

## Lern- und Arbeitsauftrag 1305 Codes, NoSQL und Backend

<b>Titel:</b>	<b>Modul 165, 306 - Gruppenarbeit</b>
<b>Modul:</b>	IMS-Lernatelier 2 Informatiker/in EFZ
<b>Autor / Version:</b>	Michael Schneider
<b>Sozialform</b>	Gruppenarbeit mit 1 bis 4 Personen
<b>Hilfsmittel:</b>	<ul style="list-style-type: none"> <li>• Alle</li> </ul>

### Ausgangslage

In diesem Auftrag sollen folgende Module vertieft werden:

165 NoSQL-Datenbanken einsetzen

347 Dienst mit Containern anwenden

426 Software mit agilen Methoden entwickeln

Suchen Sie sich ein Projekt aus, das zu Ihrem Modulablaufplan passt oder wählen Sie ein eigenes Projekt zur Repetition des Stoffes oder zum Kennenlernen von neuen Technologien aus. Besprechen Sie eigene Projekte mit der Lehrperson.

Hinweis: Es ist nicht erlaubt, Aufgaben für das Qualifikationsverfahren (LB, LBV, ...) während des Lernateliers zu realisieren. Das Lernatelier ist auch keine Aufgabenstunde.

Hinweis: Dimensionieren Sie Ihr Projekt so, dass Sie in der gegebenen Zeit fertig werden, aber auch genügend Arbeit vorhanden ist, um die Zeit zu füllen.

### Ziele

Am Schluss dieses Lern- und Arbeitsauftrages sollen Sie ein fertiges Produkt mit Projektdokumentation erstellt haben.

### Aufgaben

#### 1) Informieren

a) Schliessen Sie sich in Gruppen zusammen, die mindestens aus zwei, maximal aus vier Personen bestehen.

b) Wählen Sie in der Gruppe ein Projekt aus. Beachten Sie den Schwierigkeitsgrad, Ihren Wissensstand und die zur Verfügung stehende Zeit bei der Auswahl. Sprechen Sie eigene Ideen mit der Lehrperson ab.

#### Quiz Webapp

c) Machen Sie eine Anforderungsanalyse. Werden Sie sich in der Gruppe einig, was Sie genau realisieren möchten. Das Projekt sollte durch die Anforderungen so präzise wie möglich beschrieben sein. Legen Sie die minimalen Anforderungen und «nice-to-have»-Anforderungen fest, die bei genügend Zeit realisiert werden können.

Nr.	Anforderung	Muss/Kann	Funkt/Rand/Qua
1	Der User kann im Programm Quizze erstellen	Muss	Funktional
2	Das Programm speichert diese in eine Datenbank ab	Muss	Funktional
3	Der User kann die Fragen abrufen und sich vom Programm abfragen lassen	Muss	Funktional
4	Es gibt true/false und Wortabfragen	Muss	Funktional
5	Es gibt multiplechoice Fragen	Kann	Funktional
6	Das Programm gibt, nachdem der User das Quiz ausgefüllt hat, zurück, wie viele Fehler der User hatte.	Muss	Funktional
7	Die Webapp hat eine API	Muss	Rand
8	Die Webapp ist mit JS, Node, HTML und CSS geschrieben	Muss	Rand
9	Der User kann mehrere verschiedene Quizze erstellen	Kann	Funktional
10	Es gibt eine Login und Registrierfunktion	Muss	Funktional
11	Der User kann Quizze public oder privat machen	Muss	Funktional
12	Der User kann Quizze von anderen Usern spielen und zu seinen privaten Quizzen hinzufügen	Muss	Funktional
13	Der User kann selbst erstellte Quizze komplett löschen	Muss	Funktional
14	Der User kann on anderen Usern hinzugefügte Quizze aus seiner Liste entfernen	Muss	Funktional

d) Legen Sie fest, wo und wie Sie in der Gruppe Arbeitsergebnisse speichern und austauschen möchten.

e) Erarbeiten Sie sich noch fehlende Grundlagen für das Projekt.

Sie können die erarbeiteten Dokumente mit der Lehrperson besprechen, wenn Sie möchten.

## 2) Planen

- Teilen Sie das Projekt in klare, einzelne Aufgaben auf («Arbeitspakete»).
- Verteilen Sie diese Arbeiten an die Gruppenmitglieder und setzen Sie für jede Arbeit ein Abschlussdatum fest, dass mit dem Plan Ihrer Klasse übereinstimmt.
- Erstellen Sie eine ToDo-Liste mit verantwortlicher Person, Arbeitspaket und Abschlussdatum für das ganze Projekt.

Nr.	Arbeitspaket	Dauer	Deadline
1	HTML/CSS erstellen	3x45	05.04.2023
2	API mit CRUD Befehlen für MongoDB erstellen	4x45	03.05.2023
3.	Neue Quizze können erstellt werden mit: Geschriebenen Antworten True/False multiplechoice	3x45	03.05.2023
4	Der User kann die Quizze spielen	3x45	10.05.2023
5	Das Programm hat ein Login und eine Registrierung	2x45	10.05.2023
6	Daten werden in dem Login gespeichert und können vom Programm abgerufen werden	3x45	17.05.2023
7	Der User kann Quizze von anderen Usern spielen	1x45	17.05.2023
8	Der User kann Quizze von anderen Spielern zu seiner Liste hinzufügen	1x45	17.05.2023
9	Die Quizze können gelöscht werden	2x45	24.05.2023

Sie können diese ToDo-Liste mit der Lehrperson besprechen, wenn Sie möchten.

## 3) Entscheiden

Fällen und dokumentieren Sie wichtige Entscheidungen.

- Das Farbschema soll grün und blau sein
- Als Schriftart werde ich Roboto wählen
- Das Programm wird in Visual Studio Code geschrieben

#### 4) Realisieren

Erstellen Sie das Programm und die notwendigen Dokumente. **Erstellen Sie das Programm selbstständig und verwenden Sie nicht einfach das Resultat eines Lernvideos oder fertigen Code.**

## 5) Kontrolle

- Erstellen Sie Testfälle für das Programm.
- Testen Sie Ihr Programm und schreiben Sie ein kurzes Testprotokoll mit Testumgebung.

Testnr.	Anforderungsnr.	Vorbereitung	Eingabe	Ausgabe
1.1	1+11	Das Programm wurde gestartet und der User ist eingeloggt	1.klick auf «Create Quiz» 2.Quizname «Test» eingeben 3.Description «Testdescription eingeben» 4.Visibility «public» wählen 5. «Create Quiz» klicken 7.Im Question Type dropdown «True or False» wählen 8. 5* Wiederholen{ In Question «w» eingeben Answer «true» wählen «Add Question» klicken } 9.«Finish Quiz» klicken 11. «OK» klicken	2.localhost:3000/quizcreator wird geöffnet 6.Form mit dem Titel «Create Question» öffnet sich 10. alert(«Das Quiz wurde erfolgreich erstellt») 12. Weiterleitung auf «localhost:3000» 13. Quiz wird in der My Quizzes Liste und Public Quizzes Liste angezeigt
2.2	2	Es wurde Mindestens ein public Quiz erstellt und Postman wurde geöffnet	1.GET Befehl mit der URL «http://localhost:3000/api/getAll» ausführen	2. Alle mit public erstellten Quizze werden im JSON Format angezeigt.
3.3	3+6	Testnr. 1.1 wurde erstellt und «localhost:3000» ist geöffnet	1.Bei dem Quiz mit dem Titel «Test» auf «Play» klicken 5.Alle Fragen mit «True» beantworten 6.Auf «submit» klicken 8.Auf «OK» klicken	2.localhost:3000/play wird geöffnet 3.Es wird ein Form mit 5 Fragen angezeigt 4. Alle Fragen haben den Titel «w» und «True» oder «False» zur Auswahl 7.Kleines Fenster mit dem Titel «Correction» wird angezeigt. Auf dem Fenster sollte zusehen sein: - Ein komplett grüner Balken - Ein Text "You got 5 correct and 0 wrong!" 9.Weiterleitung auf Seite "localhost:3000"
3.4	3+6	Testnr. 1.1 wurde erstellt und «localhost:3000» ist geöffnet	1.Bei dem Quiz mit dem Titel «Test» auf «Play» klicken 5.Alle Fragen mit «False» beantworten 6.Auf «submit» klicken 8.Auf «OK» klicken	2.localhost:3000/play wird geöffnet 3.Es wird ein Form mit 5 Fragen angezeigt 4. Alle Fragen haben den Titel «w» und «True» oder «False» zur Auswahl 7.Kleines Fenster mit dem Titel «Correction» wird angezeigt. Auf dem Fenster sollte zusehen sein: - Ein komplett roter Balken - Ein Text "You got 0 correct and 5 wrong!" 9.Weiterleitung auf Seite "localhost:3000"

4.5	4	Im Quizcreator wird das «Create Question» Form angezeigt	1.Im Dropdown Menu «Write» wählen	2.Das Form verändert sich und hat: - Eine «Question» Textarea - Eine «Answer» Textarea
4.6	4	Im Quizcreator wird das «Create Question» Form angezeigt	1.Im Dropdown Menu «True or False» wählen	2.Das Form verändert sich und hat: - Eine «Question» Textarea - Ein «True» Radioknopf - Ein «False» Radioknopf
5.7	5	Im Quizcreator wird das «Create Question» Form angezeigt	1.Im Dropdown Menu «Multiplechoice» wählen	2.Das Form verändert sich und hat: - Eine «Question» Textarea - Eine «Answer» Textarea - 3 «Wrong Answers» Textareas
10.8	10	Das Programm wurde gestartet. Die Seite «localhost:3000/register» wurde geöffnet	1.Name «TestUser» eingeben 2.Email «w@w» eingeben 3.Password «www» eingeben 4.«Register» klicken 6.Email «w@w» eingeben 7.Password «www» eingeben 9.«Login» klicken	5.Weiterleitung auf «localhost:3000/login» 10.Weiterleitung auf «localhost:3000»
11.9	11	Ein public Quiz mit dem Namen «public Test Quiz» und ein private Quiz mit dem Namen «private Test Quiz» wurden erstellt	1.Seite «localhost:3000» öffnen	2.In der Liste «My Quizzes» sollten die Quizze «public Test Quiz» und «private Test Quiz» zu sehen sein. 3.In der Liste «Public Quizzes» sollte das «public Test Quiz» zu sehen sein. Das «private Test Quiz» darf nicht zu sehen sein.
12.10	12	Mit einem anderen User wurde ein Public Quiz mit dem Titel «Other User Quiz» erstellt. Eingelogggt befindet man sich auf «localhost:3000»	1.In der Liste «Public Quizzes» nach dem Quiz «Other User Quiz» suchen und auf «Play» klicken.	2.localhost:3000/play wird geöffnet 3.Alle erstellten Fragen sind zu sehen
12.11	12	Mindestens ein public Quiz wurde von einem nicht eingeloggten User erstellt. Mit einem User einloggen und auf «localhost:3000» sein	1.Ein von einem anderen User erstelltes Quiz suchen 2.Auf das Plusssymbol klicken 3. <Quizid> kopieren 4. «OK» klicken 6. «Add Quiz» klicken 8.<Quizid> in das prompt kopieren 9.«OK» klicken	3.Alert (This is the Id of the quiz: <Quizid> You can either share it or add it to your quizzes) 5.Alert verschwindet 7.prompt(Write the received Id of the Quiz in the box and press Ok) 10.Seite wird neu geladen 11.In «My Quizzes» sollte das Quiz erscheinen
13.12	13	Mindestens ein public Quiz wurde vom eingeloggten User erstellt. Mit dem User einloggen und auf «localhost:3000» sein	1.Das Public Quiz in der Liste «My Quizzes» suchen 2.Das Abfall Symbol klicken	3.Die Seite wird neu geladen 4.Das Quiz wird in der Liste «My Quizzes» und der Liste «Public Quizzes» nicht mehr angezeigt
14.13	14	Mindestens ein public Quiz wurde von einem nicht eingeloggten User erstellt. Der eingeloggte User hat das Quiz hinzugefügt. Mit einem User einloggen und auf «localhost:3000» sein	1.Das hinzugefügte Quiz von einem anderen User suchen 2.Aud das Abfall Symbol klicken	3.Seite wird neu geladen 4.In «MyQuizzes» ist das Quiz verschwunden. Es wird jedoch immer noch in der Liste «public Quizzes» angezeigt

Sie können diese Testdokumente mit der Lehrperson besprechen, wenn Sie möchten.

Testnr	Status	Tester	Datum
1.1	OK	Samuel Matarese	24.05.2023
2.2	OK	Samuel Matarese	24.05.2023
3.3	OK	Samuel Matarese	24.05.2023
3.4	OK	Samuel Matarese	24.05.2023
4.5	OK	Samuel Matarese	24.05.2023
4.6	OK	Samuel Matarese	24.05.2023
5.7	OK	Samuel Matarese	24.05.2023
10.8	OK	Samuel Matarese	24.05.2023
11.9	OK	Samuel Matarese	24.05.2023
12.10	OK	Samuel Matarese	24.05.2023
12.11	OK	Samuel Matarese	24.05.2023
13.12	OK	Samuel Matarese	24.05.2023
14.13	OK	Samuel Matarese	24.05.2023

## Bericht

Alle Tests haben funktioniert. Somit kann die Quiz App fehlerfrei ausgeführt werden.

## 6) Auswertung

Überlegen Sie sich, was gut gelaufen ist und was eher nicht. Sie können diese Informationen in Ihrem Portfolioeintrag verwenden.

Ich denke das Projekt ist wirklich sehr gut gelaufen. Die Informieren, Planen und Entscheiden Phasen konnte ich alle am ersten Tag abschliessen. Ich wollte eigentlich zuerst ein eigenes Betriebssystem entwickeln, habe aber dann gemerkt, dass dies zu lange dauern würde, da ich noch keine Erfahrung mit der dafür gebrauchten Programmiersprache Assembly x86 habe. Deshalb habe ich mich für eine simple Quizapp mit einem Login System entschieden.

Das Realisieren lief bis zum Login Teil auch gut. Ich dachte es wäre schlauer das Login erst am Ende einzubauen, da es sicherlich nicht zu kompliziert sein würde, dieses später noch einzubauen. Wie ich aber dann feststellen musste, war das ganze gar nicht so leicht. Ich musste meine ganze Applikation umstrukturieren, da meine Seiten zu Beginn nicht mit ejs, sondern HTML liefen.

Das Kontrollieren und Auswerten konnte ich dennoch am letzten Tag noch lösen.

Ich denke ich habe mir zu viel in diesem Projekt vorgenommen. Ich habe sogar einen Tag mehr daran gearbeitet als nötig war. Das dieses Projekt zu gross war, sieht man auch an diesem Portfolioeintrag. Ich wollte mich schon kurz halten, aber habe trotzdem einen sehr langen Eintrag geschrieben. Deshalb lautet mein Verbesserungsvorschlag auf nächstes Mal mir weniger vorzunehmen und nur in den geplanten Zeiten zu arbeiten.

## 7) Portfolioeintrag

Schreiben Sie den Portfolioeintrag und reichen Sie ihn ein. Gehen Sie im Portfolioeintrag auf Ihre Ziele und deren Erreichung sowie die erstellte Dokumentation ein. Beschreiben Sie im Portfolioeintrag auch Ihren Lernweg, die Probleme, Erkenntnisse und Erfolge.



## Projektvorschläge

### 165 NoSQL-Datenbanken einsetzen

#### Choose your own adventure

«Choose your own adventure» sind Bücher, in denen Ihnen eine Situation geschildert und verschiedene Entscheidungsmöglichkeiten gezeigt werden.

(<https://de.wikipedia.org/wiki/Spielbuch>). Erstellen Sie eine (Web-)Applikation, mit der Sie eine solche Geschichte durchspielen können. Speichern Sie alle Daten in einer NoSQL-Datenbank.

#### Repetitionshilfe

Erstellen Sie ein Programm, mit dem Sie Faktenwissen aus dem Unterricht repetieren können. Sie sollten Fragen und Antworten eingeben können und das Programm soll Sie abfragen.

Erweiterungen wie eine Statistik oder andere Fragetypen wie Multiple-Choice-Aufgaben sind möglich. Speichern Sie alle Daten in einer NoSQL Datenbank.

#### Medienmanager

Erstellen Sie ein Programm, in dem Sie Medien (Bilder, Video, Links, Audio, ...) speichern können. Dabei sollen alle möglichen Metadaten ausgelesen und in der Datenbank gespeichert werden können. Erstellen Sie eine Oberfläche, mit der Sie auf diese Metadaten zugreifen können.

#### Gallery-Maker

Erstellen Sie ein Programm, dem Sie eine Verzeichnishierarchie mit Bildern übergeben können und das Ihnen eine schöne Fotogalerie als statisches HTML ausgibt. Dabei soll es eine Übersicht mit Thumbnails und eine Detailansicht geben. Erweiterungsmöglichkeiten sind Texte, die zu den Bildern in einem geeigneten Format abgelegt werden können und das Auslesen der Metadaten.

### 347 Dienst mit Containern anwenden

#### Docker Visualisierung mit phaser.io

Mit phaser.io (<https://phaser.io/>) können Sie Spiele und visualisierungen für den Browser entwickeln.

Benutzen Sie die Ausgaben von «docker ps», «docker info», etc und stellen Sie Ihre Docker als 3D-Szene im Browser dar:

- Ein Docker kann ein Gebäude sein.
- Je mehr Platz/Memory er benötigt, desto grösser ist es.
- Jeder Prozess ist eine Spielfigur im Haus
- ...

Als Erweiterung könnten Sie beispielsweise Docker herunterfahren, indem Sie das Haus abbrechen (bitte mit Explosionen).

#### Dockercraft (schwieriges Projekt!)

Docker können innerhalb von Minecraft gesteuert werden: <https://github.com/docker/dockercraft>

Nehmen Sie dies als Basis für ein Projekt: Erweitern Sie Cuberite (den Minecraft Server) auf geeignete Art und Weise.

Achtung: Nur Minecraft Spielen ist kein Projekt!

## 426 Software mit agilen Methoden entwickeln

Implementieren Sie folgende Projekte mit den Design-Patterns im Hinterkopf. Versuchen Sie mindestens ein Pattern korrekt einzusetzen.

### RPG-Games

Folgende Kurse beschreiben, wie man mit C# RPG's bauen kann. Nehmen Sie diese objektorientierten Programme als Grundlage und erweitern Sie diese:

- <https://roguesharp.wordpress.com/>
- <https://scottlilly.com/learn-c-by-building-a-simple-rpg-index/>

### Textbasierte Multiuserwelt

Ein MUD (MultiUser Dungeon) ist eine textbasierte Welt, in der mehrere Benutzer gleichzeitig Abenteuer erleben, sich bekämpfen oder Aufgaben erfüllen können.

Installieren Sie <https://github.com/DavidRieman/WheelMUD>, erstellen Sie eine Welt und mit neuen Objekten.

### Augmented Reality

Mit ARCORE (<https://developers.google.com/ar>) können Sie Augmented Reality Applikationen entwickeln. Arbeiten Sie sich ein und erstellen Sie ein eigenes, einfaches Beispiel.



### Build your own x

Wollten Sie schon immer einmal ein eigenes Betriebssystem, einen eigenen Renderer, eine eigene Datenbank, einen eigenen Webserver oder eine eigene Blockchain entwickeln?

<https://github.com/codecrafters-io/build-your-own-x> hat Anleitungen dazu gesammelt.

### Gütekriterien

Der Lern- und Arbeitsauftrag ist erfüllt, wenn ...

- Wenn Sie das Projekt nach IPERKA abgearbeitet haben.
- Wenn Sie eine lauffähige Applikation produziert haben.

### Zusätzliche Angaben zum Auftrag

Keine.

### Mögliche Erweiterungsaufträge

Keine.