

Web-Entwicklung

Hausarbeit im Wintersemester 2021/22

Aufgabenstellung

Entwickeln Sie eine Web-Anwendung zur Verwaltung eines Kinos. Die Anwendung besteht aus einem Server, der die dazu nötigen Ressourcen persistent speichert und auf Anfrage zur Verfügung stellt, sowie einer Browser-Anwendung, die auf Grundlage dieser Ressourcen die folgenden Funktionalitäten für Betreiber und Kunden des Kinos bietet:

Ein Betreiber soll in der Lage sein,

- Kinosäle anzulegen
- Vorstellungen anzulegen

Ein Kunde soll in der Lage sein,

- Reservierungen von einem freien Sitzplatz oder mehreren freien Sitzplätzen für eine Vorstellung anzulegen
- QR-Codes auszudrucken, der beim Besuch von Vorstellungen als Bestätigung einer Reservierung dienen kann

Das nachträgliche Bearbeiten sowie Löschen von Kinosälen, Vorstellungen und Reservierungen ist nicht vorgesehen.

Ein Kinosaal verfügt mindestens über die Eigenschaften

- Name
- Anzahl Sitzreihen
- Anzahl Sitze pro Sitzreihe

Eine Vorstellung verfügt mindestens über die Eigenschaften

- Datum und Uhrzeit des Vorstellungsbeginnns
- Kinosaal
- Name des Films

Eine Reservierung verfügt mindestens über die Eigenschaften

- Vorstellung
- Reservierte Sitzplätze
- Name des Kunden

Ein Nutzer soll auszuwählen können, ob er Betreiber oder Kunde ist. Die Browser-Anwendung soll anschließend nur die für diese Rolle verfügbaren Funktionalitäten anbieten. Sie müssen aber weder clientseitig noch serverseitig eine Benutzerverwaltung oder Authentifizierung umsetzen – Jeder darf alle Ressourcen erstellen und einsehen.

Achten Sie darauf, dass in der Browser-Anwendung Listen von Ressourcen paginiert werden, d.h. es sollen nur so viele Einträge auf einmal angezeigt werden, wie bei aktueller Fenstergröße ohne Scrollen in das Browser-Fenster hineinpassen. Über entsprechende Schaltflächen soll zwischen den Seiten der Listen gewechselt werden können. Der aktuelle Seitenindex sowie die Gesamtanzahl der Seiten sollen angezeigt werden. Wird die Größe des Browser-Fensters durch den Nutzer verändert, sollen die Listen neu paginiert werden.

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 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. Falls Sie SQLite einsetzen, muss die Initialisierung der Datenbankstruktur automatisch durch die Server-Anwendung erfolgen.

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 ES6+-Sprachkonstrukte einsetzen oder nicht.
- Es ist Ihnen freigestellt, ob Sie ein CSS-Framework (z.B. UIKit oder Spectre.css) einsetzen oder nicht.
- Achten Sie auf eine sinnvolle Ordnerstruktur und möglichst kleinteilige Modularisierung Ihres Projekts mithilfe von CommonJS-Modulen und browserify.
- 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 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) erzeugt wird,
 - referenziert **eine einzige JS-Datei**, die das Ergebnis des Bundlings aller Abhängigkeiten mithilfe von browserify ist,
 - bietet dem Nutzer die Möglichkeit anzugeben, ob er Betreiber oder Kunde ist.
- Die serverseitige Node.js-Anwendung
 - startet 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 sowohl auf unixoiden Betriebssystemen als auch unter Microsoft Windows in einer **sh-kompatiblen Shell** ausgeführt werden kann,
 - 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 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.

Zum erstmaligen Starten der Anwendung an Port 8080 muss also lediglich `npm install && 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 28.12.2021. Die Abgabe muss erfolgen bis zum **28.12.2021 0:00 Uhr MEZ**.

Prüfung

Der Teilnahme an der Prüfung setzt die Anmeldung zur Prüfung im QIS 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.

Die Bewertung der Prüfung erfolgt auf Grundlage der Abgabe sowie eines Abnahmegesprächs.

Der Bearbeitungszeitraum für die Prüfung beginnt ebenfalls mit Veröffentlichung dieser Aufgabenstellung und endet am 14.02.2021. Die Abgabe muss erfolgen bis zum **14.02.2021 0:00 Uhr MEZ**.

In der Woche ab dem 14.02.2021 finden die Abnahmegespräche statt. Genaue Termine werden individuell vereinbart. Das Abnahmegespräch findet entweder in den Räumlichkeiten der Hochschule oder in Form einer Videokonferenz statt. Halten Sie in beiden Fällen Ihren Studierendenausweis oder den Personalausweis bereit. Das Abnahmegespräch muss unabhängig von der Bewertung der Abgabe bestanden werden. Abwesenheit führt zu der Gesamtbewertung *Nicht bestanden*.

Abgabe

Die Abgabe (sowohl der Studienleistung als auch der Prüfung) erfolgt 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 **rzht-Benutzerkennungen (nicht mit Ihren Matrikelnummern)**, 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.