



PHARMACY MANAGEMENT SYSTEM



A PROJECT REPORT

Submitted by

PRASANNA N (2303811724321082)

in partial fulfillment of requirements for the award of the course
CGB1221-DATABASE MANAGEMENT SYSTEMS

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE- 2025

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **PHARMACY MANAGEMENT SYSTEM** is the bonafide work of **PRASANNA N (2303811724321082)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr.T. AVUDAIAPPAN, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

SIGNATURE

Mrs.S. GEETHA, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on04.06.2025.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **PHARMACY MANAGEMENT SYSTEM** is the result of original work done by me and best of my knowledge, similar work has not been submitted to “ANNA UNIVERSITY CHENNAI” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1221 – DATABASE MANAGEMENT SYSTEMS**.

Signature

PRASANNA N

Place: Samayapuram

Date: 04.06.2025

ACKNOWLEDGEMENT

It is with great pride that I express my gratitude and in-debt to my institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing me with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of my study in college.

I would like to express my sincere thanks to my beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to my project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. AVUDAIAPPAN, M.E.,Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep expression and sincere gratitude to my project supervisor **Mrs.S.GEETHA, M.E.,** Department of **ARTIFICIAL INTELLIGENCE**, for her incalculable suggestions, creativity, assistance and patience which motivated me to carry out this project.

I render my sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express my special thanks to the officials and Lab Technicians of my departments who rendered their help during the period of the work progress.

INSTITUTE

Vision:

- To serve the society by offering top-notch technical education on par with global standards.

Mission:

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all – round personalities respecting moral and ethical values.

DEPARTMENT

Vision:

- To excel in education, innovation, and research in Artificial Intelligence and Data Science to fulfil industrial demands and societal expectations.

Mission

- To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.
- To collaborate with industry and offer top-notch facilities in a conducive learning environment.
- To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.
- To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

- **PEO1:** Compete on a global scale for a professional career in Artificial Intelligence and Data Science.
- **PEO2:** Provide industry-specific solutions for the society with effective communication and ethics.
- **PEO3** Enhance their professional skills through research and lifelong learning initiatives.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Capable of finding the important factors in large datasets, simplify the data, and improve predictive model accuracy.
- **PSO2:** Capable of analyzing and providing a solution to a given real-world problem by designing an effective program.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The **Pharmacy Management System** is an integrated software solution developed to improve the efficiency and accuracy of daily pharmacy operations. It allows for real-time tracking and management of medicine inventory, ensuring that stock levels are maintained and expired items are flagged in advance. The system facilitates fast and accurate billing, supports prescription handling, and maintains comprehensive records of customers and transactions. It also enables pharmacists and administrators to generate insightful reports on sales performance, stock movement, and profit margins. By automating core processes, the system reduces manual errors, enhances customer service, and ensures compliance with medical and regulatory standards, making it an essential tool for modern pharmacy management. Additionally, the system streamlines supplier management and simplifies order processing. Its user-friendly interface ensures easy adoption by pharmacy staff with minimal training.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD DATABASES FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Pharmacy Management System is an integrated software solution developed to improve the efficiency and accuracy of daily pharmacy operations. It allows for real-time tracking and management of medicine inventory, ensuring that stock levels are maintained and expired items are flagged in advance. The system facilitates fast and accurate billing, supports prescription handling, and maintains comprehensive records of customers and transactions. It also enables pharmacists and administrators to generate insightful reports on sales performance, stock movement, and profit margins. By automating core processes, the system reduces manual errors, enhances customer service, and ensures compliance with medical and regulatory standards, making it an essential tool for modern pharmacy management. Additionally, the system streamlines supplier management and simplifies order processing. Its user-friendly interface ensures easy adoption by pharmacy staff with minimal training.	PO1 3 PO2 3 PO3 3 PO4 2 PO5 3 PO6 2 PO7 1 PO8 2 PO9 2 PO10 2 PO11 2 PO12 2	PSO1 -3 PSO2 -3

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
	1.2 OVERVIEW	1
	1.3 SQL AND DATABASE CONCEPTS	2
2	PROJECT METHODOLOGY	3
	2.1 PROPOSED WORK	3
	2.2 BLOCK DIAGRAM	4
3	MODULE DESCRIPTION	5
	3.1 USER AUTHENTICATION MODULE	5
	3.2 MEDICINE MANAGEMENT MODULE	5
	3.3 ORDER MANAGEMENT MODULE	6
	3.4 SEARCH & VIEW MEDICINES MODULE	6
	3.5 ROLE-BASED UI CONTROL MODULE	6
4	CONCLUSION & FUTURE SCOPE	7
	APPENDIX A SOURCE CODE	9
	APPENDIX B SCREENSHOTS	15
	REFERENCES	17

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The objective of this project is to design and develop a Pharmacy Management System that efficiently manages inventory, sales, and customer records within a pharmacy. The system aims to enable real-time tracking of medicine stock levels, support prescription management, automate invoice generation, and provide comprehensive reports on sales and inventory. By streamlining pharmacy operations including medicine ordering, customer transactions, and regulatory compliance, the system intends to improve operational efficiency, minimize errors, and enhance customer service.

1.2 OVERVIEW

The Pharmacy Management System is designed to support the key functions of a pharmacy by managing medicine inventory, sales transactions, and customer information. It enables accurate monitoring of stock levels and helps in maintaining prescription records. The system also generates invoices and detailed reports related to sales and stock usage. Overall, it aims to simplify pharmacy operations and improve efficiency without compromising on data accuracy or regulatory compliance.

1.3 SQL AND DATABASE CONCEPTS

1. Database Design & Modeling

- Entity Identification (Medicines, Customers, Prescriptions, Sales, Suppliers)
- Attributes and Data Types (e.g., MedicineName, ExpiryDate, Quantity)
- Relationships (One-to-many between Prescriptions and Medicines)
- Entity-Relationship Diagrams (ERD)
- Normalization (1NF, 2NF, 3NF) to avoid data redundancy

2. SQL Basics

- Creating Tables (CREATE TABLE with constraints)
- Inserting Data (INSERT INTO)
- Updating Data (UPDATE)
- Deleting Data (DELETE)
- Querying Data (SELECT, WHERE clause)

3. Advanced SQL Queries

- JOINS (INNER JOIN, LEFT JOIN) to combine tables like Sales and Customers
- Aggregate Functions (SUM, COUNT, AVG) for sales totals and stock levels
- GROUP BY and HAVING clauses for summarizing data (e.g., monthly sales)
- Subqueries to find medicines below reorder levels
- Views to simplify complex queries (e.g., active prescriptions)

CHAPTER 2

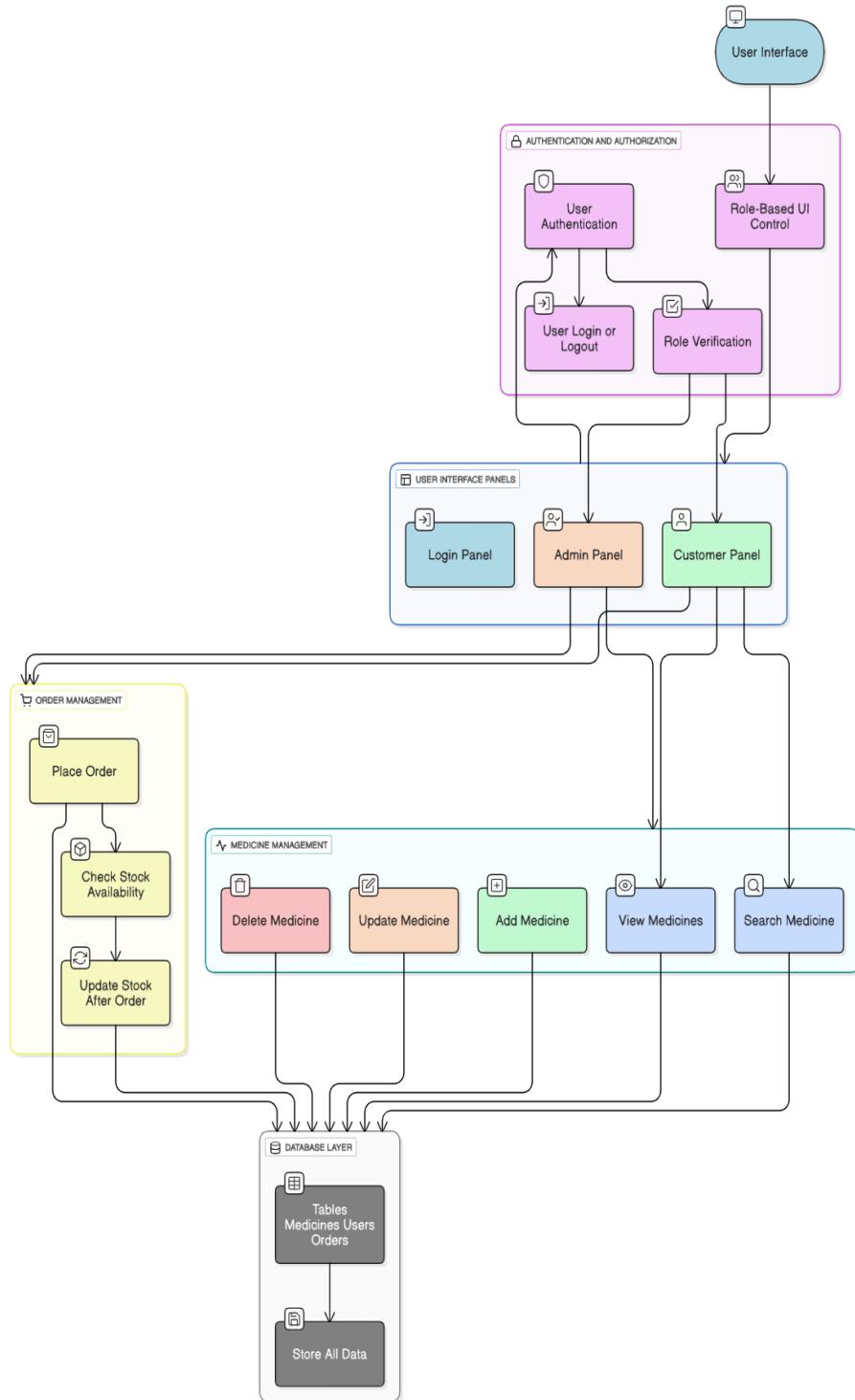
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed Pharmacy Management System aims to develop a comprehensive database-driven solution that will streamline and automate the daily operations of a pharmacy. The system will efficiently manage medicine inventory, sales transactions, customer and prescription records, and generate essential reports for better decision-making.

- **Real-time Inventory Management:** Track stock levels of medicines accurately, monitor expiry dates, and manage reorder alerts to ensure continuous availability of essential drugs.
- **Prescription Management:** Store and manage customer prescriptions, linking prescribed medicines with sales records to maintain accurate and compliant dispensing.
- **Sales and Billing:** Facilitate quick and error-free processing of customer transactions, generate detailed invoices, and maintain transaction histories.
- **Reporting and Analytics:** Provide insightful reports on sales trends, stock usage, and customer purchase history to support pharmacy management in inventory control and business planning.
- **Regulatory Compliance:** Ensure adherence to pharmaceutical regulations by maintaining detailed records and implementing validation checks for controlled substances.

2.2 BLOCK DIAGRAM



CHAPTER 3

MODULE DESCRIPTION

3.1 USER AUTHENTICATION MODULE

Purpose:

Manage user authentication and control access based on user roles within the system.

Key Functionalities:

- User login with credential verification.
- Logout functionality to end user sessions.
- Role-based UI control to present features depending on user roles (Admin, Customer).
- Secure password verification for authentication.

3.2 MEDICINE MANAGEMENT MODULE

Purpose:

Handle all operations related to medicines available in the pharmacy.

Key Functionalities:

- Add new medicines with details (name, company, price, quantity).
- View the complete list of medicines.
- Update medicine information including stock and pricing.
- Delete medicines from the inventory.
- Search medicines by name with pattern matching.

3.3 ORDER MANAGEMENT MODULE

Purpose:

Facilitate placing and tracking orders of medicines by customers.

Key Functionalities:

- Place orders specifying medicine and quantity.
- Check medicine stock availability before order confirmation.
- Update medicine quantity after successful order placement.
- Manage order records linking customers and medicines.

3.4 SEARCH & VIEW MEDICINES MODULE

Purpose:

Allow users to search for and view medicine details efficiently.

Key Functionalities:

- Display all available medicines in a tabular format.
- Search medicines by partial or full name using pattern matching.
- Provide filtered views based on user input.

3.5 ROLE-BASED UI CONTROL MODULE

Purpose:

Control the user interface components displayed to users based on their assigned roles.

Key Functionalities:

- Restrict access to administrative functions for non-admin users.
- Dynamically show/hide UI panels based on user role after login.
- Enable or disable buttons and features depending on permissions.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

CONCLUSION

The Pharmacy Management System stands as a vital solution to modernize and optimize pharmacy operations by integrating inventory control, sales processing, and prescription management into a unified platform. By leveraging a robust database design and efficient SQL-driven processes, the system ensures real-time accuracy, enhances regulatory compliance, and supports informed decision-making through comprehensive reporting. Implementing this system will significantly reduce manual errors, minimize stock-outs and wastage, and accelerate customer service, thereby elevating overall operational efficiency. Ultimately, this project promises to transform pharmacy management into a streamlined, transparent, and reliable process that benefits both the pharmacy staff and customers alike, fostering trust and ensuring the safe and timely availability of medicines.

FUTURE SCOPE

- 1. Efficient and Accurate Inventory Management:** Real-time stock tracking with expiry date monitoring reduces wastage and prevents medicine shortages, ensuring the pharmacy always maintains optimal inventory levels.
- 2. Automated and Error-free Sales Processing:** Automation minimizes manual errors in billing and prescription handling, speeding up customer transactions and enhancing overall accuracy.

3. **Enhanced Customer Experience:** Quick access to customer and prescription history allows personalized service and faster processing, improving customer satisfaction.
4. **Data-driven Business Decisions and Reporting:** Comprehensive sales and inventory reports empower management to make informed decisions for procurement and growth strategies.
5. **Scalable and Extensible System Architecture:** The system is designed to grow with the pharmacy, enabling future integration with online ordering, mobile applications, multi-branch management, and automated supplier reordering.
6. **Improved Regulatory Compliance and Security:** Detailed record-keeping combined with planned advanced security features ensures adherence to pharmaceutical regulations and protects sensitive data.

APPENDIX A – SOURCE CODE

```
!pip install gradio
import gradio as gr
import sqlite3

# Connect to SQLite DB
conn = sqlite3.connect("pharmacy.db", check_same_thread=False)
cursor = conn.cursor()

# Create tables
cursor.execute("""
CREATE TABLE IF NOT EXISTS medicines (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    company TEXT,
    price REAL,
    quantity INTEGER
)
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT UNIQUE NOT NULL,
    password TEXT NOT NULL,
    role TEXT CHECK(role IN ('admin', 'customer')) NOT NULL
)
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS orders (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    customer TEXT,
    medicine_id INTEGER,
    quantity INTEGER,
    FOREIGN KEY(medicine_id) REFERENCES medicines(id)
)
""")
```

```

    ""))
conn.commit()

# Insert test users (only once)
try:
    cursor.execute("INSERT INTO users (username, password, role) VALUES
('admin', 'admin123', 'admin')")
    cursor.execute("INSERT INTO users (username, password, role) VALUES
('john', 'john123', 'customer')")
    conn.commit()
except:
    pass

# Auth check
def login_user(username, password):
    cursor.execute("SELECT role FROM users WHERE username=? AND
password=?", (username, password))
    result = cursor.fetchone()
    if result:
        role = result[0]
        return f"Welcome {username}!", gr.update(visible=False),
gr.update(visible=(role=="admin")), gr.update(visible=(role=="customer")),
gr.update(visible=True), username, role
    else:
        return "Invalid credentials", gr.update(), gr.update(visible=False),
gr.update(visible=False), gr.update(visible=False), "", ""

# Admin Functions
def add_medicine(name, company, price, quantity):
    cursor.execute("INSERT INTO medicines (name, company, price, quantity)
VALUES (?, ?, ?, ?)",
        (name, company or "Unknown", price, quantity))
    conn.commit()
    return "Medicine added successfully!"

def view_medicines():
    cursor.execute("SELECT * FROM medicines")
    return [list(row) for row in cursor.fetchall()]

```

```

def update_medicine(med_id, name, company, price, quantity):
    cursor.execute("SELECT * FROM medicines WHERE id =?", (med_id,))
    if not cursor.fetchone():
        return "No such medicine."
    cursor.execute("UPDATE medicines SET name=?, company=?, price=?, quantity=? WHERE id=?",
                  (name, company, price, quantity, med_id))
    conn.commit()
    return "Updated successfully."

def delete_medicine(med_id):
    cursor.execute("DELETE FROM medicines WHERE id=?", (med_id,))
    conn.commit()
    return "Deleted successfully."

def search_medicine(name):
    cursor.execute("SELECT * FROM medicines WHERE name LIKE ?",
                  (f"% {name} %",))
    return [list(r) for r in cursor.fetchall()]

# Customer Functions
def place_order(customer, medicine_id, quantity):
    cursor.execute("SELECT quantity FROM medicines WHERE id=?", (medicine_id,))
    med = cursor.fetchone()
    if not med:
        return "Medicine not found."
    if med[0] < quantity:
        return "Insufficient stock."
    cursor.execute("INSERT INTO orders (customer, medicine_id, quantity) VALUES (?, ?, ?)",
                  (customer, medicine_id, quantity))
    cursor.execute("UPDATE medicines SET quantity = quantity - ? WHERE id = ?",
                  (quantity, medicine_id))
    conn.commit()
    return "Order placed!"

# UI

```

```

with gr.Blocks() as app:
    gr.Markdown("## 💊 Pharmacy Management System")

    username_state = gr.State("")
    role_state = gr.State("")

    with gr.Group(visible=True) as login_panel:
        u = gr.Textbox(label="Username")
        p = gr.Textbox(label="Password", type="password")
        login_btn = gr.Button("Login")
        login_msg = gr.Textbox(label="Status")

    with gr.Group(visible=False) as admin_panel:
        gr.Markdown("### Admin Panel")
        with gr.Tab("Add Medicine"):
            name = gr.Textbox(label="Name")
            company = gr.Textbox(label="Company")
            price = gr.Number(label="Price")
            qty = gr.Number(label="Quantity")
            add_btn = gr.Button("Add")
            add_out = gr.Textbox(label="Status")
            add_btn.click(add_medicine, [name, company, price, qty], add_out)

        with gr.Tab("View Medicines"):
            view_btn = gr.Button("Refresh")
            view_out = gr.Dataframe()
            view_btn.click(view_medicines, outputs=view_out)

    with gr.Tab("Update Medicine"):
        uid = gr.Number(label="ID")
        uname = gr.Textbox(label="Name")
        ucom = gr.Textbox(label="Company")
        upri = gr.Number(label="Price")
        uqty = gr.Number(label="Quantity")
        update_btn = gr.Button("Update")
        update_out = gr.Textbox(label="Status")
        update_btn.click(update_medicine, [uid, uname, ucom, upri, uqty],
                        update_out)

```

```

with gr.Tab("Delete Medicine"):
    did = gr.Number(label="Medicine ID")
    del_btn = gr.Button("Delete")
    del_out = gr.Textbox(label="Status")
    del_btn.click(delete_medicine, did, del_out)

with gr.Group(visible=False) as customer_panel:
    gr.Markdown("### Customer Panel")

with gr.Tab("Browse"):
    cb = gr.Button("View All")
    cb_out = gr.Dataframe()
    cb.click(view_medicines, outputs=cb_out)

with gr.Tab("Search"):
    sc = gr.Textbox(label="Search")
    sc_btn = gr.Button("Search")
    sc_out = gr.Dataframe()
    sc_btn.click(search_medicine, sc, sc_out)

with gr.Tab("Place Order"):
    mid = gr.Number(label="Medicine ID")
    oqty = gr.Number(label="Quantity")
    order_btn = gr.Button("Place Order")
    order_out = gr.Textbox()
    order_btn.click(place_order, [username_state, mid, oqty], order_out)

logout_btn = gr.Button("Logout", visible=False)

# Login handler
login_btn.click(
    fn=login_user,
    inputs=[u, p],
    outputs=[login_msg, login_panel, admin_panel, customer_panel,
    logout_btn, username_state, role_state]
)

# Logout handler
def logout():

```

```

        return gr.update(visible=True), gr.update(visible=False),
gr.update(visible=False), gr.update(visible=False), "", ""

logout_btn.click(
    fn=logout,
    outputs=[login_panel, admin_panel, customer_panel, logout_btn,
username_state, role_state]
)

app.launch()

import sqlite3
import pandas as pd

# Connect to the SQLite database
conn = sqlite3.connect("pharmacy.db", check_same_thread=False)
cursor = conn.cursor()

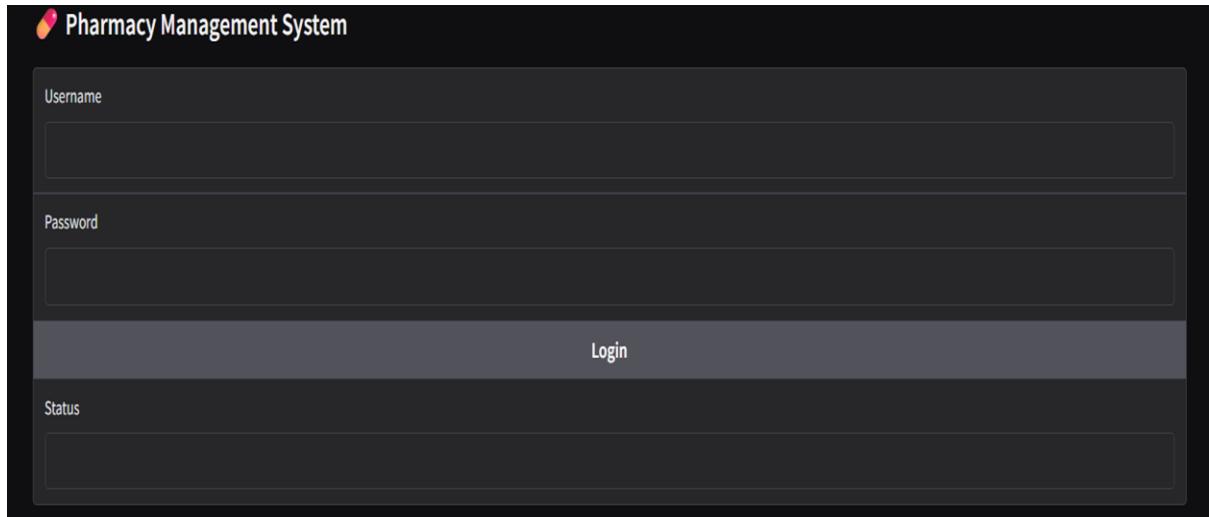
# Show all tables and their data
for table in tables:
    table_name = table[0]
    print(f"\n 📄 Table: {table_name}")

# Read table into a DataFrame
df = pd.read_sql_query(f"SELECT * FROM {table_name}", conn)

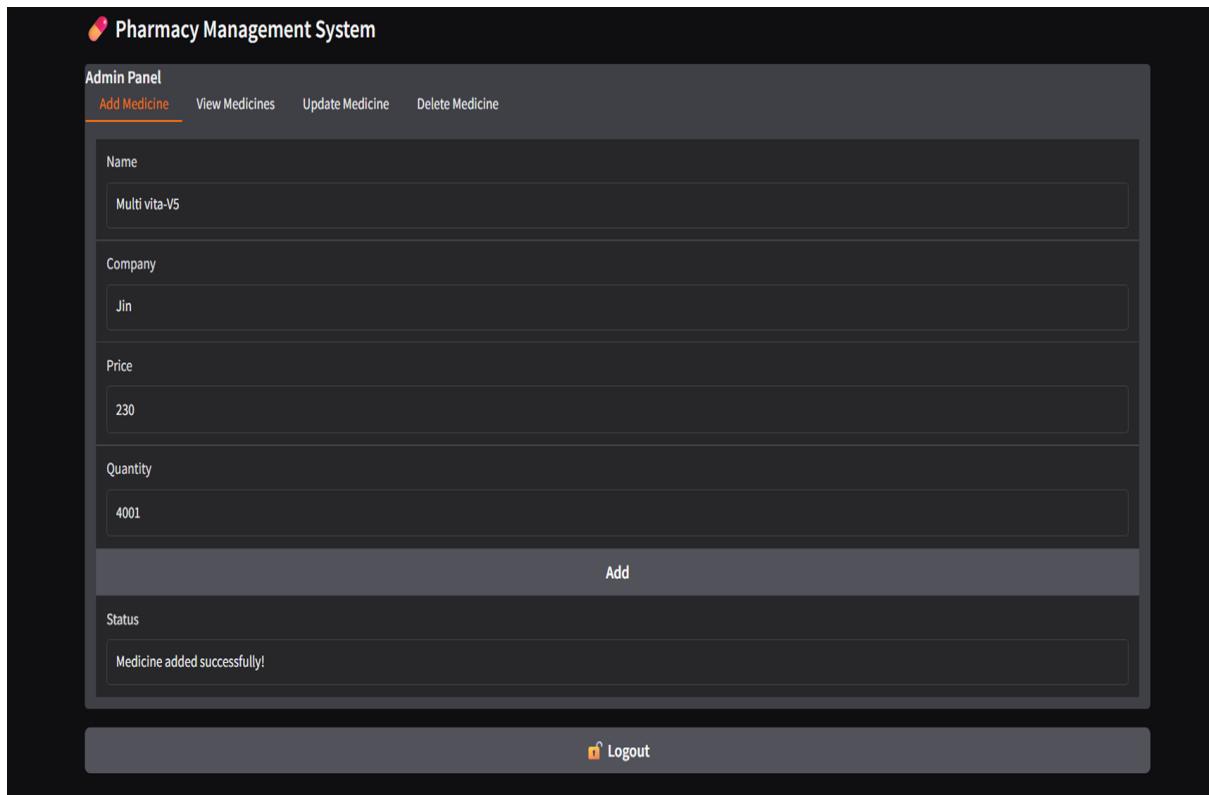
# Display the table
display(df)

```

APPENDIX B - SCREENSHOTS



The screenshot shows the login interface for the Pharmacy Management System. It features a dark-themed header with the system's logo and name. Below the header, there are two input fields: 'Username' and 'Password'. A large, prominent 'Login' button is centered below the password field. At the bottom of the screen, there is a 'Status' section which is currently empty.



The screenshot displays the Admin Panel of the Pharmacy Management System, specifically the 'Add Medicine' section. The top navigation bar includes links for 'Admin Panel', 'Add Medicine' (which is underlined, indicating it is the active page), 'View Medicines', 'Update Medicine', and 'Delete Medicine'. The main form contains four input fields: 'Name' (with 'Multi vita-V5' entered), 'Company' (with 'Jin' entered), 'Price' (with '230' entered), and 'Quantity' (with '4001' entered). Below these fields is an 'Add' button. A status message at the bottom of the form area states 'Medicine added successfully!'. At the very bottom of the screen is a 'Logout' button.

 Pharmacy Management System

Customer Panel

Browse Search Place Order

Medicine ID
1

Quantity
50

Place Order

Textbox
Order placed!

 Logout

Table: medicines					
	id	name	company	price	quantity
0	1	Paracetamol	Dolo	100.0	350
1	2	Vicks action 500	Mae	100.0	400
2	3	Optiozorolan	ikuza	154.0	4001
3	4	Multi vita-V5	Jin	230.0	4001

Table: sqlite_sequence		
	name	seq
0	users	2
1	medicines	4
2	orders	1

Table: users				
	id	username	password	role
0	1	admin	admin123	admin
1	2	john	john123	customer

Table: orders				
	id	customer	medicine_id	quantity
0	1	john	1	50

REFERENCES

1. Bhattacharyya, S., & Maitra, S. (2014). Pharmacy Management System – A Review and Proposed Design. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(7), 123-130.
 - o Classic text on relational databases, SQL, and data integrity principles.
 - o Comprehensive resource on database design, normalization, SQL, and advanced database concepts.
2. Coronel, C., Morris, S., & Rob, P. (2016). *Database Systems: Design, Implementation, & Management* (12th Edition). Cengage Learning.
 - o Covers relational database systems with practical examples, including SQL and transaction management.
3. Date, C. J. (2004). *An Introduction to Database Systems* (8th Edition). Addison-Wesley.
 - o Discusses design challenges and solutions in pharmacy database management.
4. Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th Edition). Pearson.
5. Kaur, G., & Singh, J. (2017). Design and Implementation of Pharmacy Management System Using Database. *International Journal of Computer Applications*, 167(6), 1-5.
 - o Research paper presenting a database-driven pharmacy management system.
6. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th Edition). McGraw-Hill Education.
 - o Well-known textbook for database fundamentals, including transactions, indexing, and security.