

**Navigation Database Reverse Engineering Project Documentation**

**Market Data Viewer Technical Documentation**

# Navigation Database Reverse Engineering Project Documentation

## Background

The script is a Market Data Viewer application designed to fetch and display real-time financial data for cryptocurrencies and stocks. It uses APIs to retrieve market prices, stores them in an SQLite database, and presents the data through a GUI.

# Navigation Database Reverse Engineering Project Documentation

## Requirements

Must fetch real-time data for specified cryptocurrencies and stocks.

Should store the fetched data in a local SQLite database.

Could provide a real-time updating GUI to display the fetched data.

Would benefit from logging mechanisms to track and record operational events.

# Navigation Database Reverse Engineering Project Documentation

## Method

The application is structured into several key components:

- DataFetcher class: Manages data retrieval and database updates.
- SQLite Database: Stores market data for cryptocurrencies and stocks.
- UI with tkinter: Displays the fetched data and controls the data fetching process.

# Navigation Database Reverse Engineering Project Documentation

## Implementation

1. Database Initialization: Setup SQLite database to store market data.
2. Data Fetching: Continuously fetch market data for predefined cryptocurrencies and stocks.
3. Data Display: Update the UI at regular intervals to show the latest market data.

# Navigation Database Reverse Engineering Project Documentation

## Milestones

1. Database Setup: Successful creation and initialization of market data tables.
2. Data Fetching Operational: Real-time data fetching and storage mechanisms in place.
3. UI Functionality: Real-time updating GUI displays the market data accurately.

# Navigation Database Reverse Engineering Project Documentation

## Gathering Results

Evaluate the application based on its ability to fetch, store, and display real-time market data efficiently. Assess the robustness and reliability of the data fetching and updating mechanisms. Verify the user interface for clarity, usability, and update frequency.