

1.1 Create the Database

sql

CopyEdit

```
CREATE DATABASE pharmacy_portal_db;
```

```
USE pharmacy_portal_db;
```

1.2 Tables

sql

CopyEdit

-- Users Table

```
CREATE TABLE Users (  
    userId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    userName VARCHAR(45) NOT NULL UNIQUE,  
    contactInfo VARCHAR(200),  
    userType ENUM('pharmacist', 'patient') NOT NULL  
);
```

-- Medications Table

```
CREATE TABLE Medications (  
    medicationId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    medicationName VARCHAR(45) NOT NULL,  
    dosage VARCHAR(45) NOT NULL,  
    manufacturer VARCHAR(100)  
);
```

-- Prescriptions Table

```
CREATE TABLE Prescriptions (  
    prescriptionId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    userId INT NOT NULL,  
    medicationId INT NOT NULL,  
    prescribedDate DATETIME NOT NULL,  
    dosageInstructions VARCHAR(200),  
    quantity INT NOT NULL,  
    refillCount INT DEFAULT 0,  
    FOREIGN KEY (userId) REFERENCES Users(userId),  
    FOREIGN KEY (medicationId) REFERENCES Medications(medicationId)  
);
```

-- Inventory Table

```

CREATE TABLE Inventory (
    inventoryId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    medicationId INT NOT NULL,
    quantityAvailable INT NOT NULL,
    lastUpdated DATETIME NOT NULL,
    FOREIGN KEY (medicationId) REFERENCES Medications(medicationId)
);

-- Sales Table
CREATE TABLE Sales (
    saleId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    prescriptionId INT NOT NULL,
    saleDate DATETIME NOT NULL,
    quantitySold INT NOT NULL,
    saleAmount DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (prescriptionId) REFERENCES Prescriptions(prescriptionId)
);

```

1.3 Stored Procedures, View, Trigger

sql

CopyEdit

-- Procedure: Add or Update User

DELIMITER //

```

CREATE PROCEDURE AddOrUpdateUser (
    IN p_username VARCHAR(45),
    IN p_contactInfo VARCHAR(200),
    IN p_userType ENUM('pharmacist','patient')
)
BEGIN
    IF EXISTS (SELECT * FROM Users WHERE userName = p_username) THEN
        UPDATE Users SET contactInfo = p_contactInfo, userType = p_userType WHERE
        userName = p_username;
    ELSE
        INSERT INTO Users(userName, contactInfo, userType) VALUES (p_username,
        p_contactInfo, p_userType);
    END IF;
END //
DELIMITER ;

```

-- Procedure: Process Sale

```

DELIMITER //
CREATE PROCEDURE ProcessSale (
    IN p_prescriptionId INT,
    IN p_quantitySold INT
)
BEGIN
    DECLARE m_id INT;
    DECLARE sale_amt DECIMAL(10,2);

    SELECT medicationId INTO m_id FROM Prescriptions WHERE prescriptionId =
p_prescriptionId;

    -- Calculate a dummy sale amount (replace with real pricing logic)
    SET sale_amt = p_quantitySold * 5.00;

    -- Insert Sale
    INSERT INTO Sales(prescriptionId, saleDate, quantitySold, saleAmount)
    VALUES (p_prescriptionId, NOW(), p_quantitySold, sale_amt);

    -- Update Inventory
    UPDATE Inventory SET quantityAvailable = quantityAvailable - p_quantitySold, lastUpdated
= NOW()
    WHERE medicationId = m_id;
END //
DELIMITER ;

-- View: Medication Inventory
CREATE VIEW MedicationInventoryView AS
SELECT m.medicationName, m.dosage, m.manufacturer, i.quantityAvailable
FROM Medications m
JOIN Inventory i ON m.medicationId = i.medicationId;

-- Trigger: After Prescription Insert
DELIMITER //
CREATE TRIGGER AfterPrescriptionInsert
AFTER INSERT ON Prescriptions
FOR EACH ROW
BEGIN
    UPDATE Inventory
    SET quantityAvailable = quantityAvailable - NEW.quantity,

```

```
        lastUpdated = NOW()
WHERE medicationId = NEW.medicationId;

-- Optional: Add notification logic for low stock
END //
DELIMITER ;
```

1.4 Data Population

sql

CopyEdit

-- Users Table Population

```
INSERT INTO Users (userName, contactInfo, userType) VALUES
('john_doe', 'john@example.com', 'patient'),
('mary_smith', 'mary@example.com', 'pharmacist'),
('emma_white', 'emma@example.com', 'patient');
```

-- Medications Table Population

```
INSERT INTO Medications (medicationName, dosage, manufacturer) VALUES
('Paracetamol', '500mg', 'PharmaCorp'),
('Ibuprofen', '200mg', 'MediLife'),
('Amoxicillin', '250mg', 'HealthPlus');
```

-- Inventory Table Population

```
INSERT INTO Inventory (medicationId, quantityAvailable, lastUpdated) VALUES
(1, 100, NOW()), (2, 150, NOW()), (3, 75, NOW());
```

2. PHP Implementation

2.1 Database Connection (config.php)

php

CopyEdit

```
<?php
```

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "pharmacy_portal_db";
```

```
// Create connection
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>

```

2.2 Core Logic (PharmacyDatabase.php)

php

CopyEdit

// Add or Update User

```

function addUser($userName, $contactInfo, $userType) {
    global $conn;
    $stmt = $conn->prepare("CALL AddOrUpdateUser(?, ?, ?)");
    $stmt->bind_param("sss", $userName, $contactInfo, $userType);
    $stmt->execute();
}

```

// Add Medication

```

function addMedication($medicationName, $dosage, $manufacturer) {
    global $conn;
    $stmt = $conn->prepare("INSERT INTO Medications (medicationName, dosage,
manufacturer) VALUES (?, ?, ?)");
    $stmt->bind_param("sss", $medicationName, $dosage, $manufacturer);
    $stmt->execute();
}

```

// Get User Details

```

function getUserDetails($userId) {
    global $conn;
    $stmt = $conn->prepare("SELECT * FROM Users WHERE userId = ?");
    $stmt->bind_param("i", $userId);
    $stmt->execute();
    return $stmt->get_result()->fetch_assoc();
}

```

2.3 Secure Login (login.php)

php

CopyEdit

<?php

```

session_start();

```

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get input data
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Validate user (replace this with actual DB check)
    if ($username == 'admin' && password_verify($password, password_hash('admin',
PASSWORD_DEFAULT))) {
        $_SESSION['username'] = $username;
        header('Location: dashboard.php');
    } else {
        echo "Invalid login!";
    }
}
?>

```

```

<form method="POST">
    Username: <input type="text" name="username" required><br>
    Password: <input type="password" name="password" required><br>
    <button type="submit">Login</button>
</form>

```

2.4 Dashboard Views

Patient Dashboard (dashboard_patient.php)

```

php
CopyEdit
<?php
// Patient Dashboard Logic
echo "Welcome to your Prescription Portal!";
?>

```

Pharmacist Dashboard (dashboard_pharmacist.php)

```

php
CopyEdit
<?php
// Pharmacist Dashboard Logic
echo "Welcome to the Pharmacy Dashboard!";

```

?>

3. Frontend Structure

Basic Structure Example

html

CopyEdit

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Pharmacy Portal</title>
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <h1>Pharmacy Portal</h1>
```

```
    <nav>
```

```
      <a href="login.php">Login</a>
```

```
      <a href="register.php">Register</a>
```

```
    </nav>
```

```
  </header>
```

```
</body>
```

```
</html>
```