

Sanjaya Shrestha

Martiniplatte. 1

93109 Wiesent

Tel: +49 1512 0215172

Mail: [info@sanjayashrestha.com](mailto:info@sanjayashrestha.com)

Homepage: [Sanjaya Shrestha](#)

Github: <https://github.com/pasa2002>

Demo: [SimpleCrm \(sanjayashrestha.com\)](#)

## **Dokumentation der SimpleCRM**

# Inhaltsverzeichnis

1. Einleitung
2. Problemstellung
3. Tech-Stack
4. Dateien- und Ordnerstruktur
5. Projektstart
6. Klassendiagramm

# 1. Einleitung

- SimpleCRM ist Ein Innovative Management-Tool für Basketball-Teams. Es hilft, um Basketball Teams und deren Manager in dieser anspruchsvollen Aufgabe zu unterstützen. SimpleCRM bietet eine maßgeschneiderte Lösung, die auf die spezifischen Bedürfnisse und Herausforderungen im Basketball zugeschnitten ist

## **Zielgruppe:**

- SimpleCRM richtet sich an Basketball-Teams, Trainer, die eine effiziente und übersichtliche Plattform zur Verwaltung ihrer Spieler und Investoren suchen. Das Tool ist ideal für Profi- und Amateurtteams, die ihre administrativen Aufgaben vereinfachen und gleichzeitig tiefergehende Einblicke in die Leistung und Dynamik ihrer Spieler gewinnen möchten.

# 2. Problemstellung

- Komplexe Spielerdatenverwaltung
- Finanzmanagement und Transparenz
- Investorenbeziehungen
- Informationen und Kommunikation

# 3. Tech-Stack

- Für die Entwicklung von SimpleCRM habe ich einen vielseitigen Tech-Stack gewählt, der Angular, HTML, SCSS und TypeScript umfasst, um eine reaktionsfähige und benutzerfreundliche Oberfläche zu schaffen. Für die visuelle Darstellung von Daten und Strukturen wurde Chart.js integriert, während die balldontlie API für Mannschaftsdaten genutzt wurde. Firebase dient als Backend-Lösung, die effiziente Datenverwaltung und Skalierbarkeit ermöglicht. Zusätzlich wurde Angular Material eingesetzt, um ein konsistentes und ansprechendes Design zu gewährleisten, das die Benutzerfreundlichkeit und Ästhetik der Anwendung weiter verbessert.

## 4. Dateien- und Ordnerstruktur

- Die SimpleCRM-Anwendung ist in eine modulare Struktur gegliedert, die eine klare Trennung von Komponenten, Diensten und Modellen ermöglicht. Im Folgenden wird die Struktur und Funktion der wichtigsten Dateien und Verzeichnisse erläutert:
  - **src/app**: Dieses Verzeichnis enthält alle Komponenten, Dienste und Modelle, die für die Anwendung notwendig sind.
    - **components**: Jede Komponente, wie **player-detail** oder **player-earning-chart**, repräsentiert einen spezifischen Teil der Benutzeroberfläche und ist in einem eigenen Unterordner mit einer HTML-Vorlage, SCSS-Stylesheets und einer TypeScript-Datei organisiert.
    - **services**: Die Services, definiert in diesem Ordner, enthalten die Geschäftslogik und die Datenverwaltung, die die Komponenten verwenden, um mit externen APIs wie der balldontlie API und Firebase zu interagieren.
    - **models**: In diesem Ordner werden Klassen wie **player.class.ts** definiert, die die Datenstruktur für die Spielerobjekte festlegen.
  - **src/environments**: Hier werden die Umgebungsvariablen gespeichert, die für die Entwicklung und Produktion der Anwendung unterschiedlich sein können.
  - **assets**: Dieser Ordner beinhaltet alle statischen Dateien wie Bilder und globale Styles.
  - **styles.scss**: Die globale Stylesheet-Datei definiert Design-Elemente, die in der gesamten Anwendung verwendet werden.
  - **angular.json**: Die Konfigurationsdatei für Angular, die Build- und Entwicklungsprozesse definiert.
  - **firebase.json**: Eine Konfigurationsdatei, die die Firebase-Hosting-Optionen spezifiziert.
  - **node\_modules**: Dieses Verzeichnis enthält alle Pakete, die über den Node Package Manager (npm) installiert wurden.
  - **.gitignore**: Eine Konfigurationsdatei für git, die Dateien und Verzeichnisse auflistet, die nicht in das Repository übernommen werden sollen.

## 5. Projektstart

Das Projekt lässt benötigt keinen Server oder zusätzliche Software, um gestartet zu werden. Sie müssen lediglich nur den [Link](#) folgen und sich anmelden. Sie sind dann startklar.



### Login

Email address

Password

Login

Guest Login

New to Simple CRM [Sign Up!](#)

## 6. Klassendiagramm



