Lernatelier: Projektdokumentation

Martic, Tanaskovic

# Informieren

## Ihr Projekt

Eine Webapplikation, in welcher man Vocis zusammenstellen kann und mit denen dann lernen kann.

## Quellen

<https://programmersportal.com/how-to-fit-an-image-in-a-circle-using-css/>

<https://itmodul.ch/m294/>

<https://blog.hubspot.com/website/hide-scrollbar-css>

<https://www.w3schools.com/tags/tag_textarea.asp>

<https://www.w3schools.com/howto/howto_js_trigger_button_enter.asp>

<https://betterprogramming.pub/how-to-restart-a-css-animation-with-javascript-and-what-is-the-dom-reflow-a86e8b6df00f>

<https://www.mongodb.com/languages/express-mongodb-rest-api-tutorial>

<https://www.simplilearn.com/tutorials/nodejs-tutorial/body-parser-in-express-js>

1.3. Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
|  | Muss | Rand | Mit HTML, CSS und JavaScript implementiert |
|  | Muss | Funktional | Der Nutzer kann ein Voci erstellen |
|  | Muss | Funktional | Der Nutzer kann seine Vocis sehen |
|  | Muss | Funktional | Der Nutzer kann mit einem seiner Vocis lernen. |
|  | Muss | Funktional | Der Nutzer kann wechseln, welche Sprache er übersetzen muss. |
|  | Muss | Funktional | Der Nutzer sieht das Resultat seiner Lernsitzung. |
|  | Muss | Funktional | Die Seite verfügt Client Side Routing. |
|  | Muss | Funktional | Die Seite interagiert mit einer MongoDB Datenbank |
|  | Rand | Qualität | Die Webseite beinhaltet CSS-Animationen. |

1.4 Testfälle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Anforderung** | **Voraussetzung** | **Eingabe** | **Erwartete Ausgabe** |
| 1.1 | 1 | Webseite geöffnet | Webseite geöffnet | Webseite bearbeitet mit HTML und CSS |
| 2.1 | 2 | Klick auf "Create" | Name der Liste: "Englisch" Wörtereingabe: ""  Klick auf Submit | "Es muss etwas angegeben werden." |
| 2.2 | 2 | " | Name der Liste: ""  Wörtereingabe:"Apple|Apfel"  Klick auf Submit | "Es muss etwas angegeben werden." |
| 2.3 | 3 | " | Name der Liste: "Englisch"  Wörtereingabe:"Apple|Apfel"  Klick auf Submit | Leitet zurück auf Homepage |
| 3.1 | 3 | Homepage offen Noch kein Voci erstellt | Klick auf "Study" | Leere Seite |
| 3.2 | 3 | Homepage offen  Mindestens ein Voci erstellt | Klick auf "Study" | Alle Vocis mit Titel und maximal 3 Begriffen aus dem Voci "Englisch Apple|Apfel  Baum|Tree  Tisch|Table  …" |
| 4.1 | 4 | Homepage offen Mindestens ein Voci erstellt Klick auf "Study" | Klick auf eine der Listen | Zufälliges Wort aus der List wird angezeigt. |
| 4.2 | 4 | Homepage offen Mindestens ein Voci erstellt Klick auf "Study"  Klick auf eine der Listen | "Banane" | "Falsch" Lösung wird angezeigt Frage wird später nochmal gestellt. |
| 4.2 | 4 | " | "Apfel" | "Richtig" Frage wird nicht noch einmal gestellt |
| 5.1 | 5 | " | Klick auf den Umkehrbutton | Quiz startet von vorne Diesmal stehen deutsche Begriffe in den Voci Kärtchen. |
| 6.1 | 6 | " | Alle Fragen richtig beantworten | "Du hast 7/7 beim ersten Mal richtig gehabt." |
| 6.2 | 6 | " | Alle bis auf 2 Fragen richtig beantworten | "Du hast 5/7 beim ersten Mal richtig gehabt" Wörter die man falsch gemacht hat werden in einem Format ausgegeben, sodass man daraus wieder ein neues voci erstellen kann. |
| 7.1 | 7 | Homepage ist offen | Klick auf "Study" | link der Seite wird um #lists erweitert |
| 7.2 | 7 | " | "#lists" zum Link hinzufügen | List Seite wird geöffnet |
| 7.3 | 7 | " | "#adasd" zum Link hinzufügen | Error-Seite wird gezeigt. |
| 8.1 | 8 | Homepage ist offen  Klick auf Create | Erstelle ein Voci | Voci wird zur Datenbank hinzugefügt |
| 8.2 | 8 | Homepage ist offen | Klick auf Study | Listen werden aus einer MongoDB Datenbank gezogen. |
| 8.3 | 8 | Listen werden angezeigt | Klick auf "Delete" | Voci wird aus Datenbank gelöscht |
| 8.4 | 8 | Listen werden angezeigt | Klick auf Edit | Edit Seit öffnet sich und alle Wörter der Liste werden ausgegeben. |
| 8.5 | 8 | Listen werden angezeigt Klick auf Edit bei einer Liste | Daten bearbeiten | Datensatz wird aktualisiert. |
| 9.1 | 9 | Homepage offen Mindestens ein Voci erstellt Klick auf "Study"  Klick auf eine der Listen | "Apple" | Richtig Voci Kärtchen wischt sich nach links. |
| 9.2 | 9 | " | "Banana" | Falsch. Voci Kärtchen wischt nach unten. |
| 9.3 | 9 | " | Klick auf Umkehrbutton | Knopf dreht sich. |

1. Planen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zuständiger** |
| **1.1** | **01.03.2023** | **HTML und CSS vorbereiten** | **1h** | **David und Igor** |
| **2.1** | **01.03.2023** | **Client Side Routing implementieren** | **1h** | **Igor** |
| **3.1** | **01.03.2023** | **Voci erstell Funktion implementieren** | **1h** | **David** |
| **4.1** | **01.03.2023** | **Listenanzeige implementieren** | **1h** | **Igor und David** |
| **5.1** | **08.03.2023** | **Lernfunktion implementieren** | **1h** | **Igor** |
| **6.1** | **08.03.2023** | **Resultate implementieren** | **1h** | **David** |
| **7.1** | **08.03.2023** | **Sprachwechsel implementieren** | **30min** | **Igor** |
| **8.1** | **15.03.2023** | **CSS-Animationen** | **30min** | **Igor und David** |
| **9.1** | **15.03.2023** | **Verbindung und Interaktion mit MongoDB implementieren** | **3h** | **Igor und David** |

1. Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** |
| **1.1** | **01.03.2023** | **HTML und CSS vorbereiten** | **1h** | 1h |
| **2.1** | **01.03.2023** | **Client Side Routing implementieren** | **1h** | 30min |
| **3.1** | **01.03.2023** | **Voci erstell Funktion implementieren** | **1h** | 1h |
| **4.1** | **01.03.2023** | **Listenanzeige implementieren** | **1h** | 45min |
| **5.1** | **08.03.2023** | **Lernfunktion implementieren** | **1h** | 1h |
| **6.1** | **08.03.2023** | **Resultate implementieren** | **1h** | 30min |
| **7.1** | **08.03.2023** | **Sprachwechsel implementieren** | **30min** | 15min |
| **8.1** | **15.03.2023** | **CSS-Animationen** | **30min** | 30min |
| **9.1** | **15.03.2023** | **Verbindung und Interaktion mit MongoDB implementieren** | **3h** | 3h |

1. Kontrollieren

## Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Testumgebung:** | **Laptop, Liveserver der Webseite, Google Chrome** | | |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| 1.1 | **22.03.2023** | **OK** | Igor |
| 2.1 | **22.03.2023** | **OK** | Igor |
| 2.2 | **22.03.2023** | **OK** | Igor |
| 2.3 | **22.03.2023** | **OK** | Igor |
| 3.1 | **22.03.2023** | **OK** | Igor |
| 3.2 | **22.03.2023** | **OK** | Igor |
| 4.1 | **22.03.2023** | **OK** | David |
| 4.2 | **22.03.2023** | **OK** | David |
| 4.2 | **22.03.2023** | **OK** | David |
| 5.1 | **22.03.2023** | **OK** | David |
| 6.1 | **22.03.2023** | **OK** | David |
| 6.2 | **22.03.2023** | **OK** | David |
| 7.1 | **22.03.2023** | **OK** | Igor |
| 7.2 | **22.03.2023** | **OK** | Igor |
| 7.3 | **22.03.2023** | **OK** | Igor |
| 8.1 | **22.03.2023** | **OK** | Igor |
| 8.2 | **22.03.2023** | **OK** | Igor |
| 8.3 | **22.03.2023** | **OK** | Igor |
| 8.4 | **22.03.2023** | **OK** | Igor |
| 8.5 | **22.03.2023** | **OK** | Igor |
| 9.1 | **22.03.2023** | **OK** | David |
| 9.2 | **22.03.2023** | **OK** | David |
| 9.3 | **22.03.2023** | **OK** | David |

Fazit

Alle Testfälle verlaufen wie geplant. Das Projekt ist demnach abgabebereit.

1. Auswerten

Wir haben unsere Prioritäten falsch gesetzt. Wir haben die Seite zuerst erstellt, um dann die MongoDB API im Nachhinein hinzugefügt. Stattdessen hätten wir die Seite auf den Code, der für MongoDB nötig gewesen wäre aufbauen sollen. So wie wir es gemacht haben, kostete es uns Zeit den Code umzuschreiben damit er klappt.