Criterion C - Development Narrative

Overview

I structured the development log around the three most interesting technical techniques: (1) persistent JSON da

1. JSON Datastore with Idempotent Upserts (‘src/datastore.py‘)

The client wanted the freedom to edit saved plans manually, so SQLite or Firebase felt excessive. Instead, ‘_rea

2. Algorithmic Filtering and Scoring (‘src/generator.py‘)

The generator is intentionally modular. ‘_filter_exercises‘ enforces three independent constraints: avoidance list

The ‘_score‘ function embodies algorithmic thinking: it starts at 100, subtracts points for any movement pattern w

During development I profiled different penalty weights by printing debug tables and discovered that a 10-point

3. Plan Assembly and Goal-Specific Formatting

‘generate_plan‘ starts by mapping ‘days_per_week‘ to an evidence-based split (‘SPLITS‘). Within every day loop

The accessory loop monitors both time (‘session_minutes‘) and maximum exercise count. I subtract ten minutes

4. User Interface Binding (‘src/ui_cli.py‘ and ‘src/main.py‘)

I kept the CLI minimal but resilient by wrapping list inputs (‘equipment‘, ‘avoid‘) in the ‘prompt_list‘ helper so blan

5. Testing Hooks

Although formal unit tests are TBD, I instrumented the generator by running ‘python3 -m src.main‘ with sample d

Source Attribution

No external code snippets were copied. The only references consulted were the Python 3.9 documentation for ‘