

# Pharmacy Management System

## Introduction

The Pharmacy Management System is a software solution designed to streamline and automate various tasks and operations within a pharmacy or a pharmaceutical store. This project report presents the development and implementation of a Pharmacy Management System using Python and PostgreSQL. The system provides features for inventory management, customer management, sales management, and user authentication.

## Objectives

The primary objectives of the Pharmacy Management System are as follows:

**Efficient Inventory Management:** To maintain accurate records of available medicines, their quantities, and expiration dates to ensure efficient inventory management.

**Customer Management:** To manage customer information, including their purchase history, to provide personalized services and maintain customer relationships.

**Sales Management:** To record sales transactions, generate invoices, and track revenue to monitor business performance.

## Technologies Used

Programming Language: Python

Database Management System: PostgreSQL

Graphical User Interface (GUI) Library: Tkinter

External Libraries: psycpg2, ttk (for GUI widgets)

## System Architecture

The system architecture of the Pharmacy Management System consists of three main components:

**Frontend:** The GUI developed using Tkinter library provides an interface for users to interact with the system. It includes screens for sign-in, sign-up, inventory management, customer management, and sales management.

**Backend:** The backend is implemented in Python and handles the business logic of the system. It interacts with the database to perform operations such as adding, updating, and retrieving data.

Database: PostgreSQL is used as the database management system to store data related to medicines, customers, sales transactions, and user accounts.

## **Features**

### **Inventory Management**

Allows pharmacists to add new medicines, update existing medicine details, and view the current inventory.

Provides functionality to track medicine quantities, expiration dates, and supplier information.

### **Customer Management**

Enables pharmacists to add new customers, update customer information, and view customer purchase history.

Helps in maintaining customer records for personalized services and marketing purposes.

### **Sales Management**

Facilitates recording of sales transactions, generating invoices, and calculating total sales revenue.

Allows pharmacists to view sales reports and analyze sales trends over time.

## **Implementation Details**

The implementation of the Pharmacy Management System involves the following steps:

**Database Design:** Designing the database schema to store information about medicines, customers, sales transactions, and user accounts.

**Backend Development:** Implementing the backend logic in Python to handle user authentication, inventory management, customer management, and sales management.

**GUI Development:** Developing the graphical user interface using Tkinter library to create screens for sign-in, sign-up, and other functionalities.

**Integration:** Integrating the backend logic with the GUI to enable user interaction and data processing.

**Testing and Debugging:** Testing the system for usability, functionality, and reliability. Debugging any issues or errors encountered during testing.

## **Future Enhancements**

Barcode Scanning: Integrate barcode scanning functionality to quickly add medicines to the inventory and process sales transactions.

Online Ordering: Implement online ordering capabilities to allow customers to place orders remotely and track order status.

Data Analytics: Incorporate data analytics tools to generate insights from sales data, such as identifying bestselling products and forecasting demand.

## **Conclusion**

The Pharmacy Management System provides a comprehensive solution for managing pharmacy operations efficiently. By automating tasks such as inventory management, customer management, and sales management, the system helps pharmacies improve productivity, reduce errors, and enhance customer service. With further enhancements and updates, the system can continue to meet the evolving needs of pharmacies and pharmaceutical stores.

## **References**

Tkinter Documentation: <https://docs.python.org/3/library/tkinter.html>

PostgreSQL Documentation: <https://www.postgresql.org/docs/>

psycopg2 Documentation: <https://www.psycopg.org/docs/>

Python Documentation: <https://docs.python.org/3/>

ERD:

