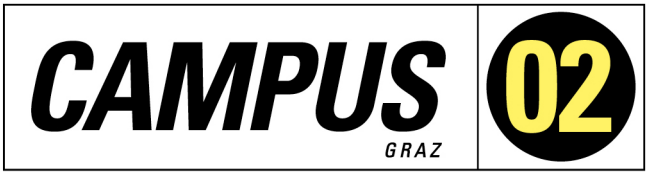

FACHHOCHSCHULE DER WIRTSCHAFT

PubQuiz-Master

Hartmann Ewald
Novosel Florian
Reisinger Lukas

Studiengang Wirtschaftsinformatik
FH Campus 02
Graz, Österreich

19. April 2022

Inhaltsverzeichnis

1 Allgemeine Beschreibung

Das Pubquiz - in vielen Grazer Lokalen ein beliebtes Mittel zum Zeitvertreib und Prokrastination für Student:innen. Die Vorbereitung dazu ist oftmals umfangreicher als vermutet. Das Projekt soll hierzu Abhilfe schaffen indem der oder die Benutzer:in auf eine bestehende Datenbank, in der Fragen inklusiver Antwortmöglichkeiten zu verschiedensten Themengebieten vorhanden sind, zugreifen kann. Weiters übernimmt die Anwendung auch die gesamte Validierung der gewählten Antwortmöglichkeit.

2 Anforderungsanalyse

- Der oder die Benutzer:in soll den Umfang (Anzahl der Fragen und dessen Schwierigkeit) sowie das jeweilige Themengebiet zu welches sich das Quiz handelt, selbst für jeden Durchlauf bestimmen können.
- Nach jeder Frage soll ein Feedback über die Richtigkeit der gegebenen Antwort dem oder der Benutzer:in geliefert werden.
- Zum Abschluss eines Quiz-Durchlaufes soll eine Statistik dem oder der Benutzer:in angezeigt werden.
- Der oder die Benutzer:in soll Voreinstellungen für die Quiz hinsichtlich Umfang und Kategorie innerhalb von JSON und XML Dateien hochladen können, beispielsweise um "Favoriten"direkt parat zu haben. Die Validität der Eingabedaten soll überprüft werden, um in weiterer Folge dem oder die Benutzer:in mittels Fehlermeldungen auf ungültige Strukturen oder Dateien hinzuweisen.

3 Systemarchitektur

Wie in der Abbildung 1 ersichtlich, dient die Businesslogik der Anwendung als zentraler Dreh- und Angelpunkt für die jeweiligen Benutzereingaben und darauffolgenden Rückmeldungen. Als Input können entweder vorgefertigte JSON und XML Files dienen, oder auch direkt durch den Benutzer über ein HTML-Formular. Die Validierung von XML Files wird über ein XSD Schema realisiert. Die Aufbereitung zur Anzeige des Eingabe XMLs übernimmt der XSLT Processor. Anfragen werden via Fetch-API an der OpentriviaDB (https://opentdb.com/) gestellt. Die Antwort wird als JSON geparsed und wiederum von der Businesslogik weiterverarbeitet.

4 Datenmodell

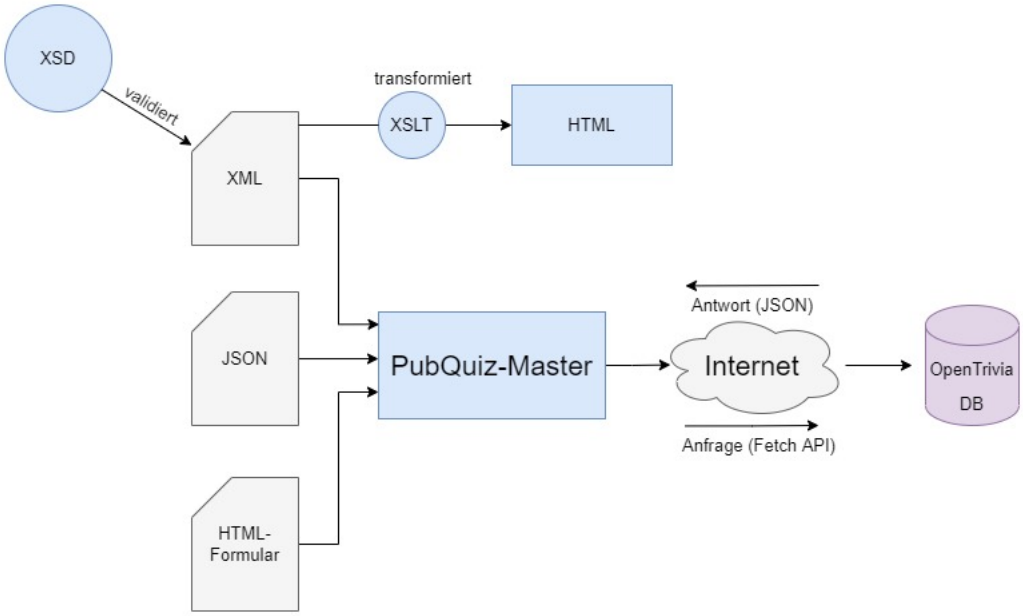

Abbildung 1: Systemarchitektur PubQuiz-Master

XML:

- **quizes:** 'quizes' beschreibt den Wurzelknoten. Dieser enthält eine beliebige Anzahl von Elementen des Typs 'Quiz', welches mindestens einmal vorkommen muss.
  - **quiz:** 'quiz' beschreibt einen jeweiligen Quizdurchlauf mit entsprechenden Einstellungen. Das Element beinhaltet die Werte 'amount', 'difficulty' und 'category'. Jedes dieser Elemente muss und darf genau einmal vorkommen.
    - **amount:** Integer Wert der die Anzahl an Fragen im Quiz definiert. Wertebereich 1 - 20. Beispielwert: 8
    - **difficulty:** String Wert der die Schwierigkeit im Quiz definiert. Mögliche Werte sind 'easy', 'medium' und 'hard'.
    - **category:** Integer Wert der die Kategorie im Quiz definiert. Der Wert beschreibt die jeweilige Kategorie-ID. Wertebereich 9-32. Beispielwert: 11

Beispiel XML:

```
<quizes>
    <quiz>
        <amount>9</amount>
        <difficulty>easy</difficulty>
        <category>12</category>
    </quiz>
</quizes>
```

Inhalte des JSON Dokumente beinhalten ein 'Quiz' Objekt, welche ebenfalls die obrigen Attribute inklusiver der im XML definierten erlaubten Wertebereiche.

Beispiel JSON:

```
{
    "Quiz": {
        "amount": 12,
        "difficulty": "easy",
        "category": 9
    }
}
```

5 Schnittstellenbeschreibung

Als Webservice wurde die von OpenTriviaDB zur Verfügung gestellte API verwendet. Die zugehörige Dokumentation kann hier abgerufen werden.

HTTP-Methode	Endpoint	Beschreibung
GET	/api_category.php	Einlesen aller verfügbaren Kategorien
GET	/api.php?amount=	Übermittlung der Anzahl an benötigten Fragen
GET	/api.php?cat=	Übermittlung der gewünschten Kategorie
GET	/api.php?difficulty=	Übermittlung der gewünschten Schwierigkeit
GET	/api.php?type=multiple	Gibt an dass nur Multiple-Choice Fragen gewünscht sind

6 Projektablauf

6.1 Tech Stack

Die grundlegende Businesslogik wurde in *Javascript* erstellt. Zur generellen Kommunikation mit der OpenTrivia Database wurde eine *Fetch API* verwendet. Das generelle Styling der Webapplikation erfolgte mittels *Tailwind CSS*.

6.2 Workflow

Zur gemeinsamen Arbeit am Code und der Versionsverwaltung wurde ein Gitlab Repository erstellt. Zur einfachen Abstimmung hat die Projektgruppe einen separaten Teams-Kanal angelegt, in dem die gesamte Kommunikation sowie diverse Statusmeetings abgehalten wurden. Die Abstimmung hierzu war sehr einfach und professionell da in dieser Gruppe bereits mehrfach Projekte umgesetzt wurden. Zu Beginn wurden mehrere Themen für mögliche Anwendungen gemeinsam gesucht und evaluiert. Im Anschluss erfolgte eine grobe Aufteilung der Arbeitspakete. Nachdem ein jeweils umfangreicherer Teil fertiggestellt wurde, gab es ein Synch-Meeting zwischen allen Projektmitgliedern.

6.3 Tasks

Ewald Hartmann übernahm den Hauptteil der Businesslogik und der Funktionalität rund um die API. FLorian Novosel war Hauptverantwortlicher für die Implementierung der Logik zur Verwendung von XML Dateien als Input sowie dessen Validierung mittels XSL sowie anschließender Transformation in ein HTML Dokument via XSL. Lukas Reisinger war Hauptverantwortlicher für die Erstellung der Projektdokumentation und Informationsbeschaffung zu LateX. Bei etwaigen aufkommenden Fragen von einzelnen Personen wurden diese jeweils kurzfristig innerhalb des Teams gemeinsam geklärt, bzw. bei Unklarheiten zum Projektumfang, mit dem Lektor der Lehrveranstaltung aufgelöst.