Budget Tracker - Software Requirements Specification (SRS)

1. Introduction

The Budget Tracker is a web application that allows users to track their income, expenses, and savings. The goal is to provide a basic and functional budget management tool for personal use.

2. Functional Requirements

2.1 User Registration and Login

Users can register for an account by providing a unique username and password.
Users can register for an account by Google signup/signin
Users can log in using their registered credentials.

2.2 Dashboard Overview

Upon logging in, users are directed to their dashboard displaying their current balance.
The dashboard will show a summary of total income, total expenses, and savings.

2.3 Income Management

Users can add new sources of income, type, specifying the income name, amount, and date received.
Users can view a list of their income transactions with options to edit or delete them.
Note: Create a central ledger to maintain income and expense records.

2.4 Expense Management

Users can add new expenses, specifying the expense name, type, amount, and date of the expense.
Users can view a list of their expense transactions with options to edit or delete them.
Note: Create a central ledger to maintain income and expense records.

2.5 Savings Management

Users can set savings goals, specifying the target amount and a deadline.
Savings should be calculated based on income and expense in a set period.
Users can view their savings goals and track their progress.

2.6 Budget Visualization

The application should display basic visualizations, such as a pie chart or bar graph, to represent the distribution of income, expenses, and savings.
The application should display reports with filters and sorting.

2.7 Budget Summary

The application should provide a monthly summary of income, expenses, and savings, allowing users to track their budget progress over time.

3. Non-functional Requirements

3.1 User Interface (UI)

The UI should be clean, intuitive, and responsive.
The design should be kept simple to expedite development.

3.2 Technology Stack

Frontend: React.js, Antd, HTML, CSS
Backend: NestJS, NodeJS, TypeScript, TypeORM
Database: Google Cloud Firestore, PostgreSQL