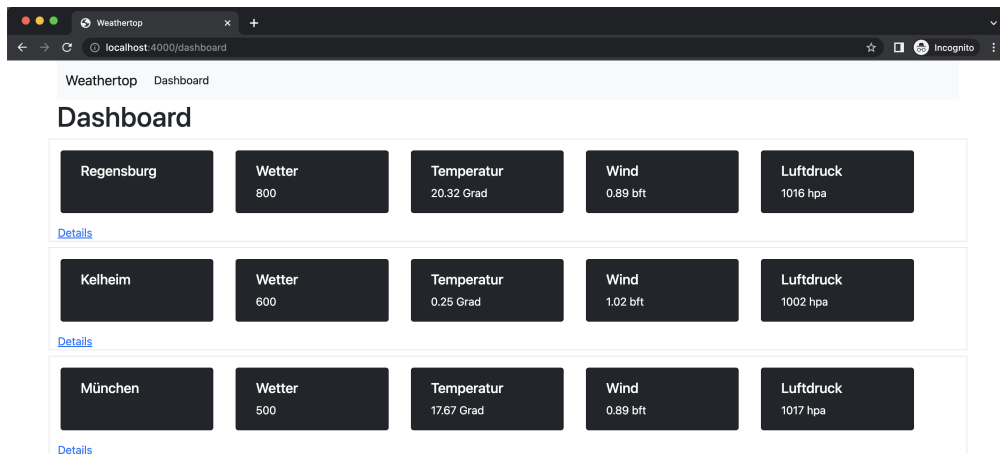


Abbildung 2: Dashboard Release 0

Wetter	Temperatur	Wind	Luftdruck
800	20.32	0.89	1016
800	20.32	0.89	1016
400	32.5	15	90
400	32.5	15	90
800	11.16	2.23	1024
800	11.16	2.23	1024
800	-6.14	1.15	1027

Abbildung 3: Details Wetterstation Release 0

Release 1

Erweitern Sie jetzt die Seite um eine dynamische Anzeige der Daten aus einer Datenbank, das User Interface ist identisch zum ersten Release.

Überlegen Sie welche Tabellen Sie in der Datenbank benötigen, um die Web-App für den ersten Release zu entwickeln.

Release 2

Ergänzen Sie Formulare für das Hinzufügen von Stationen und Readings. Zusätzlich sollen sich die Stationen und Readings löschen lassen. Ergänzen Sie Icons für den Link zu den Details der Stations, sowie für das Löschen der Stationen. Geeignete Icons finden Sie in der Bootstrap Icon-Bibliothek: <https://icons.getbootstrap.com/>. Dort finden Sie auch eine Anleitung, wie sich Icons in HTML-Dokumente integrieren lassen: <https://icons.getbootstrap.com/#usage>. Zeigen Sie auch den Zeitpunkt des Readings in der Liste der Readings an.

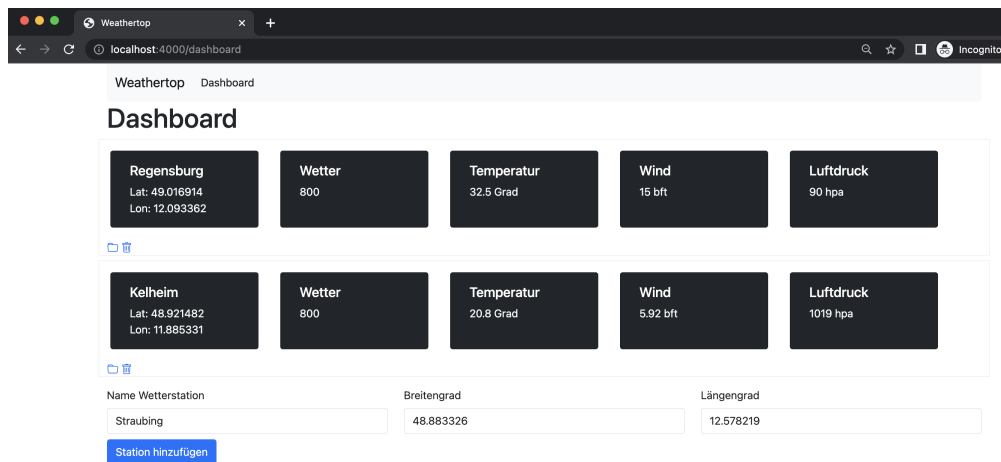


Abbildung 4: Dashboard Release 2 - Formular zum Hinzufügen von Stationen, Anzeige von Längen- und Breitengrad

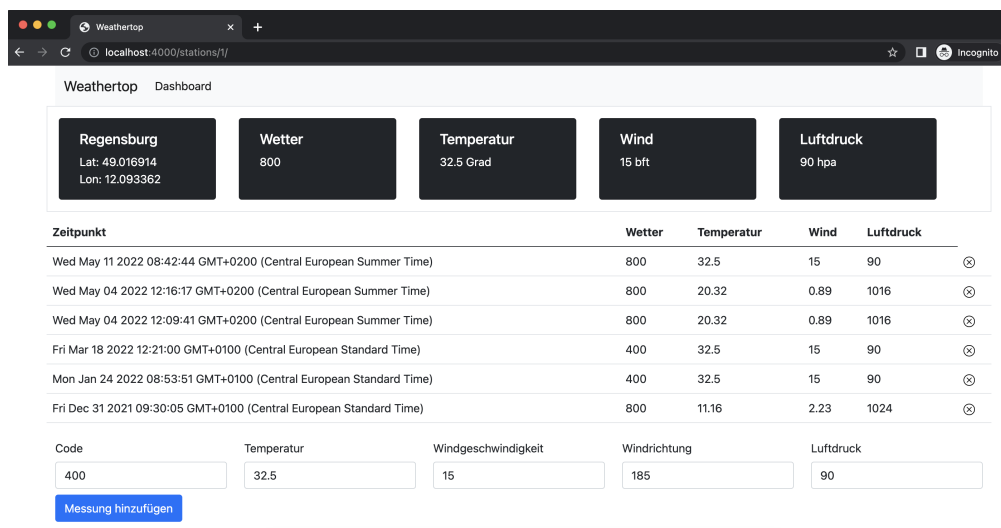


Abbildung 5: Dashboard Release 2 - Formular zum Hinzufügen von Readings und Funktion zum Löschen eines Readings

Release 3

Ergänzen Sie eine Registrierungs- und Loginfunktion für Nutzer. Die Web-App zeigt ab jetzt nur noch die Wetterstationen des eingeloggten Nutzers an. Ist kein Nutzer eingeloggt, werden auch keine Wetterstationen und Readings mehr angezeigt. Erfolgt ein Zugriff auf diese Routen, werden die Anfragen an die Loginseite weitergeleitet.

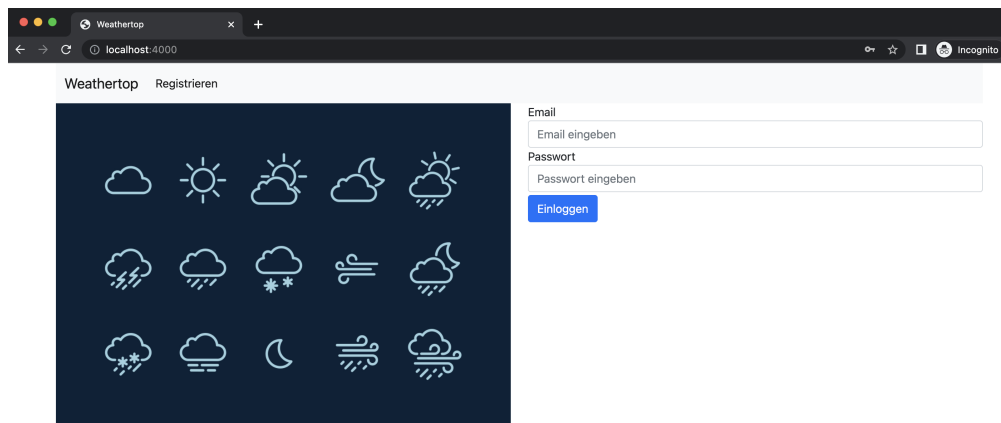


Abbildung 6: Startseite Release 3 - Formular Login

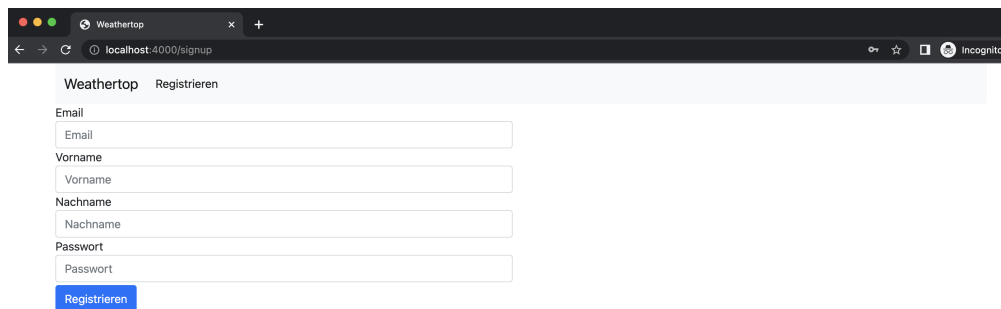


Abbildung 7: Release 3 - Formular Registrierung

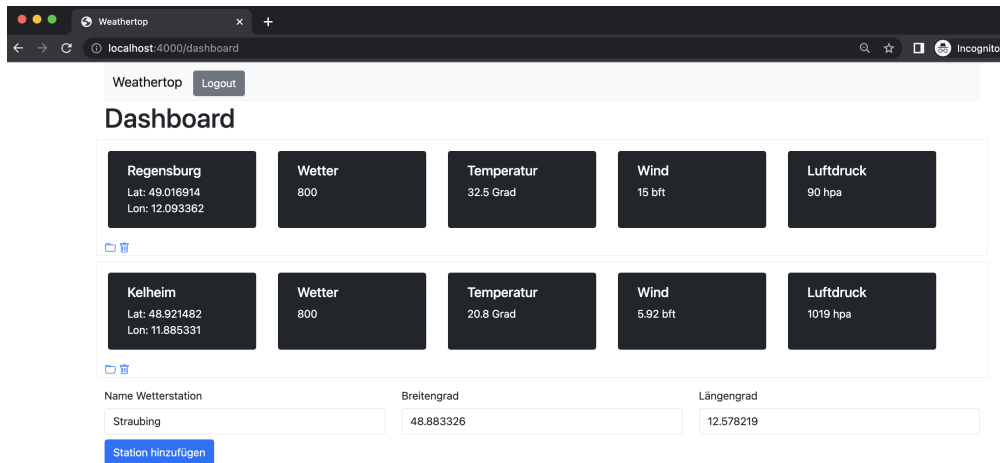


Abbildung 8: Dashboard Release 3 - Button für Logout (bei eingeloggtem Nutzer)

Release 4

Passen Sie das Dashboard an, sodass dort auch das aktuelle Wetter mit passenden Icons dargestellt wird (auch diese Icons finden Sie in der Bootstrap Icon-Bibliothek). Die Detailseite der Wetterstation sollte diese Daten ebenfalls anzeigen. Ergänzen Sie Maximal- und Minimalwerte für Temperatur, Windgeschwindigkeit und Luftdruck. Zeigen Sie das aktuelle Wetter als Text an, d.h. übersetzen Sie den Code in eine geeignete Beschreibung. Die folgende URL dient Ihnen zur Orientierung: <https://openweathermap.org/weather-conditions>.

Konvertieren Sie die Windrichtung von Grad in eine lesbare Windrichtung. Sie können sich für die Konvertierung an folgender Tabelle orientieren:

<http://snowfence.umn.edu/Components/winddirectionanddegrees.htm>.

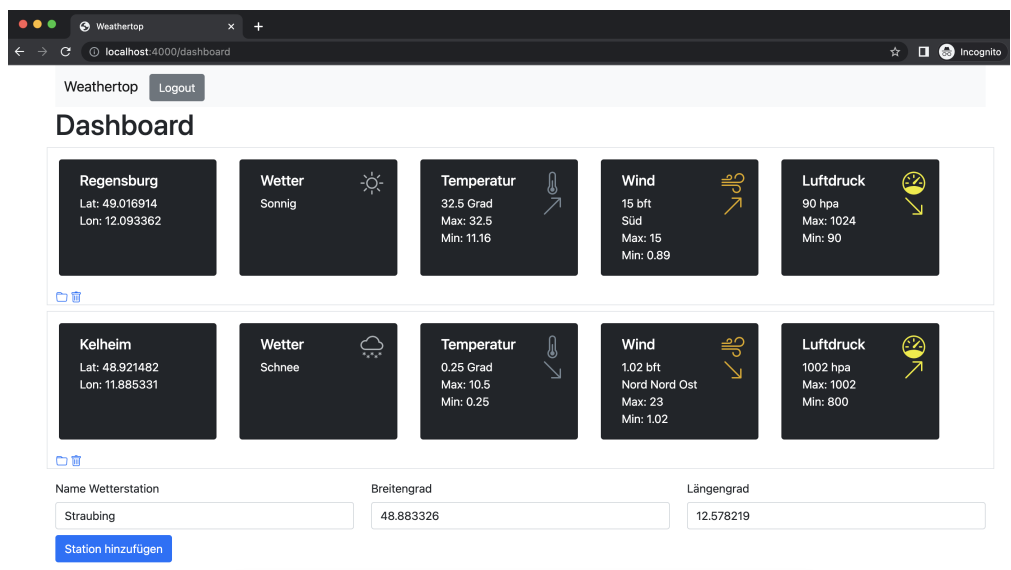


Abbildung 9: Release 4 - Dashboard mit erweiterten Wetterdaten

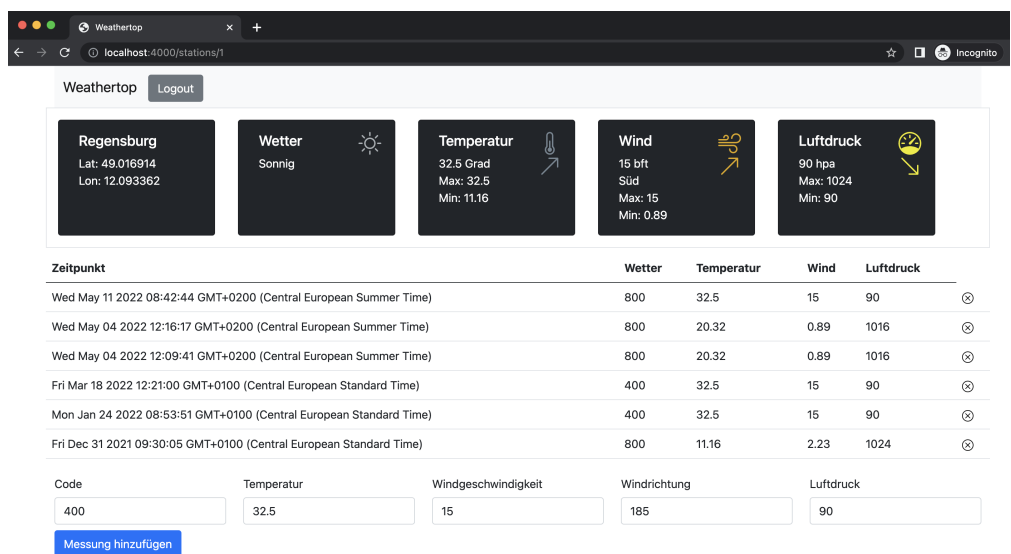


Abbildung 10: Release 4 - Erweiterte Wetterdaten auf der Detailseite der Stationen

Release 5

Erweitern Sie Ihre Web-App um **eines** der folgenden drei Features:

- Ergänzen Sie eine Karte auf dem Dashboard zur Anzeige aller Wetterstationen. Klicken die User auf einen Marker in der Karte, gelangen sie zur Detailseite der Station.

- Anzeige von Wettertrends in einem Chart. Ergänzen Sie die Wetterstation um eine Wettervorhersage, die den Trend von Temperatur und Wind für die nächsten 7 Tage anzeigt. Die Daten erhalten Sie über das API von <https://openweathermap.org/>
- Ergänzen Sie eine Funktion zur Erstellung einer automatisierten Messung. Die User können durch Klick auf einen Button aktuelle Wetterwerte über das API von Open-Weathermap einlesen und in die Datenbank hinzufügen. Somit müssen die Daten nicht mehr manuell über das Formular eingegeben werden.

Hinweis: Hier finden Sie eine Einführung zur Charts und dem Open Weather API: <https://reader.tutors.dev/#/topic/webdev-ai-ds-oth.netlify.app/topic-07-weather>. Eine Dokumentation zu Maps finden Sie hier: <https://leafletjs.com/>.

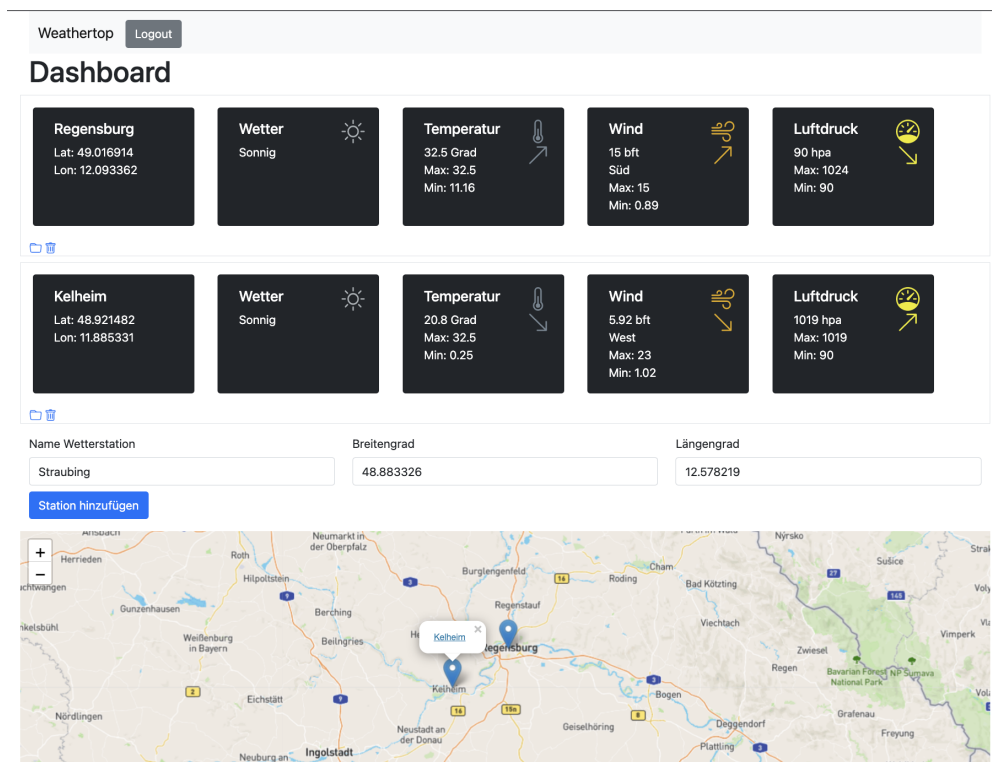


Abbildung 11: Release 5 - Anzeige der Wetterstationen auf einer Karte

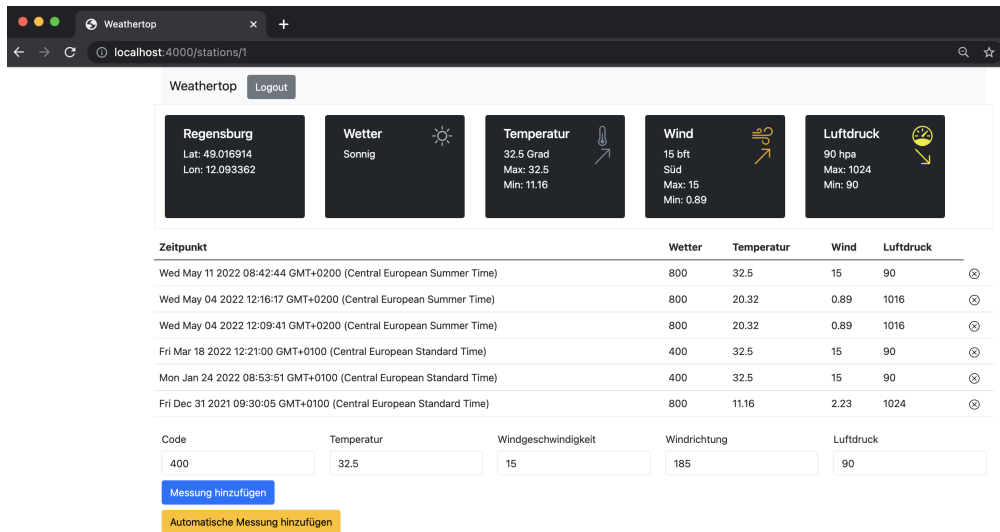


Abbildung 12: Release 5 - Button zum Hinzufügen einer automatischen Messung

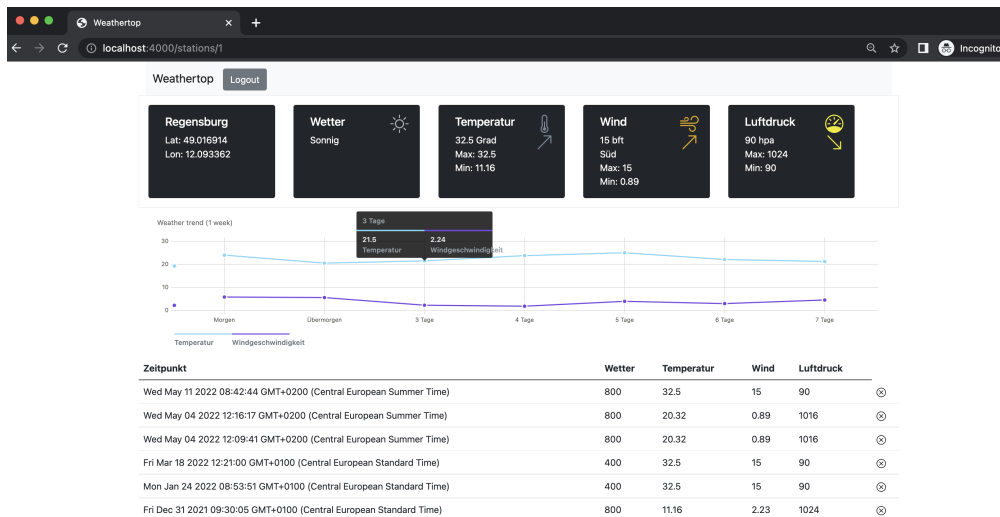


Abbildung 13: Release 5 - Wettervorhersage für die nächsten 7 Tage

Testen

Testen Sie Ihre Website ausführlich, z.B.

- Geben Sie Buchstaben in Felder ein, die nur Zahlen akzeptieren sollen.
- Geben Sie eine 0 oder negative Zahlen in Felder ein, die nur positive Zahlen akzeptieren sollen.

- Geben Sie Fließkommazahlen in Felder ein, die nur Ganzzahlen akzeptieren sollen.
- Rufen Sie Routen auf, die nur für eingeloggte Benutzer sichtbar sein sollen.

Tipps und Tricks

- Wenn Sie in Handlebars in einer **#each**-Schleife über Listen iterieren, dann können Sie mit **this** auf das jeweilige Objekt der Liste zugreifen. Manchmal wollen Sie aber auch auf weitere Variablen des Templates zugreifen. Dazu können Sie den Kontext (d.h. das aktuelle Objekt der Liste) mit **../** verlassen. Vgl. dazu hier: <https://handlebarsjs.com/guide/expressions.html#path-expressions>.
- Stellen Sie alle Fragen auf piazza:
<https://piazza.com/class/l10rl9aplb9lv>

Verpflichtende Fortschrittsbesprechungen und Zeitmanagement

Im Rahmen der zweiten Studienleistung müssen Sie Ihren Code während der Bearbeitung regelmäßig mit dem Dozierenden besprechen und diskutieren. Diese wöchentlichen Besprechungen sind verpflichtend.

Abgabe

Die Abgabe besteht aus den folgenden Deliverables:

1. Code über Github
2. Ein Video in dem Sie das UI und ihren Code in maximal 5 Minuten vorstellen. Sie können das Video mit einer Software Ihrer Wahl aufzeichnen oder die Software Snagit verwenden. Snagit ist für den Campus der OTH lizenziert und Sie können sich hier eine kostenlose Lizenz beschaffen: <https://www.studisoftware.de/shibboleth/shibdwayf?vi=19>. So fügen Sie das Video der Abgabe hinzu:
 - a) Video erstellen
 - b) Video auf der Plattform der OTH hochladen:
<https://vimp.oth-regensburg.de/>. Wählen Sie dort **Passwortgeschützt** beim Upload aus.
 - c) Fügen Sie den Link und das Passwort der Datei readme.md in Ihrem Code wie folgt hinzu:
Video: LINK ZU VIMP
Passwort: IHR_PASSWORT
3. Selbsteinschätzung - Füllen Sie das Template *Self Assessment* aus und fügen Sie dieses Dokument ebenfalls Ihrem Repository hinzu.

Stellen Sie anschließend sicher, dass die obigen Informationen Teil des Repositories sind und vergessen Sie nicht zu *committen* und zu *pushen*.