**Risk Score Analysis: Code Documentation**

**Table of Contents**

1. Introduction
2. Code Overview
3. Design and Functionality
4. Execution Flow
5. Enhancements and Future Scope

**1. Introduction**

This document provides a detailed overview of a Python script designed for analyzing financial transactions and determining their risk scores. The system leverages OpenAI's GPT-4o-mini model to assess transactions and assigns risk scores based on various factors. The goal is to identify suspicious or risky transactions effectively, providing reasoning and sub-scores for transparency.

**2. Code Overview**

The script uses OpenAI's API to interact with GPT-4o-mini. A transaction is analyzed based on a predefined system message instructing the AI to evaluate the risk, consider external sources, and provide a structured output (preferably in tabular format).

**3. Design and Functionality**

**Design Considerations:**

* **Scalability**: The system should be able to analyse multiple transactions.
* **Transparency**: It provides sub-scores and clear reasoning for the assigned risk score.
* **Extensibility**: Can be modified to include more external sources or factors.

**Functional Aspects:**

* Accepts transaction details as input.
* Utilizes an AI model to analyse transaction risk.
* Fetches additional information if required.
* Assigns a structured risk score.
* Provides reasoning for the given score.

**4. Execution Flow**

1. The script initializes the OpenAI client.
2. A system message defines the AI's role and instructions.
3. The user inputs transaction details.
4. The AI processes the request and generates a risk assessment.
5. The output is formatted and displayed for easy interpretation.

**5. Enhancements and Future Scope**

**Possible Improvements:**

* **Integration with External Databases**: Checking watchlists for suspicious entities.
* **Enhanced Scoring Model**: Incorporating machine learning to refine risk scores.
* **Multi-Transaction Analysis**: Batch processing for efficiency.