

AI Usage DISCLOSURE FORM

1. Team Details

Team Name: 404-found

Project / Product Name: NightOwl Agent

Organization / Institution (if any): RV College of Engineering

Submission Date: 8th May 2026

2. AI Usage Declaration

Did your team use any Artificial Intelligence (AI) in developing this project? Yes / No

YES

3. Purpose of AI Usage (Brief Details)

Idea generation / brainstorming: AI helped explore architecture options (out-of-process vs in-process multi-agent, polling vs webhooks, label-based vs state-based eligibility). The developer made the final calls.

Code generation or assistance: Drafted the BoardAdapter implementations (Linear GraphQL, GitHub gh CLI), the per-issue runner, the watcher orchestrator. Each diff reviewed and committed by the developer.

UI / UX design: NightOwl has no UI surface of its own — the “UX” is Telegram messages and Linear ticket comments. The wording of those messages was developer-authored with light AI editorial polish.

Content creation: README, architecture documentation, board-watcher design doc, test plan, demo script, this disclosure form — all drafted with AI assistance and edited by the developer.

Data analysis: Not applicable to this project.

Testing / debugging: Used AI to diagnose adapter bugs (e.g. a gh CLI repo-path mismatch, a claude --print argument-parsing issue, a systemd \$HOME env-propagation issue).

Other: NightOwl is an AI agent. Claude Opus 4.5 + GPT-5.5 + Whisper are part of the product, invoked on every ticket the agent processes.

4. Feature Origin Classification (Please add/delete based on the number of the feature)

1. Feature Name Telegram → PR Flow

2. Self-Generated/AI-Generated/Both Both

3. Description (include AI tools/platform Used, prompt used , Output Summary, Modification) OpenClaw's existing Telegram channel plugin handles message ingestion. The developer designed the agent persona (IDENTITY, SOUL, USER, AGENTS, TOOLS markdown files) and the three skill files (feature-flow, pr-review, security-review). Claude Code (using its cli) is invoked by openclaw at runtime to generate the actual code diffs.

AI tools / platforms used: GPT-5.5 through WARP agentic terminal (in development, for drafting skill markdown), GPT 5.5 & Claude Code Opus 4.5 (used by openclaw at runtime).

Prompts used (development): Open-ended discussion of skill structure, e.g. "Write a SKILL.md for a Telegram intent that maps to: clone, branch, edit, run tests, push, open PR, reply" - followed by extensive iteration on flow control, error handling, and tone.

Output summary: Skill files (~150 lines each markdown) describing the agent's expected behaviour. Runtime Claude calls produce the per-task diffs.

Modifications: Every skill markdown was reviewed and edited by the developer at least once before commit.

1. Feature Name Board Watcher (Linear + GitHub Issues)

2. Self-Generated/AI-Generated/Both Both

3. Description (include AI tools/platform Used, prompt used , Output Summary, Modification) Polling loop with bounded concurrency, label-based claim/release, and a pluggable BoardAdapter protocol. Architecture (concurrency model, lock files, restart recovery, failure handling) was designed by the developer, drawing inspiration from OpenAI Symphony's WORKFLOW.md pattern. The Python implementation of the orchestrator and the two adapters was drafted with Claude assistance and reviewed by the developer.

AI tools / platforms used: Anthropic Claude Code (in development).

Prompts used (development): "Implement a Python BoardAdapter for Linear that takes JSON config on stdin, returns JSON on stdout, supports list/get/comment/transition. Use only the standard library." Iterative refinement on filter semantics, label-id resolution, and error handling.

Output summary: ~300 lines of Python per adapter, ~250 lines for the watcher orchestrator, ~250 lines for the per-issue runner. All in bin/.

Modifications: Multiple bugs caught in review (path mismatches, env-var propagation under systemd, claude-code stdin-vs-arg invocation). Re-architected the secrets loading path twice.

1. Feature Name PR Review and Security Review Skills

2. Self-Generated/AI-Generated/Both Both

3. Description (include AI tools/platform Used, prompt used , Output Summary, Modification) Two read-only skills that OpenClaw executes at runtime — structured PR review with line comments + verdict, and OWASP-style security audit posting findings as a GitHub issue. Skill markdown files developer-authored with AI assistance; runtime analysis by OpenClaw(using gpt5.5 and claude code cli).

...

5. Ethical & Compliance Confirmation

AI usage complies with guidelines and policies. **Yes**

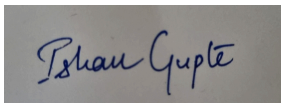
No proprietary or copyrighted data misused. **I Agree**

6. Declaration & Sign-Off

Name of Team Representative: Ishan Gupta

Role: Devleloper

Signature:



Date: 8th May 2026