# **Pharmacy Portal**

### 1. Developed By

Name: Suthfan Ahmed

### 2. Abstract

This project is a web-based application built using PHP and MySQL, designed to help pharmacists and patients manage medications, prescriptions, inventory, and sales effectively. The system features role-based access, automated inventory updates, and stored procedures for backend efficiency.

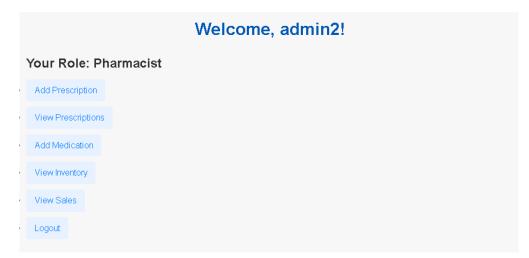
### 3. Objectives

- Provide a user-friendly portal for pharmacists and patients
- Enable secure login and role-based access control
- Maintain and manage medication inventory in real-time
- Automate record-keeping for prescriptions and sales
- Implement MySQL views, procedures, and triggers for business logic

# 4. System Screenshots

### **Home Page**

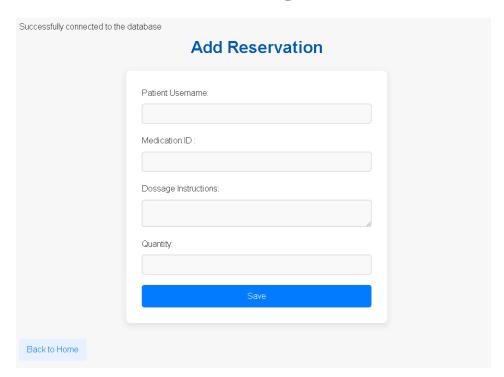




## **Login Page**



## **Add Prescription**



# **View Prescriptions**

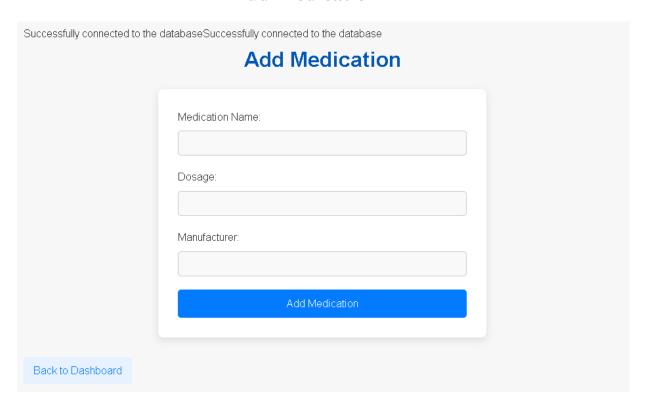
Successfully connected to the database

# **All Prescriptions**

Prescription ID	User ID	Medication ID	Medication Name	Dosage Instructions	Quantity
1	1	1	Paracetamol	Take one tablet every 4 hours	30
2	1	2	Ibuprofen	Take one tablet every 6 hours	20
13	1	1	Paracetamol	Take 1 tablet every 6 hours	30
14	2	2	Ibuprofen	Take 1 tablet daily	60

Back to Home

#### **Add Medication**



### **View Inventory**

Successfully connected to the database

### **Medication Inventory**

Medication Name	Dosage	Manufacturer	Quantity Available
Paracetamol	500mg	ABC Pharma	100
lbuprofen	200mg	XYZ Labs	150
Paracetamol	500mg	ABC Pharma	200
lbuprofen	200mg	XYZ Labs	150
Aspirin	325mg	HealthCorp	120
Amoxicillin	250mg	MediCo	80
Metformin	500mg	PharmaTech	100
Ciprofloxacin	500mg	MediHealth	200
Lisinopril	10mg	BioPharm	180
Omeprazole	20mg	PharmaBest	150
Simvastatin	20mg	TopMed	50
Furosemide	40mg	QuickPharm	300

Back to Dashboard

#### **View Sales**

Successfully connected to the database// viewSales.php Successfully connected to the database

#### Sales Data

Sale ID	Prescription ID	Sale Date	Quantity Sold	Sale Amount	Dosage Instructions	Medication Name
1	1	2025-05-05 23:28:22	2	20.00	Take one tablet every 4 hours	Paracetamol
12	1	2025-05-01 14:00:00	10	50.00	Take one tablet every 4 hours	Paracetamol
13	2	2025-05-02 16:30:00	5	25.00	Take one tablet every 6 hours	lbuprofen

Back to Home

### 5. Database Schema

#### Users

userId INT PRIMARY KEY AUTO\_INCREMENT userName VARCHAR(45) UNIQUE contactInfo VARCHAR(200) userType ENUM('pharmacist', 'patient') password VARCHAR(255)

#### **Medications**

medicationId INT PRIMARY KEY AUTO\_INCREMENT medicationName VARCHAR(45) dosage VARCHAR(45) manufacturer VARCHAR(100)

### **Prescriptions**

prescriptionId INT PRIMARY KEY AUTO\_INCREMENT userId INT medicationId INT prescribedDate DATETIME dosageInstructions VARCHAR(200) quantity INT refillCount INT DEFAULT 0

#### **Inventory**

inventoryId INT PRIMARY KEY AUTO\_INCREMENT medicationId INT quantityAvailable INT lastUpdated DATETIME

#### Sales

saleId INT PRIMARY KEY AUTO\_INCREMENT prescriptionId INT saleDate DATETIME quantitySold INT saleAmount DECIMAL(10,2)

### 6. Backend Features

- Stored Procedure: AddOrUpdateUser Manages creation or update of user records.
- Stored Procedure: ProcessSale Handles sales logic and inventory reduction.
- View: MedicationInventoryView Shows medication and realtime stock levels.
- Trigger: AfterPrescriptionInsert Reduces inventory automatically after new prescription.

### 7. Code Explanation

### PharmacyDatabase.php

The "PharmacyDatabase" class centralizes all database operations. It ensures separation of concerns by isolating database logic from UI logic.

- "connect()": Creates a mysqli connection using the credentials provided. If there's an error, it halts execution.
- "addPrescription()": First validates if the patient exists using username, then inserts prescription info tied to that user and a medication.
- "addUser()": Accepts name, contact, and role. Uses a stored procedure "AddOrUpdateUser" if implemented in SQL.
- "addMedication()": Adds new medication entries to the medications table. Could also check for duplicates.
- "MedicationInventory()": Executes a view query from "MedicationInventoryView" and returns the aggregated data.
- "validateUser()": Used in login. It retrieves a hashed password from the database and verifies the input against it using "password\_verify()".

### PharmacyServer.php

Serves as the core router/controller for web requests. Based on "\$\_GET['action']", different functions are invoked:

- "addPrescription()" and others call the relevant template and handle submitted form data.
- "handleRequest()" acts like a front controller deciding which functionality to activate.
- This structure makes the code modular and easier to extend (e.g., adding "viewSales" later).

#### login.php

Handles login logic.

- First checks if POST data is present (i.e., login attempt).

- It calls "validateUser()" to verify credentials.
- Based on the returned user role, it redirects to a different dashboard or access page.
- Secure session management is used with "session\_start()" and role-based routing.

### Templates (Forms and Views)

- Each template uses basic HTML with embedded PHP to dynamically load data.
- "addPrescription.php", "addMedication.php" use POST forms to collect user input.
- "viewPrescriptions.php", "viewInventory.php" use PHP loops to render SQL result sets as HTML tables.

#### **Triggers and Stored Procedures**

- "AfterPrescriptionInsert": Automatically updates inventory by subtracting prescription quantity after a new entry.
- "AddOrUpdateUser": Useful for login or registration when user might already exist.
- "ProcessSale": Handles sale transactions, updates inventory, and records sales in a single step for data integrity.

#### **Security Features**

- Passwords are hashed using "password\_hash()" and verified with "password\_verify()".
- Sessions protect pages from unauthorized access.
- Only logged-in users can perform sensitive actions like adding prescriptions.
- Role checks differentiate functionality access between "patient" and "pharmacist".

### **MySQL Scripts**

- Tables: Users, Medications, Prescriptions, Inventory, and Sales with foreign key relations.
- View: MedicationInventoryView shows current medication

stock.

- Procedures: AddOrUpdateUser, ProcessSale for modular database logic.
- Trigger: AfterPrescriptionInsert ensures inventory is automatically updated.

### 7. Testing & Validation

- Login and session validation tested
- Prescriptions successfully insert and show in the list
- Inventory updates tested via trigger
- View-based inventory read works accurately
- Sales tested with foreign key compliance

#### 8. Future Enhancements

- PDF exports for prescriptions and sales
- Patient email notifications
- Admin panel for analytics and management

#### 9. Conclusion

The system fulfills the objectives outlined in the project scope. It integrates PHP with MySQL effectively to handle real-world pharmacy operations while maintaining data integrity through relational design, stored procedures, views, and triggers.