FSIPS — Logging & State Model

Version 0.1

This document defines the **minimum viable state and logging model** required to support deterministic progression, fatigue management, and long-term coverage.

The goal is to maximize signal while minimizing daily friction.

---

1. Design Goals

- Logging must be fast and lightweight
- State updates must be monotonic and conservative
- The system must function even with sparse data

Perfect data is not required.

---

2. Core Persistent State

ExerciseState

Tracked per canonical exercise:

- last_practiced_date
- rolling_minutes_7d
- rolling_minutes_28d
- mastery_estimate (0.0–1.0, coarse)
- last_fatigue_profile

This is sufficient for spacing, coverage, and progression.

---

3. Optional Variant State (Deferred)

Variant-level tracking may be added later:

- variant signature (overload hash)
- last_used_date
- success score

This is explicitly non-required at v0.1.

---

4. Daily Logging Surface

Recommended minimal per-block logging:

- completed: yes / partial / no
- quality: clean / acceptable / sloppy
- optional note (free text)

Numeric metrics are optional.

---

5. Mastery Estimate Updates

Mastery estimates: - increase slowly on clean completion - hold on acceptable completion - decrease slightly on repeated failure

Estimates are intentionally fuzzy.

---

6. Failure & Regression Handling

If an exercise repeatedly fails: - overload progression halts - generator biases toward regression variants

Failure is treated as data, not error.

---

7. Cold Start Behavior

With no historical data: - conservative overloads - avoidance of F2 stacking - preference for fundamental exercises

System remains fully functional.

Status

Logging and state model frozen at v0.1