Project Report: Live Cryptocurrency Data Updater

1. Project Overview: The "Live Cryptocurrency Data Updater" is an automated system that fetches real-time cryptocurrency data from the CoinGecko API and updates a Google Sheet at regular intervals. This enables users to access the latest market data without running a script manually.

2. Objectives:

- Automate the retrieval of cryptocurrency market data.
- Store and update data in a Google Sheet.
- Provide real-time access to users via a shareable Google Sheet link.
- Perform basic data analysis on the fetched data.
- Implement Google Apps Script to run the update process automatically.

3. Workflow:

1. **Fetching Data:**
    - A Python script fetches data from the CoinGecko API.
    - It retrieves the top 50 cryptocurrencies based on market capitalization.
2. **Data Processing:**
    - The script processes the data into a structured format.
    - Identifies the top 5 cryptocurrencies by market cap.
    - Calculates the average price of the top 50 cryptocurrencies.
    - Analyzes the highest and lowest 24-hour percentage price change.
3. **Google Sheets Integration:**
    - The script uses Google Sheets API to write data into specific sheets.
    - Separate sheets are maintained for live data, top 5 market cap, and summary analysis.
4. **Automation via Google Apps Script:**
    - A trigger function in Google Apps Script schedules periodic updates.
    - The script runs even when the Python script is not active.
5. **User Access:**

- The Google Sheet is shared with users via a link, enabling them to see live updates.
- Live updates can be accessed here: Live Crypto Data Sheet

4. Technologies Used:

- **Python**: For fetching and processing cryptocurrency data.
- **CoinGecko API**: Provides real-time cryptocurrency market data.
- **Google Sheets API**: Enables integration between Python and Google Sheets.
- **Google Apps Script**: Automates the sheet updates without requiring manual execution.
- **GCP Service Account**: Used for authentication to access Google Sheets.
- **gspread Library**: Python library to interact with Google Sheets.
- **pandas Library**: Used for data analysis and processing.
- **schedule Library**: Automates periodic execution of the Python script.
- **Google Cloud Console**: For Service Account & Authentication, Google Sheets API Enablement, Permissions & Access Control, Scalability & Security.

5. Automation & Deployment:

- A Python script is initially used to set up and validate data updates.
- Google Apps Script is deployed with a trigger to update the sheet at regular intervals.
- The Python script can be removed after setting up the automation in Google Apps Script.
- The sheet remains updated without manual intervention.

6. Error Handling & Rate Limits:

- The script checks for API response status and handles errors gracefully.
- A rate limit error (429) is mitigated by reducing the update frequency or subscribing to a premium API plan.
- If an API failure occurs, the script logs the error and retries later.

7. Expected Outcome:

- A fully automated Google Sheet with real-time cryptocurrency updates.
- Users can access live market data without running any script.

- Basic analysis provides insights into market trends.
- The system remains operational even after the Python script is removed.

8. Challenges Faced:

- Configuring Google Sheets API access and permissions.
- Handling API rate limits and request failures.
- Automating periodic updates using Google Apps Script.

9. Future Enhancements:

- Implement data visualization using Google Sheets charts.
- Add alerts for significant price changes.
- Expand to track additional metrics like historical trends and trading volume.

This project ensures real-time cryptocurrency tracking with minimal manual intervention. The combination of Python, Google Sheets API, and Google Apps Script makes it a scalable and efficient solution. 🚀

Author By - Bharat Bairwa