

Civilian Management

Problem

- No payment system → civilians can spam reservations
- Pharmacies waste time + stock
- System credibility drops

Solution

- Track **uncollected reservations**
- Automatically ban abusers
- Allow limited appeals
- Admin decides final outcome

This is exactly how **real government systems work**.

Core Rules (Very Important)

Temporary Ban Rules (Enhanced)

A civilian account can be:

- **Temporarily banned at most 2 times**
- **Appeal at most 2 times (total lifetime)**

After that → **PERMANENT BAN** automatically

Temporary Ban → Permanent Ban Triggers

An account becomes **PERMANENTLY BANNED** if **ANY ONE** of these happens:

1. **✗ No appeal submitted within 14 days**
2. **✗ Appeal rejected by Admin**
3. **✗ Temporary ban count exceeds 2**
4. **✗ Appeal count exceeds 2**

No third chance. No loop. No mercy. Clean and fair.

2 Updated Status Lifecycle (Full Flow)

```
ACTIVE
  ↓ (3 uncollected orders)
TEMPORARY_BANNED (Ban #1)
  ↓
Appeal #1
├─ Approved → ACTIVE
└─ Rejected → PERMANENTLY_BANNED
```

```
ACTIVE
  ↓ (again 3 uncollected orders)
TEMPORARY_BANNED (Ban #2)
  ↓
Appeal #2
├─ Approved → ACTIVE
└─ Rejected → PERMANENTLY_BANNED
```

```
TEMPORARY_BANNED (Ban #3 attempt)
  → AUTO PERMANENTLY_BANNED ✕
```

No Temporary Ban #3 exists.

3 Appeal Rules (Very Strict, Very Clean)

- Max appeals allowed: **2 (lifetime)**
- Appeal window: **14 days**
- Appeals are **read-only after submission**
- Admin decision is **final**

If appeal limit exceeded:

- Appeal button hidden
 - Status auto-updated to **PERMANENTLY_BANNED**
-

4 Permanent Ban — NEW Behavior (Your Additions)

This is where your idea gets 🔥

Permanently Banned Accounts

Even though the account is banned:

- ✓ **Admin & Super Admin can still view it**
 - ✓ **VIVO Manage button is available** (View-In-View-Only)
 - ✗ No activation
 - ✗ No appeal
 - ✗ No editing personal data
-

Data Sanitization (Manual Control)

Via **VIVO Manage**:

- Admin/Super Admin can:
 - Mask personal details (email, phone)
 - Disable login credentials
 - Anonymize sensitive data

This is **compliance-friendly** (privacy + audit).

5 **Auto Deletion Rule (3 Months)**

Permanent Ban Auto Cleanup

After **3 months** from `permanent_banned_at`:

- System automatically:
 - Deletes the civilian account
 - Deletes login credentials
 - Keeps **minimal audit log only** (ID + action history)

No manual delete button needed.

This protects:

- Storage
- Privacy laws
- System hygiene



Civilian Management Home Page

Purpose

Central control page for Admin & Super Admin to monitor civilians and their account states.

Access

Admin Panel → Civilian Management

Components

 *Metric Cards (Top)*

1. **Total Civilians**
2. **Active Civilians**
3. **Temporarily Banned Civilians**
4. **Permanently Banned Civilians**

👉 Clicking a card **filters the main table**

👉 Default view: **Temporarily Banned Civilians**

Main Civilian Table (Single Table, Filtered)

Columns:

- Civilian ID
- Name
- Email / Contact
- Account Status
- Temp Ban Count
- Appeal Count
- Last Action Date
- **Actions**

Actions per row:

- **View / Manage** (always visible)
- **Activate** (only for eligible temp-banned users)

- ☒ No separate tables
 - ☒ Filter controlled by metric cards (tabs behavior)
-

What Changed with Your Update

- Temp ban count is visible
 - Appeal count is visible
 - Permanently banned users are **still listed**
 - No manual delete option
-

2 Civilian Details Page

Purpose

Detailed inspection of a single civilian account.

Access

Civilian Management Table → **View / Manage**

Sections Included

 *Account Overview*

- Civilian ID
 - Name
 - Email
 - Contact
 - Account Status (Active / Temp Banned / Permanent Banned)
 - Temp Ban Count
 - Appeal Count
-

Ban Information (If Banned)

- Ban reason
 - Ban date
 - Appeal deadline (14 days)
 - Warning message if:
 - Last appeal
 - Final ban threshold reached
-

Action Buttons (Role + Status Based)

If Active

- No ban-related actions

If Temporarily Banned

- **View Appeal Details**
- **Back**

✗ No direct activate here

✗ Activation only via Appeal decision

If Permanently Banned

- **VIVO Manage** (View-In-View-Only)
- **Back**

✗ No activate

✗ No appeal

✗ No edit

What Changed

- Permanent ban is final
 - Appeal & ban counts visible
 - Countdown logic tied to backend
-

Appeal Details Page

Purpose

Admin decision page for a civilian's appeal.

Access

Civilian Details Page → **View Appeal**

Content

Appeal Information

- Appeal ID
 - Appeal Number (1 or 2)
 - Submitted date
 - Appeal message
 - Attachments (if any)
-

Civilian History Summary

- Total reservations
 - Uncollected orders
 - Temp ban count
 - Appeal count remaining
-

Action Buttons (Admin / Super Admin)

1. **Activate Account**
 2. **Permanently Ban Account**
 3. **Back**
-

What Changed

- If appeal count exceeds 2 → auto permanent ban
- If this is the **last appeal**, warning banner shown

- Decision is final
-

4 Permanently Banned Civilian Page (Filtered View)

This is NOT a separate page — it's a filtered state of the main table

Purpose

Audit & compliance view of permanently banned civilians.

Behavior

- Only **View / VIVO Manage** available
 - No activation
 - No appeal
-

Extra Info Shown

- Permanent ban date
 - Auto-delete countdown (e.g. "Deletes in 48 days")
-

5 Permanently deleted accounts details Page (View-In-View-Only)

Purpose

Safe inspection of permanently banned accounts without altering data.

Access

Civilian Details → **VIVO Manage**

What's Visible

- Masked personal data
 - Reservation history (read-only)
 - Ban & appeal history
 - System actions log
-

What's Allowed

- Disable login credentials
 - Anonymize sensitive fields (manual)
- ✗ No edits
- ✗ No activation
-

What Changed

- Added because of legal + privacy reasons
 - Supports 3-month retention policy
-

6 Civilian Reports & Inquiries Management Page

- **Purpose:** Allow Admin to monitor, manage, and filter all reports and inquiries submitted by civilians.
 - **Access:** Civilian Management Home Page → Click “Reports & Inquiries” button.
 - **Visible:**
 - Metric cards showing counts: Pending, Ongoing, Resolved, Rejected.
 - Main table with columns: Reference ID, Civilian Name, Type (Report / Inquiry), Title, Priority, Status, Date Submitted, Actions.
 - Actions column: **View-Manage** button only.
 - **Behavior:** Metric cards filter the table; resolved/rejected entries auto-delete after 60 days.
-

7 Reports & Inquiries Details Page

- **Purpose:** View full details of a single report or inquiry and manage its status.
- **Access:** Civilian Reports & Inquiries Management Table → Click **View-Manage**.
- **Visible:**
 - Report/Inquiries Information: Reference ID, Civilian Info, Type, Title, Description, Priority, Attachments, Status, Date Submitted.

- Admin Response Section: Textarea + optional file attachment.
 - Action buttons: **Mark In Progress, Resolve, Reject.**
- **Behavior:** Admin can update status; actions are logged; resolved/rejected entries auto-delete after 60 days.

8. Auto Deletion Process (No UI Page)

Purpose

System hygiene & legal compliance.

Trigger

- 3 months after permanent ban
-

Behavior

- Account is deleted automatically
- Login credentials removed
- Minimal audit log retained

✗ No admin button

✗ No reversal

Final Summary (Big Picture)

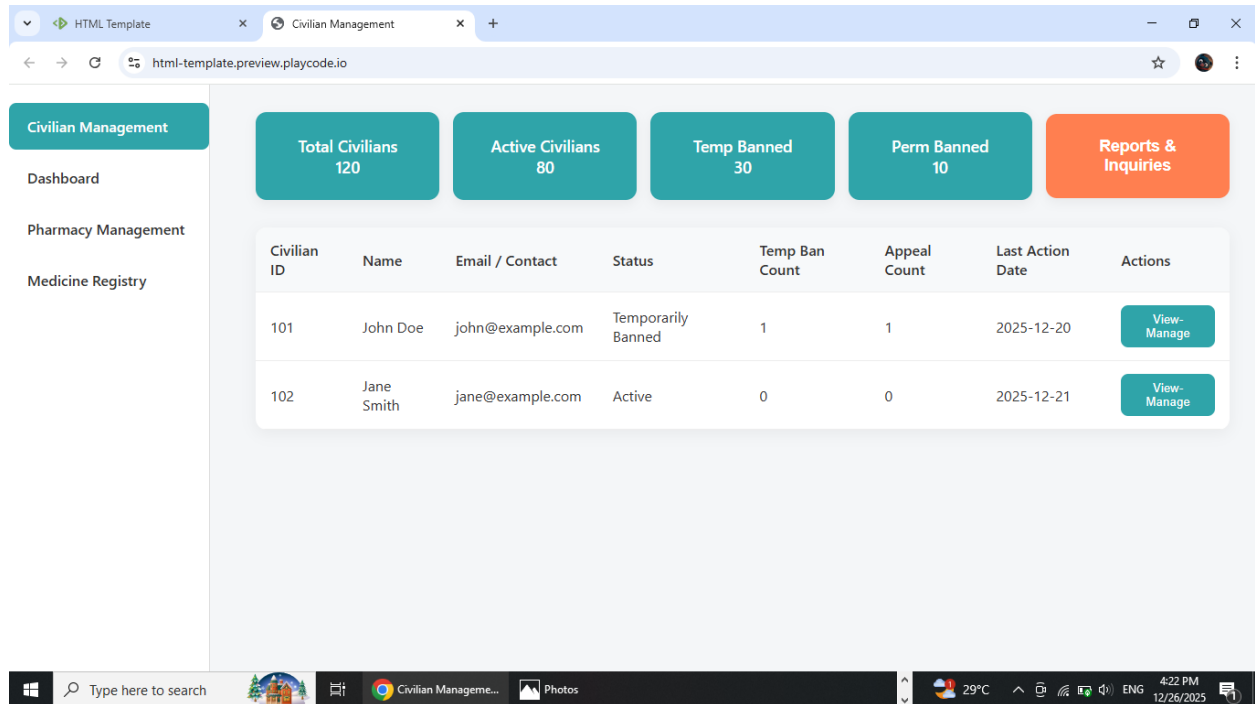
You now have:

- ✓ Clear user lifecycle
- ✓ Strict abuse prevention
- ✓ No infinite loops
- ✓ Automatic enforcement
- ✓ Admin workload control
- ✓ Legal & privacy compliance

This is **production-ready logic**, not just academic.

Details of all pages

1 Civilian Management Home Page



Access:

- Admin Panel → Civilian Management

Purpose:

- Central dashboard for monitoring civilians and their account statuses.
- Provides quick filtering and insights into active, temporarily banned, and permanently banned civilians.

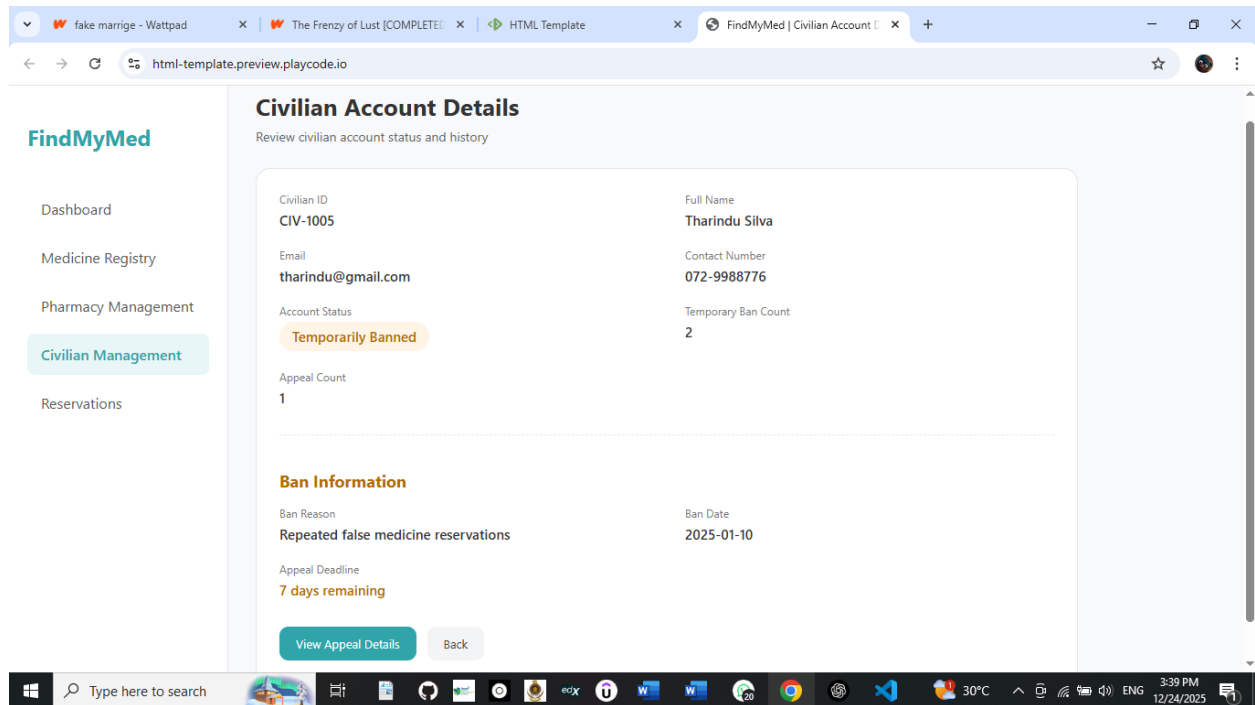
Behavior:

- Default table view: **Temporarily Banned Civilians**
- Clicking **metric cards** filters the main table by status
- Single table approach (no separate tables) with tabs-like behavior
- Only eligible temporary banned civilians can be activated via **Activate button** in the table row

Components & Contents (Bullet-wise):

- **Metric Cards (Top)**
 - Total Civilians
 - Active Civilians
 - Temporarily Banned Civilians
 - Permanently Banned Civilians
 - Reports and inquiries
 - Clicking each card filters the table below
- **Main Civilian Table**
 - Columns:
 - Civilian ID
 - Name
 - Email / Contact
 - Account Status (Active / Temp Banned / Permanent Banned)
 - Temp Ban Count
 - Appeal Count
 - Last Action Date
 - Actions
 - Actions per row:
 - View / Manage (always visible)
 - Activate (visible only for eligible temporary banned civilians)
- **Filtering & Sorting**
 - Table is automatically filtered based on metric card selection
 - Table updates dynamically when statuses change
- **Updates & Info**
 - Temp ban count and appeal count displayed
 - Permanently banned users remain listed for audit purposes

2 Civilian Details Page



Access:

- From **Civilian Management Table** → **View / Manage**

Purpose:

- Detailed inspection of a single civilian account
- View account history, ban info, and appeal deadlines

Behavior:

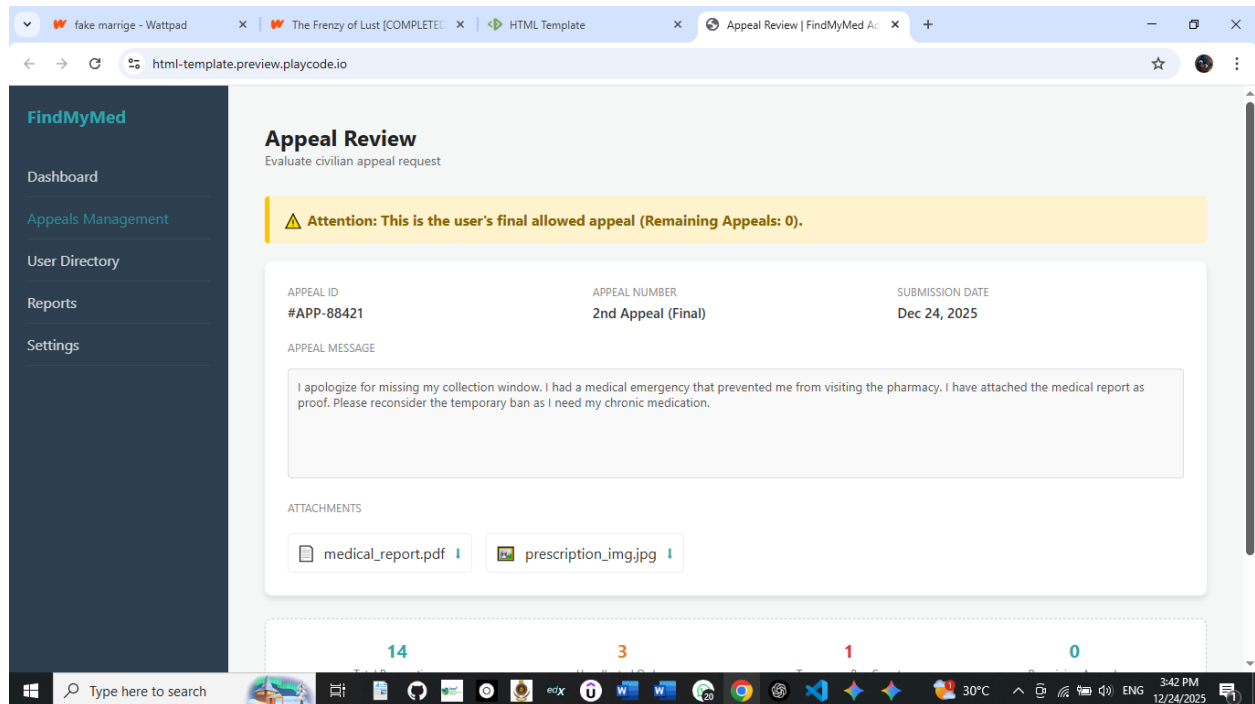
- Displays all core civilian details
- Shows ban info if account is banned
- Action buttons depend on account status:
 - Temp banned → only **View Appeal Details & Back**
 - Permanently banned → only **VIVO Manage & Back**
 - Active → no ban-related actions

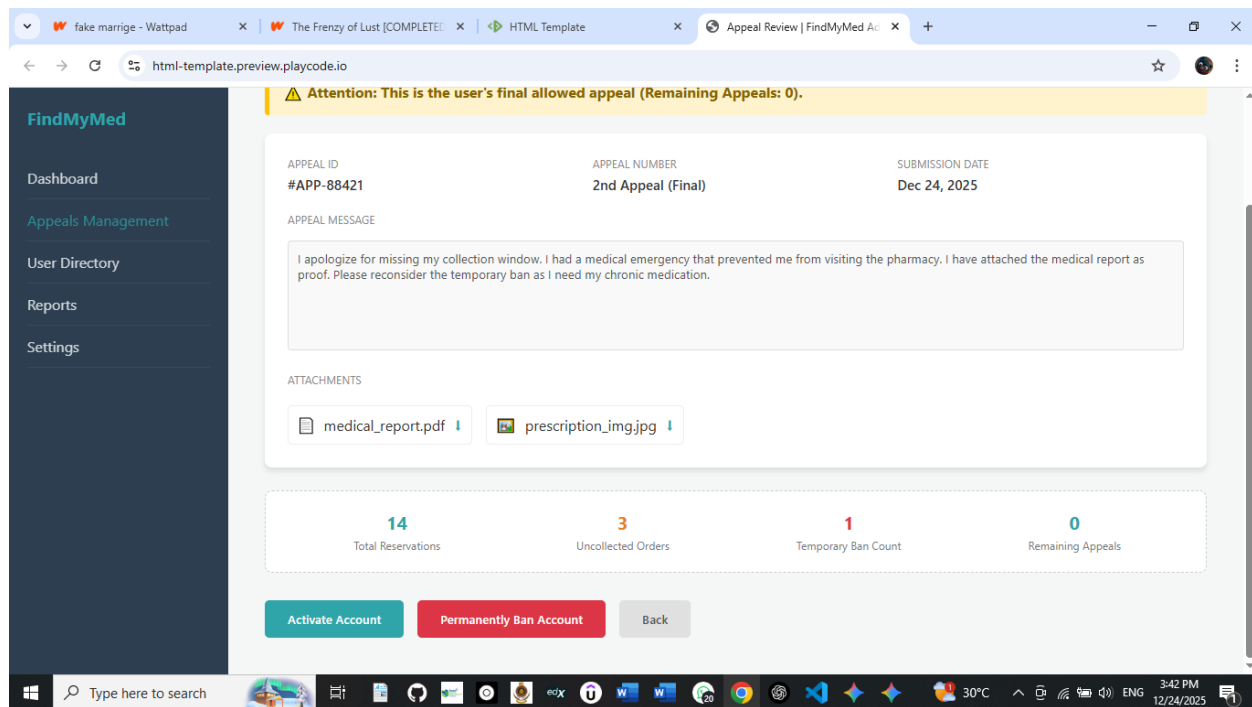
Components & Contents (Bullet-wise):

- **Account Overview**
 - Civilian ID
 - Name
 - Email
 - Contact
 - Account Status (Active / Temp Banned / Permanent Banned)

- Temp Ban Count
- Appeal Count
- **Ban Information (if banned)**
 - Ban Reason
 - Ban Date
 - Appeal Deadline (14-day countdown)
 - Warning if:
 - Last allowed appeal
 - Maximum temp bans reached
- **Action Buttons**
 - Temporarily Banned: View Appeal Details, Back
 - Permanently Banned: VIVO Manage, Back
 - Active: no ban actions
- **Notes**
 - Countdown logic handled by backend
 - No direct activation from this page

3 Appeal Details Page





Access:

- Civilian Details Page → **View Appeal Details**

Purpose:

- Admin / Super Admin evaluates civilian appeals
- Decide whether to activate account or permanently ban

Behavior:

- Shows appeal content and civilian history summary
- Buttons based on admin decision:
 - Activate Account
 - Permanently Ban Account
 - Back

Components & Contents (Bullet-wise):

- **Appeal Information**
 - Appeal ID
 - Appeal Number (1 or 2)
 - Submission Date
 - Appeal Message (read-only)
 - Attachments (downloadable)
- **Civilian History Summary**

- Total reservations
- Uncollected orders
- Temp Ban Count
- Remaining Appeals
- **Action Buttons**
 - Activate Account
 - Permanently Ban Account
 - Back
- **Special Notes**
 - Last allowed appeal triggers warning
 - Auto permanent ban if appeal count exceeds 2

4 Permanently Banned Civilian Page (Filtered Table)

The screenshot displays the 'Permanently Banned Civilians' page within the FindMyMate application. The browser window shows multiple tabs, including 'fake marriage - Wattpad', 'The Frenzy of Lust [COMPLETE]', 'HTML Template', and 'Permanently Banned Civilians'. The address bar indicates the URL 'html-template.preview.playcode.io'.

FindMyMate

Permanently Banned Civilians

Metric Cards:

- Total Civilians: 1,235
- Active Civilians: 985
- Temporary Banned: 120
- Permanently Banned: 130

Table of Permanently Banned Civilians:

Civilian ID	Name	Email	Status	Reason	Actions
CIV-00123	John Doe	john.doe@example.com	Permanently Banned	3 Uncollected Reservations	View Details
CIV-00124	Jane Smith	jane.smith@example.com	Permanently Banned	Exceeded Appeal Limit	View Details
CIV-00125	Mike Johnson	mike.johnson@example.com	Permanently Banned	Multiple Violations	View Details

Navigation Sidebar:

- Dashboard
- Medicine Registry
- Pharmacy Management
- Civilian Management**
- Reports & Inquiries
- Reservations

Windows Taskbar:

Search: Type here to search. System tray shows 30°C, 3:52 PM, 12/24/2025, and ENG.

Access:

- Civilian Management Home → Metric Card: Permanently Banned Civilians

Purpose:

- Audit & compliance view of permanently banned civilians
- Read-only actions

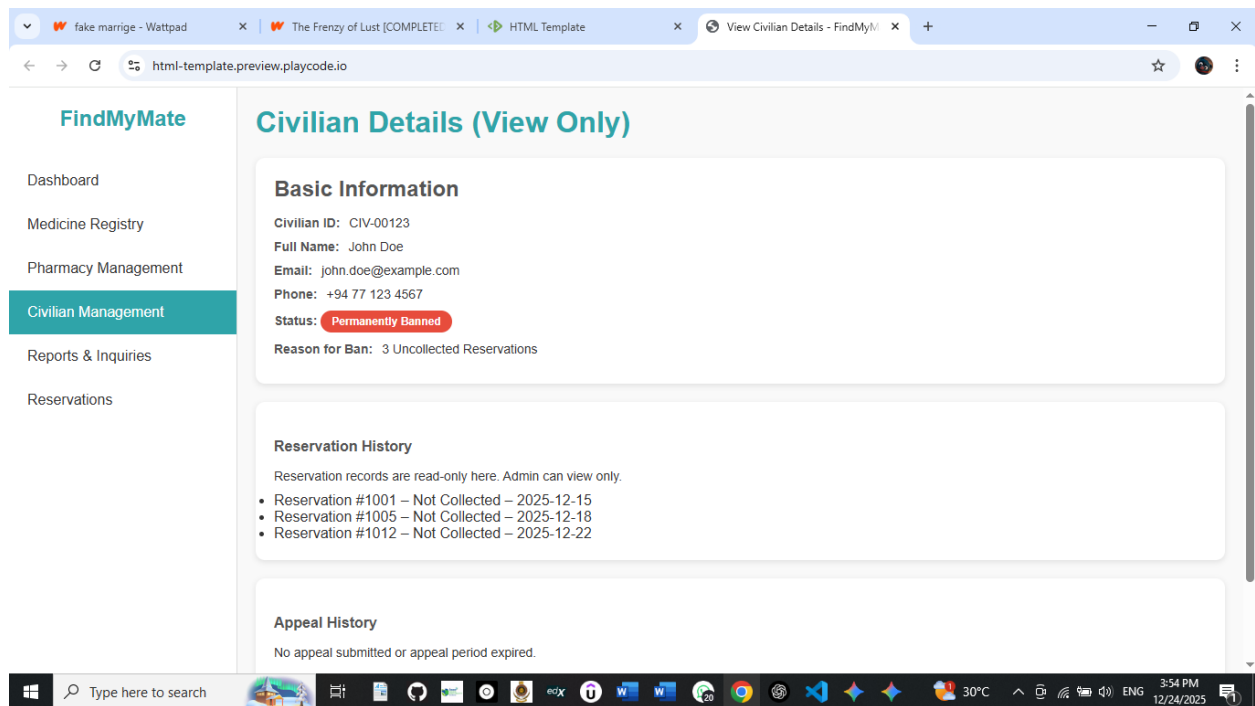
Behavior:

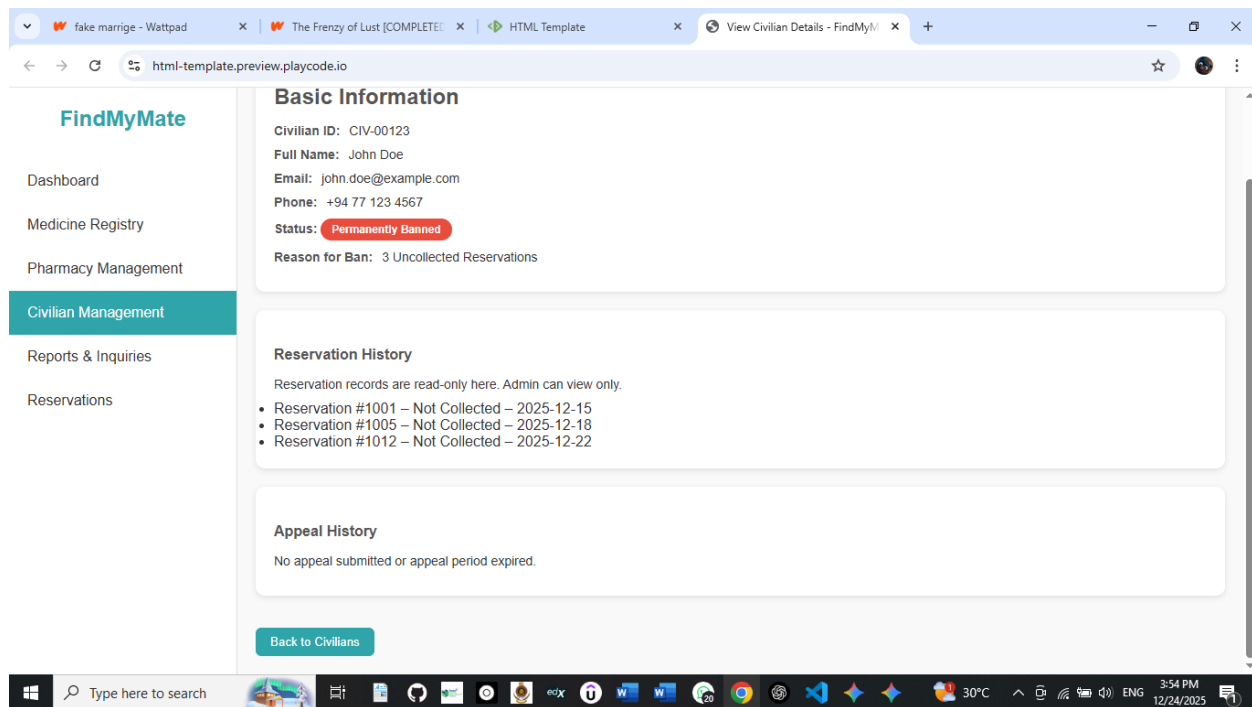
- Table filtered to show only permanently banned civilians
- Actions limited to **View / VIVO Manage**
- No activation, no appeal

Components & Contents (Bullet-wise):

- Metric Card: Total Permanently Banned Civilians
- Table Columns:
 - Civilian ID
 - Name
 - Email
 - Permanent Ban Date
 - Auto Deletion Countdown
 - Actions (View / VIVO Manage)
- Info:
 - Auto deletion countdown (e.g., “Deletes in 48 days”)

5 Permanently Deleted Civilians (VIVO Manage Page)





Access:

- Civilian Details → **VIVO Manage**

Purpose:

- Safe inspection of permanently banned accounts
- Legal compliance and data privacy

Behavior:

- Read-only access to all sensitive info
- Admin can **disable login** or **anonymize fields**
- No edits or activation allowed

Components & Contents (Bullet-wise):

- **Masked Civilian Info**
 - Civilian ID
 - Masked Name / Email
- **Read-Only History**
 - Reservation history
 - Ban & appeal history
 - System action logs

- **Action Button**
 - Back
 - **Legal Compliance**
 - Supports 3-month retention policy
 - Auto deletion triggered after retention period
-

6 Auto Deletion Process (No UI Page)

Purpose:

- System hygiene and legal compliance

Behavior:

- Triggered automatically **3 months after permanent ban**
- Deletes account login credentials
- Minimal audit logs retained
- No manual override
- No admin UI required

Components & Contents:

- Triggered automatically by backend scheduler
 - Logs deletion in system history
 - No UI components
-

☑ This summary now covers:

- Access paths for each page
- Purpose of each page
- Behavior rules per account status
- Bullet-wise contents and components
- Role-based visibility for action buttons
- Auto processes like countdowns, appeals, and deletion

Backend part

1 Civilian Management Home Page

Backend Logic & Steps:

1. **Fetch Civilian Data for Metric Cards**
 - Query the `civilians` table to get counts:
 - Total civilians
 - Active civilians
 - Temporarily banned civilians
 - Permanently banned civilians
 - Return counts to frontend to display metric cards
 2. **Fetch Civilian Table Data**
 - Query `civilians` table for required columns:
 - Civilian ID, Name, Email, Account Status, Temp Ban Count, Appeal Count, Last Action Date
 - Apply filtering based on metric card selected (status filter)
 - Return filtered dataset to the frontend table
 3. **Activate Button Logic**
 - Only show **Activate** button if:
 - Civilian status = Temporarily Banned
 - Appeals remaining > 0
 - Clicking **Activate** triggers an API call:
 1. Validate eligibility
 2. Update `civilians.status = Active`
 3. Reset temporary ban count if needed
 4. Log the action in `civilian_history` table
-

2 Civilian Details Page

Backend Logic & Steps:

1. **Fetch Civilian Details**
 - Query `civilians` table by ID
 - Fetch:
 - Basic info
 - Account status
 - Temp ban count
 - Appeal count
 - Ban info (if banned)
 - Appeal deadline (calculate from `ban_date + 14 days`)

2. Conditional Actions

- Based on `civilians.status`:
 - Temp Banned → show **View Appeal Details**
 - Permanently Banned → show **VIVO Manage**
 - Active → no ban actions

3. Countdown Logic for Appeals

- Backend calculates `remaining_days = 14 - (current_date - ban_date)`
 - If `remaining_days ≤ 0` → automatically move account to permanent ban
-

3 Appeal Details Page

Backend Logic & Steps:

1. Fetch Appeal Details

- Query `appeals` table for selected appeal ID
- Include:
 - Appeal ID, Appeal Number, Submission Date, Message, Attachments
 - Civilian reservation history summary
 - Temp ban count
 - Remaining appeals

2. Admin Decision Actions

- **Activate Account**
 1. Validate appeal count ≤ 2
 2. Update `civilians.status = Active`
 3. Increment temp ban count
 4. Log action in `civilian_history` table
- **Permanently Ban Account**
 0. Update `civilians.status = Permanently Banned`
 1. Log action with timestamp
 2. Trigger **auto-deletion scheduler** (3 months countdown)
- **Back** → no backend action

3. Appeal Count Enforcement

- Before activation, check if appeal count exceeds max (2)
 - If yes → automatically mark as permanent ban
-

4 Permanently Banned Civilian Page (Filtered Table)

Backend Logic & Steps:

1. Fetch Permanently Banned Civilians

- Query `civilians` table where `status = Permanently Banned`

- Include:
 - ID, Name, Email, Permanent Ban Date, Auto Deletion Countdown
 - 2. **VIVO Manage Button**
 - API call fetches read-only details from `civilians + civilian_history`
 - No modification allowed
-

5 Permanently Deleted Civilians (VIVO Manage Page)

Backend Logic & Steps:

1. **Fetch Deleted Account Details**
 - Query `deleted_civilians` table (archived data)
 - Mask sensitive info (e.g., hash email, mask name)
 - Fetch reservation history (read-only)
 - Show ban & appeal history
 - Admin actions only include:
 - Disable login credentials
 - Optional anonymization
 2. **Deletion Policy**
 - Triggered by **scheduler** 3 months after permanent ban
 - Remove login access and personal identifiers
 - Retain minimal audit logs
-

6 Auto Workflows

a) Temporary Ban → Permanent Ban Auto Workflow

- Backend scheduler runs daily:
 1. Query `civilians` where `status = Temporarily Banned`
 2. Calculate `days_since_ban = current_date - ban_date`
 3. If `days_since_ban > 14` and no appeal submitted:
 - Update `status = Permanently Banned`
 - Log action

b) Auto Deletion of Permanently Banned Accounts

- Backend scheduler runs daily:
 1. Query `civilians` where `status = Permanently Banned`
 2. Check if `current_date - permanent_ban_date >= 90 days`
 3. Move record to `deleted_civilians` table
 4. Remove login credentials

5. Keep minimal audit info

7 Tables / Database Structure (Simplified)

- `civilians`
 - `id`, `name`, `email`, `contact`, `status`, `temp_ban_count`, `appeal_count`, `ban_date`, `created_at`, `updated_at`
- `appeals`
 - `id`, `civilian_id`, `appeal_number`, `message`, `attachments`, `submitted_at`
- `civilian_history`
 - `id`, `civilian_id`, `action_type` (Temp Ban / Permanent Ban / Activation), `action_by`, `timestamp`
- `deleted_civilians`
 - All historical info of permanently deleted civilians (masked sensitive info)
- `scheduler_logs`
 - Optional: to track automatic actions like bans and deletions

✓ Key Points for Implementation

1. Use **role-based checks**: Admin vs Super Admin for button visibility and actions.
2. Use **scheduler or cron jobs** for:
 - 14-day appeal countdown
 - 3-month auto deletion
3. Maintain **history logs** for audit and compliance.
4. Enforce **max 2 temp bans per civilian**.
5. All table updates trigger **real-time UI refresh** if using SPA or via API response.

Backend part of the core rules explanation

1 Temporary Ban Rules (Enhanced)

Rule Explanation:

- A civilian can only be temporarily banned **2 times in their lifetime**.
- A civilian can only submit **2 appeals in total**.
- After exceeding these limits → automatic **PERMANENT BAN**.
- No exceptions, no third chances.

Backend Implementation Steps:

1. Create fields in `civilians` table:
 - `temp_ban_count` → counts how many times temporarily banned
 - `appeal_count` → counts total appeals submitted
 2. When a civilian triggers a temporary ban (e.g., 3 uncollected orders):
 - Check `temp_ban_count < 2`
 - If yes → update status to `TEMPORARY_BANNED`
 - Increment `temp_ban_count += 1`
 - If no → automatically update status to `PERMANENTLY_BANNED`
 3. When a civilian submits an appeal:
 - Check `appeal_count < 2`
 - If yes → allow submission, increment `appeal_count += 1`
 - If no → block appeal, auto-permanent ban
 4. All changes logged in `civilian_history` table.
-

2 Temporary Ban → Permanent Ban Triggers

Rule Explanation:

- Account becomes **PERMANENTLY BANNED** if ANY of these happens:
 1. No appeal submitted within **14 days**.
 2. Appeal rejected by Admin.
 3. Temporary ban count exceeds 2.
 4. Appeal count exceeds 2.

Backend Implementation Steps:

1. **14-day appeal deadline check**
 - Add `ban_date` field in `civilians`.
 - Daily scheduler checks:
 - `WHERE status = TEMPORARY_BANNED AND CURRENT_DATE - ban_date > 14`
 - If true → update `status = PERMANENTLY_BANNED`, log action.
 2. **Appeal rejected by Admin**
 - When Admin rejects appeal → immediately update `status = PERMANENTLY_BANNED`.
 3. **Temp ban count exceeded**
 - Already handled in previous rule: check `temp_ban_count > 2`.
 4. **Appeal count exceeded**
 - Already handled in previous rule: check `appeal_count > 2`.
-

3 Updated Status Lifecycle

Rule Explanation:

- Lifecycle of a civilian's account is clean and controlled:

Flow Example:

1. Active → 3 uncollected orders → TEMPORARY_BANNED #1
 - Appeal #1 → approved → Active
 - Appeal #1 → rejected → PERMANENTLY_BANNED
2. Active → 3 uncollected orders → TEMPORARY_BANNED #2
 - Appeal #2 → approved → Active
 - Appeal #2 → rejected → PERMANENTLY_BANNED
3. Attempt TEMPORARY_BANNED #3 → automatic PERMANENTLY_BANNED (no temp ban #3 allowed)

Backend Implementation Steps:

1. Every time a civilian triggers a temporary ban:
 - Check `temp_ban_count < 2` → allow temp ban
 - Else → auto permanent ban
 2. Appeal results update status:
 - Approved → `status = ACTIVE`
 - Rejected → `status = PERMANENTLY_BANNED`
 3. Log every status change in `civilian_history`.
-

4 Appeal Rules (Very Strict)

Rule Explanation:

- Maximum 2 appeals per lifetime.
- Appeal must be submitted within **14 days** of temp ban.
- Once submitted, appeals are read-only.
- Admin decision is final.
- If appeal limit exceeded → no appeal button, auto permanent ban.

Backend Implementation Steps:

1. On temp ban, calculate `appeal_deadline = ban_date + 14 days`.
2. Backend validates on appeal submission:
 - `appeal_count < 2` → allow submission

- `appeal_count >= 2` → reject submission, update `status = PERMANENTLY_BANNED`
 - 3. Store appeal in `appeals` table:
 - Fields: `appeal_id`, `civilian_id`, `appeal_number`, `message`, `attachments`, `submitted_at`
 - 4. Admin actions:
 - Approve → activate account
 - Reject → permanent ban
-

5 Permanent Ban — Behavior

Rule Explanation:

- Permanently banned accounts:
 - Can be **viewed by Admin/Super Admin**
 - **VIVO Manage button** available (read-only)
 - Cannot activate, appeal, or edit

Backend Implementation Steps:

1. Query `civilians` where `status = PERMANENTLY_BANNED`.
 2. Only show read-only details on the UI.
 3. Log all view actions for audit purposes.
 4. Admin can:
 - Mask personal info
 - Disable login
 - Anonymize sensitive data
-

6 Data Sanitization (Manual Control)

Rule Explanation:

- Admin/Super Admin can manually:
 - Mask email, phone
 - Disable login credentials
 - Anonymize personal data

Backend Implementation Steps:

1. Add fields in `civilians` table for masked data:
 - `masked_email`, `masked_phone`, `is_login_disabled`

2. Backend updates these fields on VIVO Manage action.
 3. Original data can be archived in `civilian_history` for compliance.
-

7 Auto Deletion Rule (3 Months)

Rule Explanation:

- After **3 months** from permanent ban:
 - System deletes the account
 - Deletes login credentials
 - Keeps minimal audit log (ID + actions)

Backend Implementation Steps:

1. Add `permanent_ban_date` in `civilians`.
 2. Daily scheduler:
 3. `WHERE status = PERMANENTLY_BANNED AND CURRENT_DATE - permanent_ban_date >= 90`
 - Move record to `deleted_civilians` table (archive)
 - Remove login credentials
 - Keep minimal audit log
 - Delete from main `civilians` table
-

✓ Summary of Backend Tables Needed:

- `civilians` → main table with status, ban & appeal counts
- `appeals` → records of appeals
- `civilian_history` → audit of all actions (ban, activate, appeal decisions)
- `deleted_civilians` → permanently removed accounts archive
- Scheduler/cron jobs → auto permanent ban, auto deletion

Civilian Ban Modals

Purpose:

Provide Admins and Super Admins with a controlled interface to temporarily or permanently ban civilian accounts based on system rules.

Key Points:

- **Temporary Ban Modal:**

- Used when a civilian has 3 uncollected orders or policy violations.
- Requires reason for ban.
- Starts a 14-day appeal window.
- Tracks ban count (max 2 temporary bans allowed).
- **Permanent Ban Modal:**
 - Triggered automatically or manually when rules are violated:
 - Appeal rejected or not submitted in 14 days
 - Exceeded max temporary bans or appeals
 - Requires reason for permanent ban.
 - Triggers 3-month auto-deletion schedule.

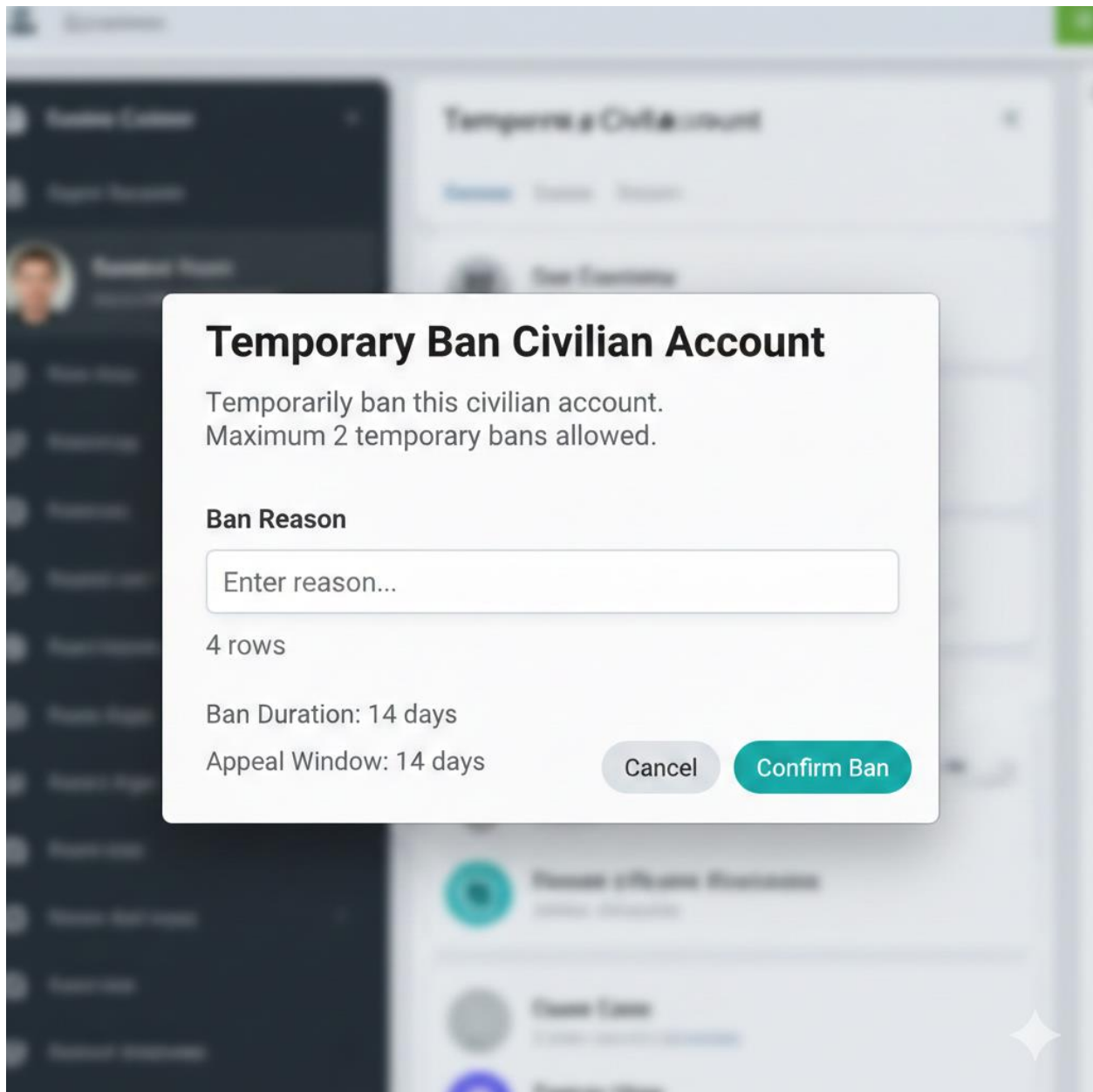
Behavior & Integration:

- Modals **update the account status** in the backend (TEMPORARY_BANNED / PERMANENTLY_BANNED).
- Sends **notifications** to civilians about the action.
- **Audit logs** are created for all ban actions.
- Backend checks **rules automatically** to enforce max bans, appeals, and auto-deletion.

Benefits:

- Ensures **rules are consistently applied**.
- Keeps **audit trail and legal compliance** intact.
- Provides a **clear, deliberate admin workflow** for banning users.

1 Temporary Ban Modal



Purpose

- Temporarily ban a civilian account that violates system rules (e.g., 3 uncollected orders).
- Provides a 14-day ban window during which the civilian can submit an appeal.
- Maximum 2 temporary bans per civilian; exceeding this leads to permanent ban.

Access

- Accessed from: **Civilian Details Page** → Temporary Ban button.
- Visible only to: **Admin & Super Admin**.

Components

- **Title:** Temporary Ban Civilian Account
- **Description:** Explains why the ban is applied and duration.
- **Ban Reason Textarea:** Required field where admin inputs reason for ban.
- **Duration Info:** Shows ban period (14 days) and appeal window (14 days).
- **Buttons:**
 - Cancel → Closes modal.
 - Confirm Ban → Executes the temporary ban logic.

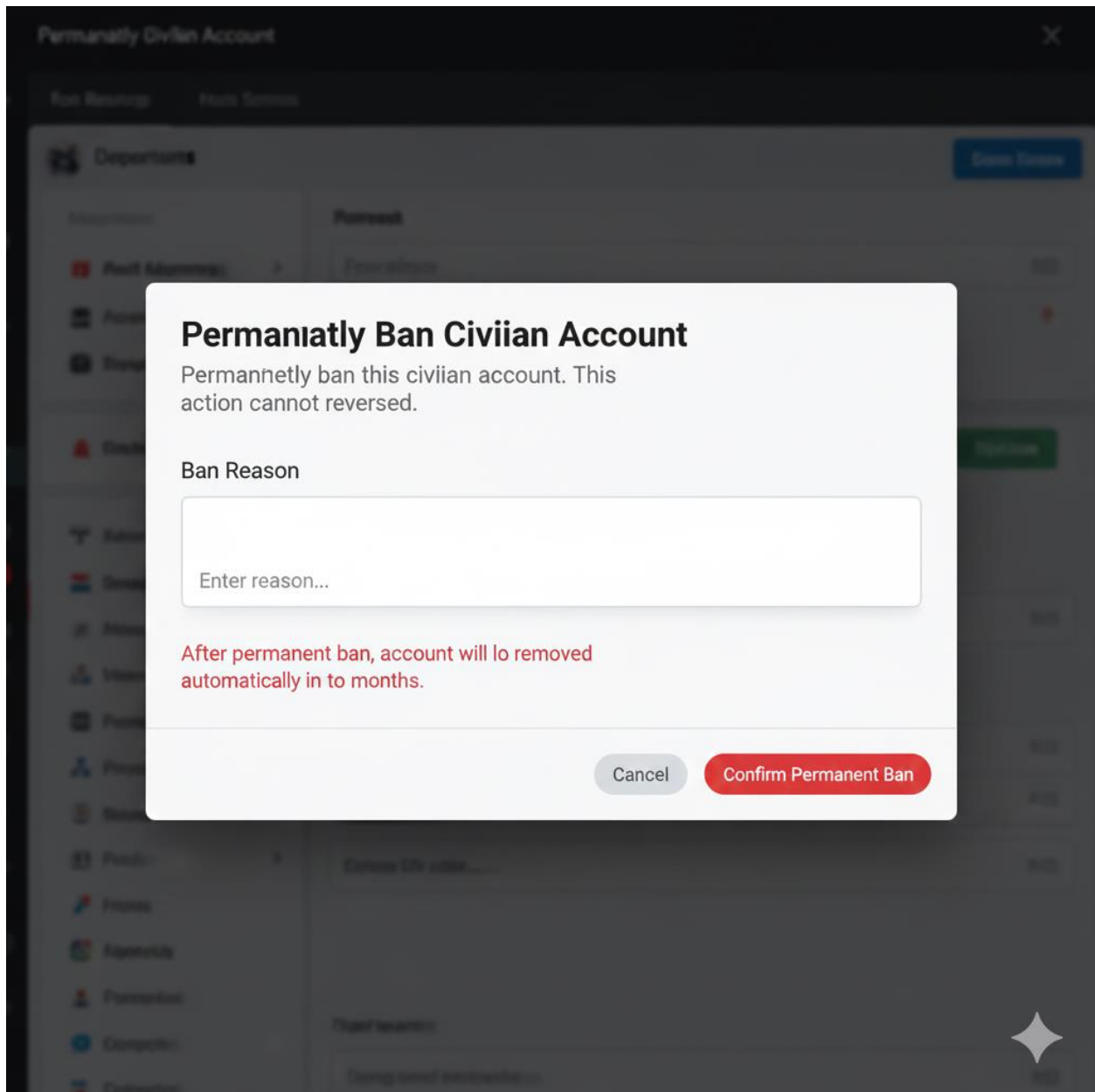
Behavior

- Admin fills the reason → clicks Confirm → the system updates civilian status to `TEMPORARY_BANNED`.
- The civilian receives a notification about temporary ban.
- Ban count and appeal deadline are updated in the database.
- If civilian does not appeal within 14 days, backend triggers **auto-permanent ban**.

Backend Implementation Steps

1. **Fetch civilian record** from database using ID.
2. **Validate:** Ensure `tempBanCount < 2`. If not, send error → next ban is permanent.
3. **Update civilian status:**
 - `status = TEMPORARY_BANNED`
 - `tempBanCount += 1`
 - `banReason = admin input`
 - `banDate = now`
 - `appealDeadline = now + 14 days`
4. **Log action** in `CivilianHistory` table for audit.
5. **Notify civilian** about temporary ban via notification/email.
6. **Return response** to frontend for confirmation toast.

2 Permanent Ban Modal



Purpose

- Permanently ban a civilian account.
- Triggered when temporary ban rules are exceeded, appeal rejected, or admin manually decides.
- Accounts are auto-deleted after 3 months (retention period).

Access

- Accessed from: **Civilian Details Page** → Permanent Ban button.
- Visible only to: **Admin & Super Admin**.

Components

- **Title:** Permanently Ban Civilian Account
- **Description:** Explains permanence and consequences.
- **Ban Reason Textarea:** Required field for admin to input reason.
- **Warning Text:** Explains auto-deletion after 3 months.
- **Buttons:**
 - Cancel → Closes modal.
 - Confirm Permanent Ban → Executes permanent ban logic.

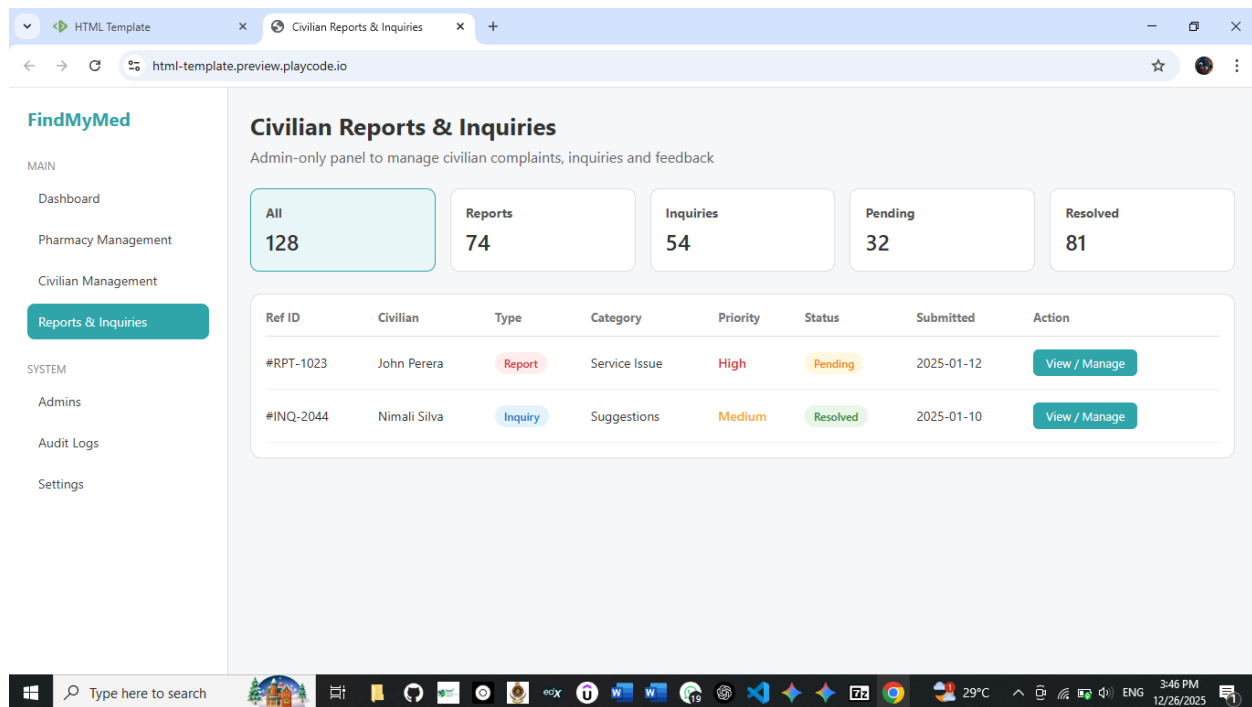
Behavior

- Admin fills reason → clicks Confirm → system updates civilian status to `PERMANENTLY_BANNED`.
- Login credentials are disabled immediately.
- Civilian receives notification about permanent ban.
- Admin can use **VIVO Manage** to anonymize sensitive data.
- System schedules account for auto-deletion after 3 months.

Backend Implementation Steps

1. **Fetch civilian record** from database using ID.
2. **Update civilian status:**
 - `status = PERMANENTLY_BANNED`
 - `permanentBanDate = now`
 - `banReason = admin input`
 - `isLoginDisabled = true`
3. **Log action** in `CivilianHistory` for auditing.
4. **Notify civilian** about permanent ban.
5. **Schedule auto-deletion** job for 3 months later.
6. **Return response** to frontend for confirmation toast.

Civilian Reports and Inquiries Management Page



1 How the Page Can Be Accessed

- **Navigation:** Admin Panel → Civilian Management → “Reports & Inquiries” button.
- **Permissions:** Only **Admin** has access. Super Admin does not handle this page.
- **Purpose:** To let the Admin view, manage, and respond to all reports and inquiries submitted by civilians.

2 Page Components

1. Metric Cards (Top Section)

- Display counters for:
 - Total Reports & Inquiries
 - Reports
 - Inquiries
 - Pending
 - Resolved
 - Rejected
- Clicking a card filters the main table automatically by type/status.

2. Main Table

- **Columns:**
 - Report / Inquiry ID
 - Civilian Name & ID
 - Type (Report / Inquiry)
 - Issue Category (Technical / Inventory / Reservation / Other)

- Priority (High / Medium / Low)
- Status (Pending / In Progress / Resolved / Rejected)
- Submitted Date
- Actions (View / Manage button)
- **Behavior:**
 - Default view: Pending reports/inquiries.
 - Clicking **View/Manage** opens the detailed report/inquiry page.
 - Table auto-updates when status changes.
 - Filters controlled via metric cards.
 - Resolved and rejected reports and inquiries will be automatically deleted after 60 days from the system.
 - User will get a notification after their report or inquiry being resolved or rejected.

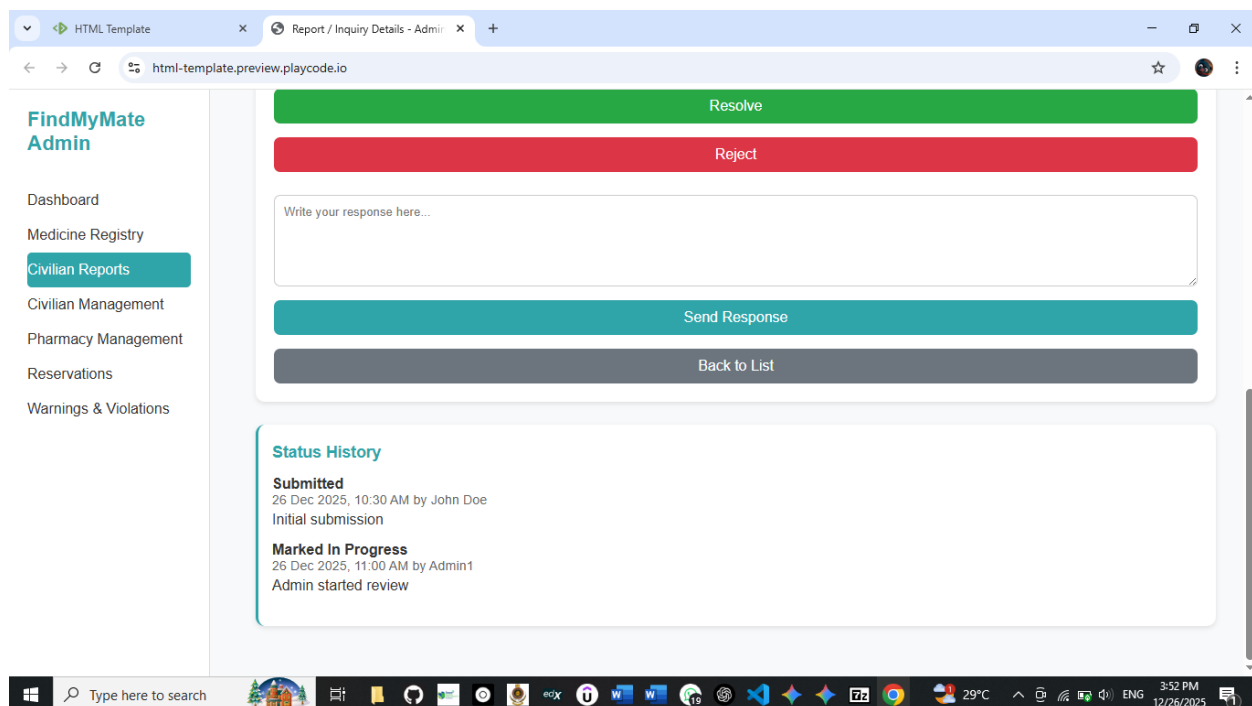
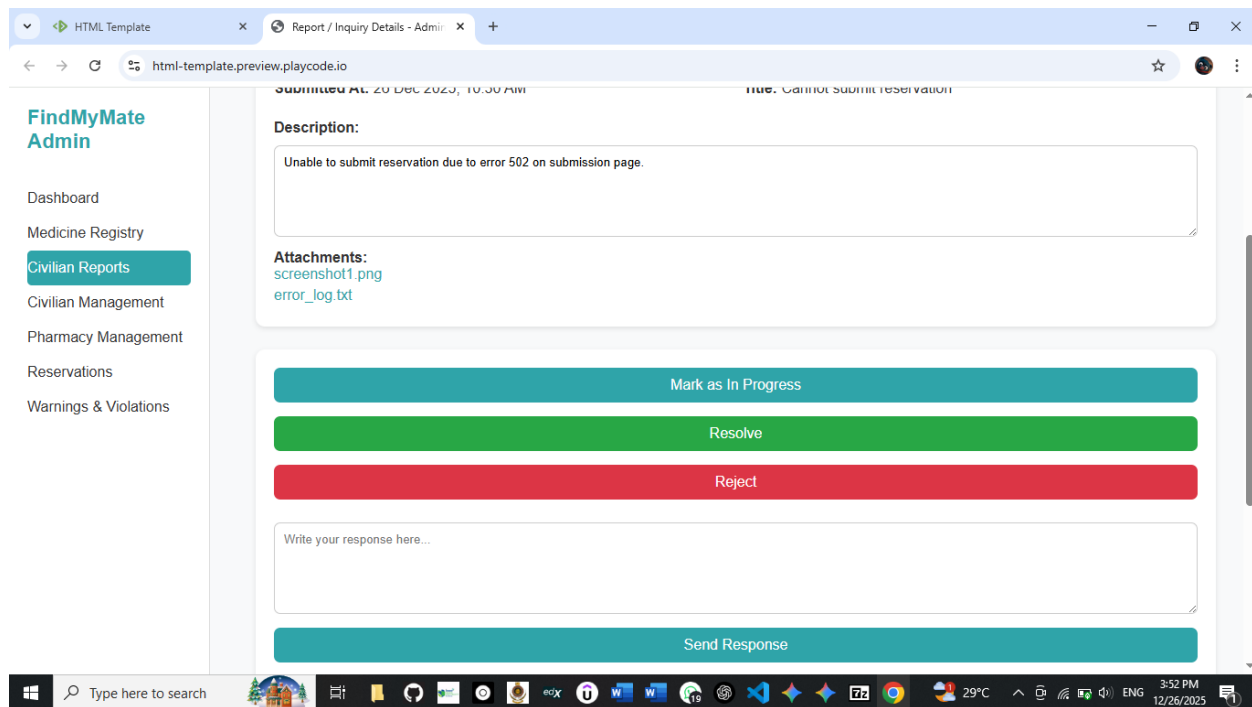
Civilian Report / Inquiry Detail Page

The screenshot displays the 'FindMyMate Admin' interface. On the left is a sidebar with navigation links: Dashboard, Medicine Registry, Civilian Reports (highlighted), Civilian Management, Pharmacy Management, Reservations, and Warnings & Violations. The main content area is titled 'Report / Inquiry Details' with the subtitle 'Review and manage civilian submissions'. It features a 'Report Information' section with the following details:

- Reference ID:** #RPT12345
- Type:** Inquiry
- Priority:** High (indicated by an orange tag)
- Submitted At:** 26 Dec 2025, 10:30 AM
- Submitted By:** John Doe (ID: CIV1001)
- Issue Category:** Technical
- Status:** In Progress (indicated by a blue tag)
- Title:** Cannot submit reservation

The **Description:** field contains the text: 'Unable to submit reservation due to error 502 on submission page.' Below this, the **Attachments:** section lists 'screenshot1.png' and 'error_log.txt'. At the bottom of the report card, there is a button labeled 'Mark as In Progress'.

The browser's address bar shows 'html-template.preview.playcode.io'. The Windows taskbar at the bottom indicates the system time is 3:52 PM on 12/26/2025, with a temperature of 29°C.



1 How it Can Be Accessed

- Accessed by clicking “**View / Manage**” button in the **Civilian Reports & Inquiries Table** on the main admin page.

- Only accessible to **Admin** (Super Admin cannot handle this page).
-

2 Purpose

- Allow Admin to **review, manage, and respond** to each report or inquiry submitted by a civilian.
 - Track **status changes, admin responses, and attachment files**.
 - Ensure **audit trail** of all actions taken on the report/inquiry.
-

3 Page Layout & Sections

A. Report / Inquiry Information Section

This is the main info card about the submission. Read-only except for admin actions.

Content to Include:

- **Reference ID / Report ID:** Unique ID of the report/inquiry.
- **Civilian Info:** Name, ID, email/contact.
- **Type:** “Report” or “Inquiry”.
- **Issue Category:** Technical, Inventory, Reservation, Other.
- **Priority Level:** High / Medium / Low (badge style).
- **Status:** Pending / In Progress / Resolved / Rejected (badge style, color-coded).
- **Title / Subject:** Short descriptive title of the report.
- **Description:** Full description text explaining the issue.
- **Attachments (if any):** Downloadable files (PDF, image, etc.) displayed as cards with file name and download icon.
- **Submitted Date & Time:** Timestamp when the report/inquiry was created.

Behavior Notes:

- Read-only for Admin to prevent accidental editing.
 - Attachments can be downloaded but not edited.
 - Status badge updates instantly when admin performs an action.
-

B. Admin Action Section

This panel is where Admin can take actions. Should be clearly separated and visible, often aligned on the right or top of the page.

Action Buttons:

- **Mark as In Progress:** Changes status to “In Progress”. No response required.
- **Resolve Report / Inquiry:** Changes status to “Resolved”. Optional response message can be added.
- **Reject Report / Inquiry:** Changes status to “Rejected”. Mandatory reason or message should be added in a modal.
- **Send Response / Message to Civilian:** Admin can type a message to the civilian, optionally attach a file.
- **Back to List:** Returns to the main table.

Behavior Notes:

- Clicking **Resolve** or **Reject** triggers a confirmation modal.
 - Any status change updates the **status_changed_at** timestamp for tracking and auto-deletion purposes.
 - Messages sent by admin are stored in **civilian_report_responses** table.
-

C. Admin Response Section

- Card below the main info or side panel for optional communication.
 - **Content to Include:**
 - Textarea for **response message**.
 - Optional file attachment upload.
 - Button: **Send Response**.
 - Response is saved and linked to the report/inquiry.
 - Civilians can view this response on their portal (optional feature if you have a civilian portal).
-

D. Status & History / Timeline Section

- Timeline-style panel showing **all actions** taken on the report/inquiry.
 - **Each entry includes:**
 - Action (Submitted, Status Changed, Response Sent)
 - Date & Time
 - Performed by (Admin)
 - Optional comment / message
 - Helps in auditing and keeping track of the workflow.
-

4.Quick Content Checklist

Report / Inquiry Info Section:

- Reference ID
- Civilian Name / ID
- Type
- Issue Category
- Priority
- Status
- Title / Subject
- Description
- Attachments
- Submission Date & Time

Admin Action Section:

- Mark as In Progress
- Resolve Report / Inquiry
- Reject Report / Inquiry
- Send Response
- Back to List

Timeline / History Section:

- All past actions with timestamps
- Admin responses
- Status changes

1 Civilian Reports & Inquiries Main Page – Backend Logic

Purpose

Display all reports and inquiries in a table with filtering, metric cards, and status tracking.

Backend Implementation Steps

Step 1: Database Tables

- `civilian_reports` table:
 - `id` (PK)
 - `civilian_id` (FK to `civilians` table)
 - `type` (Report / Inquiry)

- category (Technical / Inventory / Reservation / Other)
 - priority (High / Medium / Low)
 - status (Pending / In Progress / Resolved / Rejected)
 - title
 - description
 - attachments (JSON / separate table for files)
 - submitted_at (datetime)
 - status_changed_at (datetime)
- civilian_report_responses table:
 - id, report_id, admin_id, response_message, attachment, created_at

Step 2: Fetch Data for Page

- Query civilian_reports table to get all reports and inquiries.
- Count metrics:
 - Total reports/inquiries
 - Count by type (Report / Inquiry)
 - Count by status (Pending / In Progress / Resolved / Rejected)
- Apply filters based on **metric card clicks**.

Step 3: Status Update APIs

- Admin actions trigger API calls:
 - Mark as In Progress → update status = 'In Progress', status_changed_at = now()
 - Resolve → status = 'Resolved', status_changed_at = now()
 - Reject → status = 'Rejected', status_changed_at = now(), store rejection reason

Step 4: Automatic Deletion Cron Job

- **Purpose:** Delete resolved/rejected reports & inquiries after 60 days.
- **Implementation:**

```
DELETE FROM civilian_reports
WHERE status IN ('Resolved', 'Rejected')
AND status_changed_at <= NOW() - INTERVAL 60 DAY;
```

- Run via server cron job daily.
- Also delete linked responses and attachments.

Step 5: Return Data to Frontend

- Return table data with:
 - Filtered by type/status
 - Metric counts
 - Pagination if needed

2 Civilian Report / Inquiry Details Page – Backend Logic

Purpose

Show full details of one report/inquiry and allow admin to manage it (status change, response, attachments).

Backend Implementation Steps

Step 1: Fetch Full Report Data

- Query `civilian_reports` table by id.
- Include:
 - Civilian info (name, ID, contact)
 - Type, category, priority, status, title, description
 - Submitted date/time
 - Attachments
- Query `civilian_report_responses` table to fetch all admin responses.

Step 2: Admin Actions

- **Mark as In Progress**

```
UPDATE civilian_reports
SET status = 'In Progress', status_changed_at = NOW()
WHERE id = <report_id>;
```

- **Resolve Report / Inquiry**

```
UPDATE civilian_reports
SET status = 'Resolved', status_changed_at = NOW()
WHERE id = <report_id>;
```

```
INSERT INTO civilian_report_responses(report_id, admin_id, response_message,
created_at)
VALUES (<report_id>, <admin_id>, <optional_message>, NOW());
```

- **Reject Report / Inquiry**

```
UPDATE civilian_reports
SET status = 'Rejected', status_changed_at = NOW(), rejection_reason =
'<reason>'
WHERE id = <report_id>;
```

```
INSERT INTO civilian_report_responses(report_id, admin_id, response_message,
created_at)
VALUES (<report_id>, <admin_id>, '<optional_message>', NOW());
```

- Ensure **mandatory rejection reason** is submitted.

Step 3: Response Handling

- Admin can send a message (with optional attachment)

```
INSERT INTO civilian_report_responses(report_id, admin_id, response_message,
attachment, created_at)
VALUES (<report_id>, <admin_id>, '<message>', '<file>', NOW());
```

Step 4: Automatic Deletion

- Same cron job as main table:

```
DELETE FROM civilian_reports
WHERE status IN ('Resolved', 'Rejected')
AND status_changed_at <= NOW() - INTERVAL 60 DAY;

DELETE FROM civilian_report_responses
WHERE report_id NOT IN (SELECT id FROM civilian_reports);
```

- This ensures orphaned responses are also deleted.

Step 5: Return Data to Frontend

- Full report details
- Admin responses (timeline)
- Status history
- Attachments for download

3 Notes / Best Practices

- **Atomic Operations:** Ensure status update and response insertion happen in the same transaction.
- **Audit Trail:** Always log `admin_id` and `timestamp`.
- **Validation:** Only allow **Admin** to update status or send responses.
- **Notifications:** Optional – send email or system notification to civilian when status changes.
- **Security:** Validate access to report IDs; users cannot fetch reports of other civilians.

✓ With this setup:

- Main table shows all reports & inquiries with filtering and metric cards.
- Details page allows full inspection, admin response, and status changes.

- Resolved / Rejected items are automatically cleaned up after 60 days to reduce storage and maintain compliance.