Codora Timetable — Product Requirements Document (PRD)

1. Overview

Product Name

Codora Timetable

One-line Description

Codora Timetable is a system that generates **conflict-free college timetables** and safely handles changes using **minimal auto-repair with clear explanations**.

Core Promise

> "Generate a correct timetable, handle real-world changes safely, and always explain what changed and why."

This is **not** an "AI timetable generator".
Correctness and trust come before automation.

---

2. Problem Statement

College timetables are hard to manage because:

- many constraints exist (faculty, rooms, capacity, shifts)
- small changes cause large disruptions
- most systems require full regeneration
- decisions are not explainable to HODs or faculty

Manual handling leads to:

- errors
- repeated rework
- loss of trust

Codora Timetable solves this by combining:

- deterministic scheduling
- incremental conflict repair
- human-in-the-loop approvals

---

3. Target Users & Roles

Primary Users

- **Admin** (Academic Office): uploads data, generates and publishes timetables
- **HOD**: reviews changes, approves repairs
- **Class Advisor**: proposes edits or change requests

Secondary Users

- **Faculty**: views personal timetable and receives change notifications

---

4. MVP Scope (What We ARE Building)

4.1 Inputs

Admins can upload structured data:

- Sections & student strength
- Faculty & availability
- Courses & weekly load
- Rooms & capacity/type
- Time configuration (shifts, periods, recess)

4.2 Core Capabilities

- Generate a **conflict-free weekly timetable**
- Ensure:
  - no double booking (faculty / room / section)
  - room capacity ≥ section strength
  - room type compatibility (lab/lecture)
  - shift timing & recess rules
- View timetables:
  - section-wise
  - faculty-wise
  - room-wise

4.3 Manual Changes (Controlled)

- Authorized users can propose changes:
  - move class
  - change room
  - change faculty
- System detects conflicts immediately

4.4 Auto-Repair (Key Differentiator)

When conflicts occur, the system:

- freezes unaffected parts of the timetable
- re-solves only the impacted portion
- minimizes number of changes

- presents repair options for approval

4.5 Explainability & Trust

For every change:

- system explains **what changed**
- explains **why it changed**
- lists **who is affected**

4.6 Versioning

- Every generation or repair creates a new version
- Versions can be:
  - reviewed
  - published
  - rolled back

---

5. Explicit Non-Goals (MVP)

The MVP will **NOT** include:

- student-facing mobile apps
- real-time live scheduling
- substitute teacher optimization
- preference-based optimization (fairness, gaps)
- ERP/SIS integrations
- cross-university scheduling

If it’s not listed in the MVP scope, it is **out of scope**.

---

6. Phases (High-Level Only)

Phase 1 — Foundation

Generate a correct timetable from uploads and display it (read-only).

Phase 2 — Auto-Repair Core

Support controlled edits and minimal-disruption repairs with explanations.

Phase 3 — Trust & Publish

Versioning, rollback, approvals, and notifications.

---

7. Design Principles (Non-Negotiable)

- **Correctness > Cleverness**

- **Solver decides schedules, not AI**
- **No silent changes**
- **Every change must be explainable**
- **Human approval is required before publish**

---

8. Definition of Success (MVP)

The MVP is successful if:

- real college data produces a valid timetable
- a single change does not break the whole schedule
- admins understand why changes happened
- published timetables can be rolled back safely

---

9. Ownership & Scope Control

- This PRD defines **what is being built**
- GitHub Issues define **what is being worked on**
- Research and architecture docs are **reference only**
- Any feature outside this PRD requires explicit approval

---

10. Final Note

Codora Timetable is built to **reduce human stress**, not replace humans.

Trust, correctness, and clarity come first.
Automation comes second.