IT Assets Management Solution

The Manual: IT & POS Hardware Inventory System
Category: Developer Documentation
Version: 1.12.29

---

1. Executive Summary & Purpose

The **Enterprise Asset & POS Management Solution** is a specialized system designed to manage the full lifecycle of IT assets (monitors, keyboards, PCs) and POS hardware (scanners, cash drawers).

Unlike standard inventory systems, this platform focuses on **Chain of Custody**. It tracks not only current stock levels but also physical location, user assignment, and the real-time health status of every hardware unit.

---

2. Technical Architecture

The system is built using a **Procedural-Modular** approach to ensure ease of maintenance and logical separation of concerns.

Backend Logic (PHP)

- **Session Management:** Security is handled at the page level. Every file within `web_content` includes a verification script to ensure an active session; unauthorized users are redirected to `index.php`.
- **Database Layer:** Managed via `/render/connection.php`. The system utilizes a persistent connection with the `utf8mb4` charset to preserve technical serial numbers and special characters.
- **Component Rendering:** To maintain DRY (Don't Repeat Yourself) principles, `modal.php` serves as a central template library for reusable UI elements like "Delete" or "Edit" buttons.

Frontend & UI (Bootstrap 5 & JS)

- **Responsive Grid:** Utilizing the Bootstrap 5 Grid System, inventory tables are optimized for mobile devices, allowing managers to conduct stock takes on the floor.
- **State Persistence:** The `dark_mode_script.js` utilizes browser `localStorage` to ensure the user’s theme preference persists across sessions and refreshes.

---

3. Database Data Dictionary

The system relies on a relational structure of 12 tables. Below are the primary schemas for core operations.

Table: `users` (Security Hub)

Column	Type	Purpose
`user_id`	INT (11)	Primary Key. Auto-incremented.
`role_id`	VARCHAR (15)	Foreign Key; defines user permissions.
`username`	VARCHAR (999)	Unique login identifier.
`password`	VARCHAR (255)	Stored as a secure hash.
`is_active`	TINYINT (1)	Boolean: 1 = Active, 0 = Disabled.
`last_login`	DATETIME	Audit trail for user access.

Table: `branch_logs` (Logistics Hub)

Column	Type	Purpose
`log_id`	INT (11)	Primary Key.
`action_type`	ENUM	Restricts to: 'Transfer', 'Restock', 'Damage', 'Adjustment'.
`origin_branch`	VARCHAR	Departure point (e.g., "Warehouse").
`destination`	VARCHAR	Arrival point (e.g., "Branch 05").

Column	Type	Purpose
`remarks`	TEXT	Detailed notes on the movement or condition.

Table: `products` (Asset Hub)

- **`product_id`**: Unique database identifier.
- **`category_id`**: Links to the classifications table.
- **`status`**: Enumerable state: *Available, Allocated, Damaged, Under Repair*.

---

4. Advanced Operational Logic

The system automates business workflows to ensure data integrity during hardware movement.

The "Chain of Custody" Flow

1. **Request:** User initiates movement via `allocation.php`.
2. **Validation:** System verifies the asset status is "Available."
3. **Update:** `product_allocations` links the asset to a specific Branch.
4. **Logging:** `branch_logs.php` triggers an automatic audit record (Action: Transfer).

Damage & Maintenance Flow

- **Reporting:** Staff submits a report via `damage_monitoring.php`.
- **State Change:** Asset status is updated to "Damaged" in the `products` table.
- **Audit Trail:** Dual logs are created in `damaged_products` (specifics) and `system_audit_logs` (general asset change).

---

5. Developer’s Maintenance Guide

Modifying the System

- **New Categories:** Never hardcode categories into the HTML. Use the `categories.php` interface; this ensures all dropdowns in `inventory.php` update dynamically.
- **Scripting:** Place new JavaScript in `/src/script/`. Always link scripts at the bottom of the `<body>` tag to optimize page load speeds.

Data Safety (Soft Deletes)

The system utilizes **Soft Deletes** (`deleted_at` timestamp) instead of `DELETE` queries. This prevents "orphaned" data where transaction logs would point to missing user or product IDs.

- **To Restore:** Change the `deleted_at` value in the database to `NULL`.

---

6. Disaster Recovery & Migration

To migrate the system to a new server:

1. **File Transfer:** Move the project folder to the new `htdocs`.
2. **SQL Setup:**
    - Create a database matching the name in `connection.php`.
    - Import the latest `.sql` file from `/data_backups/`.
3. **Permissions:** Verify the MySQL user has "ALL PRIVILEGES" to ensure log tables can be updated.

---

7. Troubleshooting FAQ

- **Error: "Connection failed: Access denied"**
    - *Solution:* Check `render/connection.php`. Ensure the username and password match your local environment (usually `root` with no password).
- **Error: "Page not found" on Nav Link**
    - *Solution:* Inspect `nav/header.php`. Verify the relative paths to the `web_content/` folder.
- **Issue: New logo/styling not showing**
    - *Solution:* Clear browser cache. Bootstrap 5 assets are often heavily cached by the browser.