

Abschlussprojekt

Alexander Bazo

Abschlussprojekt WhatsApp, Goethe?

Im Rahmen Ihres Abschlussprojektes implementieren Sie eine Webanwendung, die der Erfassung und Darstellung von Briefwechseln Johann Wolfgang von Goethes mit verschiedenen anderen Personen dient. Zur Implementierung des Projekts setzen Sie die im Kurs erworbenen Fähigkeiten und die dort besprochenen Technologien ein. Das Projekt muss bis zum 31. August abgegeben werden. Die Abgabe kann zu einem früheren Zeitpunkt erfolgen. In dem Fall müssen Sie schriftlich bestätigen, dass Sie auf die verbleibende Bearbeitungszeit verzichten.

Anforderungen

Das wesentliche Ziel des Projektes ist es, die bereitgestellten Briefwechsel durch den Benutzer in das System einzupflegen und im Anschluss für diesen – aufbereitet – zugänglich zu machen. Die zu implementierende Anwendung erfüllt dazu drei Funktionen:

1. Auf einer Startseite wird den Nutzer*innen eine Übersicht über den Datenbestand präsentiert. Hier werden die Autor*innen und Adressaten der Briefe vorgestellt und relevante quantitative Informationen, wie z.B. die Anzahl der Briefe und die Zeitspanne der Briefwechsel, aufgeführt. Die Darstellung erfolgt auf graphisch und text-basiert und wird durch Visualisierungen (auf Basis von *Chart.js*) unterstützt.
2. Von der Startseite wechseln die Nutzer*innen zu eigentlichen Sammlung der Briefwechsel. Dieser werden, sortiert nach Briefpartner*innen, in Form einer an moderne Chat-Programme angelegten Darstellung zugänglich gemacht. Die Darstellung innerhalb eines ausgewählten Briefwechsels erfolgt chronologisch und unterscheidet visuelle zwischen Autor*in und Empfänger*in.
3. Über geeignete Schaltflächen können die Nutzer*innen sowohl neue Briefpartner*innen als auch neue Briefe ergänzen, die dann dauerhaft in der Datenbank gespeichert und in der Anwendung angezeigt werden.

Aufgabenstellung

Erstellen der Datenbank: Ihre erste Aufgabe ist das Importieren der als JSON-Dateien vorliegenden Briefwechsel in eine *SQLite*-Datenbank. Diese dient als Datengrundlage für die im zweiten Teil implementierte Webanwendung. Nutzen Sie für die Umsetzung des Import-Vorgangs die im Ordner `importer` vorgegebenen Strukturen aus dem Starterpaket. Das dort bereits vorhanden, und von Ihnen zu erweiternde, *Node.js*-Skript, muss beim jedem Ausführen eine neue *SQLite*-Datenbank mit den vorgegebenen Briefen erzeugen. Sie können den Skript über

die Eingabe von `npm start` auf der Kommandozeile (aus dem Unterordner `importer` heraus) starten.

Implementieren der Webanwendung: Der zweite Teil des Projekts ist die eigentliche Webanwendung, in die Sie die Datenbank aus Teil 1 einbinden und anschließend die in den Anforderungen gelisteten Funktionalitäten umsetzen. Denken Sie daran, ggf. in der Datei `Database.js` den Pfad zur erstellten Datenbankdatei anzupassen. Implementieren Sie Ihre Lösung auf Basis der Vorgaben im Ordner `app` des Starterpakets. Sie können Server und Client der Anwendung über die Eingabe von `npm start` auf der Kommandozeile (aus dem Unterordner `app` heraus) starten.

Vorgaben

Datengrundlage

Die Datenbasis umfasst Briefe, die im Rahmen unterschiedlicher Briefwechsel zwischen Goethe und anderen Personen entstanden sind. Ein Briefwechsel findet immer zwischen exakt zwei Personen – Goethe und einer anderen Person – statt und besteht aus einer beliebigen, chronologisch sortierten Anzahl von Briefen an bzw. von den beteiligten Personen. Innerhalb der Anwendung wird eine Person durch Name, Vorname sowie einen kurzen Beschreibungstext repräsentiert. Ein einzelner Brief wird durch den eigentlichen Text, einen Empfänger, einen Absender sowie dem Absendedatum und -ort repräsentiert.

Starterpaket (Anwendung)

Im *Starterpaket* finden Sie ein angefangenes Projekt, welches bereits die grobe Ordner- und Dateistruktur der Anwendung vorgibt, inkl. des *Importers*.

Verwenden Sie das Ihnen zur Verfügung gestellte Starter-Paket, um die Webanwendung (Teil 2 des Projekts) zu implementieren. Das Starter-Paket enthält die Ihnen bekannten node.js Module (`index.js`, `Database.js`) zur Realisierung des Webserver sowie Module, die für die clientseitigen Funktionalitäten der Anwendung zuständig sind (`app.js`, `FetchHelper.js`). Nutzen sie die bestehenden Module und entwickeln sie diese weiter, und erweitern Sie die Anwendung darüber hinaus auch um zusätzliche Module, um Ihre Applikation modularisiert aufzubauen.

Über die bereits existierende Klasse `FetchHelper` können Sie vordefinierte Anfragen (GET/POST) an Ihren Webserver senden. Es bleibt Ihnen allerdings offen, ob Sie die Client-seitige Kommunikation mit dem Webserver eigenständig implementieren und/oder die bereits bestehende Klasse verwenden und ggfs. abändern und so auf Ihre Bedürfnisse zuschneiden.

Die zu persistierenden Daten werden in einer SQLite-Datenbank gespeichert. Zur Gestaltung der Benutzeroberfläche verwenden Sie HTML & CSS. Die graphische Aufbereitung auf der Übersichtsseite erfolgt mit der Javascript-Bibliothek Chart.js [<http://www.chartjs.org/>].

Starten der Anwendung

Um die Anwendung korrekt auszuführen, wird ein lokaler Webserver benötigt, welcher direkt über die *Node.js*-Umgebung gestartet werden kann. Installieren Sie *Node.js* über die entsprechende Installationsdatei für Ihr Betriebssystem, die Sie [hier](#) herunterladen können. Öffnen Sie dann den Projektordner in *Visual Studio Code* und Starten Sie das *integrierte Terminal*. Führen Sie dort den Befehl `npm install` aus, um das Projekt vorzubereiten. Dannach können Sie über die

Eingabe des Befehls **npm start** den Server starten und den Client im Browser über die Adresse **http://localhost:8080/** aufrufen. Wenn Sie im integrierten Terminal die Tastenkombination **STRG + C** drücken und die Eingabe bestätigen, wird der Server beendet.

Skizzen der Benutzeroberfläche

Die folgenden Skizzen beschreiben die wesentlichen Bestandteile der Benutzeroberfläche. Bei der Implementierung der Anwendung müssen Sie sich an diesen Vorgaben orientieren. Bitte beachten Sie dabei jedoch, dass die Skizzen lediglich die Struktur und den Inhalt der Benutzeroberfläche beschreiben, nicht aber deren finales oder vollständiges Aussehen.

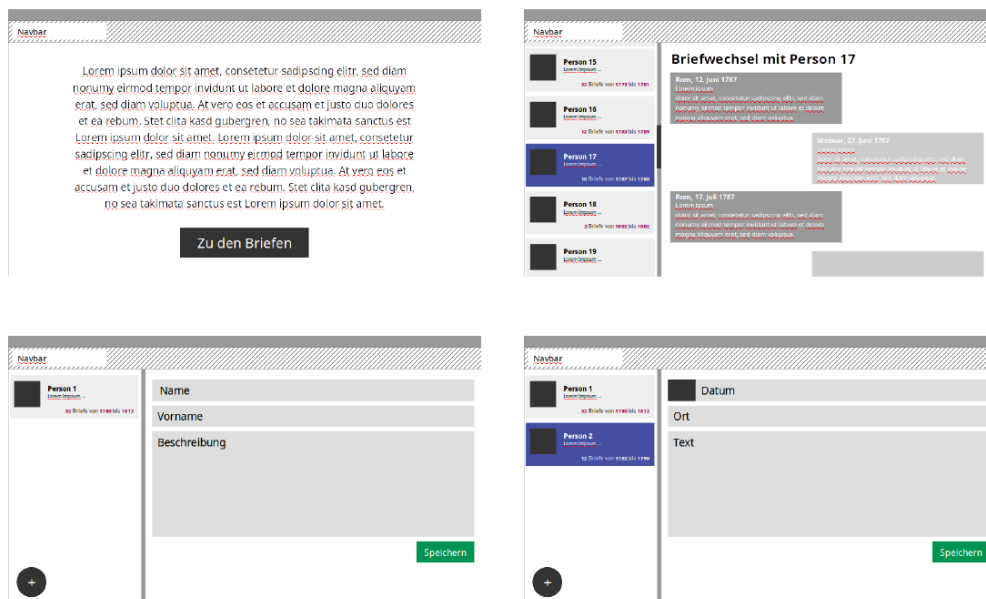


Figure 1: Startseite

Abgabekriterien

Stellen Sie Ihr gesamtes Projekt bis spätestens 31.08.2022 (23:59 Uhr) als zip-komprimierten Ordner per E-Mail an Alexander Bazo (alexander.bazo@ur.de) bereit. Bitte reichen Sie bei der Abgabe die erstellte und befüllte Datenbank mit ein.

Der Name der Zip-Datei ergibt sich aus dem Präfix **Projekt_WT_SS22**, ihrem Vor- und Nachnamen jeweils getrennt durch **_**.

Beispiel: **Projekt_WT_SS22_Max_Mustermann.zip**

Bewertungskriterien

Wesentliches Kriterium zur Bewertung Ihres Projektes ist die Umsetzung aller beschriebenen Funktionen. Ihre Anwendung muss für die definierten Anforderungen einen ernst gemeinten, erkennbaren Lösungsvorschlag beinhalten. Zusätzlich bewerten wir den Aufwand und die Qualität, die bei der Umsetzung der Funktionen erkennbar sind. Dazu gehört die Verwendbarkeit, Fehlertoleranz und ästhetische Qualität der Benutzeroberfläche sowie die Qualität (Struktur) des Datenbankschemas.

Wir wünschen Ihnen viel Erfolg!