**Cryptocurrency Price Tracker**

**Automated Live Price Tracker and Analyzer**

**Author: Gowtham R** (Team 6),

**Project Type:** Individual Project

**EXECUTIVE SUMMARY:**The Automated Cryptocurrency Scraper & Analyzer is an advanced Python tool designed to provide a comprehensive, real-time snapshot of the cryptocurrency market. The script automates the extraction of live data for the top 20 cryptocurrencies from CoinMarketCap using Selenium WebDriver. It runs in a continuous 10-seconds loop, capturing key metrics like price, 24-hour change, and market capitalization, and appends this data with a timestamp to a CSV file for historical analysis.

Beyond simple data collection, the tool performs a multi-layered analysis directly in the terminal, presenting all findings in clean, formatted tables. Key analytical features include identifying top gainers and losers, calculating market sentiment and volatility, and providing a full statistical summary. The script culminates in a data-driven "Recommendation Assistant" that suggests a coin to research based on a scoring model and provides speculative long-term growth projections under conservative, moderate, and aggressive scenarios. This project serves as a powerful, end-to-end solution for investors and analysts seeking automated, real-time market intelligence.

**PROJECT OVERVIEW:**

**a) Objective:**

To design, develop, and implement an autonomous Python script that not only scrapes real-time cryptocurrency data but also performs a deep, multi-faceted analysis. The primary goal is to transform raw data into actionable insights, providing users with a comprehensive market overview, trend indicators, and data-driven recommendations directly in the terminal.

**b) Scope:**

* Scrape data for the **top 20 cryptocurrencies** from CoinMarketCap.
* Extract key data points: Name, Price (USD), 24h Change (%), and Market Cap.
* Perform and display **four levels of analysis**: a raw data sample, top gainers/losers, advanced metrics, and highly advanced statistical summaries.
* Generate a **single coin recommendation** with reasoning and a long-term growth outlook.
* Append the newly scraped data with a precise timestamp to a local CSV file for persistent historical logging.  
    
  **c) Key Features:**
* **Continuous Auto-Updating:** The script operates in a continuous loop, automatically refreshing market data every 10 seconds without manual intervention.
* **Live Web Scraping:** Utilizes Selenium to reliably extract data from the dynamic, JavaScript-rendered content on CoinMarketCap.
* **Multi-Layered Terminal Analysis:** Presents a full spectrum of analytics, from basic price tables to advanced metrics like market volatility and weighted sentiment.
* **Data-Driven Recommendation Assistant:** Employs a scoring model to suggest a cryptocurrency for further research and provides scenario-based growth projections.
* **Formatted Terminal UI:** Leverages the tabulate library to display all data and analysis in clean, bordered, grid-style tables for enhanced readability.
* **Historical CSV Logging:** Automatically appends each scraped dataset with a timestamp to a CSV file, building a historical database for future analysis.
* **Headless Operation:** Capable of running in the background without a visible browser window, making it ideal for continuous, low-resource monitoring.

**TECHNICAL IMPLEMENTATION:**

**a) Architecture:**

The system is built on a robust, sequential pipeline that executes within a continuous loop:

1. **Initialization:** Launch Chrome WebDriver and configure Selenium options (headless or visible mode).
2. **Data Extraction:** Identify table elements and fetch key details (rank, name, price, 24h change, market cap).
3. **Data Processing:** Clean, format, and structure the scraped data using pandas.  
   4. **Analysis & Visualization:** A series of dedicated functions perform sequential analysis on the DataFrame. The results of each analysis are immediately formatted into tables using tabulate and printed to the terminal.  
   **5. Data Storage:** Append timestamped records to a CSV file for historical tracking.  
   **6. Loop Management:** Continuously monitor and refresh data at configurable time intervals (default 10 seconds).

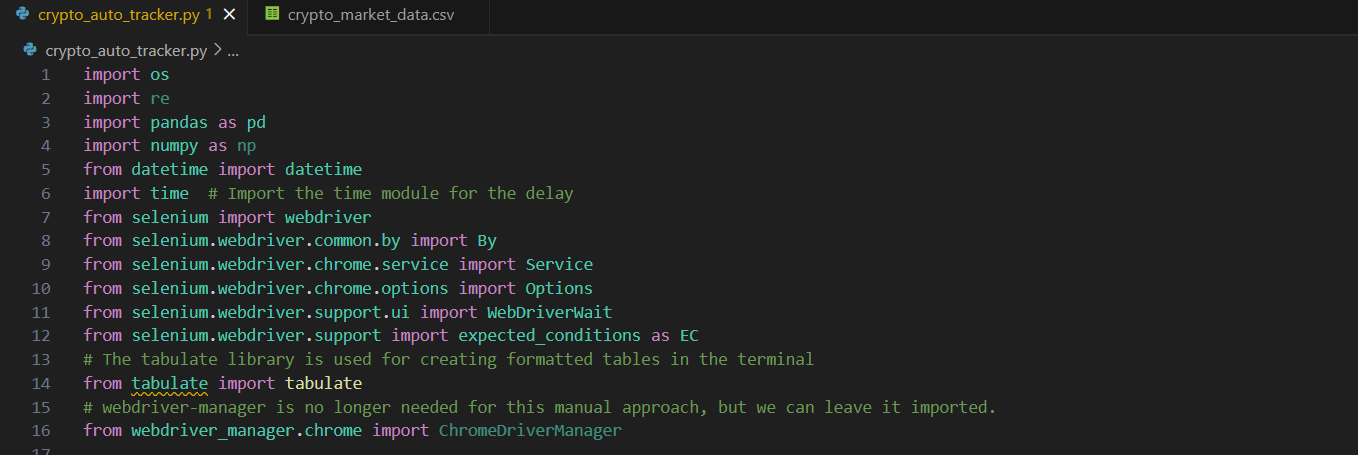
**b) Technology Stack:**

* **Python 3.x:** Core programming language for automation and data handling.
* **Selenium WebDriver:** Automates browser actions and scrapes websitesite data.
* **Webdriver Manager:** Handles automatic ChromeDriver installation and version management.
* **Pandas:** Processes, filters, and exports structured coin data into CSV format.
* **Datetime Module**: Generates and formats timestamps for each data entry.
* **OS Module:** Manages file handling and CSV operations.
* **Time Module:** Controls scraping frequency and loop intervals for continuous operation.
* **Google Chrome:** Browser platform controlled by Selenium for data extraction.

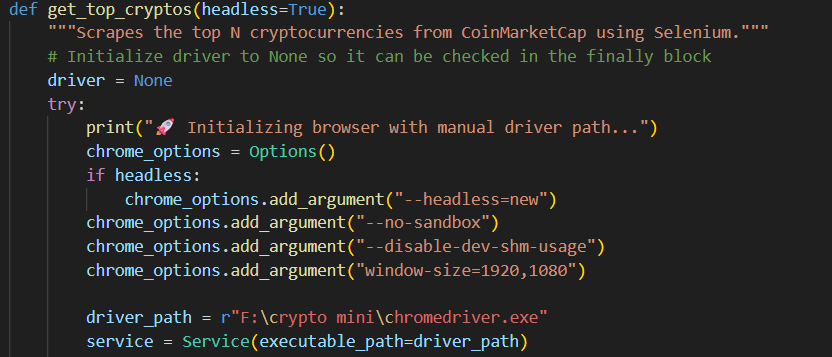
**c) Performance Metrics:**

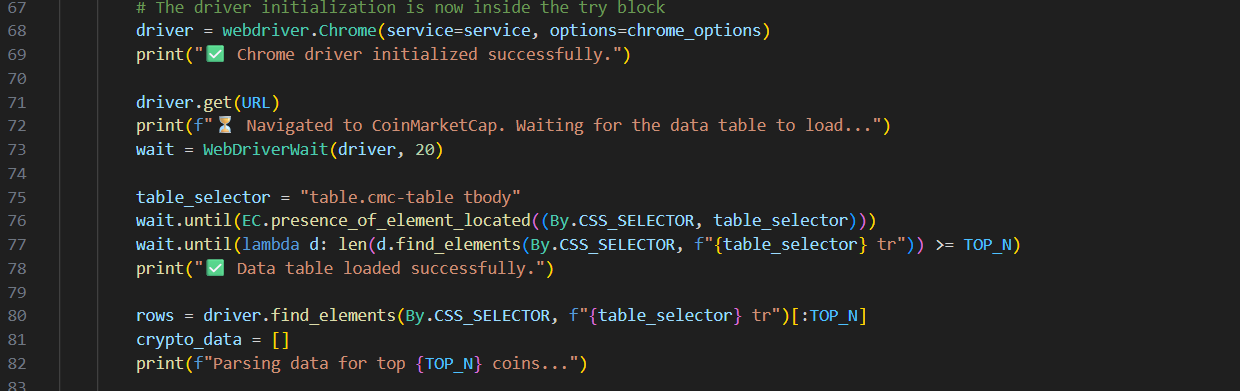
* **Processing Capacity:** Handles data for the **top 10–50 cryptocurrencies** per cycle.
* **Execution Time:** Completes one full data scrape and CSV update in **5–10 seconds** (in headless mode).
* **Success Rate:** High stability with built-in exception handling for loading or timeout errors.
* **Output Format:** Structured and timestamped **CSV file** suitable for trend analysis and dashboard integration.
* **Update Frequency:** Fully configurable; supports **as low as 10 seconds per refresh** for real-time tracking.

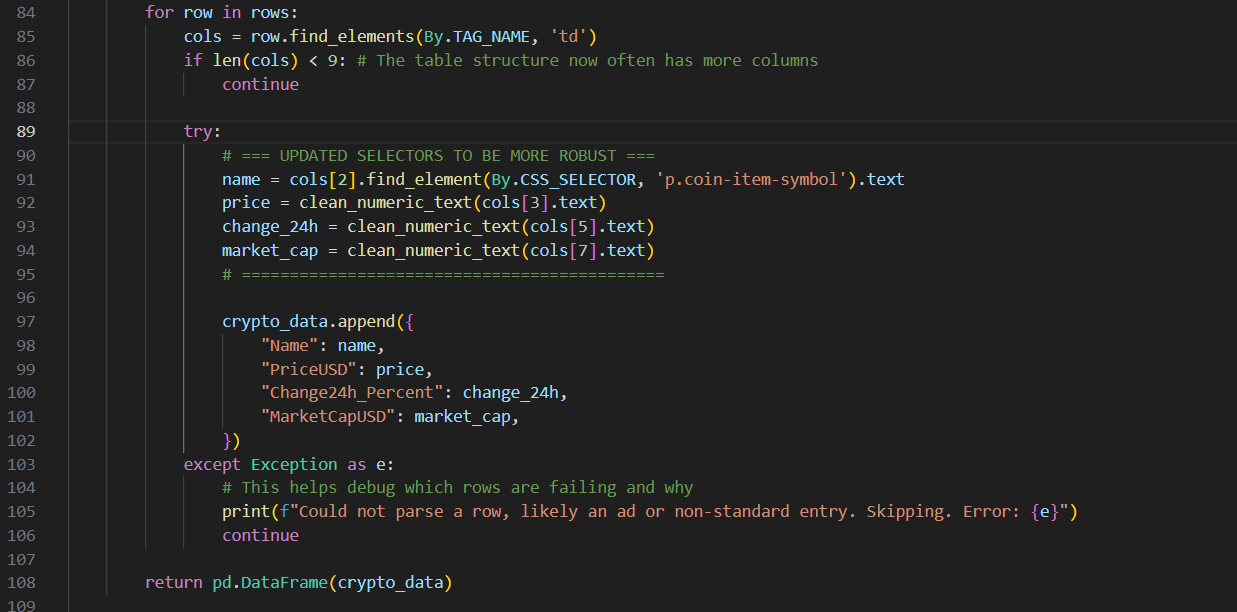
**SOURCE CODE IMPLEMENTATION (Screenshot):**

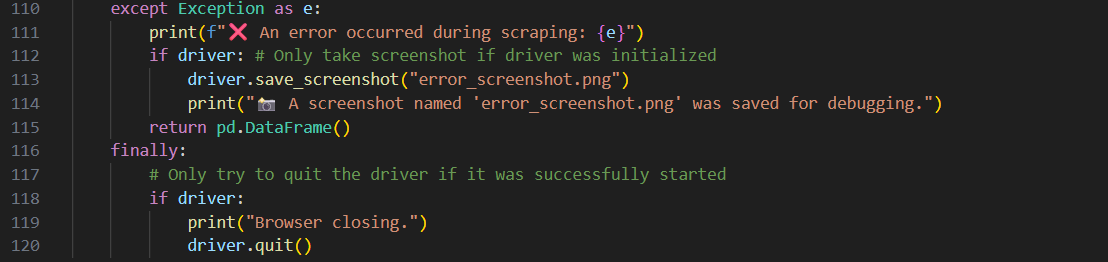


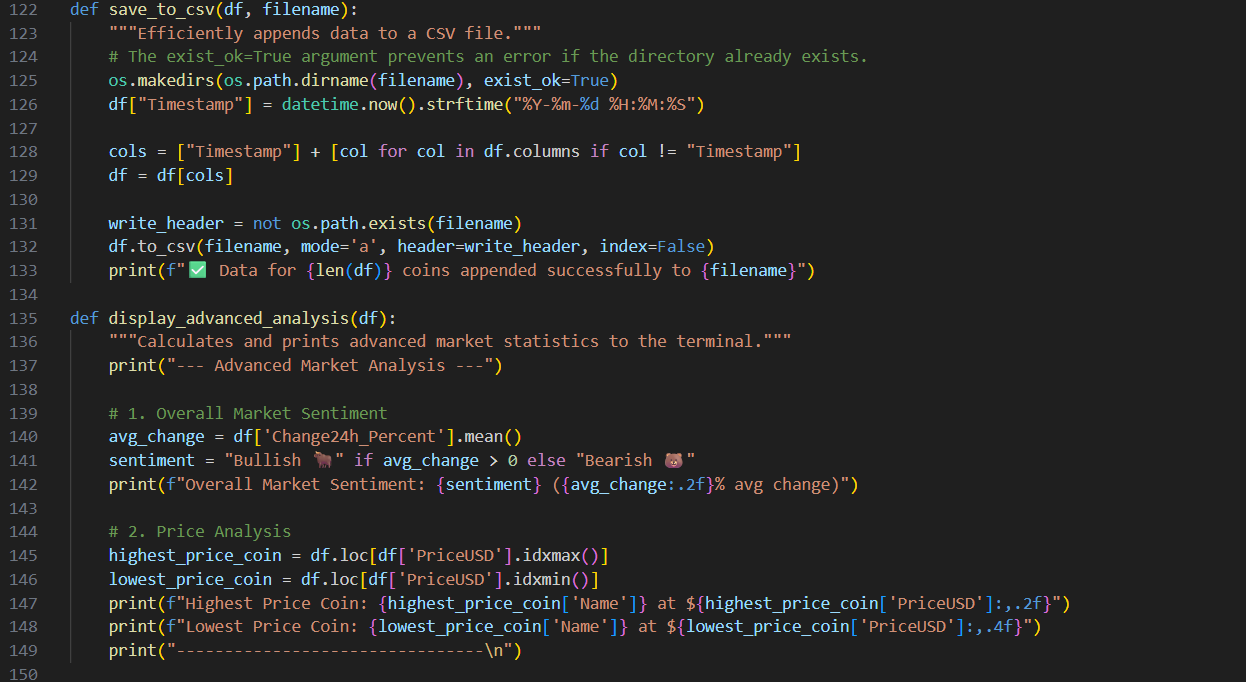


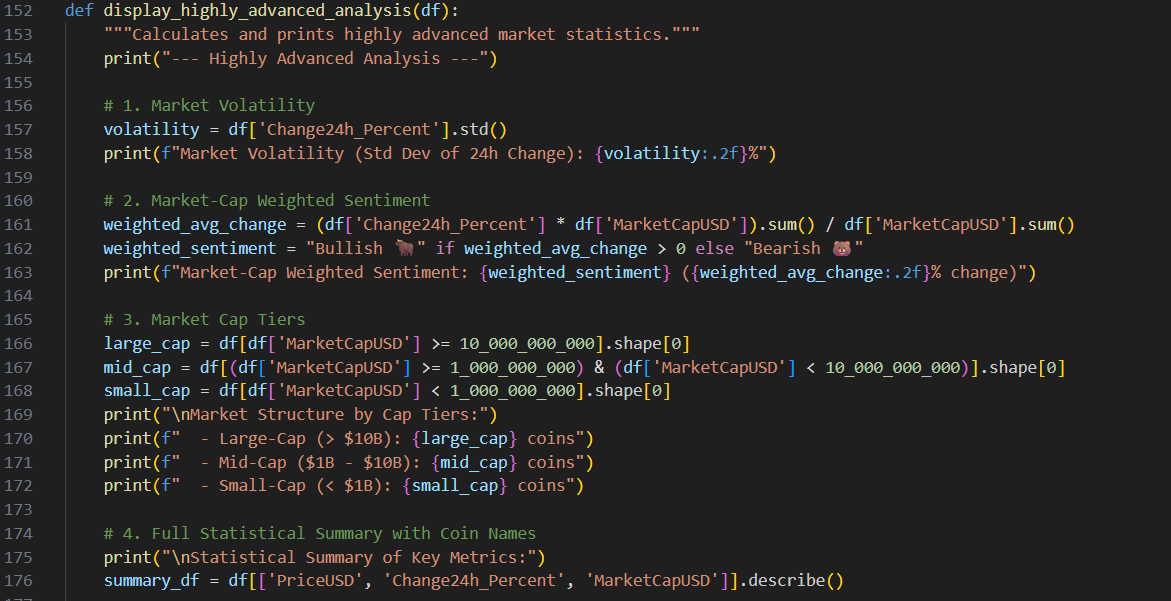


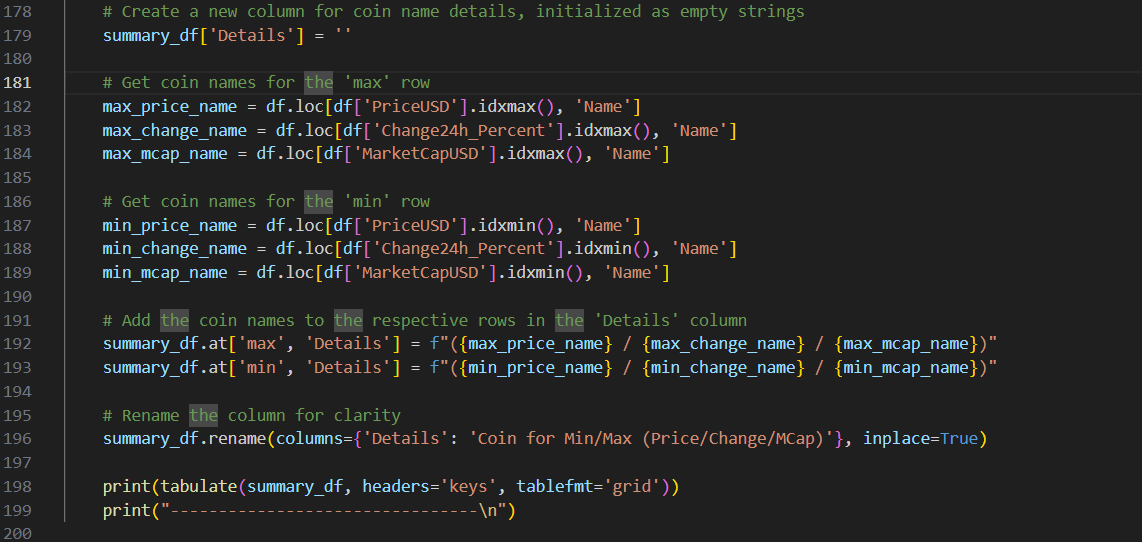


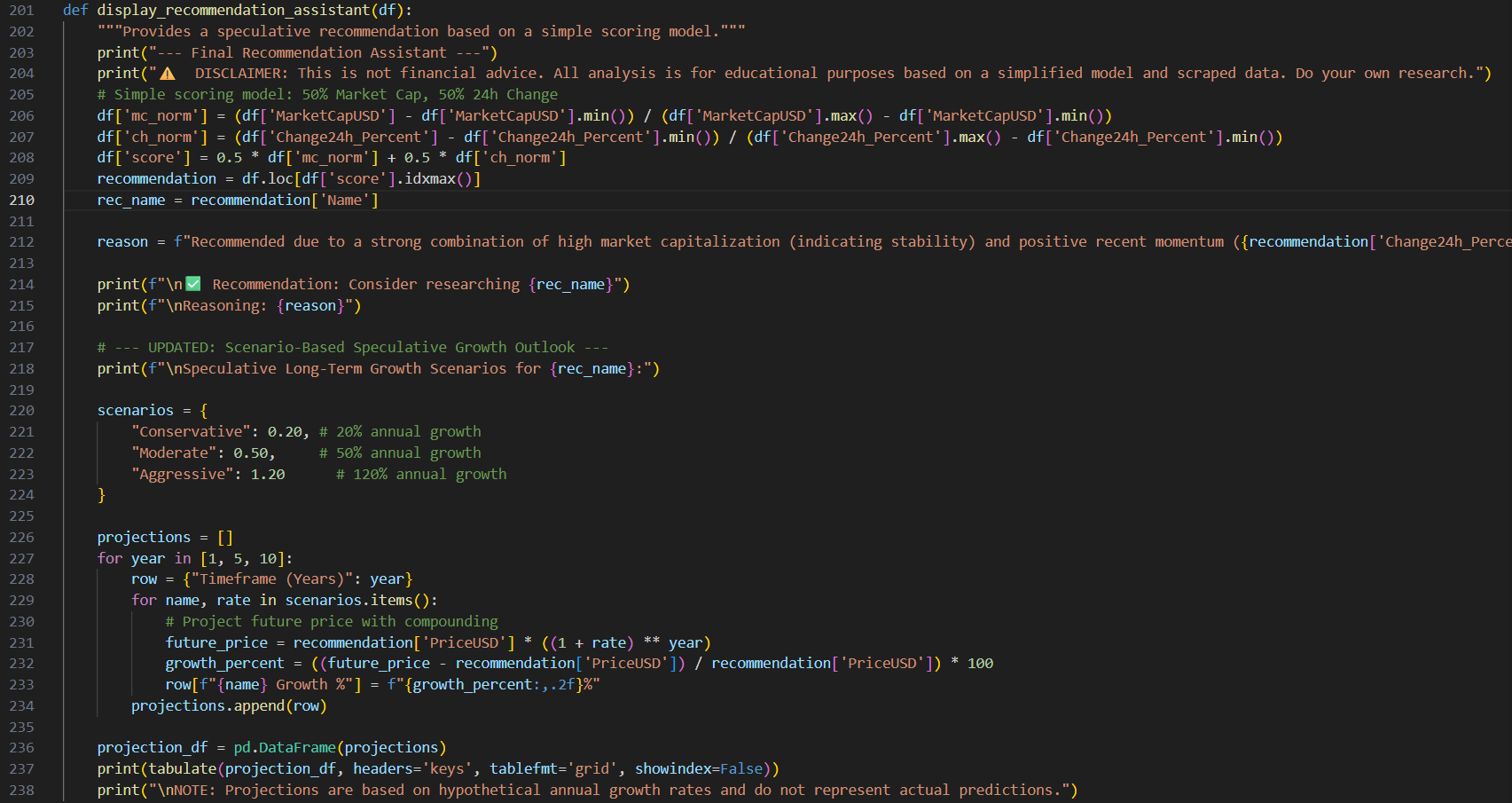


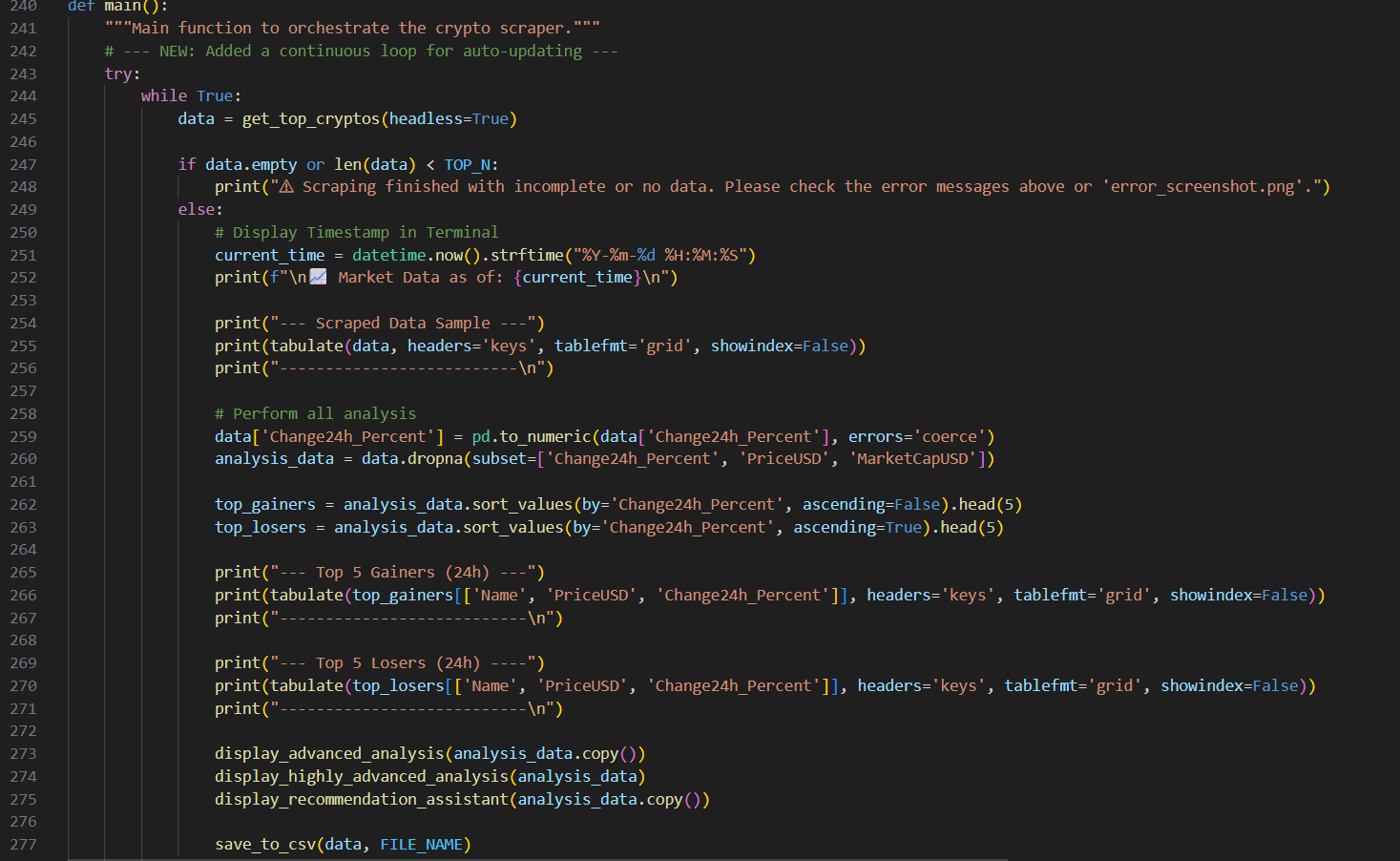


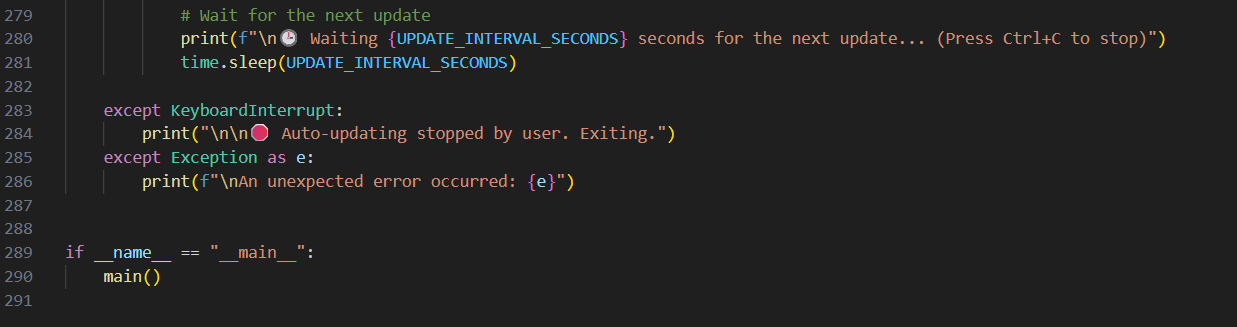




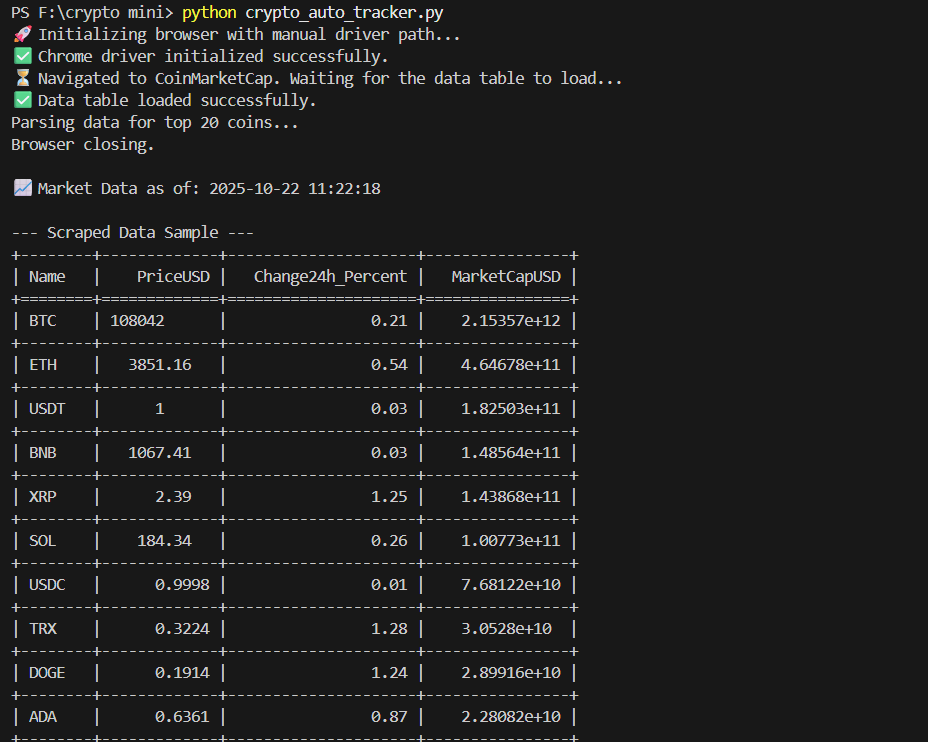


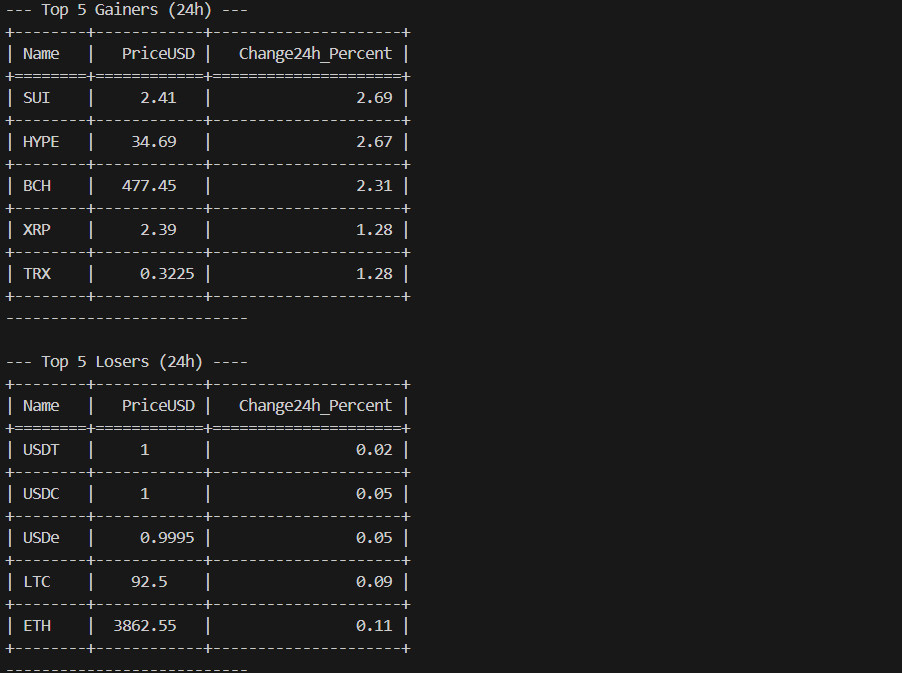


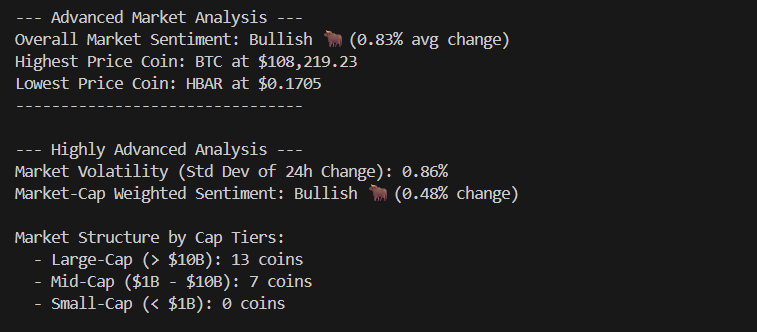


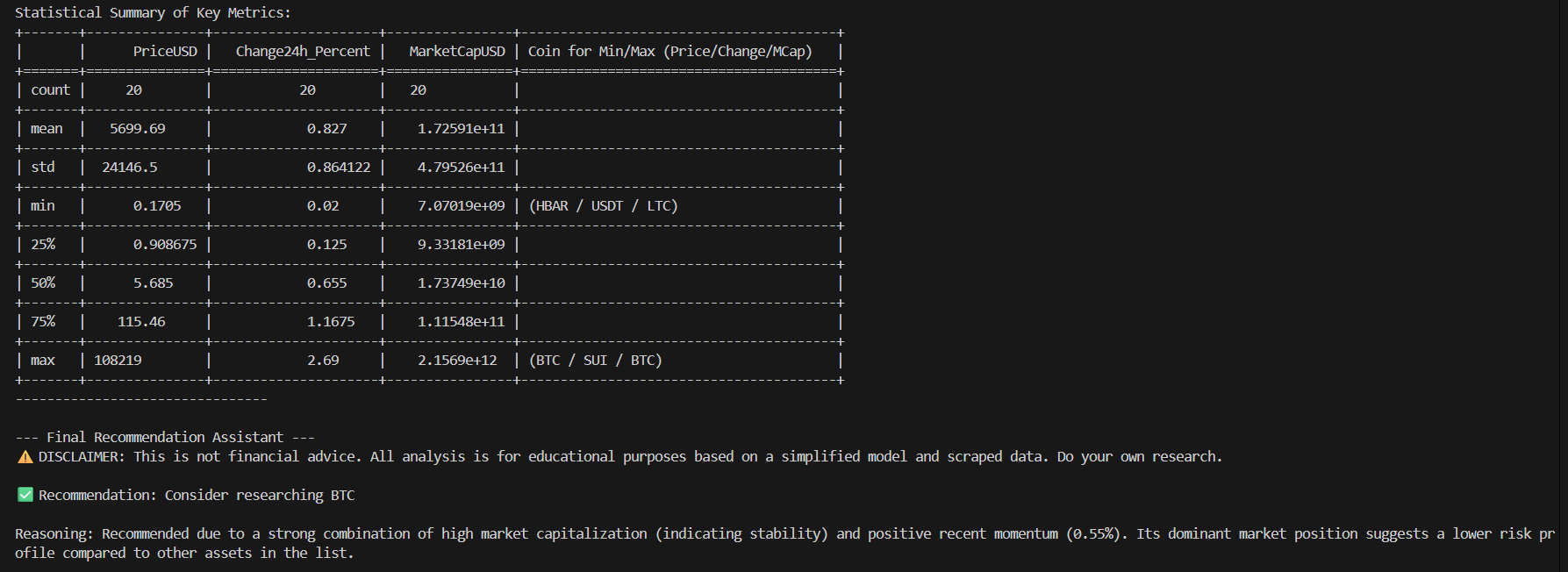


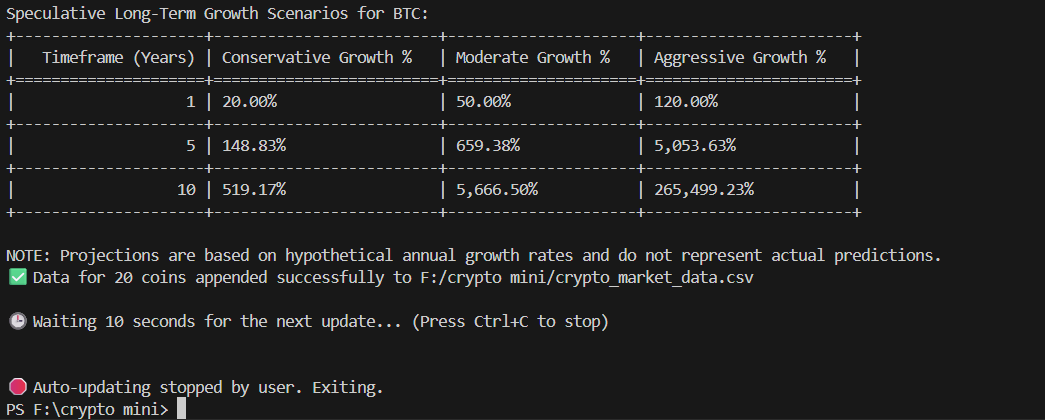
**Result:**

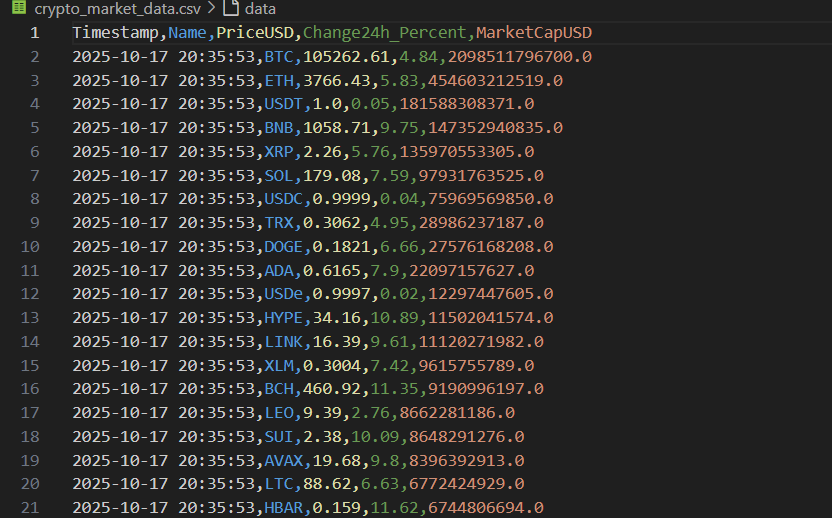










**Saved in crypto\_market\_data.csv  
**

**KEY ACHIEVEMENTS:**

**a) Technical Accomplishments:**

· **Successful Web Automation:** Seamless integration of **Selenium WebDriver** with **CoinMarketCap** for dynamic data extraction.

· **Real-Time Monitoring:** Live tracking of top cryptocurrency prices with configurable refresh intervals (default 10 seconds).

· **Robust Error Handling:** Automated exception management for timeouts, missing elements, and connectivity issues.

· **Efficient Data Processing:** Optimized use of **pandas** for structuring, filtering, and formatting large datasets.

· **Headless Operation:** Fully functional background execution without browser visibility for low-resource automation.

· **Scalable Architecture:** Supports configurable coin limits, scraping intervals, and advanced filtering logic.

· **Historical Data Management:** Continuous CSV logging with timestamps for trend analysis.

**b) Process Automation Benefits:**

· **Consistency:** Automated and standardized data scraping at fixed intervals without manual effort.

· **Real-Time Insights:** Instant visibility of live market changes, gainers, losers and recommendation assist.

· **Offline Analysis:** Exported CSV files enable long-term tracking and offline data analysis.

· **Investment Support:** Quick identification of top-performing and underperforming cryptocurrencies.

· **Trend Tracking:** Historical data supports visual analysis and integration with analytical dashboards.

**LIMITATIONS AND CONSTRAINTS:**

1. **Current Limitations:**

* **Single Source Dependency**: Currently limited to CoinMarketCap as the primary data source.
* Dynamic Page Load Dependency: Relies on Selenium WebDriver for rendering JavaScript content, which may cause slow loading on unstable networks.
* Refresh Interval: Requires a minimum 10-second delay between scrapes to prevent IP blocking or excessive browser load.
* Session-Based Logging: Historical data is recorded only during active script execution sessions.
* **No Automated Alerts:** The system does not currently support notifications for significant price movements or thresholds.

1. **Resource Constraints:**

* System Performance: Dependent on system specifications and browser performance; may slow down on low-end machines.
* Network Latency: Scraping speed and data freshness vary with internet connection quality.
* Storage Limitation: Uses local CSV files for storage; lacks database or cloud integration for large-scale data.
* Memory Usage: Selenium and Chrome instances consume moderate memory during long continuous runs.
* Concurrency: Designed for a single-threaded operation; parallel data extraction for multiple sources is not yet implemented.

**PROJECT IMPACT AND VALUE:**

**a) Immediate Benefits:**

* **Time Efficiency:** Completely automates the laborious task of manual data collection and analysis.
* **Real-Time Market Awareness:** Provides an immediate and comprehensive snapshot of the market's state, including top movers and overall sentiment.
* **High-Quality Data:** Ensures data is consistent, structured, and accurately timestamped for reliable analysis.
* **Informed Decision-Making:** The multi-layered analysis provides the necessary context for users to make more informed trading or investment decisions.

**b) Long-term Potential:**

* **Historical Database Creation:** The continuous CSV logging builds a valuable historical dataset, which can be used to back-test trading strategies or train machine learning models.
* **Foundation for Advanced Tools:** The script serves as a powerful backend for future enhancements, such as a web-based dashboard or an automated alert system.
* **Predictive Analytics:** The collected data is ideal for developing more sophisticated predictive models to forecast market trends.

**FUTURE ENHANCEMENTS:**

**Planned Improvements:**

1. **Historical Data Integration:** Incorporate an API to fetch historical data, enabling the calculation of more reliable long-term indicators like moving averages.
2. **Machine Learning Models:** Implement time-series forecasting models (e.g., ARIMA, LSTM) for more advanced price predictions.
3. **Web Dashboard:** Develop a web-based interface using a framework like Flask or React to visualize the data with interactive charts and graphs.
4. **Automated Alert System:** Integrate services like Twilio or SendGrid to send SMS or email alerts when certain price thresholds or market conditions are met.
5. **Cloud Deployment:** Host the entire system on AWS or Google Cloud Platform (GCP) for 24/7 continuous monitoring and scalability.
6. **Mobile Application**: Develop a cross-platform mobile app to provide users with live market

updates, alerts, and portfolio access anytime, anywhere.

**CONCLUSION:**

* The Automated Cryptocurrency Scraper & Analyzer project successfully demonstrates the practical use of web automation, dynamic data extraction, and real-time market monitoring through Selenium. It effectively addresses the need for continuous cryptocurrency Scraper & Analyzer, providing a strong foundation for advanced analytics and informed investment decisions.
* The project’s technical implementation showcases proficiency in Python programming, Selenium WebDriver automation, data parsing, and structured data management using pandas. By capturing live market data at 10-second intervals and maintaining timestamped historical
* The Automated Cryptocurrency Scraper & Analyzer project successfully achieves its objective of creating a comprehensive, real-time market intelligence tool. It demonstrates a strong command of web automation, data analysis, and software engineering principles by building a script that is not only functional but also robust, readable, and feature-rich. By transforming raw web data into a sophisticated, multi-layered analysis and presenting it in a clean terminal interface, the project provides immense value to anyone interested in monitoring the cryptocurrency market. Its continuous operation and historical logging capabilities establish a solid foundation for even more advanced, data-driven applications in the future.