

Replit Agent Prompt: Enhance AI Generation & Reporting for Classroom Solutions

You are enhancing the **Classroom Solutions** function of the C2C app.

This must be implemented behind a **feature flag** (`classroom_solutions_enabled`) so it does not disrupt existing subscription workflows.

Requirements

1. Submission Handling

- Teachers access public submission form via QR code:
 - Collects required fields (anonymized student info, request type, severity, concerns, actions taken).
 - Teachers explicitly select **Differentiation OR Tier 2 Intervention** task.
 - On submit:
 - Validate teacher against enrolled list (max 5 requests per month, auto-reset monthly).
 - Save submission record in database with `status: pending`.
 - Trigger **AI draft generation** immediately via DeepSeek API.
 - Store AI draft with submission.
-

2. Delayed Delivery Logic

- After submission, teacher sees confirmation:

“Your personalized strategies are being prepared and will be delivered shortly.”
 - AI draft delivery is **delayed by 30 minutes** (configurable).
 - During delay:
 - Admin can view/edit submission in dashboard.
 - If admin edits → teacher receives **reviewed version with additional professional input**.
 - If no action → system auto-sends AI draft as “**AI Initial Guidance**.”
-

3. Urgent & Safeguard Flow

- Teacher selects **Severity**: Mild, Moderate, Urgent.
- If **Urgent**:
 - Teacher response includes:

“An urgent case requires rapid support. Initial strategies are provided now. **Please notify your student support department immediately.**”

- Admin receives **distinct urgent ping** and red flag in dashboard.
- If submission contains keywords (e.g., *suicide, self-harm, kill, harm others*):
 - Bypass delay — teacher immediately receives:

“URGENT: This case involves potential harm. **Consult your school’s child protection protocol immediately.** Initial strategies are provided here for reference only.”

- Admin instantly alerted with highest-priority ping.
 - Auto-sending of full draft is disabled until admin manually approves.
-

4. Universal Disclaimer

Every output (AI draft or reviewed version) must include this disclaimer at the bottom:

Disclaimer: The strategies provided here are **AI-assisted suggestions** intended to support your professional practice. They are not a substitute for your school’s policies, student support team, or child protection protocols.

- For Mild/Moderate concerns: Please review and adapt recommendations according to your classroom and school context.
 - For Urgent concerns: Notify your student support department immediately.
 - For any indication of suicide, self-harm, or harm to others: **Consult your school’s child protection protocol immediately.**
-

5. Admin Dashboard Enhancements

Add a **Submissions Review** section under “Classroom Solutions”:

- List all submissions with filters: Pending, Reviewed, Urgent, Completed.
- Show teacher info, anonymized student profile, concerns, severity, AI draft.
- Admin actions:
 - **Edit & Send** (customize response)

- **Send As-Is** (AI Initial Guidance)
 - **Request More Info**
 - **Escalate / Hold** (prevent auto-send)
 - Urgent cases always appear at top with red badge.
-

6. Reporting & Analytics

Enhance reporting module to track Classroom Solutions:

- Submissions per teacher per month.
 - Request type distribution (Differentiation vs. Tier 2).
 - Severity levels distribution.
 - Urgent flag counts.
 - Average admin review time vs. auto-send rate.
-

Technical Notes

- Keep all functions gated under `classroom_solutions_enabled`.
 - Ensure all teacher usage resets on the 1st of each month (5 requests default).
 - Build in **configurable delay** (default 30 minutes).
 - Use webhooks or socket pings for admin notifications (distinct sound for urgent).
 - Ensure compliance with student privacy → no full names stored, only age/grade + initials.
-

Deliverables:

- Updated backend endpoints for submission handling, AI generation, urgent keyword scanning, and delayed delivery.
- Admin dashboard with Submissions Review UI.
- Automated email delivery to teachers with disclaimer + severity-specific messaging.
- Reporting dashboard updates for Classroom Solutions analytics.

Database Schema Changes for Classroom Solutions

1. classroom_submissions (new table)

Holds every teacher submission.

Field	Type	Notes
id	UUID (PK)	Unique submission ID
teacher_id	UUID (FK → teachers.id)	Who submitted
first_name	VARCHAR(50)	Anonymized student first name (initials allowed)
last_initial	CHAR(1)	Student last initial
student_age	INT	Required
student_grade	VARCHAR(10)	Required
task_type	ENUM(differentiation, tier2_intervention)	Required
learning_profile	JSON	Array of profile tags (IEP, EAL, Gifted, etc.)
concern_types	JSON	Array of selected concern categories
concern_description	TEXT	Teacher free text
severity_level	ENUM(mild, moderate, urgent)	Required
actions_taken	JSON	Array of teacher's actions
status	ENUM(pending, reviewed, auto_sent, urgent_flagged, completed)	Default = pending

Field	Type	Notes
ai_draft	TEXT	Auto-generated intervention text
reviewed_text	TEXT	Admin-updated intervention text
sent_text	TEXT	Final version sent to teacher
disclaimer_attached	BOOLEAN	Always TRUE for sent outputs
urgent_flag	BOOLEAN	Auto-set if keyword trigger or teacher marked urgent
created_at	TIMESTAMP	Submission time
auto_send_time	TIMESTAMP	When draft will be auto-sent (created_at + 30min by default)
sent_at	TIMESTAMP	When actually sent to teacher
admin_id	UUID (FK → admins.id)	Null if auto-sent

2. teacher_limits (new table or extend existing teachers)

Tracks teacher usage quota (5 per month).

Field	Type	Notes
teacher_id	UUID (PK, FK → teachers.id)	
month_year	VARCHAR(7)	Format YYYY-MM
submissions_used	INT	Default 0
limit	INT	Default 5 (configurable)

3. admin_notifications (optional helper table)

Stores pings for urgent cases & follow-ups.

Field	Type	Notes
id	UUID (PK)	
submission_id	UUID (FK → classroom_submissions.id)	Related case
admin_id	UUID (nullable)	If assigned
type	ENUM(urgent, reminder, followup)	
status	ENUM(unread, read, resolved)	
created_at	TIMESTAMP	

4. classroom_reports (aggregate table, optional but useful)

Pre-computed reporting for dashboards.

Field	Type	Notes
id	UUID (PK)	
month_year	VARCHAR(7)	Reporting cycle
total_submissions	INT	Count
differentiation_requests	INT	
tier2_requests	INT	
mild_cases	INT	
moderate_cases	INT	
urgent_cases	INT	
auto_sent_rate	FLOAT	%
avg_review_time_minutes	FLOAT	

Flow with Schema

1. Submission saved in **classroom_submissions**.
2. AI draft stored in **ai_draft**.
3. If admin edits → **reviewed_text** populated, **status** → reviewed.
4. Final sent version saved in **sent_text**, with disclaimer appended.
5. Teacher usage tracked in **teacher_limits**.
6. Urgent keyword triggers → **urgent_flag** = TRUE + record in **admin_notifications**.
7. Reports built from **classroom_reports** or direct aggregation.

```
-- =====
-- 1. classroom_submissions table
-- =====
CREATE TABLE classroom_submissions (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  teacher_id UUID NOT NULL REFERENCES teachers(id),
  first_name VARCHAR(50) NOT NULL,
  last_initial CHAR(1) NOT NULL,
  student_age INT NOT NULL,
  student_grade VARCHAR(10) NOT NULL,
  task_type VARCHAR(20) NOT NULL CHECK (task_type IN
('differentiation', 'tier2_intervention')),
  learning_profile JSONB,
  concern_types JSONB,
  concern_description TEXT,
  severity_level VARCHAR(10) NOT NULL CHECK (severity_level IN
('mild', 'moderate', 'urgent')),
  actions_taken JSONB,
  status VARCHAR(20) NOT NULL DEFAULT 'pending' CHECK (status IN
('pending', 'reviewed', 'auto_sent', 'urgent_flagged', 'completed')),
  ai_draft TEXT,
  reviewed_text TEXT,
  sent_text TEXT,
  disclaimer_attached BOOLEAN NOT NULL DEFAULT TRUE,
  urgent_flag BOOLEAN NOT NULL DEFAULT FALSE,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT now(),
  auto_send_time TIMESTAMP WITH TIME ZONE,
  sent_at TIMESTAMP WITH TIME ZONE,
  admin_id UUID REFERENCES admins(id)
);

-- Index for faster lookups by teacher
CREATE INDEX idx_classroom_submissions_teacher ON
classroom_submissions(teacher_id);

-- =====
-- 2. teacher_limits table
```

```

-- =====
CREATE TABLE teacher_limits (
    teacher_id UUID NOT NULL REFERENCES teachers(id),
    month_year VARCHAR(7) NOT NULL, -- Format YYYY-MM
    submissions_used INT NOT NULL DEFAULT 0,
    limit INT NOT NULL DEFAULT 5,
    PRIMARY KEY (teacher_id, month_year)
);

-- =====
-- 3. admin_notifications table
-- =====
CREATE TABLE admin_notifications (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    submission_id UUID NOT NULL REFERENCES classroom_submissions(id),
    admin_id UUID,
    type VARCHAR(20) NOT NULL CHECK (type IN
('urgent', 'reminder', 'followup')),
    status VARCHAR(20) NOT NULL DEFAULT 'unread' CHECK (status IN
('unread', 'read', 'resolved')),
    created_at TIMESTAMP WITH TIME ZONE DEFAULT now()
);

-- Index to quickly find unread urgent notifications
CREATE INDEX idx_admin_notifications_type_status
    ON admin_notifications(type, status);

-- =====
-- 4. classroom_reports table (optional)
-- =====
CREATE TABLE classroom_reports (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    month_year VARCHAR(7) NOT NULL,
    total_submissions INT NOT NULL DEFAULT 0,
    differentiation_requests INT NOT NULL DEFAULT 0,
    tier2_requests INT NOT NULL DEFAULT 0,
    mild_cases INT NOT NULL DEFAULT 0,
    moderate_cases INT NOT NULL DEFAULT 0,
    urgent_cases INT NOT NULL DEFAULT 0,
    auto_sent_rate FLOAT DEFAULT 0.0,
    avg_review_time_minutes FLOAT DEFAULT 0.0
);

-- Index for fast reporting by month
CREATE INDEX idx_classroom_reports_month ON classroom_reports(month_year);

```

Notes:

1. classroom_submissions.auto_send_time is calculated at submission (created_at + 30 minutes **default**).
2. urgent_flag is set automatically by keyword detection or teacher severity = Urgent.
3. teacher_limits tracks monthly quotas; resets should be automated at start of month.
4. admin_notifications supports real-time alerts for urgent and follow-up cases.

5. `classroom_reports` can be built as a **materialized view** or updated via scheduled jobs.