Crypto Dashboard

Real-Time Cryptocurrency Analytics and Reporting Platform

Naveen Prabakar

August 2025

Contents

1 Abstract

The *Crypto Dashboard* is a full-stack, cloud-deployed cryptocurrency analytics application providing real-time monitoring, advanced statistical analysis, and automated professional reporting of major digital assets. The system integrates a modern React-based frontend, a scalable backend written in Go, and a distributed Cassandra database optimized for time-series data. By combining automated ingestion pipelines, AI-driven insights, and PDF-based reporting with email subscription support, the platform is designed to offer extensible and professional-grade support for financial analysis, research, and monitoring of cryptocurrency markets.

2 Introduction

The rapid rise of cryptocurrencies has created demand for real-time analytics tools that provide insights beyond simple price tracking. The volatility of digital assets requires platforms capable of handling large volumes of live data, conducting advanced statistical analyses, and delivering automated professional reporting.

This project addresses those challenges by building a robust cryptocurrency analytics system with cloud-native deployment. Leveraging a React and TypeScript frontend, Go-based backend services, and an Apache Cassandra database, the platform enables analysts, developers, and researchers to explore actionable insights in digital markets.

3 System Features

The platform offers the following key features:

- **Live Price Tracking:** Access to real-time and historical pricing data for major cryptocurrencies via CoinGecko API.
- **Advanced Analytics:** Includes volatility analysis, trends, min/max, averages, and detection of top market movers.
- **AI-Powered Queries:** Allows natural language inputs that are automatically converted into Cassandra Query Language (CQL).
- **Automated PDF Reports:** Daily professional-grade PDF reports containing charts, metrics, and AI-generated insights.
- **Email Subscription:** Users can opt-in to receive daily reports through an automated mailing system.
- **Interactive Frontend UI:** Responsive React dashboard with real-time charting and reporting capabilities.

4 System Architecture

The architecture is divided into several layers to ensure scalability and separation of concerns:

4.1 Frontend

- Built with React, TypeScript, and Vite.
- Provides interactive dashboards and data visualizations using charting libraries.
- Deployed on Vercel with SSL and continuous integration from GitHub.

4.2 Backend

- Written in Golang with modular services: ingestion, analytics, reporting, and AI.
- Exposes RESTful endpoints for price data, statistics, AI queries, and reports.
- Hosted on Render for auto-scaling and health-monitored deployments.

4.3 Database

- Apache Cassandra (AstraDB) as scalable, fault-tolerant time-series database.
- Separate tables for cryptocurrency data, email subscribers, and verification.

4.4 Ingestion

- Dedicated Go worker that schedules CoinGecko API fetches.
- Deployed as an AWS EC2 systemd service with CloudWatch monitoring.

4.5 AI Integration

- Uses OpenAI GPT for two primary tasks:
  1. Translating natural language into CQL queries.
  2. Summarizing daily analytics reports with professional narratives.

5 Project Structure

```
Crypto-Dashboard/
          Backend/
                crypto/backend/
                    AI.go
                    analytics.go
                    crypto.go
                    Report.go
                    main.go
                    ...
          Database/
                Create_Crypto_table.cql
                Email_subscribers.cql
                Email_Verify_table.cql
          Documents/
                Dashboard_demo.mp4
                Demo_video.txt
                report_example.pdf
          Frontend/crypto
                src/
                vite.config.ts
          README.md
```

6 Implementation Details

6.1 Database Schema

- **Crypto Tables:** Time-series price data including coin_id, timestamp, and price.
- **Subscriptions:** Stores active user emails with verification workflow.

6.2 API Reference

The backend exposes several endpoints:

- `/latest/{coin_id}` – Get latest coin price.
- `/history/{coin_id}` – Get N minutes of historical data.
- `/average/{coin_id}` – Compute average between time ranges.
- `/range/{coin_id}` – Find min/max prices in range.
- `/trend/{coin_id}` – Regression-based trend analysis.
- `/subscribe`, `/unsubscribe` – Manage email reports.
- `/report` – Download daily analytics PDF.

6.3 PDF Report Generation

Daily report generation involves:

1. Collecting daily statistics such as min, max, average, and volatility.
2. Creating trend and volatility visualizations with `gonum/plot`.
3. Producing AI-generated commentary to summarize daily movements.
4. Compiling into professional multi-page PDF with title, charts, and tables.

6.4 Email Delivery

- Subscribers receive daily emails with attached PDF.
- Email handling logic in `Report.go`.

7 Cloud Deployment

The platform is entirely cloud-hosted:

- **Frontend:** Vercel hosting with GitHub-based CI/CD.
- **Backend:** Render with auto-scaling, health checks, monitored logs.
- **Database:** DataStax AstraDB with globally distributed Cassandra for resilience.
- **Ingestion Worker:** AWS EC2 instance with systemd service and CloudWatch integration.

8 Development Workflow

- **Frontend:**
  - `npm run dev` – launch dev server
  - `npm run build` – build production build
  - `npm run lint` – linting checks
- **Backend:**
  - `go run main.go analytics.go AI.go Report.go` – API server
  - `go run crypto.go` – ingestion worker
  - Dockerfile and scripts for containerized deployment

9 Demonstration

Supporting artifacts include:

- *Demo Video:* Showcases frontend dashboard functionality (see `bottom of README page`).
- *Sample Report:* Example of auto-generated PDF report (see `report_example.pdf`).

10 Conclusion & Future Work

The Crypto Dashboard successfully integrates real-time ingestion, advanced analytics, AI interpretation, and automated reporting into one cohesive platform. It demonstrates strong potential for applications in both financial monitoring and educational use in data-driven systems.

Future Work:

- Integration of ML models for short-term forecasting.
- Enhanced account management with authentication and user roles.
- Integration with crypto exchanges and wallet APIs for portfolio monitoring.
- Multi-language reporting with localization options.
- Expansion to other asset classes (commodity, equities).

11 License

This project is licensed for educational and non-commercial use. Dependencies retain their respective licenses.