SDE

🧠 TokenWise — Real-Time Wallet Intelligence on Solana

🔍 Project Overview

TokenWise is a real-time intelligence tool designed to monitor and analyze wallet behavior for specific tokens on the **Solana blockchain**. Your goal is to build a system that tracks the **top 60 token holders**, captures their transaction activity in real time, and visualizes market trends through a clean dashboard.

> 💡 *Target Token:*
>
> *Contract Address:* `9BB6NFEcjBCtnNLFko2FqVQBq8HHM13kCyYcdQbgpump` *(Solana)*

🎯 Objectives

- 📃 **Discover** the top 60 wallets holding a specific Solana token
- 🔄 **Real-time monitoring** of token-related transactions (buys/sells)
- 🛠️ **Identify** protocols used (e.g., Jupiter, Raydium, Orca)
- 📈 **Generate insights dashboard** for market trend analysis
- 🕰️ **Support historical analysis** with custom time filters

---

💻 Technical Requirements

1. Top Wallet Discovery

- Connect to Solana via RPC
- Fetch and display top 60 wallet addresses holding the token
- Store balance, wallet address, and token quantity

2. Real-Time Transaction Monitoring

- Track token buys and sells by these wallets in real time
- Capture:
  Timestamps
  Amounts
  Buy/Sell indicator
  Protocol used (Jupiter, Raydium, Orca)
  Wallet address

3. Insights Dashboard

- Create a simple dashboard (can be local or web) that shows:
  Total number of buys vs sells
  Net direction (buy-heavy, sell-heavy)
  Wallets with repeated activity
  Protocol usage breakdown

4. Historical Analysis

- Enable querying of past activity with time filters

- Exportable reports (CSV/JSON) with transaction summaries

🧱 Tech Stack

Use tools you're most comfortable with. Here are recommendations:

Component	Suggested Tools
Backend	Node.js with TypeScript
Blockchain Interface	`@solana/web3.js` or `@solana/kit`
DB	SQLite (simple) or PostgreSQL (preferred)
Dashboard	Streamlit / React.js / Flask + Chart.js

📦 Submission Requirements

To evaluate both your technical skills and communication, please submit:

🛠️ Technical Deliverables

- ✅ Working codebase + README
- ✅ Setup instructions and dependencies
- ✅ Sample output data (JSON/CSV)
- ✅ Working **dashboard** with insights

📽️ Presentation & Documentation

- 🎥 **2-Min Strategy Video** — explain your overall approach
- 🎥 **2-Min Code Walkthrough Video** — explain architecture & key logic
- 🎥 **2-Min results from dashboard video** — explain the features of dashboard
- 📊 **PPT Deck (3–5 slides)** — covering:
  Design and thought process
  Assumptions and tradeoffs
  Possible future improvements

📄 Final Step

👉 **Submit your project and videos via this Google Form:**

➡️ **SUBMIT HERE**

⏱️ Timeline

- Ideal completion time: **Wednesday 11:59pm IST**
- Focus on real-time logic, insights dashboard, and clean architecture

---

💬 Collaboration & Support

- Use telegram for doubts Tg Group Link - https://t.me/+nRIHhNPHXe04ZjE9

💡 Tips

- Focus on clean MVP first; add enhancements later
- Use caching/snapshotting wisely for real-time fetches
- Think deeply about user needs for the dashboard
- Communicate assumptions in your videos/slides

🔗 Resources

- Solana Developer Docs
- Jupiter Aggregator
- Solana Web3.js

🚀 Let's Go!

This is the first step in our hiring process, and it's your opportunity to stand out.

We're not just looking for code – we're looking for creativity, clarity of thought, and how you approach real-world problems.

Build something powerful, simple, and insightful.

Creativity is always appreciated – show us what makes you different!
