## ARIA TRADING SIGNAL BOT/AI –

DAY 2 – 10/03/2025

Awesome Akshay 👏 — huge progress so far! Let me give you a **complete, in-depth recap** of everything we've built together for **Aria Trading Signal OCR System**, including every technical and strategic decision we’ve made — no matter how small.

**✅ 🧠 Project Name: Aria Trade Signal OCR**

**🎯 Goal:**

To create a lightweight, real-time trading assistant that:

* Reads your **live Zerodha/TradingView chart** from screen
* Analyzes **option chain data from NSE**
* Provides **minute-level trade signals** with **AI scoring**, Telegram alerts, and a beautiful UI dashboard
* Helps **low-budget traders (₹2k base)** find scalping entries
* Operates **without any API**, powered by **screen OCR and web scraping**
* Runs **locally and securely**, supporting both **Windows (GTX 1050 Ti)** and **Mac (M2)**

**✅ 📦 TECH STACK & MODULES STRUCTURE**

| **Component** | **Description** |
| --- | --- |
| **OCR** | pytesseract + mss reads selected screen region every second |
| **AI** | Optional scoring via **Ollama + Mistral model (4.1GB)** |
| **Option Chain Scraper** | NSE data scraped live using requests + session headers |
| **Signal Generator** | Basic filters or AI-verification used to score good options |
| **Trade Logger** | Saves each recommendation with confidence, comment, and timestamp |
| **UI (Flask)** | Dashboard with real-time updates, confidence heatmap, trend placeholder |
| **Replay Viewer** | Image history for past 1 hour auto-cleared |
| **Telegram Bot** | Sends trade calls instantly to mobile |
| **Backtesting (soon)** | Mock replays + P&L simulation in local memory |
| **Trend Engine (soon)** | Strategy module for Supertrend/EMA combo |
| **Dual-mode screen selector** | Detects **portrait screen 2** automatically if available |

**✅ 🔧 DEV TOOLS & USER EXPERIENCE**

| **Feature** | **Description** |
| --- | --- |
| ✅ Region selector | region\_selector.py opens a translucent window to select screen area |
| ✅ Multi-screen capture | Uses screeninfo to always prioritize **screen 2 (portrait)** |
| ✅ Platform support | Mac (brew, M2 chip) and Windows (venv, GitHub Desktop) support |
| ✅ Easy Git Sync | All code works well with GitHub Desktop flow |
| ✅ Configurable config.py | User can change screen region, premium limits, Tesseract path, etc. |
| ✅ Adjustable investment | ₹2000 set with PER\_TRADE = DAILY\_INVEST / 2 logic |
| ✅ Clean temp folder | /temp/ocr\_images auto-cleans after 3600 snapshots (~1 hour) |
| ✅ Cross-platform Tesseract | Installed via brew on Mac, .exe on Windows with path set |

**✅ 🖥️ UI Features (Current Dashboard)**

| **Section** | **What it Shows** |
| --- | --- |
| 📷 OCR Snapshot | Latest captured screen image with timestamp |
| 💰 Investment | Daily budget, amount used/remaining |
| 🔍 AI Signals | Parsed trade suggestions with color-coded confidence |
| 📈 Simulated P&L (soon) | Win rate and profit per day |
| 📉 Trend Engine (soon) | Visual chart + trend prediction block |
| 🕒 Replay Viewer | Scrollable thumbnail previews of hourly OCRs |
| 🌓 Dark/Light Mode | Toggle theme easily with persistent localStorage |
| ⏱️ Countdown | Shows time till next scan, auto-refreshes every second |
| 📲 Telegram | Push alerts for signals with emoji/markup |

**✅ 🤖 AI STRATEGY INTEGRATION**

| **What** | **How it works** |
| --- | --- |
| **Ollama AI** | Verifies options using natural language and returns JSON response |
| **Fallback** | If AI is not found, uses basic filter (e.g., premium range, volume) |
| **Modular AI** | ai\_verifier.py checks AI, handles subprocess, JSON output parsing |
| **Offline safe** | Ollama not required, system works with or without it |

**✅ MOCK TRADING & BACKTEST (Coming Up)**

| **Module** | **Purpose** |
| --- | --- |
| **Simulated P&L** | Log which trades worked based on strike/premium movement |
| **Trade replay** | Build mock trades from past signals to test win rate |
| **Backtest** | Optionally run through saved OCR + option chains and simulate |

**✅ STRATEGY ENGINE (Next Block)**

We'll add:

* 🔄 **EMA 9/21/50** Crossover for trend strength
* 📉 **Supertrend** for entry confirmation
* ⚖️ **RSI** to avoid overbought traps
* 💥 Configurable stoploss/take-profit per trade
* 💡 Used for either **signal enhancement** or **AI override fallback**

**✅ What's Working Right Now:**

* 🟢 Fully working screen OCR every second
* 🟢 Option chain data scraped in real-time
* 🟢 Telegram alerts configured
* 🟢 Live Flask dashboard running and responsive
* 🟢 Replay viewer shows recent chart views
* 🟢 Logs being created for each signal
* 🟢 Works cross-platform with no hard dependencies
* 🟢 AI scoring integrated if Ollama is detected

**🧠 Next Steps**

Here’s what I suggest we do now:

| **Step** | **Module** | **Priority** |
| --- | --- | --- |
| ✅ 1 | trend\_analysis.py for Supertrend + EMA logic | 🔥 HIGH |
| ✅ 2 | pnl\_tracker.py — Simulated P&L log | 🔥 HIGH |
| ✅ 3 | UI section to show live chart (trend overlay or simulated returns) | MEDIUM |
| ✅ 4 | Allow pause/resume of main loop from UI | LOW |
| ✅ 5 | Allow daily investment set from UI (input box or config edit) | LOW |

Would you like me to start with **Supertrend + EMA logic** as a new module?

I’ll include:

* trend\_analysis.py
* config.py updates
* main.py enhancement
* Possibly an indicators.py helper file too

Just confirm and I’ll drop the full set in one go!