

# Projektdokumentation

---

Dominik Kreiselmaier

Datum	Version	Änderung	Autor
16.12.2022	0.0.1	Erste Version	Kreiselmaier
23.12.2022	0.0.2	Zweite Version	Kreiselmaier
13.01.2022	0.0.3	Dritte Version	Kreiselmaier
20.01.2022	0.0.4	Vierte Version	Kreiselmaier
27.01.2022	1.0.0	Finale Version	Kreiselmaier

## 1. Informieren

---

### 1.1 Ihr Projekt

---

Projektvorschlag vom Lehrer:

**Noten:** Erstellen Sie eine API in dem Sie Ihre Noten speichern und das Ihnen die Durchschnitte pro Fach zurückgibt.¶

(Mit Backend und Frontend)

Ich habe eine SPA erstellt wo man seine Noten eintragen kann und es diese dann als Durchschnitt zurückgibt. Ich muss für das Projekt ein Backend erstellen, welches mit den CRUD-Anfragen klarkommt und die Noten in einem Array speichert. Für das Frontend musste ich JavaScript, HTML und CSS. Man sollte die Noten selber eingeben können.

### 1.2 Quellen

---

[https://www.youtube.com/watch?v=Znh8EIl\\_ICk](https://www.youtube.com/watch?v=Znh8EIl_ICk) --> Coole Pizza

<https://chat.openai.com/chat> --> wurde zum Debuggen verwendet, also ich habe den Fehler eingegeben und eine Antwort und so konnte ich besser nach Lösungen suchen.

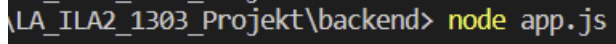
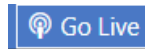


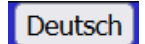
### 1.3 Anforderungen

Nummer	Muss / Kann?	Funktional? Qualität? Rand?	Beschreibung
1.	Muss	Funktional	Es gibt einen Header der die Fächer und einen Knopf, um die Noten hinzuzufügen beinhalten.
2.	Muss	Funktional	Man sieht seine Noten. (schon gespeicherte)
3.	Muss	Funktional	Man kann den Durchschnitt unter den Noten sehen.
4.	Muss	Funktional	Man kann Noten eingeben, wo man das Fach und Note eingeben muss, um eine Note hinzuzufügen.
5.	Muss	Qualität	Die Eingaben werden überprüft, ob sie die Anforderungen erfüllen (Überprüfung mit JavaScript und nicht mit HTML)
6.	Muss	Funktional	Die eingegebenen Noten werden durch das Backend in einer JS-Datei in einem Array gespeichert.
7.	Kann	Qualität	Es gibt ein Routingsystem, wo jede Seite einen eigenen hash (URL, also Deutsch, Mathematik, Note hinzufügen und Error) hat.
8.	Kann	Qualität	Wenn die URL falsch ist, bekommt man eine Fehlermeldung
9.	Muss	Funktional	Man kann Noten Löschen, indem man auf den Löschen Knopf von der richtigen Note drückt.


### 1.4 Diagramme

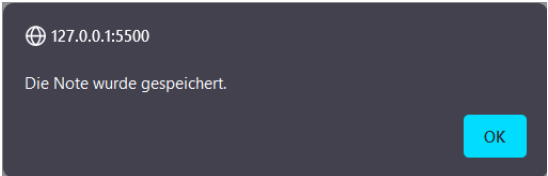
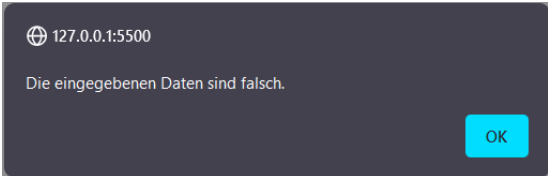
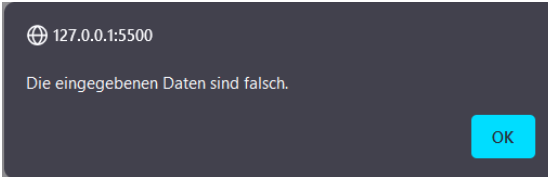
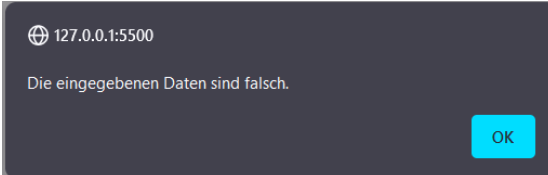
Es gibt keine Diagramme zu diesem Projekt.

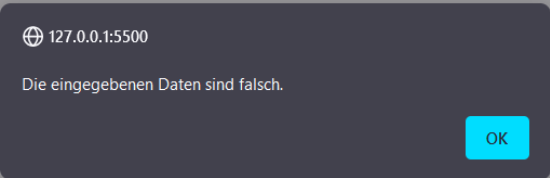
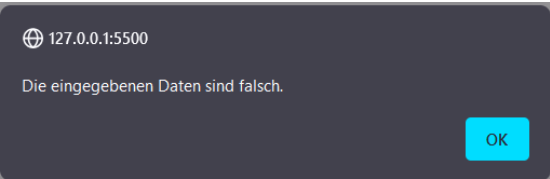
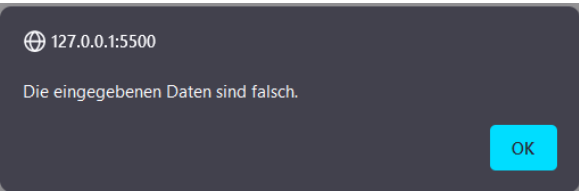
### 1.5 Testfälle

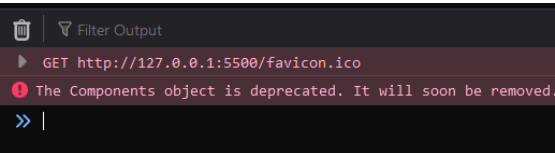
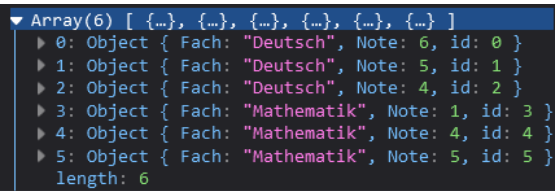
Nummer	Vorbereitung	Eingabe	Erwartete Ausgabe
1	Node installiert Live Server Extension in Visual Studio Code installiert (v5.7.9)	1. Im Terminal im Backendordner "node app.js" eingeben.  3. In Visual Studio Code rechts unten in der blauen Linie auf "Go Live" klicken. 	2. Im Terminal sollte man folgendes sehen:  4. Im Header der Webseite sollte man folgendes sehen. 
2.1	Nummer 1	1. Auf "Deutsch" im Header klicken. 	2. Man sieht im body der Webseite folgendes:

			<b>Note:</b> 6 Löschen <b>Note:</b> 5 Löschen <b>Note:</b> 4 Löschen
2.2	Nummer 1	1. Auf "Mathematik" im Header klicken. Mathematik	2. Man sieht im body der Webseite folgendes: <b>Note:</b> 1 Löschen <b>Note:</b> 4 Löschen <b>Note:</b> 5 Löschen
3.1	Nummer 1	1. Auf "Deutsch" im Header klicken. Deutsch	2. Man sieht im body der Webseite folgendes:

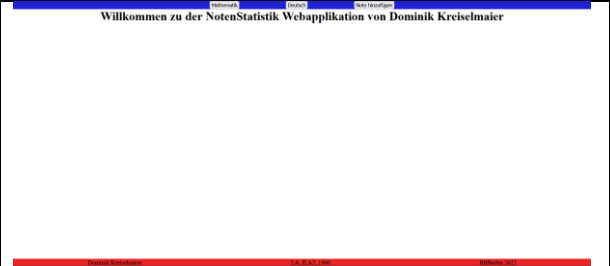
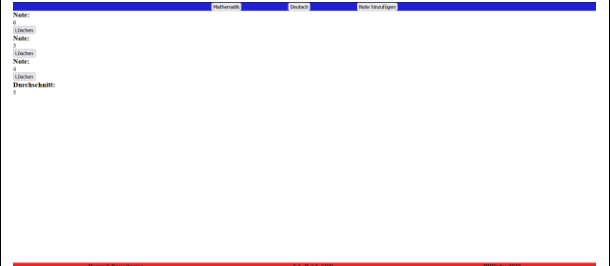
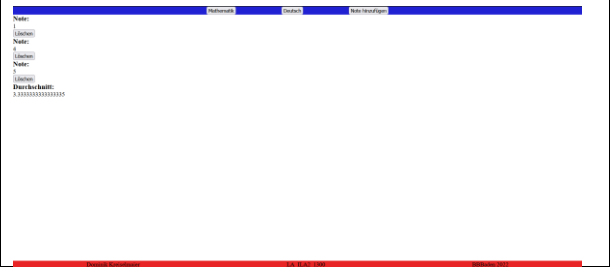
			<b>Note:</b> 6 Löschen <b>Note:</b> 5 Löschen <b>Note:</b> 4 Löschen <b>Durchschnitt:</b> 5
3.2	Nummer 1	1. Auf "Mathematik" im Header klicken. Mathematik	2. Man sieht im body der Webseite folgendes: <b>Note:</b> 1 Löschen <b>Note:</b> 4 Löschen <b>Note:</b> 5 Löschen <b>Durchschnitt:</b> 3.3333333333333335
4	Nummer 1	1. Auf "Note hinzufügen" im Header klicken. Note hinzufügen 3. In dem Eingabefeld "Fach" "Deutsch" eingeben und bei "Note" "5".	2. Man sieht im body der Webseite folgendes: Fach: <input type="text"/> Note: <input type="text"/>  Submit

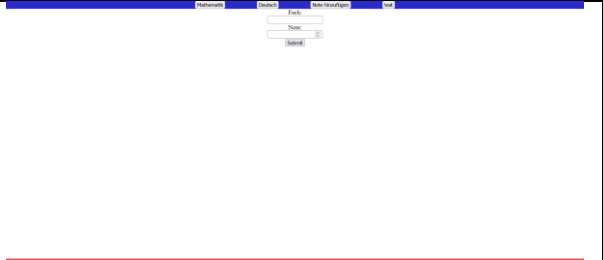
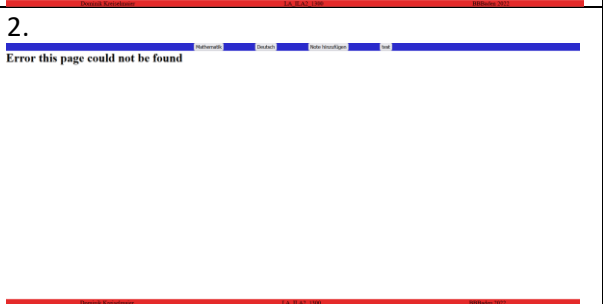
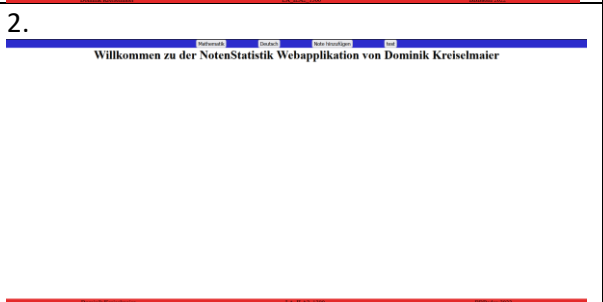
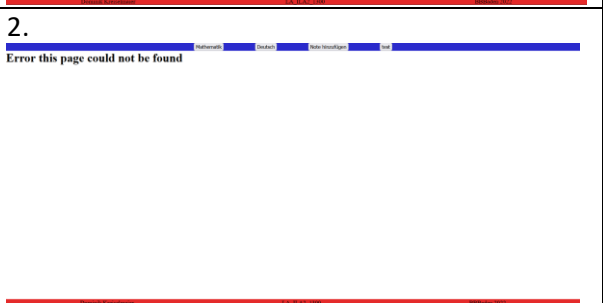
		<p>Fach:</p> <input type="text" value="Deutsch"/> <p>Note:</p> <input type="text" value="5"/> <input type="button" value="Submit"/> <p>4. Auf "Submit" klicken.</p>	<p>5.</p> 
5.1	Bei Nummer 4 und nur Punkt 1 gemacht	<p>1. In dem Eingabefeld "Fach" " asfsdf" eingeben und bei "Note" "1".</p> <p>Fach:</p> <input type="text" value="asfsdf"/> <p>Note:</p> <input type="text" value="1"/> <input type="button" value="Submit"/> <p>2. Auf "Submit" klicken.</p>	<p>3.</p> 
5.2	Bei Nummer 4 und nur Punkt 1 gemacht	<p>1. In dem Eingabefeld "Fach" " " eingeben und bei "Note" "1".</p> <p>Fach:</p> <input type="text" value=" "/> <p>Note:</p> <input type="text" value="1"/> <input type="button" value="Submit"/> <p>2. Auf "Submit" klicken.</p>	<p>3.</p> 
5.3	Bei Nummer 4 und nur Punkt 1 gemacht	<p>1. In dem Eingabefeld "Fach" "" eingeben und bei "Note" "".</p> <p>Fach:</p> <input type="text" value=""/> <p>Note:</p> <input type="text" value=""/> <input type="button" value="Submit"/>	<p>3.</p> 

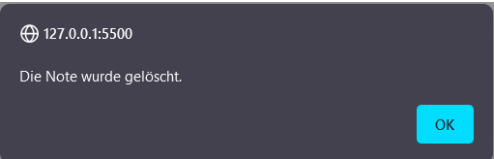
		2. Auf "Submit" klicken.	
5.4	Bei Nummer 4 und nur Punkt 1 gemacht	<p>1. In dem Eingabefeld "Fach" "Deutsch" eingeben und bei "Note" "0".</p> <p>Fach:</p> <input type="text" value="Deutsch"/> <p>Note:</p> <input type="text" value="0"/> <input type="button" value="Submit"/> <p>2. Auf "Submit" klicken.</p>	<p>3.</p> 
5.5	Bei Nummer 4 und nur Punkt 1 gemacht	<p>1. In dem Eingabefeld "Fach" "Deutsch" eingeben und bei "Note" "7".</p> <p>Fach:</p> <input type="text" value="Deutsch"/> <p>Note:</p> <input type="text" value="7"/> <input type="button" value="Submit"/> <p>2. Auf "Submit" klicken.</p>	<p>3.</p> 
5.6	Bei Nummer 4 und nur Punkt 1 gemacht	<p>1. In dem Eingabefeld "Fach" "Deutsch" eingeben und bei "Note" "".</p> <p>Fach:</p> <input type="text" value="Deutsch"/> <p>Note:</p> <input type="text" value=""/> <input type="button" value="Submit"/> <p>2. Auf "Submit" klicken.</p>	<p>3.</p> 
6.1	Der Backendordner ist offen	<p>1. Den Ordner "API_Handler" öffnen</p> <p>▼ API_Handler</p> <p>2. Die Datei "data.js" öffnen.</p> <p>JS data.js</p>	<p>3.</p>

			<pre> {   "Fach": "Deutsch",   "Note": 6, }, {   "Fach": "Deutsch",   "Note": 5, }, {   "Fach": "Deutsch",   "Note": 4, }, {   "Fach": "Mathematik",   "Note": 1, }, {   "Fach": "Mathematik",   "Note": 4, }, {   "Fach": "Mathematik",   "Note": 5, }, </pre>
6.2	Nummer 1	<p>1. "Ctrl" + "Shift" + "I" drücken</p> <p>3. In dem Terminal folgendes eingeben:</p> <pre> fetch('http://localhost:3000/api/getall')   .then(response =&gt; response.json())   .then(data =&gt; console.log(data))   .catch(error =&gt; console.error(error)); </pre> <pre> &gt;&gt; fetch('http://localhost:3000/api/getall')   .then(response =&gt; response.json())   .then(data =&gt; console.log(data))   .catch(error =&gt; console.error(error)); </pre> <p>4. "Enter" drücken</p> <p>6. Auf das Dreieck neben dem Wort "Array" drücken.</p>	<p>2.</p>  <p>5.</p> <pre> &gt;&gt; Array(6) [ {...}, {...}, {...}, {...}, {...}, {...} ] </pre> <p>7.</p> 
7.1	Nummer 1	<p>1. In das URL-Feld vom Browser folgendes eingeben:</p> <p><a href="http://127.0.0.1:5500/frontend/">http://127.0.0.1:5500/frontend/</a></p>	<p>2.</p>



			 <p>3. In der URL sollte folgendes stehen: <code>http://127.0.0.1:5500/frontend/#Home</code></p>
7.2	Nummer 1	1. In das URL-Feld vom Browser folgendes eingeben: <code>http://127.0.0.1:5500/frontend/#German</code>	<p>2.</p> 
7.3	Nummer 1	1. In das URL-Feld vom Browser folgendes eingeben: <code>http://127.0.0.1:5500/frontend/#Math</code>	<p>2.</p> 
7.4	Nummer 1	1. In das URL-Feld vom Browser folgendes eingeben: <code>http://127.0.0.1:5500/frontend/#GradesInput</code>	<p>2.</p>

			
7.5	Nummer 1	1. In das URL-Feld vom Browser folgendes eingeben: <code>http://127.0.0.1:5500/frontend/#Error</code>	2. 
7.6	Nummer 1	1. In das URL-Feld vom Browser folgendes eingeben: <code>http://127.0.0.1:5500/frontend/#Home</code>	2. 
8	Nummer 1	1. In das URL-Feld vom Browser folgendes eingeben: <code>http://127.0.0.1:5500/frontend/#dsfljdgkjld</code>	2. 

9	Nummer 4	<p>1. Im Header auf "Deutsch" klicken.</p> <p>Deutsch</p> <p>3. Auf den "Löschen" Knopf drücken, von der untersten Note.</p> <p><b>Note:</b></p> <p>6</p> <p>Löschen</p> <p><b>Note:</b></p> <p>5</p> <p>Löschen</p> <p><b>Note:</b></p> <p>4</p> <p>Löschen</p> <p><b>Note:</b></p> <p>5</p> <p>Löschen</p>	<p>2.</p> <p><b>Note:</b></p> <p>6</p> <p>Löschen</p> <p><b>Note:</b></p> <p>5</p> <p>Löschen</p> <p><b>Note:</b></p> <p>4</p> <p>Löschen</p> <p><b>Note:</b></p> <p>5</p> <p>Löschen</p> <p><b>Durchschnitt:</b></p> <p>5</p> <p>4.</p>  <p>5.</p> <p><b>Note:</b></p> <p>6</p> <p>Löschen</p> <p><b>Note:</b></p> <p>5</p> <p>Löschen</p> <p><b>Note:</b></p> <p>4</p> <p>Löschen</p> <p><b>Durchschnitt:</b></p> <p>5</p>
---	----------	--	---

## 2. Planen

Ein Arbeitspaket ist 45min lang und es werden alle von Dominik Kreiselmaier gemacht, da es ein Einzelprojekt ist.

Nummer	Frist	Beschreibung	Zeit (geplant)
1.	16.12.2022	Projekt aussuchen	2
2.	16.12.2022	Über das Projekt Informieren und schauen, was für Technologien benutzen kann.	2
3.	23.12.2022	Für Technologien entscheiden.	½
4.	23.12.2022	Anforderungen, Testfälle und Zeitplan erstellen.	2 ½
5.	13.01.2023	HTML mit CSS erstellen	1
6.	20.01.2023	Backend mit JavaScript erstellen (Daten als Array speichern/ CRUD Anfragen Handler)	6
7.	27.01.2023	Frontend mit JavaScript erstellen (Die Noten als Durchschnitt zurückgeben/ Request an das Backend machen (getall, addone, delete)/ Eingaben erlauben und bei falschen einen Error zurückgeben)	6
8.	27.01.2023	Testprotokoll und Testfazit erstellen	1
9.	27.01.2023	Auswertung erstellen	1
10.	27.01.2023	Portfolio erstellen (Video erstellen/ Links zu Dokumentation und Projekt einbinde/, Ziel, Reflexion und Validierung erstellen/ Projekterklärung schreiben)	3
TOTAL:			5 * 5 * 1 = 25 Arbeitspakete


Ich habe weniger für das Entscheiden eingeplant, da fast alles klar war was ich möchte an Technologien, als ich mich informiert habe und ich mich nicht mit anderen Personen einigen musste.

## 3. Entscheiden


Ich habe mich für die DIE Visual Studio Code entschieden, mit den Add-ons Live Server (v5.7.9) und REST Client (v.0.25.1).

Ich habe mich auch für die Technologie NodeJS entschieden.

```
C:\Users\doeme>node -v
v18.13.0
```



**Live Server** v5.7.9  
Ritwick Dey | 30,227,321 | ★★★★★ (424)  
Launch a development local Server with live reload feature for static & dynamic pages  
Uninstall ⚙️  
⚠️ This extension has been disabled because the current workspace is not trusted.



**REST Client** v0.25.1  
Huachao Mao | 3,053,861 | ★★★★★ (330)  
REST Client for Visual Studio Code  
Uninstall ⚙️  
⚠️ This extension has been disabled because the current workspace is not trusted.



#### 4. Realisieren

Nummer	Datum	Beschreibung	Zeit (geplant)	Zeit (effektiv)
1.	16.12.2022	Projekt aussuchen	2	1
2.	16.12.2022	Über das Projekt Informieren und schauen, was für Technologien benutzen kann.	2	1
3.	23.12.2022	Für Technologien entscheiden.	½	1
4.	23.12.2022	Anforderungen, Testfälle und Zeitplan erstellen.	2 ½	2
5.	13.01.2023	HTML mit CSS erstellen	1	3
6.	20.01.2023	Backend mit JavaScript erstellen (Daten als Array speichern/ CRUD Anfragen Handler)	6	8
7.	27.01.2023	Frontend mit JavaScript erstellen (Die Noten als Durchschnitt zurückgeben/ Request an das Backend machen (getall, addone, delete)/ Eingaben erlauben und bei falschen einen Error zurückgeben)	6	6
8.	27.01.2023	Testprotokoll und Testfazit erstellen	1	1
9.	27.01.2023	Auswertung erstellen	1	1
10.	27.01.2023	Portfolio erstellen (Video erstellen/ Links zu Dokumentation und Projekt einbinde/, Ziel, Reflexion und Validierung erstellen/ Projekterklärung schreiben)	3	4

Ich habe beim Portfolio erstellen ein wenig mehr gebraucht, da ich es zu Hause noch ein bisschen überarbeiten musste. Ich habe, aber auch so ein paar wenige Sachen zu Hause gemacht, wegen zeitlichen Gründen.

#### 5. Kontrollieren

##### 5.1 Testprotokoll


###### Testumgebung:

Die Tests wurden von Dominik Kreiselmaier per Hand durchgeführt mit FireFox, Visual Studio Code und der Extension Live Server und ich hatte NodeJS installiert. Die Tests wurden auf dem Betriebssystem Windows 11 Home ausgeführt.

Ein paar Tests wurden auch mit der Selenium IDE durchgeführt (Browser Extension), diese wurde aber auch nochmal per Hand geprüft.

Windows 11, version 22H2  
Successfully installed on 06/11/2022

Name

 Microsoft Visual Studio Code (User)

Publisher

Microsoft Corporation

Installed On


11/01/2023


Size

353 MB

Version

1.74.3


Name	Publisher	Installed On	Size	Version
 Mozilla Firefox (x64 de)	Mozilla	20/01/2023	338 MB	109.0



**Live Server** v5.7.9
 

Ritwick Dey
 | 30,227,321
 | ★★★★★ (424)

Launch a development local Server with live reload feature for static & dynamic pages

Uninstall
 

⚠ This extension has been disabled because the current workspace is not trusted.

Nummer	Datum	Resultat	Durchgeführt
1	27.01.2023	OK	Kreiselmaier
2.1	27.01.2023	OK	Kreiselmaier
2.2	27.01.2023	OK	Kreiselmaier
3.1	27.01.2023	OK	Kreiselmaier
3.2	27.01.2023	OK	Kreiselmaier
4	27.01.2023	OK	Kreiselmaier
5.1	27.01.2023	OK	Kreiselmaier
5.2	27.01.2023	OK	Kreiselmaier
5.3	27.01.2023	OK	Kreiselmaier
5.4	27.01.2023	OK	Kreiselmaier
5.5	27.01.2023	OK	Kreiselmaier
5.6	27.01.2023	OK	Kreiselmaier
6.1	27.01.2023	OK	Kreiselmaier
6.2	27.01.2023	OK	Kreiselmaier
7.1	27.01.2023	OK	Kreiselmaier
7.2	27.01.2023	OK	Kreiselmaier
7.3	27.01.2023	OK	Kreiselmaier
7.4	27.01.2023	OK	Kreiselmaier
7.5	27.01.2023	OK	Kreiselmaier
7.6	27.01.2023	OK	Kreiselmaier
8	27.01.2023	OK	Kreiselmaier
9	27.01.2023	OK	Kreiselmaier

## **5.2 Testfazit**

---

Es funktioniert alles und die Applikation läuft so wie sie soll.

## **6. Auswerten**

---

Das Projekt ist nicht so gut gelaufen, da ich mehr als geplant für das Backend gebraucht habe, da ich nicht so verstanden habe, was ich machen musste. Dafür ist mir das Erstellen des Portfolios besser gelaufen, da ich mich bei Kollegen und Lehrer erkundigt habe was ich wie verbessern kann und habe es auch umgesetzt.