**Real Estate Investment Assistant Documentation**

**Overview**

The Real Estate Investment Assistant is a Streamlit application designed to analyze and visualize the financial performance of real estate investments. Users input hypothetical investment scenarios, which the app translates into a series of financial transactions. These transactions are then used to project cash flows and evaluate the investment's financial viability.

**Application Workflow**

User Input:

The application starts with the user inputting a real estate investment scenario via the Streamlit interface.

LLM Interpretation:

The language model interprets the user's narrative into a structured format, specifically a list of dictionaries representing transactions, each with action and year keys.

Dataframe Construction:

The app constructs a DataFrame, organizing transactions by year and associating each with a specific financial action.

Financial Analysis:

The application calculates cash flows and other financial metrics for each action and year, assessing the overall performance of the investment.

Visualization:

Results, including cash flow analysis and investment performance metrics, are visualized in the Streamlit dashboard, providing users with interactive and comprehensive insights.

**Key Modules and Their Functions**

**streamlit\_app.py**

The main driver of the Streamlit application, handling user interactions and coordinating the flow of data through the app.

**functions\_cashflow\_dataset.py**

Contains functions for generating and manipulating datasets related to cash flows in real estate transactions.

**functions\_cashflow\_for\_transaction\_action.py**

Provides functions that calculate cash flows based on different transaction actions within the investment lifecycle.

**functions\_cashflows\_calculation.py**

Includes core financial calculation functions, such as mortgage payments, interest calculations, and cash flow simulations.

**investment\_analysis.py**

Core module for analyzing the investment, calculating return on investment, net present value, and other key financial indicators.

**parameters.py**

Defines initial investment parameters and configurations used across the application for financial calculations.

**prep\_functions.py**

Prepares and preprocesses the data, converting user inputs into a format suitable for financial analysis.

**llm\_response.py**

Manages the interaction with the language model, processing user inputs into structured transaction data for analysis.

**llm\_settings.py**

Configures settings for the language model and other environmental variables necessary for the app's operation.