

**A
REPORT
on
“ Live Crypto Analyzer ”**

Submitted to :

Department Of Computer Science & Engineering

Partial fulfilment of the degree of

Bachelors of Technology in Computer Science Engineering

Submitted by :

Yeshwant Waware

Vinay Jawade

Chetan Patil

Under the Guidance of

Prof. Priyanka Choudhary



NAAC Accredited with A+ Grade

ISO9001:2015 Certified

Vidarbha Bahu Uddeshiya Shikshan Sanstha's

Tulsiramji Gaikwad Patil College of Engineering & Technology, Nagpur-

441108

(An Autonomous Institute affiliated to RTMNU, Nagpur)

Session 2024-25

CERTIFICATE

This is to certify that project work described in this report entitled, “**Live Crypto Analyzer**” was carried out by **Yeshwant Waware , Chetan Patil, Vinay Jawade**, in Tulsiramji Gaikwad-Patil College of Engineering & Technology, Nagpur under my supervision and guidance in partial fulfillment of the requirement for the degree of **Computer Science & Engineering** of Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur.

This work is the own work of the candidate, completed in all respect and is of sufficiently high standard to warrant its submission to the said degree. The assistance and resources used for this work are duly acknowledged.

Under the Guidance

Project Cordinator

Head of the Department
(CSE)

Date : / / 2025



NAAC Accredited

Vidarbha Bahu-uddeshiya Shikshan Sanstha's

TULSIRAMJI GAIKWAD-PATIL
College of Engineering & Technology

— An Autonomous Institution —



DECLARATION

I hereby declare that this project titled “**Live Crypto Analyzer**” is abona fide and authentic record of the work done by me under supervision of **Prof. Priyanka Choudhary** during academic session 2024-25

the work presented here is not duplicated from any other source and also not submitted earlier for any other degree/diploma of any university. I understand that any such duplication is liable to be punished in accordance with the university rules. The source material, data used in this research have been duly acknowledged.

Date :

Group Member :

Place :

Yeshwant Waware

Vinay Jawade

Chetan Patil



NAAC Accredited

Vidarbha Bahu-uddeshiya Shikshan Sanstha's

TULSIRAMJI GAIKWAD-PATIL
College of Engineering & Technology

— An Autonomous Institution —



ACKNOWLEDGEMENT

With profound feeling of immense gratitude and affection, I would like to thank my guide **Prof. Priyanka Choudhary** (CSE) Department, for his continuous support, motivation, enthusiasm and guidance. His encouragement, supervision with constructive criticism and confidence enabled me to complete this project.

I also wish to extend my reverences to **Prof. Lowlesh Yadav** Computer Science & Engineering (HOD) for providing necessary facilities to complete my project. I am also thankful to all the faculty members and all non-teaching staff of the department & college for their cooperation throughout the project work. I also put forth my/our deepest sense of gratitude towards the Principal, TGPCET for constant motivation and providing necessary infrastructure.

Date :

GROUP MEMBER

Place :

Yeshwant Waware

Vinay Jawade

Chetan Patil

ABSTRACT

The rapid growth and volatility of the cryptocurrency market have created a pressing need for tools that can provide real-time insights into market movements. Traditional dashboards often fail to deliver live updates or come with significant latency, limiting their usefulness for active traders. This project addresses that challenge by developing a real-time cryptocurrency market analysis dashboard using Python and the Binance WebSocket API.

The dashboard provides live updates on market metrics such as current price, 24-hour percentage change, trading volume, and high/low prices for all listed trading pairs on Binance. By utilizing WebSocket technology, the system ensures continuous data streaming without the need for repeated HTTP requests, significantly improving performance and reducing latency. The backend is implemented using Python's asynchronous programming libraries, `asyncio` and `websockets`, to handle data efficiently in real time.

Incoming JSON data is structured and processed using the `pandas` library, enabling the extraction of meaningful statistics such as the top gainers, top losers, and most traded coins by volume. Additionally, specific data for the BTC/USDT trading pair is extracted to generate an interactive candlestick chart, offering deeper insight into price movement trends over time.

The front-end is built using Streamlit, a Python-based web app framework that allows rapid development of interactive dashboards. For data visualization, Plotly is used to render dynamic bar charts and candlestick graphs. The dashboard is divided into two main sections: a "Live Analysis" panel displaying the charts, and an "About" section providing project information. To maintain smooth and non-blocking updates, an Update Manager uses `asyncio.sleep(1)` to refresh data every second. State management is handled using `st.session_state`, ensuring consistent chart behavior and preventing unnecessary flickering or duplication of chart keys.

Overall, the project demonstrates the successful integration of real-time data streaming, asynchronous processing, and modern data visualization to create a fast, informative, and user-friendly crypto dashboard. It serves as a practical tool for traders and also as a proof of concept for building real-time financial applications using open-source technologies.

Table of Contents

Introduction.....	1
Objective.....	2
Literature Review.....	4
Methodology.....	5
System Designed.....	7
Implimentation.....	9
Result.....	10
Conclusion.....	11
Future Scope.....	12
References.....	14

INTRODUCTION

In recent years, the cryptocurrency market has emerged as one of the most dynamic and fast-paced financial sectors in the world. Operating around the clock, it presents unique challenges and opportunities for investors and analysts alike. Due to its decentralized and volatile nature, the prices of cryptocurrencies can change dramatically within seconds. As a result, real-time data analysis becomes critical for making informed trading decisions and understanding market trends. Most existing crypto dashboards provide delayed or limited data updates, which can hinder traders from responding effectively to rapid market movements. To overcome this limitation, this project focuses on building a real-time crypto market analysis dashboard that delivers live data visualization using modern Python-based technologies and APIs.

The primary objective of this project is to design and implement a dashboard that streams live cryptocurrency data from the Binance WebSocket API and presents it in a clear, interactive, and insightful manner. The dashboard includes multiple features such as the top gainers and losers based on 24-hour price change, the highest-volume trading pairs, and a candlestick chart for detailed price movement of a selected pair (BTC/USDT). The backend of the system uses Python's `websockets` and `asyncio` libraries to receive real-time data asynchronously. Once the JSON-formatted data is received, it is parsed and transformed into structured pandas DataFrames, enabling efficient data manipulation and analysis.

The front-end is built using Streamlit, a lightweight and fast web app framework for Python. It serves as an interactive platform for users to view market updates in real time. Visualizations such as bar charts and candlestick charts are rendered using Plotly, offering an engaging and informative user experience. To maintain seamless updates and avoid UI flickering, the project uses `st.session_state` to manage component states and an asynchronous update mechanism that refreshes data every second without blocking the main thread. A simple sidebar allows users to navigate between sections, such as Live Market Analysis and an informational "About" section.

This project not only demonstrates the application of real-time data processing and visualization in a practical context but also highlights the growing importance of accessible and responsive financial tools in the digital economy. By combining real-time APIs, asynchronous programming, data analysis, and modern web technologies, the dashboard serves as a powerful tool for crypto traders, enthusiasts, and researchers looking to stay ahead in a fast-moving market.

OBJECTIVE

The primary aim of this project is to design and implement a **real-time cryptocurrency market analysis dashboard** that leverages live data streaming and visual analytics to assist users in tracking market performance efficiently. Given the dynamic and volatile nature of the cryptocurrency market, the system is intended to provide rapid insights that are both informative and visually engaging.

The **specific objectives** of this project are as follows:

◆ 1. Real-Time Data Integration

To integrate live cryptocurrency data from the **Binance WebSocket API**, enabling continuous, second-by-second updates on price movements, volume, and other key trading metrics.

◆ 2. Market Trend Analysis

To analyze the **Top 10 gainers and losers** in the market by computing 24-hour percentage change in prices, offering a quick snapshot of the best and worst performing cryptocurrencies.

◆ 3. High Volume Coin Monitoring

To track and display the **Top 10 cryptocurrencies by trading volume**, which helps users identify coins with the highest liquidity and trading interest during a given period.

◆ 4. Candlestick Chart Visualization

To implement a **real-time candlestick chart for BTC/USDT**, showing Bitcoin's high, low, open, and close prices to support technical analysis of market behavior.

◆ 5. Visual Dashboard Development

To build a **visually appealing and interactive dashboard** using **Streamlit** and **Plotly**, allowing users to interact with charts and observe live updates in a web-based environment.

◆ 6. Asynchronous Programming for Stability

To employ **asynchronous functions using Python's `asyncio` module** for handling live data streams efficiently and ensuring that chart updates do not interrupt the application's responsiveness.

🔗 7. Session Management for Smooth Updates

To implement a mechanism using **Streamlit's `session_state`** to manage chart keys and preserve update counts, ensuring stable and flicker-free chart rendering during live updates.

🔗 8. Educational and Informative Add-ons

To enrich the user experience by displaying **crypto facts and trivia**, enhancing user knowledge and engagement beyond raw data presentation.

🔗 9. Expandability and Future Integration

To design the system in a **modular and scalable** way, allowing for future features such as user alerts, portfolio tracking, predictive analytics, or multi-exchange data support.

This set of objectives ensures that the dashboard is not only functional and responsive, but also valuable from both a practical trading perspective and an educational standpoint. for adapting the technology for different

LITERATURE REVIEW

The cryptocurrency market, known for its volatility and 24/7 operation, has drawn considerable attention from researchers, developers, and investors alike. As digital assets continue to gain mainstream traction, the need for real-time data analytics and visualization tools has become increasingly important. Various studies and projects have explored the intersection of real-time data streaming, financial analysis, and user-friendly interfaces to aid decision-making in the crypto domain.

Chitlange and Tambade (2023) conducted a comprehensive survey on visual dashboards for real-time market analysis. Their work highlighted the value of live data streaming combined with dynamic visualizations to interpret financial trends efficiently. They emphasized the integration of frontend libraries like Chart.js and backend tools such as Flask, a model similar to the architecture used in this project, albeit with Python and Streamlit.

In a related project, Raj (2024) explored crypto data analysis using Python and Streamlit. The study demonstrated how Streamlit's rapid development capabilities and simplicity in handling user interaction make it suitable for building analytical dashboards. His project focused on plotting historical trends and analyzing token performances, laying a foundational approach for visualization in crypto analytics.

Another notable project, "Cryptonik" by Chauhan and Jain (2023), presented a live cryptocurrency tracker that allowed users to monitor real-time values of multiple tokens. The project used REST APIs for data collection and highlighted the importance of timely information in a volatile market. However, the lack of asynchronous data handling presented limitations in performance and scalability, which this Live Crypto Analyzer addresses through Python's `asyncio` and `websockets`.

More advanced studies like "CryptoVisualizer+" (Langer & Maier, 2023) integrated structured visual analytics for time-series crypto data. They introduced sophisticated charting and technical indicators, showcasing how powerful visual tools can support traders in making strategic decisions. Similarly, Roy and Paul (2024) proposed a machine learning framework for forecasting cryptocurrency prices, pushing the analytical boundary further from just visualization to intelligent prediction.

These works collectively establish the growing interest in real-time analytics and the need for accessible, interactive platforms. While many tools exist for data tracking and prediction, the majority either lack real-time WebSocket-based streaming or do not focus on user experience and dynamic updates. The Live Crypto Analyzer aims to fill this gap by offering a WebSocket-powered, fully interactive, and real-time dashboard using open-source technologies like Streamlit and Plotly.

This review of existing literature confirms the relevance and necessity of real-time crypto analytics tools and provides a solid foundation upon which the current project builds and innovates.

METHODOLOGY

The methodology for developing the Real-Time Cryptocurrency Dashboard is divided into several key phases, ranging from requirement gathering and tool selection to implementation and testing. The overall approach follows an **incremental development model** where each component is developed, tested, and integrated progressively to ensure functionality and stability.

3.1 Requirements Analysis

- Identified the need for a real-time crypto monitoring tool that can visualize top market trends.
- Determined the use of **Binance WebSocket API** as the data source due to its reliability and real-time capabilities.
- Selected **Streamlit** for front-end development to simplify deployment and enable quick UI prototyping.
- Chose **Plotly** for creating interactive, aesthetically appealing visualizations.

3.2 Data Collection using WebSockets

- Utilized **WebSocket connections** to subscribe to Binance's live ticker data streams.
- Parsed incoming JSON messages to extract essential fields such as:
 - Symbol
 - Last price
 - 24h price change percentage
 - 24h trading volume
 - High and low prices
- Converted the raw data into a **pandas DataFrame** for easier manipulation and analysis.

3.3 Data Processing and Filtering

- Computed 24-hour percentage price change to rank the **top gainers and losers**.
- Sorted and filtered data to identify the **top 10 cryptocurrencies by trading volume**.
- Extracted price values for **BTC/USDT** to create a real-time candlestick chart.

3.4 Visualization with Plotly

- Used **Plotly Express and Graph Objects** to build:
- Bar charts for top gainers, losers, and high-volume coins

- Candlestick charts for BTC/USDT
 - Applied color scales (green, red, blue) to represent different categories intuitively.
-

3.5 Dashboard Implementation with Streamlit

- Created a multi-page layout in **Streamlit**, separating live analysis and project overview.
 - Integrated **Plotly charts** into Streamlit using `plotly_chart()` with unique keys to support stable live updates.
 - Used `st.session_state` to track update counts and maintain consistency in dynamic rendering.
-

3.6 Real-Time Data Handling

- Implemented asynchronous logic using **Python's asyncio module** for:
 - Non-blocking WebSocket connections
 - Real-time updates every second
 - Ensured smooth UI performance even with continuous streaming and refreshing of charts.
-

3.7 Informational Add-ons

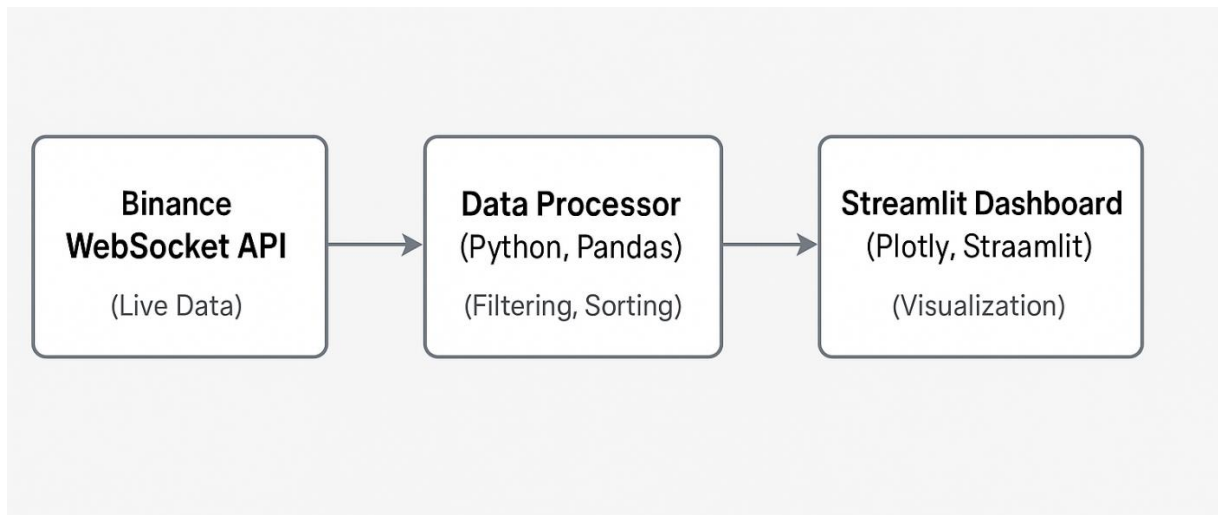
- Added an "**About**" page to display rotating **crypto facts** using the `random` module.
 - Enhanced user engagement by offering educational content alongside live charts.
-

3.8 Testing and Debugging

- Conducted extensive testing to validate:
 - WebSocket connection stability
 - Chart update smoothness
 - Correctness of data filtering and sorting logic
 - Resolved issues related to chart flickering and data inconsistencies during live updates.
 - minimal latency and maximum clarity for the end-user.
-

SYSTEM DESIGNED

The system is designed as a **real-time data visualization dashboard** that connects to the Binance exchange via WebSocket, processes incoming data, and displays it through interactive visualizations using a web-based interface. The architecture follows a **modular design**, separating data acquisition, processing, and presentation layers to ensure flexibility and scalability.



4.2 Component Descriptions

◆ 1. Binance WebSocket API (Data Source)

- Provides **real-time ticker data** for all listed trading pairs.
- Emits JSON-formatted messages that include:
- Current price
- 24h price change (%)
- Volume
- High/Low prices

◆ 2. Data Processor (Back-End Logic)

- **Receives data** asynchronously using Python's `websockets` and `asyncio` libraries.
- Transforms JSON into structured format using **pandas DataFrames**.
- Applies **business logic** to:
-

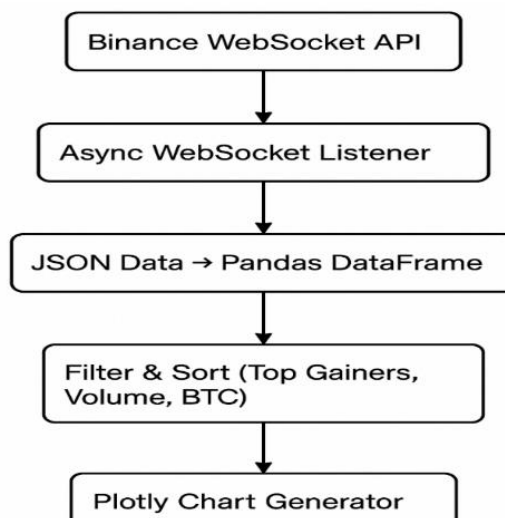
- Calculate top gainers/losers by 24h change
 - Sort coins by volume
 - Extract specific pair data for candlestick chart (BTC/USDT)
-

◆ 3. Streamlit Dashboard (Front-End)

- Serves as the **user interface** for visualizing data.
 - Renders dynamic charts using **Plotly**:
 - Bar charts for gainers, losers, and volume
 - Candlestick chart for BTC/USDT
 - Uses **st.session_state** to manage chart keys and ensure consistent updates.
 - Integrates a navigation sidebar with two main sections:
 - Live Analysis
 - About (Facts & Info)
-

◆ 4. Update Manager (Async & Session Control)

- Ensures **non-blocking real-time updates** every second using `asyncio.sleep(1)`.
- Maintains **update count** using `st.session_state.update_count`, preventing chart flicker or key duplication.



IMPLEMENTATION

The implementation of the Live Crypto Analyzer involves a combination of asynchronous data handling, structured processing, and real-time visualization. The system is divided into four main components: the Binance WebSocket API (data source), Data Processor (backend logic), Streamlit Dashboard (frontend interface), and the Update Manager (async control). Each component plays a crucial role in delivering a seamless and responsive user experience.

1. Data Collection via Binance WebSocket API

The data stream originates from Binance's public WebSocket API, which provides real-time ticker data for all trading pairs. The WebSocket connection is handled using Python's `websockets` and `asyncio` libraries. A dedicated async function establishes a persistent connection to Binance's stream endpoint, listening for JSON-formatted messages containing live price, 24h change, volume, and high/low data.

2. Data Processing with Pandas

The received JSON data is parsed and converted into a structured pandas DataFrame. The backend logic filters relevant fields and applies sorting logic to:

- Identify the top 10 gainers and losers by 24-hour percentage change.
- Sort trading pairs by highest volume.
- Extract data for specific symbols like BTC/USDT for candlestick charting.

This processing is done asynchronously, ensuring that the app remains responsive during continuous data inflow.

3. Front-End Visualization using Streamlit & Plotly

The user interface is implemented using Streamlit, allowing rapid development of an interactive and web-based dashboard. Visualizations are rendered using Plotly, including:

- Bar charts for gainers, losers, and volume leaders.
- A candlestick chart that displays historical OHLC data for BTC/USDT.

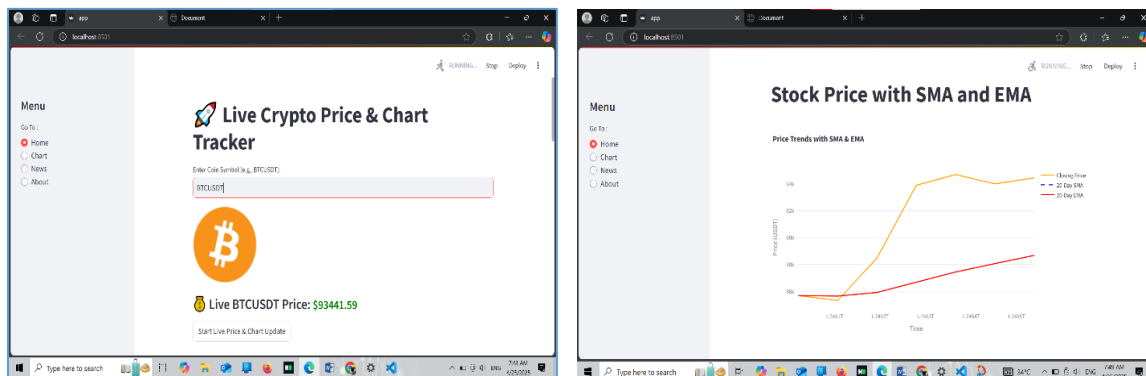
The dashboard layout includes a sidebar with two main sections: **Live Analysis** (main visuals) and **About** (project info, features, and technology stack).

4. Real-Time Update Management

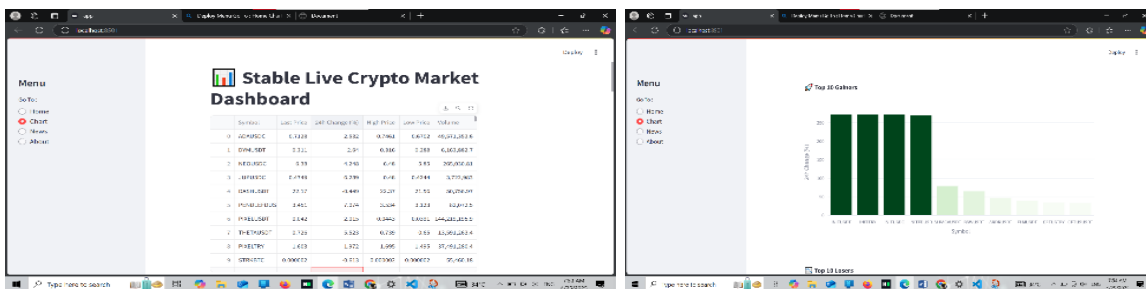
To ensure the dashboard updates without flicker or UI lag, an Update Manager is implemented using `asyncio.sleep(1)` for timed refreshes. The `st.session_state` is used to track updates and manage chart keys.

RESULT

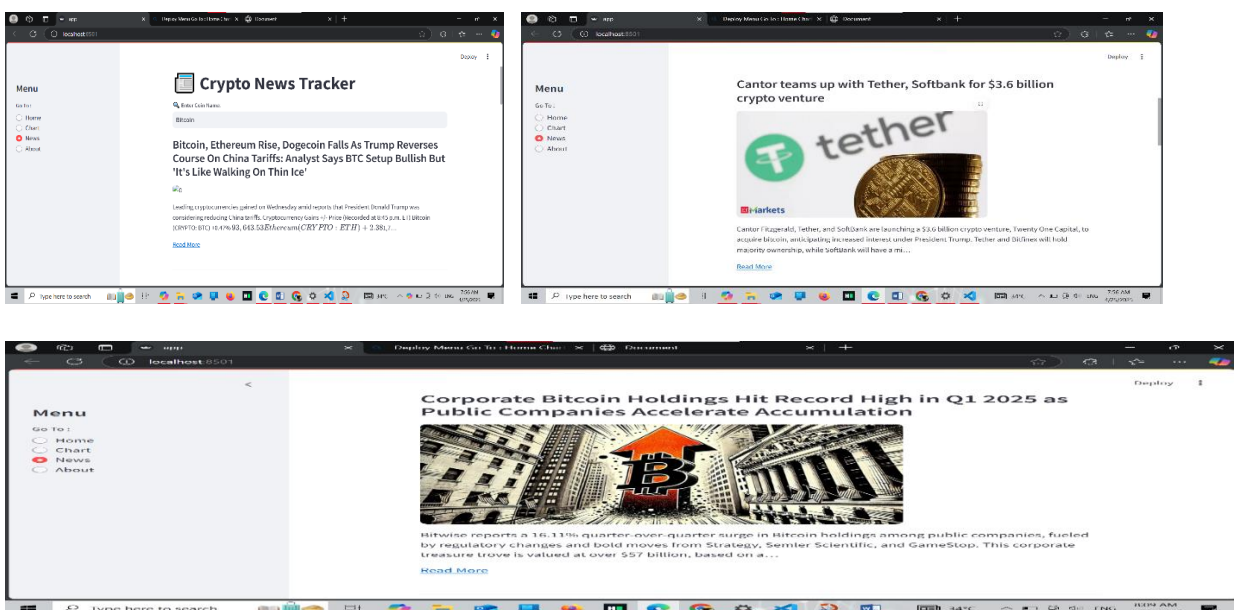
1. Home Page



2. Chart Page



3. News Page



CONCLUSION

The Live Crypto Analyzer project successfully demonstrates the practical implementation of real-time data processing, analysis, and visualization in the context of the fast-paced cryptocurrency market. By leveraging the Binance WebSocket API, the system receives continuous and accurate ticker data for all listed trading pairs, ensuring that users have access to the latest market information at any given moment.

The backend, built using Python's `asyncio` and `websockets` libraries, efficiently handles the asynchronous data flow without blocking the UI, while `pandas` enables robust data transformation and logic application. The dashboard identifies top gainers, losers, and high-volume pairs and presents this data using interactive charts powered by Plotly. Special attention is given to the BTC/USDT trading pair through a dynamic candlestick chart, helping users perform technical analysis within the dashboard itself.

The front-end is built with Streamlit, offering a simple yet powerful interface that updates every second without performance issues, thanks to the `Update Manager` and `st.session_state`. This ensures a smooth user experience with consistent chart rendering and session tracking.

Overall, the project provides a strong foundation for real-time financial analysis applications. It not only delivers a functional and informative user tool but also showcases modern techniques in data engineering, asynchronous programming, and visualization. With further enhancements such as alerts, technical indicators, and deployment, the Live Crypto Analyzer can evolve into a complete solution for both beginner and professional cryptocurrency traders.

FUTURE SCOPE

While the current version of the Live Crypto Analyzer successfully delivers real-time market insights and interactive visualizations, there is significant potential for further development and enhancement. Several key areas for future work have been identified to improve functionality, user experience, and scalability.

One of the primary enhancements would be to **add support for user-defined filters and custom alerts**. This would allow users to set thresholds for price changes, volume spikes, or specific trading pairs, and receive real-time notifications via email, SMS, or app pop-ups.

Another valuable addition would be **integration of technical indicators** such as Moving Averages (MA), Relative Strength Index (RSI), and Bollinger Bands into the candlestick chart. These tools would assist users in performing deeper technical analysis directly from the dashboard.

To enhance performance and scalability, the backend can be extended using **Fask API or Flask**, allowing it to support multiple users simultaneously and improve request handling. Additionally, storing data in a **local or cloud database** (like PostgreSQL or MongoDB) would allow users to track historical trends and perform time-series analysis.

Finally, converting the dashboard into a **deployed web application** (using Streamlit Cloud, Heroku, or AWS) would allow global access without requiring users to run the script locally. With these upgrades, the Live Crypto Analyzer could evolve from a local tool to a powerful, production-ready crypto monitoring platform.

- **Real-Time Data Streaming**

Uses WebSockets to provide continuous live updates without manual refresh.

- **Interactive Visualizations**

Displays price movements, gainers/losers, and volume analysis using dynamic Plotly charts.

- **User-Friendly Interface**

Built with Streamlit for a clean, responsive, and easy-to-navigate UI.

- **Efficient Asynchronous Processing**

Handles real-time data smoothly using Python's `asyncio` and `websockets` for non-blocking performance.

- **Data Filtering and Sorting**

Automatically identifies top-performing and underperforming coins based on 24h change and volume.

- **Candlestick Chart Integration**

Visualizes technical trends of BTC/USDT pair, assisting in basic technical

- **Custom Chart Management**

Utilizes `st.session_state` to manage updates and prevent chart flickering or duplication.

- **Lightweight and Open Source**

Can be run on local systems without the need for heavy software or cloud setup.

- **Scalable Architecture**

Easy to expand by adding more trading pairs, technical indicators, or data sources.

- **Great for Educational Use**

Helps students and beginners understand real-time data handling, visualization, and crypto markets.

REFERENCES

1. Chitlange, V., & Tambade, A. (2023). *A Survey on Real-Time Market Dynamics Through Visual Dashboards*. Retrieved from Academia.edu
2. Raj, K. (2024). *Crypto Analysis and Visualization Using Python and Streamlit*. International Journal of Scientific Development and Research (IJS DR), Vol. 9, Issue 5. Retrieved from <https://www.ijedr.org/papers/IJEDR2405202.pdf>
3. Chauhan, M., & Jain, R. (2023). *Cryptonik – A Live Cryptocurrency Tracker*. International Journal of Innovative Science and Research Technology (IJISRT), Volume 8, Issue 6. Retrieved from <https://ijisrt.com/assets/upload/files/IJISRT23JUN227.pdf>
4. Langer, J., & Maier, M. (2023). *CryptoVisualizer+: Enhancing Visual Analysis for Cryptocurrency Time-Series Data*. In: Gschwandtner, T., Federico, P. (eds) Time-Oriented, Systematic Visual Analytics. Springer. https://link.springer.com/chapter/10.1007/978-3-031-62245-8_8
5. Roy, P., & Paul, R. (2024). *An Integrated Framework for Cryptocurrency Price Forecasting and Trend Analysis Using Machine Learning*. Applied Sciences, 15(4), 1864. <https://www.mdpi.com/2076-3417/15/4/1864>
6. Shah, A., & Mishra, A. (2022). *Time-Series Prediction of Cryptocurrency Market Using Machine Learning Techniques*. Retrieved from Academia.edu
7. Tripathy, N., & Hazarika, A. (2020). *KryptoOracle: A Real-Time Cryptocurrency Price Prediction Platform Using Twitter Sentiments*. arXiv preprint arXiv:2003.04967. <https://arxiv.org/abs/2003.04967>

