
# Karachi Institute of Economics & Technology

**Database Management System - 116395**
**Fall-2024**

# Pharmacy POS

**Submitted by:**

**Student ID: 66008**
**Name: Noman Shoukat**

**Student ID: 65787**
**Name: Muhammad Arif**

College of Computing & Information Sciences

**2024**

# INTRODUCTION

The Pharmacy Ordering System is an innovative solution crafted to optimize pharmacy operations, with a focus on streamlining purchasing and reporting processes. Developed using PHP and SQL Database, the system provides a comprehensive platform for managing customer transactions, inventory, and reporting from a single, user-friendly interface. Its robust features enable efficient invoice generation, precise inventory tracking, and insightful reports, ensuring seamless daily operations for pharmacy staff while delivering actionable analytics for management decision-making.

# OBJECTIVES

The primary goals of the Pharmacy Ordering System are:

**Facilitate Accurate Transactions:** Provide a reliable and user-friendly platform for pharmacy staff to process customer purchases, ensuring error-free billing and seamless invoice generation.

**Streamline Inventory Management:** Enhance inventory control by tracking stock levels in real-time, managing product expiry dates, and optimizing reordering processes to prevent shortages or overstocking.

**Centralize Pharmacy Operations:** Integrate multiple functions, such as order management, inventory control, and reporting, into a unified, easy-to-access system to reduce complexity and boost operational efficiency.

**Enable Data-Driven Insights:** Offer comprehensive reporting features that provide detailed insights into sales trends, profit margins, customer activity, and inventory turnover, supporting informed decision-making.

# FEATURES

1. **User Authentication**

   **Login System**: Pharmacy Ordering System features a secure login system that ensures authorized access for staff. Users can be created and managed by the super admin, who has full access to all system functionalities. Roles include:

- **Super Admin:** Has unrestricted access to all features and administrative functions.
- **Store Manager:** Can manage inventory, sales, and customer data but cannot perform high-level administrative tasks.
- **Employee:** Limited access to perform sales transactions and view relevant customer information.

This role-based access control ensures operational security and maintains accountability within the system.

2. **Inventory Management**

   - **Add and Manage Inventory:** Staff can efficiently add new products, update stock quantities, set pricing, and adjust costs, ensuring accurate tracking and streamlined inventory control.

3. **Sales and Billing Management**

   - **Order Creation and Management**: Staff can create new orders by selecting items, adjusting quantities, and entering customer information. The system automatically calculates totals and generates an invoice for each order, simplifying checkout and record-keeping.

4. **Reporting**

   - **Item-Specific Report**: Generate detailed reports on each item, tracking stock levels, sales volume, and profitability for individual products.
   - **Sales and Profit by Date**: View daily, weekly, or monthly reports on total sales and profits, enabling management to analyze performance trends over time.
   - **Customer Purchase History**: Access detailed reports on individual customer purchases, providing insights into buying behavior and frequent purchases.
   - **Overall Store Report**: Summarize key metrics across the store, including total sales, top-performing products, and overall profitability, offering a complete picture of store performance.

# TECH STACK

- Programming PHP
- Database: Microsoft SQL Database
- Development Environment: Apache/VS Code

# DATABASE

## Entities and Relations

### *1.* Bill Items

- **Relation:**
  - Many-to-One (Many bill_items belong to one medicine batch)
  - Many-to-One (Many bill_items belong to one medicine bill)
  - Many-to-One (Many bill_items belong to one medicine)
- **Description:**
  This table stores detailed information about the items included in a bill, such as the quantity, sale price, discounts, taxes, and the final price.

### *2.* Customers

- **Relation:**
  - Standalone entity.
- **Description:**
  This table contains customer data, including personal details like first and last name, email, mobile, and gender, as well as account status and creation date.

### *3.* Login Attempts

- **Relation:**
  - Standalone entity.
- **Description:**
  Tracks login attempts by storing user-related details such as email, IP address, and timestamps of the attempts.

### *4.* Medicine Batch

- **Relation:**
  - Many-to-One (Each batch belongs to a medicine).
  - Many-to-One (Each batch is linked to a purchase record).

- **Description:**
  Stores batch-specific data for medicines, including batch number, expiry date, stock quantity, sale/purchase price, discount details, and associated purchase records.

### *5.* Medicine Bill

- **Relation:**
  - Many-to-One (Each bill is linked to a customer).
  - Many-to-One (Each bill has a payment method).
- **Description:**
  Stores information about the bills issued for medicines, including subtotal, taxes, discounts, and the final amount. Also tracks customer and payment method information.

### *6.* Medicine Category

- **Relation:**
  - Standalone entity.
- **Description:**
  Categorizes medicines into different groups for better organization.

### *7.* Medicine Purchase

- **Relation:**
  - Many-to-One (Each purchase is linked to a supplier).
- **Description:**
  Stores records of medicine purchases, including supplier information, purchase total, tax, discount, and payment details.

### *8.* Medicines

- **Relation:**
  - Many-to-One (Each medicine belongs to a category).
- **Description:**
  This table includes details about medicines, such as name, generic name, company, tax, minimum/reorder levels, and packaging details.

### *9.* Payment Method

- **Relation:**
  - Standalone entity.
- **Description:**
  Lists different payment methods available in the system.

### *10.* Suppliers

- **Relation:**
  - Standalone entity.

- **Description:**
  Contains supplier details, such as name, email, phone, and address.

### *11.* User Role

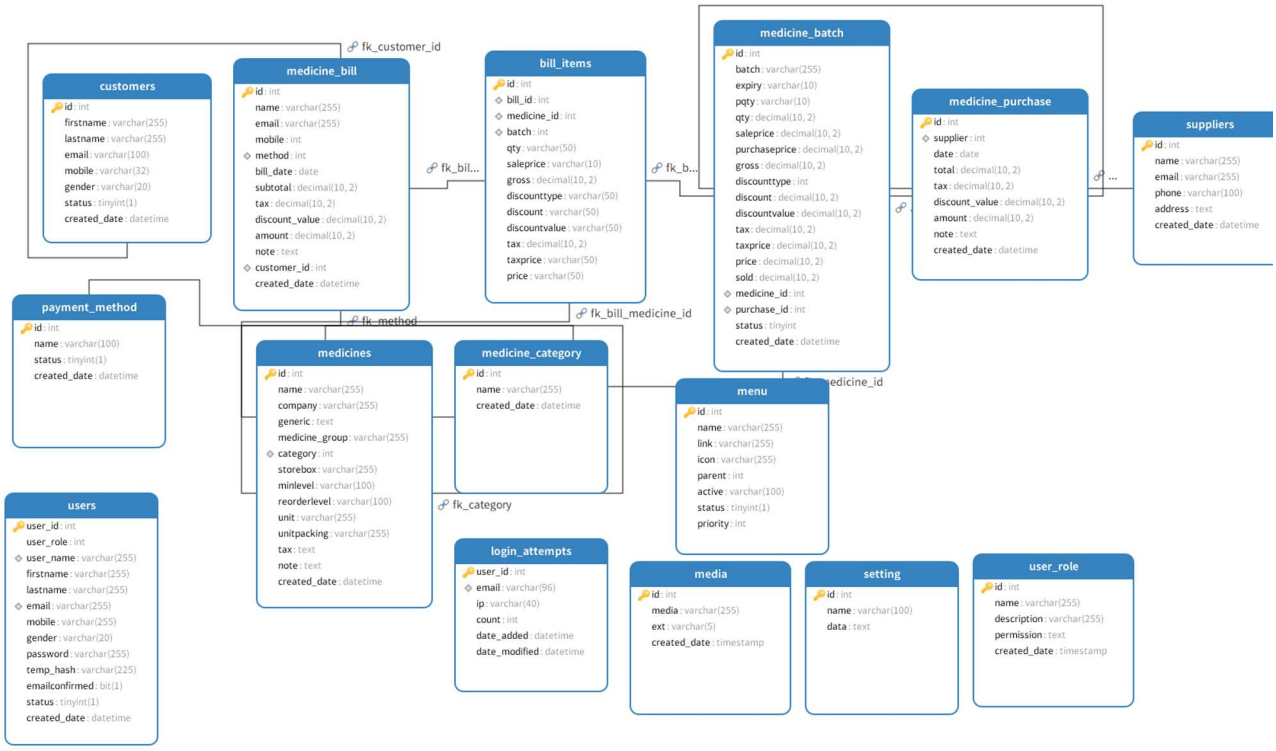
- **Relation:**
  - Standalone entity.
- **Description:**
  Manages user roles, including permissions and descriptions.

### *12.* Users

- **Relation:**
  - Many-to-One (Each user has a role).
- **Description:**
  Stores user information, including credentials, personal details, and role assignment.

## Summary of Relationships

- **Bill Items**:
  - Links to medicine_batch, medicine_bill, and medicines.
- **Medicine Batch**:
  - Linked to medicines and medicine_purchase.
- **Medicine Bill**:
  - Linked to customers and payment_method.
- **Users**:
  - Linked to user_role.




# Conclusion

The Pharmacy Ordering System is a robust solution designed to enhance pharmacy operations with a user-friendly interface. It streamlines processes such as sales processing, inventory management, and customer interaction tracking, reducing manual errors and increasing efficiency. A key feature is real-time inventory management, ensuring accurate stock records, automated reordering, and reduced wastage. The system also integrates pricing and cost management tools, allowing pharmacies to track product margins and optimize pricing strategies. Additionally, the system offers detailed reporting and data analytics, providing insights into sales trends, customer preferences, and inventory performance. This data-driven approach helps improve decision-making and business operations. With features like billing, discounts, and tax calculations, the system simplifies the checkout process, enhancing the customer experience.

In summary, the Pharmacy Ordering System is an essential tool for improving pharmacy efficiency, profitability, and customer satisfaction, supporting growth and scalabilit