

Audit 3

Finanzratgeber

im Rahmen des Entwicklungsprojekts

mit Schwerpunkt Web Development

von Hendrik Runde und Alexander Zellmann



Inhalt

- Anforderungen
 - Use Cases
 - Proof of Concepts
 - Tech Stack
 - Externe API's
 - Rapid Prototyp
 - Ausblick Audit 4
-

Anforderungen

- **Manuelle Eingabe von Kontodaten**

Das System muss eine benutzerfreundliche Oberfläche bereitstellen, um manuell Einkommens- und Ausgabedaten eingeben zu können.

Das System sollte dem Benutzer ermöglichen, relevante Informationen präzise und einfach einzutragen.

Das System muss eine Plausibilitätsprüfung der manuell eingegebenen Daten durchführen, um sicherzustellen, dass sie den erwarteten Format- und Wertebereichen entsprechen.

- **Übersichtliche Darstellung**

Das System sollte verschiedene Diagrammtypen verwenden, um die freie Liquidität, Einnahmen und Ausgaben klar und leicht verständlich darzustellen.

Das System sollte dem Benutzer eine klare und visuelle Darstellung seiner finanziellen Situation anzeigen.

Das System muss die Prognosen visuell ansprechend und leicht verständlich darstellen, um den Benutzer für die Altersvorsorge zu motivieren.

- **Ausgaben kategorisieren**

Das System muss vordefinierte Ausgabekategorien bereitstellen.

Das System sollte dem Benutzer ermöglichen, Ausgaben bestimmten Kategorien zuzuordnen, um einen detaillierten Überblick zu erhalten.

Das System muss dem Benutzer die Möglichkeit bieten, auch manuell Ausgabekategorien definieren zu können.

- **Prognosen erstellen**

Das System ermöglicht es dem Benutzer, manuell seine Sparrate einzutragen.

Das System muss dem Benutzer die Möglichkeit bieten, verschiedene Anlageoptionen auszuwählen.

Das System muss die Prognosen in einer zeitlichen Entwicklung bis zur Rente darstellen.

Das System sollte dem Benutzer klare Erläuterungen zu den Annahmen und Berechnungen der Prognosen bieten.

Use Cases

Registrierung

use case Registrierung

actors Benutzer

precondition

Der Benutzer besitzt einen Computer oder ein mobiles Endgerät mit Internetzugang.

Die Webanwendung ist zudem erreichbar und funktionsfähig.

main flow

1. Der Benutzer ruft die Webseite auf, um sich zunächst zu registrieren.
2. Der Benutzer gibt seine Informationen ein (Vorname, Nachname, Alter, E-Mail-Adresse und ein Passwort seiner Wahl)
3. Das System prüft, ob die eingegebenen Daten korrekt sind, z.B. ob die E-Mail-Adresse gültig und ob das Passwort sicher genug ist.
4. Wenn die Daten akzeptiert werden, erstellt das System einen neuen, personalisierten Account für den Benutzer.
5. Der Benutzer bekommt eine Bestätigungsmail, um die Registrierung zu verifizieren.
6. Nach erfolgreicher Bestätigung wird der Benutzer zur Hauptseite, bereits angemeldet, weitergeleitet.
7. Der Benutzer definiert langfristige und mittelfristige Ziele im Bereich der Altersvorsorge.

postcondition Ein neuer Benutzer-Account wurde erstellt.

end Registrierung

Use Cases

Anmeldung

use case Anmeldung

actors Benutzer

precondition Benutzer besitzt einen Account, ist jedoch noch nicht angemeldet.

main flow

1. Der Benutzer gibt seine E-Mail und sein Passwort ein.
2. Das System prüft die eingegebenen Daten auf Korrektheit.
3. Der Benutzer ist angemeldet und wird auf die Hauptseite weitergeleitet.

postcondition

Erfolgreiche Anmeldung des Benutzers

exceptional flow Fehlerhafte Daten

- 2a. Das System stellt eine Inkonsistenz der Daten fest.
4. Der Benutzer wird auf den Fehler hingewiesen.

postcondition Fehlerhafte Anmeldedaten

Der Benutzer wurde nicht angemeldet und auch darüber informiert.

end Anmeldung

Use Cases

Manuelle Eingabe von Kontodaten

use case Manuelle Eingabe

actors Benutzer

precondition Der Benutzer hat sich erfolgreich angemeldet und möchte nun seine Kontodaten einpflegen.

main flow

1. Der Benutzer gibt Einkommens- (Jobbezeichnung + Betrag) und Ausgabedaten (Ausgabebeträge in passenden Ausgabekategorien) manuell in die benutzerfreundliche Oberfläche ein.
2. Das System führt eine Plausibilitätsprüfung der manuell eingegebenen Daten durch.
3. Der Benutzer besitzt die Berechtigung, seine Daten jederzeit anzupassen.
4. Der Benutzer bestätigt die manuelle Eingabe seiner Kontodaten.

postcondition Die manuell eingegebenen Kontodaten wurden erfolgreich erfasst und sind im System einsehbar.

exceptional flow Fehlerhafte oder unvollständige Daten

- 3a. Das System erkennt fehlerhafte oder unvollständige Daten.
- 3b. Der Benutzer wird auf die Fehler hingewiesen.
- 3c. Der Benutzer hat die Möglichkeit, fehlerhafte Daten zu korrigieren.

postcondition Fehlerhafte Daten

Die manuelle Eingabe von Kontodaten konnte nicht abgeschlossen werden, der Benutzer wurde darüber informiert.

end Manuelle Eingabe

Use Cases

Prognosen erstellen

use case Prognosen erstellen

actors Benutzer

precondition Der Benutzer hat sich erfolgreich angemeldet und seine Kontodaten manuell eingegeben.

main flow

1. Der Benutzer navigiert sich zur Startseite.
2. Der Benutzer trägt manuell seine Sparrate ein.
3. Der Benutzer wählt eine der möglichen Anlageoptionen aus.
4. Das System stellt die Kapitalentwicklung auf den Zeitraum bis zum Renteneintritt visuell dar.

postcondition Prognose wurde erstellt

Dem Benutzer wurde erfolgreich eine Prognose erstellt und angezeigt.

end Prognosen erstellen

Use Cases

Übersichtliche Darstellung

use case Übersichtliche Darstellung

actors Benutzer

precondition Der Benutzer hat sich erfolgreich angemeldet, seine Kontodaten manuell eingegeben und möchte sich seine Kontodaten anzeigen lassen.

main flow

1. Der Nutzer navigiert sich zur Startseite.
2. Das System verwendet verschiedene Diagrammtypen, um die freie Liquidität, Einnahmen und Ausgaben klar und leicht verständlich darzustellen.
3. Der Benutzer kann sich eine klare und visuelle Darstellung seiner finanziellen Situation anzeigen lassen.

postcondition

Der Benutzer hat einen übersichtlichen Überblick über seine freie Liquidität, Einnahmen und Ausgaben.

end Übersichtliche Darstellung

Proof of Concepts

| Was wird umgesetzt ? |
|---|
| Manuelle Eingabe von Kontodaten |
| Der Benutzer kann manuell Einkommens- und Ausgabendaten über eine benutzerfreundliche Oberfläche eingeben. |
| Exit-Kriterien (Erfolgskriterien) |
| Einkommens- und Ausgabendaten werden erfolgreich eingegeben und im System gespeichert. Der Benutzer erhält eine Bestätigung über die erfolgreiche manuelle Eingabe. |
| Fail Kriterien |
| Der Benutzer lässt erforderliche Felder leer. Die eingegebenen Daten entsprechen nicht den definierten Formaten. |
| Fallbacks |
| Bei unvollständigen Feldern oder unzulässigen Daten zeigt das System eine klare Fehlermeldung an und gibt dem Benutzer Anleitungen zur Problembeseitigung. Technische Probleme während des Registrierungs- vorgangs werden durch eine allgemeine Fehlermeldung behandelt, und der Benutzer wird aufgefordert, es zu einem späteren Zeitpunkt erneut zu versuchen. |

| Was wird umgesetzt ? |
|--|
| Übersichtliche Darstellung |
| Das System stellt verschiedene Diagrammtypen bereit, um freie Liquidität, Einnahmen und Ausgaben in klaren und leicht verständlichen Grafiken darzustellen. Diese Diagramme sind zoom- und filterbar. |
| Exit-Kriterien (Erfolgskriterien) |
| Die Diagramme werden korrekt generiert und auf der Benutzeroberfläche angezeigt. Benutzer können erfolgreich zoomen und filtern, um detaillierte Informationen zu erhalten. |
| Fail Kriterien |
| Diagramme werden nicht korrekt generiert oder angezeigt. Zoom- und Filterfunktionen funktionieren nicht wie erwartet. |
| Fallbacks |
| Bei Problemen mit der Diagrammerstellung oder Anzeige erhält der Benutzer eine klare Fehlermeldung und wird dazuaufgefordert, es erneut zu versuchen. Technische Schwierigkeiten mit Zoom- oder Filterfunktionen werden durch eine allgemeine Fehlermeldung behandelt, und der Benutzer wird aufgefordert, es später erneut zu versuchen. |

| Was wird umgesetzt ? |
|---|
| Ausgaben kategorisieren |
| Das System stellt vordefinierte Kategorien für Ausgaben bereit und ermöglicht es dem Benutzer, Ausgaben diesen zuzuordnen. |
| Exit-Kriterien (Erfolgskriterien) |
| Der Benutzer kann Ausgaben erfolgreich den vordefinierten Kategorien zuordnen. Das System speichert die Zuordnung korrekt in der Datenbank. |
| Fail Kriterien |
| Der Benutzer kann keine Ausgaben den vordefinierten Kategorien zuordnen. Technische Schwierigkeiten führen zu einer fehlerhaften Speicherung der Zuordnung. |
| Fallbacks |
| Bei Fehlern während des Zuordnungsprozesses erhält der Benutzer eine klare Fehlermeldung und wird aufgefordert, es erneut zu versuchen. Technische Probleme bei der Speicherung werden durch eine allgemeine Fehlermeldung behandelt, und der Benutzer wird darauf hingewiesen, es später erneut zu versuchen. |

Proof of Concepts

| Was wird umgesetzt ? |
|--|
| Prognosen erstellen |
| Das System ist fähig, Prognosen zu erstellen. Die Prognosen zeigen, wie sich das Kapital über die Jahre entwickelt, basierend auf der festgelegten monatlichen Sparrate und Zinsseszins. Die Prognosen beziehen sich auf die gewählte Art der Altersvorsorge. |
| Exit-Kriterien (Erfolgskriterien) |
| Das System erstellt Prognosen erfolgreich. Die erstellten Prognosen zeigen die Kapitalentwicklung über die gewählte Zeitspanne und berücksichtigen die gewählte Altersvorsorge. |
| Fail Kriterien |
| Das System kann keine Prognosen erstellen. Die erstellten Prognosen berücksichtigen nicht die gewählte Altersvorsorge oder sind fehlerhaft. |
| Fallbacks |
| Bei Problemen mit der Prognoseerstellung erhält der Benutzer eine klare Fehlermeldung und Anweisungen zur Fehlerbehebung. Falls die Prognosen Fehler aufweisen, wird dem Benutzer angezeigt, dass die Einstellungen überprüft werden müssen, und klare Anweisungen zur Korrektur werden bereitgestellt. |

| Was wird umgesetzt ? |
|---|
| Visuelle Darstellung |
| Das System kann Prognosen visuell darstellen, um den Benutzer für die Altersvorsorge zu motivieren. |
| Exit-Kriterien (Erfolgskriterien) |
| Die visuelle Darstellung der Prognosen ist ansprechend und leicht verständlich. Benutzer können die Entwicklung ihres Kapitals über die Jahre klar erkennen. |
| Fail Kriterien |
| Die visuelle Darstellung der Prognosen ist unklar oder fehlerhaft. Benutzer können die Informationen nicht leicht verstehen. |
| Fallbacks |
| Bei Problemen mit der visuellen Darstellung erhalten Benutzerklare Anweisungen zur Fehlerbehebung. Falls die Darstellung nicht den erwarteten Anforderungen entspricht, wird dem Benutzer erklärt, wie er die Einstellungen anpassen kann. |

Tech Stack und externe API`s

Frontend

HTML

- weitverbreitete textbasierte Auszeichnungssprache
- Darstellung von Webseiten in Browsern

CSS

- Computersprache
- Aufbauen und Strukturieren von Webseiten

JavaScript

- Programmiersprache
 - ermöglicht Interaktion zwischen Anwender und System
-

Tech Stack und externe API's

Frameworks

Vue.js

- JavaScript-Webframework
- Erstellen von Single-Page-Webanwendungen

Bootstrap

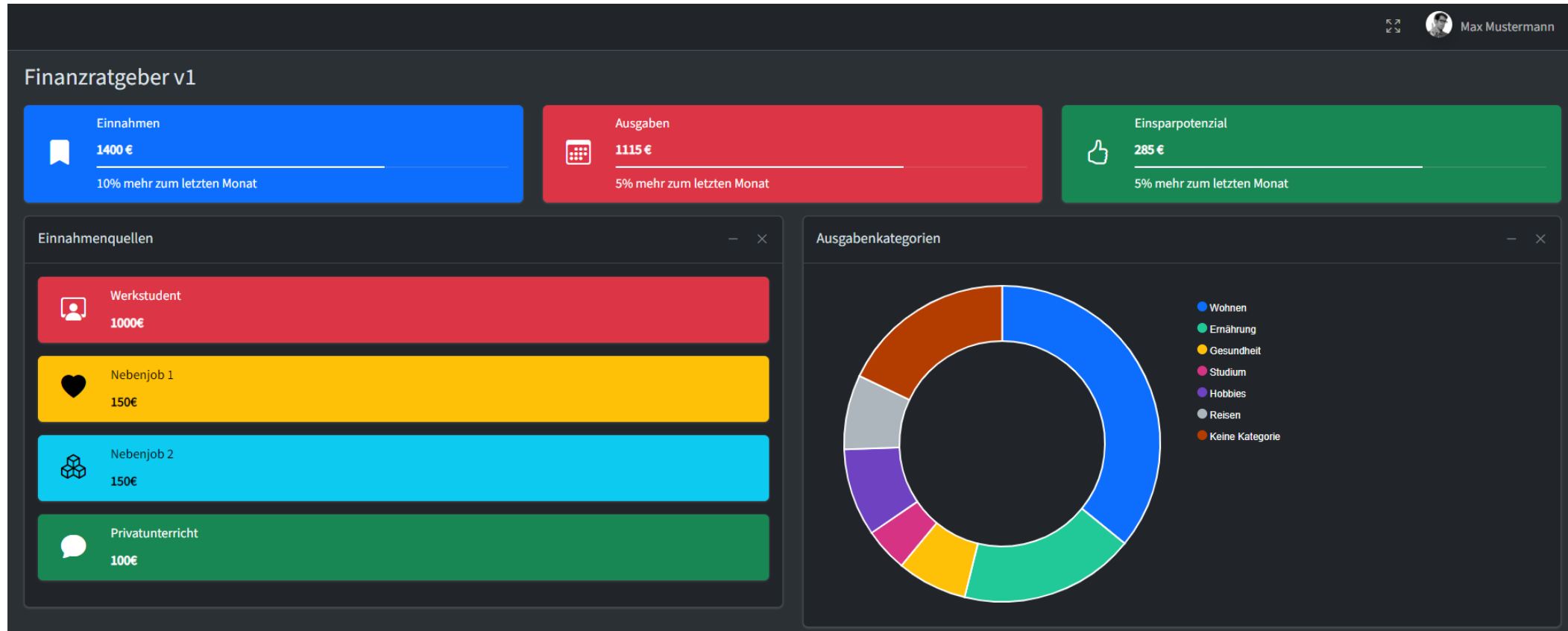
- Frontend-CSS-Framework
- Gestaltungsvorlagen basierend auf HTML und CSS
- meistverbreitete Oberflächengestaltungselemente wie z.B. Formulare, Buttons, Tabellen, usw.

Bibliotheken

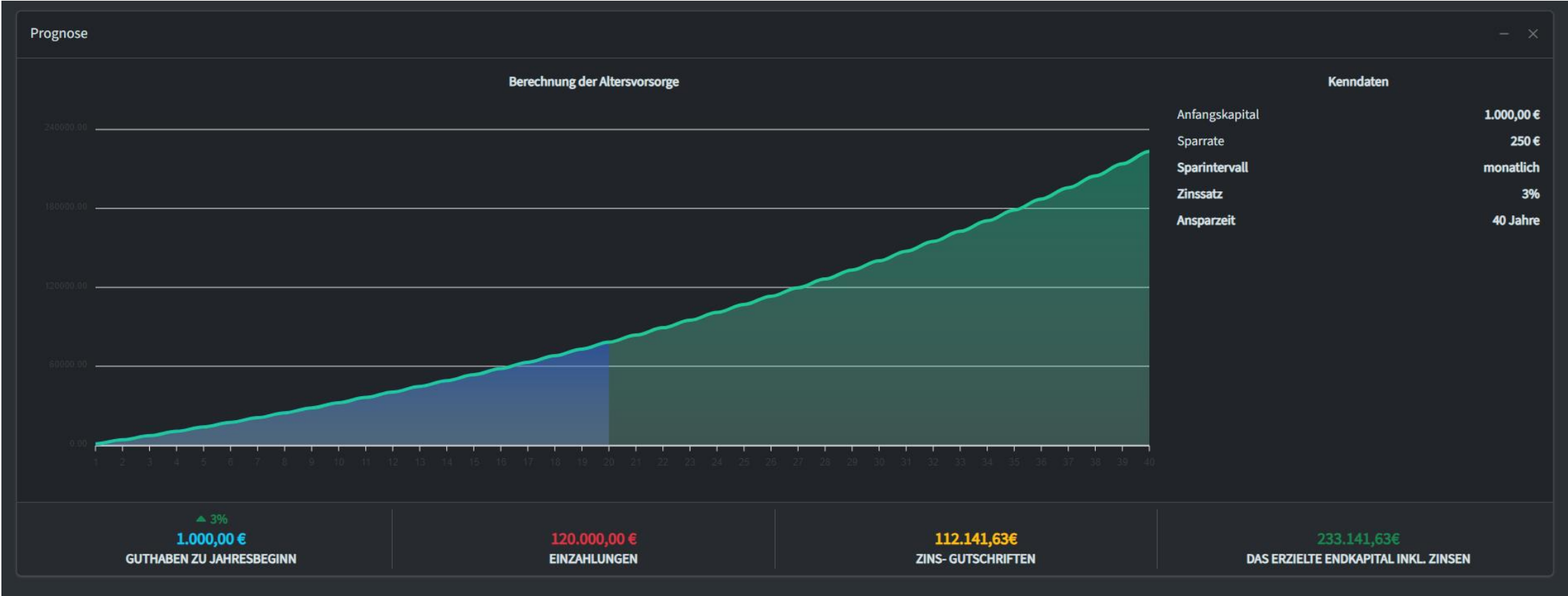
Chart.js

- Open-Source-JavaScript-Bibliothek
 - Datenvisualisierung
 - unterstützt 8 Diagrammtypen: Balken, Linien, Flächen, usw.
-

Rapid Prototyp



Rapid Prototyp



Ausblick Audit 4

- Weitere Ausarbeitung unseres Prototyps: Erweiterung um Registrierung, Anmeldung, manuelle Dateneingabe, Berechnung der Prognosen unter Berücksichtigung verschiedenster Anlagestrategien mithilfe unseres Tech Stacks
 - Codeinspektion unseres finalen Prototyps
 - Fazit und kritische Einordnung unseres Projekts anhand unserer ursprünglichen Zielsetzung
 - Abschließendes Poster für die finale Präsentation
-