AI Trader / Journal — Execution Plan & Templates

Date: 2025-10-01

# AI Trader / Journal — Execution Plan & Templates (2025-10-01) ## Objectives - Reboot the codebase with professional structure and quality gates. - Ship via small PRs with test coverage and explicit change control. - Reduce regression risk when chat context resets. ## Branching & Flow - `main` = stable - `feat/<slug>`, `fix/<slug>`, `docs/<slug>` - PRs must include tests and CHANGELOG updates. ## Required Return Format (from ChatGPT) 1) Rationale 2) Unified diff patch for edited files 3) New files as separate blocks 4) Test plan & commands 5) Migration notes ## Change Request Template (you paste in chat) ``` PROJECT: AI Trader / Journal App STACK: Python, Streamlit, yfinance, SQLite, pytest ENTRY: app.py (or ./src/app.py) GOAL: <what you want> CONTEXT: <recent behavior / errors> RETURN FORMAT REQUIRED: Unified diff patch + new files as full blocks. Include test updates. ``` ## Git & Patch Application (PowerShell) ``` ni changes.patch -ItemType File -Force | Out-Null Set-Content -Path .\changes.patch -Value @' <PASTE PATCH HERE> '@ git checkout -b feat/<slug> git apply --check .\changes.patch git apply .\changes.patch pytest -q streamlit run app.py git add -A && git commit -m "feat: <summary>" && git push -u origin HEAD ``` ## PR Checklist - [ ] Scope is one logical change - [ ] CI/tests pass - [ ] Backward compatible (or migration notes) - [ ] Screens/GIF for UI changes - [ ] CHANGELOG updated ## Issue Templates See `.github/ISSUE\_TEMPLATE/` in repo for Bug/Feature templates. ## Next Steps - Init Git repo, push to GitHub - Enable pre-commit hooks - Add GitHub Actions for lint/test - Build Journal UI CRUD - Add caching and rate-limit-aware fetchers