

# Changelog

All notable changes to this project are documented in this file. Version numbers follow semantic versioning and dates use `YYYY-MM-DD`.

## [1.0.1] - 2026-01-01

### Added

- Support for shared balancing quotas for specified postings (e.g., GRM and MedComm at TTSH).
- SR preference mapping to ensure residents are allocated to their preferred SR supervisor.
- Multi-select dropdown for postings with balancing deviations.
- Button to set default deviations for ED, GRM, and GM postings.

### Changed

- Redesigned CSV ingestion workflow for improved validation and error handling.
- Removed the now-redundant Optimisation Scorecard from the UI.
- Updated documentation for constraints and developer onboarding.

### Fixed

- Resolved a bug in post-processing that affected timetable accuracy.
- Corrected logic to exclude residents on leave from the assigned block count.
- Addressed an issue with the end-of-year career-track constraint.

## [1.0.0] - 2025-12-16

### Added

- Residency Rotation Scheduler initial release with constraint-based timetable optimisation powered by Google OR-Tools CP-SAT.
- CSV ingestion and validation for residents, resident history, resident preferences, SR preferences, postings, and optional resident leaves, plus weightages, pinned assignments, and a solver time-limit override.
- FastAPI backend exposing `/api/solve`, `/api/save`, and `/api/download-csv`, with in-memory dataset caching to support iterative solves and current-year edits.
- React + Vite + Tailwind frontend for uploading CSVs, tuning weightages, pinning assignments, running the solver, reviewing timetables/cohort statistics, and exporting the final timetable; includes a sample CSV generator for quick smoke tests.
- Constraint documentation and developer handoff guidance covering the hard/soft rules (see `constraints.md` and `DEVELOPER_GUIDE.md`), plus notes for hosted deployments (stateless flows with optional PostgreSQL session persistence).

### Known issues

- No automated backend test suite yet; validate changes manually with representative CSVs.
- Local in-memory cache resets when the API process restarts; `/api/save` requires a prior solve/upload within the same process.
- Solver runtime scales with dataset size; use `max_time_in_minutes` when iterating or debugging feasibility.