Lernquiz

Team Blank

**2019**

Luis Karch, Marcel Enslinger, Friedrich Stilo

KOS

08.01.2019



Gliederung der Dokumentation

[Daily Scrum 2](#_Toc4137160)

[Daily-Scrum Ablauf: 2](#_Toc4137161)

[Daily-Scrum Regeln: 2](#_Toc4137162)

[Wahl zum Scrum-Master 2](#_Toc4137163)

[Was soll das Quiz können? 3](#_Toc4137164)

[Ablauf einer Runde: 3](#_Toc4137165)

[Zusätzliche Features: 4](#_Toc4137166)

[Verwendete Technologien: 4](#_Toc4137167)

[Umsetzung / Retrospektive 4](#_Toc4137168)

[Login und Registrierung 4](#_Toc4137169)

[Chat und Räume 4](#_Toc4137170)

[Spiel an sich 5](#_Toc4137171)

[Nicht erreichte Features die geplant waren und mögliche Ausbaustufen 5](#_Toc4137172)

[Punkteverteilung 5](#_Toc4137173)

[Punkte 5](#_Toc4137174)

# Daily Scrum

## Daily-Scrum Ablauf:

1. Aktueller Stand wird überprüft.
2. Treten Probleme auf?
3. (evtl. Ansätze zur Behebung!?)
4. Was ist das Heutige Ziel?

## Daily-Scrum Regeln:

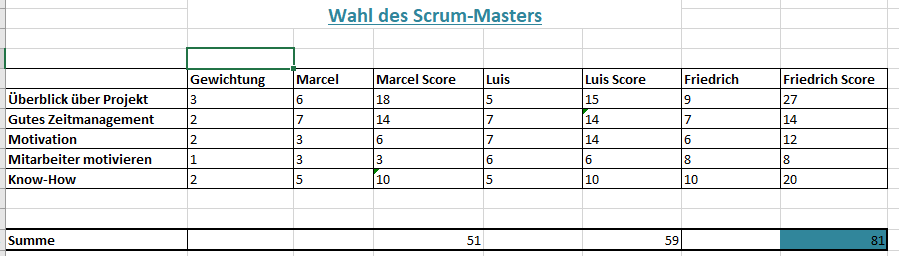
1. Jedes Teammitglied kommt der Reihe nach dran.
2. Wer dran ist, geht den ganzen obigen Ablauf durch.
3. Erst zum Schluss werden Fragen oder Anregungen mitgeteilt.

# Wahl zum Scrum-Master

Bei der Wahl des Scrum-Masters wurden von der Gruppe „Blank“ verschiedene Aspekte für sehr nötig befunden. Diese erhielten eine Gewichtung von „1“ bis „3“, wobei letztere Wertung für besonders wichtig steht.  
Der erste Aspekt „Mitarbeiter motivieren“ wurde ausgewählt, da der Erhalt der Motivation im Team zum Erreichen der gesetzten Ziele beiträgt. Allerdings wurde dieser nur mit der Wichtigkeit „1“ bewertet, weil die meiste Zeit der Arbeit eher in Einzelarbeit stattfindet und daher die „eigene Motivation“, welche zugleich der zweite Aspekt ist, als notwendiger angesehen wurden. Sie wurde, aufgrund des hohen Anteils an Einzelarbeit mit einer Wertung von „2“ eingestuft und sollte sicherstellen, dass der Scrum-Master auch selbst völlig hinter dem Projekt steht und auch die zusätzlichen Aufgaben seines Amtes motiviert bewältigt. Ein weiterer Punkt für diese Gewichtung, den die Gruppe als wichtig erachtete, ist das nötige „Know-How“ über die verschiedenen Programmiersprachen und anderen Anwendungen, wie beispielsweise „Trello“. Das Team wählte diesen Aspekt, weil der Scrum-Master bei ihnen auch als erster Ansprechpartner bei Fragen oder Problemen fungieren sollte. Auch sollte er möglichst gut und vorrausschauend planen können, was „Blank“ zu ihren letzten zwei Kriterien führte: ein „Gutes Zeitmanagement“ mit der Gewichtung „2“, sowie den „Überblick über das Projekt“. Ersteres ist damit zu begründen, dass der Zeitplan der Projektidee sehr Zeitaufwendig ist und daher ein überlegtes Konzept ausgearbeitet werden muss. Ohne einen gut ausgearbeiteten Zeitplan ist ein Projekt kaum umzusetzen. Als wichtigsten Kritikpunkt wurde der „Überblick über das Projekt“ ausgearbeitet und mit einer Wertung von „3“ beziffert. Die Gruppe entschied sich diesen Aspekt als wichtigsten zu nennen, da der Scrum-Master der Hauptplaner im Projekt sein sollte und über alle Vorgange Bescheid wissen sollte. Er sollte derjenige sein, der die wöchentlichen Fortschritte festhält und das weitere Vorgehen beschreibt.

Aufgrund der oben genannten Aspekte konnte sich Friedrich als Scrum-Master behaupten und konnte vor allem in den Bereichen „Know-How“ und „Überblick über das Projekt“ überzeugen (siehe Beiligendes Excel-Dokument „Wahl zum Scrum-Master“). Speziell im Bereich „Know-How“ erhielt er die volle Punktzahl, da er sich nicht nur in jeder genutzten Programmiersprache bestens auskennt, sondern auch schon im benutzten Scrum-Board „Trello“ Erfahrungen sammeln konnte.

Hier unsere Wahl per Excel-Tabelle:



# Was soll das Quiz können?

* Login mit Username
* Teilnahme an bereits bestehenden Spiel
* Wenn noch kein Spiel läuft, warten auf weitere Mitspieler (mit Möglichkeit auch alleine zu spielen)
* Bestenliste und Aufzählung der letzten Spiele

# Ablauf einer Runde:

Am Anfang der Runde kann man auswählen, aus welchem Bereich die Fragen kommen sollen (Bereiche: AWP, VSY, ITS und BWP). In Jedem Bereich gibt es verschiedene Schwierigkeitsgrade, welche unterschiedlich viel Punkte geben. Man hat nach der Auswahl eines Kästchens eine gewisse Zeit um die Frage zu beantworten. Löst man diese richtig, bekommt man die Punkte die vorher auf dem Kästchen standen. Bei Versagen werden die Punkte nicht abgezogen. Das Kästchen ist nicht wieder wählbar. Bei einer Falschantwort wird das Kästchen ausgegraut und mit einem roten X versehen. Bei einer richtigen Antwort kommt der Name des Spielers in das Kästchen + die Punkte. Danach ist die Runde für diesen Spieler vorbei und der nächste Spieler ist dran. Das Quiz ist vorbei, wenn alle Kästchen dran waren. Gewonnen hat der Spieler mit den meisten Punkten. Sollte man alleine Spielen, kann man sich in einer Bestenliste eintragen. Diese beinhaltet die besten 10 Spiele und eine weitere, welche die letzten 10 Spiele anzeigt. Diese werden auch mit einem Datum versehen.

# Zusätzliche Features:

Bei einer erst Nutzung muss man sich einen Account erstellen, welcher den Username und ein Passwort beinhaltet. Das Passwort wird verschlüsselt in die Datenbank gesichert. Neben dem Frageboard gibt es auch noch eine Chatbox, in der man mit seinem Kontrahenten schreiben kann. In diesem wird der Username vom Login benutzt, wenn der Spieler eine Nachricht abschickt

# Verwendete Technologien:

Die Oberfläche wird mit HTML und CSS realisiert. Für die Fragen, Antwortmöglichkeiten, Bestenliste, speicherbare Accounts und den Chat werden mehrere Datenbanktabellen mit MySQL eingerichtet. Um die Datenbank mit der Oberfläche zu verbinden wird PDO verwendet. Dies ist eine Library für PHP

# Umsetzung / Retrospektive

Zur Versionsverwaltung wurde Git verwendet. Damit sich die drei Programmierbereiche nicht stören und es zu Merge-Konflikten kommt, wurde das Projekt gleich am Anfang in drei Hauptklassen aufgeteilt (unter dem Ordner Classes). Eine Database.php für alle MySQL abfragen, eine Engine.php für die Core-Logik und eine Output.php für die Darstellung.

## Login und Registrierung

Zuerst wurde eine simple Login und Registrier-Logik geschrieben, welche es erlaubt Benutzer anzulegen und sich einzuloggen. Hier wurde das PHP-Eigene Session-Management verwendet, sowie die Passwort-Hash Funktion von PHP. In MySQL wurde als Primary Key der Accounts-Tabelle der Loginname verwendet, sodass es immer nur einen Benutzer mit diesem Login geben kann. Später hat sich diese Vereinfachung (gegenüber einem generiertem Primary Key) als sehr nützlich herausgestellt.

## Chat und Räume

Dann wurde die Funktion mehrere Räume zu haben, sowie der dazugehörige Chat programmiert. Zuerst war eine simplere Version geplant, nämlich nur einen Raum zu haben, jedoch wurde es später auf mehre Räume aufgeteilt, was die Sache etwas kompliziert machte, und man im Nachhinein vielleicht doch gleich von Anfang an das Projekt auf mehrere Räume ausgelegt haben sollte.

Hier wurde auch zuerst mit JavaScript gearbeitet, der Chat funktioniert so, dass im Hintergrund immer AJAX-Requests an den Server geschickt werden, welcher dann die letzten 50 Nachrichten zurückliefert. Zum Senden an den Server wurde auch AJAX verwendet.

## Spiel an sich

Das Herzstück des Projekts bildet ein Game-Log. Anstatt den State des derzeitigen Spiels (oder mehrerer Spiele gleichzeitig) in eine Tabelle zu speichern, funktioniert der Game-Loop mit Log-Einträgen. Das heißt jede Aktion wird mit festgelegten Keywords in die Datenbank gespeichert, sodass das Programm später nur alle Log-Einträge anschauen muss, um sich den derzeitigen Spielstand herzuleiten. Wenn z.B. am Anfang des Spieles die anfänglich versteckten Fragen generiert werden, wird dies mit „setQuestion“ abgespeichert. Wenn ein Spieler auf eine Frage klickt wird dies mit „uncoverQuestion“ festgehalten. Außerdem wird, wenn ein Spieler am Zug ist, ein Eintrag mit „playerTurn“ generiert. Zudem gibt es noch weitere Einträge. Die Core-Logik muss dann nur über alle Einträge gehen und weiß genau was als nächstes passieren darf. So wird z.B. überprüft, ob ein Spieler gerade am Zug ist, wenn auf das Spielfeld geklickt wird.

Die dynamische Interaktion mit dem Spielfeld wurde auch mit AJAX-Requests und JavaScript gelöst. Jeder Klick auf eine Box im Spielfeld generiert eine Anfrage an den Server, welcher dann prüft was zu tun ist. Zudem wird jede Sekunde eine Anfrage losgeschickt, welche den aktuellen Spielstand abfragt und die HTML-Elemente mit jQuery updatet.

# Nicht erreichte Features die geplant waren und mögliche Ausbaustufen

Erreicht wurde eigentlich alle geplanten Punkte, jedoch hätte man das Programm noch weiter ausbauen können. Zum Beispiel ein besseres Sessionmanagement ohne zu große Betonung auf die $\_SESSION Variable. Auch Teilnahme in mehreren Räumen gleichzeitig, sowie Notifications wenn man am Zug ist.

Zudem lassen sich Fragen nur über die Datenbank direkt einbauen und editieren, eine weitere Ausbaustufe wäre ein Interface zum Pflegen und editieren der Frage und Antworten.

# Punkteverteilung

Die Aufgaben waren gleichmäßig verteilt, und jeder hatte in seinem Bereich (Marcel für HTML/CSS, Luis für MySQL und Friedrich für PHP) genug zu tun, sodass sich nur für kleine Abweichungen in der Punkteverteilung entschieden wurde. Zuerst sollte der Scrum-Master einen Bonus bekommen, dies fiel also auf Friedrich. Da Friedrich auch den generellen Überblick über das Projekt hatte und den anderen oft Fragen beantwortet entschied sich das Team dies leicht in der Punkteverteilung darzustellen.

## Punkte

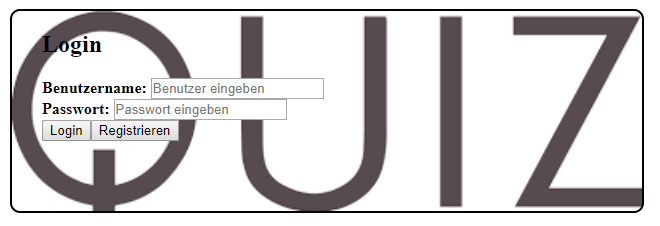
|  |  |
| --- | --- |
| Marcel | 97.5% |
| Luis | 97.5% |
| Friedrich | 105% |

# Installationsanleitung

Der Ordner muss im XAMPP im htdocs Ordner platziert werden, dann muss in der MySQL Workbench das Script Misc/Database.sql ausgeführt werden um die Datenbank zu installieren.

# Vorstellung Programm mit Bildern

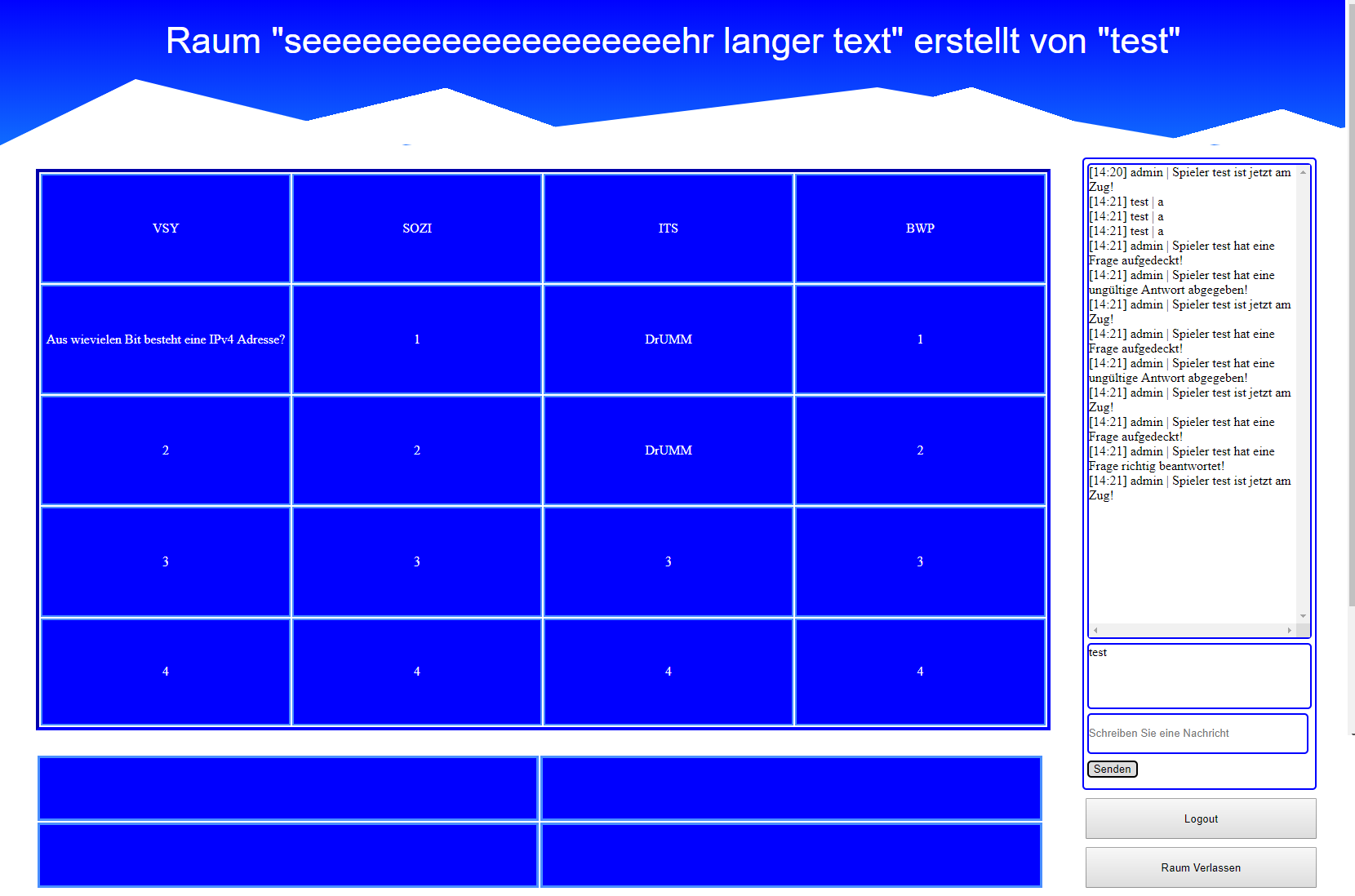
### Login



Raumübersicht

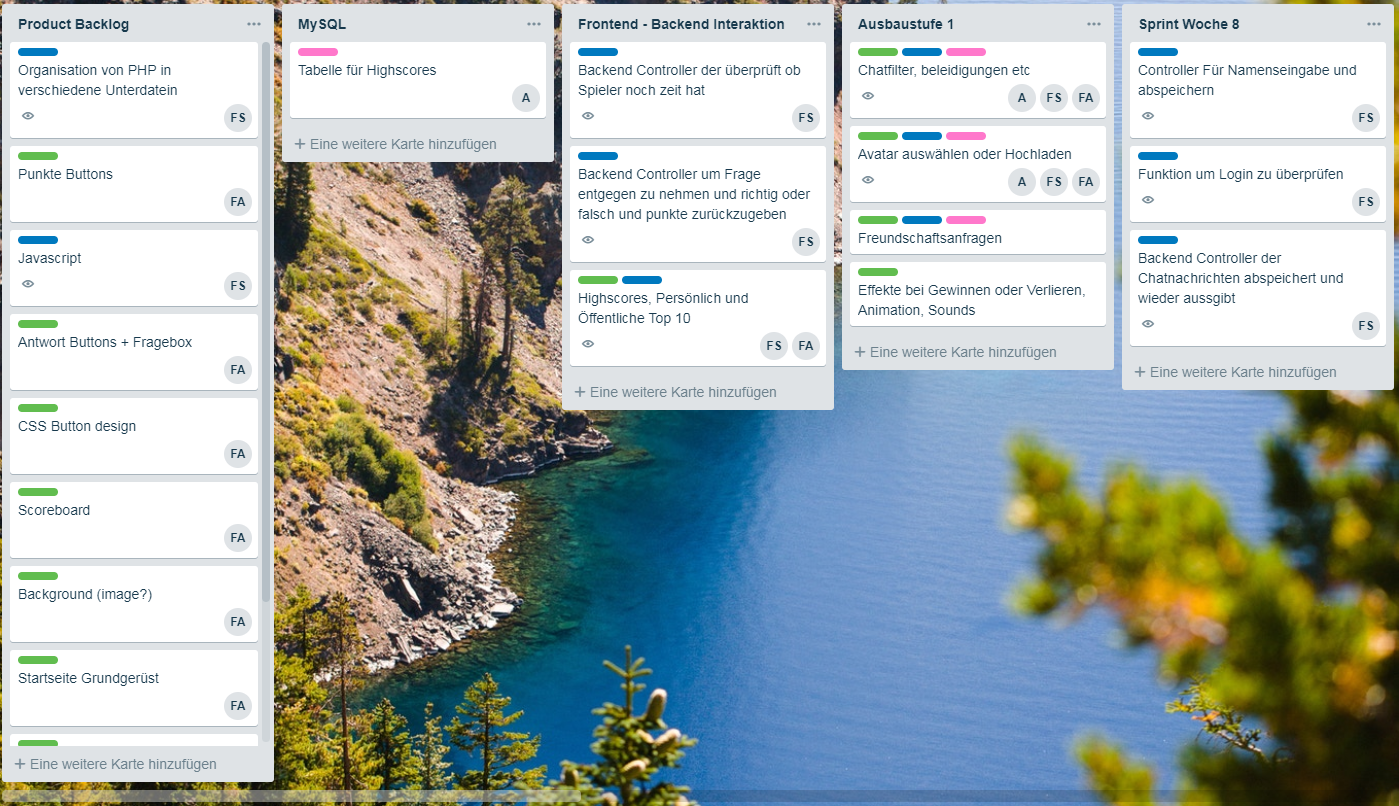


Raum und Spielfeld (beta)

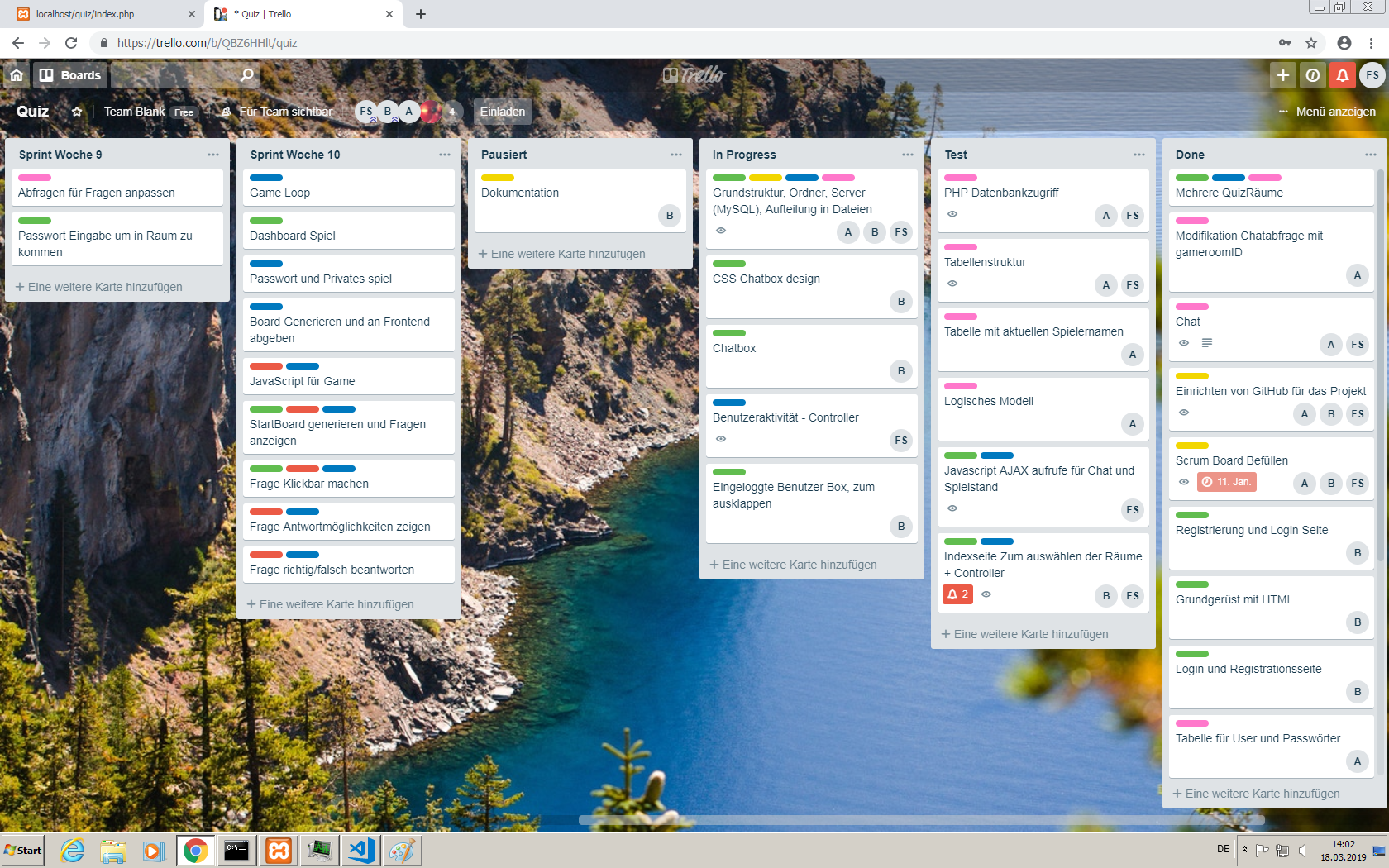


# Scrum-Board Übersichten

Woche 8



Woche 9-10



Woche 10

