Project Constitution — PMO RAID Register

> The non-negotiable principles that govern every plan, task, and line of code in this project. The AI coding agent MUST adhere to these articles. Any plan that violates an article must be revised before implementation. When an article and a feature request conflict, the article wins until formally amended below.

Version: 1.0.0 **Ratified:** 2026-06-14 **Author:** Apurv Duhan, PMP

Article I — Spec-First & Traceability (NON-NEGOTIABLE)

No code is written without a corresponding requirement in `spec.md`. Every functional requirement carries a stable ID (FR-XXX). Every task and every test references the requirement ID it satisfies. The result is a continuous trace from intent → spec → task → test, reconstructable at any time. If a request has no requirement, the spec is amended first.

Article II — Auditability of Data Changes (NON-NEGOTIABLE)

Every create, update, or delete to a RAID item MUST be recorded in an append-only audit log capturing: timestamp (UTC), actor, item ID, field changed, old value, new value. Audit records are never editable or deletable through the application. Absence of an audit trail for any data mutation is a defect.

Article III — Test-First

Acceptance scenarios in `spec.md` (Given/When/Then) are translated into automated tests before the implementing code exists. A feature is "done" only when its referenced tests pass. No requirement ships without at least one test mapped to its FR ID.

Article IV — Simplicity & No Premature Architecture

Start with the simplest design that satisfies the spec. No microservices, no message queues, no caching layer, no third-party dependency unless a requirement demonstrably needs it. Every added dependency must be justified in the plan against this article.

Article V — Data Privacy & Minimization

Collect only the data the spec requires. No personally identifiable information beyond a display name and email for the item owner. No analytics, telemetry, or third-party trackers. Data stays within the application's own datastore.

Article VI — Security Baseline

No secrets, keys, tokens, or credentials in source code or commit history — environment variables only. All user input is validated and treated as untrusted. The application performs no destructive action without explicit user confirmation in the UI.

Article VII — Accessibility

The interface meets WCAG 2.1 AA: full keyboard operability, visible focus states, sufficient color contrast, and labels on all interactive elements. RAID status MUST NOT be conveyed by color alone (use text/icon + color).

Article VIII — Observability

The application logs meaningful events (startup, errors, data mutations) in a structured, machine-readable format. Errors surface a clear, non-technical message to the user and a detailed entry to the log. Silent failures are prohibited.

Article IX — Plain-Language Stakeholder Readability

`spec.md` MUST remain readable by a non-technical PMO stakeholder. Implementation detail belongs in the plan, never the spec. If a stakeholder cannot understand what the product does from the spec alone, the spec has failed.

Governance

- This constitution supersedes all other practices. Plans are checked against every article during the `/analyze` quality gate.
- **Amendments** require: a documented rationale, a version bump (semantic), and an updated ratification date. Amendments are committed as their own change so the governance history is itself auditable.
- Reviewers (human-in-the-loop) verify article compliance at plan review — *before* implementation, not after.

Compliance is not a final checklist; it is the condition under which the agent is permitted to work.
