

Praktikumsbericht

Joshua Coelho Mestre

2. September 2019

Inhaltsverzeichnis

1	Einleitung	3
2	Kalenderwoche 23	4
3	Kalenderwoche 24	5
4	Kalenderwoche 25	6
5	Kalenderwoche 26	7
6	Kalenderwoche 27	8
7	Kalenderwoche 28	8
8	Kalenderwoche 29	8
9	Kalenderwoche 30	8
10	Kalenderwoche 31	8
11	Kalenderwoche 32	8
12	Kalenderwoche 34	8
13	Fazit	8

1 Einleitung

Scholz & Volkmer ist eine Kreativagentur zur digitalen Markenführung mit ungefähr 130 Mitarbeitern und Firmensitz in Wiesbaden und Berlin.

Die Agentur arbeitet im Kundenauftrag und erstellt Websites, Mobile Apps und weitere digitale Produkte zur Verbesserung der digitalen Präsenz.

In meiner Zeit bei Schovo arbeitete ich vorwiegend als Frontend Entwickler für den Kunden Adidas, der neben seiner Corporate Seite auch einen Blog bei Schovo entwickeln lässt.

Während meines Praktikums war ich für die Umsetzung eines umfangreichen A/B-Tests der auf dem oben erwähnten Blog namens "Gameplan A" durchgeführt werden sollte.

2 Kalenderwoche 23

Das Praktikum begann mit einem großen Meeting bei dem ich die Gelegenheit bekam mich der ganzen Firma vorzustellen.

Darauf folgte der Erhalt meines Arbeitsrechners in der IT-Abteilung sowie das treffen mit meiner Praktikumsbetreuerin.

Meine Praktikumsbetreuerin zeigte mir meinen Arbeitsplatz und gab mir eine Einführung in die firmeneigene Softwarestruktur und Abläufe. Darunter zum Beispiel den zum Dateien-Austausch genutzten Fileserver, das häufig genutzte Ticketsystem Jira und das für den Informationsaustausch genutzte Confluence.

Ich machte mich danach mit den eben genannten Systemen vertraut und richtete meinen Arbeitsrechner ein.

Am folgenden Tag durfte ich mich mit dem Code der firmeneigenen Website vertraut machen. Dazu klonete ich das, das sich auf dem Gitlab-Server befindliche Projekt, und lies es mittels Docker-Container auf meinem Arbeitsrechner laufen.

Das Setup des Projekts erfolgte mittels im Repository enthaltener Scripte.

Da geplant war mich als Backend-Developer einzusetzen und es sich bei der S-V.de Webseite um ein Django Projekt handelt, absolvierte ich den Rest der Woche ein detailliertes Django Tutorial um meine im Studium erhaltenen Django-Kenntnisse aufzufrischen und zu verbessern.

Das Ziel des Tutorials war es, abgesehen von der Verbesserung meiner Django Kenntnisse, eine Website zu erstellen, die es den Usern erlaubt Links zu posten und diese einer Bestimmten Category zuzuordnen.

Zusätzlich dazu sollte es den Usern ermöglicht werden direkt auf der Seite das Internet nach neuen Seiten anhand von Suchbegriffen zu durchforsten und bereits vorhandene Links zu liken".

Das Tutorial befasste sich mit allen relevanten Gesichtspunkten der Webentwicklung und vermittelte zusätzlich zu dem Basiswissen der Django-Programmierung auch Inhalte wie

die Nutzung von CSS-Frameworks wie Bootstrap und die Nutzung externer APIs um die Funktionalität der Webseite um eine Websuche zu erweitern.

3 Kalenderwoche 24

Zu Beginn der zweiten Woche beschäftigte ich mich mit einem auf der s-v.de auftretenden Problem.

Im Falle einer Unerreichbarkeit der CDN-Server auf denen die Medien der Seite Abgelegt sind sollten diese aus dem Cache des Django Caches geladen werden, was jedoch nicht zuverlässig geschah.

Ich beschäftigte mich zunächst mit der lokalen Reproduktion des Fehlers um ihn dann schrittweise zu lokalisieren. Auf Grund der Projektlage wurde ich jedoch von dem s-v.de-Projekt abgezogen und auf die Adidas Corporate Seite angesetzt.

Der Kunde wünschte sich ein Update für das sich auf der Seite befindliche Medien-Archiv.

Das für die Filterung der Medienformate benutzte Dropdown-Menü, welches es dem Nutzer ermöglicht aus Audio, Video und Bilder zu wählen, sollte durch drei nutzerfreundlichere Checkboxes ersetzt werden.

Nach dem Klonen und lokalen Einrichten des Projekts bestand meine Arbeit zunächst daraus mich mit dem Projekt und dem verwendeten, firmeneigenen Javascript Framework, vertraut zu machen.

Nach meiner Orientierung begann ich mit der Umsetzung des Updates, indem ich zunächst das unerwünschte Dropdown-Menü aus dem betroffenen HTML-Template entfernte und das Markup für die geforderten Checkboxes inklusive des, von der Konzepterin geforderten, Stylings hinzufügte.

Der nächste Schritt beinhaltete Implementierung der Filter-Funktionalität der Checkboxes, wobei sich das Hauptproblem darin ergab, dass statt wie im Falle des Dropdown nun nicht mehr nur eine Kategorie Medien ausgewählt werden kann sondern mehrere oder sogar gar keine Checkbox ausgewählt sein kann.

4 Kalenderwoche 25

Um das Problem der Mehrfachfilterung der Medien durch die Checkboxes zu lösen habe ich mich intensiv mit dem bereits vorhandenen Filter-Modul beschäftigt.

Die Anforderung war das bereits existierende Modul mit möglichst geringer Kopplung und möglichst minimal inversiv dahingehend zu verändern, dass es mit der Eingabe über die Checkboxes die gewünschten Medien einblendet.

Da das Filter-Modul mehrfach auf der Seite benutzt wird und bis dahin ausschließlich über die Dropdown-Menüs gesteuert wurde stand ich vor der Herausforderung die Funktionalität des Moduls um die Checkbox zu erweitern jedoch das verarbeiten der Eingaben durch die Dropdowns nicht zu behindern.

Um der Anforderung gerecht zu werden habe ich innerhalb des Moduls eine Funktion implementiert, die im Falle einer Eingabe durch die Checkboxes aufgerufen wird und diese soweit verarbeitet, dass sie danach von wie gewohnt von dem Filter weiter genutzt werden kann.

Durch die selbst implementierte Funktion musste ich anschließend nur noch geringfügige Anpassungen an dem Modul vornehmen, da das nun nicht mehr nur noch ein Filter aktiv sein kann sondern mehrere.

Die Änderung beschränkte sich auf die Erweiterung des Moduls um die Möglichkeit durch ein Filter-Array zu iterieren statt nur ein einziges Filterwort zu akzeptieren.

Nachdem ich den Task beendet hatte ließ ich meinen Code von einem Senior Developer reviewen, der mich auf einige Sachen hinwies, die ich nicht bedacht hatte.

Unter anderem hatte ich Funktionen benutzt, die von Internet Explorer 11 nicht unterstützt werden. Dies ist vor allem problematisch, da viele Rechnerau Seiten des Kunden diesen Browser noch benutzen.

Ich behob die durch die Code-Review entdeckten Fehler und nutzte die Gelegenheit um noch weitere kleine Verbesserungen vorzunehmen, die ich durch mein verbessertes Verständnis für Javascript und das benutzte Framework entdeckt habe.

5 Kalenderwoche 26

Da ich mich nun in der Projekt Adidas Corporate Seite, und insbesondere in das Medien-Archiv, eingearbeitet hatte übernahm ich weitere Tickets diesbezüglich.

Ich kümmerte mich in dieser Woche um einen bekannten Bug, der zur Auswirkung hatte, dass wenn man sich das Medien-Archiv im apple-eigenen Browser, Safari, anschauen wollte, keine Bilder zu sehen waren.

Ich begann damit den Code zu debuggen und mir parallel dazu sowohl die für die Bilder benutzten Templates, als auch das für das laden der Bilder verantwortliche Javascript-Modul zu Gemüte zu führen.

Mir fiel beim Debuggen auf, dass wenn ich das loadedEvent für die Bilder händisch abfeuerte, alle Bilder korrekt angezeigt wurden.

Dieses Verhalten wies darauf hin, dass die Bilder bereits alle geladen waren, jedoch auf Grund des fehlenden Events "hidden"blieben. Dies entsprach dem Code den ich im Bilder-Modul vorfand, wo mit einem Eventlistener auf das Laden der Bilder gewartet wurde um dann alle gleichzeitig anzeigen zu können.

Das Laden der Bilder übernimmt ein Lazyloader, welcher das src-Attribut der Images erst setzt, wenn sie benötigt werden, sodass unnötige clientseitige Operationen und Request gespart werden können.

Die Vermutung lag nahe, dass das Event gefeuert wird noch bevor das Bilder-Modul den Eventlistener anmelden kann.

Durch weiteres Debugging fiel schlussendlich auf, dass bereits im Template das src-Attribut gesetzt wurde anstatt die Quelle in das data-src-Attribut zu schreiben, was zum laden der Bilder führte noch bevor das Javascript initialisiert war wodurch nicht auf das Event reagiert werden konnte, da es nicht gefeuert wurde, da durch das bereits gesetzte Attribut der Lazyloader übergangen wurde.

Eine Korrektur des Fehlers im Template eliminierte den Bug und ermöglichte eine Korrekte anzeige in allen gängigen Browsern.

6 Kalenderwoche 27

7 Kalenderwoche 28

8 Kalenderwoche 29

9 Kalenderwoche 30

10 Kalenderwoche 31

11 Kalenderwoche 32

12 Kalenderwoche 34

13 Fazit

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

Online-Quellen