

## Dokumentation der Projektarbeit zur Erstellung einer interaktiven Webseite

Gruppe: B5

Mitglieder: Robert Witzke, Marten Kaffler, Mohammed Hamid Omar, Nico Laschner, Dennis Podkolsin, Marcus Koppelman

Die interaktive Webseite kann als ein Prototyp für eine Reiseplattform verstanden werden, wobei man mehrere Städte mit Buchungsoption zur Auswahl hat. Wenn man über den localhost:8080 auf die Webseite zugreift, wird einem zu Beginn die Login-Seite angezeigt, insofern man sich vorher noch nicht angemeldet hat bzw. nicht angemeldet ist. Anschließend gelangt man, nach Eingabe des Benutzernamens, auf die Startseite und kann dann über verschiedene Aktionen mit der Webseite interagieren. Bei der Anmeldung des Nutzers wird im backend unter `app.post(„/login“, ...)` ein Cookie „username“ erstellt, der den Anmeldenamen als Wert trägt. Somit kann der Browser auch geschlossen werden und die Daten bleiben dennoch bestehen.

Auf der Startseite befinden sich die Städte Barcelona, New York, Kapstadt und Tokio, einen Bereich für Kommentare und einige Links in dem Header der Seite. Klickt man auf die einzelnen Städte, wird man auf die jeweilige Seite der Stadt weitergeleitet und erhält deutlich mehr Details und Informationen. Bei jedem Klick auf eine Stadt wird im Backend die Funktion `app.post(„/count“, ...)` aufgerufen. Dabei wird von jedem der Nutzer (in dem Falle Cassidy, Bryn und Kim) gespeichert, wie oft sie eine Stadt besucht haben und dieser Wert bei jedem Klick um eins erhöht. Über die Funktion `app.get(„/countdata“, ...)` wird dann die Liste mit den Daten zu den jeweiligen Aufrufen eines Nutzers zurückgegeben und diese Daten werden von der Funktion `addCounterListItem()` gefetcht. Die Logik ist dabei so aufgebaut, dass nur Städte als relevant betrachtet werden, die überdurchschnittlich viel aufgerufen wurden (meistbesuchte Inhalte). Für diese Städte werden anschließend in der Funktion `addCounterListItem()` Listeneinträge erstellt, wobei der Aufruf der Funktion in der `overview.html` stattfindet, da sich dort auch die Liste befindet.

Um die Funktionalität der Kommentare zu gewährleisten, gibt es im backend die Datenstruktur *comments*, welche alle Kommentare des jeweiligen Nutzers beinhaltet. Sobald ein Kommentar verfasst und abgesendet wird, wird dieser dem aktuell angemeldeten Nutzer zugeordnet. Um den Kommentar anzuzeigen, gibt es, wie auch bei den meistbesuchten Inhalten, die Funktion `app.get(„/commentsdata“, ...)` und eine Funktion namens `addComment()`, die davon Daten fetcht. `app.get(„/commentsdata“, ...)` gibt die Liste von Kommentaren des aktuell angemeldeten Nutzers zurück. Nur wenn der Nutzer „Bryn“ angemeldet ist, dann werden alle Kommentare aller Nutzer zusammengefügt und zurückgegeben. In der Funktion `addComment()` werden die Daten dann gefetcht und anschließend wieder in einer Liste gespeichert, wofür einzelne Listeneinträge erstellt werden.

Bei den Favoriten ist das Prinzip ähnlich. Nutzer können auf die einzelnen Seiten der Städte navigieren und dort auf den Button „Favorit“ klicken. Dadurch wird im Backend die Route `/favorit` aufgerufen, welche den Namen der Stadt in der Favoritenliste *data* dem jeweiligen Nutzer zuordnet und die Liste der Favoriten im Cookie „favorites“ speichert. Auf der Startseite, also der `overview.html`, gibt es dann wieder einen Bereich (`<div>`), in dem eine Liste von Favoriten ausgegeben wird. Dafür gibt es im Backend die Funktionen `app.get(„/data“, ...)` und `addListItem()`, wobei man sich über die Route „/data“ die Liste der Favoriten des aktuell angemeldeten Nutzers holen kann. In der Funktion `addListItem()` werden diese Daten wieder gefetcht und für jeden Favoriten wird ein Listeneintrag auf der Startseite erzeugt.

Durch diese Implementierung ist ebenfalls gewährleistet, dass Bryn alle Kommentare aller Sitzungen sehen kann, jedoch aber jeder Nutzer und jede Nutzerin nur seine/ihre eigenen Favoriten und meistbesuchten Inhalte.

Die Webseite enthält zusätzlich noch einen Button zum Ausloggen, wobei im backend die Funktion `app.get(„/logout“, ...)` aufgerufen wird, die den Cookie „username“ und „favorites“ löscht.

Um das Dockerimage zu einem Dockercontainer zu bauen, müssen folgende Befehle im Verzeichnis des Dockerfiles ausgeführt werden:

```
$ docker build --no-cache . -t reiseplattform
```

--no-cache wird verwendet, um keine im Cache gespeicherte Version zu nutzen und -t, um dem Image einen Namen zu geben. Um das Dockerimage dann laufen zu lassen, wird folgender Befehl benötigt:

```
$ docker run -p 8080:8080 -d reiseplattform
```

Der Parameter -p navigiert einen öffentlichen Port auf einen privaten Port innerhalb des Docker-Containers. Der Parameter -d lässt den Container auch im Hintergrund weiterlaufen. Anschließend kann der Server über <http://localhost:8080/> erreicht werden.

### **Zur Aufgabenverteilung in der Gruppe:**

Die HTML-Seiten inklusive des Stylings wurden in mehreren gemeinsamen Meetings erstellt. Die Javascript-Funktionalitäten sowie serverseitige Anfragen wurden in der Gruppe wie folgt aufgeteilt:

1. Login mit cookie-Erstellung und Zuweisung im Backend hat Marcus Koppelman übernommen.
2. Die Darstellung der meistbesuchten Inhalte im Frontend haben Dennis Podkolsin und Nico Laschner übernommen.
3. Die Darstellung der Favoriten und Kommentare im Frontend hat Mohammad Hamid Omar erledigt.
4. Die Zuweisung der meistbesuchten Inhalte, Favoriten und Kommentare zu dem/der jeweilige(n) Nutzer\*in haben Marten Kaffler und Robert Witzke gemacht.
5. Den Prozess für die Erstellung eines Docker-Images und des Docker-Containers haben Marcus Koppelman und Nico Laschner übernommen.
6. Die Dokumentation haben Robert Witzke und Marten Kaffler geschrieben.