**Project Files**

1. Source Code Files

* app.component.ts: Hauptkomponente der Anwendung. Diese Komponente stellt das Hauptlayout der App dar und bindet andere Komponenten ein.
* app.component.html: HTML-Template für die Hauptkomponente, das den visuellen Aufbau der Hauptseite definiert.
* app.component.css: Stile für die Hauptkomponente. Definiert das Aussehen der Hauptseite.
* app.module.ts: Hauptmodul, das alle Komponenten, Module und Dienste registriert, um die App zu starten.
* main.ts: Einstiegspunkt der Anwendung, der die App initialisiert und Angular bootstrapped (startet).
* environment.ts: Konfigurationsdatei für Umgebungsvariablen, die spezifische Einstellungen für die Entwicklungsumgebung enthält (z. B. API-URLs, Debugging-Optionen).

1. Routing Files

* app-routing.module.ts: Definiert die Routen der App, also welche Komponente bei welcher URL geladen wird. Dies ermöglicht die Navigation zwischen verschiedenen Seiten.

1. Component Files

* budget.component.ts/html/css: Budget-Komponente, die für die Darstellung und Verwaltung von Budgetinformationen zuständig ist.
* budget-dialog.component.ts/html/css: Dialog-Komponente für die Erstellung oder Bearbeitung von Budget-Einträgen.
* categories.component.ts/html/css: Diese Komponente zeigt und verwaltet verschiedene Kategorien, die in der App verwendet werden.
* dashboard.component.ts/html/css: Dashboard-Komponente, die eine Übersicht über wichtige Daten oder Kennzahlen der App bietet.
* main-site.component.ts/html/css: Hauptseite der Anwendung, die als Container für alle anderen Seiten fungiert.
* reports.component.ts/html/css: Komponente zur Anzeige und Erstellung von Berichten basierend auf Daten aus der App.
* settings.component.ts/html/css: Komponente für die Verwaltung der Benutzereinstellungen und Konfigurationen.
* sign-in.component.ts/html/css: Login-Seite, über die sich Benutzer in die App einloggen können.
* sign-up.component.ts/html/css: Registrierungsseite, über die sich neue Benutzer anmelden können.
* transaction-dialog.component.ts/html/css: Dialog-Komponente zum Hinzufügen oder Bearbeiten von Transaktionen.
* transactions.component.ts/html/css: Zeigt alle Transaktionen in einer Liste oder Tabelle an.
* transaction-service.component.ts/html/css: Service-Komponente zur Verwaltung von Transaktionen und deren Interaktion mit der Datenbank.
* year-dialog.component.ts/html/css: Dialog zur Auswahl oder Anzeige von Jahresübersichten, z. B. zur Budgetplanung.

1. Models

* budget-entry.model.ts : Modellklasse, die die Struktur eines Budget-Eintrags definiert, z. B. mit Feldern wie Name, Betrag und Kategorie.
* transaction-entry.model.ts: Modellklasse, die die Struktur eines Transaktions-Eintrags definiert, z. B. mit Feldern wie Datum, Betrag und Beschreibung.

1. Service Files

* auth.guard.ts: Authentifizierungsschutz, der sicherstellt, dass nur angemeldete Benutzer auf bestimmte Routen zugreifen können.
* auth-service.service.ts: Dienst zur Verwaltung der Benutzeranmeldung, Registrierung und Authentifizierung.
* budget.service.ts: Dienst zur Verwaltung der Budget-Daten, der die Kommunikation mit der Datenbank übernimmt.
* categories.service.ts: Dienst zur Verwaltung von Kategorien, der Datenbankabfragen und andere Logik für Kategorien bereitstellt.
* firestore.service.ts: Dienst, der die Interaktion mit der Firestore-Datenbank von Firebase ermöglicht.
* report.service.ts: Dienst zur Erstellung und Verwaltung von Berichten basierend auf den App-Daten.
* settings.service.ts: Dienst zur Verwaltung und Speicherung von Benutzereinstellungen und Konfigurationen.

1. Assets

* i18n/de.json: Übersetzungsdatei für die deutsche Sprache.
* i18n/en.json: Übersetzungsdatei für die englische Sprache.
* images/berichte.png: Bild, das möglicherweise für Berichte oder Dashboard verwendet wird.
* images/budget.png: Bild für die Budget-Seite oder -Funktionalität.
* images/dashboard.png: Bild für das Dashboard.
* images/doc\_icon.svg: Symbol für Dokumente, wahrscheinlich in Berichten oder Exportfunktionen.
* images/einstellungen.png: Bild für die Einstellungen.
* images/jpeg\_icon.svg: Symbol für JPEG-Dateien, möglicherweise in einer Exportfunktion.
* images/kategorien.png: Bild für die Kategorien-Seite.
* images/logo.jpg: Das Hauptlogo der App.
* images/pdf\_icon.svg: Symbol für PDF-Dateien, wahrscheinlich in einer Exportfunktion.
* images/plus.png: Symbol für "Hinzufügen" oder "Neue Einträge".
* images/png\_icon.svg: Symbol für PNG-Dateien, wahrscheinlich in einer Exportfunktion.
* images/transaktionen.png: Bild für die Transaktionen-Seite.
* manifest.webmanifest: Konfigurationsdatei für Progressive Web Apps (PWA), die Meta-Informationen wie Icons und Start-URLs enthält.

1. Environment Files

* src/environments/environment.ts: Entwicklungsumgebungskonfiguration, die lokale Datenquellen, APIs und andere Umgebungsvariablen enthält.

1. Configuration Files

* angular.json: Konfigurationsdatei für das Angular CLI. Definiert Build-Prozesse, Projektpfade und andere Einstellungen.
* package.json: Definiert die Projektabhängigkeiten, Skripte und Metadaten (wie den Projektnamen und die Version).
* tsconfig.json: Konfigurationsdatei für TypeScript, die Compiler-Optionen und Dateieinstellungen definiert.

1. Firebase Configuration

* firebase.json: Konfigurationsdatei für Firebase Hosting und andere Firebase-Dienste wie Firestore, Authentication und Storage.
* .firebaserc: Firebase-Projektkonfiguration, die verschiedene Firebase-Projekte und Umgebungen referenziert.

1. Service Worker

* ngsw-config.json : Konfigurationsdatei für den Angular Service Worker, die bestimmt, wie die App offline funktioniert und welche Dateien zwischengespeichert werden sollen.

1. Dokumentation

* README.md: Eine Datei mit Informationen zum Projekt, wie Installationsanweisungen, Nutzungshinweisen und Projektzielen.
* LICENSE: Eine Datei, die die Lizenz des Projekts beschreibt und die Nutzungsbedingungen festlegt.

1. Tests

* app.component.spec.ts: Testdatei für die app.component.ts, die sicherstellt, dass die Hauptkomponente korrekt funktioniert.
* budget.component.spec.ts: Testdatei für die budget.component.ts, um deren Funktionen zu überprüfen.
* budget-dialog.component.spec.ts: Testdatei für den Dialog zur Budgetbearbeitung.
* categories.component.spec.ts: Testdatei für die Kategorien-Komponente.
* dashboard.component.spec.ts: Testdatei für die Dashboard-Komponente.
* main-site.component.spec.ts: Testdatei für die Hauptseite.
* reports.component.spec.ts: Testdatei für die Berichts-Komponente.
* settings.component.spec.ts: Testdatei für die Einstellungen.
* sign-in.component.spec.ts: Testdatei für die Login-Komponente.
* sign-up.component.spec.ts: Testdatei für die Registrierungs-Komponente.
* transaction-dialog.component.spec.ts: Testdatei für den Transaktionsdialog.
* transactions.component.spec.ts: Testdatei für die Transaktionen-Komponente.
* transaction-service.component.spec.ts: Testdatei für den Transaktions-Service.
* year-dialog.component.spec.ts: Testdatei für den Jahresauswahl-Dialog.
* auth.guard.spec.ts: Testdatei für die Authentifizierungslogik.
* auth-service.service.spec.ts: Testdatei für den Authentifizierungsdienst.
* firestore.service.spec.ts: Testdatei für den Firestore-Dienst.

1. Build and Deployment

* dist/: Ordner, in dem das fertige Build nach ng build abgelegt wird.
* firebase.json: Konfigurationsdatei für Firebase Hosting und Deployment-Prozesse.