Web-Entwicklung
Hausarbeit im Wintersemester 2023/24

Aufgabenstellung

Entwickeln Sie gemeinsam mit einem Team-Partner eine Web-Anwendung, die zur Suche nach gefälligen Vornamen eingesetzt werden kann. Die Anwendung besteht aus einem Server, der die dazu nötigen Ressourcen persistent speichert und auf Anfrage zur Verfügung stellt, sowie einer Browser-Anwendung, die dem Nutzer auf Grundlage dieser Ressourcen die folgenden Funktionalitäten bietet:

Ein Nutzer soll in der Lage sein,

- eine lexikografisch aufsteigend sortierte Liste aller bekannten Namen zu durchsuchen
  - optional gefiltert nach Geschlecht (m/w)
  - optional beginnend oder nicht beginnend mit Präfix
  - optional endend oder nicht endend mit Suffix
  - optional gefiltert nach Silbenanzahl
- gefällige Namen von dieser Liste auf einen persönlichen Merkzettel zu übertragen
- die Namen auf dem persönlichen Merkzettel
  - optional gefiltert nach Geschlecht (m/w) anzuzeigen
  - zu priorisieren
  - wieder zu entfernen

Achten Sie darauf, dass in der Browser-Anwendung die Liste der bekannten Namen paginiert wird, d.h. es sollen immer nur eine konstante von Ihnen wählbare Anzahl von Namen beim Server angefragt und auf einmal angezeigt werden. Über entsprechende Schaltflächen soll zwischen den Seiten der Liste gewechselt werden können. Der aktuelle Seitenindex sowie die Gesamtanzahl der Seiten sollen angezeigt werden. Werden bei einer Suche mehrere der o.g. Filter eingesetzt, sind diese Und-verknüpft.

Ein mithilfe von Node.js und Express realisierter HTTP-Server soll einerseits die Browser-Anwendung als statische Dateien an den Browser ausliefern, andererseits die Persistierung der o.g. Ressourcen (bekannte Namen, Namen auf dem persönlichen Merkzettel) in einer Datenbank sowie die Auslieferung der Ressourcen über eine REST-konforme HTTP-Schnittstelle zur Verfügung stellen.

Zur Persistierung der Ressourcen haben Sie die Wahl zwischen einer MongoDB- oder SQLite-Datenbank. Implementieren Sie ein von der Web-Anwendung unabhängiges Node.js-Skript, welches die initiale Befüllung der Datenbank mit den bekannten Namen aus der beigefügten CSV-Datei[1] übernimmt. Setzen Sie das npm-Modul syllabificate[2] ein, um die Anzahl der Silben für jeden Namen zu bestimmen.

Sie müssen weder clientseitig noch serverseitig eine Benutzerverwaltung oder Authentifizierung umsetzen – Jeder darf alle Ressourcen erstellen und einsehen und in der Datenbank muss nur ein einziger Merkzettel persistiert werden.

---

[1] Gesamt_Vornamen_Koeln_2010_2022_cleaned.csv

[2] https://www.npmjs.com/package/syllabificate

Weitere Anforderungen sind:

- Ihre Anwendung muss zumindest in aktuellen Versionen von Google Chrome und Mozilla Firefox funktionsfähig sein.
- Sie dürfen **keinen** JavaScript-Präprozessor (z.B. TypeScript) einsetzen.
- Sie dürfen **kein** MV*-Framework (z.B. Angular, React oder Vue.js) oder jQuery einsetzen.
- Es ist Ihnen freigestellt, ob Sie ein CSS-Framework (z.B. UIKit) einsetzen oder nicht.
- Achten Sie auf eine sinnvolle Ordnerstruktur und möglichst kleinteilige Modularisierung Ihres Projekts mithilfe von ESM-Modulen und esbuild.
- Sie dürfen neben ggf. explizit genannten auch weitere npm-Module einsetzen, insofern diese in den Build-Prozess (s.u.) eingebunden sind.
- Ihr Code darf auf Grundlage der semistandard-Regeln keine Fehler aufweisen. Anderenfalls führt dies bei der Prüfung automatisch zu einer Abwertung um einen Notenschritt.

Studienleistung

Voraussetzung zur Zulassung zur Prüfung ist der Nachweis der Studienleistung. Der Erwerb der Studienleistung setzt die Anmeldung zur Studienleistung im QIS sowie die Mitteilung der Team-Zusammensetzung (via E-Mail oder per Listeneintrag in den Präsenzveranstaltungen) voraus.

Um die Studienleistung zu erbringen, muss ein Grundgerüst der Hausarbeit erstellt und fristgerecht abgegeben werden. Dieses Grundgerüst muss über folgende Komponenten verfügen und die folgenden Anforderungen erfüllen:

- Die clientseitige Browser-Anwendung
  - referenziert **eine einzige CSS-Datei**, die aus einer oder mehreren Less-Datei(en) mithilfe von lessc erzeugt wird,
  - referenziert **eine einzige JS-Datei**, die das Ergebnis des Bundlings aller Abhängigkeiten mithilfe von esbuild ist.
- Die serverseitige Node.js-Anwendung
  - startet mittels Express einen HTTP-Server an einem Port, der als Kommandozeilenargument übergeben werden kann,
  - liefert die clientseitige Anwendung als statische Dateien an den Browser aus.
- Einen npm-Build-Prozess,
  - der in einer **sh-kompatiblen Shell** ausgeführt werden kann,
  - der durch Aufruf von `npm run initdb` das Skript zur initialen Befüllung der Datenbank ausführt,
  - der durch Aufruf von `npm run clean` das Projekt bereinigt, d.h. **alle** Dateien löscht, welche durch den Build-Prozess heruntergeladen oder generiert wurden,
  - der durch Aufruf von `npm run lint` **alle** JS-Dateien im Projekt (sowohl Client als auch Server) mithilfe von semistandard überprüft,
  - der durch Aufruf von `npm run debug` das gesamte Projekt erzeugt,
  - der durch Aufruf von `npm run build` ebenfalls das gesamte Projekt erzeugt und dabei die CSS- und JS-Datei mithilfe von less-plugin-clean-css bzw. terser minifiziert,
  - der das Erzeugen des Projekts mit `npm run debug` oder `npm run build` abbricht, falls semistandard Fehler aufdeckt,
  - der durch Aufruf von `npm run start` oder `npm start` den HTTP-Server an Port 8080 startet.

*1 Die Browser-Anwendung muss noch keinerlei Funktionalität bieten und auch das Skript zur initialen Befüllung der Datenbank muss noch nicht implementiert sein.

Zum erstmaligen Starten der Anwendung an Port 8080 muss also lediglich `npm install && npm run initdb && npm run build && npm start` ausgeführt werden. **Bricht dieser Prozess ab oder ist das Ausführen der Anwendung im Browser anschließend nicht möglich, gilt die Studienleistung als *Nicht bestanden*.**

Der Bearbeitungszeitraum für die Studienleistung beginnt mit Veröffentlichung dieser Aufgabenstellung und endet am 02.01.2024. Die Abgabe muss erfolgen bis zum **02.01.2024 0:00 Uhr[3] MEZ**.

Prüfung

Der Teilnahme an der Prüfung setzt die Anmeldung zur Prüfung im QIS, die Mitteilung der Team-Zusammensetzung (via E-Mail oder per Listeneintrag in den Präsenzveranstaltungen) sowie den Erwerb der Studienleistung voraus.

Der minimale Funktionsumfang der finalen Abgabe ergibt sich aus der zu Beginn dargestellten Aufgabenstellung sowie den Anforderungen in den Abschnitten *Aufgabenstellung* und *Studienleistung*.

Hinsichtlich der Gestaltung sowie der Nutzerinteraktionsmechanismen gibt es keine weiteren Vorgaben oder Einschränkungen.

Der Bearbeitungszeitraum für die Prüfung beginnt ebenfalls mit Veröffentlichung dieser Aufgabenstellung und endet am 26.02.2024. Die Abgabe muss erfolgen bis zum **26.02.2024 0:00 Uhr[3] MEZ**.

Abgabe

Die Abgabe (sowohl der Studienleistung als auch der Prüfung) erfolgt via Stud.IP in den entsprechend benannten Ordner der Lehrveranstaltung. Sie umfasst das **bereinigte Projekt als ZIP-Datei (nicht als RAR-Datei o.ä.)**. Versehen Sie den Dateinamen mit Ihren **lesbaren Namen**, um eine eindeutige Zuordnung zu ermöglichen.

Eine fehlende, unvollständige oder verspätete Abgabe führt zur Gesamtbewertung der Studienleistung bzw. der Prüfung mit *Nicht bestanden*. Maßgeblich ist jeweils die Zeitangabe in Stud.IP.

[3] https://de.wikipedia.org/wiki/24-Stunden-Z%C3%A4hlung