

# Entwicklerdokumentation für das Web-Quiz-Projekt

## Belegarbeit Internettechnologien

- Dozent: Herr Prof. Dr.-Ing. Jörg Vogt
- Abgabedatum: 02.06.2024
- Bearbeiter: Özer, Taylan

## Inhaltsverzeichnis

### 1. Projektübersicht

Das Web-Quiz-Projekt ist ein interaktives, webbasiertes Quizspiel, das in vier Themenbereiche unterteilt ist: Allgemeinwissen, Internettechnologien, Mathematik und Notenlehre. Ziel des Projekts ist es, Benutzern die Möglichkeit zu geben, ihr Wissen in diesen Bereichen auf unterhaltsame Weise zu testen und zu erweitern. Das Quiz ist modular aufgebaut und nutzt moderne Webtechnologien für eine optimale Benutzererfahrung.

Für das Design der Website wird eine CSS-Datei (mvp.css) verwendet, die für ein klares und ansprechendes Layout sorgt. Die Fragen und Antworten sind in einer JSON-Datei (data.JSON) gespeichert, die sowohl lokale Fragen als auch externe Fragen von einem HTW-Webserver einbindet. Die JavaScript-Dateien (main.js und mvp.js) enthalten die Hauptfunktionen des Spiels, wie das Laden der Fragen, die Verwaltung des Spielablaufs und die Anpassung der mathematischen Darstellungen mit KaTeX.

Ein Manifest-Ordner sorgt für die Erstellung eines Webicons, um die Benutzerfreundlichkeit auf mobilen Geräten zu verbessern. Das Projekt beinhaltet auch eine umfassende Dokumentation, die sowohl eine Anwenderdokumentation als auch eine Entwicklerdokumentation umfasst. Die index.html-Datei ist die zentrale Datei des Projekts und integriert alle notwendigen Ressourcen und Skripte. Insgesamt bietet das Web-Quiz-Projekt eine gut strukturierte, erweiterbare und benutzerfreundliche Plattform für Wissensspiele im Web.

### 2. Dateien und Ordner

#### 2.1 index.html

Die **index.html-Datei** ist das Hauptdokument für das Web-Quiz-Projekt und definiert die Struktur und das Layout der Webseite. Im Kopfbereich (<head>) sind wichtige Meta-Tags enthalten, die die Darstellung und Funktionalität der Seite auf mobilen Geräten optimieren. Dazu gehören Links zu den CSS- und JavaScript-Dateien, die das Design und die Funktionalität des Quiz bereitstellen.

Das <body>-Element umfasst den Hauptinhalt der Seite, einschließlich eines Headers, der eine Nachricht anzeigt, wenn JavaScript deaktiviert ist. Die Seite ist in verschiedene Container unterteilt,

darunter eine Navigationsleiste, einen Hauptbereich für das Spiel und eine Seitenleiste für Statusmeldungen.

Externe Bibliotheken wie VexFlow und KaTeX werden eingebunden, um die musikalischen und mathematischen Darstellungen zu ermöglichen. Die Manifest- und Icon-Links sorgen dafür, dass die Seite als Progressive Web App funktioniert. Im Footer werden die Urheberrechtsinformationen und der Name des Projekterstellers, Taylan Özer, angezeigt.

## 2.2 CSS

Die **mvp.css-Datei** definiert das visuelle Erscheinungsbild und das Layout der Web-Quiz-Seite. Der Hintergrund des gesamten Dokuments ist in einem hellen Orange-Ton (#FFBB98) gehalten, während das Hauptlayout durch Flexbox-Container strukturiert ist.

Der Header ist schwarz mit einem Bild-Hintergrund und weißem Text, wodurch er sich deutlich vom restlichen Inhalt abhebt. Die Navigationsleiste und Schaltflächen sind in einem Blauton gestaltet, der sich beim Hover-Effekt in ein helleres Grau ändert.

Das Hauptinhaltsbereich, die Seitenleiste und der Footer teilen eine gemeinsame hellorange Hintergrundfarbe und sind jeweils mit abgerundeten Ecken und einem Rand versehen. Für die Spielmechanik und das Layout der Quizfragen werden spezifische CSS-Klassen verwendet, die eine ansprechende und funktionale Benutzeroberfläche gewährleisten.

Die responsive Gestaltung der Seite wird durch mehrere Media Queries unterstützt, die das Layout je nach Bildschirmgröße anpassen, um eine optimale Benutzererfahrung auf verschiedenen Geräten zu ermöglichen

## 2.3 JavaScript

### 2.3.1. main.js

Die **main.js-Datei** enthält den JavaScript-Code zur Registrierung eines Service Workers für das Web-Quiz-Projekt. Sie beginnt mit dem Import einer Konfigurationsdatei (conf.js), aus der der Cache-Name (cacheName) abgerufen wird. Beim Laden der Seite wird geprüft, ob der Browser Service Worker unterstützt; wenn ja, wird der Service Worker (sw\_site.js) registriert, um die Offline-Funktionalität der Webseite zu gewährleisten. Die Datei definiert auch eine Funktion (getDataFromCache), die versucht, Daten aus dem Cache abzurufen, indem sie den Cache nach einer bestimmten URL durchsucht und, falls gefunden, die Daten als Blob zurückgibt. Zudem werden DOM-Elemente wie `netz_img` und `netz_text` referenziert, um den Netzwerkstatus anzuzeigen.

### 2.3.2 mvp.js

Die **mvp.js-Datei** implementiert das Model-View-Presenter (MVP) Pattern für das Web-Quiz-Projekt. Nach dem Laden der Seite wird ein Model, ein Presenter und eine View instanziiert und miteinander verknüpft. Das Model verwaltet die Daten und lädt sie sowohl aus internen JSON-Dateien als auch aus externen Quellen, einschließlich eines externen HTW-Servers. Die Presenter-Klasse koordiniert die Logik, darunter die Auswahl und Mischung der Fragen, das Verwalten des Timers und das Überprüfen der Antworten. Die View-Klasse ist für die Darstellung der

Benutzeroberfläche verantwortlich, einschließlich des Menüs, der Fragen und der Antwortbuttons. Beim Start eines Quiz wird die Benutzeroberfläche entsprechend eingerichtet, und die Fragen werden aus dem Modell abgerufen und angezeigt. Das System unterstützt sowohl Multiple-Choice-Fragen als auch externe Fragen, die vom HTW-Server geladen und überprüft werden. Die Interaktion mit den Fragen erfolgt durch Anklicken von Antwortbuttons, wobei der Presenter die Richtigkeit der Antworten überprüft und die Punktzahl entsprechend aktualisiert. Bei falschen Antworten oder abgelaufener Zeit wird das Spiel beendet und die entsprechenden Nachrichten angezeigt.

## 4 JSON

Diese Sammlung umfasst vier thematische Quizbereiche und einen Bereich für externe Fragen. Der Bereich "Allgemeinwissen" enthält Fragen zu verschiedenen Themen wie Geographie, Geschichte und berühmten Persönlichkeiten, um das Allgemeinwissen zu testen. "Internettechnologie" fokussiert sich auf technisches Wissen rund um das Internet, einschließlich Protokolle, Sicherheitsmechanismen und Webentwicklung. Der Bereich "Mathematik" stellt mathematische Aufgaben, die grundlegende Rechenoperationen und mathematische Konzepte abdecken. Im Abschnitt "Notenlehre" werden musikalische Noten und Akkorde abgefragt, um das Verständnis der Musiktheorie zu prüfen. Der Bereich "Externe HTW-Fragen" ist für spezifische Fragen von einem externen Server der HTW gedacht und woraus bereits vorher hochgeladene Fragen sich zusätzlich in das Web-Quiz hineinholen lassen.

## 5.Funktionalitäten

- Quizfragen zu verschiedenen Themenbereichen beantworten
- Auswahl von Antworten aus mehreren vorgegebenen Optionen
- Speichern und Auswerten der richtigen Antworten
- Integration externer Fragen in das bestehende Quizsystem

## 6.Technologien

Hier habe ich folgende Technologien verwendet:

- Microsoft Visual Studio Code (Entwicklungsumgebung),
- Ajax-Schnittstelle (HTW-Server),
- Katex,
- Vexflow,
- (Browser: Opera, Google Chrome)

## 7.Nutzung

# Anwenderdokumentation

## Belegarbeit Internettechnologien

- Dozent: Herr Prof. Dr.Ing. Jörg Vogt
- Abgabedatum: 02.06.2024
- Bearbeiter: Özer, Taylan

## Start der Webanwendung

- Öffne Visual Studio Code:

→ Live Server Extension installieren (gehe zu den Extensions, suche nach "Live Server" und installiere die Extension),

→ Verbindung zu GitHub herstellen (optional) (Stelle sicher, dass du eine Verbindung zu deinem GitHub-Repository hergestellt hast, falls du deine Änderungen verfolgen und versionieren möchtest)

### Allgemein

Das Web-Quiz umfasst folgende Themenbereiche:

- Allgemeinwissen,
- Internettechnologien,
- Mathematik,
- Musiktheorie/Notenlehre

### Aufgabenbereich Allgemeinwissen:

- Aufgaben zu verschiedenen Themen des Allgemeinwissens

### Aufgabenbereich Internettechnologien:

- Aufgaben zu verschiedenen Themen der Internettechnologien

### Aufgabenbereich Mathematik:

- Aufgaben zu verschiedenen mathematischen Themen

### Musiktheorie/Notenlehre:

- Aufgaben zur Musiktheorie/Notenlehre. Beispiel: Note C wird vorgegeben und der Nutzer, darf nach bestem Wissen diesen Note auswählen. Notenblatt mit Noten sind ebenfalls vorhanden.

### Weitere Informationen:

Jede Fragestellung bietet vier Auswahlmöglichkeiten, von denen nur eine immer richtig ist und mit

einer Punktanzeige visuell auch dargestellt wird. Am Ende des Quiz wird über eine Progressbar angezeigt, wie viele Punkte du erreicht hast und ob du gewonnen oder verloren hast. Neustarten ist auch möglich, auf den für hier vorgesehenen Button.

### **Verbesserungsvorschläge:**

Im allgemeinen lässt sich folgendes zum Thema Verbesserungen sagen: Eine bessere Strukturierung des CSS-Designs ist durchaus denkbar, wenn man viel Wert auf die optische Ansicht und die User-Experience legt.

### **Bekannte Probleme bzw. Auffälligkeiten:**

- keine

### **Browser:**

- Auf allen hier genannten Browsern lief die Webanwendung problemlos: Mozilla Firefox, Google Chrome, Opera

## **8. Erweiterungsmöglichkeiten**

Dieses Projekt bietet eine solide Grundlage für ein webbasiertes Quiz, das leicht erweitert und an verschiedene Anforderungen angepasst werden kann. Die modulare Struktur ermöglicht eine einfache Wartung und Erweiterung des Codes, während die Verwendung von Standardtechnologien wie HTML, CSS und JavaScript eine breite Kompatibilität sicherstellt. Durch die Einbindung von Katex und VexFlow können komplexe mathematische und musikalische Inhalte ansprechend dargestellt werden.

Für zukünftige Entwicklungen könnten zusätzliche Features wie Benutzeranmeldungen, erweiterte Statistikfunktionen und soziale Features (z.B. Teilen von Ergebnissen) implementiert werden, um die Benutzererfahrung weiter zu verbessern.