ProjektDokumentation

Elvis Bruhin, Thikal Vincent Robert, Kenan Bajramovic

Berufsfachschule Baden

LA1304

# Versionen der Applikation

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Autor | Bemerkung |
| 3.2.23 | 0.1.0 | Alle Zusammen | Keine |
| 24.2.23 | 0.2.0 | Elvis Bruhin | Keine |
| 3.3.23 | 0.3.0 | Thikal Vincent Robert | Keine |
| 10.323 | 0.4.0 | Elvis Bruhin | Kleiner Rückstand bei der Funktion „Check Answer“ |
| 17.3.23 | 0.5.0 | Kenan Bajramovic | Keine |
| 24.3.23 | 1.0.0 | Alle Zusammen | Keine |

# Informieren

Bei diesem Projekt haben wir uns für ein allgemeines Quiz-Game entschieden.

Von der Website kopiert https://getcssscan.com/css-buttons-examples \*/  
<https://www.youtube.com/watch?v=riDzcEQbX6k>

# Anforderungsliste

|  |  |  |  |
| --- | --- | --- | --- |
| ***Anf.-Nr.*** | ***Muss/Kann*** | ***Bedingungsart*** | ***Anforderung*** |
| 1 | Muss | Funktional | Die Quiz Game Website soll in der Lage sein, Fragen anzuzeigen, |
| 2 | Kann | Funktional | Der Nutzer kann zwischen vier Blöcke eine auswählen. |
| 3 | Muss | Qualitativ | Der richtige Block wird mit grün markiert, falls der Nutzer draufklickte. |
| 4 | Muss | Qualitativ | Falls der Nutzer auf einen falschen Block geklickt hatte, dann wird dieser mit rot markiert. |
| 5 | Kann | Qualitativ | Oben rechts auf der Website wird die Anzahl richtige Antworten aufgezeigt. |
| 6 | Muss | Qualitativ | Das Projekt soll über VS-Code programmiert werden. |
| 7 | Muss | Qualitativ | Das Projekt soll mit dem IPERKA-Verfahren durchgeführt werden. |
| 8 | Muss | Rand | Das Grundgerüst der Quiz Game Website soll mit Hilfe von HTML und der funktionale Teil mit JavaScript erstellt werden. |
| 9 | Muss | Rand | Das Design der Quiz Game Website soll mit Hilfe von CSS erstellt werden. |
| 10 | Muss | Funktional | Das Backend funktioniert auf das ganze Programm. |

# Entscheiden

Als Datenbank für den Lernauftrag 1304 haben wir uns für MongoDB entschieden. Wir haben die Mongoose-Library zur Modellierung von Objektdaten verwendet, um verschiedene Beziehungen zu verwalten. Um die Vermeidung von Anmeldedaten in einem für jedermann zugänglichen Online-Repository zu sichern, haben wir .env-Files verwendet.

# Testfallspezifikationen (Planung)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nummer | Nummer Anforderung | Voraussetzung | Eingabe | Erwartete Ausgabe |
| 1. | 1. | Man ist auf der Website | Klick auf Button „You want to play lets play“ | Anzeige von Frage und vier Antwortfelde |
| 2. | 2,3 | Nr 1. Wurde erfüllt | Man drückt bei der Frage „Whats the opposite of hot“ auf „cold“ | Button wird kurz grün und nächste Frage wird angezeigt |
| 3. | 2,4 | Nr 1. Wurde erfüllt | Man drückt bei der Frage „Whats the opposite of hot“ auf „wet“ | Button wird kurz rot und nächste Frage wird angezeigt |
| 4. | 10 | Man hat VS-Code offen und ist im Terminal auf dem Pfad bei welchem mongodb.js drinnen ist. | node mongodb.js | Server running on Port 3000  Connected to Database |
| 5. | 5 | Man ist auf der Website | Den richtigen Block anklicken | Punktzahl wird geändert |

## Testprotokoll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nummer | Testfall NR. | Datum | Resultat | Durchgeführt |
| 1. | 1 | 24.3.2023 | OK | Kenan Bajramovic |
| 2. | 2 | 24.3.2023 | OK | Kenan Bajramovic |
| 3. | 3 | 24.3.2023 | OK | Kenan Bajramovic |
| 4. | 4 | 24.3.2023 | OK | Kenan Bajramovic |
| 5. | 5 | 24.3.2023 | NOK | Kenan Bajramovic |

## Reflexion

Anfangs gab es einige Schwierigkeiten bei der Arbeit mit MongoDB die uns in den Rückstand gebracht haben, letztendlich konnten wir diese dann mehr oder weniger aufholen.