Praktikum Web-Programmierung

Aufgabe 4

1 Themenbereiche

Clientseitige Webanwendungen

- DOM-Manipulation
- Ereignisbehandlung
- Asynchrone Anfragen
- JSON

Web-Komponenten

- HTML-Templates und Includes
- Shadow-DOM

2 Aufgabenstellung

In dieser Aufgabe sollen Sie Ihren Web-Shop um ein paar nützliche, clientseitige Komponenten erweitern, die insgesamt das Kundenerlebnis verbessern sollen.

2.1 Echtzeit Suchvorschläge

Eine oft anzutreffende Eigenschaft moderner Webseiten ist die, dass in Suchfeldern bereits Vorschläge zur Vervollständigung der Suchanfragen angezeigt werden, während der Nutzer oder die Nutzerin noch die Anfrage eingibt. Dies wird in der Regel dadurch realisiert, dass der Web-Browser die Eingabeereignisse abfängt und dann die aktuelle Eingabe als asynchrone Anfrage [6] an einen Web-Dienst sendet, welcher mit möglichen Vervollständigungen der Suchanfrage antwortet. Verfahren, die versuchen, bestmögliche Vorschläge zu liefern, können durchaus sehr komplex werden.

Für den Prototypen des Web-Shops sollen Sie daher nur eine sehr einfache Variante der Vorschlagsliste implementieren. Sobald der Nutzer oder die Nutzerin anfängt, Eingaben in das Suchfeld einzugeben, soll eine asynchrone Anfrage an den Webserver mit den aktuellen Inhalten des Suchfelds gesendet werden. Die Suche selbst kann nach wie vor über eine einfache SQL “LIKE” Abfrage durchgeführt werden [3]. Die ersten fünf Ergebnisse der Abfrage sollen dann in einer Popup-Box unterhalb des Suchfeldes zur Auswahl dargestellt werden.

Wählt der Nutzer oder die Nutzerin der Webseite eine der Möglichkeiten aus oder wird die Eingabe der Suchanfrage mit der Eingabetaste beendet, soll die ganz reguläre Suche durchgeführt werden.

2.1.1 Such-Web-Service

Um die Echtzeit-Suchvorschlagsliste zu implementieren, müssen Sie einen Web-Service bereitstellen. Dieser soll an einer von Ihnen gewählten URL Suchanfragen verarbeiten und die entsprechenden Ergebnisse zur Verfügung stellen. Als Austauschdatenformat sollen Sie JSON nutzen [7]. Wie Ihre konkreten Anfrageparameter und Ergebnisstrukturen aufgebaut sind, bleibt dabei Ihnen überlassen. Der Suchservice sollte REST-konform sein [2].

Es bietet sich im Rahmen dieser Aufgabe an, den Such-Service als PHP-Skript zu implementieren. Als Alternative können Sie den Dienst aber auch in JavaScript mit Node.js umsetzen [5]. Die nötigen Um- und Weiterleitungen, damit der Dienst dann auch letzendlich mit der Shop-Webseite funktioniert, müssen Sie dann natürlich ebenfalls implementieren und bereitstellen.

2.2 Ein Warenkorb-Popup als Web-Komponente

Als zweites “Feature” der Shop-Seite sollen Sie eine Popup-View implementieren, die immer dann angezeigt wird, wenn der Nutzer oder die Nutzerin mit der Maus über den Warenkorb-Link fährt.

Innerhalb der View soll der aktuelle Inhalt des Warenkorbs angezeigt werden. Es genügt, wenn die Ansicht aus einer einfachen Textliste mit Einträgen für den Inhalt besteht. Da so eine Liste möglicherweise auch für andere Zwecke eingesetzt werden kann, sollen Sie diese Anzeige als wiederverwendbare Web-Komponente umsetzen [1].

Dazu müssen Sie also ein entsprechendes HTML-Template erzeugen und dafür sorgen, dass das Design der Komponente innerhalb ihres eigenen Shadow-DOMs gekapselt ist. Den Namen der Komponente können Sie selbst wählen. Ebenfalls können Sie selbst entscheiden, ob Sie das Befüllen der Liste durch die Angabe von Kindelementen der Komponente umsetzen oder durch die Bereitstellung und den Aufruf entsprechender JavaScript-APIs.

2.3 Hinweise

- Überlegen Sie sich vor dem Beginn der Implementierung, welche Anfragen der Such-Service verarbeiten und wie der Datenaustausch zwischen Web-Service und JavaScript im Client aussehen soll.

- Beachten Sie, dass der Web-Service *keine* HTML-Seite zurückliefert.
- Verwenden Sie für asynchrone Anfragen aus dem Browser am besten die neuere **fetch**-API [4].
- Schicken Sie zu viele asynchrone Anfragen zu schnell hintereinander an den Web-Service, entstehen Ihnen möglicherweise Probleme beim Verarbeiten der Anfragen und deren Antworten. Versuchen Sie also, dies zu vermeiden.
- Wenn Sie Ihre Web-Komponente mit einer API ausstatten wollen, achten Sie darauf, dass Sie den Code für die Initialisierung nicht in einer IIFE kapseln.
- Das Verwenden externer Bibliotheken und/oder Frameworks ist nicht erlaubt.

3 Bewertungskriterien

Für diese Aufgabe können Sie maximal 20 Punkte erhalten, sofern die folgenden Kriterien erfüllt sind:

8 Punkte	Die Webseite erfüllt alle funktionalen Anforderungen der Aufgabenstellung.
2 Punkte	Der Quellcode ist klar strukturiert, sowie sinnvoll und ausreichend kommentiert. Die HTML-Dokumente enthalten, außer in begründeten Ausnahmefällen, keine nichtsemantischen Elemente.
4 Punkte	Die Lösung der Aufgabe wird im Rahmen eines Vortrages präsentiert. Der Vortrag ist sinnvoll strukturiert und enthält alle wesentlichen Inhalte.
6 Punkte	Während des Vortrages ist erkennbar, daß sich alle Gruppenmitglieder mit den fachlichen Inhalten auseinandergesetzt haben und ein Verständnis für die Themengebiete der Aufgabe entwickelt haben.

Literatur

[1] The World Wide Web Consortium (W3C). *Web Components Specifications*. Webseite. 2018. URL: https://github.com/w3c/webcomponents/. Abgerufen am 03.06.2018.

[2] The World Wide Web Consortium (W3C). *Web Services Architecture*. Webseite. 2004. URL: https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/. Abgerufen am 03.06.2018.

[3] Oracle Corporation. *MySQL 8.0 Reference Manual: Pattern Matching*. Webseite. 2018. URL: https://dev.mysql.com/doc/refman/8.0/en/pattern-matching.html. Abgerufen am 24.04.2018.

[4] Mozilla MDN web docs. *Fetch API*. Webseite. 2018. URL: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API. Abgerufen am 03.06.2018.

[5] The Node.js Foundation. *Node.js*. Webseite. 2018. URL: https://nodejs.org/en/. Abgerufen am 03.06.2018.

[6] Jesse James Garrett. *Ajax: A New Approach to Web Applications*. Webseite. 2005. URL: http://adaptivepath.org/ideas/ajax-new-approach-web-applications/. Abgerufen am 03.06.2018.

[7] ECMA International. *ECMA-404 - The JSON Data Interchange Syntax*. Webseite. 2017. URL: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf. Abgerufen am 03.06.2018.