Needle Management Project

# Scope

To create a web-based system to replace current needle management system that we don’t have any control. We need to have multiple user logins with multiple user permission levels and user rolls. The system must have the ability to create reports that can help the auditing.

Have to capture image of the broken needle and keep them in a database to use to create reports

Currently using droidcam

# Rolls scenarios

## Operator

* Needle Distribute
* Needle refill
* Needle Request
* Send request to manager for approval
* Check available stock

## Manager

* Approve the needle requests
* Check all transactions related to their department
* Generate reports from available data

## Stores

* Check the needle request and available quantity
* Distribute requested needles
* Place order to new needles

## Purchasing

* Check the new needle request from stores
* Enter whether new needle request is processed or not

## Observer

* Check all transactions
* Generate reports from available data

## Admin

* Add new employees
* Add new sewing machines
* Adding new embroider machines
* Adding new needles
* Creating and modifying the operator account and giving permissions
* Needle Distribute
* Needle refill
* Needle Request
* Check available stock
* Check the needle request and available quantity
* Distribute requested needles

## Super admin

* Add new employees
* Add new sewing machines
* Adding new embroider machines
* Adding new needles
* Adding new rolls
* Adding new departments
* Adding new managers and assigning departments to them
* Creating and modifying the operator account and giving permissions
* Needle Distribute
* Needle refill
* Needle Request
* Check available stock
* Check the needle request and available quantity
* Distribute requested needles

# User Accounts

* Nirmal – Super Admin
* Vindana - Admin
* Hansani - Observer
* Prasangi- Observer
* Madushanka - Admin
* Thushara - Stores
* Lorena - Operator
* Subashini - Operator
* TM1 - Operator
* TM2 - Operator
* Kobeigane – Operator

# Work Flow

Starting from the super admin, Once super admin created the

* Admin account
* Manager account
* Operator account
* Observer account
* Departments

Super admin need to assign the rolls and departments to the needed accounts. After that admin need to add the entire machine models, needle models. After admin have to map the needle types that needed to all the specific machines and departments.

After that operator can start their work from their side, operator need to scan the needle by using given android phone that installed droidcam after the scan the captured image and needle details need to enter after entering operator need to request approval from their manager in order to request new needle replacement.

After operator placed the request Stores user can proceed to distribute the requested needles and give the order request to the purchasing department. When new needle received from purchasing department stores user need to mark that new needle was received.

In each and every stage needle quantity need to change accordingly

After everything the observer accounts can check all the transactions and all the reports

## Needle Management System - Final Project Report

This document provides a complete technical summary and deployment guide for the Needle Management System application. It includes the full database schema, server configuration steps for enabling HTTPS, and the complete source code for all PHP files.

### Section 1: Database Setup (SQL)

Run the following SQL queries in phpMyAdmin or any MySQL client to create the database and all necessary tables.

-- Create the database

CREATE DATABASE `needle\_management\_system`;

USE `needle\_management\_system`;

-- Create the roles table

CREATE TABLE `roles` (

`role\_id` int(11) NOT NULL AUTO\_INCREMENT,

`role\_name` varchar(50) NOT NULL,

PRIMARY KEY (`role\_id`),

UNIQUE KEY `role\_name` (`role\_name`)

) ENGINE=InnoDB;

-- Insert the roles

INSERT INTO `roles` (`role\_name`) VALUES

('Super Admin'), ('Admin'), ('Manager'), ('Operator'), ('Stores'), ('Purchasing'), ('Observer');

-- Create the departments table

CREATE TABLE `departments` (

`dept\_id` int(11) NOT NULL AUTO\_INCREMENT,

`dept\_name` varchar(100) NOT NULL,

PRIMARY KEY (`dept\_id`),

UNIQUE KEY `dept\_name` (`dept\_name`)

) ENGINE=InnoDB;

-- Create the users table

CREATE TABLE `users` (

`user\_id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(50) NOT NULL,

`full\_name` varchar(100) NOT NULL,

`password\_hash` varchar(255) NOT NULL,

`role\_id\_fk` int(11) NOT NULL,

`dept\_id\_fk` int(11) DEFAULT NULL,

`is\_active` tinyint(1) NOT NULL DEFAULT 1,

`created\_at` timestamp NOT NULL DEFAULT current\_timestamp(),

PRIMARY KEY (`user\_id`),

UNIQUE KEY `username` (`username`),

KEY `role\_id\_fk` (`role\_id\_fk`),

KEY `dept\_id\_fk` (`dept\_id\_fk`),

CONSTRAINT `users\_ibfk\_1` FOREIGN KEY (`role\_id\_fk`) REFERENCES `roles` (`role\_id`),

CONSTRAINT `users\_ibfk\_2` FOREIGN KEY (`dept\_id\_fk`) REFERENCES `departments` (`dept\_id`)

) ENGINE=InnoDB;

-- Create the machines table

CREATE TABLE `machines` (

`model\_id` int(11) NOT NULL AUTO\_INCREMENT,

`model\_name` varchar(100) NOT NULL,

`serial\_number` varchar(100) NOT NULL,

`model\_type` varchar(50) DEFAULT NULL,

PRIMARY KEY (`model\_id`),

UNIQUE KEY `serial\_number` (`serial\_number`)

) ENGINE=InnoDB;

-- Create the needle\_types table

CREATE TABLE `needle\_types` (

`needle\_type\_id` int(11) NOT NULL AUTO\_INCREMENT,

`needle\_sku` varchar(100) NOT NULL,

`needle\_size` varchar(50) NOT NULL,

`manufacturer` varchar(100) DEFAULT NULL,

PRIMARY KEY (`needle\_type\_id`),

UNIQUE KEY `needle\_sku` (`needle\_sku`)

) ENGINE=InnoDB;

-- Create the machine\_needle\_compatibility table

CREATE TABLE `machine\_needle\_compatibility` (

`map\_id` int(11) NOT NULL AUTO\_INCREMENT,

`machine\_id\_fk` int(11) NOT NULL,

`needle\_type\_id\_fk` int(11) NOT NULL,

PRIMARY KEY (`map\_id`),

UNIQUE KEY `machine\_needle\_pair` (`machine\_id\_fk`,`needle\_type\_id\_fk`),

KEY `needle\_type\_id\_fk` (`needle\_type\_id\_fk`),

CONSTRAINT `machine\_needle\_compatibility\_ibfk\_1` FOREIGN KEY (`machine\_id\_fk`) REFERENCES `machines` (`model\_id`),

CONSTRAINT `machine\_needle\_compatibility\_ibfk\_2` FOREIGN KEY (`needle\_type\_id\_fk`) REFERENCES `needle\_types` (`needle\_type\_id`)

) ENGINE=InnoDB;

-- Create the needle\_requests table

CREATE TABLE `needle\_requests` (

`request\_id` int(11) NOT NULL AUTO\_INCREMENT,

`requesting\_user\_id\_fk` int(11) NOT NULL,

`machine\_model\_id\_fk` int(11) NOT NULL,

`needle\_type\_id\_fk` int(11) NOT NULL,

`quantity\_requested` int(11) NOT NULL,

`change\_reason\_type` varchar(50) DEFAULT NULL COMMENT 'e.g., Broken, Blunted',

`change\_reason` text DEFAULT NULL,

`broken\_needle\_image\_path` varchar(255) NOT NULL,

`request\_status` varchar(50) NOT NULL DEFAULT 'Pending Approval',

`manager\_approver\_id\_fk` int(11) DEFAULT NULL,

`stores\_issuer\_id\_fk` int(11) DEFAULT NULL,

`created\_at` timestamp NOT NULL DEFAULT current\_timestamp(),

`approved\_at` timestamp NULL DEFAULT NULL,

`issued\_at` timestamp NULL DEFAULT NULL,

PRIMARY KEY (`request\_id`),

KEY `requesting\_user\_id\_fk` (`requesting\_user\_id\_fk`),

KEY `machine\_model\_id\_fk` (`machine\_model\_id\_fk`),

KEY `needle\_type\_id\_fk` (`needle\_type\_id\_fk`),

KEY `manager\_approver\_id\_fk` (`manager\_approver\_id\_fk`),

KEY `stores\_issuer\_id\_fk` (`stores\_issuer\_id\_fk`),

CONSTRAINT `needle\_requests\_ibfk\_1` FOREIGN KEY (`requesting\_user\_id\_fk`) REFERENCES `users` (`user\_id`),

CONSTRAINT `needle\_requests\_ibfk\_2` FOREIGN KEY (`machine\_model\_id\_fk`) REFERENCES `machines` (`model\_id`),

CONSTRAINT `needle\_requests\_ibfk\_3` FOREIGN KEY (`needle\_type\_id\_fk`) REFERENCES `needle\_types` (`needle\_type\_id`),

CONSTRAINT `needle\_requests\_ibfk\_4` FOREIGN KEY (`manager\_approver\_id\_fk`) REFERENCES `users` (`user\_id`),

CONSTRAINT `needle\_requests\_ibfk\_5` FOREIGN KEY (`stores\_issuer\_id\_fk`) REFERENCES `users` (`user\_id`)

) ENGINE=InnoDB;

-- Create the needle\_inventory table

CREATE TABLE `needle\_inventory` (

`inventory\_id` int(11) NOT NULL AUTO\_INCREMENT,

`needle\_type\_id\_fk` int(11) NOT NULL,

`quantity` int(11) NOT NULL,

`last\_updated` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

PRIMARY KEY (`inventory\_id`),

UNIQUE KEY `needle\_type\_id\_fk` (`needle\_type\_id\_fk`),

CONSTRAINT `needle\_inventory\_ibfk\_1` FOREIGN KEY (`needle\_type\_id\_fk`) REFERENCES `needle\_types` (`needle\_type\_id`)

) ENGINE=InnoDB;

-- Create the purchase\_requests table

CREATE TABLE `purchase\_requests` (

`prequest\_id` int(11) NOT NULL AUTO\_INCREMENT,

`needle\_type\_id\_fk` int(11) NOT NULL,

`quantity\_requested` int(11) NOT NULL,

`requesting\_user\_id\_fk` int(11) NOT NULL,

`request\_notes` text DEFAULT NULL,

`request\_status` varchar(50) NOT NULL DEFAULT 'Pending',

`created\_at` timestamp NOT NULL DEFAULT current\_timestamp(),

PRIMARY KEY (`prequest\_id`),

KEY `needle\_type\_id\_fk` (`needle\_type\_id\_fk`),

KEY `requesting\_user\_id\_fk` (`requesting\_user\_id\_fk`),

CONSTRAINT `purchase\_requests\_ibfk\_1` FOREIGN KEY (`needle\_type\_id\_fk`) REFERENCES `needle\_types` (`needle\_type\_id`),

CONSTRAINT `purchase\_requests\_ibfk\_2` FOREIGN KEY (`requesting\_user\_id\_fk`) REFERENCES `users` (`user\_id`)

) ENGINE=InnoDB;

-- Create the purchase\_orders table

CREATE TABLE `purchase\_orders` (

`order\_id` int(11) NOT NULL AUTO\_INCREMENT,

`purchase\_request\_id\_fk` int(11) DEFAULT NULL,

`needle\_type\_id\_fk` int(11) NOT NULL,

`quantity\_ordered` int(11) NOT NULL,

`supplier\_name` varchar(255) DEFAULT NULL,

`order\_notes` text DEFAULT NULL,

`order\_status` varchar(50) NOT NULL DEFAULT 'Placed',

`placing\_user\_id\_fk` int(11) NOT NULL,

`placed\_at` timestamp NOT NULL DEFAULT current\_timestamp(),

`received\_at` timestamp NULL DEFAULT NULL,

PRIMARY KEY (`order\_id`),

KEY `needle\_type\_id\_fk` (`needle\_type\_id\_fk`),

KEY `placing\_user\_id\_fk` (`placing\_user\_id\_fk`),

KEY `purchase\_request\_id\_fk` (`purchase\_request\_id\_fk`),

CONSTRAINT `purchase\_orders\_ibfk\_1` FOREIGN KEY (`needle\_type\_id\_fk`) REFERENCES `needle\_types` (`needle\_type\_id`),

CONSTRAINT `purchase\_orders\_ibfk\_2` FOREIGN KEY (`placing\_user\_id\_fk`) REFERENCES `users` (`user\_id`),

CONSTRAINT `purchase\_orders\_ibfk\_3` FOREIGN KEY (`purchase\_request\_id\_fk`) REFERENCES `purchase\_requests` (`prequest\_id`) ON DELETE SET NULL

) ENGINE=InnoDB;

### Section 2: Server Configuration (HTTPS Setup)

To enable webcam access, your server must use a secure https:// connection.

#### For Linux (Ubuntu/Debian with standard Apache)

1. **Enable the SSL Module:**
2. sudo a2enmod ssl
3. **Create a Directory for Certificates:**
4. sudo mkdir -p /etc/apache2/ssl
5. **Generate the Certificate and Key:**
6. sudo openssl req -x509 -newkey rsa:4096 -keyout /etc/apache2/ssl/server.key -out /etc/apache2/ssl/server.crt -days 365 -nodes

When asked for the "Common Name", enter your server's IP address (e.g., *10.0.10.70*).

1. **Configure the SSL Virtual Host:**
   * Edit the file: sudo nano /etc/apache2/sites-available/default-ssl.conf
   * Find and update these two lines to point to your new certificate files:
   * SSLCertificateFile /etc/apache2/ssl/server.crt
   * SSLCertificateKeyFile /etc/apache2/ssl/server.key
   * Save and close the file.
2. **Enable the SSL Site and Restart Apache:**
3. sudo a2ensite default-ssl.conf
4. sudo systemctl restart apache2

#### For Windows (XAMPP)

1. **Open Command Prompt as Administrator.**
2. **Navigate to the Apache** bin **directory:**
3. cd C:\xampp\apache\bin
4. **Generate the Certificate:**
5. openssl req -x509 -newkey rsa:4096 -keyout server.key -out server.crt -days 365 -nodes

When asked for the "Common Name", enter your server's IP address (e.g., *10.0.10.70*).

1. **Move the Certificate Files:**
   * Create folders: C:\xampp\apache\conf\ssl.key and C:\xampp\apache\conf\ssl.crt
   * Move server.key to C:\xampp\apache\conf\ssl.key\
   * Move server.crt to C:\xampp\apache\conf\ssl.crt\
2. **Configure Apache:**
   * Edit C:\xampp\apache\conf\extra\httpd-ssl.conf.
   * Update the paths:
   * SSLCertificateFile "conf/ssl.crt/server.crt"
   * SSLCertificateKeyFile "conf/ssl.key/server.key"
   * Edit C:\xampp\apache\conf\httpd.conf.
   * Make sure these lines are uncommented (no # at the beginning):
   * LoadModule ssl\_module modules/mod\_ssl.so
   * Include conf/extra/httpd-ssl.conf
3. **Restart Apache** from the XAMPP Control Panel.

### Section 3: PHP Application Code

This section contains the complete source code for all the PHP files in the application.

File structure:

/needle-system/

|-- add\_compatibility\_process.php

|-- add\_department\_process.php

|-- add\_machine\_process.php

|-- add\_needle\_process.php

|-- add\_order\_process.php

|-- add\_purchase\_request\_process.php

|-- add\_request\_process.php

|-- add\_stock\_process.php

|-- add\_user\_process.php

|-- approve\_requests.php

|-- dashboard.php

|-- db\_connect.php

|-- fulfill\_requests.php

|-- get\_compatible\_needles.php

|-- index.php

|-- login\_process.php

|-- logout.php

|-- manage\_compatibility.php

|-- manage\_departments.php

|-- manage\_inventory.php

|-- manage\_machines.php

|-- manage\_needles.php

|-- manage\_users.php

|-- my\_requests.php

|-- nav.php

|-- new\_request.php

|-- place\_orders.php

|-- request\_purchase.php

|-- style.css

|-- update\_fulfillment\_status.php

|-- update\_order\_status.php

|-- update\_request\_status.php

|-- view\_all\_requests.php

|-- view\_all\_transactions.php

|-- view\_department\_reports.php

|-- /uploads/ (This folder must be created and be writable by the server)

```php

<?php

// =================================================================

// FILE: add\_compatibility\_process.php

// =================================================================

session\_start();

require\_once 'db\_connect.php';

// --- Protection ---

if (!isset($\_SESSION['user\_id'])) {

header("Location: index.php");

exit();

}

$allowed\_roles = ['Super Admin', 'Admin'];

if (!in\_array($\_SESSION['role\_name'], $allowed\_roles)) {

header("Location: dashboard.php");

exit();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$machine\_id = $\_POST['machine\_id'];

$needle\_type\_id = $\_POST['needle\_type\_id'];

if (empty($machine\_id) || empty($needle\_type\_id)) {

header("Location: manage\_compatibility.php?status=error&message=" . urlencode("Both a machine and a needle must be selected."));

exit();

}

$stmt\_check = $conn->prepare("SELECT map\_id FROM machine\_needle\_compatibility WHERE machine\_id\_fk = ? AND needle\_type\_id\_fk = ?");

$stmt\_check->bind\_param("ii", $machine\_id, $needle\_type\_id);

$stmt\_check->execute();

$stmt\_check->store\_result();

if ($stmt\_check->num\_rows > 0) {

$stmt\_check->close();

header("Location: manage\_compatibility.php?status=error&message=" . urlencode("This link already exists."));

exit();

}

$stmt\_check->close();

$sql = "INSERT INTO machine\_needle\_compatibility (machine\_id\_fk, needle\_type\_id\_fk) VALUES (?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("ii", $machine\_id, $needle\_type\_id);

if ($stmt->execute()) {

header("Location: manage\_compatibility.php?status=success&message=" . urlencode("Compatibility link added successfully."));

} else {

header("Location: manage\_compatibility.php?status=error&message=" . urlencode("Failed to add link: " . $stmt->error));

}

$stmt->close();

$conn->close();

} else {

header("Location: manage\_compatibility.php");

exit();

}

?>

<?php

// =================================================================

// FILE: add\_department\_process.php

// =================================================================

session\_start();

require\_once 'db\_connect.php';

// --- Protection ---

if (!isset($\_SESSION['user\_id'])) {

header("Location: index.php");

exit();

}

if ($\_SESSION['role\_name'] !== 'Super Admin') {

header("Location: dashboard.php");

exit();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$dept\_name = trim($\_POST['dept\_name']);

if (empty($dept\_name)) {

header("Location: manage\_departments.php?status=error&message=" . urlencode("Department name cannot be empty."));

exit();

}

$stmt\_check = $conn->prepare("SELECT dept\_id FROM departments WHERE dept\_name = ?");

$stmt\_check->bind\_param("s", $dept\_name);

$stmt\_check->execute();

$stmt\_check->store\_result();

if ($stmt\_check->num\_rows > 0) {

$stmt\_check->close();

header("Location: manage\_departments.php?status=error&message=" . urlencode("Department name already exists."));

exit();

}

$stmt\_check->close();

$sql = "INSERT INTO departments (dept\_name) VALUES (?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("s", $dept\_name);

if ($stmt->execute()) {

header("Location: manage\_departments.php?status=success&message=" . urlencode("Department added successfully."));

} else {

header("Location: manage\_departments.php?status=error&message=" . urlencode("Failed to add department: " . $stmt->error));

}

$stmt->close();

$conn->close();

} else {

header("Location: manage\_departments.php");

exit();

}

?>

<?php

// =================================================================

// FILE: add\_machine\_process.php

// =================================================================

session\_start();

require\_once 'db\_connect.php';

// --- Protection ---

if (!isset($\_SESSION['user\_id'])) {

header("Location: index.php");

exit();

}

$allowed\_roles = ['Super Admin', 'Admin'];

if (!in\_array($\_SESSION['role\_name'], $allowed\_roles)) {

header("Location: dashboard.php");

exit();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$model\_name = trim($\_POST['model\_name']);

$serial\_number = trim($\_POST['serial\_number']);

$model\_type = trim($\_POST['model\_type']);

if (empty($model\_name) || empty($serial\_number) || empty($model\_type)) {

header("Location: manage\_machines.php?status=error&message=" . urlencode("All fields are required."));

exit();

}

$stmt\_check = $conn->prepare("SELECT model\_id FROM machines WHERE serial\_number = ?");

$stmt\_check->bind\_param("s", $serial\_number);

$stmt\_check->execute();

$stmt\_check->store\_result();

if ($stmt\_check->num\_rows > 0) {

$stmt\_check->close();

header("Location: manage\_machines.php?status=error&message=" . urlencode("A machine with this serial number already exists."));

exit();

}

$stmt\_check->close();

$sql = "INSERT INTO machines (model\_name, serial\_number, model\_type) VALUES (?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("sss", $model\_name, $serial\_number, $model\_type);

if ($stmt->execute()) {

header("Location: manage\_machines.php?status=success&message=" . urlencode("Machine added successfully."));

} else {

header("Location: manage\_machines.php?status=error&message=" . urlencode("Failed to add machine: " . $stmt->error));

}

$stmt->close();

$conn->close();

} else {

header("Location: manage\_machines.php");

exit();

}

?>

<?php

// =================================================================

// FILE: add\_needle\_process.php

// =================================================================

session\_start();

require\_once 'db\_connect.php';

// --- Protection ---

if (!isset($\_SESSION['user\_id'])) {

header("Location: index.php");

exit();

}

$allowed\_roles = ['Super Admin', 'Admin'];

if (!in\_array($\_SESSION['role\_name'], $allowed\_roles)) {

header("Location: dashboard.php");

exit();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$needle\_sku = trim($\_POST['needle\_sku']);

$needle\_size = trim($\_POST['needle\_size']);

$manufacturer = !empty(trim($\_POST['manufacturer'])) ? trim($\_POST['manufacturer']) : NULL;

if (empty($needle\_sku) || empty($needle\_size)) {

header("Location: manage\_needles.php?status=error&message=" . urlencode("SKU and Size are required."));

exit();

}

$stmt\_check = $conn->prepare("SELECT needle\_type\_id FROM needle\_types WHERE needle\_sku = ?");

$stmt\_check->bind\_param("s", $needle\_sku);

$stmt\_check->execute();

$stmt\_check->store\_result();

if ($stmt\_check->num\_rows > 0) {

$stmt\_check->close();

header("Location: manage\_needles.php?status=error&message=" . urlencode("A needle with this SKU already exists."));

exit();

}

$stmt\_check->close();

$sql = "INSERT INTO needle\_types (needle\_sku, needle\_size, manufacturer) VALUES (?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("sss", $needle\_sku, $needle\_size, $manufacturer);

if ($stmt->execute()) {

header("Location: manage\_needles.php?status=success&message=" . urlencode("Needle type added successfully."));

} else {

header("Location: manage\_needles.php?status=error&message=" . urlencode("Failed to add needle type: " . $stmt->error));

}

$stmt->close();

$conn->close();

} else {

header("Location: manage\_needles.php");

exit();

}

?>

<?php

// =================================================================

// FILE: add\_order\_process.php

// UPDATED - Now handles linking a purchase order to a purchase request.

// =================================================================

session\_start();

require\_once 'db\_connect.php';

// --- Protection ---

if (!isset($\_SESSION['user\_id'])) {

header("Location: index.php");

exit();

}

$allowed\_roles = ['Super Admin', 'Purchasing'];

if (!in\_array($\_SESSION['role\_name'], $allowed\_roles)) {

header("Location: dashboard.php");

exit();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$needle\_type\_id = $\_POST['needle\_type\_id'];

$quantity\_ordered = $\_POST['quantity\_ordered'];

$supplier\_name = !empty(trim($\_POST['supplier\_name'])) ? trim($\_POST['supplier\_name']) : NULL;

$order\_notes = !empty(trim($\_POST['order\_notes'])) ? trim($\_POST['order\_notes']) : NULL;

$placing\_user\_id = $\_SESSION['user\_id'];

$purchase\_request\_id = isset($\_POST['purchase\_request\_id']) ? $\_POST['purchase\_request\_id'] : NULL;

// --- Validation ---

if (empty($needle\_type\_id) || empty($quantity\_ordered) || !is\_numeric($quantity\_ordered) || $quantity\_ordered <= 0) {

header("Location: place\_orders.php?status=error&message=" . urlencode("Please select a needle and enter a valid quantity."));

exit();

}

$conn->begin\_transaction();

try {

// 1. Insert the new purchase order

$sql\_order = "INSERT INTO purchase\_orders (purchase\_request\_id\_fk, needle\_type\_id\_fk, quantity\_ordered, supplier\_name, order\_notes, placing\_user\_id\_fk)

VALUES (?, ?, ?, ?, ?, ?)";

$stmt\_order = $conn->prepare($sql\_order);

$stmt\_order->bind\_param("iisssi", $purchase\_request\_id, $needle\_type\_id, $quantity\_ordered, $supplier\_name, $order\_notes, $placing\_user\_id);

$stmt\_order->execute();

$stmt\_order->close();

// 2. If it came from a purchase request, update that request's status

if ($purchase\_request\_id) {

$sql\_pr = "UPDATE purchase\_requests SET request\_status = 'Ordered' WHERE prequest\_id = ?";

$stmt\_pr = $conn->prepare($sql\_pr);

$stmt\_pr->bind\_param("i", $purchase\_request\_id);

$stmt\_pr->execute();

$stmt\_pr->close();

}

$conn->commit();

header("Location: place\_orders.php?status=success&message=" . urlencode("Purchase order placed successfully."));

} catch (Exception $e) {

$conn->rollback();

header("Location: place\_orders.php?status=error&message=" . urlencode("Database error: " . $e->getMessage()));

}

$conn->close();

} else {

header("Location: place\_orders.php");

exit();

}

?>

<?php

// =================================================================

// FILE: add\_purchase\_request\_process.php

// NEW FILE - Handles the purchase request logic.

// =================================================================

session\_start();

require\_once 'db\_connect.php';

// --- Protection ---

if (!isset($\_SESSION['user\_id'])) {

header("Location: index.php");

exit();

}

$allowed\_roles = ['Super Admin', 'Stores'];

if (!in\_array($\_SESSION['role\_name'], $allowed\_roles)) {

header("Location: dashboard.php");

exit();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$needle\_type\_id = $\_POST['needle\_type\_id'];

$quantity\_requested = $\_POST['quantity\_requested'];

$request\_notes = !empty(trim($\_POST['request\_notes'])) ? trim($\_POST['request\_notes']) : NULL;

$requesting\_user\_id = $\_SESSION['user\_id'];

// --- Validation ---

if (empty($needle\_type\_id) || empty($quantity\_requested) || !is\_numeric($quantity\_requested) || $quantity\_requested <= 0) {

header("Location: request\_purchase.php?status=error&message=" . urlencode("Please select a needle and enter a valid quantity."));

exit();

}

// --- Insert into Database ---

$sql = "INSERT INTO purchase\_requests (needle\_type\_id\_fk, quantity\_requested, request\_notes, requesting\_user\_id\_fk)

VALUES (?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("iisi", $needle\_type\_id, $quantity\_requested, $request\_notes, $requesting\_user\_id);

if ($stmt->execute()) {

header("Location: request\_purchase.php?status=success&message=" . urlencode("Purchase request submitted successfully."));

} else {

header("Location: request\_purchase.php?status=error&message=" . urlencode("Database error: " . $stmt->error));

}

$stmt->close();

$conn->close();

} else {

header("Location: request\_purchase.php");

exit();

}

?>

<?php

// =================================================================

// FILE: add\_request\_process.php

// UPDATED - Now handles the new "change\_reason\_type" field.

// =================================================================

session\_start();

require\_once 'db\_connect.php';

// --- Protection ---

if (!isset($\_SESSION['user\_id'])) {

header("Location: index.php");

exit();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// --- Get Form Data ---

$requesting\_user\_id = $\_SESSION['user\_id'];

$machine\_id = $\_POST['machine\_id'];

$needle\_type\_id = $\_POST['needle\_type\_id'];

$quantity = $\_POST['quantity'];

$change\_reason\_type = $\_POST['change\_reason\_type']; // New field

$change\_reason = trim($\_POST['change\_reason']); // Optional details

$captured\_image\_data = $\_POST['captured\_image\_data'];

$image\_file = $\_FILES['broken\_needle\_image'];

$image\_path = null;

// --- Validation ---

if (empty($machine\_id) || empty($needle\_type\_id) || empty($quantity) || empty($change\_reason\_type)) {

header("Location: new\_request.php?status=error&message=" . urlencode("Please fill in all required fields."));

exit();

}

// --- Image Handling ---

$upload\_dir = '/var/www/html/needle-system/uploads/';

// Prioritize captured base64 data from webcam

if (!empty($captured\_image\_data)) {

$img\_data = str\_replace('data:image/jpeg;base64,', '', $captured\_image\_data);

$img\_data = str\_replace(' ', '+', $img\_data);

$decoded\_image = base64\_decode($img\_data);

$new\_file\_name = uniqid('', true) . '.jpg';

$destination = $upload\_dir . $new\_file\_name;

if (file\_put\_contents($destination, $decoded\_image)) {

$image\_path = 'uploads/' . $new\_file\_name;

} else {

header("Location: new\_request.php?status=error&message=" . urlencode("Failed to save captured image. Check folder permissions."));

exit();

}

}

// Fallback to file upload

elseif (isset($image\_file) && $image\_file['error'] === UPLOAD\_ERR\_OK) {

$file\_tmp\_name = $image\_file['tmp\_name'];

$file\_name = $image\_file['name'];

$file\_ext = strtolower(pathinfo($file\_name, PATHINFO\_EXTENSION));

$allowed\_ext = ['jpg', 'jpeg', 'png', 'gif'];

if (in\_array($file\_ext, $allowed\_ext)) {

$new\_file\_name = uniqid('', true) . '.' . $file\_ext;

$destination = $upload\_dir . $new\_file\_name;

if (move\_uploaded\_file($file\_tmp\_name, $destination)) {

$image\_path = 'uploads/' . $new\_file\_name;

} else {

header("Location: new\_request.php?status=error&message=" . urlencode("Failed to move uploaded file. Check permissions."));

exit();

}

} else {

header("Location: new\_request.php?status=error&message=" . urlencode("Invalid file type."));

exit();

}

} else {

header("Location: new\_request.php?status=error&message=" . urlencode("You must provide an image via Webcam or file upload."));

exit();

}

// --- Insert into Database ---

$sql = "INSERT INTO needle\_requests (requesting\_user\_id\_fk, machine\_model\_id\_fk, needle\_type\_id\_fk, quantity\_requested, change\_reason\_type, change\_reason, broken\_needle\_image\_path)

VALUES (?, ?, ?, ?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("iiissss", $requesting\_user\_id, $machine\_id, $needle\_type\_id, $quantity, $change\_reason\_type, $change\_reason, $image\_path);

if ($stmt->execute()) {

header("Location: new\_request.php?status=success&message=" . urlencode("Request submitted successfully."));

} else {

header("Location: new\_request.php?status=error&message=" . urlencode("Failed to submit request: " . $stmt->error));

}

$stmt->close();

$conn->close();

} else {

header("Location: new\_request.php");

exit();

}

?>

<?php

// =================================================================

// FILE: add\_stock\_process.php

// UPDATED - Uses a safer SQL query to prevent race conditions.

// =================================================================

session\_start();

require\_once 'db\_connect.php';

// --- Protection ---

if (!isset($\_SESSION['user\_id'])) {

header("Location: index.php");

exit();

}

$allowed\_roles = ['Super Admin', 'Stores'];

if (!in\_array($\_SESSION['role\_name'], $allowed\_roles)) {

header("Location: dashboard.php");

exit();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$needle\_type\_id = $\_POST['needle\_type\_id'];

$quantity\_to\_add = $\_POST['quantity\_to\_add'];

// --- Validation ---

if (empty($needle\_type\_id) || empty($quantity\_to\_add) || !is\_numeric($quantity\_to\_add) || $quantity\_to\_add <= 0) {

header("Location: manage\_inventory.php?status=error&message=" . urlencode("Please select a needle and enter a valid quantity."));

exit();

}

// --- Use INSERT ... ON DUPLICATE KEY UPDATE for a single, safe query ---

$sql = "INSERT INTO needle\_inventory (needle\_type\_id\_fk, quantity) VALUES (?, ?)

ON DUPLICATE KEY UPDATE quantity = quantity + VALUES(quantity)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("ii", $needle\_type\_id, $quantity\_to\_add);

if ($stmt->execute()) {

header("Location: manage\_inventory.php?status=success&message=" . urlencode("Stock updated successfully."));

} else {

header("Location: manage\_inventory.php?status=error&message=" . urlencode("Database error: " . $stmt->error));

}

$stmt->close();

$conn->close();

} else {

header("Location: manage\_inventory.php");

exit();

}

?>

<?php

// =================================================================

// FILE: logout.php

// This file destroys the session and logs the user out.

// =================================================================

session\_start();

$\_SESSION = array();

session\_destroy();

header("location: index.php");

exit;

?>