



Anforderungsspezifikation zum Projekt

ShareList

Dokumentname	T3-ShareList-Lastenheft_v1.0
Version	1.0
Datum	22.03.2022
Status	Vervollständigt
Verteiler	SCHAUER Adrian, ERMLER Raymond, FIDANCI Mirac, LASSER Fabian, GRGIC Luka, KOFLER Robert, BIERBAMER Peter

Zusammenfassung

Das Ziel des Projekts ist es eine Webseite zu erstellen bei welcher die Möglichkeit geboten wird Listen zu erstellen. Allerdings ist nicht nur das Erstellen der Listen ein Punkt der Kompetenzen welche ShareList erfüllt, sondern auch das Teilen und Rollen zuweisen für die jeweiligen Nutzer. Ganz stark im Fokus liegt das Teilen der Listen, daher auch der Name ShareList. Das Team besteht aus insgesamt 5 Entwicklern: 2 für das Frontend¹ und 3 für das Backend², wobei bei dieser Unterteilung es nicht bedeutet, dass die Backendentwickler ausschließlich am Backend arbeiten, sondern auch Aufgaben für das Voranschreiten des Frontends bewältigen und andersherum genauso.



Änderungshistorie

Version Nr.	Autor	Datum	Änderungen, Grund der Änderungen
0.1	T03-ShareList	14.03.2022	Ersterstellung (Ausgangsversion)
1.0	T03-ShareList	22.03.2022	Ausgangsversion vollendet



Inhaltsverzeichnis

1.	Beschreibung des Ist- und Sollzustandes	4
1.1.	Istzustand	4
1.2.	Veranlassung für das Projekt	4
1.3.	Sollzustand (Produktbeschreibung)	4
1.4.	Zielbestimmung	4
1.5.	Produkteinsatz	4
2.	Funktionale Anforderungen	4
2.1.	Featureliste	4
2.2.	Use-Case Beschreibung zum Gesamtsystem	5
2.3.	Produktfunktionen	5
3.	Nicht funktionale Anforderungen	9
4.	Produktdaten	10
5.	Skizze der Systemarchitektur	10
6.	Schnittstellen	10
7.	Lieferumfang	11
8.	Abnahmekriterien (optional)	11
9.	Glossar	11



1. Beschreibung des Ist- und Sollzustandes

1.1. Istzustand

Backend ist soweit fertig, es fehlt noch das Frontend sowie das UI³.

1.2. Veranlassung für das Projekt

Anlass des Projekts war unsere Aufgabe im Fach PRE eine Website zu erstellen und Projektmanagement-Basierte Aufgaben zur Website zu bearbeiten.

1.3. Sollzustand (Produktbeschreibung)

Der Sollzustand des Projekts ist die Fertigstellung des Front- und Backends. Dabei soll man Listen erstellen können, auf welche auch andere Leute zugreifen können und sofern sie die Berechtigung dafür haben, auch von diesen bearbeitet werden können.

1.4. Zielbestimmung

1.4.1. Ziele

Unser Ziel ist es, eine Website zu erstellen, mit der es einfach ist, Listen zu erstellen und teilen. Außerdem sollen Berechtigungen vergeben werden können, wer auf die Liste Zugriff hat oder die Liste bearbeiten darf. Unser Muss-Ziel ist das erfolgreiche Erstellen der Website bis zum Abgabedatum. Die Zielgruppe sind alle Menschen, die eine einfache, teilbare Liste erstellen wollen.

1.4.2. Nutzen

Mit ShareList können ganz einfach Listen für sich selbst, oder eine Gruppe erstellen. Es ist möglich, Mitglieder hinzufügen, die ebenfalls den Inhalt der Liste bearbeiten können. Oder man erlaubt anderen nur Einsicht auf die Liste.

1.5. Produkteinsatz

1.5.1. Anwendungsbereiche

Die Liste könnte beispielsweise als ToDo-Liste⁴ für ein Projekt sein, den Einkauf notieren, oder um sich Ziele zu setzen.

1.5.2. Zielgruppen

Unsere Zielgruppe sind Menschen, die sich Dinge für sich selbst, sowohl auch für andere Personen notieren wollen. Wir versuchen, unsere Website möglichst intuitiv zu gestalten, um möglichst vielen Altersgruppen und Menschen mit verschiedenen technischen Niveaus, eine Plattform zu bieten.

2. Funktionale Anforderungen

2.1. Featureliste

1: ShareList bietet die Möglichkeit mehrere Benutzer gleichzeitig an den Listen arbeiten zu lassen

2: Der Ersteller der Liste kann anderen Personen Rollen zuweisen, welche Ihnen bestimmte Rechte geben

3: Man kann nicht nur eine Liste haben, sondern mehrere, was bei der Vielfalt der bestimmten Dinge nützlich ist die man haben will

2.2. Use-Case Beschreibung zum Gesamtsystem

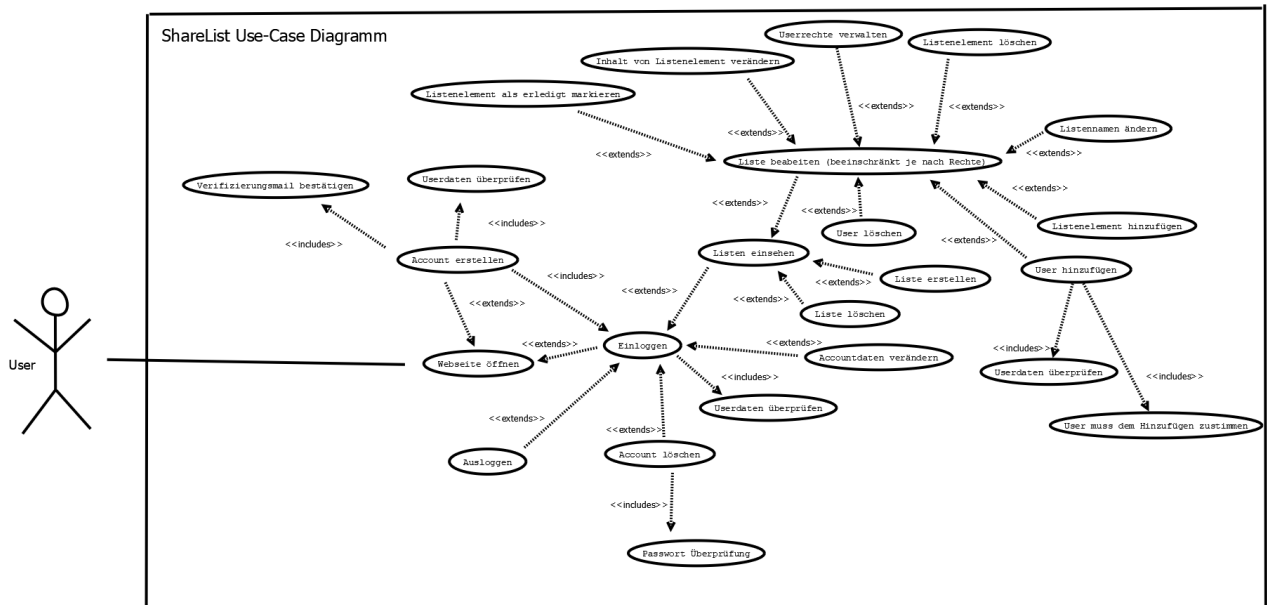


Abbildung 2.2 Skizze des Use-Case Diagramms

2.3. Produktfunktionen

Use Case 1	
Name	Account erstellen
Kurzbeschreibung	Ein User ⁵ erstellt einen persönlichen Account.
Akteure	User
Auslöser	User will einen Account erstellen.
Ergebnis(se)	Account wird für den User erstellt.
Vorbedingungen	Account mit gegebener E-Mail darf nicht bereits existieren.
Eingehende Daten	E-Mail und Passwort
Ablaufbeschreibung	User gibt E-Mail und Passwort an, daraufhin wird überprüft, ob die E-Mail bereits genutzt wird, wenn nicht wird der Account erstellt und der User kriegt eine Bestätigungsmail zugesendet.
Fehlerverhalten	E-Mail wird bereits genutzt, daraufhin wird kein Account erstellt.
Variationen	/

Use Case 2	
Name	Einloggen
Kurzbeschreibung	User meldet sich mit seinen Account Daten beim System an.
Akteure	User
Auslöser	User möchte sich bei der Website mit seinem Account anmelden.
Ergebnis(se)	User wird im System angemeldet.
Vorbedingungen	Account für den User existiert bereits.
Eingehende Daten	E-Mail und Passwort
Ablaufbeschreibung	User gibt E-Mail und Passwort an, daraufhin wird überprüft, ob die Daten in Ordnung sind, wenn ja wird dem User einen persönlichen Token ⁶ zurückgegeben.
Fehlerverhalten	Account existiert nicht oder die Anmelde Daten sind falsch.
Variationen	/



Use Case 3	
Name	Ausloggen
Kurzbeschreibung	User meldet sich vom System ab.
Akteure	User
Auslöser	User möchte sich bei der Website mit seinem Account abmelden.
Ergebnis(se)	User wird im System abgemeldet.
Vorbedingungen	User ist bereits angemeldet.
Eingehende Daten	/
Ablaufbeschreibung	User meldet sich von der Website ab.
Fehlerverhalten	/
Variationen	/

Use Case 4	
Name	Account löschen
Kurzbeschreibung	User löscht seinen Account.
Akteure	User
Auslöser	User möchte seinen Account löschen.
Ergebnis(se)	Der Account vom User wird gelöscht.
Vorbedingungen	User muss angemeldet sein und der Account muss existieren.
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User gibt Passwort an, daraufhin wird überprüft, ob das Passwort in Ordnung ist und darauf hin wird dann der Account gelöscht.
Fehlerverhalten	Passwort ist falsch.
Variationen	/

Use Case 5	
Name	Accountdaten verändern
Kurzbeschreibung	User verändert seine Accountdaten.
Akteure	User
Auslöser	User möchte seine Accountdaten ändern.
Ergebnis(se)	Accountdaten werden geändert
Vorbedingungen	User muss angemeldet sein und der Account muss existieren.
Eingehende Daten	Token, Passwort und neues Passwort
Ablaufbeschreibung	User gibt neues und altes Passwort an, daraufhin wird überprüft, ob das Passwort in Ordnung ist, und daraufhin werden die Accountdaten geändert auf das neue Passwort.
Fehlerverhalten	Passwort ist falsch.
Variationen	/

Use Case 6	
Name	Listen einsehen
Kurzbeschreibung	Listen werden eingesehen.
Akteure	User
Auslöser	User möchte Listen einsehen.
Ergebnis(se)	Listen werden angezeigt.
Vorbedingungen	User muss angemeldet sein.
Eingehende Daten	Token
Ablaufbeschreibung	User möchte sich die Listen ansehen, daraufhin werden alle Listen die zum Token korrespondieren zurückgegeben.
Fehlerverhalten	User hat keine Rechte zu der Liste.
Variationen	/



Use Case 7	
Name	Liste löschen
Kurzbeschreibung	Liste wird gelöscht.
Akteure	User
Auslöser	User möchte Liste löschen.
Ergebnis(se)	Liste wird gelöscht.
Vorbedingungen	User muss angemeldet sein und Besitzer der Liste sein.
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User gibt Passwort an, daraufhin wird überprüft, ob das Passwort in Ordnung ist und ob der User Besitzer der Liste ist. Wenn das alles zutrifft, wird die Liste gelöscht.
Fehlerverhalten	Passwort falsch oder User ist nicht Besitzer der Liste.
Variationen	/

Use Case 8	
Name	Liste erstellen
Kurzbeschreibung	Eine Liste wird erstellt
Akteure	User
Auslöser	User möchte eine Liste erstellen
Ergebnis(se)	Liste wird erstellt
Vorbedingungen	User muss angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User loggt sich ein und erstellt eine Liste
Fehlerverhalten	Passwort ist falsch
Variationen	/

Use Case 9	
Name	Liste bearbeiten
Kurzbeschreibung	Liste wird von bestimmten Personen bearbeitet
Akteure	User
Auslöser	User möchte die Liste bearbeiten
Ergebnis(se)	Liste wird bearbeitet nach Vorstellung des Users
Vorbedingungen	User muss angemeldet sein und entsprechende Rechte für die Bearbeitung besitzen
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User wählt eine Liste, welche er bearbeiten kann und verändert diese dann seinen eigenen Wünschen entsprechend
Fehlerverhalten	Passwort falsch
Variationen	/

Use Case 10	
Name	User hinzufügen
Kurzbeschreibung	Admin ⁷ fügt einen User hinzu
Akteure	Admin
Auslöser	Admin möchte einen weiteren User der Liste hinzufügen
Ergebnis(se)	Neuer User wird hinzugefügt
Vorbedingungen	Admin muss angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	Admin wählt Liste und fügt einen neuen User hinzu
Fehlerverhalten	Passwort falsch
Variationen	/



Use Case 11	
Name	User löschen
Kurzbeschreibung	Ein User wird gelöscht
Akteure	Admin
Auslöser	Admin entschließt sich einen User der Liste zu entfernen
Ergebnis(se)	User wird von der Liste entfernt
Vorbedingungen	Admin muss angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	Admin wählt den User, welchen er entfernt haben möchte und entfernt ihn
Fehlerverhalten	Passwort ist falsch
Variationen	/

Use Case 12	
Name	Userrechte verwalten
Kurzbeschreibung	Die Userrechte der User einer Liste werden verwaltet
Akteure	Admin
Auslöser	Admin möchte den Usern bestimmte Rechte zuteilen
Ergebnis(se)	Den Usern werden die Rechte zugeteilt, welche den Vorstellungen des Admins entsprechen
Vorbedingungen	Admin muss angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	Admin geht auf den User seiner Wahl und fügt ihm die Rechte zu, welche er auf diesem User haben möchte oder entfernt ihm die Rechte
Fehlerverhalten	Passwort ist falsch
Variationen	/

Use Case 13	
Name	Listenelement löschen
Kurzbeschreibung	User löscht ein Listenelement
Akteure	User
Auslöser	Ein überflüssiges Listenelement soll gelöscht werden
Ergebnis(se)	Listenelement wird gelöscht
Vorbedingungen	User muss entsprechende Rechte besitzen und angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User wählt ein Listenelement, welches er entfernen will und entfernt es dann
Fehlerverhalten	Passwort ist falsch oder User hat nicht die entsprechenden Rechte
Variationen	/

Use Case 14	
Name	Listenelement hinzufügen
Kurzbeschreibung	Ein Listenelement hinzufügen
Akteure	User
Auslöser	Der User möchte ein Listenelement zur Liste hinzufügen
Ergebnis(se)	Ein Listenelement wird hinzugefügt
Vorbedingungen	User muss entsprechende Rechte besitzen und angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User geht auf die Liste und fügt ein Listenelement hinzu
Fehlerverhalten	Passwort ist falsch oder der User hat nicht die entsprechenden Rechte
Variationen	/



Use Case 15	
Name	Inhalt von Listenelement verändern
Kurzbeschreibung	Die Inhalte des Listenelements werden verändert
Akteure	User
Auslöser	Der User möchte einen bestimmten Inhalt vom Listenelement ändern
Ergebnis(se)	Der Inhalt des Listenelements wird verändert
Vorbedingungen	User muss entsprechende Rechte besitzen und angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User geht auf die Liste und verändert den Inhalt seinen Wünschen entsprechend
Fehlerverhalten	Passwort ist falsch oder der User hat nicht die entsprechenden Rechte
Variationen	/

Use Case 16	
Name	Listennamen ändern
Kurzbeschreibung	Die Änderung vom Namen der Liste
Akteure	User
Auslöser	Der User möchte einen neuen Listennamen haben
Ergebnis(se)	Der neue Name der Liste wird nun angezeigt
Vorbedingungen	User muss entsprechende Rechte besitzen und angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User geht auf die Liste und fügt einen neuen Listennamen ein
Fehlerverhalten	Passwort ist falsch oder der User hat nicht die entsprechenden Rechte
Variationen	/

Use Case 17	
Name	Listenelement als erledigt markieren
Kurzbeschreibung	Ein Listenelement als erledigt markieren, sobald es erledigt ist
Akteure	User
Auslöser	Der User möchte ein Listenelement, welches vollbracht wurde markieren
Ergebnis(se)	Ein Listenelement wird als erledigt markiert
Vorbedingungen	User muss entsprechende Rechte besitzen und angemeldet sein
Eingehende Daten	Token und Passwort
Ablaufbeschreibung	User geht auf die Liste und wählt ein Listenelement welches vollbracht wurde, um es als erledigt zu markieren
Fehlerverhalten	Passwort ist falsch oder der User hat nicht die entsprechenden Rechte
Variationen	/

3. Nicht funktionale Anforderungen

Usability – Der Nutzer sollen die Web-Applikation⁸ einfach nutzen können, mithilfe von gutem Design, welches dem Nutzer das Bedienen erleichtern soll.

Reliabilität – Die Website soll immer verfügbar sein und möglichst bis gar keine Fehler aufweisen. Was durch gute Validation⁹ und ohne Bugs¹⁰ erreicht werden kann.

Performance – Die Website soll auch eine sehr kurze Antwort Zeit haben. Was dadurch erfüllt wird, indem möglichst schneller und guter Code geschrieben wird.

Sicherheit – Alle Passwörter der Nutzer sollen verschlüsselt werden.

Portierbarkeit – Die verschiedenen Schnittstellen¹¹ für die verschiedenen Aufgaben: Frontend, Backend.



Wartbarkeit – Code so schreiben das man leicht und schnell neue Sachen hinzuaddieren kann.

Wiederverwendung von Systemteilen – zum Beispiel Footer¹² und Header¹³ der Website da dieser öfter benötigt wird.

Skalierbarkeit – Das die Server auch mehr Nutzer aushalten, falls die Idee mehr Aufmerksamkeit erlangt.

Lokalisierung - Das alles an das Österreichische Gesetz angepasst ist.

Architektur – Privater Server bei Easynome (<https://www.easynome.at/de>)

4. Produktdaten

- Daten:
 - Nutzerdaten (Anzahl: 10000+)
 - Hashed¹⁴ Passwort
 - Listendaten (keine Angaben zu Zahlen)
 - Listitems (keine Angaben zu Zahlen)
 - Besitzer (gehört Liste)
 - Benutzer (nimmt bei Liste teil)
- Datenbank:
 - Vorgesehen, aber privat (aus Sicherheitsgründen)

5. Skizze der Systemarchitektur

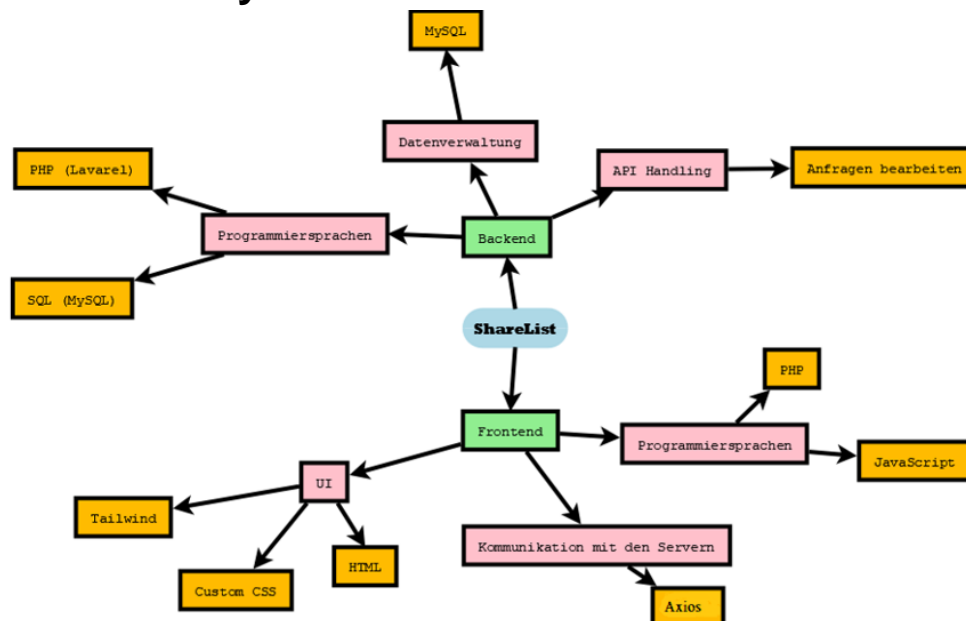


Abbildung 5.0 Skizze der Systemarchitektur

6. Schnittstellen

Wie bereits im Diagramm der Systemarchitektur sichtbar, wird die UI mit CSS¹⁵, HTML¹⁶ und mit dem Framework¹⁷ Tailwind¹⁸ erarbeitet. Für die Kommunikation nutzen wir Axios¹⁹ und als Programmiersprachen wird PHP²⁰ und JavaScript²¹ genutzt. Im Backend für die Datenverwaltung MySQL²² genutzt und als Programmiersprachen SQL²³ und PHP (Laravel²⁴).



7. Lieferumfang

- Vollständiges Backend
 - MySQL Server
 - REST²⁵ Server
- Vollständiges Frontend
 - HTML, PHP, JavaScript ... Dateien
 - Assets (statische Bilder, Konfigurationen, ...)

8. Abnahmekriterien (optional)

Man muss in der Lage sein sich einzuloggen, Listen zu erstellen und Listenelemente hinzuzufügen, aber auch User in die eigene Liste hinzuzufügen. Das Ändern des Namens einer Liste sowie das Löschen eines Accounts wären wichtige Punkte, aber nicht essenziell. Außerdem ist festgelegt, dass die Lieferung das ganze Projekt umschließt, welches am 07.06.2022 geliefert wird.

9. Glossar

Nr.	Begriff	Links zur Erklärung
1	Frontend	https://www.datacenter-insider.de/was-ist-ein-frontend-und-ein-backend-a-714429/
2	Backend	https://www.datacenter-insider.de/was-ist-ein-frontend-und-ein-backend-a-714429/
3	UI	https://kulturbanause.de/faq/ui/
4	ToDo-Liste	https://www.microtech.de/erp-wiki/todo-liste/#:~:text=Die%20ToDo-Liste%20ist%20der,der%20Listen%20zählen%20zur%20Ablauforganisation
5	User	https://praxistipps.chip.de/was-ist-ein-user-einfach-und-verstaendlich-erklart_41367
6	Token	https://www.blockchain-insider.de/was-ist-ein-token-a-854928/
7	Admin	https://praxistipps.chip.de/was-ist-ein-admin-einfach-erklart_41693
8	Web-Applikation	https://www.dev-insider.de/was-ist-eine-web-app-a-596814/
9	Validation	https://www.igi-global.com/dictionary/code-validation/69141
10	Bugs	https://praxistipps.chip.de/was-ist-ein-bug-bugfix-einfach-erklart_41528
11	Schnittstellen	https://www.it-business.de/was-ist-eine-schnittstelle-a-714831/
12	Footer	https://praxistipps.chip.de/header-und-footer-was-ist-das-einfach-erklart_41286
13	Header	https://praxistipps.chip.de/header-und-footer-was-ist-das-einfach-erklart_41286
14	Hashed	https://www.ibau.de/akademie/glossar/hashwert/
15	CSS	https://www.ionos.at/digitalguide/websites/webdesign/was-ist-css/
16	HTML	https://www.seo-kueche.de/lexikon/html/
17	Framework	https://praxistipps.chip.de/was-ist-ein-framework-einfach-erklart_41348
18	Tailwind	https://www.twigbit.com/glossar/tailwind#:~:text=Tailwind%20ist%20ein%20Utility-First,einfach%20eigene%2C%20einzigartige%20Designs%20erstellen
19	Axios	https://de.karabayyazilim.com/blog/javascript/axios-nedir-2020-05-21-115817
20	PHP	https://www.php.net/manual/en/intro-what-is.php
21	JavaScript	https://developer.mozilla.org/de/docs/Learn/JavaScript/First_steps/What_is_JavaScript
22	MySQL	https://www.bigdata-insider.de/was-ist-mysql-a-614184/
23	SQL	https://www.sqlcourse.com/beginner-course/what-is-sql/
24	Laravel	https://kinsta.com/de/wissensdatenbank/was-ist-laravel/
25	REST	https://www.redhat.com/de/topics/api/what-is-a-rest-api