ToEnder

Todo / Kalender Web

Schneeberger Patrick

Inhaltsverzeichnis

[1.0 Projektbeschreibung 3](#_Toc60911144)

[1.1 Vorwort 3](#_Toc60911145)

[1.2 Ausgangsituation 3](#_Toc60911146)

[1.3 Umsetzung 3](#_Toc60911147)

[1.4 Ergebnis 3](#_Toc60911148)

[2.0 Planung 4](#_Toc60911149)

[2.1 Grobplanung 4](#_Toc60911150)

[2.1 Arbeitsjournal 4](#_Toc60911151)

[2.2 Probleme 5](#_Toc60911152)

[2.3 User Stories 5](#_Toc60911153)

[3.0 Mockup 6](#_Toc60911154)

[4.0 Testen 7](#_Toc60911155)

[5.0 Limitierungen der Webseite 8](#_Toc60911156)

[6.0 Schlusswort 8](#_Toc60911157)

[7.0 Anhang 9](#_Toc60911158)

[7.1 Continuous Deployment Einrichten 9](#_Toc60911159)

[7.1.1 Github 9](#_Toc60911160)

[7.1.2 Deploy.yml 9](#_Toc60911161)

1. Projektbeschreibung
   1. Vorwort

Im Modul-150 haben wir den Auftrag bekommen eine Projekt umzusetzen das wie eine praktische Arbeit im Betrieb gewertet wird. Dies Arbeit dient zur Vorbereitung auf unsere praktische Arbeit im Betrieb.

* 1. Ausgangsituation

Im Geschäft programmiere ich mit Delphi. Wir entwickeln Client-Applikationen. Somit habe ich fast keine Erfahrung in der Webentwicklung. Einfache Grundlagen von HTML/CSS sind mir bekannt. Javascript ist für mich neu. Ich werde dieses Projekt alleine durchführen.

* 1. Umsetzung

Da ich in der Webentwicklung fast keine Erfahrung habe werde ich eine statische Webseite erstellen. Entsprechende Frameworks zu suchen und zu lernen wird viel zu zeitaufwendig.

Dazu sollen wir Git als Versionsverwaltung verwenden. Weiter sollen wir CI/CD einrichten.

Daher werde ich mich auf diese zwei Punkte konzentrieren.

* statische Webseite (html/js/css)
* Git
* CI/CD
  1. Ergebnis

Mir sind Begriffe wie CI/CD unbekannt. Auch mit Git habe ich wenig Erfahrung.

Mein Ziel ist es mich mit der Webentwicklung vertrauter zu machen. Daher habe ich mir überlegt eine Webseite zu erstellen bei der ich Aufgaben erfassen und Darstellen kann. Dies möchte ich mit einem Kalender kombinieren.

Github: <https://github.com/sInt3g3r/ToEnder>

1. Planung
   1. Grobplanung

|  |  |  |
| --- | --- | --- |
| Datum | Arbeit | Geschätzt |
| 19.11.2020 | Dokumentation vorbereiten,  Planung, Beginn mit dem Projekt, Mokup erstellen, User-Stories erstellen | 3h |
| 26.11.2020 | ToDo erfassen fertig | 3h |
| 03.12.2020 | ToDo bearbeiten, löschen, Status fertig | 3h |
| 10.12.2020 | ToDo's einem Datum zuweisen und anzeigen | 3h |
| 17.12.2020 | Projekt beenden. Beginn mit dem Manuellen Test | 3h |
| 17.12.2020  Hausaufgabe | Testen des Projekts. Bugfixing | 3h |
| 07.01.2021 | Dokumentation / Abgabe | 3h |
|  | **Total** | **21h** |

* 1. Arbeitsjournal

|  |  |  |
| --- | --- | --- |
| Datum | Arbeit | Dauer |
| 19.11.2020 | Continuous Integration  Continuous Deployment grob kennen gelernt.  Angefangen dieses einzurichten. | 3h |
| 26.11.2020 | Continuous Deployment fertig eingerichtet.  Beginn mit der Webseite | 3h |
| 03.12.2020 | add.html erstellt für das Erfassen von ToDo's | 3h |
| 10.12.2020 | add.html funktion fertig in Local Storage. Mit der Darstellung des Kalenders begonnen | 3h |
| 17.12.2020 | Leere Cards werden in der Monatsansicht dargestellt  -> Mehrere Tasks am selben Tag ? | 3h |
| 21.12.2020 | Mehrere Tasks am selben Tag darstellen mit ID, nicht fertig | 1.5h |
| 23.12.2020 | Mehrere Tasks am selben Tag darstellen mit ID, Fertig  Löschen eines Task funktioniert.  Umbennenen von Card / Cards zu task / tasks | 1.5h |
| 26.12.2020 | Editieren Funktion angefangen | 45min |
| 01.01.2021 | Editieren Funktion beendet | 4h |
| 02.01.2021 | Dialoge beim löschen hinzugefügt. Aktualisieren des Monats nach dem löschen hinzugefügt. Style änderungen an Buttons vorgenommen | 3h |
| 04.01.2021 | Dokumentation überarbeitet | 2h |
| 05.01.2021 | Tests bei der Dokumentation erfasst | 1h |
| 06.01.2021 | Status Inikator hinzugefügt | 1h |
|  | Zwischenresultat | 29h 45m |
| 07.01.2021 | Dokumentation überarbeitet, Style überarbeitet,  Bugs gefixt, Debug code entfernt, Bereit zur Abgabe | 3h |

* 1. Probleme

26.11.2020 - Continous Integration benötigt zu viel Zeit einzurichten.

Somit werde ich dies nicht einrichten.

17.12.2020 - Es wurde nur einen Task an einem Tag angezeigt. Dies ist nicht realistisch.

Es sollten mehrere Tasks an einem Tag angezeigt werden können.

Dazu benötige ich zum Editieren oder löschen eine ID um einen Taks eindeutig zu identifizieren.

05.01.2021 - Tests werden Manuell gemacht aufgrund Zeitmangels.

* 1. User Stories

Beispiel:

**"Als [Kundentyp] [möchte] ich, [damit]."**

*Als Max möchte ich meine Freunde einladen, damit wir diesen Service gemeinsam nutzen können.*

*Als Sascha möchte ich meine Arbeit organisieren, damit ich mehr Kontrolle darüber habe.*

Die User Stories werden im Github repository als Issues erfasst.

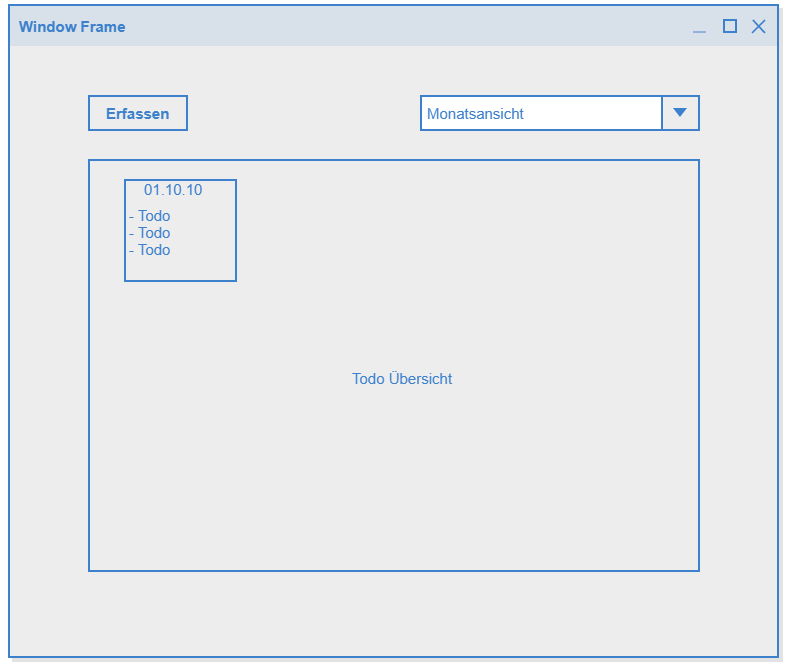
<https://github.com/sInt3g3r/ToEnder>

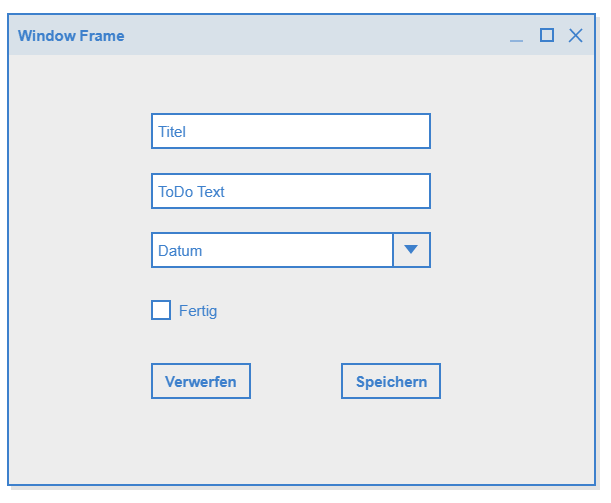
|  |  |  |
| --- | --- | --- |
| **Nummer** | **Story** | **Gewicht** |
| 1 | Als Benutzer möchte ich eine Arbeit erfassen können. | 1 |
| 2 | Als Benutzer möchte ich eine Arbeit einem Datum zuweisen können. | 1 |
| 3 | Als Benutzer möchte ich eine Übersicht der Arbeiten eines Monats sehen. | 2 |
| 4 | Als Benutzer möchte ich eine Übersicht meiner Arbeiten an einem Tag sehen. | 2 |
| 5 | Als Benutzer möchte ich eine Arbeit löschen können. | 3 |
| 6 | Als Benutzer möchte ich eine Arbeit Bearbeiten können | 4 |
| 7 | Als Benutzer möchte ich der Arbeit einen Status zuweisen können. | 5 |
| 8 | Als Benutzer möchte ich bei den Übersichten offene/fertige Arbeiten sehen. | 5 |
| 9 | (Optional) Die erfassten Arbeiten sollen persistent gespeichert werden | 6 |

Je kleiner die Gewichtung desto wichtiger ist des diese User-Story zu implementieren.

Somit werde ich nach der Reihenfolge der Gewichtung die einzelnen Features implementieren.

1. Mockup





1. Testen

Aufgrund des Zeitaufwandes habe ich mich dazu entschieden die Webseite manuell zu Testen.

Wie auf dem Arbeitsjournal zu sehen ist habe ich 6h gebraucht um das Continuous Deployment einzurichten. Daher verzichte ich auf Continuous Integration.

|  |  |  |
| --- | --- | --- |
| **Nummer** | **Story** | **Gewicht** |
| 1 | Als Benutzer möchte ich eine Arbeit erfassen können. | 1 |
| 2 | Als Benutzer möchte ich eine Arbeit einem Datum zuweisen können. | 1 |
| 3 | Als Benutzer möchte ich eine Übersicht der Arbeiten eines Monats sehen. | 2 |
| 4 | Als Benutzer möchte ich eine Übersicht meiner Arbeiten an einem Tag sehen. | 2 |
| 5 | Als Benutzer möchte ich eine Arbeit löschen können. | 3 |
| 6 | Als Benutzer möchte ich eine Arbeit Bearbeiten können | 4 |
| 7 | Als Benutzer möchte ich der Arbeit einen Status zuweisen können. | 5 |
| 8 | Als Benutzer möchte ich bei den Übersichten offene/fertige Arbeiten sehen. | 5 |
| 9 | (Optional) Die erfassten Arbeiten sollen persistent gespeichert werden | 6 |

|  |  |  |
| --- | --- | --- |
| **Story** | **Test beschreibung** | **Erfolgreich** |
| 1 | 1) Auf "Erfassen" klicken  2) Titel : "Abwaschen" Beschreibung: "Alle Teller abwaschen" Datum: "Heute"  3) Auf "Speichern" klicken.  Wenn nun im "Local Storage" diese Einträge vorhanden sind war der Test erfolgreich. | JA |
| 2 | Siehe Test 1 | JA |
| 3 | Beim klicken auf einen Monat werden die Tage als Kärtchen angezeigt. | JA |
| 4 | 1) Arbeit wie bei Test 1 Erfassen 2) Weitere Arbeit mit folgenden Werten erfassen: Titel: "Staubsaugen" Beschreibung: "Wohnzimmer" Datum: "Heute"  3) Die gleichen Arbeiten nochmals für Morgen erfassen. Der Test ist erfolgreich wenn auf der index.html alle 4 Aufgaben am richtigen Tag dargestellt werden. | JA |
| 5 | 1) Arbeit Erfassen wie bei Test 4  2) Im Local Storage die ID eines Eintrags auslesen und merken.  3) Auf "Löschen" klicken beim entsprechenden Eintrag.  Der Test ist ist erfolgreich wenn im "Local Storage" der korrekte Eintrag nicht mehr vorhanden ist. | JA |
| 6 | 1) Arbeit wie bei Test 1 Erfassen  2) Auf "Bearbeiten" klicken  3) Folgende Werte ändern:  Titel: "Abwaschen Wichtig"  Beschreibung: "Alle Teller und Gabeln abwaschen"  Datum: "Morgen"  4) Speichern  Der Test ist erfogreich wenn der Eintrag abgeändert wurde.  Kann im "Local Storage" geprüft werden. | JA |
| 7 | 1) Arbeit wie bei Test 1 erfassen aber als Abgeschlossen.  2) Speichern  Der Test ist erfogreich wenn "done" im Eintrag auf 1 gesetzt wurde.  Wieder im Local Storage überprüfen. | JA |
| 8 | 1) Zwei Arbeiten erfassen wie bei Test 4 Wobei die zweite Arbeit auf "Fertig" gesetzt wird.  Die Arbeit die auf fertig gesetzt wrude sollte einen grünen Punkt aufweisen. Die andere Arbeit wird mit einem orangen Punkt angezeigt. | JA |
| 9 | Nicht implementiert | NEIN |

1. Limitierungen der Webseite

* Erfasste Aufgaben sind nur im Browser gespeichert
* Es können nur Aufgaben im aktuellen Jahr angezeigt werden
* Kein responsives Design

1. Schlusswort

Gerade der Anfang dieser Arbeit war sehr zeitintensiv. Ich musste mir Informationen zu CI/CD beschaffen und mich damit vertraut machen. Weiter musste ich die Schritte planen, User Stories erfassen, Mockup und Dokumentation erstellen. Danach konnte ich mit der eigentlicher Arbeit beginnen.

Ich habe mich bewusst gegen ein Framework entschieden aus folgenden Gründen:

* Ich habe wenig Erfahrung in der Webentwicklung
* Es gibt eine grosse Auswahl an Frameworks
* Ein Framework muss eingerichtet werden was mit Arbeit verbunden ist
* Ich habe fast keine Erfahrung mit Frameworks
* Javascript ist neu für mich und ich möchte mich damit beschäftigen

Ich konnte alle geplanten Punkte umsetzten. Daher bin ich zufrieden mit meinem Projekt.

Mit dem Design der Webseite bin ich nicht zufrieden. Es sieht nicht nach einer moderen Webseite aus. Mir ist aber der praktische Aspekt wichtiger. Das Design kann nachträglich immer noch angepasst werden.

Ich werde mir überlegen wie ich die Zeitplanung und Arbeitsjournal besser darstellen kann.

Dies möchte ich an der IPA übersichtlicher gestalten.

Ich konnte bei diesem Projekt vieles lernen. Gerade bei Javascript habe ich viel gelernt.

In der Zukunft möchte ich mich mehr mit Frameworks beschäftigen und da mein Wissen erweitern.

1. Anhang
   1. Continuous Deployment Einrichten
      1. Github

1) Projekt Repository erstellen -> Code pushen

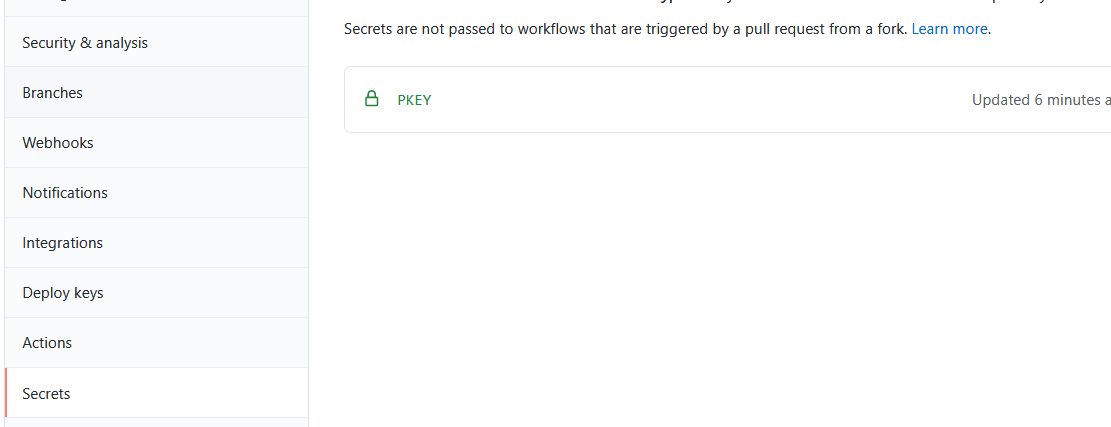
2) Deployment Repository erstellen  
 **Achtung beim Repository: "<username>.github.io"**

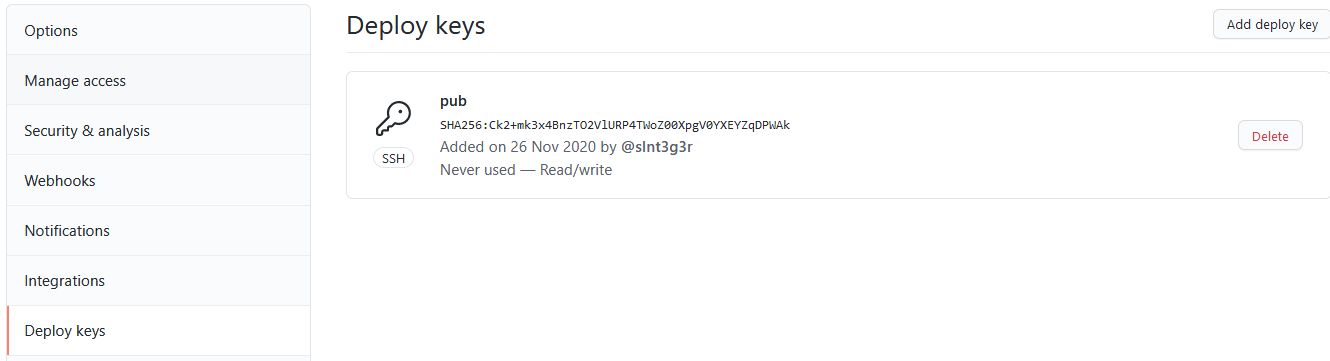
muss verwendet werden sonst funktioniert es nicht!

z.b sint3g3r.github.io

3) mit cmd "ssh-keygen -t rsa" neuen public/private key erstellen

4) Im Projekt Repository unter Settings->Secrets den Private key eintragen.



5) Im Deployment Repository unter Settings->Deploy keys den Public key eintragen. 

* + 1. Deploy.yml

name: CI only

on:

  push:

    branches:

      - main

jobs:

  deploy-test-pages:

    runs-on: ubuntu-latest

    steps:

    - uses: actions/checkout@v2

    - name: Deploy

      uses: peaceiris/actions-gh-pages@v3

      with:

        deploy\_key: ${{ secrets.PKEY }}

        external\_repository: sInt3g3r/sInt3g3r.github.io

        publish\_branch: main

        publish\_dir: ./frontend/src

<https://github.com/actions/checkout>

<https://github.com/peaceiris/actions-gh-pages>