

# Dokumentation

## Modul 294

Fach: Modul 294

Dozent: Lukas Müller

Autoren: Nelo Nisslé, Muhammet Topal

Datum: 23.11.24

25. November 2024

## Inhaltsverzeichnis

Einführung .....	3
Informieren .....	4
Ausgangslage: .....	4
Ressourcen: .....	4
Ziel Auswertung .....	5
Planen .....	7
Projektstruktur: .....	7
Mockup und Skizzen .....	8
Entscheiden .....	10
Warum Bootstrap? .....	10
Farbliche Gestaltung: .....	10
Technologieentscheidung .....	10
Realisieren .....	11
Code Der Webseite .....	11
Hilfs Mittel bei der Umsetzung .....	11
Verbindung mit JavaScript: .....	12
Kontrollieren .....	15
Test mit Postman: .....	15
Test mit REST Extension: .....	16
Vollständigkeit liste: .....	17
Auswerten .....	18
Lessons Learned: .....	18
Verbesserungen in der Zukunft .....	18
Programm Richtig starten und Nutzen .....	19
Fazit: .....	19
Referenzen & Quellen .....	20

## Einführung

Hier werden wir sie kurz einführen in unsere Projekt Arbeit.

In unserem Projekt geht es darum eine Interactive Webseite zu erstellen die mit einem backend kommunizieren muss um Daten auszutauschen die wir dann mit Get überprüfen konnten. Die Webseite muss zu dem mit einem Framework erstellt werden wir haben «Bootstrap» als Framework genommen da es uns die perfekten Funktionen Bietet die wir benötigen um unsere Webseite am besten wie möglich zu designen und Fertig zu stellen. Um das Ganze so schnell wie möglich in die Realität umzusetzen haben wir uns nach IPERKA Orientiert . Ausserdem mussten wir uns an den Vorgegeben Zielen Orientieren und diese Nach Dringlichkeit auswerten. Ausserdem hatten wir uns auch eine grobe Skizzen Planung erstellt für unsere Webseite.

## Informieren

In dem Ersten Schritt «Informieren» haben wir uns vor allem auf die Information Beschaffung und auf die Zielsetzung Fokussiert. Damit wir in den späteren teilen eine festes ziel haben auf das wir uns voll konzentrieren können.

### Ausgangslage:

Eine Vollständige Interaktive Webseite die mit einem Backend Kommunizieren kann. Die Webseite muss mit einem Framework erstellt werden wir haben Dafür Bootstrap genommen. Die Webseite musste eine Anmelde Formular haben die Daten an den Serverschickt und dies müssen wir überprüfen in dem wir entweder eine VSC Extension benutzen oder Postman mit einer «Get» Anfrage.

### Ressourcen:

Wir haben Verschiedenste Tools und andere Anwendungen verwendet um unser Projekt umzusetzen.

- Visual Studio Code: Für die Entwicklung der Webseite und des Frontends.
- Bootstrap: Zur Umsetzung eines responsiven Webdesigns und für vorgefertigte UI-Komponenten.
- Postman: Für das Testen der REST-API und die Überprüfung der Datenübertragung.
- GitHub: Zur Versionskontrolle und Zusammenarbeit im Projekt.

Wir haben uns Hier Auch noch Gedanken gemacht bezüglich Verwendung von Hilfs mitteln und sind zum Entschluss gekommen das wir möglichst ohne Hilfsmittel arbeiten wollen aber wenn es Dazu kommen sollte das wir welche benötigen brauchen wir diese dann auch.

## Ziel Auswertung

Hier haben wir erstmals die ziele und auch gleich in einer weiteren Tabelle die ziel Auswertung und Priorität.

Hier die Anforderungen an unsere Webseite:

### **Nr. Beschreibung**

A1	Landing Page mit den erforderlichen Elementen, Gestalterische Grundsätze berücksichtigt (Barrierefreies Design)
A2	Responsives Webdesign, Korrekte Wahl des Designs
A3	Kontaktseite (Adresse, Telefon, Anfahrt usw.)
A4	Formular für die Onlineanmeldung eines Serviceauftrags inkl. Validierung der Eingabedaten.
A5	An den Server müssen nachfolgende Informationen eines Serviceauftrags gesendet werden:
	- Kundenname
	- E-Mail
	- Telefon
	- Priorität
	- Dienstleistung
	- Erfassungsdatum
	- Abholdatum
A6	Auswahl der Dienstleistungen bei der Serviceanmeldung
A7	Intuitive Bedienung, benutzerfreundliche und verständliche Anzeige der ausgewählten Dienstleistung
A8	Bestätigung der Onlineanmeldung
A9	Testprotokoll zur Eingabvalidierung
A10	Test API-Projekt in Postman oder VS REST-Client angelegt.
A11	Ganzes Projektmanagement muss nach IPERKA dokumentiert sein

Das sind Die Vorgegeben Anforderungen die an unsere Webseite gerichtet sind. Jetzt werde ich diese Ziele Noch in einer weiteren Tabelle nach Ihrer Priorität auswerten so das wir ein möglichst effizientes Vorgehens Modell haben.

<b>NR. Auswertung nach Priorität.</b>	
A1	Hoch
A2	Hoch
A3	Mittel
A4	Hoch
A5	Hoch
A6	Mittel
A7	Hoch
A8	Mittel
A9	Hoch
A10	Hoch
A11	Mittel

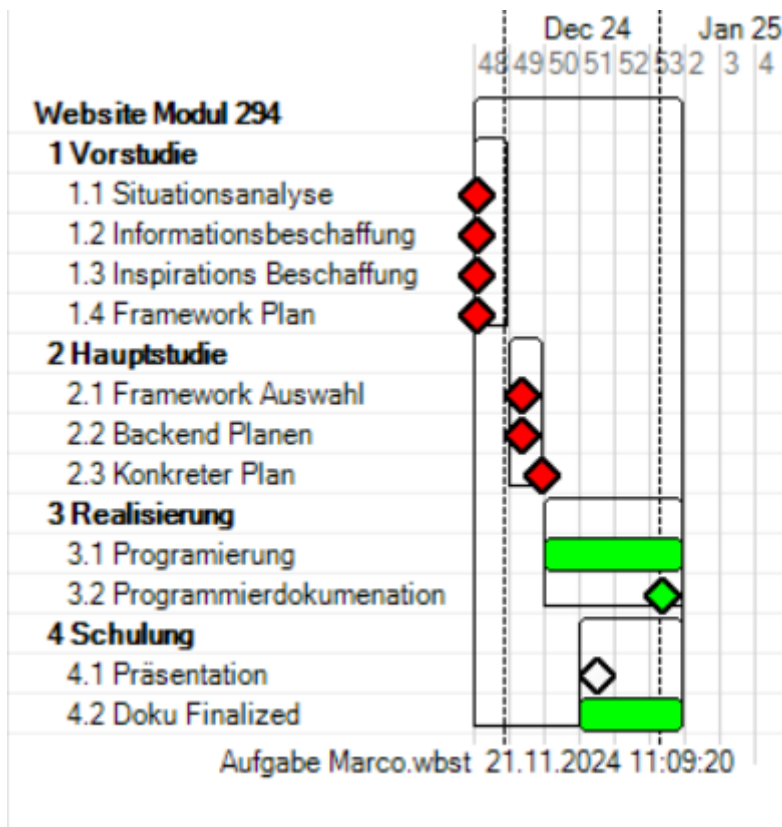
## Planen

In der Planung haben wir vor allem punkte gemacht wie die Zeitplanung, Aufgaben Aufteilung und erste Skizzen unserer Webseite. Das ist für uns essentiell da es uns eine Grundlage gibt für die effiziente Erarbeitung unseres Projekts.

### Projektstruktur:

Hier haben wir einen Zeitplan Entworfen und unsere Aufgaben Einteilung erstellt das wir so schnell wie möglich Arbeiten und beide Ihre stärken aufbringen und anwenden können.

Hier ist ein Bild unsers Zeitplanes den wir erstellt haben und nach dem wir vorgegangen sind:



Das ist unser Grober Zeitplan den wir Entwickelt haben. Wie man unschwer erkennen kann ist er sehr grob da wir unsere Arbeit eigentlich beide auf verschiedenste zeiten eingeteilt haben und die aufgaben auch sehr Individuell erarbeitet haben.

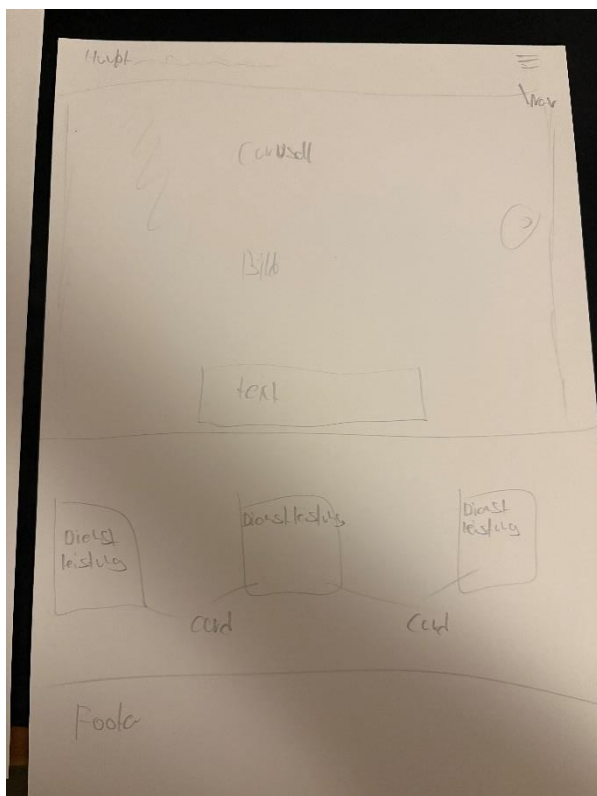
Und Auch haben wir uns hier noch eingeteilt wer welche aufgaben übernimmt so das wir am einfachsten arbeiten konnten auch wen wir gerade nicht mit einander sprechen konnten. So waren wir nicht nur extrem effizient sondern auch extrem Flexibel. Diese Aufgaben Einteilung sah wie Folgend aus:

Aufgabe	Aufteilung
Frontend	Mali/ Nelo
Kommunikation mit dem Server	Nelo
Dokumentation	Nelo/ Mali
Präsentation	Nelo/ Mali
Testing	Nelo/ Mali
Optimierung des Forntends	Mali

## Mockup und Skizzen

Hier haben wir uns dann erstmalig um das Design richtige Gedanken gemacht und uns dann auch mit Hilfe von Skizzen Gerade ein festes Design überlegt das wir umsetzten wollen. Deshalb haben wir uns da auch zusammen gesetzt und viel überlegt wie wir unsere Webseite gestalten wollen. Insgesamt haben wir uns dann zwei Hauptskizzen gemacht die erste für die «Startseite» und die Zweite für die «Anmeldungsseite» so sind wir auch automatisch auf das design der «Contact» und «About us» Seite Gekommen.

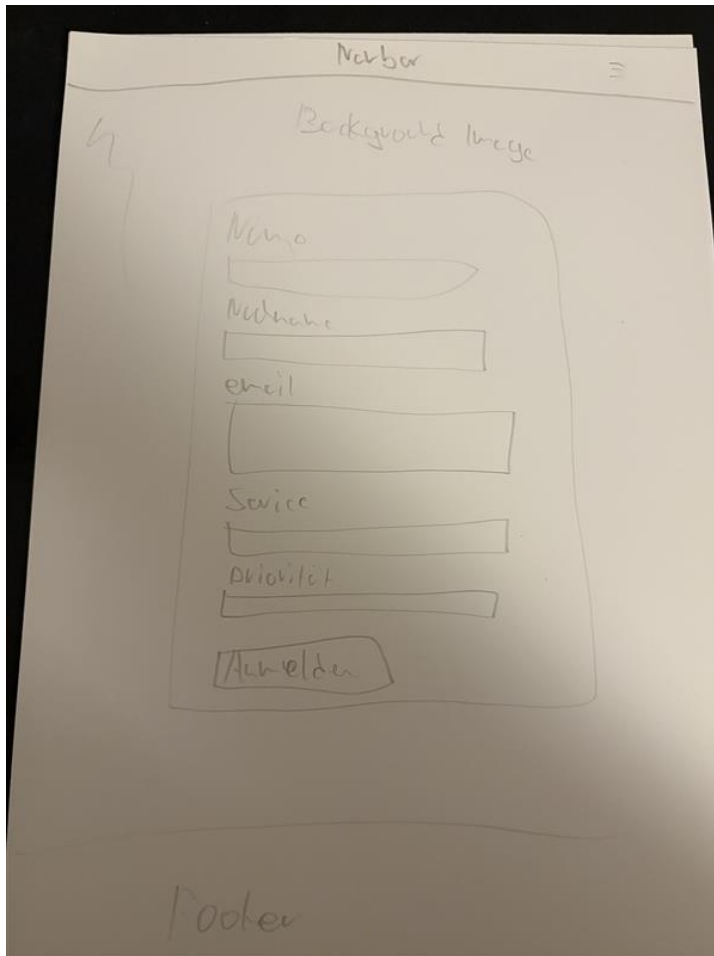
Hier ist die Skizze der Hauptseite:



Ja wir wissen das man die skizze nicht gut sieht aber wir wollten sie verwenden da es die Original Skizzen sind die wir verwendet haben.



Hier ist die zweite skizze:



Diese Skizze ist etwas besser zu erkennen aber natürlich nicht ganz so gut wir haben uns aber aus dem selben Grund entschieden sie zu verwenden wie bei der Letzten Skizze.

Der Grund wieso wir keine Online Skizze gemacht haben war weil wir uns beide der Meinung waren das es auf Blatt authentischer ist und wir uns beim erstellen der skizze deutlich mehr gesagt haben als gezeichnet.

## Entscheiden

Hier sind wir auf die Punkte eingegangen Warum Bootstrap, Farbliche Gestaltung, Technologieentscheidung.

### Warum Bootstrap?

Wir haben uns zu dem ganz viele Gedanken gemacht und sind dann zum Beschluss gekommen das uns Bootstrap genau die Elemente liefert die wir auf unserer Seite verwenden wollen. Das waren Elemente wie Cards, Carousel und viele mehr, auch lieferte Bootstrap perfekte «Examples» die wir als Beispiel sehen konnten und auch ähnlich anpassen und übernehmen konnten. Es gab aber natürlich auch Punkte bei denen wir nicht ganz so happy waren von Bootstrap vor allem wegen dem CSS das war zum Teil echt extrem nervig da es nicht ganz so einfach und Effizient ist. Das waren die Hauptgründe wieso wir Bootstrap verwendet haben.

### Farbliche Gestaltung:

Das war für uns schon ganz am Anfang klar das wir auf eine winterliche Farbpalette gesetzt haben und diese auch verwendet haben. Diese beinhaltete hauptsächlich weiße und blaue Farbelemente. Als Schriftfarbe haben wir uns ganz einfach auf schwarz entschieden da es überall gut lesbar war und auch gepasst hat.

### Technologieentscheidung

Hier haben wir festgelegt welche Tools und welche Applikationen wir verwendet haben.

Das waren die einfachsten und schnellsten Entscheidungen da wir diese schon beherrscht haben. Diese technologischen Entscheidungen waren folgende:

- Arbeits Umgebung: Visual Studio Code da wir extrem viel schon damit gemacht haben und uns dementsprechend sehr gut damit auskannten.
- Testing Programm: Postman weil wir damit schon gearbeitet haben und es mit Abstand die umfassendste Umgebung war um unsere Überprüfungen zu machen.
- Bootstrap: Für uns am besten passendes Framework.
- JavaScript: mussten wir für den Backend verwenden und haben wir in dem Modul angeschaut.

## Realisieren

Hier werden wir auf den Tiel der Realisation eingehen aber nicht auf alles ganz so genau da das meiste wie HTML eigentlich selbsterklärend sind. Darum gehen wir aber genauer auf den teil der Server Verbindung ein und erklären den so gut wie möglich.

### Code Der Webseite

Für den Code Der Webseite den wir mit Bootstrap gemacht haben werden wir nicht näher erklären da dieser selbst erklärend sein sollte. Auch Da dieser Code in den Files selber noch erklärt ist mit den Elementen. Dennoch werden wir kurz über die Arbeit in Bootstrap reden gerade was die verschiedenen Elemente angeht

Wir haben Verschiedenste Elemente genutzt aber die wichtigsten sind Sachen wie die Navigation zwischen den Seiten. Diese Navigation ist über ein Element «nav» geregelt klar muss man die verschiedene Seiten oder abschnitte in der navbar selber noch verlinken so das man sie auch benutzen kann.

Ein weiters wichtiges Element unsere Hauptseite ist das Carousel gewesen. Das Carousel ist nichts anderes als ein Komponente das man mit verschiedensten bildern Austaten kann. Das haben wir genommen da es grafisch sehr gut zu unsere seite gepasst hat.

### Hilfs Mittel bei der Umsetzung

Wir haben so gut wie keine Hilfsmittel genommen da es unser ziel war so viel wie möglichst selbstständig zu machen. Dennoch haben wir Seiten wie Bootstrap genommen um unsere Codes herzuholen. Auch haben wir Hilfe genommen bei Seiten die uns halfen eine gute Farbpalette zu erstellen.

Bei Problemen haben wir einfach gegoogelt und so meist eine sehr gute und schnelle Lösung bekommen. Natürlich haben wir aber auch ein zwei mall ChatGPT verwendet für eine Hilfestellung die wir sonst nicht so schnell kapiert hätten. Da es auch gelogen wäre hätten wir gesagt das wir kein ChatGPT verwendet haben ist es klar das wir es angeben.

## Verbindung mit JavaScript:

Da wir das schon in der Präsentation erklärt habe gehe ich hier genau gleich vor und werde das ganze in abschnitte unterteilen die mit Abschnitt 1 anfängt und so weitergeht. Das ganze ist aufbauend da es im Abschnitt 1 um den ersten block im Code geht und immer so weiter.

### Abschnitt 1:

```
document.getElementById("serviceForm").addEventListener("submit", function(e) { });
```

Der Code wartet auf das event, dass das Formular mit der ID serviceForm abgeschickt wird. Wenn das Formular abgeschickt wird, wird die angegebene Funktion ausgeführt.

### Abschnitt 2:

```
e.preventDefault();
```

Das Standardverhalten des Formulars (die Seite neu zu laden) wird verhindert, sodass der Code die Daten programmgesteuert verarbeiten kann.

### Abschnitt 3:

```
const name = document.getElementById("name").value;
const email = document.getElementById("email").value;
const phone = document.getElementById("phone").value;
const service = document.getElementById("service").value;
const priority = document.getElementById("priority").value;
```

Die Werte aus den inputfeldern (name, email, phone, service, priority) werden abgerufen und in die Variablen gespeichert. so Können wir die daten später ausslesen unhd posten.

### Abschnitt 4:

```
const priorityDays = {
  "Tief": 12,
  "Standard": 7,
  "Express": 5
};
```

Ein Objekt wird erstellt, das für jede Priorität (Tief, Standard, Express) eine Anzahl von Tagen enthält, die zu einem bestimmten Datum hinzugerechnet werden.

#### Abschnitt 5:

```
const today = new Date();  
today.setDate(today.getDate() + priorityDays[priority]);
```

Ein neues Datum (today) wird erstellt, das den aktuellen Tag repräsentiert. Basierend auf der gewählten Priorität wird die Anzahl der Tage (aus dem priorityDays-Objekt) zum aktuellen Datum hinzugefügt.

#### Abschnitt 6:

```
const pickupDate = today.toISOString().split('T')[0];
```

Das Datum wird in ISO-Format (z.B. "2024-11-20T12:34:56.789Z") umgewandelt und anschließend mit .split('T')[0] nur das Datum (ohne Uhrzeit) extrahiert.

#### Abschnitt 7:

```
const data = {  
  name,  
  email,  
  phone,  
  service,  
  priority,  
  created_at: new Date().toISOString().split('T')[0],  
  pickup_date: pickupDate  
};
```

Ein data-Objekt wird erstellt, das alle erfassten Formulardaten enthält. Zusätzlich wird das Erstellungsdatum (created\_at) in einem ähnlichen Format wie das Pickup-Datum gespeichert.

#### Abschnitt 8:

```
console.log(data);
```

Die erstellten Daten werden in der Konsole ausgegeben, um zu überprüfen, was gesendet wird.

#### Abschnitt 9:

```
fetch("http://localhost:5000/api/registration", {  
  method: "POST",  
  headers: { "Content-Type": "application/json" },  
  body: JSON.stringify(data)  
})
```

Eine POST-Anfrage wird an das Backend gesendet. Die URL `<http://localhost:5000/api/registration>` ist der Endpunkt, an den die Formulardaten gesendet werden. Der Content-Type-Header gibt an, dass die gesendeten Daten im JSON-Format sind. Der body enthält die Formulardaten, die in einen JSON-String umgewandelt werden.

#### Abschnitt 10:

```
.then(response => response.json())  
  .then(result => alert("Anmeldung erfolgreich!"))  
  .catch(error => console.error("Fehler:", error));
```

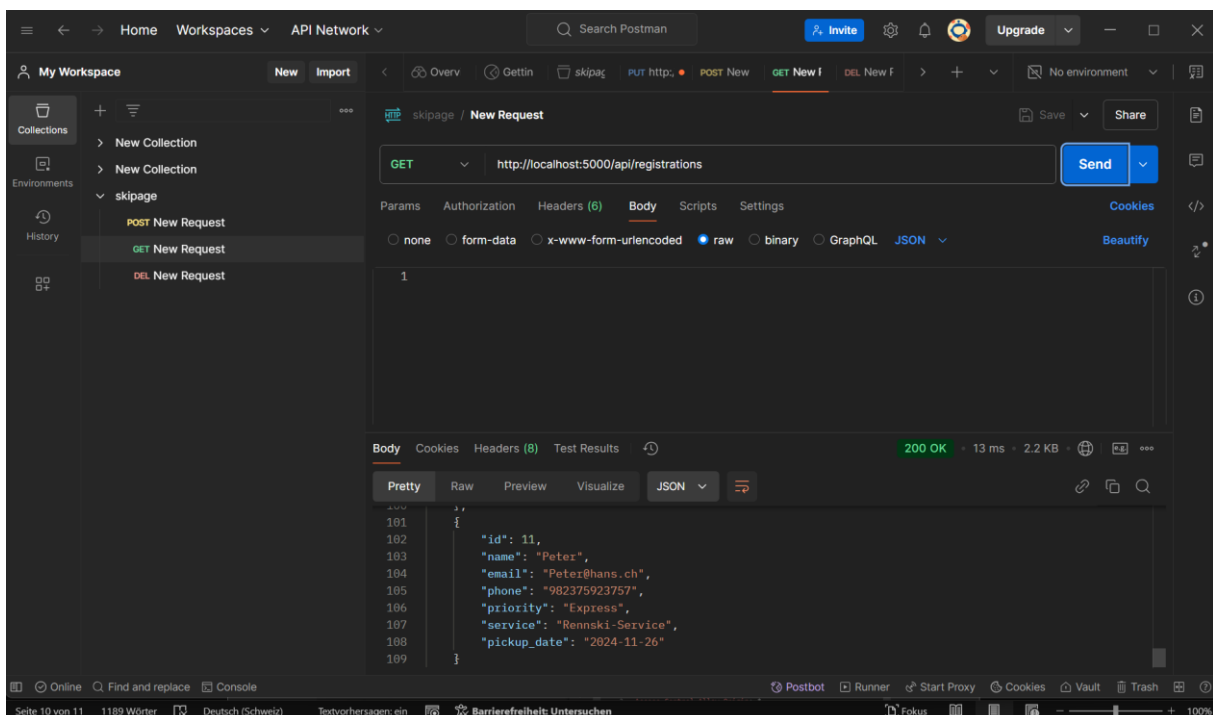
Wenn die Anfrage erfolgreich ist, wird die Antwort als JSON verarbeitet und eine Erfolgsmeldung angezeigt. Wenn ein Fehler auftritt, wird dieser in der Konsole protokolliert.

## Kontrollieren

Bei dem Kapitel Kontrollieren sind wir vor allem auf unsere Kontrollen mit Postman eingegangen da wir unsere Testes so gut wie nur über Postmans gemacht haben und mit der «REST» Extension in Visual Studio Code gemacht haben. Mehr hatten wir Bei den Tests auch gar nicht gemacht da diese Tests Alle Erfolgreich waren und wir dem entsprechen auch kein Test Diagramm erstellen mussten da es Direkt Funktioniert hat.

### Test mit Postman:

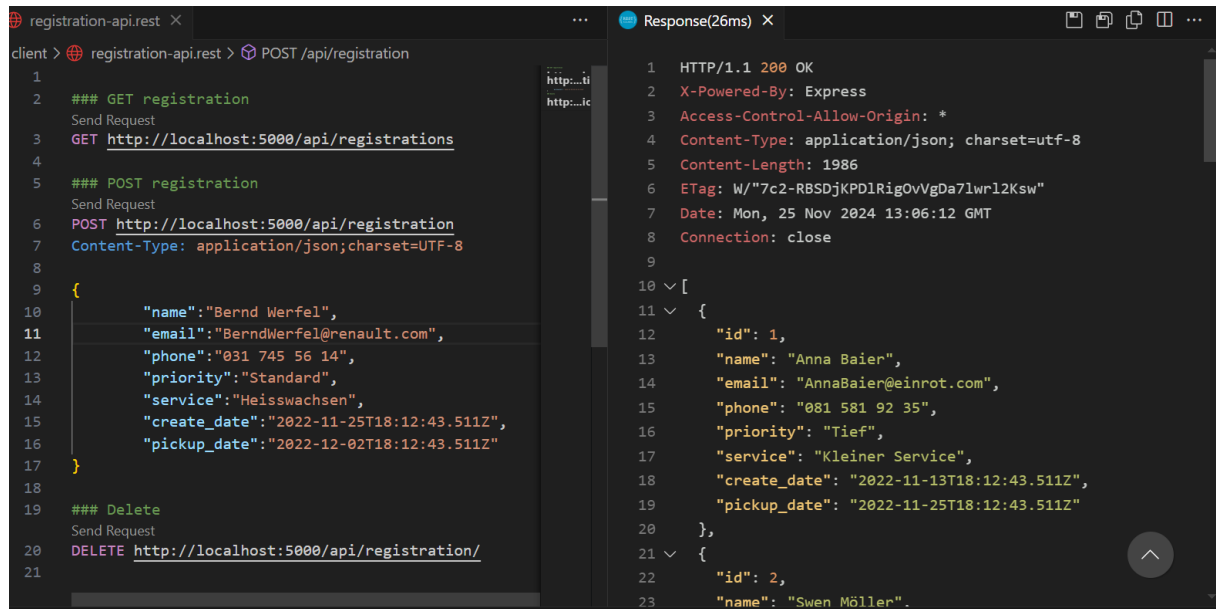
Die Tests in Postman haben wir Durchgeführt mit dem wir die URL geprüft haben wo de JSON eintrage erstellt wurden das Sah dann in Etwa so Aus:



Das Ganze Funktioniert so das wir im Oberen teil die GET Funktion ausgewählt haben und dann eine URL eingeben mussten wo die Daten drunter waren. Dann musste man nur noch «Send» Drücken und man hat die Einträge gesehen wie man unten im Bild sehen kann.

## Test mit REST Extension:

Die Tests mit REST sahen dann in etwa so aus:



The screenshot shows a REST client interface with two panels. The left panel displays a series of REST API requests, and the right panel shows the response for the most recent request.

**Left Panel (Requests):**

```
1 client > registration-api.rest > POST /api/registration
2
3 ### GET registration
4 Send Request
5 GET http://localhost:5000/api/registrations
6
7 ### POST registration
8 Send Request
9 POST http://localhost:5000/api/registration
10 Content-Type: application/json; charset=UTF-8
11
12 {
13   "name": "Bernd Werfel",
14   "email": "BerndWerfel@renault.com",
15   "phone": "031 745 56 14",
16   "priority": "Standard",
17   "service": "Heisswachsen",
18   "create_date": "2022-11-25T18:12:43.511Z",
19   "pickup_date": "2022-12-02T18:12:43.511Z"
20 }
21
22 ### Delete
23 Send Request
24 DELETE http://localhost:5000/api/registration/
```

**Right Panel (Response):**

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Access-Control-Allow-Origin: *
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 1986
6 ETag: W/"7c2-RBSDjKPDlRigOvVgDa7lwr12Ksw"
7 Date: Mon, 25 Nov 2024 13:06:12 GMT
8 Connection: close
9
10 [
11   {
12     "id": 1,
13     "name": "Anna Baier",
14     "email": "AnnaBaier@einrot.com",
15     "phone": "081 581 92 35",
16     "priority": "Tief",
17     "service": "Kleiner Service",
18     "create_date": "2022-11-13T18:12:43.511Z",
19     "pickup_date": "2022-11-25T18:12:43.511Z"
20   },
21   {
22     "id": 2,
23     "name": "Sven Möller".
```

Rechts im Bild sieht man den File der REST Extension mit dem GET Request wo wir die daten bekommen haben die eingetragen wurden Dies lief über eine URL ab wie man sehen kann. Links im Bild sieht man dann wie so Daten ausgesehen haben und das sie mit der «id» sortiert wurden Dies war ebenfalls erfolgreich wie man sehen kann.



## Vollständigkeit liste:

Auch haben wir noch geschaut ob wir dann alles erfüllt haben mit einer kleinen Checkliste der Ziele die auf GitHub waren das sah dann in etwa so aus:

NR	CHECK
A1	Vollständig
A2	Vollständig
A3	Vollständig
A4	Vollständig
A5	Vollständig
A6	Vollständig
A7	Vollständig
A8	Vollständig
A9	Vollständig
A10	Vollständig
A11	Vollständig

Anhand dieser Grafik Konntet wir unsere abriet mit Gutem gewissen abgeben und so auch sehr beruhigt sein. Zu dieser liste haben wir aber auch noch zusammen die Bewertung nageschaut und uns damit Gedanken gemacht und mit Hilfe dieser Tabelle auch zu frieden sagen das wir für uns das Projekt abgeschlossen haben.

## Auswerten

Bei der Auswertung ging es darum was wir gelehrt haben was wir verbessern könnten in Zukunft und noch ein Schluss Fazit. Auch kommt noch kurz ein Teil wie man das ganze startet und Benutzt.

### Lessons Learned:

Wir haben uns für diesen Punkt zusammengesetzt und wir haben beide Unterschiedliches gelehrt. Deshalb haben wir uns beide kurz ran gesetzt und entsprechend aufgeschrieben was wir gelehrt haben.

Nelo:

Ich habe vor allem gelehrt wie ich solche Frontend Applikationen an ein backend binden kann in JavaScript Gerade was das mit den verschiedenen Anfragen angeht und wie wir diese dann mit Post man und REST überprüfen können. Deshalb finde ich das ich wichtige Sachen neu gelehrt habe und jetzt auch gut verstehe.

Mali:

Das Projekt hat mir nicht nur großen Spaß bereitet, sondern mir auch geholfen, mein Verständnis für strukturiertes und zielorientiertes Arbeiten deutlich zu vertiefen. Obwohl ich bereits mit Bootstrap vertraut war, konnte ich mein Wissen erweitern und neue Ansätze für die Gestaltung und Umsetzung von Frontends kennenlernen. Dieses Projekt hat meine Begeisterung für Design und Frontend-Programmierung wieder neu entfacht und mir gezeigt, wie viel kreatives Potenzial in diesem Bereich steckt.

### Verbesserungen in der Zukunft

Wir sind beide zum Entschluss gekommen das wir in der Zukunft sicherlich besser eine so grosse Arbeit zu richtig zu planen und nicht nur «Freestyle zu arbeiten» Das war so der Hauptpunkt sonst fanden wir war alles ganz gut was unsere Arbeit anging Vor allem haben wir uns unter einander extrem gut verstanden und konnten uns gegenseitig sehr gut ergänzen gerade mit unseren verschiedenen stärken.

## Programm Richtig starten und Nutzen

Damit das Ticket System Funktioniert muss man nicht nur die Webseite starten sondern auch den Server starten dafür muss man den Server öffnen und dann die Console starten dann mit `cd server` in das Verzeichnis wechseln und dann `node server.js` eingeben um den Server zu starten. Wenn der Server gestartet ist kann man ihn automatisch mit dem Frontend verwenden und so das ticket System zum laufen bringen.

## Fazit:

Das Projekt hat uns beiden sehr gefallen und wir konnten uns extrem gut ergänzen. Gerade in der Zusammenarbeit hatten wir richtig spass und mussten uns nicht irgendwie durchkämpfen sondern konnten immer zusammen etwas fragen und immer gut zusammensetzen. Auch konnten wir unsere stärken im Team sehr gut ausbauen und so den anderen ergänzen und auch korrigieren. Die Kreativität wie wir an das Projekt angegangen sind war perfekt den wir haben wenn wir ein Problem hatten immer den Partner der einen anderen Lösungsansatz hatte oder wusste wie man das Problem beheben konnte.

Wir haben dem entsprechend viel gelehrt gerade auch was das zusammenarbeiten angeht gerade wenn es zur Kommunikation kommt mit GitHub dem pushen/ Pullen wie man mit dem umgeht aber auch der Verständigung im Team. Das war für uns beide etwas sehr spannendes was wir so in Zukunft sicher auch noch sagen können das es uns sehr geholfen hat.

## Referenzen & Quellen

Als Referenzen und auch Quellen Material haben wir am meisten das GitHub von Modul294 von Lukas Müller verwendet. Die Genauen quellen die wir verwendet haben sind Bei Quellen in einer Tabelle aufgelistet.

### Quellen:

Das waren so unsere Haupt Quellen die wir verwendet haben.

Name	Link
<b>Bootstrap</b>	<a href="#">Bootstrap · The most popular HTML, CSS, and JS library in the world.</a>
<b>Youtube</b>	<a href="#">(221) YouTube</a>
<b>Swagger</b>	<a href="#">Swagger Editor</a>
<b>ChatGPT</b>	<a href="#">ChatGPT</a>