

## **Projektdokumentation**

### **LAGOM**

Unsere Gruppe besteht aus Thao Nguyen, Lars Obist und Yuliya Litvin und wir haben das Projekt mit dem Namen "Lagom" entwickelt.

In diesem Projekt handelt es sich um einen digitalen Bullet Journal, der jedem Nutzer ein individuelles Workspace bietet, in dem dieser sich sein Leben strukturieren kann. Somit kann er sich Erinnerungen anlegen, ToDo-Listen erstellen, ein Tagebuch führen, sich Ziele setzen und Termine in seinen Kalender eintragen.

Um unseren Nutzer zudem täglich zu motivieren, findet er bei jedem erneuten Neuladen der Seite einen anderen motivierenden Spruch auf seiner Workspace-Page und auch immer wieder drei neue Bilder in seiner Galerie.

Für die Umsetzung unseres Projekts hatten wir zunächst für jede Entität (wie z. B. Kalender usw.) eine Tabelle mit den entsprechenden Attributen mit Hilfe von der MySQL Workbench in unserer Datenbank erstellt.

Um die Inhalte in der Webanwendung darstellen zu können, haben wir Node.js und das auf TypeScript basierende Framework Angular genutzt.

### **Abweichungen von unserer konzeptionellen Idee**

Bevor wir in das tatsächliche Programmieren unseres Projekts eingestiegen sind, stand bereits fest, dass wir die Hauptseite, also das Workspace des Nutzers, in sieben unterschiedliche Bereiche aufteilen wollen, wie es in unserer Konzeption vorher festgelegt hatten.

Somit war für jeden einzelnen dieser Bereiche eine bestimmte Art der Umsetzung und Funktionalität vorgesehen, wobei einige davon sich im Laufe der Projektbearbeitung aus zeitlichen Gründen ein wenig veränderten und deshalb zum Teil anders gehandhabt wurden als wie zuvor geplant.

Beispielsweise war für den Bereich "Motivation" zuvor vorgesehen, dass sich Nutzer auch selbst ihre persönlichen Sprüche reinschreiben können. Allerdings dachten wir im Laufe des Projekts an dieser Stelle um und entschieden, dass wir dem Nutzer stattdessen unterschiedliche Motivationssprüche von unserer Seite aus anzeigen wollen, um ihn aktiv zu motivieren und Abwechslung neben seinen eigenen Einträgen zu bieten. Denn oftmals motivieren Worte eines anderen Menschen mehr als die eigenen. Dies wollten wir insofern erreichen, indem bei jedem Neuladen der Seite ein neuer Spruch im Workspace des Nutzers auftaucht.

Ähnlich wie beim Bereich "Motivation" hat der Nutzer auch bei der Galerie keine Möglichkeit eigene Bilder in diese hinzuzufügen. Hier sind wir wie auch bei der Motivation von unserer eigentlichen konzeptionellen Idee abgewichen und veränderten diese. Dementsprechend bekommt der Nutzer bei jedem Neuladen der Seite andere zufällige Bilder in seiner Galerie angezeigt.

Somit ist die Galerie auch zukünftig weiter ausbaubar und wir wollten diese zur jetzigen Abgabe - trotz der eigentlich fehlenden Hinzufügen-Option des Nutzers - veranschaulichen.

Zurzeit können weitere Bilder nur durch das Reinladen in den Assets-Ordner hinzugefügt werden, was nicht unserer eigentlichen Umsetzungsidee entspricht. Stattdessen sollte der Nutzer eigentlich die Möglichkeit haben, beliebige Bilder in seinen Workspace zum Beispiel per Drag-and-Drop hinzuzufügen und in der Galerie zu speichern. Leider fehlte uns für die Umsetzung dessen am Ende doch ein wenig die Zeit. Dennoch haben wir die Struktur und die Tabellen in unserer Datenbank beibehalten, sodass eine Weiterentwicklung unserer ursprünglichen Idee problemlos möglich wäre.

Die Einträge beim Kalender werden zurzeit im Stil der anderen Bereiche wie zum Beispiel das Tagebuch und die Erinnerung und weitere dargestellt. Demnach ist die Funktionalität auch identisch zu ähnlichen im Workspace gegebenen Bereichen.

Einen Kalender mit Monatsansicht haben wir mit Hilfe eines Frameworks in das Workspace eingebunden. Somit kann man die Termine zeitgleich neben einem Kalender (mit gewohnter Darstellung) sehen. Dies bereitet die Möglichkeit, einen Überblick zu behalten, an welchen Tagen Termine eingetragen sind. Zeitlich haben wir es jedoch nicht mehr geschafft, Termine innerhalb dieser Kalenderansicht anzeigen zu lassen, was in einer möglichen Ausarbeitung umgesetzt werden könnte.

## **Gruppenarbeit**

Die Bearbeitung des gesamten Projekts erfolgte wöchentlich gemeinsam Remote (da die Corona-Situation es zurzeit nicht anders erlaubt). Somit ist es für uns kaum möglich abzugrenzen, welches Gruppenmitglied für welchen Bereich zuständig war, da wir immer zusammen gearbeitet haben.

Für das Design und die Gestaltung des Projekts, also des Workspaces, kann aber deutlich hervorgehoben werden, dass Thao Nguyen hier die leitende Person war.

## Technische Punkte

*Welche Frameworks habt ihr eingesetzt und warum?*

Wir haben für die Entwicklung der webbasierten Datenbankanwendung das Framework Angular 9 verwendet, da dies zu Beginn des Semesters empfohlen wurde.

Für das Gestalten des generellen Designs unserer Webseite haben wir keine Frameworks eingesetzt, sondern alles selbst implementiert. Das Favicon (Icon oben im Browser haben wir passend zu unserem *Lagom - Bullet Journal* Corporate Design selbst illustriert. Dies bedeutete für uns zwar mehr Aufwand, bereitete uns jedoch die Möglichkeit komplett die Kontrolle über unser Design und das Verständnis unseres Codes beizubehalten.

Lediglich innerhalb der Component "Kalender" haben wir ein Framework verwendet, da dies das Erstellen des Kalenders bzw. dessen Monatsansicht und das Umsetzen der entsprechenden Funktionalität deutlich vereinfacht hat und man es zudem nicht neu erfinden musste. Das Design der Kalenderansicht haben wir jedoch an unsere Plattform angepasst.

*Beschreibt was ihr selber gemacht habt und was ihr von woanders übernommen habt?*

Für die Funktionalität der Webanwendung haben wir als Komplikationen oder Schwierigkeiten bei der Implementierung der Anwendung aufgetreten sind, auf den Seiten von Angular, StackOverflow und YouTube nach Lösungsansätzen gesucht und für unseren Fall spezifisch angewendet und angepasst.

Folgende Quellen wurden verwendet:

Angular-Seite und *Heroes-Beispiel*: <https://angular.io/tutorial>

Youtube: *Socket-Beispiel*: <https://www.youtube.com/watch?v=66T2A2dvplY>

*Socket-Beispiel*:

<https://medium.com/iamdeepinder/creating-a-real-time-app-with-angular-8-and-socket-io-with-nodejs-af63bd59a47f>

*Kalender-Framework/Vorlage*:

<https://ej2.syncfusion.com/angular/documentation/calendar/customization/>

*Eine Beschreibung wie eure Webanwendung auszuführen ist:*

Um unsere Anwendung verwenden zu können, müssen zunächst die nötigen Node-Modules installiert werden. Dafür muss der Befehl *npm install* in einem Terminal oder der Powershell ausgeführt werden.

Um das Projekt im Browser ausführen zu können, müssen zwei Server gestartet werden, da wir unsere Anwendung über zwei Ports (Port 8080 für den Server und Port 4200 für die Anwendung selbst) laufen lassen.

Um die Anwendung zu starten, muss es beim ersten Mal "gebildet" werden. Dafür muss im Terminal oder in der Powershell der Befehl *ng build* eingegeben und gestartet werden. Anschließend muss der Befehl *ng serve* folgend eingetragen und durchgeführt werden.

Um den Server zu starten, muss lediglich der Befehl *node server.js* in einem weiteren Terminal- oder Powershell-Fenster eingetippt und ausgeführt werden. Falls doch noch Änderungen im Code auftreten sollten, sollte der Befehl *npm run serve* angewandt werden. So werden Änderungen, die in der Datei "server.js" vorgenommen wurden, direkt erkannt und der Server muss nicht noch einmal manuell gestartet werden.

Nun kann ein beliebiger Browser (getestet wurde auf Chrome und Firefox) geöffnet werden und die URL <http://localhost:4200> eingegeben werden. Die Anwendung startet und man kann sie nun benutzen. Um das beste Nutzererlebnis zu genießen, empfiehlt es sich unsere Anwendung im Browser im Vollbildmodus zu nutzen.

Zudem haben wir ein paar Beispiel Accounts erstellt, anhand welcher wir zeigen möchten, wie ein Workspace in Nutzung beispielsweise aussehen könnte. Folgende Beispiel Accounts haben wir erstellt:

*Beispielnutzer 1:*

Nutzername: ylitvin  
Passwort: 12345

*Beispielnutzer 2:*

Nutzername: pnguyen  
Passwort: 12345

*Beispielnutzer 3:*

Nutzername: lobist  
Passwort: 12345

*Beispielnutzer 4:*

Nutzername: btannert  
Passwort: benjaminT

Das automatische Aktualisieren der Webseite lässt sich anhand von zwei offenen Tabs darstellen. Dafür muss man in beiden Tabs mit dem gleichen Nutzer-Account angemeldet sein. Dies hat den Grund, dass der Inhalt eines individuellen Workspaces nur von dem Nutzer einsehbar ist, der dieses Workspace besitzt. Daher gibt es in unserem Projekt keinen Use-Case, in dem zwei verschiedene Nutzer zur gleichen Zeit Aktualisierungen sehen müssen bzw. in unseren Fällen auch nicht können.

Wenn beide Tabs mit dem gleichen angemeldeten Nutzer-Account geöffnet und am besten nebeneinander gestellt sind, können nun Aktionen wie "Eintrag erstellen", "Eintrag bearbeiten" oder "Eintrag löschen" ausgeführt werden. Der ausgeführten Aktion entsprechend, werden in dem jeweiligen Workspace Veränderungen sichtbar.

*Wie greift man auf die Datenbank zu?*

Wir haben die von Dir bereitgestellte Datenbank für dieses Semesterprojekt genutzt. Um auf diese Datenbank zuzugreifen und sich einzelne Datensätze anzuschauen, kann man nachdem man den Server mit dem Befehl `node server.js` oder `npm run serve` gestartet hat, im Browser beispielsweise die URL <http://localhost:8080/nutzer> eingeben und sich damit die in unserer Datenbank eingetragenen Nutzer mit ihren Attributen anzeigen lassen.

Wenn Daten in der Datenbank manuell verändert werden sollen, kann dies über die MySQL Workbench getan werden. Dazu muss man den Hostnamen der Datenbank, den Port, den Benutzernamen und das Zugriffs Passwort dafür kennen, um die MySQL Workbench mit dieser Datenbank zu verbinden.