**Chain Report AI: Enabling On-Demand Cryptocurrency Analysis**

**(ChainReportAI: A MODE Hackathon entry DEC 2024)**

Chain Report AI is an innovative system designed to provide users with comprehensive, hyper-personalized and automated cryptocurrency news and token reports. The ideal user scenario begins on Social platforms like Twitter/X, where a user tags **@ChainReportAI** in a public post, requesting a Token Report for a specific cryptocurrency such as $MODE or $SOL. Leveraging the Eliza AI framework, the ChainReportAI agent monitors social media interactions for these mentions and intelligently processes requests using its tokenreportAction functionality, a core feature of the **plugin-chainreportai** module.

Upon detecting a valid request, the Eliza agent initiates an asynchronous fetch() to the Apps Script-hosted API endpoint, sending a payload containing {tokenSymbol, requestor, ticketId}. The API enrolls the job in its spreadsheet-based task queue and responds to the agent with an acknowledgment. Concurrently, the Apps Script environment triggers periodic processing to compose reports. Using advanced AI models, relevant news, and token data are gathered, analyzed, and formatted into Markdown for efficient styling. This content is rendered into a styled Google Doc and exported as a PDF. The PDF is shared via a "Anyone with Link" policy, with the original Google Doc discarded for security.

The **plugin-chainreportai** periodically queries the Apps Script endpoint to check the status of queued tasks. Once a report is marked complete, the Eliza evaluator generates a public tweet tagging the requestor and including the secure report URL. In an advanced version, the agent could handle payment processing before initiating the report, using pay-to-address details provided by GOAT's Solana and ERC-20 plugins. Upon completion and payment verification, the agent would privately deliver the report URL to the user via direct message (if permissible).

This streamlined system integrates Eliza, Google Apps Script, and ams to integrate blockchain technologies like GOAT to deliver real-time, user-centric solutions for cryptocurrency news and analysis, enhancing the accessibility and usability of financial data on demand.

**Personal Impressions:**

By no means am I an expert (I literally learned about this hackathon and the Eliza framework on the 17th), I am disappointed in the level of frustration I encountered along the way. Much of my own making. specially the amount of time I spent port-fiddling and trying to debug errors and omissions before I could get the project to cleanly build and stand-up the chat client. I've watched and rewatched the devschool Ep 1a/1b, 2, and 3 and a few others on YouTube. I have high hopes for the framework and its unfolding roadmap. The GOAT, Twitte/X, and BirdEye as still a bit opaque for me, but that will resolve over time.

**Live Staged client (low probability of success at this stage)**

<https://enough-reindeer-similarly.ngrok-free.app/>

**Shared Folder in Google Drive, showing rudimentary MVP example reports:**

<https://drive.google.com/drive/folders/1lRWvhopDmGnnWdOYZnsexl5-3e9H1807?usp=sharing>

**Token Report Job Queue Spreadsheet:**

<https://docs.google.com/spreadsheets/d/18aN9oqBKOIgTWrfhTq5lzmXJ6Fr_fGPnn6QmWPySx2E/edit?usp=sharing>

**Google Doc containing the bound Apps Script endpoints that are targetted by the Eliza character;**

<https://docs.google.com/document/d/1E4JwHBFtIugwqiftRmJJDOFNDRBilDYkCoWBQdrDM_8/edit?usp=sharing>

**Postman Test Scenario;**

[**https://docs.google.com/document/d/1VTmKMA8VFo8Ha0dt249Ak2CjT\_DVGgeodVHysLUvRM0/edit?usp=sharing**](https://docs.google.com/document/d/1VTmKMA8VFo8Ha0dt249Ak2CjT_DVGgeodVHysLUvRM0/edit?usp=sharing)

**Character composition file:**

<https://docs.google.com/document/d/1kObmcf5pZkM4a8tyoCV7el2ppilAiYFdHAPlgNX3hew/edit?usp=sharing>