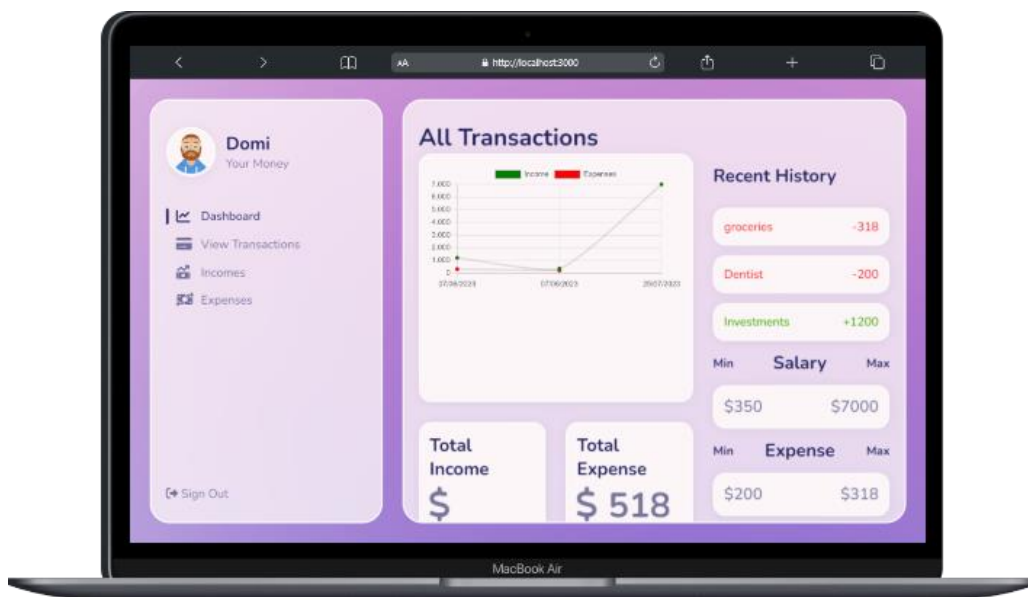


ZLI-Basislehrjahr

Abschlussprojekt

Ausgaben Tracker



Dokumentation

1. INHALTSVERZEICHNIS

Inhalt

1. INHALTSVERZEICHNIS	2
1.1 Änderungstabelle	3
1.2 Einleitung	4
2. Informieren	6
2.1 Beschreibung der Idee	6
3. Planen	6
3.1 Beschreibung der Use Cases, wer kann was machen	6
3.2 Beschreibung von allfälligen Risiken	6
3.3 Konzept für das UI mit Wireframes oder Mockups und Flow-Chart	7
3.4 Konzept für die Umsetzung in Form eines Klassendiagramms	8
.....	8
3.5 Flow-Diagramm	8
3.6 Aufstellung der verwendeten Technologien und Tools die Sie einsetzen werden	9
3.7 Liste mit Dingen, die Sie für dieses Projekt neu lernen müssen	9
3.8 Aufteilung der voraussichtlichen Arbeiten auf Arbeitspakete	9
3.9 Zeitplanung, z.B. in Form eines Gantt-Diagramms	10
4. Entscheiden	10
4.1 Logo auswahl	10
4.2 Farben auswahl	11
4.3 Technologie und Tools auswählen	11
5. Realisieren	12
5.1 Dashboard	12
5.2 Wichtige Komponenten	13
6. Kontrollieren	14
6.1 Testfälle Ausgaben Tracker	14
7. Inbetriebnahme	17
8. FactSheet	19
9. Auswerten	20
9.1 Reflexion	20
9.2 Quellenangaben	21

1.1 Änderungstabelle

	Datum	Geänderter Inhalt	Phase
1	05.06.2023	Dokument erstellt, Trello erstellt, Repository erstellt	Informieren
2	06.05.2023	Projektkonzept, Flow Chart, Trello Gantt diagramm erstellt	Planen
3	07.05.2023	Projektkonzept fertig, Klassendiagramm erstellt, Use Cases erstellt	Planen
4	12.05.2023		Planen
5	13.05.2023	Entscheiden und erstellen von Index	Entscheiden
6	14.05.2023	Logo und Farben, Technologien und Tools	Entscheiden
7	19.05.2023	Wichtige Komponenten	Realisieren
8	20.05.2023	Dashboard	Realisieren
9	21.05.2023	Wichtige Komponenten überarbeitet	Realisieren
10	26.05.2023	Testfälle durchgeführt und aufgeschrieben	Kontrollieren
11	27.05.2023	FactSheet erstellt	Factsheet
12	28.05.2023	Inbetriebnahme, Reflexionm Quellcode	Auswertung

1.2 Einleitung

Willkommen zur Dokumentation des Ausgaben Tracker's.

Diese Dokumentation dient dazu, einen umfassenden Überblick über mein Programmierprojekt zu geben und dabei zu helfen, die Funktionalitäten, das Design und die Umsetzung besser zu verstehen.

In dieser Dokumentation werde ich die Ziele und den Zweck des Projekts erläutern sowie die Herausforderungen, mit denen ich konfrontiert war. Ich werde einen detaillierten Einblick in die Architektur und den Aufbau des Programms geben und die verschiedenen Technologien und Tools vorstellen, die ich verwendet habe.

Ich hoffe, dass diese Dokumentation dabei hilft, das Projekt besser zu verstehen und es effektiv zu nutzen.

Ich wünsche Ihnen viel Erfolg bei der Verwendung des Ausgaben Tracker's.

Projekttitel	
Ausgaben Tracker	
Projektleiter (Name, E-Mail)	Coach (Name, E-Mail)
Dominik Schweizer, dominikschweizer06@gmail.com	Fabian Dennler
Projektstart	Projektende
05.06.2023	28.06.2023
Idee / Kurzbeschreibung (Um was geht es bei diesem Projekt?)	
Im Projekt geht es darum, seine Ausgaben mithilfe einer Applikation zu verfolgen. In der Applikation kann man verschiedene Einkommensquellen definieren und festlegen. Es sollte ein Dashboard geben, auf dem man mithilfe eines Diagramms genau sehen kann, wie viel man ausgibt und einnimmt.	
Ausgangslage (Was ist los? Warum wird der Projektauftrag gestellt?)	
Ich möchte meine Kenntnisse auf die Probe stellen und mich weiter in das Thema Backend vertiefen, da ich darin noch nicht so gut bin. Dazu gehören React und Node.js. Das Endprodukt sollte mir auch in Zukunft von Nutzen sein.	
Projektziele (Erwartete Resultate. Forderungen und Wünsche)	Prior.¹
Ich erwarte ein Endprodukt, welches man einfach bedienen kann.	A
Das Endprodukt sollte mindestens eine Grafik/Diagramm enthalten.	B
Man kann seine Einkommensquellen definieren.	A
Man sieht vergangene Ausgaben/Einnahmen.	B
Die Daten sollten in einer Datenbank gespeichert werden.	A
Vorgehensziele (Forderungen und Rahmenbedingungen)	Prior.¹
Ich möchte gerne die Pomodoro-Technik anwenden, denn damit kann ich mich gut konzentrieren und meine Ziele erfolgreich definieren.	B
Ich versuche so viel wie möglich an der Dokumentation zu arbeiten.	B
Lernziele (Welche Kompetenzen werden angestrebt?)	Prior.¹
Ich lerne React besser kennen und kann es anwenden.	A
Ich kann mir einen guten Zeitplan erstellen, an den ich mich halte.	B
Ressourcen (Personal, Sachmittel, Material?)	Menge
Laptop	1
Zeit	12 Tage
Projektentscheidung (Freigabeunterschrift)	Datum
Freigabe durch: Fabian Dennler	05.06.2023

2. Informieren

2.1 Beschreibung der Idee

Im Projekt geht es darum, seine Ausgaben mithilfe einer Applikation zu verfolgen. In der Applikation kann man verschiedene Einkommensquellen definieren und festlegen. Es sollte ein Dashboard geben, auf dem man mithilfe eines Diagramms genau sehen kann, wie viel man ausgibt und einnimmt.

Ich plane auch meine Daten in einer Datenbank zu speichern. Ich habe mich bei meinen Mitlernenden ein wenig informiert und bin zum Entschluss gekommen MongoDB zu nutzen.

3. Planen

3.1 Beschreibung der Use Cases, wer kann was machen

Wenn man die Applikation öffnet, erhält man einen umfassenden Überblick über Ausgaben und Einnahmen. Man kann zwischen verschiedenen Bereichen navigieren. Geplante Bereiche sind das Dashboard, Einnahmen, Ausgaben und ein Verlauf. Im Bereich Einnahmen sollte es möglich sein, Einnahmen zu definieren. Im Bereich Ausgaben sollte man die zuletzt getätigten Ausgaben eintragen können. Ebenfalls geplant ist, dass man seine Einnahmen und Ausgaben in verschiedene Kategorien einordnen kann. Falls man etwas falsch eingetragen hat, oder man den Eintrag aus der Datenbank löschen will, möchte ich noch eine DELETE Methode einbauen, welche sich hinter einem Papiorkorb-Button versteckt.

3.2 Beschreibung von allfälligen Risiken

Bei der Umsetzung könnten Fehler mit der Implementierung des Codes vorkommen. Es kann sein, dass der Code nicht funktioniert, oder es können unvorhergesehene Fehlerfunktionen auftreten.

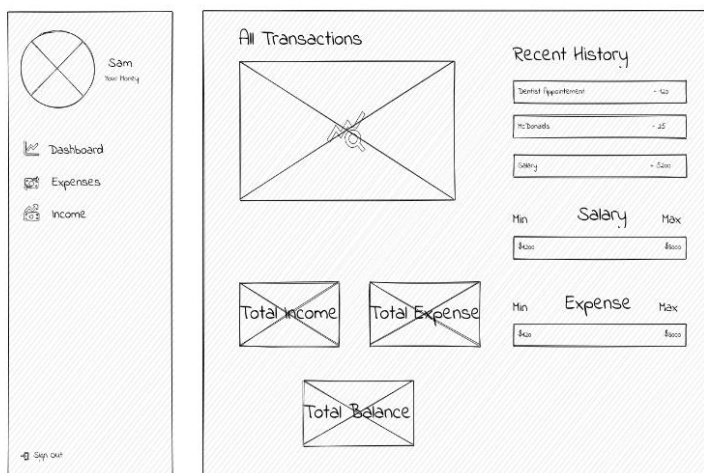
Ein weiteres Risiko könnte sein, dass ich zu wenig Zeit für die Umsetzung des Projektes habe, da ich ein eher komplexeres Projekt durchführen möchte.

Da ich noch nie so richtig eine Datenbank aufgesetzt habe, kann mir dies auch zum Verhängnis werden.

3.3 Konzept für das UI mit Wireframes oder Mockups und Flow-Chart

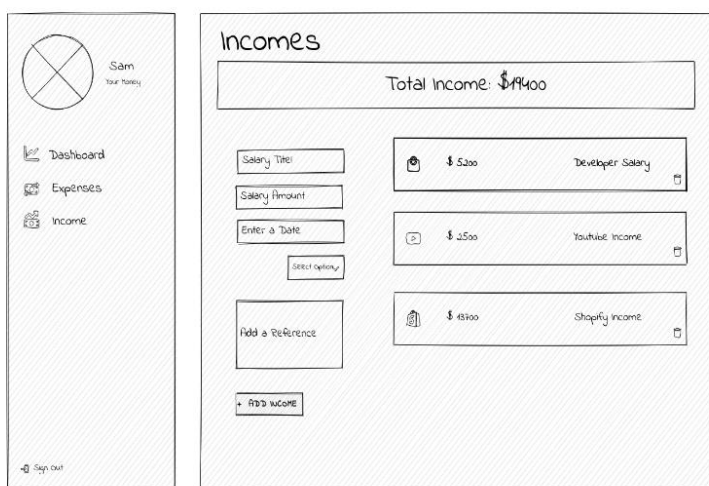
Ein Wireframe ist hilfreich, um die grundlegende Struktur und das Layout einer Website oder Anwendung zu visualisieren, bevor sie entwickelt wird. Es ermöglicht eine klare Kommunikation und Zusammenarbeit zwischen Designern, Entwicklern und anderen Projektbeteiligten. Durch die Verwendung eines Wireframes können Probleme frühzeitig erkannt und korrigiert werden, was letztendlich zu einem effizienteren und benutzerfreundlichen Endprodukt führt.

Ich habe mein Wireframe auf Miro erstellt. Auf Miro kenne ich mich schon ein wenig von vergangenen Projekten aus. Es gibt viele Vorlagen, welche man gut für ein Wireframe nutzen kann.



So stelle ich mir mein Dashboard vor. Auf der linken Seite, eine Navigation mit einem Profilbild und Name.

Auf der rechten Seite sieht man ein Diagramm mit den Ein- und Ausgaben und sogar mit einem Verlauf der letzten getätigten transaktionen.

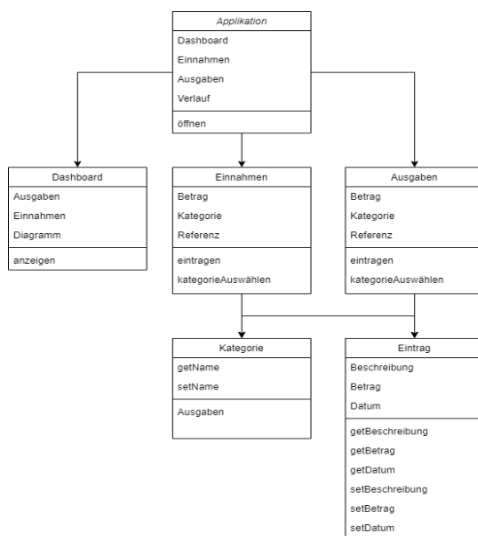


Im Income/Expenses Tab, kann man seine Daten eintragen und mit einer Kategorie versehen. Jede Kategorie sollte ein individuelles Icon haben, sodass man es besser erkennen kann.

Auf der rechten Seite kann man seine Daten löschen und ansehen.

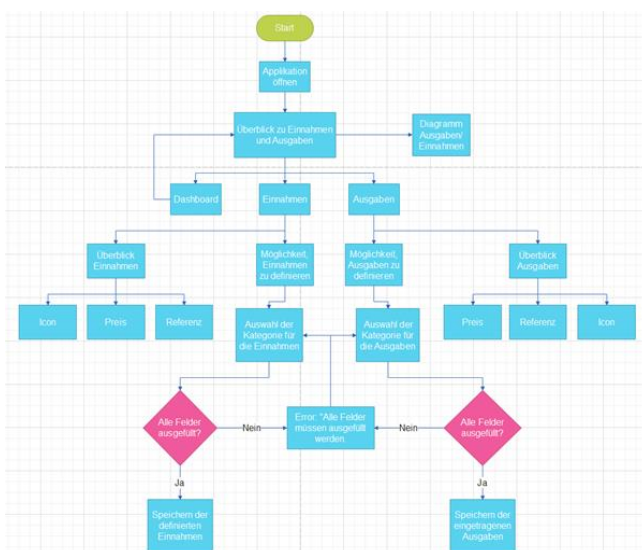
3.4 Konzept für die Umsetzung in Form eines Klassendiagramms

Ein Klassendiagramm visualisiert die Struktur eines Softwaresystems durch Rechtecke, die Klassen repräsentieren, und Pfeile, die deren Beziehungen darstellen. Es dient dazu, die Klassen, Attribute, Methoden und deren Verbindungen auf einen Blick zu erfassen und unterstützt bei der Entwicklung und Implementierung des Systems.



Ich habe ein Klassendiagramm auf Miro erstellt, welches zeigt, was in jeder Klasse steckt. Als oberste Klasse habe ich die ganze Applikation, welche sich danach auf drei verschiedene Klassen unterteilt. Da die Einnahmen und Ausgaben fast gleich sind, teilen sie sich die Klasse Kategorie und Eintrag.

3.5 Flow-Diagramm



Ein Flowdiagramm, ist ein visuelles Hilfsmittel zur Darstellung von Prozessen, Abläufen oder Entscheidungsstrukturen. Es besteht aus verschiedenen Symbolen und Linien, die den Fluss von Informationen, Aufgaben oder Entscheidungen innerhalb eines Systems oder Prozesses anzeigen.

Bei meinem Diagramm sieht man den Ablauf und die Möglichkeiten, welche man bei meiner Webapplikation hat.

3.6 Aufstellung der verwendeten Technologien und Tools die Sie einsetzen werden

In meinem Projekt werde ich in **Visual Studio Code** mit **React** arbeiten. Dazu werde ich noch eine **Mongo Datenbank** aufsetzen. Für mein Wireframe benutze ich **WonderShare EdrawMax**. Für meine Zeitplanung werde ich **Trello** benutzen. Trello kann mir gleichzeitig auch ein Gantt-Diagramm erstellen. Für die Datenbank abfrage werde ich **Postman** benutzen.

3.7 Liste mit Dingen, die Sie für dieses Projekt neu lernen müssen



mongo DB

Für mein Projekt werde ich MongoDB neu lernen müssen, da ich es noch nie gebraucht habe.



POSTMAN

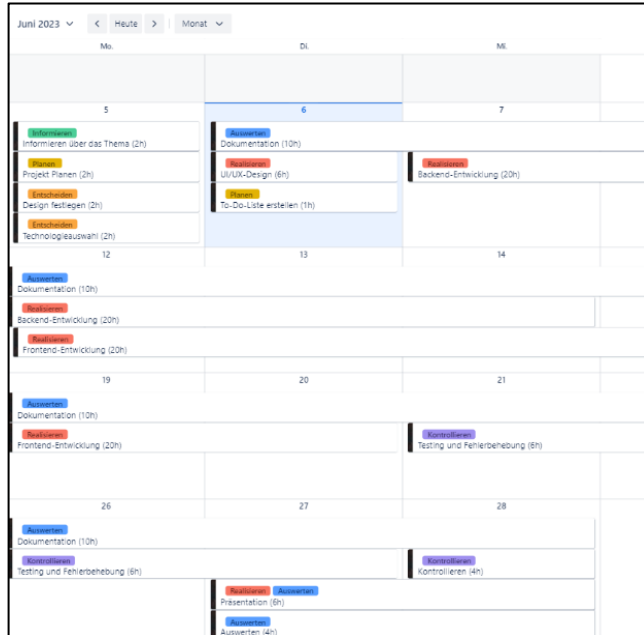
Zwar haben wir Postman schon im ÜK angeschaut, aber ich habe es noch nicht ganz verstanden, somit werde ich Postman nochmals genauer anschauen und verstehen.

3.8 Aufteilung der voraussichtlichen Arbeiten auf Arbeitspakete



Ich habe die Aufteilung der Arbeitspakete in Trello bei meiner ToDoListe integriert.

3.9 Zeitplanung, z.B. in Form eines Gantt-Diagramms



In Trello kann man die Karten zu einem gewissem Datum zuweisen. Wenn man danach auf die Kalender Ansicht wechselt, sieht man ein Gantt-Diagramm.

Das Gannt-Diagramm kann mir sehr hilfreich sein, wenn ich nicht weis was ich zurzeit machen soll oder als nächstes machen werde.

4. Entscheiden

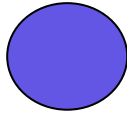
4.1 Logo auswahl

Da ich noch kein Logo habe, wollte ich noch kurz eines von einem AI generieren lasse. Ich gab meine Präferenzen an. Da ich ein sehr kleines und schlichtes Logo möchte, entschied ich mich für das unterste.

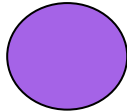


4.2 Farben auswahl

Primärfarbe =



Sekundärfarbe =



4.3 Technologie und Tools auswählen



Ich habe mich entschieden das Frontend mit React zu gestalten, da es sehr verbreitet ist und einfach zu bedienen ist.



Für meine Laufzeitumgebung habe ich mich für Node.js entschieden, da wir im ZLI schon viel mit dem gearbeitet haben und ich vertraut mit Node.js bin.



Um eine Datenbank aufzusetzen habe ich mich für MongoDB als Technologie entschieden, da ich Mitlernende habe, welche mir das Erklären können und es einfach zu bedienen ist. Bei MongoDB kann man auch ein gratis Cluster erstellen, welches ich gut nutzen kann.



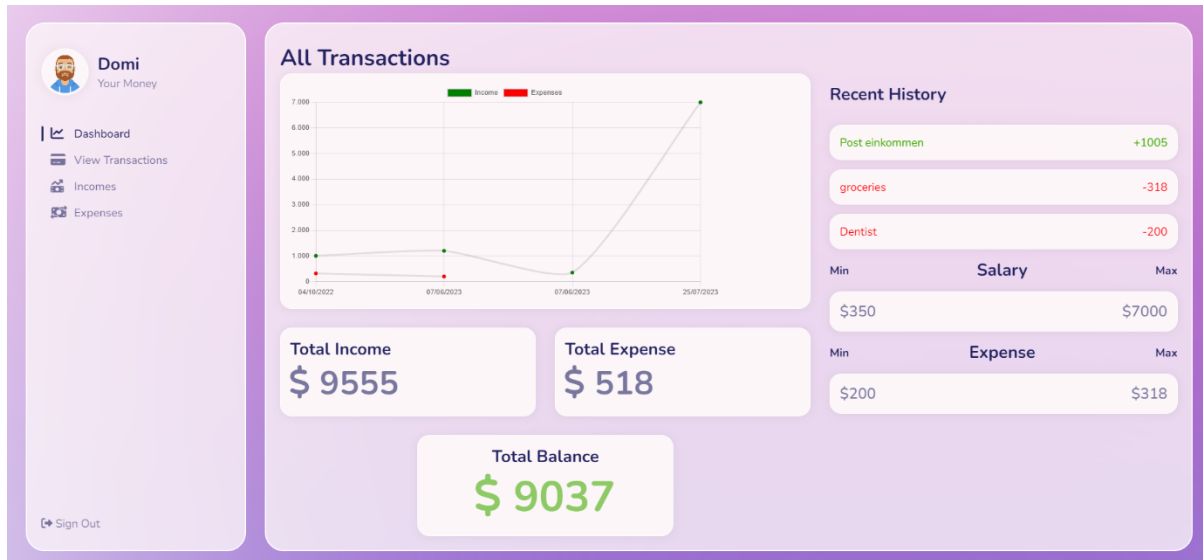
Ich bin sehr mit VisualStudioCode vertraut und arbeite auch gerne damit, deswegen habe ich mich entschieden mein Projekt mit VSC zu machen.



Postman hat ein schönes UI und ich finde es einfach zu bedienen. Es wird mir helfen POST, GET und DELETE Methoden zu testen.

5. Realisieren

5.1 Dashboard



Das Dashboard spielt eine zentrale Rolle in der Applikation, da es alle relevanten Daten auf einer einzigen Seite zusammenführt und diese in Form eines übersichtlichen Diagramms darstellt. Es bietet einen schnellen und einfachen Überblick über wichtige Kennzahlen und ermöglicht es den Benutzern, die Informationen auf einen Blick zu erfassen.

Ein herausragendes Merkmal des Dashboards ist die Berechnung und Anzeige der Gesamtsumme der Balance. Diese Kennzahl gibt den Benutzern eine schnelle Übersicht über den gesamten finanziellen Status. Diese Funktion ermöglicht es den Benutzern, die finanzielle Gesamtsituation besser zu verstehen.

Ein weiterer wichtiger Aspekt des Dashboards ist seine Anpassungsfähigkeit an die vorhandenen Daten. Das bedeutet, dass sich das Dashboard je nach den vorliegenden Daten automatisch anpasst und aktualisiert. Neue Datenpunkte oder Änderungen in den vorhandenen Daten führen zu entsprechenden Aktualisierungen im Dashboard. Dadurch bleibt das Dashboard immer aktuell und liefert den Benutzern stets die neuesten Informationen.

Insgesamt bietet das Dashboard der Applikation eine umfassende und flexible Darstellung der Daten. Es ermöglicht es den Benutzern, wichtige Informationen zu überblicken, Vergleiche anzustellen und fundierte Entscheidungen zu treffen. Durch die dynamische Anpassung an die vorliegenden Daten bleibt das Dashboard stets aktuell und relevant für die Benutzer.

5.2 Wichtige Komponenten

```
exports.addIncome = async(req, res) => {
  const { title, amount, category, description, date} = req.body

  const income = IncomeSchema({
    title,
    amount,
    category,
    description,
    date
  })
}
```

Hier sieht man eine sehr wichtige Funktion. Das ist die addIncome() Funktion, mit der man später die Daten, welche unter IncomeSchema steht in die Datenbank POSTEN kann.

```
router.post('/add-income', addIncome)
  .get('/get-incomes', getIncomes )
  .delete('/delete-income/:id', deleteIncome)
  .post('/add-expense', addExpense)
  .get('/get-expenses', getExpense )
  .delete('/delete-expense/:id', deleteExpense)

module.exports = router
```

Hier sehen wir die «Routes» also die Endpoint. Wir bestimmen hier auch die HTTP-Methode bestimmt, also wird unsere addIncome() mit einer POST-Methode versehen, hierzu wird noch der Endpoint (URL) bestimmt.

```
const mongoose = require('mongoose')

const db = async () => {
  try {
    mongoose.set("strictQuery", false)
    await mongoose.connect(process.env.MONGO_URL)
    console.log("Db connected")
  } catch (error) {
    console.log("Db connection error" + error)
  }
}

module.exports = {db}
```

Hier sieht man den Code, der sich mit der Datenbank verbindet. Da unsere «MONGO_URL» im .env File gespeichert ist, müssen wir zuerst darauf zugreifen. Im Code sieht man auch, dass wenn es einen Fehler gibt, ein String kommt, welcher dir sagt, dass es nicht geklappt hat sich mit der Datenbank zu verbinden.

6. Kontrollieren

6.1 Testfälle Ausgaben Tracker

Testfall Nr.	1
Beschreibung	Zwischen den verschiedenen Seiten wechseln (Dashboard, Incomes, Expenses)
Voraussetzung	Keine
Testschritte	<ul style="list-style-type: none"> - Mit Mauszeiger über Dashboard und linke Maustaste drücken. - Mit Mauszeiger über Incomes und linke Maustaste drücken. - Mit Mauszeiger über Expenses und linke Maustaste drücken.
Erwartetes Ergebnis	Wechselt zur ausgewählten Seite.
Testergebnis	erfüllt
Testdatum	27.06.2023
Bemerkungen	

Testfall Nr.	2
Beschreibung	Man kann bei den Seiten Incomes und Expenses ein neues Einkommen oder eine neue Ausgabe posten.
Voraussetzung	Es müssen alle Felder korrekt ausgefüllt sein.
Testschritte	Man wechselt zur Seite Incomes oder Expenses und trägt auf der linken Seite den Titel, Betrag, Datum, Kategorie und eine Beschreibung korrekt ein. Danach drückt man auf «+ Add Income/Expense»
Erwartetes Ergebnis	Wird auf die Datenbank gepostet und auf der Rechten Seite der Jeweiligen Seite angezeigt.
Testergebnis	erfüllt
Testdatum	27.06.2023
Bemerkungen	

Testfälle Ausgaben Tracker

Testfall Nr.	3
Beschreibung	Bei einem Post in die Datenbank wird ein Individuelles Icon für jede Kategorie angezeigt.
Voraussetzung	Es müssen alle Felder korrekt ausgefüllt sein.
Testschritte	Man füllt alle Felder Korrekt ein und wählt eine Kategorie, danach drückt man auf «+ Add Income/Expense» und schaut ob auf der Rechten Seite das passende Icon angezeigt wird. Dieses Verfahren wiederholt man für jedes Icon.
Erwartetes Ergebnis	Es gibt für jede Kategorie ein individuelles Icon.
Testergebnis	erfüllt
Testdatum	27.06.2023
Bemerkungen	

Testfall Nr.	4
Beschreibung	Wen man einen Falschen Wert in einem InputFeld eingibt, gibt es eine Meldung. (Validierung)
Voraussetzung	Keine
Testschritte	Man füllt alle Felder korrekt aus, ausser bei «Salary/Expense Amount» gibt man anstatt einer Zahl, einen oder mehrere Buchstaben ein.
Erwartetes Ergebnis	Es kommt eine Fehlermeldung.
Testergebnis	erfüllt
Testdatum	27.06.2023
Bemerkungen	

Testfälle Ausgaben Tracker

Testfall Nr.	5
Beschreibung	Man kann einen Post löschen. Es wird von allen Seiten gelöscht. (aus der Datenbank)
Voraussetzung	Keine
Testschritte	<ul style="list-style-type: none"> - Wechselt zu Incomes oder Expenses. - Drückt mit der linken Maustaste auf den Papierkorb. - Wechselt zu Dashboard und schaut, ob es noch angezeigt wird.
Erwartetes Ergebnis	Der Post wird mit einer Delete Methode aus der Datenbank gelöscht.
Testergebnis	erfüllt
Testdatum	27.06.2023
Bemerkungen	

Testfall Nr.	6
Beschreibung	Diagramm ändert sich bei Post oder Delete.
Voraussetzung	Keine
Testschritte	<ul style="list-style-type: none"> - Wechselt zu Incomes oder Expenses. - Macht einen Post oder Delete. - Wechselt zu Dashboard und schaut, ob sich das Diagramm ändert.
Erwartetes Ergebnis	Es wird angepasst und verändert sich je nach Wert.
Testergebnis	erfüllt
Testdatum	27.06.2023
Bemerkungen	Man kann beim Diagramm einstellen, ob man nur die Ansicht von Income oder Expenses haben möchte.

7. Inbetriebnahme

Um das Projekt zu starten, muss man es zuerst klonen.

```
git clone https://github.com/domixsch/abschlussprojekt\_zli.git
```

Um auf die Datenbank zugreifen zu können, muss man zuerst noch ein Kostenfreies Cluster bei MongoDB erstellen und den Connection String in das bereits erstellte .env File einfügen. Danach muss man nur noch das "Password" mit seinem Passwort ersetzen.

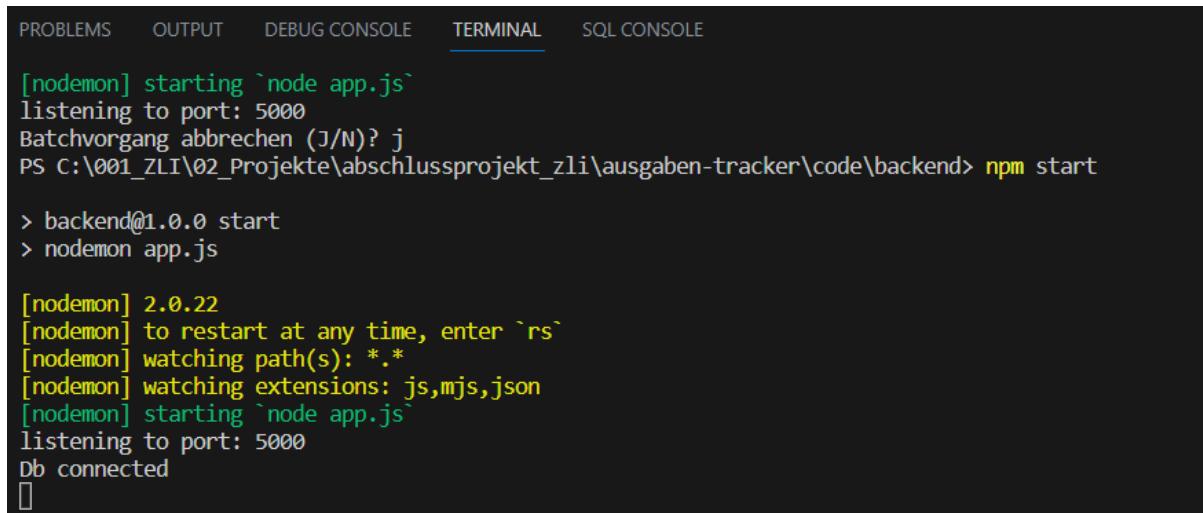
Danach sollte es wie folgt aussehen.

```
backend > .env
1  PORT = 5000
2
3  MONGO_URL = mongodb+srv://Exampelusername:<Password>@cluster0.fxw6zpd.mongodb.net/?retryWrites=true&w=majority
4
```

Danach geht man im Terminal in das Backendverzeichnis und gibt die folgenden Befehle ein:

```
cd backend  
npm install  
npm start
```

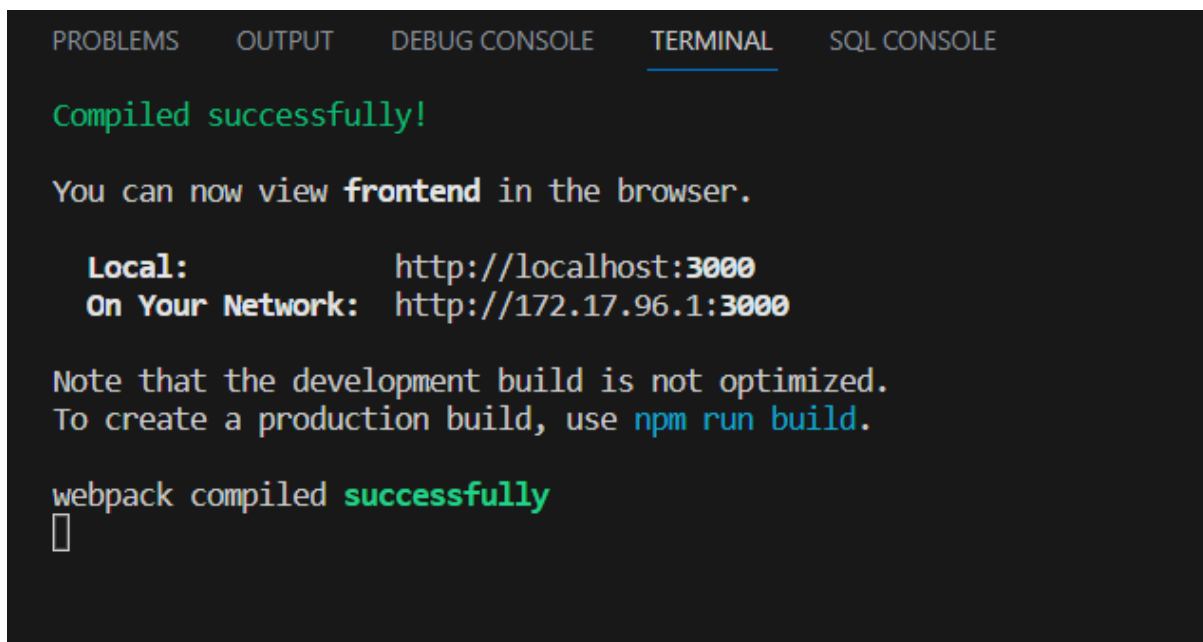
Das sollte dann in etwa so aussehen:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  SQL CONSOLE  
  
[nodemon] starting `node app.js`  
listening to port: 5000  
Batchvorgang abbrechen (J/N)? j  
PS C:\001_ZLI\02_Projekte\abschlussprojekt_zli\ausgaben-tracker\code\backend> npm start  
  
> backend@1.0.0 start  
> nodemon app.js  
  
[nodemon] 2.0.22  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,json  
[nodemon] starting `node app.js`  
listening to port: 5000  
Db connected  
█
```

Nachdem man diese Schritte erledigt hat, öffnet man ein neues Terminal und wechselt in das Frontendverzeichnis und führt die folgenden Commands aus:

```
cd ..  
cd frontend  
npm install  
npm start
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  SQL CONSOLE  
  
Compiled successfully!  
  
You can now view frontend in the browser.  
  
Local: http://localhost:3000  
On Your Network: http://172.17.96.1:3000  
  
Note that the development build is not optimized.  
To create a production build, use npm run build.  
  
webpack compiled successfully  
█
```

8. FactSheet

DOMINIK SCHWEIZER

ZLI

Ausgaben Tracker



Das Projekt

Wie der Name es schon sagt, kann man seine Ein- und Ausgaben erfassen. Es war noch nie so einfach, seine Finanzen im Überblick zu behalten. Man kann in Kürze eine Ein- und Ausgabe erfassen und diese in der Datenbank speichern. Damit es noch übersichtlicher wird, wurde ein Diagramm in der Webapplikation eingebaut. Es wurden auch viele Icons zu jeder Kategorie zugeteilt, um sich das viele lesen zu sparen.

Idee

Ich kam auf die Idee, einen Ein- und Ausgaben Tracker zu programmieren, da ich mich sehr für das Thema Finanzen interessiere und gerne eine Übersicht darüber habe. Ich wollte mich selbst auch mit diesem Projekt auf die Probe stellen, den in diesem Projekt habe ich ein Backend mit einer Datenbank sowohl auch ein Frontend mit React programmiert.

Technologien



React



Node.js



MongoDB

Facts



95.2% Javascript



4.8% HTML

Features

Finanz Diagramm

Animierter Hintergrund

Sauberes User-Interface




→


9. Auswerten

9.1 Reflexion

Insgesamt bin ich mit meinem Ausgabentracker-Projekt zufrieden und betrachte es als eine wertvolle Erfahrung, ich habe vieles dabei gelernt. Die Umsetzung des Backends war definitiv ein Erfolg. Es war spannend, die verschiedenen Routen und Datenbankoperationen zu implementieren, um die Funktionalität des Trackers zu ermöglichen. Durch die Auseinandersetzung mit dem Backend konnte ich meine Kenntnisse in Bezug auf Datenbankstrukturen, API-Entwicklung und serverseitige Logik erweitern.

Besonders stolz bin ich auch auf meine Fähigkeit, das Projekt zu planen und den Zeitrahmen einzuhalten. Durch die detaillierte Planung und die Aufteilung des Projekts in Meilensteine konnte ich effizient arbeiten und sicherstellen, dass ich im Zeitplan blieb. Die Planung hat mir geholfen, den Überblick zu behalten und Prioritäten zu setzen, was sich letztendlich positiv auf das Endprodukt ausgewirkt hat.

Es gab jedoch einige Herausforderungen im Umgang mit MongoDB und der Verbindung dazu. Ich musste mich intensiv mit der Konfiguration und den entsprechenden Treibern auseinandersetzen, um eine reibungslose Verbindung herzustellen. Diese Probleme haben zwar etwas Zeit gekostet, aber sie haben mir auch die Möglichkeit gegeben, meine Fehlerbehebungsfähigkeiten zu verbessern und ein besseres Verständnis für die Interaktion mit Datenbanken zu entwickeln.

Ein weiterer Aspekt, den ich als Erfolg betrachte, ist das Factsheet. Die Gestaltung und Präsentation des Factsheets ist mir gut gelungen und es macht einen professionellen Eindruck. Ich habe viel Wert auf eine klare und ansprechende Darstellung der Informationen gelegt, um den Benutzern eine benutzerfreundliche Erfahrung zu bieten.

Insgesamt war die Entwicklung meines Ausgabentrackers ein lehrreiches und bereicherndes Projekt. Ich habe nicht nur technische Fähigkeiten verbessert, sondern auch gelernt, wie wichtig es ist, Projekte zu planen und den Zeitrahmen im Auge zu behalten. Die Herausforderungen mit MongoDB haben mir gezeigt, wie wichtig es ist, flexibel zu sein und Probleme aktiv anzugehen. Ich bin stolz auf das erreichte Ergebnis und freue mich darauf, mein Projekt weiterzuentwickeln und neue Erkenntnisse zu gewinnen.

9.2 Quellenangaben

Link zur Quelle	Beschreibung
https://www.mongodb.com/docs/	MongoDB Dokumentation
https://nodejs.org/en/docs	Node.js Dokumentation
https://devdocs.io/react/	React Dokumentation
https://chatgpt.ch/	Probleme und Fehler beheben
https://www.youtube.com/watch?v=mhM-bITHBz8&t=146s&ab_channel=DailyTuition	Backend Tutorial
https://www.youtube.com/watch?v=i0JesTevAcA&t=170s&ab_channel=TheCodeDealer	React Frontend Tutorial
https://fontawesome.com/	Icons und Fonts