

Studienleistung - OTHer

Abgabe bis Freitag, den 20.01.2023, 23:00 Uhr über GitHub Classroom

Medieninformatik Webtechnologien - Prof. Dr. Markus Heckner

Zum Bearbeiten der 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/9o5NvwUr>

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 in diesem Dokument vorgestellten Releases.

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 vier wöchentliche Besprechungen des Projektfortschritts mit dem Dozenten während des Bearbeitungszeitraums verpflichtender Teil der Studienleistung sind. Die Besprechungen finden während der regulären Kurszeiten statt. Wählen Sie dazu einen Zeitslot in Grips, um eine gleichmäßige Verteilung zwischen den Terminen sicherzustellen. Sobald Sie mit ihrem Projekt fertig sind, müssen Sie es natürlich nicht mehr mit dem Dozenten besprechen, dies kann auch schon nach weniger als vier Besprechungen sein.

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!

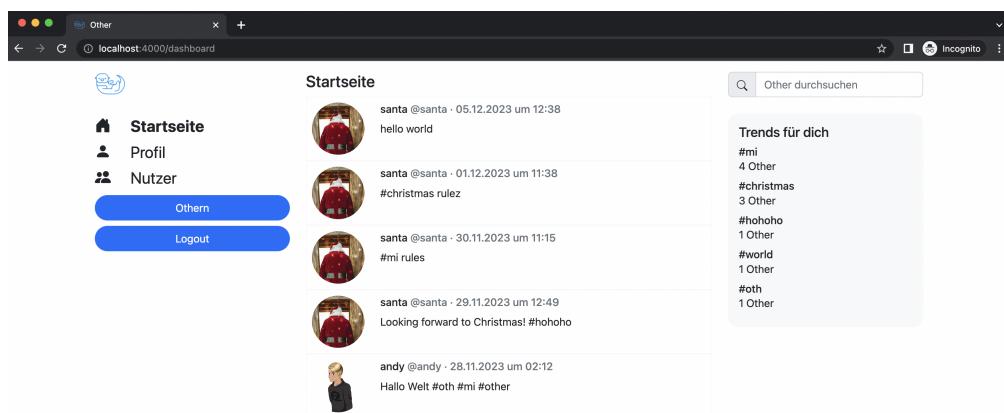
Die Studienleistung ist ein Projekt, bei dem Sie sich selbstständig in Webtechnologien einarbeiten sollen. Alle Abgaben werden automatisiert untereinander verglichen. Bearbeiten Sie die Studienleistung alleine. Liegt ein Plagiat vor, können Sie den Kurs dieses Semester nicht bestehen.

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

OTHer - Überblick und Vorbereitungen

Der Kurznachrichtendienst Twitter wird seit kurzer Zeit kontrovers in Presse und Öffentlichkeit diskutiert. Es ist daher Zeit, dass Sie eine Alternative schaffen: In dieser Aufgabe implementieren Sie ihren eigenen Kurznachrichtendienst *OTHer*. Mit *OTHer* ist es möglich eigene Nachrichten (*OTHer*) zu veröffentlichen (*othern*) und anderen Nutzern zu folgen, um deren Nachrichten zu lesen. Weiterhin zeigt *OTHer* die aktuell im Trend liegenden Hashtags an, damit Sie immer über aktuelle Ereignisse informiert sind. *OTHer* bietet auch eine Suchfunktion, mit der Sie Nutzer suchen können, um sich deren *Other* anzusehen oder ihnen zu folgen.

Die folgende Abbildung zeigt einen beispielhaften Screen von *OTHer*.



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

Technologiestack

Verwenden Sie die folgenden Technologien:

- Node.js und Express
- Template Engine
- PostgreSQL
- Bootstrap

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

Ich empfehle Ihnen die Verwendung von Bootstrap in der aktuellen Version 5.2:

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Verwenden Sie die Entwicklungsumgebung Webstorm oder eine andere JavaScript-IDE, mit der sich Node.js Projekte debuggen lassen (z.B. Visual Studio Code).

Releases im Überblick

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 OTHer in den folgenden Releases:

Release 0 - Walking Skeleton

Im ersten Release sollen Sie eine erste Version der Seite mit Node.js und Express implementieren, die über ausgewählte Views verfügt, aber noch keine Datenbankanbindung und implementierte Logik enthält. Mit diesem *Walking Skeleton* können Sie die erste Funktion von OTHer zeigen, ohne diese im Detail implementiert zu haben.

Features:

- **Startseite** (später nur für eingeloggte User sichtbar) - Diese Seite enthält eine Navigation in der linken Spalte. In der Navigation befindet sich ein Link zu dieser Startseite selbst und ein Link zu einer Übersicht aller in OTHer registrierten Nutzer. In der Mitte werden alle in OTHer erstellten Others angezeigt. In der rechten Spalte befindet sich eine Suchleiste, mit der man später nach registrierten Usern suchen kann.
- **Nutzerübersicht** - Die linke und rechte Spalte entsprechen dem Dashboard. Die mittlere Spalte enthält eine alphabetisch sortierte Übersicht aller in OTHer registrierten User.

Release 1 - User und Others aus der Datenbank

In diesem Release importieren Sie bestehende Daten von Usern und Others in eine eigene Datenbank auf den Servern der OTH Regensburg.

Features:

- **Anzeige aller Others auf der Startseite** - Die Startseite zeigt jetzt alle Others aller in der Datenbank befindlichen User an.
- **Nutzerübersicht mit allen Nutzern** - Auch die Nutzerseite zeigt jetzt keine Dummydaten mehr, sondern zeigt alle in der Datenbank enthaltenen User an.

Release 2 - Login und Logout

In diesem Release ergänzen Sie OTHer um Funktionen zur Registrierung und zum Einloggen.

Features:

- **Landingpage** - Die Landingpage wird allen nicht registrierten Usern und allen ausgelogten Usern angezeigt. Die Navigationsleiste enthält lediglich einen Link zur Startseite (in diesem Fall der Landingpage) und zwei Buttons zur Registrierung und zum Login.
- **Registrierung** - Diese Seite zeigt ein Registrierungsformular für neue User an. Neue User werden nach Registrierung in der Datenbank angelegt.
- **Login** - Diese Seite ermöglicht es registrierten Usern sich bei OTHer anzumelden. Nach dem Login erscheint die im vorigen Release entwickelte Startseite.
- **Profilseite** - Die Profilseite zeigt Detailinformationen (Bild, Username, Geburtsdatum, Kurze Biographie und Registrierungsdatum) des eingeloggten Users an. Ergänzen Sie auch einen Link zur Profilseite in der Navigation.
- **Anpassungen Nutzerübersicht** - Klickt man auf den Nutzernamen eines Users auf der Übersicht aller Nutzer wird das entsprechende Profil des Nutzers angezeigt.

Release 3 - Othern

In diesem Release ergänzen Sie OTHer um die Funktion Others zu erstellen.

Features:

- **Navigation ergänzen** - Ergänzen Sie die Navigation um einen Button Othern. Bei Klick auf den Button erscheint die Seite Other erstellen.
- **Other erstellen** - Diese Seite zeigt ein Textfeld und einen Button othern. Ein Klick auf den Button erstellt einen neuen Other für den eingeloggten User.
- **Anpassungen Anzeige aller Others auf der Startseite** - Der erstellte Other wird jetzt ebenfalls in der Liste aller Others angezeigt, d.h. zusätzlich zu den Others von anderen Usern.
- **Anpassungen Profil** - Ergänzen Sie das Userprofil um eine Anzeige der Anzahl der erstellten Others des jeweiligen Users.

Release 4 - Folgen und Entfolgen

In diesem Release erweitern Sie OTHer um eine Funktion anderen Usern zu folgen und diese wieder zu entfolgen.

Features:

- **Anpassungen Nutzerübersicht** - Die Seite Nutzer ermöglicht es jetzt Nutzern zu folgen und diese zu entfolgen. Nach Klick auf den jeweiligen Button wird man zum Dashboard weitergeleitet. Klickt man in den Navigation auf die Nutzerseite, zeigt diese ge-

änderten Zustand bei dem entsprechenden User an. D.h. Sie *Folgen* einem User, dann wird dieser User auf der Nutzerseite mit einem Button *Entfolgen* angezeigt. Klicken Sie auf *Entfolgen*, zeigt die Nutzerseite bei diesem User wieder den *Folgen* Button an.

- **Anpassungen Startseite** - Die Startseite sieht wie vorher aus, lediglich die angezeigten Daten ändern sich: In der Mitte sollen jetzt nur die Others von Usern angezeigt werden, denen der eingeloggte User folgt. Somit erscheinen für neue User jetzt keine Others mehr auf der Startseite, da neue User noch keinen Usern folgen.
- **Anpassungen Nutzerprofil** - Passen Sie die Nutzerprofilseite entsprechend an, so dass dort angezeigt wird wie viele Follower ein User hat, bzw. wie vielen Usern ein bestimmter User folgt.

Release 5 - Suche

In diesem Release ergänzen Sie eine Suche nach Nutzern.

Features:

- **Suche starten** - In die Suchleiste kann jetzt ein Username eingegeben werden. Schickt man die Suche ab, werden die Suchergebnisse angezeigt.
- **Suchergebnisse** - Die Suchseite zeigt die Suchergebnisse der Suche an. Die Seite ist ähnlich zur Nutzerübersicht aufgebaut, enthält aber nur zur Suche passende User.

Release 6 - Trends und Profilbild aktualisieren

In diesem Release erweitern Sie OTHer um eine Anzeige von *trendenden Hashtags* und die Möglichkeit das Profilbild zu ändern.

Features:

- **Trends** - In der rechten Spalte findet sich eine Übersicht aller im Trend liegenden Hashtags: Die Anzeige enthält eine absteigend sortierte Liste der 5 häufigst vergebenen Hashtags.
- **Profilbild aktualisieren** - Die Nutzer haben die Möglichkeit ihr Profilbild zu ändern.

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

Bewertung

Jeder Release entspricht einem Notenspektrum, vgl. die folgende Tabelle zur Orientierung:

Release	Bewertung
0	5,0
1	5,0
2	3,7 - 5,0
3	3,0 - 3,7
4	2,3 - 3,0
5	1,7 - 2,3
6	1,0 - 1,7

Die Tabelle besagt beispielsweise, dass Sie, wenn Sie alle Features der Releases 0 bis 5 umgesetzt haben, eine Note zwischen 1,7 und 2,3 erhalten. Die teilweisen Überlappungen ergeben sich aus den folgenden zusätzlichen Bewertungskriterien:

- **Korrektheit** - Funktioniert der Code wie er soll?
- **Codequalität** - Ist der Code gut verständlich? Werden Wiederholungen vermieden?

Sehr starke Empfehlung:

Bitte entwickeln Sie OTHer 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

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
```

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 (Web-)projekt 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.

Releases im Detail

Release 0 - Walking Skeleton

Beginnen Sie mit einer statischen Website auf Basis von Node.js und Express. Implementieren Sie die Route `/dashboard` und zeigen dort die Startseite an. Implementieren Sie die Route `users` für die Anzeige der Nutzerübersicht.

Diese Seite bestimmt die Grundstruktur von OTHer. Alle weiteren Seiten folgen diesem dreispaltigen Muster: In der linken Spalte befindet sich immer die Navigation, in der mittleren Spalte werden die Inhalte (z.B. Others, Userprofil, Formulare für Login oder Registrierung) angezeigt und in der rechten Spalte immer die Suche und bei Bedarf weitere Zusatzinformationen (vgl. spätere Releases).

Für die Navigation und die Suchleiste benötigen Sie Icons aus der Bootstrap Iconbibliothek (<https://icons.getbootstrap.com/>). Anstatt die Bootstrap Icons als `svg`-Elemente wie Bilder in HTML einzubinden, können Sie diese auch recht einfach über einen Icon-Font einbinden.

Dazu müssen Sie im Head der HTML-Datei die Schriftart, zusätzlich zu Bootstrap, wie folgt einbinden:

```
1 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.3.0/font/bootstrap-icons.css">
```

Fügen Sie anschließend die Icons als Bootstrap Icon-Font wie folgt in HTML ein:

```
1 <i class="bi-house-door-fill"></i>
```

Mit dem folgenden Code können Sie ein Icon und danebenstehenden Text vertikal ausrichten (z.B. für die Navigationsleiste):

```

1 <div class="fs-3">
2   <i class="bi bi-house-door-fill"></i>
3   <span class="align-middle"> Startseite </span>
4 </div>

```

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

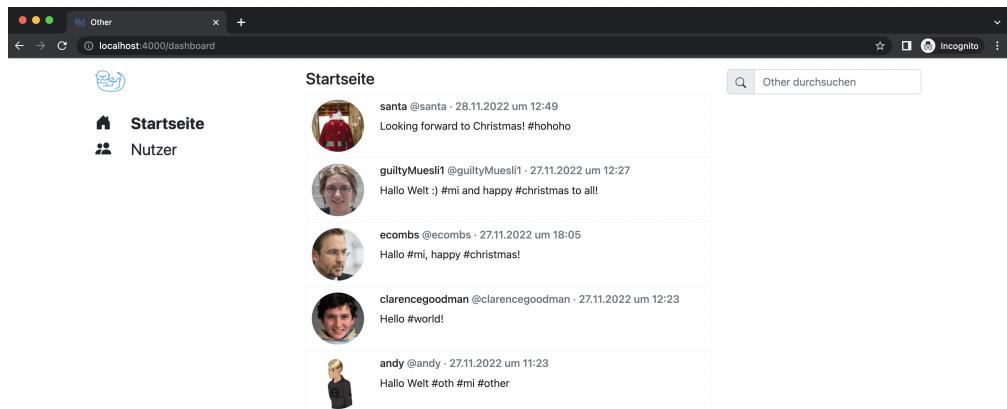


Abbildung 1: Release 0 - Dashboard

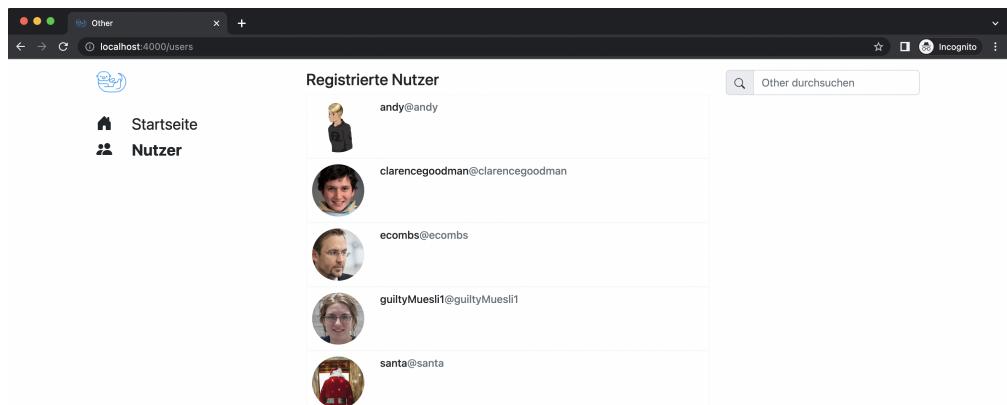


Abbildung 2: Release 0 - Nutzerübersicht

Release 1 - User und Others aus der Datenbank

Für diesen Release benötigen Sie eine PostgreSQL Datenbank. Diese wird Ihnen kostenlos von der OTH bereitgestellt. Um diesen Server nutzen zu können, müssen Sie **zwingend** mit dem VPN der OTH-Regensburg verbunden sein.

Die Verbindung zur Datenbank können Sie z.B. mit den Tool *DBeaver* herstellen, das Sie auf der folgenden Seite herunterladen können: <https://dbeaver.io/download/>. Mit diesem

Tool können Sie SQL-Queries an den Server schicken oder sich den Inhalt und die Struktur von Tabellen per Mausklick anzeigen lassen.

Der folgende Screenshot zeigt die Verbindungsparameter für *DBeaver* an:

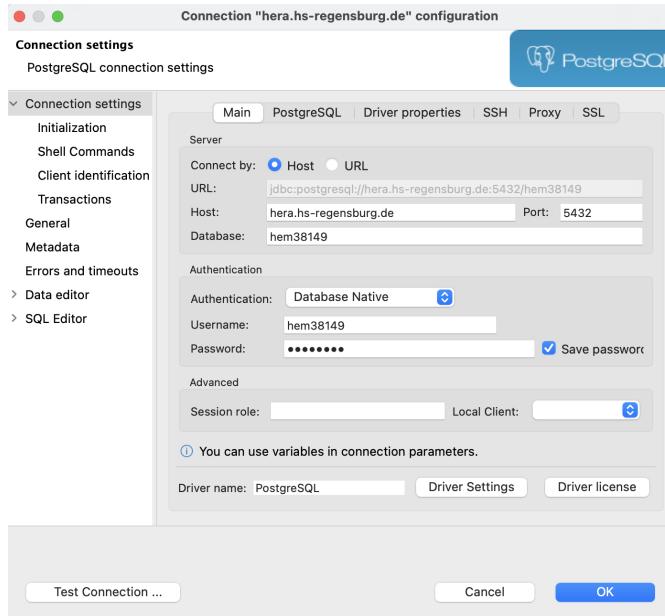


Abbildung 3: Verbindungsparameter *DBeaver*

Im Feld Passwort tragen Sie das voreingestellte Passwort ein, dieses entspricht Ihrem NDS-Kürzel.

Achtung: Ändern Sie direkt im Anschluss unbedingt Ihr Passwort, wie auf dem folgenden Screenshot dargestellt:

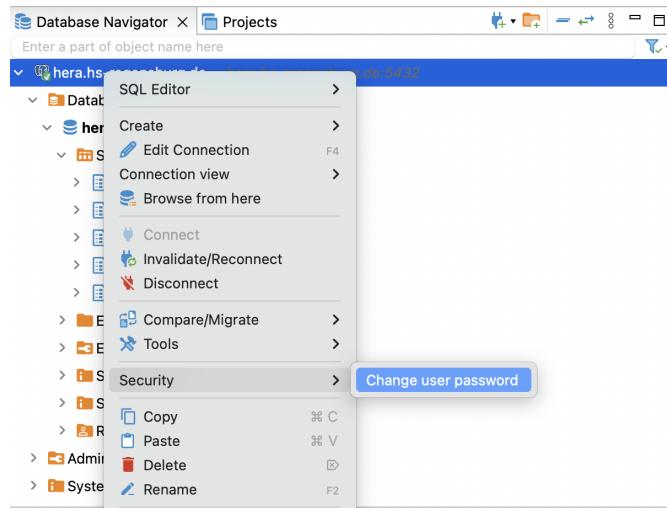


Abbildung 4: DBeaver - Passwort ändern

Klicken Sie jetzt auf den Button SQL, um die SQL-Console zu öffnen, in der sie SQL-Befehle eingeben können. Die erstellten Tabellen erscheinen unter public, wie auf dem folgenden Screenshot dargestellt:

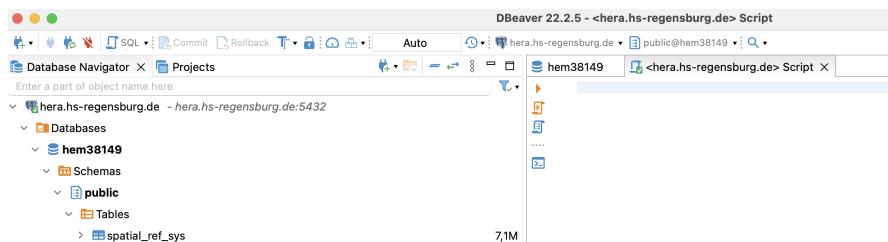


Abbildung 5: Datenbanktabellen in DBeaver

Über Node.js und dem Package `pg` können Sie wie folgt auf die Datenbank zugreifen (ersetzen Sie `abc12345` und `password` durch ihre eigenen Daten):

```
postgres://abc12345:password@hera.hs-regensburg.de:5432/abc12345
```

Ergänzen Sie die Datei `.env` um die folgende Zeile, um die Verbindungsdaten über Node.js auslesen zu können:

```
postgres://abc12345:password@hera.hs-regensburg.de:5432/abc12345
```

In diesem Release müssen Sie die Daten für User und deren Others aus 2 verschiedenen CSV-Dateien in Ihre Datenbank importieren. Die Datei `users.csv` enthält die User für OTHer, die Datei `others.csv` enthält die Others, die diese User erstellt haben. Diese Dateien enthalten pro Zeile einen Datensatz, dessen Attribute bzw. Spalten durch Kommas getrennt werden (daher der Name **CommaSeparatedValue**).

Die Datei `users.csv` ist dabei spaltenweise wie folgt aufgebaut:

- **user_id** - Eindeutige ID des Users
- **name** - Username
- **password** - Passwort
- **birthday** - Geburtsdatum
- **profile_pic** - Name des Bildes für das Profil (Dieser Name verweist auf eine png-Datei im Ordner **public/img/users** des Webprojekts (vgl. Starterprojekt). Die Bilder selbst sind also bereits im Webprojekt vorhanden und müssen nicht in die Datenbank importiert werden. Die Namen für die Bilder stimmen dabei nicht mit dem Namen der User überein, Sie müssen aber trotzdem keine Bilder umbenennen und können die Namen so verwenden, wie in der CSV-Datei angegeben.)
- **bio_text** - Kurzer Text über den User für das Profil
- **created** - Zeitpunkt der Registrierung des Users

Die Datei **others.csv** ist dabei spaltenweise wie folgt aufgebaut:

- **post_id** - Eindeutige ID des others
- **user_id** - ID des Users, der den Other erstellt hat
- **text** - Text des Others
- **created** - Erstellungszeitpunkt des Others

Der folgende Code-Ausschnitt zeigt das Einlesen eines CSV-Files und die Ausgabe auf der Kommandozeile:

```

1 const fs = require("fs");
2 const csvparser = require("csv-parse");
3
4 const processFile = async () => {
5     const parser = fs
6         .createReadStream("users.csv")
7         .pipe(csvparser.parse());
8     for await (const record of parser) {
9         console.log(record[1]);
10    }
11 };
12
13 (async () => {
14     await processFile();
15 })();

```

Erstellen Sie zunächst die Tabelle **users** in ihrer Datenbank und importieren dann die Daten aus der entsprechenden CSV-Datei. Gehen Sie anschließend analog für die Tabelle **others** vor. Erstellen Sie dazu die Datei *import.js* in Ihrem Projekt. Verwenden Sie dazu das Package **csv-parse** (vgl. https://csv.js.org/parse/recipes/async_iterator/).

Dabei ist **record** ein Array, das jeweils einer Zeile der Datei entspricht. Mithilfe des Indizes können Sie auf die einzelnen Spalten der Zeile zugreifen.

Die folgenden Screenshots zeigen ein mögliches User Interface für das Release 1:

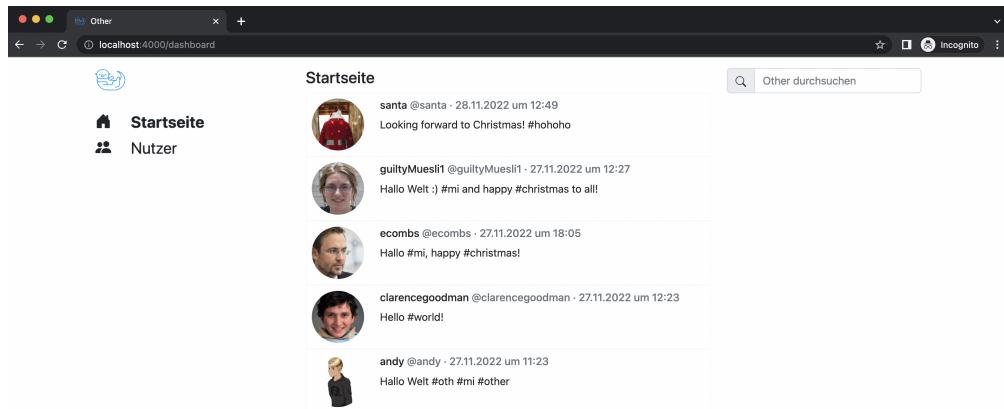


Abbildung 6: Release 1 - Dashboard

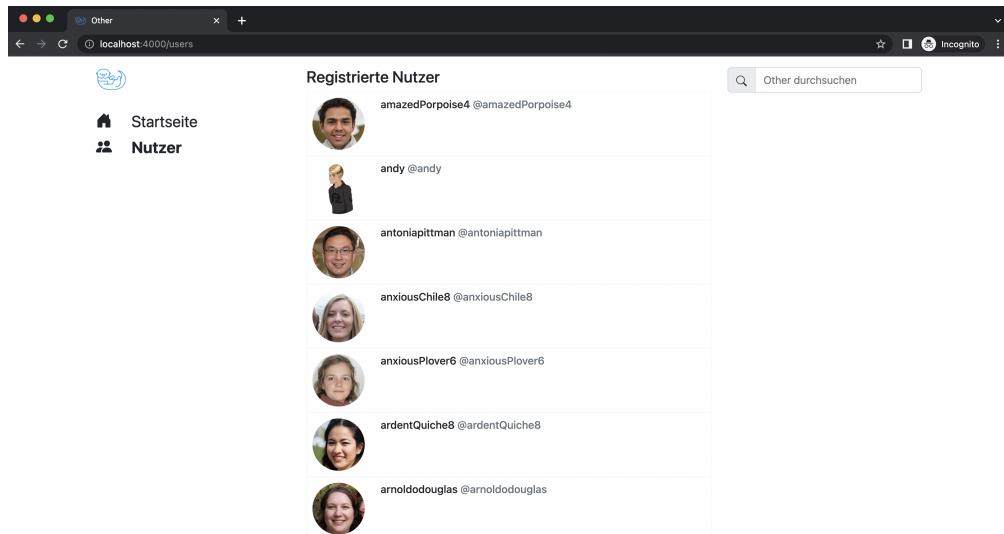


Abbildung 7: Release 1 - Nutzerübersicht

Tipp: Die Formatierung des Datums können Sie in JavaScript mithilfe des Objekts **Date** umsetzen. Vgl. dazu https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date.

Release 2 - Login und Logout

Für die Landingpage, die OTHer bei ausgeloggtem Zustand anzeigen soll, benötigen Sie eine neue Route `/`. Ergänzen Sie auch die Formulare für Registrierung (Route `/register`) und

den Login (Route `login`). Nach Registrierung werden die User zur Loginseite weitergeleitet, nach erfolgreichem Login zum Dashboard. Das Dashboard ist ab jetzt nur für eingeloggte User sichtbar.

Ergänzen Sie ebenfalls die notwendige Funktionalität für die Profilseite und die angepasste Nutzerübersicht.

Zusätzlich benötigt das Dasboard jetzt eine Funktion zum Ausloggen. Erstellen Sie dazu die entsprechende Route `/logout`.

Hinweis: Zur Eingabe des Geburtsdatums bei der Registrierung kann Ihnen ein HTML Input-Element vom Typ `date` helfen: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/date>.

Die folgenden Screenshots zeigen ein mögliches User Interface für das Release 2:

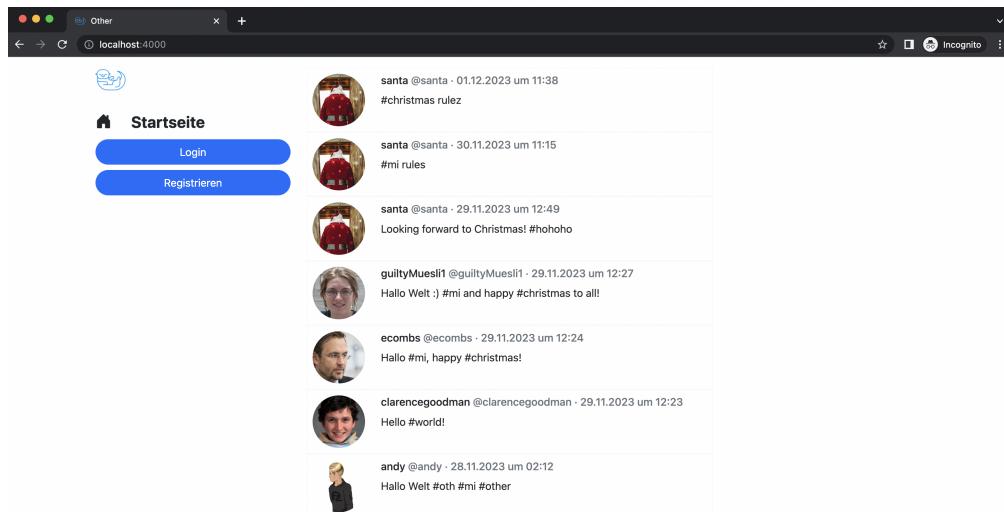


Abbildung 8: Release 2 - Landingpage

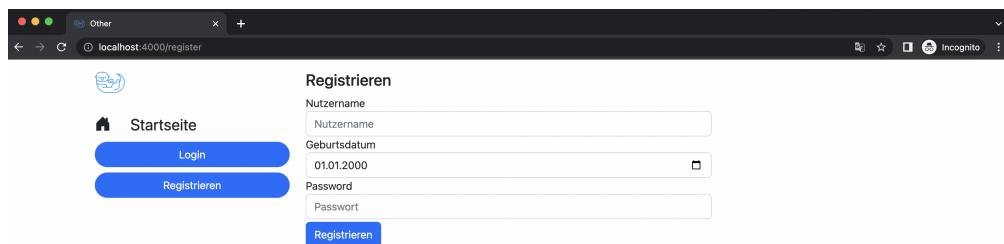
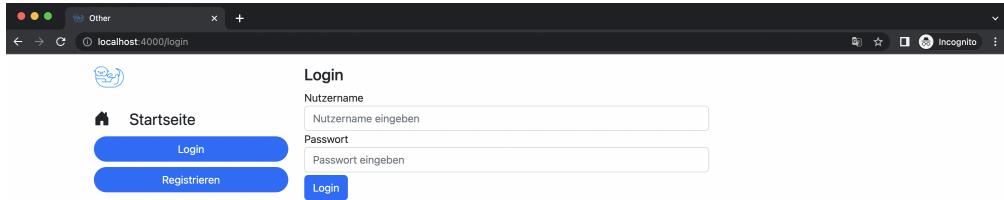
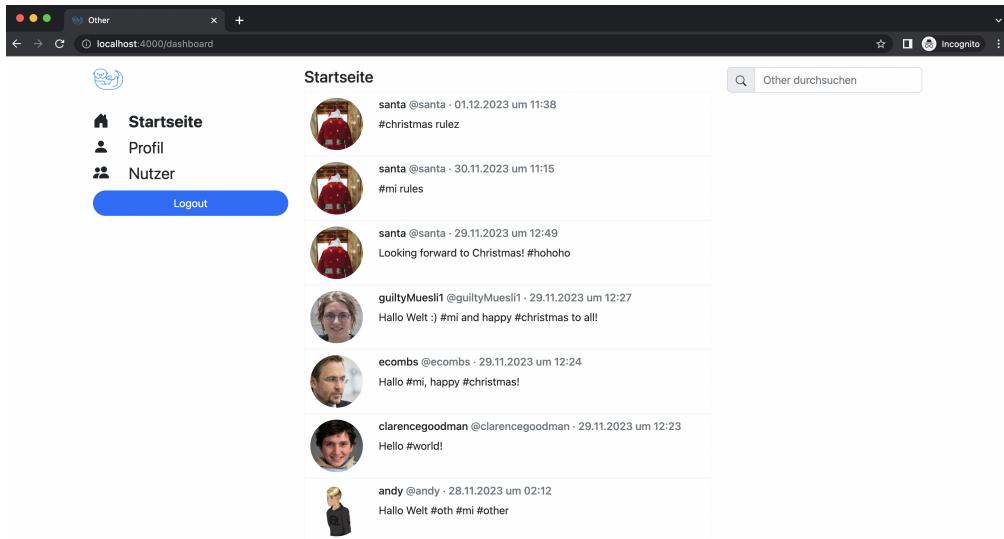


Abbildung 9: Release 2 - Registrierung



The screenshot shows a login form with a blue header bar. The main content area has a light gray background. On the left, there is a sidebar with icons for Home, Profile, and User, and buttons for 'Logout' and 'Login'. The right side contains a 'Login' section with fields for 'Username' and 'Password', and a 'Login' button.

Abbildung 10: Release 2 - Login



The screenshot shows a dashboard page with a blue header bar. The main content area has a light gray background. On the left, there is a sidebar with icons for Home, Profile, and User, and a 'Logout' button. The right side displays a list of recent posts from users like santa, guiltyMuesli1, ecombs, clarencegoodman, and andy, each with a timestamp and a short message.

User	Message	Timestamp
santa	#christmas rulez	01.12.2023 um 11:38
santa	#mi rules	30.11.2023 um 11:15
santa	Looking forward to Christmas! #hohoho	29.11.2023 um 12:49
guiltyMuesli1	Hello Welt :) #mi and happy #christmas to all!	29.11.2023 um 12:27
ecombs	Hello #mi, happy #christmas!	29.11.2023 um 12:24
clarencegoodman	Hello #world!	29.11.2023 um 12:23
andy	Hello Welt #oth #mi #other	28.11.2023 um 02:12

Abbildung 11: Release 2 - Dashboard

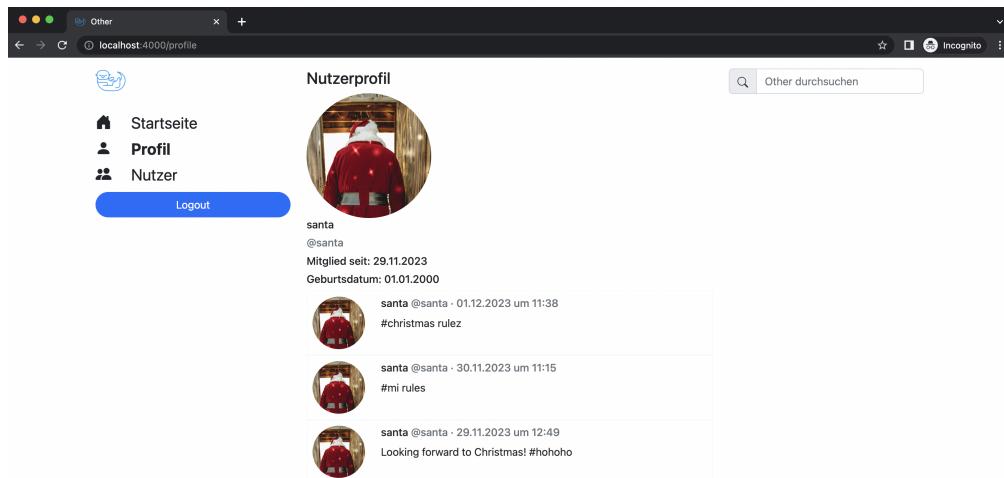


Abbildung 12: Release 2 - Profilseite (Achtung: Der Text zur Biographie ist nicht im Screenshot ersichtlich. Überlegen Sie sich selbst eine geeignete Position.)

Release 3 - Othern

Ergänzen Sie die Funktionalität eigene Others zu erstellen. Die erstellten Others sollen nach der Erstellung direkt angezeigt werden. Außerdem soll das Profil die insgesamt erstellten Others eines Users anzeigen.

Die folgenden Screenshots zeigen ein mögliches User Interface für das Release 3:

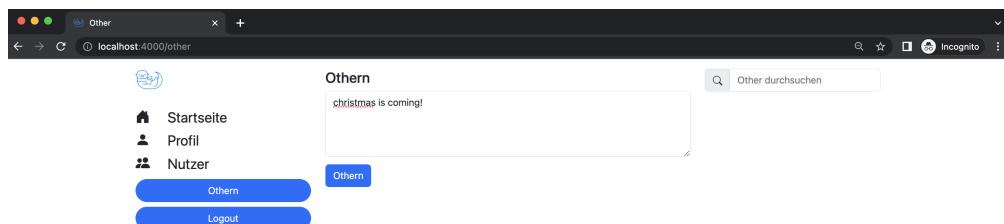


Abbildung 13: Release 3 - Neuen Other erstellen

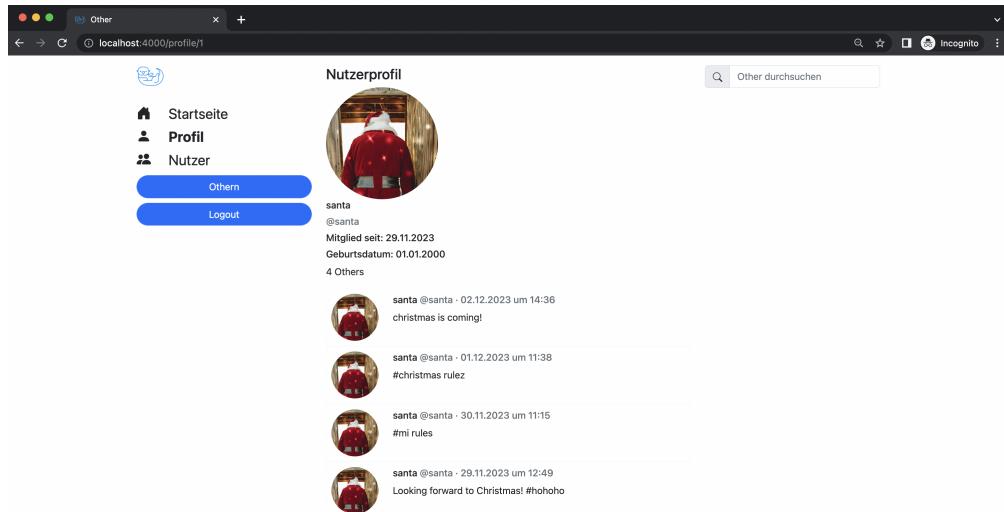


Abbildung 14: Release 3 - Aktualisiertes Profil mit neuem Other und Anzahl der erstellten Others eines Users

Release 4 - Folgen und Entfolgen

In diesem Release sollen Sie die Funktionalität ergänzen, einem anderen User zu folgen. Dazu benötigen Sie eine neue Datenbanktabelle. Diese können Sie mit dem folgenden Befehl erstellen:

```

1 CREATE TABLE follows (
2     follower INT NOT NULL REFERENCES users ,
3     followee INT NOT NULL REFERENCES users ,
4     created TIMESTAMPTZ NOT NULL DEFAULT NOW() ,
5     PRIMARY KEY(follower, followee)
6 );

```

Die Tabelle **follows** setzt Follower und Followee durch die beiden Fremdschlüssele (follower und followee) in Beziehung. Wenn Sie jetzt einen neuen Eintrag in dieser Tabelle erstellen, bedeutet es, dass der User **follower**, dem User **followee** folgt. Das Attribut **created** muss bei einem **INSERT** nicht angegeben werden und erhält immer den Zeitpunkt des Einfügens als Wert. Die Zeile **PRIMARY KEY ...** sorgt dafür, dass die Datenbank sicherstellt, dass Sie nur eindeutige Kombinationen aus **follower** und **followee** eintragen können. Versuchen Sie eine bereits existierende *Folgebeziehung* in die Tabelle einzutragen, wird dies durch die Datenbank verhindert.

Die Anzeige der Nutzerseite mit den jeweiligen Buttons für Folgen und Entfolgen ist etwas knifflig umzusetzen.

Sie benötigen eine Übersicht aller User, die pro User anzeigt, ob der eingeloggte diesem User bereits folgt oder nicht.

Diese können Sie beispielsweise mit der folgenden SQL-Abfrage erzeugen:

```

1 SELECT name, EXISTS (
2   SELECT 1 FROM follows
3   WHERE follower = 1
4   AND followee = users.user_id
5 )
6 FROM users
7 WHERE user_id != 1;

```

Diese Abfrage holt sich die Namen aller User aus der Usertabelle, außer dem eingeloggten User (in diesem Fall mit der **user_id 1**). Um in diesem Ergebnis jetzt feststellen zu können, ob man einem User bereits folgt oder nicht, wird eine Subanfrage (**EXISTS**) ergänzt: Dies prüft für jeden Eintrag aus **users**, ob es der eingeloggte User bereits dem anderen User folgt. Ist das der Fall (d.h. es gibt einen Eintrag in **follows**), gibt **EXISTS** den Wert **true** zurück, andernfalls **false** (d.h. es gibt keinen Eintrag in **follows**).

Die folgenden Screenshots zeigen ein mögliches User Interface für das Release 4:

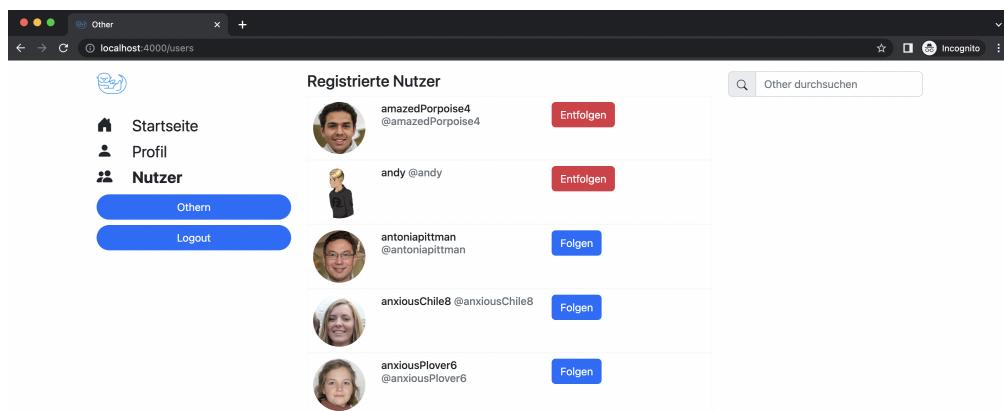


Abbildung 15: Release 4 - Aktualisierte Nutzerseite mit der Möglichkeit Usern zu *followen* und diese zu *entfollowen*

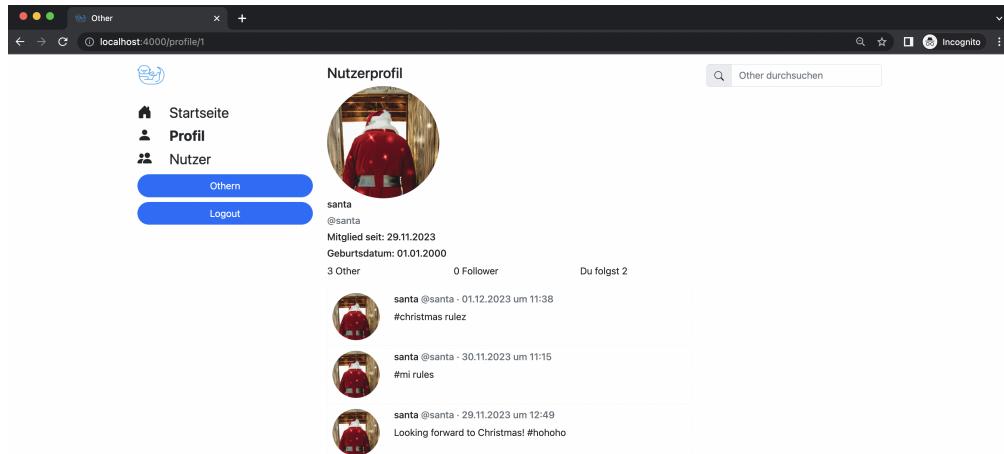


Abbildung 16: Release 4 - Angepasstes Profil mit Anzahl der *Follower* und *Followees*

Release 5 - Suche

Die folgenden Screenshots zeigen ein mögliches User Interface für das Release 5:

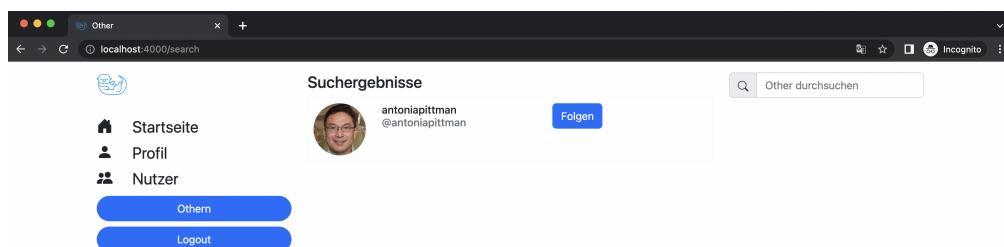


Abbildung 17: Release 5 - Beispielergebnisse für eine Suche nach *ant* - Santa ist der einge-loggte User und damit nicht in den Suchergebnissen enthalten

Hinweis: Eine Suche mit LIKE können Sie mit dem folgenden Platzhalter für das SQL-Query umsetzen:

```
const values = [`${title}`];
```

Release 6 - Trends und Profilbild aktualisieren

In diesem Release setzen Sie eine Anzeige der trendenden Hashtags um. Hierzu werden alle Other nach Hashtags durchsucht und eine Rangliste der häufigsten Hashtags ermittelt. Sie müssen dafür keine neuen Tabellen in der Datenbank anlegen.

Für die Änderung des Profilbild sollen Sie sich selbst in die Dokumentation einlesen. Die beiden folgenden Links können Ihnen helfen:

- <https://www.npmjs.com/package/express-fileupload>
- <https://wiki.selfhtml.org/wiki/HTML/Tutorials/Formulare/input/Datei-Upload>

Die folgenden Screenshots zeigen ein mögliches User Interface für das Release 6:

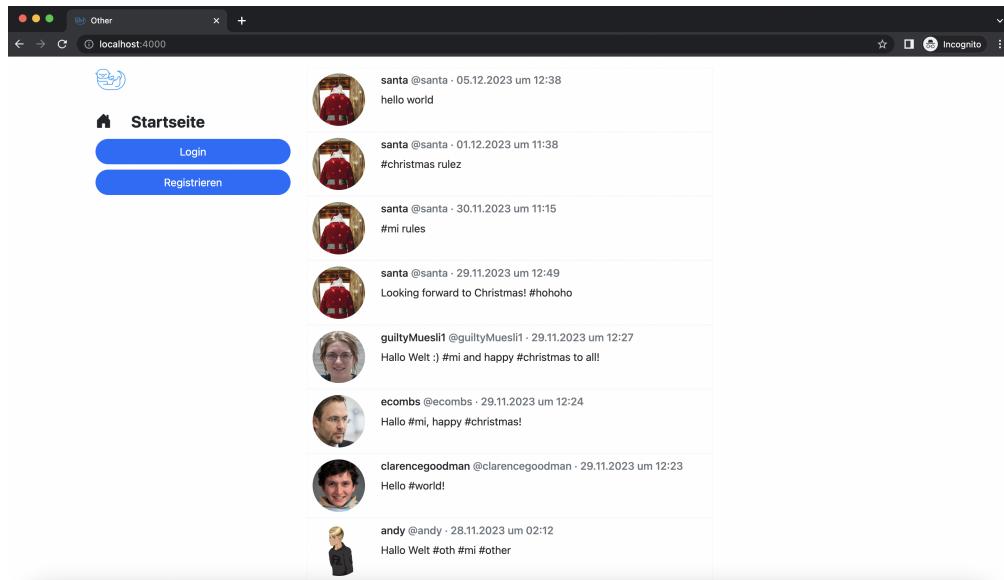


Abbildung 18: Release 6 - Übersicht aller Other zur Verdeutlichung der Trends

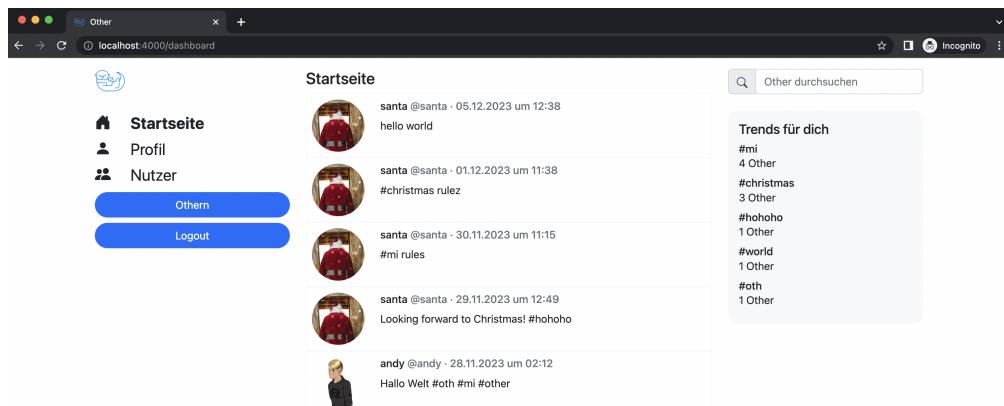


Abbildung 19: Release 6 - Anzeige der Trends

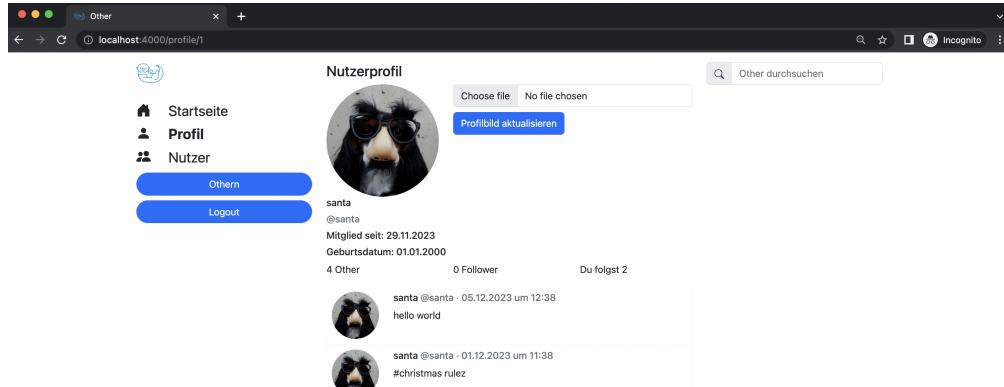


Abbildung 20: Release 6 - Geändertes Profilbild

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. Die folgende Liste ist ein Auswahl möglicher Tools:
 - Zoom-Meeting und Aufnahme starten
 - Quicktime (Mac, sollte bereits installiert sein)
 - OBS Studio (Mac / Windows)
 - Evtl. Xbox Game Bar (Tastenkombination Windows + G) (nicht getestet)
 a) Video erstellen

 b) Video über Dropbox hochladen (Sie benötigen keinen Dropbox-Account (!)):

<https://www.dropbox.com/request/Ds5T1guo41cib6fahL25>. Nennen Sie das Video **Other** und geben Ihren echten Namen, sowie Ihre studentische Mailadresse beim Upload an.
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 *commit*ten und zu *push*en.

Mögliche Erweiterungen

Die Möglichkeiten OTHer zu erweitern sind grenzenlos, hier ein paar Ideen, die natürlich alle optional sind:

- Ermöglichen Sie das Löschen eines Others.
- Hashtags einfärben und mit Funktionalität hinterlegen - Färben Sie die Hashtags in den Others blau ein und machen Sie diese klickbar. Ein Klick auf einen Hashtag führt zu einer Seite, die nur Others mit dem entsprechenden Hashtag enthält.
- Ergänzen Sie eine Validierung der Registrierung per Mail: Ihre Webanwendung verschickt eine Mail nach Registrierung. Ein Login ist erst möglich, wenn die User auf einen Bestätigungslink in der Mail geklickt haben.
- Implementieren Sie eine Like-Funktion, bei der User andere Others liken können.
- Vorschläge für Nutzer - Ergänzen Sie eine Vorschlagsliste der neu in OTHer registrierten Nutzer in der Seitenleiste, denen User folgen können.
- Ermöglichen Sie die Integration von Bildern oder Videos in Othern.
- Ergänzen Sie eine Möglichkeit andere User in Other zu erwähnen. Other werden so klickbar und man gelangt auf das Profil der Nutzer, die erwähnt wurden.
- Ergänzen Sie eine Kommentarfunktion für OTHer.
- Ergänzen Sie die Möglichkeit von Re-Others.
- Passwörter sollten nicht unverschlüsselt in der Datenbank abgelegt werden. Ergänzen Sie eine Passwortverschlüsselung. Eventuell kann Ihnen das npm-Paket **bcrypt** behilflich sein.
- und und und

Testen

Testen Sie Ihre Website ausführlich, z.B.

- Testen Sie ob das Folgen oder Entfolgen auch die entsprechenden Auswirkungen auf die entsprechende Datenbanktabelle hat.
- Testen Sie, ob die richtigen Others angezeigt werden, insbesondere, wenn Sie das Feature anderen Usern zu folgen implementiert haben.
- Rufen Sie Routen auf, die nur für eingeloggte User sichtbar sein sollen.