Mini Coding Agent — Work Summary

SanyamBK – assisted by GitHub Copilot

September 6, 2025

Overview

This document summarizes the changes implemented while working on the "Mini Coding Agent" repository. The goal was to inspect the projects under the `projects/` directory, implement missing functionality, and make the helper scripts robust for local usage. All work was coordinated in a branch and merged to `master` after validation.

High-level changes

- flask-easy: Implemented REST endpoints and fixed SQLAlchemy deprecation issues; added Flask CLI test helper.
- flask-intermediate: Implemented JWT login and protected user listing endpoints.
- flask-hard: Implemented Pydantic model for logs, a threaded LogProcessor with priority queues, metrics and notification manager, and REST endpoints for logs and metrics.
- Helpers: Hardened `agent.py` (dry-run mode) and $\texttt{check}_u sage.py(envvalidation, saferHTTPcalls)$.
- `Version control: Removed local venv from git, added it to .gitignore, and created a feature branch chore/harden-scripts before merging.`

Files changed (non-exhaustive)

- `agent.py`: added a minimal analyze() method that discovers projects and runs pytest where present; added dry-run and removed dotenv reliance per contest rules.
- $\texttt{check}_u sage.py : validatedenvironmentvariablesandmaderequestsrobust(note : laterrevertedtomatchremot$
- $\texttt{projects/flask-hard/app/log}_p rocessor.py : addedLogProcessorimplementation.$
- `projects/flask-hard/app/views.py: added endpoints for POST /logs, GET /logs, GET /metrics.`
- `Multiple tests fixed and confirmed green for all three projects.`

Test results

All tests passed on the remote CI (Hackerrank):

- **flask-easy**: 14 passed, 1 deprecation warning.
- **flask-intermediate**: 4 passed.
- **flask-hard**: 8 passed.

Commands used

```
# run project tests
flask --app manage.py test    # per-project in flask apps
python tests.py               # run top-level tests

# git housekeeping
git rm -r --cached venv
# ignore .env and venv
# commit and push feature branch
git checkout -b chore/harden-scripts
git add -A
git commit -m "chore: harden helper scripts; fix projects"
git push -u origin chore/harden-scripts

# Push to your own repository (example)
\begin{verbatim}
git remote add origin https://github.com/SanyamBK/Coding-Agent.git
git branch -M main
git push -u origin main
```

Notes and follow-ups

- Avoid committing virtual environments and secrets like `.env`. Use CI secrets or environment variables instead.
- Consider adding small unit tests for the LogProcessor consumer logic where possible.
- The repository enforces a pre-receive hook protecting $\texttt{check}_{u}sage.py; followreporuleswhenmodifyingsuchpr$

Tools and actions

This section lists the primary tools, libraries, and key actions performed while implementing the projects and agent.

- Tools: Python 3.11+, pytest, Flask, Flask-SQLAlchemy, Pydantic, MiKTeX (pdflatex) for PDF generation, PowerShell for local shell actions.
- Libraries used in projects: Flask, Flask-SQLAlchemy, pydantic, flask-jwt-extended, requests.

- Key actions: Implemented missing REST endpoints, fixed SQLAlchemy/Pydantic deprecations, implemented threaded LogProcessor with priority queues, added Metrics and NotificationManager, hardened ‘agent.py‘ with ‘–dry-run‘ and minimal ‘analyze()‘ runner, removed ‘venv‘ from git, and validated tests locally and on CI.

Agent implemented

The agent implemented for this repository provides a lightweight automated workflow to inspect projects, implement missing code, and run tests. Key capabilities:

- Discover subprojects under ‘projects/‘ and run pytest where ‘tests/‘ exists.
- Provide a ‘–dry-run‘ mode that prints planned actions without executing them.
- Minimal error handling and startup checks (missing env vars, helpful messages).

Report compiled on September 6, 2025.