Web-Programmierung, 4-3131-TIT20B

Kurs B

Gruppe 1

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

# Aufgabe 3 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 meist besuchten 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 Inhalten 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

Um aus diesem Dockerfile ein Image zu erstellen muss das zuerst Dockerfile und die Projektseite heruntergeladen werden. Danach kann über eine Kommandozeile zum Dockerfile navigiert werden und zum Beispiel mit dem Befehl **$ docker build -t gruppe\_b1\_projektarbeit .** das Image erstellt werden.   
Das Image kann mit **$ docker run –p 8080:8080 gruppe\_b1\_projektarbeit** oder über Docker Desktopgestartet werden. Achtung in Docker Desktop muss beim Start in den optionalen Einstellungen noch der localhost Port 8080 eingegeben werden.

# Merkmal für Personalisierung

Wenn man die Webseite besucht, 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.

# Meist besuchte Inhalte

# Kommentare

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

„addComment“ erhält per POST Anfrage „username“ und „kommentar“, der username ist der gewählt Name des Benutzers, dieser wird auch nachträglich auf vergangene Kommentare angewandt, der Name kann also geändert werden, aber nur für alle Kommentare.  
BenutzerIDs und die Namen werden als Key-Value-Paar in einem Objekt gespeichert, Kommentare selber werden in einem Objekt gespeichert, dass als Attribute die verschiedenen Inhaltsseiten hat. Diese Attribute sind Arrays in denen die Kommentare als String gespeichert werden. Die Strings haben die Form BenutzerID: Kommentar.

Wenn nun eine Inhaltsseite Aufgerufen wird, dann wird per AJAX über GET Anfrage der Inhalt des entsprechenden Kommentar Arrays aufgerufen, die BenutzerIDs ausgetauscht und dann in einer Liste auf der Seite angezeigt.

# Favoriten