

# Web-Programmierung, 4-3131-TIT20B

Kurs B,

Gruppe 1

Mitglieder: Oskar Droescher, Kai Arne Möbert, Mario Schenkewitz, Florian Schoon, Jos Trautermann

## Projektarbeit Dokumentation

### Anforderungen:

#### Installation:

- Wenn Kim die docker CLI oder docker compose CLI Befehle exakt Schritt für Schritt ausführt wie in der Beschreibung, dann wird die Webanwendung erfolgreich gestartet.
- Wenn Kim die URL <http://localhost:8080/> aufruft, dann wird Kim die Startseite der Webanwendung geladen und angezeigt.

#### Aufrufen:

- Wenn Cassidy die URL <http://localhost:8080/> aufruft, oder Cassidy die URL <http://localhost:8080/index.html> aufruft, dann wird Cassidy die Startseite der Webanwendung geladen und angezeigt.
- Wenn die Anwendung vollständig geladen wurde, dann wird ihr ein Merkmal für die Personalisierung zugeordnet.
- Wenn die Anwendung vollständig geladen wurde, dann wird Cassidy's Anwesenheit durch ein Merkmal auch nach dem Schließen des Browsers wiedererkannt.
- Wenn die Startseite vollständig geladen wurde, dann werden Cassidy ihre meistbesuchten Inhalte angezeigt.

#### Inhaltsseiten:

- Wenn eine Inhaltsseite vollständig geladen wurde, dann kann Cassidy Kommentare zu Inhalten zu erstellen und der Kommentar wird auf dem Backend ihrer Sitzung zugeordnet.
- Wenn eine Inhaltsseite vollständig geladen wurde, dann kann Cassidy Inhalte als Favorit zu markieren und die Markierung wird auf dem Backend ihrer Sitzung zugeordnet.
- Wenn eine Inhaltsseite vollständig geladen wurde, dann werden Cassidy ihre favorisierten Inhalte anhand ihres Personalisierungsmerkmals angezeigt.

#### Öffentliche Infos:

- Wenn die Anwendung vollständig geladen wurde, dann werden Bryn alle Kommentare aller Sitzungen angezeigt.
- Wenn die Anwendung vollständig geladen wurde, dann werden Bryn keine Favoriten angezeigt. (erster Aufruf)

#### Notizen:

- Es wird kein Neustart des Servers durchgeführt. Es ist keine Persistenz notwendig.

## Starten der App

Die App kann mit Hilfe von Docker gestartet werden.

### Dockerfile (Implementierung: Mario Schenkewitz, Jos Trautermann)

Um aus diesem Dockerfile ein Image zu erstellen, muss zuerst das Dockerfile heruntergeladen werden. Docker Engine (bzw. Docker Desktop) muss gestartet sein. Danach kann über eine Kommandozeile zum Dockerfile navigiert werden (Ordner in dem das Dockerfile liegt) und mit dem Befehl **docker build --no-cache -t gruppe\_b1\_projektarbeit .** das Image erstellt werden.

Das Image kann mit **docker run -p 8080:8080 gruppe\_b1\_projektarbeit** oder über Docker Desktop gestartet werden. Die Seite kann jetzt über <http://localhost:8080/> erreicht werden.

Achtung: In Docker Desktop muss beim Start in den optionalen Einstellungen noch der localhost Port 8080 angegeben werden.

## Merkmal für Personalisierung (Implementierung: Mario Schenkewitz, Kai Arne Möbert)

Beim Besuchen der verschiedenen Seiten, wird die Funktion „userCheck“ ausgeführt. Wenn kein Cookie mit dem Namen „username“ existiert, dann wird ein Cookie mit einer generierten ID erstellt, diese wird genutzt, um die verschiedenen Benutzer zu identifizieren. Außerdem wird ein Objekt vom Typ „user“ erstellt, es hat die Attribute id, name, kommentare[], favoriten[], funfacts, catwatch und katzenspielzeug. Dieses neue Objekt mit der generierten ID wird in eine Liste „userlist“ eingetragen.

## Kommentare (Implementierung: Mario Schenkewitz, Oskar Droescher)

Kommentare werden über die Funktionen „addComment“ und „getComment“ realisiert.

„addComment“ erhält per POST Anfrage „name“ und „kommentar“, der name ist der gewählt Name des Benutzers bei der Eingabe und kann bei jedem Kommentar geändert werden, um Anonymität zu gewährleisten. Der Kommentar wird in der Form „Name: Kommentar“ in einem Array gespeichert, das im Objekt „comments“ enthalten ist.

Der Kommentar wird zusätzlich in das entsprechende user-Objekt in der „userlist“ gespeichert, um Kommentare der Session zuzuordnen.

Wenn eine Inhaltsseite aufgerufen wird (funfacts, catwatch, katzenspielzeuge), dann wird per AJAX über GET Anfrage der Inhalt des entsprechenden Kommentar-Arrays abgerufen und als Liste auf der Seite dargestellt.

## Favoriten (Implementierung: Mario Schenkewitz, Florian Schoon)

Auf den Inhaltsseiten (funfacts, catwatch, katzenspielzeuge), befindet sich eine Schaltfläche, um die Seite seinen Favoriten hinzuzufügen. Dafür wird der Schaltfläche ein „eventListener“ für „click“ hinzugefügt (siehe ./src/scripts/addfav.js). Bei Betätigen der Schaltfläche wird eine Anfrage per POST an den Server geschickt, dieser speichert die Seite im entsprechenden user-Objekt in der „userlist“ und sendet als Antwort den Text der Schaltfläche, der bei erfolgreicher Antwort geändert wird. Aufrufen der Startseite (index) sucht in der „userlist“ nach dem entsprechenden user-Objekt und stellt dann alle Einträge des „favoriten“ Arrays dar. Falls das Array eine Länge von 0 hat, wird ein Platzhaltertext angezeigt.

## Meistbesuchte Inhalte (Implementierung: Mario Schenkewitz)

Wie beim Merkmal für Personalisierung wird auch bei den meistbesuchten Inhalten eine Funktion beim Besuchen der verschiedenen Seiten ausgeführt. Die Funktion „addVisit“ überprüft die derzeitige URL, wenn diese mit einer der Inhaltsseiten übereinstimmt (funfacts, catwatch, katzenspielzeuge), dann wird der Integer wert im entsprechenden user-Objekt um 1 erhöht. Besuchen der Startseite ruft wieder analog die Werte aus dem user-Objekt ab, fügt sie formatiert einem Array hinzu und stellt die Werte dar.

## Arbeitsaufteilung

1) Siehe unterhalb der Überschriften, die rot markierten Texte.

2) HTML-Seiten und Styling:

<b>Bearbeiter</b>	:	<b>HTML-Seite &amp; Styling</b>
Mario Schenkewitz	:	Home, Funfacts
Oskar Droescher	:	Catwatch
Kai Arne Möbert	:	Katzenspielzeug
Florian Schoon	:	Unser Team

3) Jos Trautermann hat die Seiten auf die korrekte Funktionsweise getestet und ggf. Verbesserungen vorgeschlagen.