Project Documentation: Cryptocurrency Dashboard
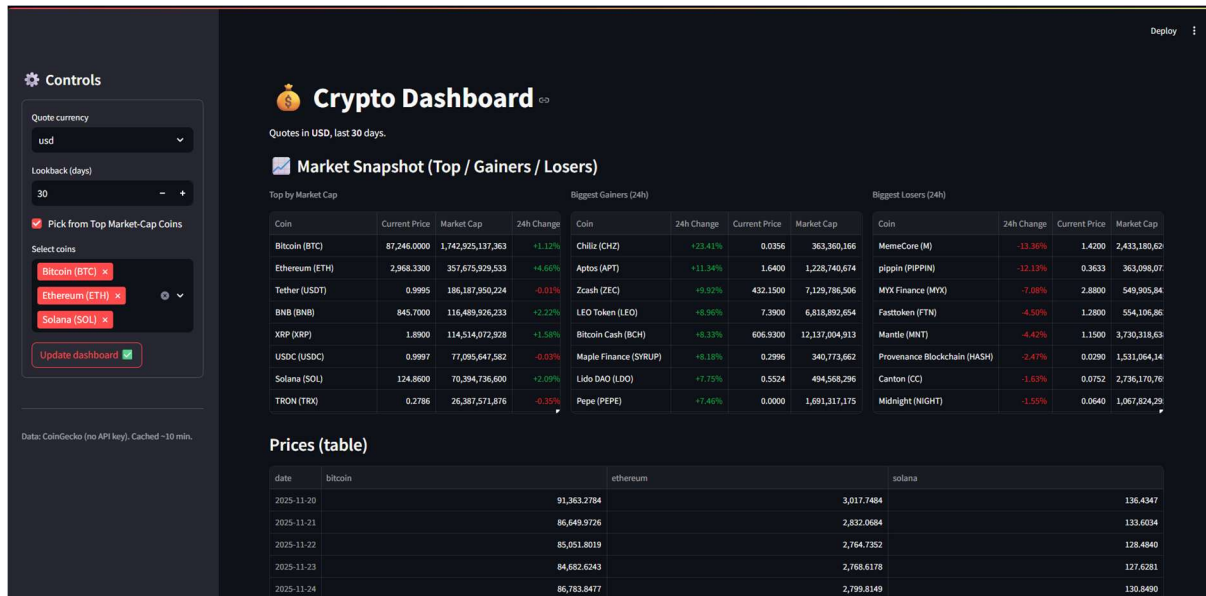
By Gonzalo Constans

Figure 1: Project result

1. Introduction

The Cryptocurrency Dashboard project is an interactive web-based application developed in Python using the Streamlit framework. The purpose of the project is to demonstrate how real-world data can be retrieved from a public API, processed, analyzed, and visualized. The dashboard connects directly to the CoinGecko public API to provide live information about the cryptocurrency market, including price trends, volatility, correlations between assets, and educational "recommendations" based on technical indicators.

We combined several programming concepts: data acquisition from external sources, data cleaning and analysis using pandas, visualization using matplotlib, and interactive web deployment using Streamlit.

2. Project Aim and Objectives

1. Demonstrate how to connect Python to an external RESTful API and retrieve structured JSON data.

2. Implement data processing techniques to organize and summarize large quantities of financial time-series information.

3. Create a responsive interface that allows users to filter, visualize, and interpret data dynamically.

4. Introduce simple analytical metrics (such as volatility, correlation, and technical indicators) to provide insight into market behavior.

3. System Architecture and Development Process

We created the dashboard entirely with python. It consists of three parts:

3.1 Data Acquisition

The application retrieves data using the requests library from CoinGecko endpoints. I used two main endpoints:

- The /coins/markets endpoint provides an overview of the top cryptocurrencies by market capitalization, their prices, market values, and 24-hour changes.

- The /coins/{id}/market_chart endpoint provides historical price data for selected coins over a user-defined time range (e.g., 30, 90, or 180 days).

The API responses are in JSON format and are parsed into pandas DataFrames for further analysis. To respect rate limits and avoid HTTP 429 ("Too Many Requests") errors, the program uses a delay between consecutive calls

3.2 Data Analysis

Once the raw data is collected, several computations are performed:

- **Volatility (Standard Deviation of Daily Returns):** measures the dispersion of returns, indicating how risky or stable each cryptocurrency has been over the selected time window.

- **Correlation Matrix:** shows how closely different coins move together, which helps identify diversification opportunities.

- **Cumulative Returns:** computed over the selected period to summarize overall performance.

- **Recommendation Signals:** generated using a set of simple technical indicators, including moving averages, seven-day momentum, and the Relative Strength Index (RSI). These signals classify each coin as "Buy," "Hold," or "Sell"

3.3 Visualization and User Interface

We built the visualization with Streamlit. The user interface is organized into a sidebar and several main sections:

- The **sidebar** allows users to choose the quote currency, define the lookback period, and select which coins to analyze. To prevent unnecessary API calls and rate-limit issues, the dashboard uses an "Update Dashboard" button so that the application only reloads data when explicitly requested.
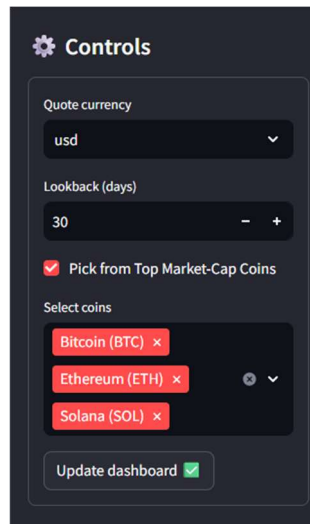
Figure 2: Sidebar controls

- The **market snapshot section** displays three tables side by side: the top coins by market capitalization, the day's biggest gainers, and the biggest losers. Each table shows the coin name, current price, market capitalization, and 24-hour percentage change. The percentage change column is color-coded (green for positive changes, red for negative)

💰 Crypto Dashboard

Quotes in **USD**, last **30** days.

📈 Market Snapshot (Top / Gainers / Losers)

Top by Market Cap				Biggest Gainers (24h)				Biggest Losers (24h)			
Coin	Current Price	Market Cap	24h Change	Coin	24h Change	Current Price	Market Cap	Coin	24h Change	Current Price	Market Cap
Bitcoin (BTC)	87,246.0000	1,742,925,137,363	+1.12%	Chiliz (CHZ)	+23.41%	0.0356	363,360,166	MemeCore (M)	-13.36%	1.4200	2,433,180,62
Ethereum (ETH)	2,968.3300	357,675,929,533	+4.66%	Aptos (APT)	+11.34%	1.6400	1,228,740,674	pippin (PIPPIN)	-12.13%	0.3633	363,098,07
Tether (USDT)	0.9995	186,187,950,224	-0.01%	Zcash (ZEC)	+9.92%	432.1500	7,129,786,506	MYX Finance (MYX)	-7.08%	2.8800	549,905,84
BNB (BNB)	845.7000	116,489,926,233	+2.22%	LEO Token (LEO)	+8.96%	7.3900	6,818,892,654	Fasttoken (FTN)	-4.50%	1.2800	554,106,86
XRP (XRP)	1.8900	114,514,072,928	+1.58%	Bitcoin Cash (BCH)	+8.33%	606.9300	12,137,004,913	Mantle (MNT)	-4.42%	1.1500	3,730,318,63
USDC (USDC)	0.9997	77,095,647,582	-0.03%	Maple Finance (SYRUP)	+8.18%	0.2996	340,773,662	Provenance Blockchain (HASH)	-2.47%	0.0290	1,531,064,14
Solana (SOL)	124.8600	70,394,736,600	+2.09%	Lido DAO (LDO)	+7.75%	0.5524	494,568,296	Canton (CC)	-1.63%	0.0752	2,736,170,76
TRON (TRX)	0.2786	26,387,571,876	-0.35%	Pepe (PEPE)	+7.46%	0.0000	1,691,317,175	Midnight (NIGHT)	-1.55%	0.0640	1,067,824,29

Figure 3: Market snapshot with top market cap, gainers and losers

- The **price chart section**. Here I provide a table with the historical prices of the selected coins.

Prices (table)

date	bitcoin	ethereum	solana
2025-11-20	91,363.2784	3,017.7484	136.4347
2025-11-21	86,649.9726	2,832.0684	133.6034
2025-11-22	85,051.8019	2,764.7352	128.4840
2025-11-23	84,682.6243	2,768.6178	127.6281
2025-11-24	86,783.8477	2,799.8149	130.8490
2025-11-25	88,229.3574	2,953.3276	138.3677
2025-11-26	87,310.3316	2,955.0465	139.0062
2025-11-27	90,474.2268	3,027.4772	142.9225
2025-11-28	91,279.0606	3,015.6118	140.8299
2025-11-29	90,950.3770	3,033.1356	137.4683

Figure 4: Table of historical prices

- The **volatility and correlation section** presents two charts side by side: a bar chart ranking the coins by their 30-day volatility, and a heatmap showing the pairwise correlations of daily returns. These helps understand both individual risk levels and inter-coin relationships.

Figure 5: Volatility and Correlation graphs in the app

- Finally, the **recommendation section** summarizes each selected coin's technical indicators and displays an educational "Buy," "Hold," or "Sell" label, along with supporting metrics such as short-term and long-term moving averages, RSI, and recent performance.

Coin	Recommendation	Score	Last Price	SMA 7	SMA 21	SMA 50	RSI 14	7d Return
bitcoin	Sell	-2	87,247.6661	88,049.1264	89,927.7958	None	30.7	-5.67%
ethereum	Sell	-2	2,971.3902	2,999.1399	3,060.0104	None	36.3	-8.18%
solana	Sell	-2	124.8636	128.5316	133.2679	None	34.7	-8.48%

Figure 7: Recommendation UI

4. Features and Functionality

The dashboard includes several key features that make it interactive and educational:

1. **Dynamic Coin Selection:** users can either manually input coin IDs or select from the top market-cap coins retrieved automatically from CoinGecko.

2. **Customizable Time Frames:** the lookback period can be adjusted between 7 and 365 days

3. **Market Snapshot Tables:** provide an immediate overview of the market's current state.

4. **Technical Indicator-based Recommendations:** illustrate how signals can be generated from time-series data

5. **Downloadable Data:** users can export the raw prices, returns, and summary statistics as CSV files for further analysis.

6. **Caching and Rate-Limit Protection:** We used Streamlit's caching mechanism (@st.cache_data) and built-in pauses to ensure the app runs efficiently without overloading the API.

5. Development Process

The project was developed incrementally over several stages:

1. **Initial Setup:** establishing API connectivity and verifying data retrieval using the requests library.

2. **Data Processing:** cleaning and structuring the JSON responses into consistent DataFrames.

3. **Visualization:** prototyping charts with matplotlib and integrating them into a Streamlit layout.

4. **Interface Enhancements:** adding sidebar controls, update buttons, and formatted tables.

5. **Analytics Expansion:** introducing volatility, correlation, and simple recommendation logic.

6. **User Experience Refinements:** color-coded tables, side-by-side charts, improved layout, and caching for performance.

6. Challenges and Solutions

The main challenge during development was the CoinGecko API's strict rate limit, which can trigger HTTP 429 errors if too many requests are made in quick succession. This was resolved by implementing a pause between API calls and a retry mechanism with backoff.

Another challenge was ensuring that charts and tables remained readable despite the wide variation in cryptocurrency prices and volatilities. To solve it we used log scaling

Finally, because Streamlit reruns the entire script upon any interface interaction, we introduced an "Update Dashboard" button to prevent redundant API calls when adjusting parameters like lookback days.

7. Conclusion

With this project we showed how Python can be used to build a full-stack application that integrates real-time data, analysis, and visualization in a browser-based format.

From a learning perspective, it shows how different technical layers(data access, computation, and presentation)can be combined. The application can easily be extended in future iterations, for example by adding more advanced technical indicators, historical backtesting, or alternative data sources such as social media sentiment.

Overall, the project has provided a practical and engaging framework for understanding how computer science concepts translate into real-world applications.