Projektdokumentation epiphanies

Delia Reho

|  |  |  |
| --- | --- | --- |
| **Datum** | **Version** | **Änderung** |
| 22.12.2022 | 1.1 | Krankheitstag |
| 12.01.2023 | 1.2 | Informieren |
| 19.01.2023 | 1.3 | Informieren |
| 26.01.2023 | 1.4 | Firebase aufsetzen |
| 02.02.2023 | 1.5 | Grobes HTML geschrieben |
| 23.02.2023 | 1.6 | Leichte CSS Bearbeitung |
| 02.03.2023 | 1.7 | Login Logik |
| 09.03.2023 | 1.8 | Music Player Logik |
| 16.03.2023 | 1.9 | Music Player Logik |
| 19.03.2023 | 1.10 | Testen |

# 

# Inhaltsverzeichnis

[Inhaltsverzeichnis 1](#_Toc118360321)

[1. Informieren 2](#_Toc118360322)

[1.1 Projekt 2](#_Toc118360323)

[1.2 Quellen 2](#_Toc118360324)

[1.3 Anforderungen 3](#_Toc118360325)

[2. Planen 4](#_Toc118360327)

[2.1 Arbeitspakete/To-Do 4](#_Toc118360328)

[3. Entscheiden 5](#_Toc118360330)

[4. Realisieren 6](#_Toc118360331)

[5. Kontrollieren 6](#_Toc118360332)

[5.1 Testfälle 7](#_Toc118360333)

[5.2 Testprotokoll 7](#_Toc118360334)

[5.3 Testfazit 7](#_Toc118360335)

[6. Auswerten 7](#_Toc118360336)

# Informieren

## Projekt

Ich möchte einen Music Player erstellen, bei dem man eigene Dateien einfügen kann und dann diese abspielen lassen kann. Diese Audios sollten in einer Speicherablage gespeichert werden und dann wenn man sich einloggt sollen die Lieder schon dort sein. Man kann Lieder entfernen und hinzufügen. Wenn möglich, kann man auch mehrere Playlists erstellen.

## Quellen

* [Alles was mit Firebase zu tun hat](https://firebase.google.com/)
* [Audio Files abspielen](https://www.youtube.com/watch?v=b89pR5L4qZc) und [in Playlist sortieren](https://youtu.be/Er6gIU_BN4s)

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | Muss | Funktional | Hinzugefügte Lieder werden auf der Webseite dargestellt |
| 2 | Muss | Funktional | Man kann das Lied pausieren |
| 3 | Muss | Funktional | Man kann das Spiel abspielen lassen |
| 4 | Muss | Funktional | Man kann zum nächsten Lied springen |
| 5 | Muss | Funktional | Man kann zum vorherigen Lied springen |
| 6 | Muss | Funktional | Der Name des Lieds, welches gerade läuft wird angezeigt |
| 7 | Muss | Funktional | Man kann sich mit Username, E-Mail und Password registrieren |
| 8 | Muss | Funktional | Die Registrationsdaten werden in der Datenbank gespeichert |
| 9 | Muss | Funktional | Die eingefügten Lieder werden in der Firebase Storage Ablage abgelegt. |

# Planen

## Arbeitspakete/To-Do

Erster Arbeitstag, 22. Dezember 2022 konnte nichts erledigt werden, da ich an Corona erkrankt war.

|  |  |  |  |
| --- | --- | --- | --- |
| **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** |
| 12.01.2023 | Dokumentation erstellen und IPE ausfüllen | 90 | 90 |
| 12.01.2023 | Entscheidungen für Sprache treffen | 45 | 90 |
| 12.01.2023 | Komptabilität von Firebase und HTML, CSS und Javascript absichern | 45 | 45 |
| 19.01.2023 | Informieren wie man einen Music Player aufsetzt, ob man mp3 Files in eine Firebase speichern kann usw. | 90 | 210 |
| 26.01.2023 | Firebase aufsetzen | 90 | 210 |
| 02.02.2023 | HTML schreiben | 90 | 210 |
| 23.02.2023 | Grobes GUI erstellt, CSS bearbeitet | 100 | 210 |
| 02.03.2023 | Login umgesetzt | 210 | 210 |
| 09.03.2023 | Music Player umgesetzt | 210 | 210+ |
| 16.03.2023 | Music Player umgesetzt | 210 | 210 |
| 19.03.2023 | Testen | 50 | 30 |
| 19.03.2023 | Erweiterung Dokumentation | 30 | 20 |
| 19.03.2023 | Portfolio | 60 |  |

# Entscheiden

Farben

|  |
| --- |
| **Hex Codes** |
| #f3ccbd |
| #eebcbb |
| #e7acc2 |
| #d6b5d4 |
| #c9bcda |
| #b2c9e9 |

# Realisieren

Die Datenbank wird kostenlos auf Firebase gehostet.

# Kontrollieren

Ich teste nur die Anforderungen, die erfüllt wurden und nicht zur Kategorie «*Rahmenbedingungen*» zählen:

|  |  |
| --- | --- |
| **Anf. Nr.** | **Beschreibung** |
| 1 | Hinzugefügte Lieder werden auf der Webseite dargestellt |
| 2 | Man kann das Lied pausieren |
| 3 | Man kann das Spiel abspielen lassen |
| 4 | Man kann zum nächsten Lied springen |
| 5 | Man kann zum vorherigen Lied springen |
| 7 | Man kann sich mit Username, E-Mail und Password registrieren |
| 8 | Die Registrationsdaten werden in der Datenbank gespeichert |

## Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Voraussetzung** | **Eingabe** | **Erwartete Ausgabe** |
| 1.1 | Benutzer ist eingeloggt | 1. «add songs» drücken  2. «test.mp3» einfügen | 3. «test.mp4» wird in der Liste angezeigt |
| 2.1 | Benutzer ist eingeloggt | 1. «add songs» drücken  2. «test.mp3» einfügen  3. Abspiel-Knopf drücken  4. Pausier-Knopf drücken | 5. Lied stoppt |
| 3.1 | Benutzer ist eingeloggt | 1. «add songs» drücken  2. «test.mp3» einfügen  3. Abspiel-Knopf drücken | 4. Lied wird abgespielt |
| 4.1 | Benutzer ist eingeloggt | 1. «add songs» drücken  2. «test.mp3» und «test2.mp3» einfügen  3. «next» Knopf drücken | 4. «test2.mp3» wird abgespielt |
| 5.1 | Benutzer ist eingeloggt | 1. «add songs» drücken  2. «test.mp3» und «test2.mp3» einfügen  3. «next» Knopf drücken  4. «previous» Knopf drücken | 5. «test.mp3» wird abgespielt |
| 7.1 | Registrations-Formular ist geöffnet | 1. «beispiel» als username eingeben  2. «beispiel@test.ch» als e-mail eingeben  3. «beispielpasswort» als password eingeben | 4. Wird in Datenbank gespeichert  5. Benutzer wird zum Music Player geschickt |
| 8.1 | Registrations-Formular ist geöffnet | 1. «beispiel» als username eingeben  2. «beispiel@test.ch» als e-mail eingeben  3. «beispielpasswort» als password eingeben | 4. Wird in Datenbank gespeichert |

## Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| **1.1** | 19.03.2023 | **OK** | Delia |
| **2.1** | 19.03.2023 | **OK** | Delia |
| **3.1** | 19.03.2023 | **OK** | Delia |
| **4.1** | 19.03.2023 | **OK** | Delia |
| **5.1** | 19.03.2023 | **OK** | Delia |
| **7.1** | 19.03.2023 | **OK** | Delia |
| **8.1** | 19.03.2023 | **OK** | Delia |

## Testfazit

Alle getesteten Anforderungen funktionieren, alle nicht getesteten Funktionieren nicht.

# Auswerten

* *Siehe Portfolios*