Kristu Jayanti College
AUTONOMOUS Bengaluru
Reaccredited A++ Grade by NAAC | Affiliated to Bengaluru North University

Pharmacy Management System

Project Report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS (BCA)

Department of Computer Science (UG)

Submitted by

Tenzin Choenzom
23BCAA63

Under the guidance of

Prof. SONA K.V

DEPARTMENT OF COMPUTER SCIENCE (UG)

BCA PROGRAMME

KRISTU JAYANTI COLLEGE (Autonomous)

K. Narayanapura, Kothanur P.O., Bangalore – 560077

Kristu Jayanti College
AUTONOMOUS Bengaluru
Reaccredited A++ Grade by NAAC | Affiliated to Bengaluru North University

DEPARTMENT OF COMPUTER SCIENCE (UG)

CERTIFICATE OF COMPLETION

This is to certify that the project entitled "**Pharmacy Management System"** has been satisfactorily completed by **Tenzin Choenzom**, 23BCAA63 in partial fulfilment of the award of the Bachelor of Computer Applications degree requirements prescribed by Kristu Jayanti College (Autonomous) Bengaluru (Affiliated to Bangalore North University) during the academic year 2024-25.

Internal Guide ***Head of the Department***

Valued by Examiners

1: _____ ***Centre:*** **Kristu Jayanti College**

2: _____ ***Date:***

DECLARATION

I, Tenzin Choenzom, 23BCAA63 hereby declare that the project work entitled "Pharmacy Management System" is an original project work carried out by me, under the guidance of Prof.Sona K.V

This project work has not been submitted earlier either to any University / Institution or any other body for the fulfillment of the requirement of a course of study.

Signature
Tenzin Choenzom

Bengaluru
Date:

ACKNOWLEDGEMENT

The success of the project depends upon the efforts invested. It's my duty to acknowledge and thank the individuals who has contributed to the successful completion of the project.

I take this opportunity to express my profound and wholehearted thanks to **Rev. Fr. Dr. AUGUSTINE GEORGE, PRINCIPAL, AND Rev. Fr. LIJO P. THOMAS, VICE PRINCIPAL & CHIEF FINANCE OFFICER IN KRISTU JAYANTI COLLEGE**, **BANGALORE** for providing ample facilities made to undergo my project successfully.

I express my deep sense of gratitude and sincere thanks to our **DEAN PROF. CALISTUS JUDE AL** for his valuable support

I express my deep sense of gratitude and sincere thanks to our Head of the Department **Prof. SEVUGA PANDIAN** for his valuable advice.

I feel immense pleasure to thank my respected guide **Prof.Sona K.V** for sustaining Interest and providing dynamic guidance in aiding me to complete this project immaculately and impeccably and for being the source of my strength and confidence.

It is my duty to express my thanks to all Teaching and Non-Teaching Staff members of Computer science department who offered me help directly or indirectly by their suggestions. The successful completion of my project would not have been possible without my parent's Sacrifice, guidance, and prayers. I take this opportunity to thank everyone for their continuous Encouragement. I convey my thankfulness to all my friends who were with me to share my happiness and agony.

Last but not the least I thank almighty God for giving me strength and good health throughout my project and enabling me to complete it successfully.

SYNOPSIS

Pharmacy Management System application to help pharmacist to manage pharmacy in the systematic ways. Pharmacy Management System can make the work easier by giving the details of medicine when its name is entered. It manages tasks like keeping track of medicines, handling prescriptions, billing, and maintaining records of customers and patients. This system reduces mistakes, saves time, and makes sure medicines are always available. A computer gives the details of the medicine like rate of medicine ,and expiry date of medicines manually, so by using this pharmacy management system we can maintain the records of all the medicines. It is fed with the information whenever new medicine are brought and it is provided with expiry date with search option. When we entire the name of medicine it gives the details of medicine. One of the most important responsibilities of pharmacy management is to supervise and manage the pharmacy employees in order to ensure healthy working relationships and outcomes. Each of these function is critical to the pharmacy's operation and should be explained by management.

However, most pharmacies faced problems such as insufficient service promotions, lack of coherence of pharmacy services in hospitals, poor drug information systems and the inconsistency of the pharmacy information management due to its manual processes. Now these are the problems that must be solved with this Pharmacy Management System Project Proposal.

TABLE OF CONTENTS

S L. No	Topic	Page No
1.	**Introduction**	**1-4**
1.1	Problem Definition	1
1.2	Scope of the Project	2-4
2	**System Study**	**5-8**
2.1	Existing System	6
2.2	Feasibility Study	6
2.3	Proposed System	8
3	**System Design**	**9-36**
3.1	ER Diagram	13-16
3.2	DFD [lev0, lev1]	17-20
3.3	Gantt Chart	24
3.4	Input / Output Design	26-36
4	**System Configuration**	**37**
4.1	Hardware Requirements	37
4.2	Software Requirements	37
5.	**Details of Software**	**38-39**
5.1	Overview of Frontend	38
5.2	Overview of Backend	38
5.3	About the Platform	39
6	**Testing**	**40-42**
7	**Conclusion and Future Enhancement**	**43**
8	**Bibliography**	**44**
9	**APPENDICES A-Table Structure**	**45-48**
10	**APPENDICES B-Screenshots**	**49-54**
11	**APPENDICES C-Sample Report of test cases**	**55**
12	**APPENDICES D-Source Code**	**56-80**

INTRODUCTION

1. INTRODUCTION

1.1 PROBLEM DEFINITION

Managing a pharmacy using manual processes often leads to challenges such as inaccurate inventory tracking, delayed billing, difficulty in monitoring expired medicines, and inefficient handling of customer and prescription records. These issues can result in financial losses, poor customer service, and non-compliance with medical regulations. There is a need for a reliable, automated system that can streamline pharmacy operations, reduce human error, and improve overall efficiency. The Pharmacy Management System aims to address these problems by providing a digital platform that ensures accurate stock management, fast billing, proper prescription handling, and organized data storage for suppliers and customers.

1.2 SCOPE OF THE PROJECT

The Pharmacy Management System aims to efficiently manage the ore functions of a pharmacy, including inventory control, customer transaction tracking, billing, and income monitoring. It is intended to stream line daily operations by reducing manual errors and improving data accuracy and accessibility. The system allows for the organized storage and retrieval of medicine details, customer information, and sales records. It supports features such as secure login for authorized access, dynamic record viewing, and structured data management. The project is suitable for use in standalone pharmacies and can be expanded to meet the needs of larger operations in the future.

Module in the Software

1. **Login:**

- The pharmacist will get the access to the system by use of this module. They will need to provide the information about the user ID and the password given to them at the time of registration.

2. **Patient Module :**

- stores customer information, including contact details.
- Patient Profiles: Stores detailed patient information, including contact details and medical history.

3. **Drug Module:**

- manages details about medicines like name, price, and expiry dates.
- Stock Control: Tracks the quantity of each medicine in stock, updates in real-time as medicines are sold or received from suppliers.
- Batch Tracking: Keeps track of medicine batches (especially important for tracking expiration dates and recalls).

4. **Drug Manufacture Module:**

- handles supplier information and contacts.
- This module helps manage all the suppliers and their details for efficient stock ordering and procurement.
- Supplier Details: Store basic info like name and contact, Order Management: Create and track
- purchase orders to suppliers when stock is low.

5. **Bill Module:**

- generates invoices for sales, including amount and date.
- This module manages all invoices related to purchases and sales, keeping financial records organized.

6. Sales Module:

- records each sale, including medicine, quantity, and sale date.
- Sales Transactions: Tracks sales made, including payments, discounts, and taxes applied
- Payment Gateway Integration: Allows integration with payment gateways for processing electronic payments (credit/debit cards, mobile payments).

7. Doctor

- Prescription Reference: Verifies prescriptions to authorized doctors.
- Prescription Validation: Ensures prescriptions are valid and accurate.
- Inventory Management: Helps manage stock based on prescribed medications.
- Reporting: Generates reports on prescriptions by each doctor for analysis

8. Prescription Module:

- Allows pharmacists to enter prescription details (patient information, prescribed drugs, etc.).
- **Prescription Validation:**

  Checks for drug interactions and alerts the pharmacist of any issues before dispensing.

- **Refill Management:**

  Tracks refills for patients and reminds pharmacists when refills are due.

SYSTEM STUDY

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

The existing system of pharmacy typically involves a combination of manual and digital processes to manage drug inventory, prescriptions, billing, and patient records. Most modern pharmacies use pharmacy management software that helps streamline tasks such as stock control, drug interaction alerts, and electronic prescriptions. However, in many smaller or rural pharmacies, manual record-keeping is still common, leading to challenges like errors, inefficiencies, and limited data access. Overall, while advancements have improved accuracy and speed, there remains a need for more integrated and automated systems to enhance efficiency and patient safety.

Disadvantage of existing system

- Human errors in prescription and dispensing
- Poor inventory management
- Lack of real-time data access
- Risk of data loss
- Inefficient patient record handling

2.2 FEASIBILITY STUDY

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. A feasibility study tests the viability of an idea, a project or even a new business. The goal of a feasibility study is to place emphasis on potential problems that could occur if a project is pursued and determines if, after all significant factors are considered, the project should be pursued. Feasibility studies also allow a business to address where and how it will operate, potential obstacles, competition and the Funding needed to get the business up and running.

This project "**Pharmacy Management System**" has undergone the following Feasibility study:

- Every project Economic Feasibility
- Technical Feasibility
- Behavioural Feasibility
- Schedule Feasibility

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Feasibility and risk analysis are related in many ways. The feasibility analysis in this project has been discussed below based on the above-mentioned components of feasibility.

1. Technical feasibility:

Technical feasibility focusses on the technology used. It means the computerized system is technically feasible i.e., it doesn't have any technical fault and work properly in the given environment. The system is technically feasible if it provides the required output.

2. Economic feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of the computerized system. We analyze the computerized system is feasible as than the manual system because it saves the money, time and manpower. It is also feasible according to cost benefit analysis.

3. Behavioural feasibility:

Behavioural feasibility is the analysis of behaviour of the computerized system. In this we analysis that the computerized system is working properly or not. If working then it is communicating properly with the environment or not. All the matters are analyzed and a good computerized system is prepared.

4. Schedule feasibility:

Time evaluation is the most important consideration in the development of project. The time schedule required for the development of the project is very important since more development time affect machine time, cost and cause delay in the development of other systems.

2.3 PROPOSED SYSTEM

Manual Data Entry:

• Proposed Solution: Implement automated data entry using barcode scanners and digital record-keeping to reduce human errors.

Time-Consuming Processes:

• Proposed Solution: Introduce automation for inventory management, billing, and report generation to improve efficiency.

Limited Reporting Capabilities:

• Proposed Solution: Use real-time, automated reports to generate accurate data for inventory, sales, and financial management.

Difficulty in Tracking Stock and Expiry Dates:

• Proposed Solution: Use automated inventory management that provides alerts for low stock and expired products.

Data Security Risks:

• Proposed Solution: Digitally store all patient and medication records with encrypted data and secure access controls to protect sensitive information.

Limited Scalability:

• Proposed Solution: Use a scalable system that can handle increased data and transactions as the pharmacy expands.

Inefficient Customer Service:

• Proposed Solution: Speed up customer service by automating prescription processing, billing, and maintaining digital customer records

SYSTEM DESIGN

3. SYSTEM DESIGN

In the design phase the architecture is established. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture. The architecture defines the components, their interfaces and behaviours. The deliverable design document is the architecture.

The design document describes a plan to implement the requirements. This phase represents the ``how" phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established. The design may include the usage of existing components. Analysing the trade-offs of necessary complexity allows for many things to remain simple which, in turn, will eventually lead to a higher quality product. The architecture team also converts the typical scenarios into a test plan.

In our approach, the team, given a complete requirement document, must also indicate critical priorities for the implementation team. A critical implementation priority leads to a task that has to be done right. If it fails, the product fails. If it succeeds, the product might succeed. At the very least, the confidence level of the team producing a successful product will increase. This will keep the implementation team focused. Exactly how this information is conveyed is a skill based on experience more than a science based on fundamental foundations.

System design is the process of defining the architecture components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

If the broader topic of product development “blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured.

Systems design is therefore the Process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s, systems design had a crucial and respected role in the data processing industry. In the 1990s, standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modelling software systems and is increasingly used for high designing non-software systems and organizations.

LOGICAL DESIGN:

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

PHYSICAL DESIGN:

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified /authenticated, how it is processed, and how it is displayed.

In physical design, the following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing requirements,
5. System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. It is concerned with how the data is represented and stored within the system. Finally, **Process Design** is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system.

At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase. Physical design, in this context, does not refer to the tangible physical design of an information system.

To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and control processor. The H/S personal specification is developed for the proposed system.

E-R DIAGRAM

3.1 E-R DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

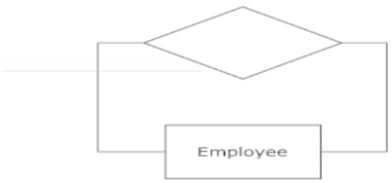
Structure of an Entity Relationship Diagram with Common ERD Notations

An entity relationship diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

- **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.

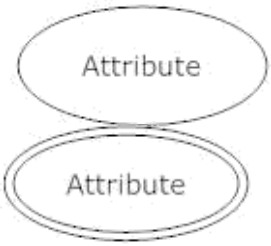
- **Weak entity** is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

- **Actions**, which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.



- **Relationship**: The degree of a relationship is the number of entity types that participate in the relationship.

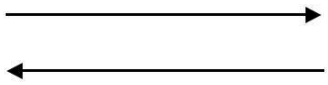
- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



- **Multi-valued attribute** can have more than one value. For example, an employee entity can have multiple skill values.

- **Derived attribute** is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

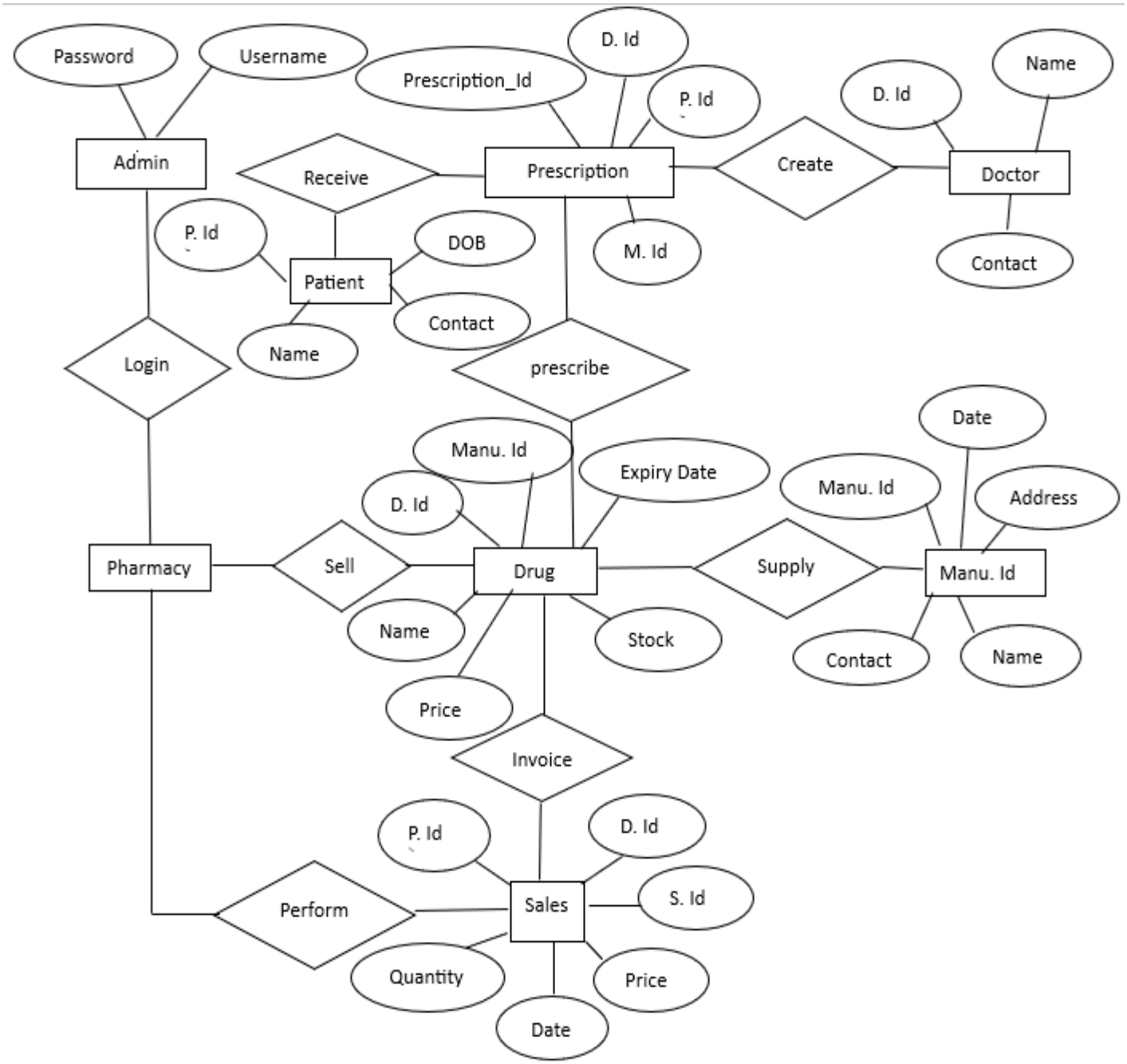
- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.



- **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinarily is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and cordiality specifies the absolute minimum number of relationships.

  - One to One
  - One to Many
  - Many to One
  - Many to Many

E-R DIAGRAM FOR PHARMACY MANAGEMENT SYSTEM



3.2 DATA FLOW DIAGRAM (level 0 and level 1)

The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and are transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs.

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

PHYSICAL VS LOGICAL DFD

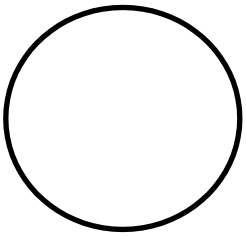
A logical DFD captures the data flows that are necessary for a system to operate. It describes the processes that are undertaken, the data required and produced by each process, and the stores needed to hold the data. On the other hand, a physical DFD shows how the system is actually implemented, either at the moment (Current Physical DFD), or how the designer intends it to be in the future (Required Physical DFD).

Thus, a Physical DFD may be used to describe the set of data items that appear on each piece of paper that move around an office, and the fact that a particular set of pieces of paper are stored together in a filing cabinet. It is quite possible that a Physical DFD will include references to data that are duplicated, or redundant, and that the data stores, if implemented as a set of database tables, would constitute an un-normalized (or de-normalized) relational database. In contrast, a Logical DFD attempts to capture the data flow aspects of a system in a form that has neither redundancy nor duplication.

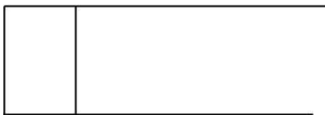
DATA FLOW SYMBOLS AND THEIR MEANINGS: -

An entity: A source of data or a destination for data.

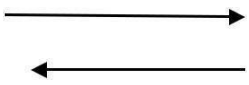
Source/Sink: Represented by rectangles in the diagram. Sources and Sinks are external entities which are sources or destinations of data, respectively.



Process: Represented by circles in the diagram. Processes are responsible for manipulating the data. They take data as input and output an altered version of the data.



Data Store: Represented by a segmented rectangle with an open end on the right. Data Stores are both electronic and physical locations of data. Examples include databases, directories, files, and even filing cabinets and stacks of paper.



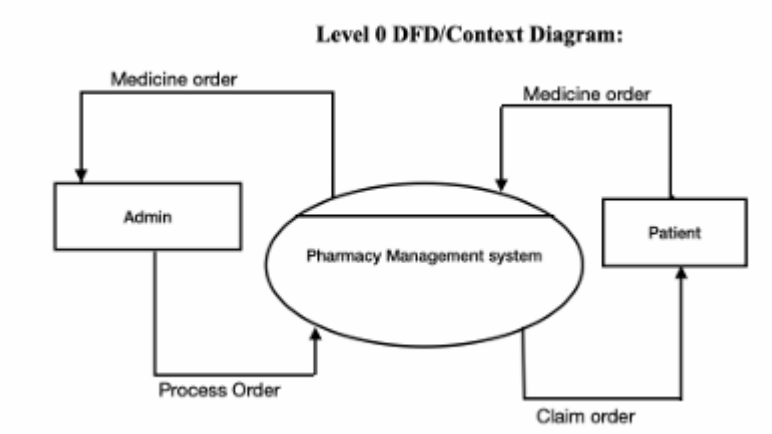
Data Flow: Represented by a unidirectional arrow. Data Flows show how data is moved through the System. Data Flows are labelled with a description of the data that is being passed through it.

A level-0 DFD is the most basic form of DFD. It aims to show how the entire system works at a glance. There is only one process in the system and all the data flows either into or out of this process.

Level-0 DFD's demonstrates the interactions between the process and external entities. They do not contain Data Stores.

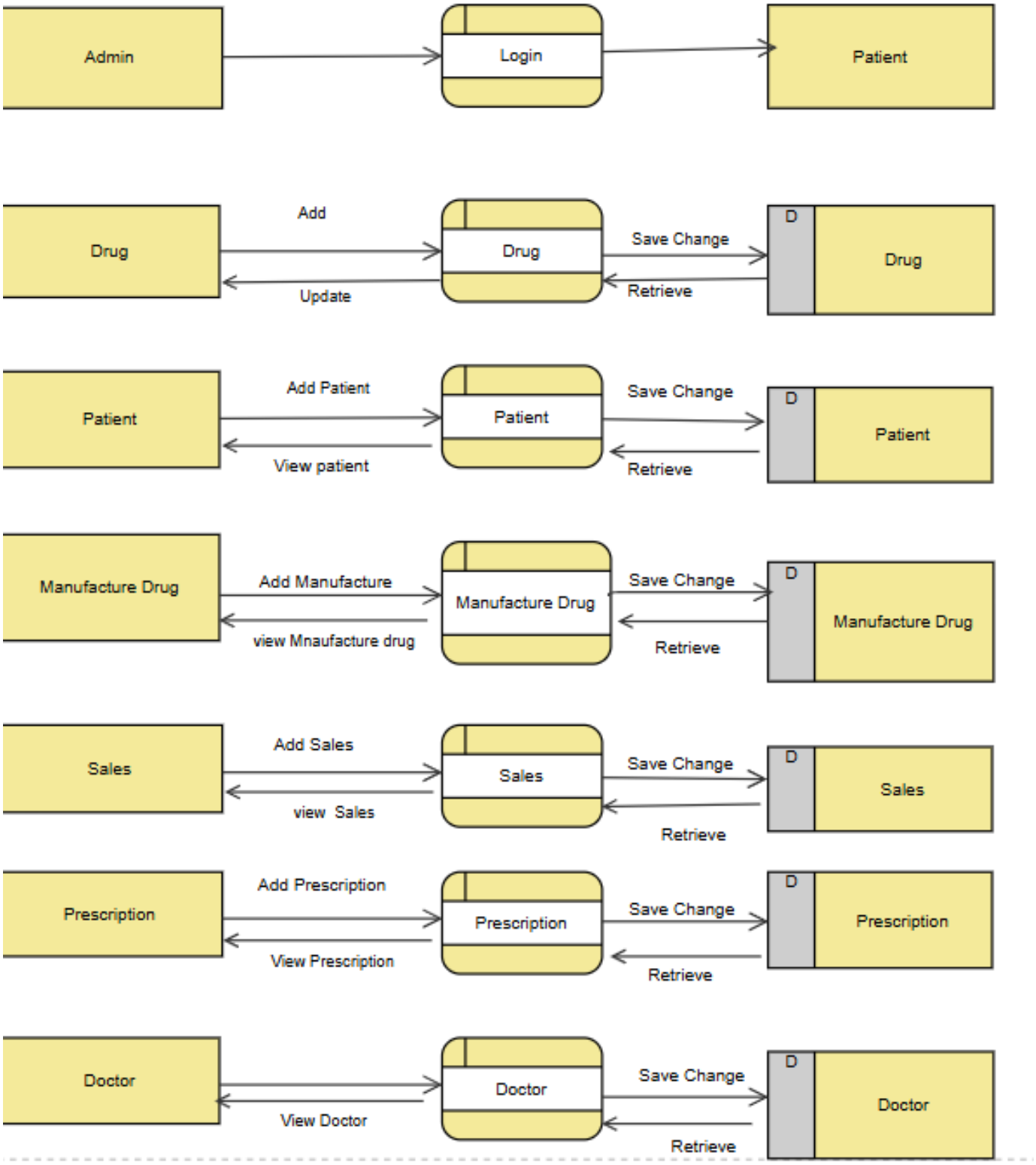
When drawing Level-0 DFD's, we must first identify the process, all the external entities and all the data flows. We must also state any assumptions we make about the system. It is advised that we draw the process in the middle of the page. We then draw our external entities in the corners and finally connect our entities to our process with the data flows.

Level 0 DFD/Context Diagram:



Level 1 DFD:

Level 1 DFD's aim is to give an overview of the full system. They look at the system in more detail. Major processes are broken down into sub-processes. Level 1 DFD's also identifies data stores that are used by the major processes. When constructing a Level 1 DFD we must start by examining the Context Level DFD. We must break up the single process into its subprocesses. We must then pick out the data stores from the text we are given and include them in our DFD. Like the Context Level DFD's, all entities, data stores and processes must be labelled. We must also state any assumptions made from the text.


Admin
Login
Patient
Drug
Add
Update
Drug
Save Change
Retrieve
D
Drug
Patient
Add Patient
View patient
Patient
Save Change
Retrieve
D
Patient
Manufacture Drug
Add Manufacture
view Mnaufacture drug
Manufacture Drug
Save Change
Retrieve
D
Manufacture Drug
Sales
Add Sales
view Sales
Sales
Save Change
Retrieve
D
Sales
Prescription
Add Prescription
View Prescription
Prescription
Save Change
Retrieve
D
Prescription
Doctor
View Doctor
Doctor
Save Change
Retrieve
D
Doctor

3.3 ACTIVITY DIAGRAM

An activity diagram visually represents the series of actions or flow of control in a system similar to a flow chart or data flow diagram. Activity diagram are often used in business processing modelling. They can also describe steps in a used case diagram. The activity diagram for Admin module and User module of College Gadget Booking is given below.

3.4 GANTT CHART

A Gantt chart is a type of bar chart, devised by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities.

HISTORICAL DEVELOPMENT:

The first known tool of this type was developed in 1896 by Karol Adamiecki, who called it a Harmon gram. Adamiecki did not publish his chart until 1931, however, and only in Polish, which limited both its adoption and recognition of his authorship. The chart is named after Henry Gantt (1861–1919), who designed his chart around the years 1910–1915. One of the first major applications of Gantt charts was by the United States during World War I, at the instigation of General William Crozier in the 1980s, personal computers allowed widespread creation of complex and elaborate Gantt charts. The first desktop applications were intended mainly for project managers and project schedulers. With the advent of the Internet and increased collaboration over networks at the end of the 1990s, Gantt charts became a common feature of web-based applications, including collaborative groupware.

GANTT CHART BENEFITS:

Clarity:

One of the biggest benefits of a Gantt chart is the tool's ability to boil down multiple tasks and timelines into a single document. Stakeholders throughout an organization can easily understand where teams are in a process while grasping the ways in which independent elements come together toward project completion.

Communication:

Teams can use Gantt charts to replace meetings and enhance other status updates. Simply clarifying chart positions offers an easy, visual method to help team members task progress.

Motivation:

Some teams or team members become more effective when faced with a form of external motivation. Gantt charts offer teams the ability to focus work at the front of a task timeline, or at the tail end of a chart segment. Both types of team members can find Gantt charts meaningful as they plug their own work habits into the overall project schedule

Coordination:

For project managers and resource schedulers, the benefits of a Gantt chart include the ability to sequence events and reduce the potential for overburdening team members. Some project managers even use combinations of charts to break down projects into more manageable sets of tasks.

Creativity:

Sometimes, a lack of time or resources forces project managers and teams to find creative solutions. Seeing how individual tasks intertwine on Gantt charts often encourages new partnerships and collaborations that might not have evolved under traditional task assignment systems.

Time Management:

Most managers regard scheduling as one of the major benefits of Gantt charts in a creative environment. Helping teams understand the overall impact of project delays can foster stronger collaboration while encouraging better task organization

Flexibility:

Whether you use Excel to generate Gantt charts or you load tasks into a more precise chart generator the ability to issue new charts as your project evolves lets you react to unexpected changes in project scope or timeline. While revising your project schedule too frequently can eliminate some of the other benefits of Gantt charts, offering a realistic view of a project can help team members recover from setbacks or adjust to other changes.

Manageability:

For project managers handling complex assignments, like software publishing or event planning, the benefits of Gantt charts include externalizing assignments. By visualizing all of the pieces of a project puzzle, managers can make more focused, effective decisions about resources and timetables.

Efficiency:

Another one of the benefits of Gantt charts is the ability for teams members to leverage each other's deadlines for maximum efficiency. For instance, while one team member waits on the outcome of three other tasks before starting a crucial piece of the assignment, he or she can perform other project tasks. Visualizing resource usage during projects allows managers to make better use of people, places, and things.

Accountability:

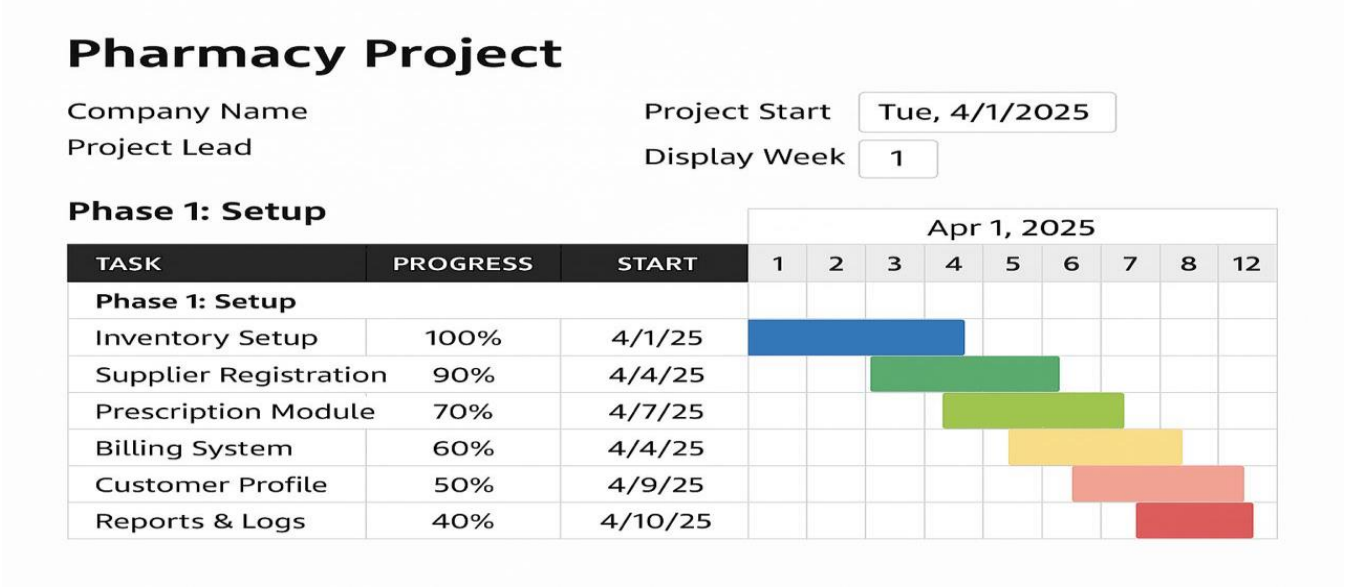
When project teams face major organizational change, documenting effort and outcomes becomes crucial to career success. Using Gantt charts during critical projects allows both project managers and participants to track team progress, highlighting both big wins and major failures during professional review periods; team members who frequently exceed expectations can leverage this documentation into larger raises or bonuses.

Gantt chart Importance:

The project's summary and terminal elements, which combine to form the project's internal structure, are shown on the Gantt chart. Many charts will also depict the precedence rankings and dependencies of various tasks within the project. The charts can illustrate the start and finish project terminal elements in project management. It can also show summary elements and terminal

dependencies. The smallest task tracked as part of the project effort is known as a terminal element. Gantt chart represents the tasks in most modern project scheduling packages. However other management applications use simpler communication tools such as message boards, to-do lists and simple scheduling etc., therefore, they do not use Gantt charts as heavily.

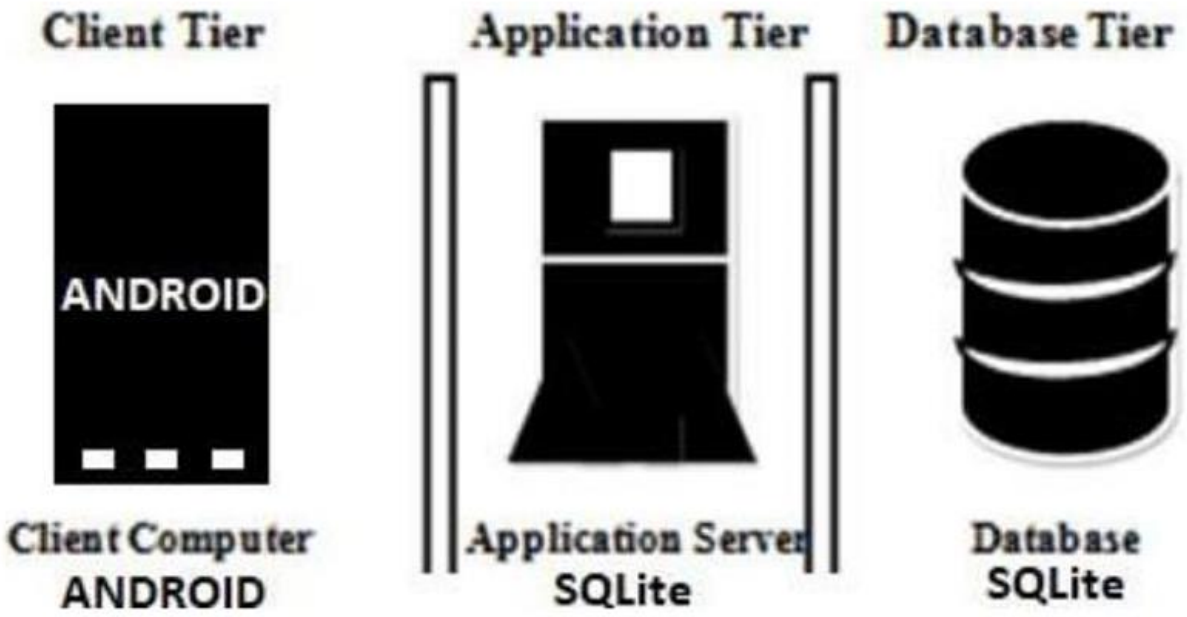
The way to create this chart begins by determining and listing the necessary activities.Next, sketch out how you expect the chart to look. List which items depend on others and what activities take place when. For each activity, list how many man-hours it will require, and who is responsible. Lastly, determine the throughput time.



This technique's primary advantage is its good graphical overview that is easy to understand for nearly all project participants and stakeholders. Its primary disadvantage is its limited applicability for many projects, since projects are often more complex than can be effectively communicated with this chart.

3.5 ARCHITECTURAL DESIGN

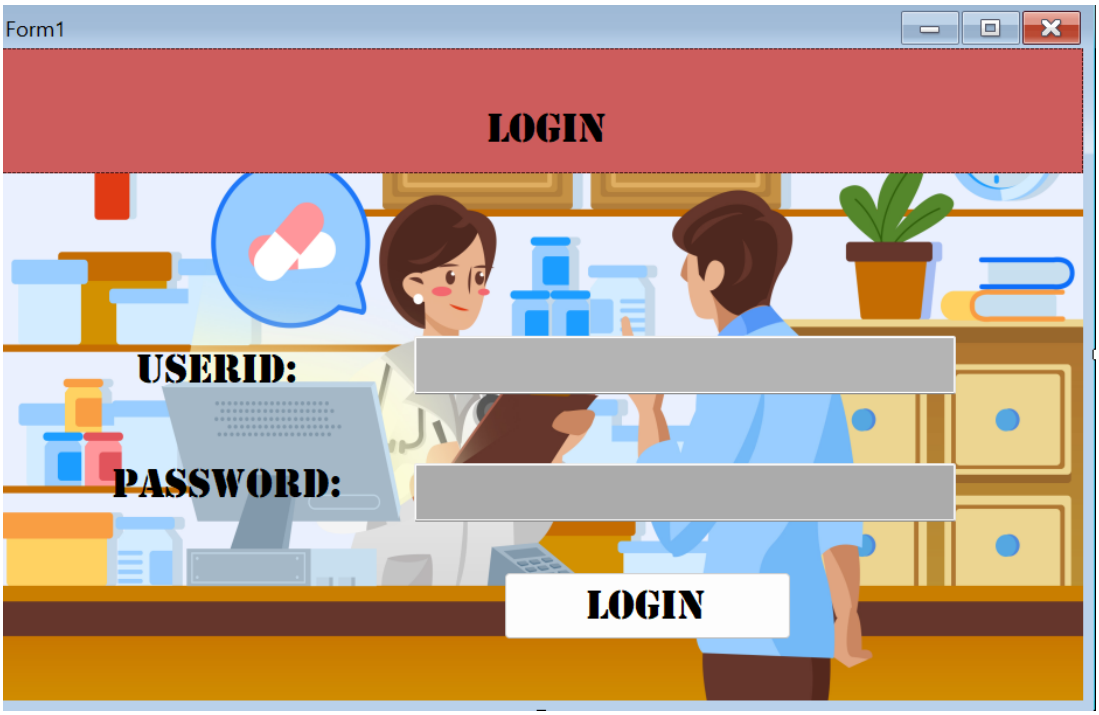
An architectural model (in software) is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system or ecosystem.

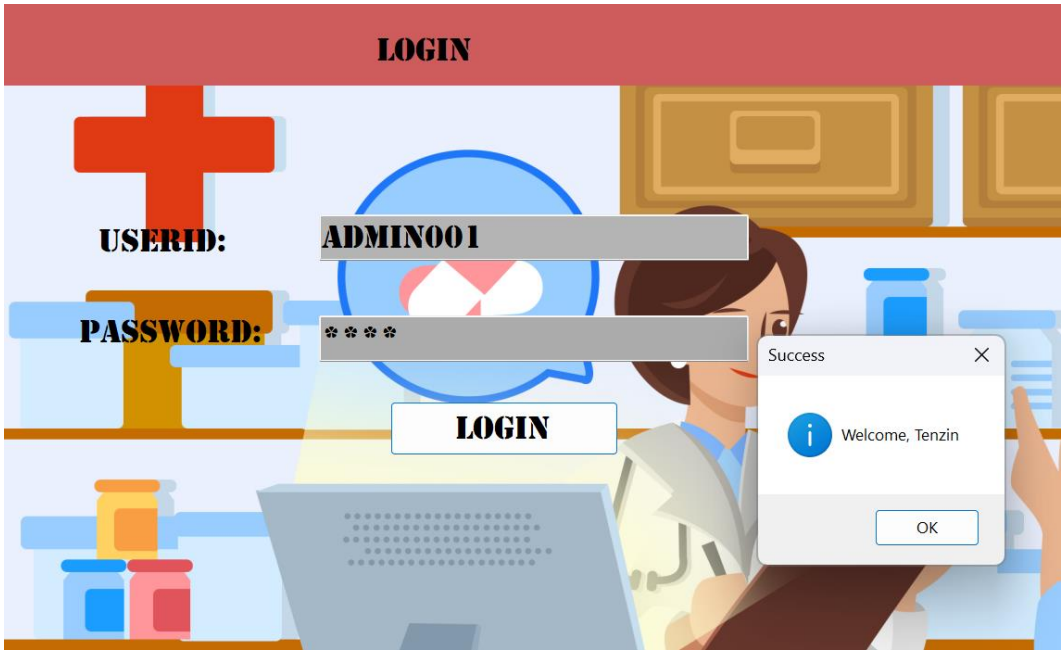


INPUT/OUTPUT DESIGN

3.6 INPUT/OUTPUT DESIGN

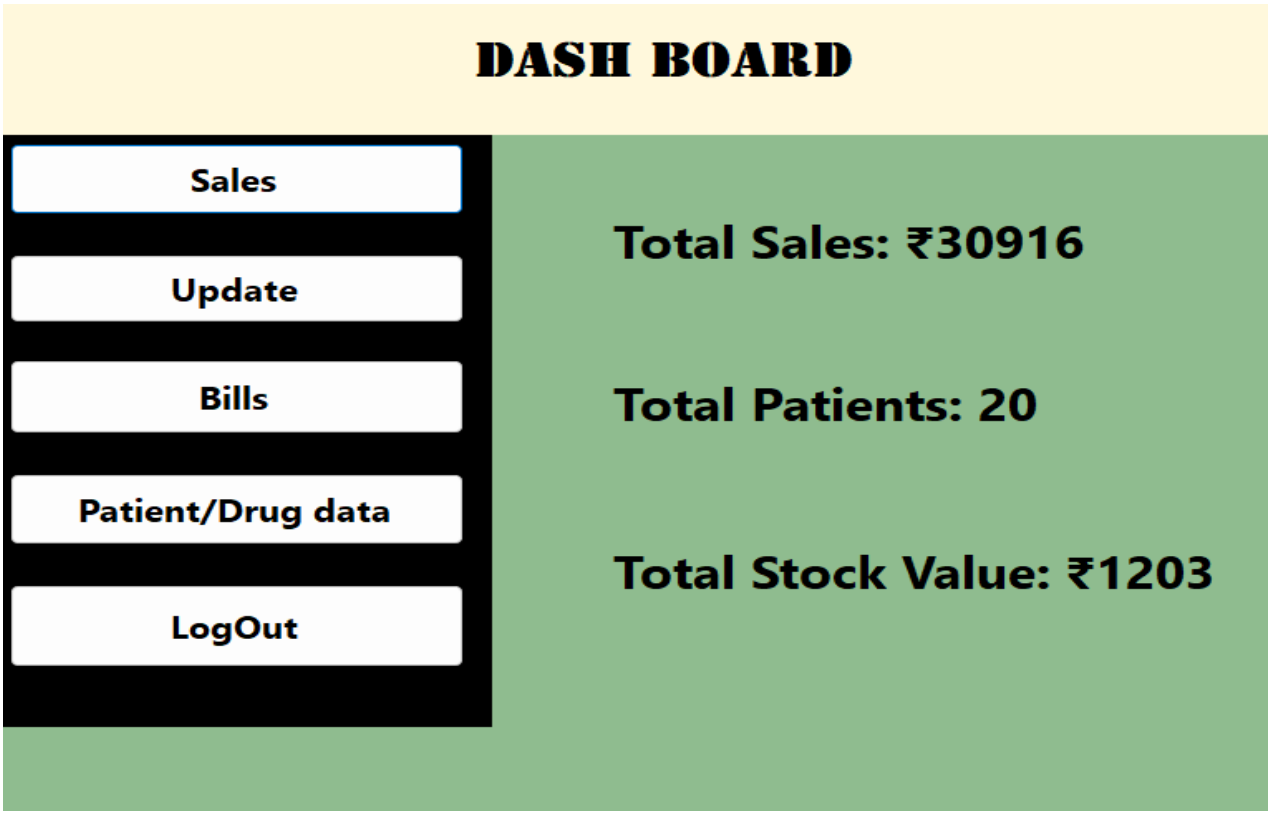
Login:



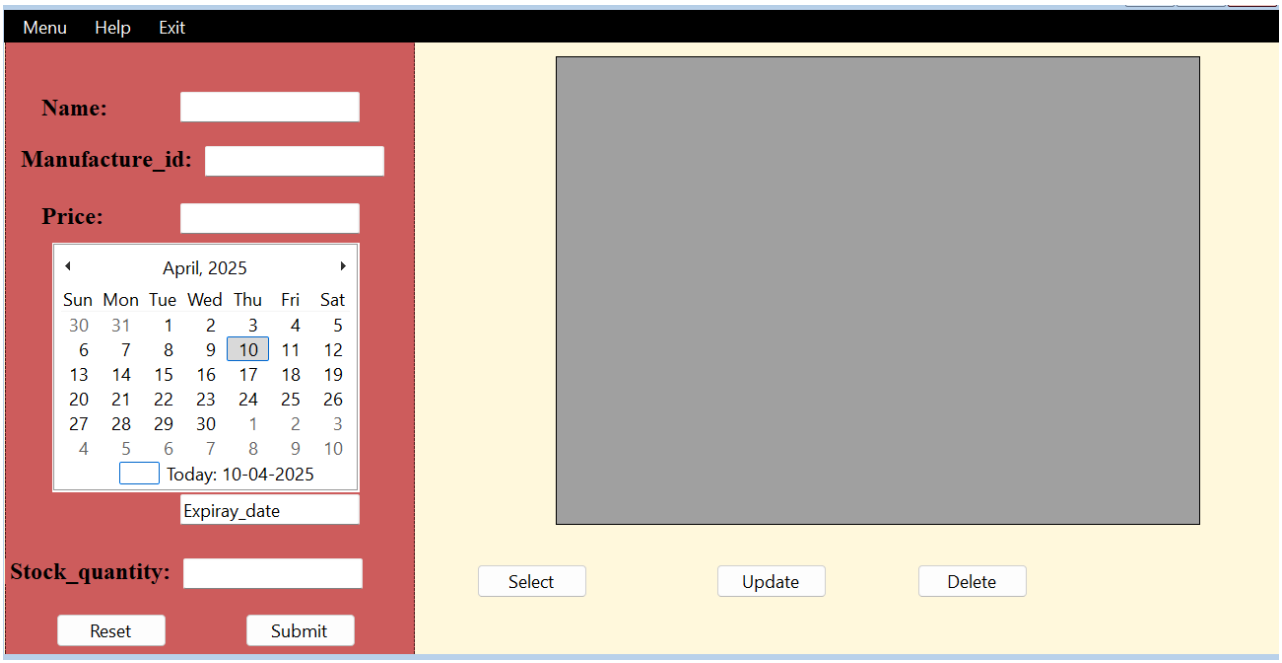


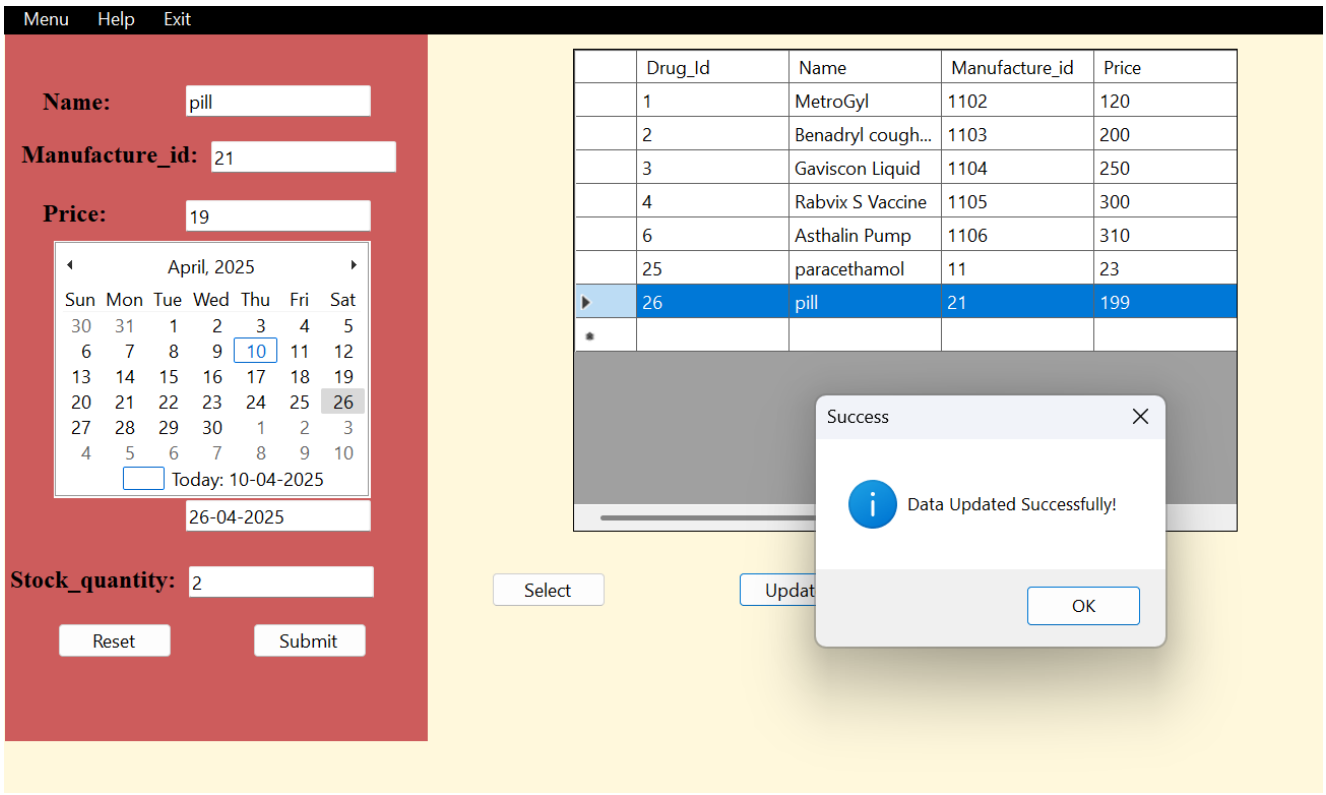
UserId	UserName	Password	Click to Add
admin	karma	1234	
Admin001	Tenzin	1234	
Admin002	Sam Saheb	1234	
Admin003	abhi	1234	
*			

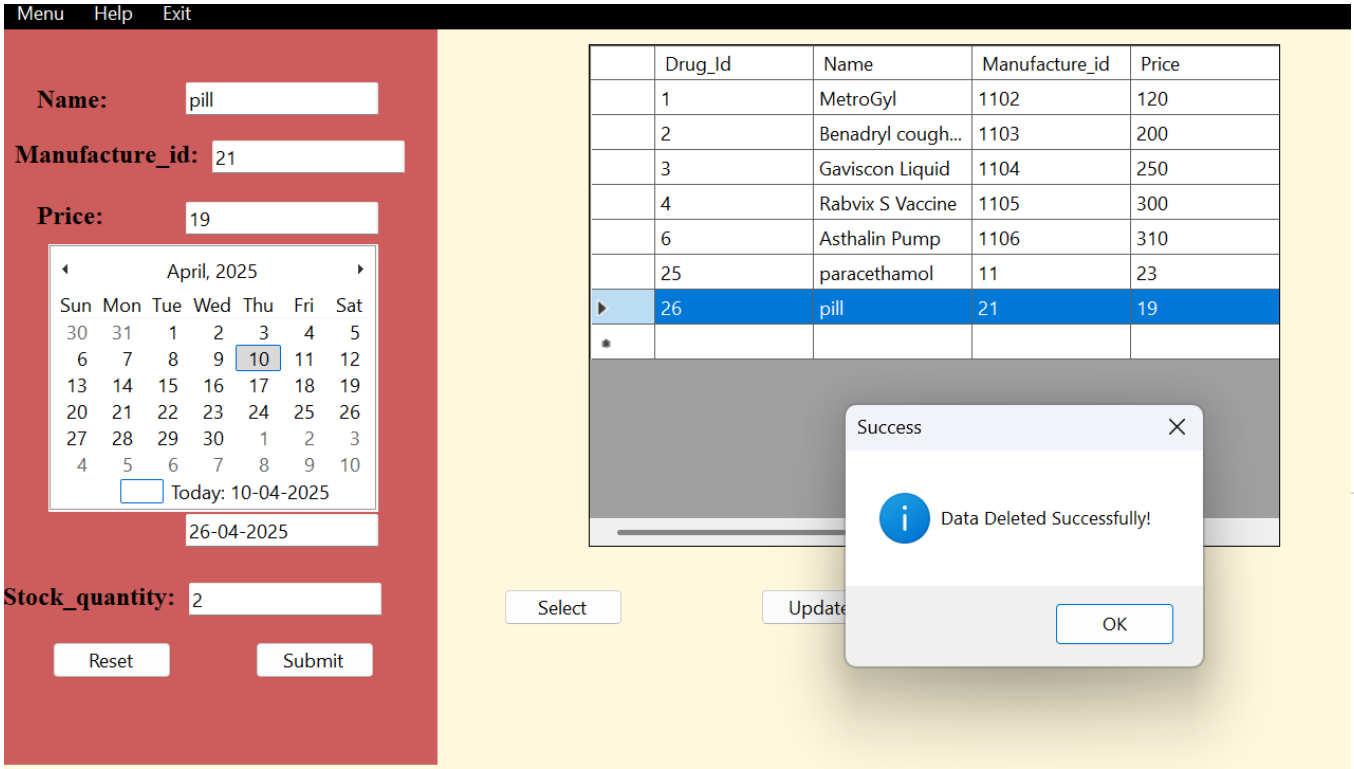
Dashboard:



Add/delete and update the data Drug

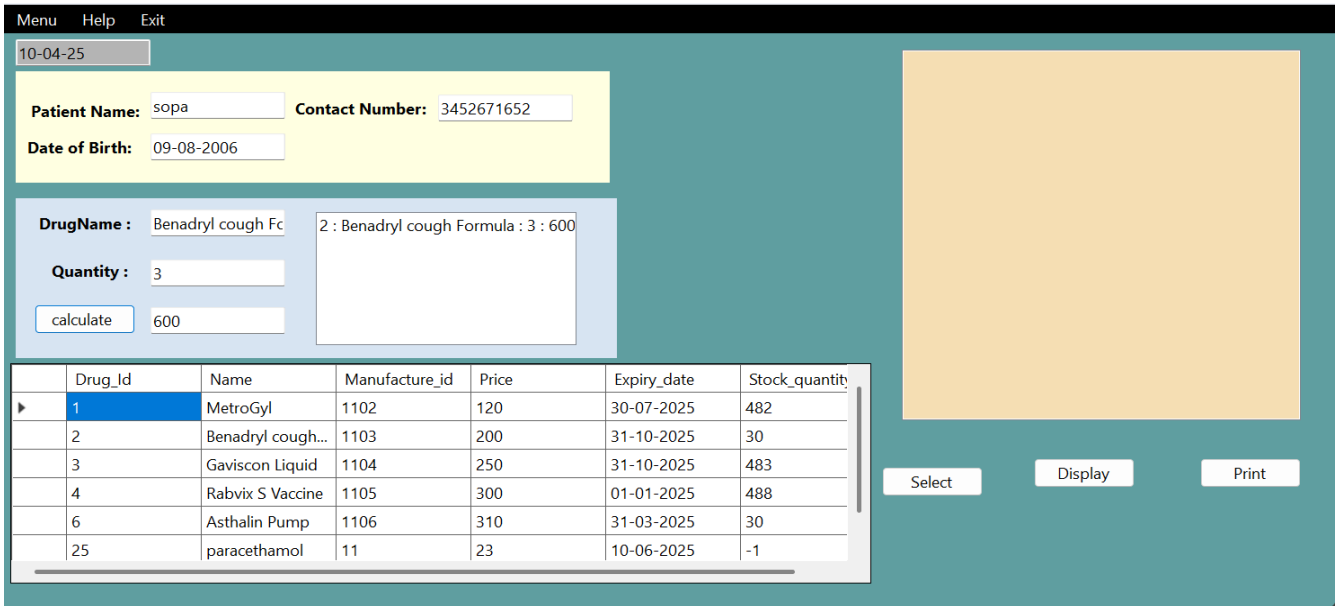


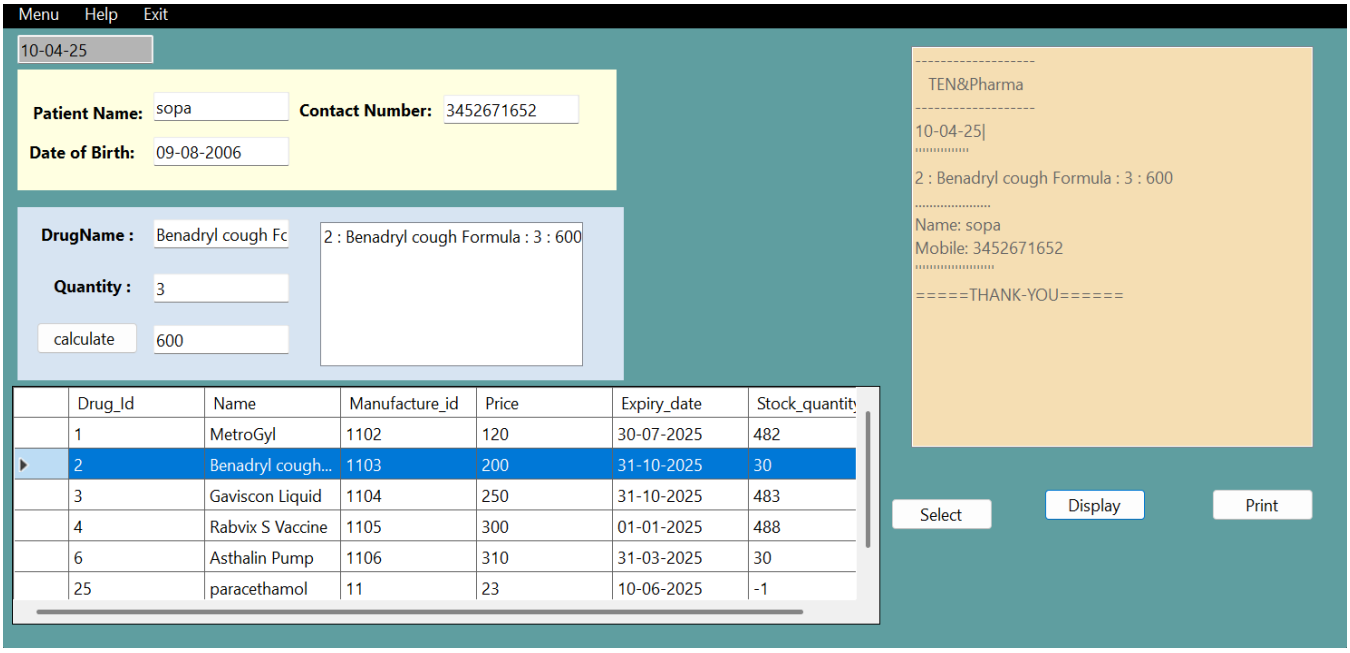


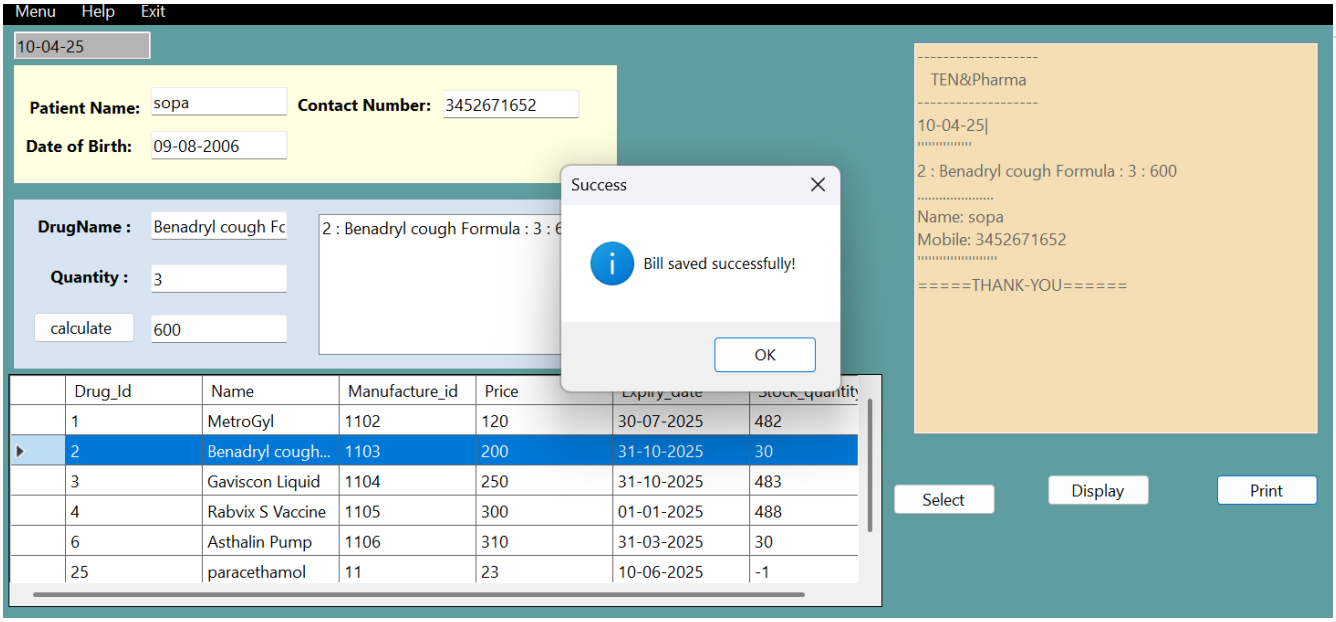


Drug_Id	Name	Manufacture	Price	Expiry_date	Stock_quant	Click to Add
1	MetroGyl	1102	120	30-07-2025	492	
2	Benadryl cough	1103	200	31-10-2025	86	
3	Gaviscon Liquic	1104	250	31-10-2025	498	
4	Rabvix S Vaccin	1105	300	01-01-2025	493	
6	Asthalin Pump	1106	307	31-03-2025	30	
(New)						

Sales:

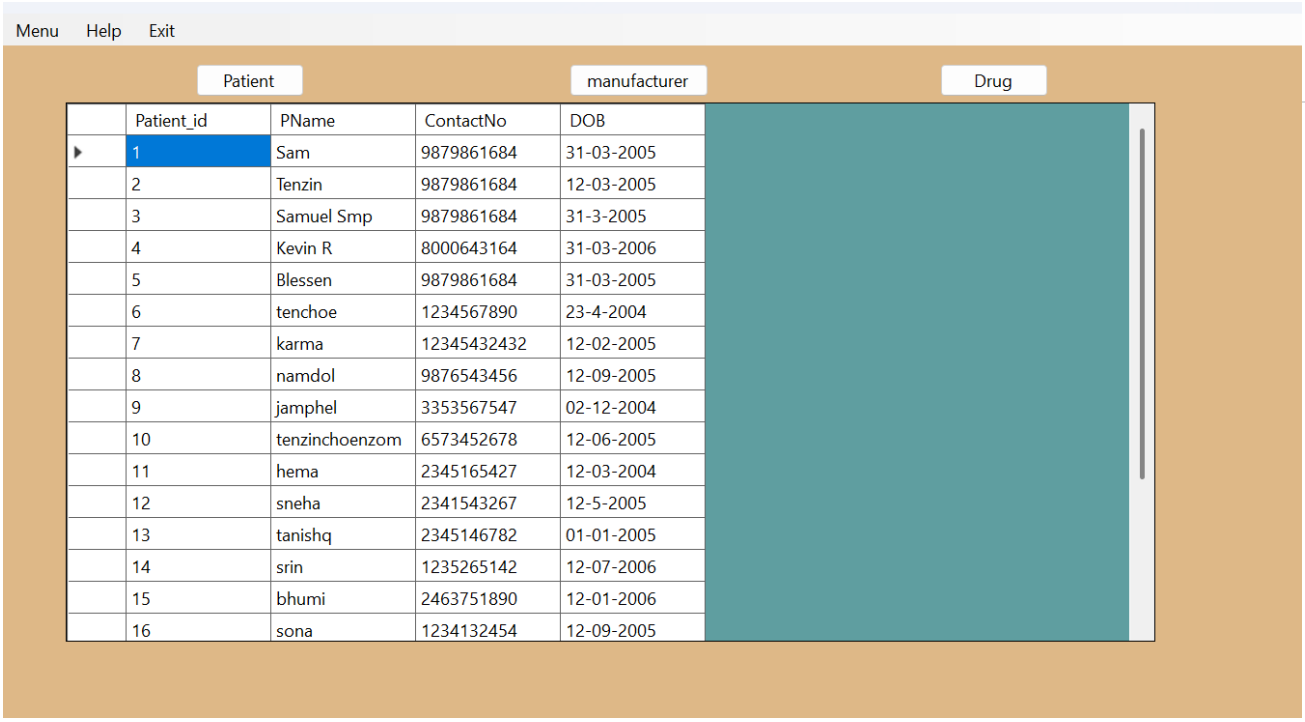

Menu Help Exit
10-04-25
Patient Name: sopa
Contact Number: 3452671652
Date of Birth: 09-08-2006
DrugName : Benadryl cough Fc
2 : Benadryl cough Formula : 3 : 600
Quantity : 3
calculate 600
Drug_Id Name Manufacture_id Price Expiry_date Stock_quantity
1 MetroGyl 1102 120 30-07-2025 482
2 Benadryl cough... 1103 200 31-10-2025 30
3 Gaviscon Liquid 1104 250 31-10-2025 483
4 Rabvix S Vaccine 1105 300 01-01-2025 488
6 Asthalin Pump 1106 310 31-03-2025 30
25 paracethamol 11 23 10-06-2025 -1
Select
Display
Print


Menu Help Exit
10-04-25
Patient Name: sopa
Contact Number: 3452671652
Date of Birth: 09-08-2006
DrugName : Benadryl cough Fc
2 : Benadryl cough Formula : 3 : 600
Quantity : 3
calculate 600
TEN&Pharma
10-04-25
2 : Benadryl cough Formula : 3 : 600
Name: sopa
Mobile: 3452671652
=====THANK-YOU======
Drug_Id Name Manufacture_id Price Expiry_date Stock_quantity
1 MetroGyl 1102 120 30-07-2025 482
2 Benadryl cough... 1103 200 31-10-2025 30
3 Gaviscon Liquid 1104 250 31-10-2025 483
4 Rabvix S Vaccine 1105 300 01-01-2025 488
6 Asthalin Pump 1106 310 31-03-2025 30
25 paracethamol 11 23 10-06-2025 -1
Select
Display
Print



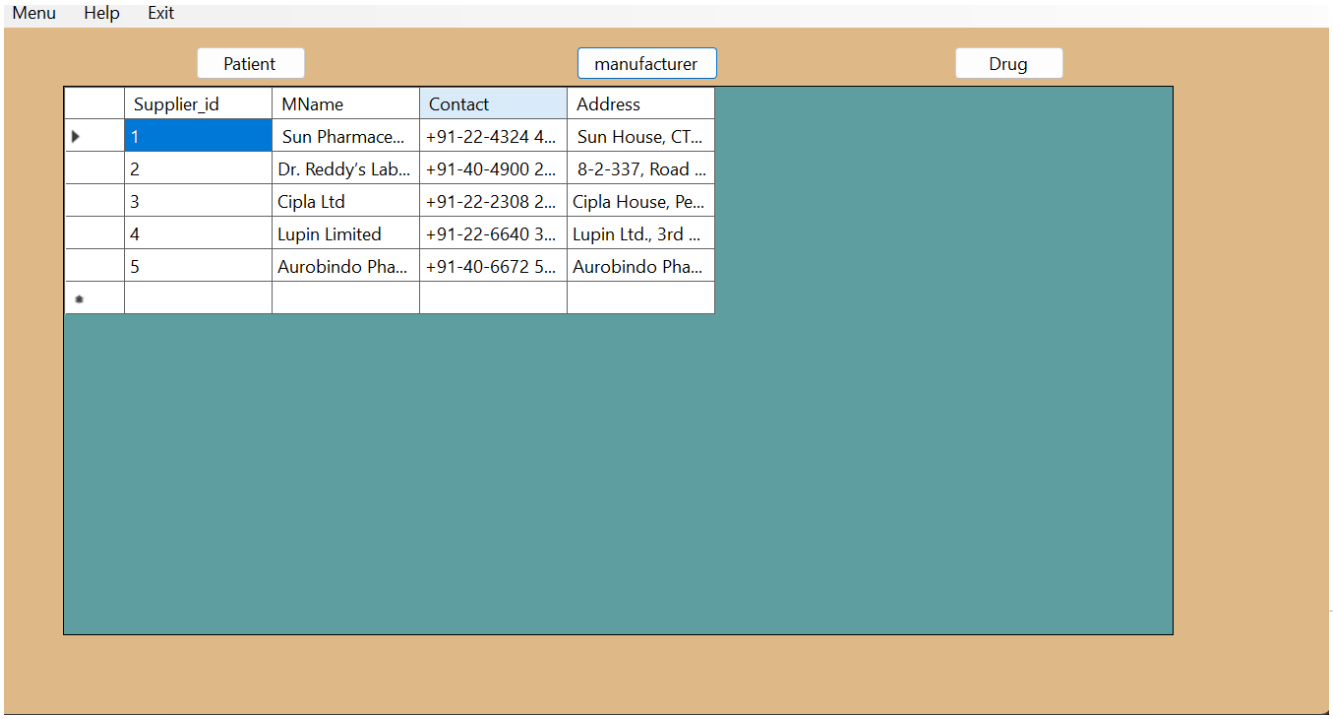
Sales_id	SalesDate	PatientName	DrugName	TotalAmoun	Quantity
1	25-03-25	Sam	1 : MetroGyl : 2	890	
2	26-03-25	Tenzin	1 : MetroGyl : 2	840	
3	26-03-25	Samuel Smp	4 : Rabvix S Vac	970	
4	26-03-25	Kevin R	7 : Asthaline Ta	440	
5	26-03-25	Blessen	2 : Benadryl cou	2320	
6	26-03-25	tenchoe	1 : MetroGyl : 2	640	
7	26-03-25	karma	4 : Rabvix S Vac	3300	
(New)					

Patient, manufacturer and Drug detail:

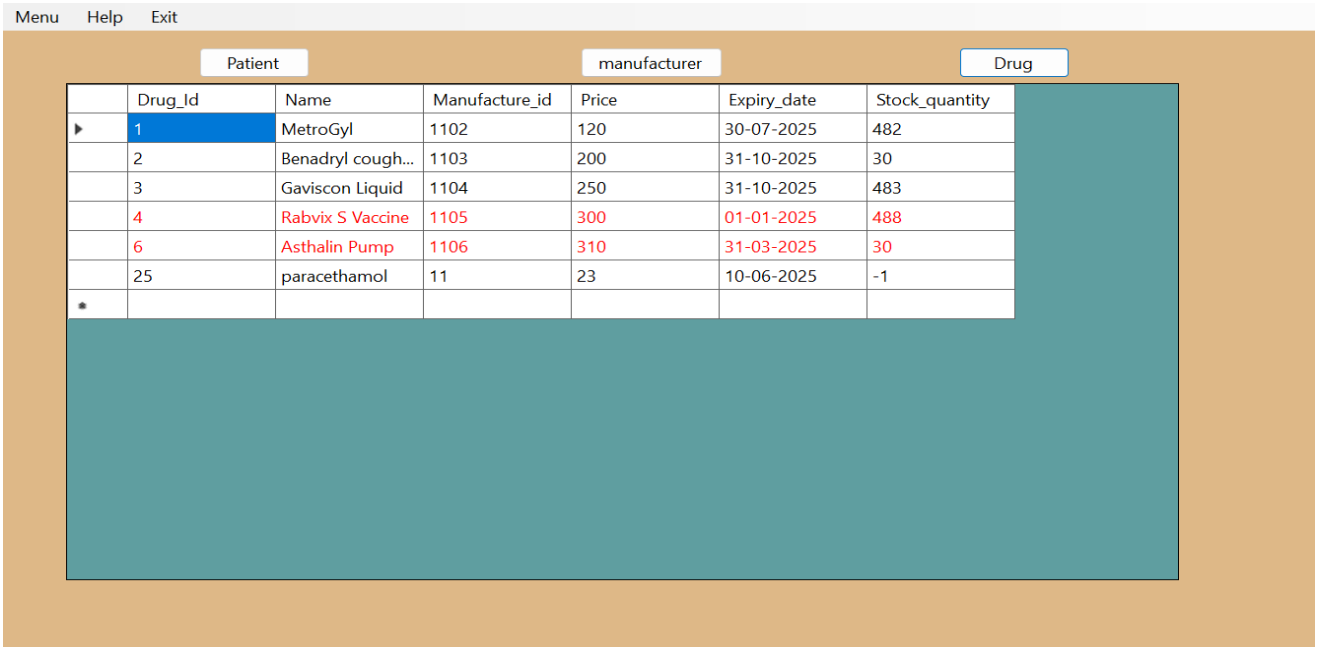


Patient_id	PName	ContactNo	DOB
1	Sam	9879861684	31-03-2005
2	Tenzin	9879861684	12-03-2005
3	Samuel Smp	9879861684	31-3-2005
4	Kevin R	8000643164	31-03-2006
5	Blessen	9879861684	31-03-2005
6	tenchoe	1234567890	23-4-2004
7	karma	12345432432	12-02-2005
8	namdol	9876543456	12-09-2005
9	jamphel	3353567547	02-12-2004
10	tenzinchoenzom	6573452678	12-06-2005
11	hema	2345165427	12-03-2004
12	sneha	2341543267	12-5-2005
13	tanishq	2345146782	01-01-2005
14	srin	1235265142	12-07-2006
15	bhumi	2463751890	12-01-2006
16	sona	1234132454	12-09-2005

Patient_id	PName	ContactNo	DOB
1	Sam	9879861684	31-03-2005
2	Tenzin	9879861684	12-03-2005
3	Samuel Smp	9879861684	31-3-2005
4	Kevin R	8000643164	31-03-2006
5	Blessen	9879861684	31-03-2005
6	tenchoe	1234567890	23-4-2004
7	karma	12345432432	12-02-2005
(New)			



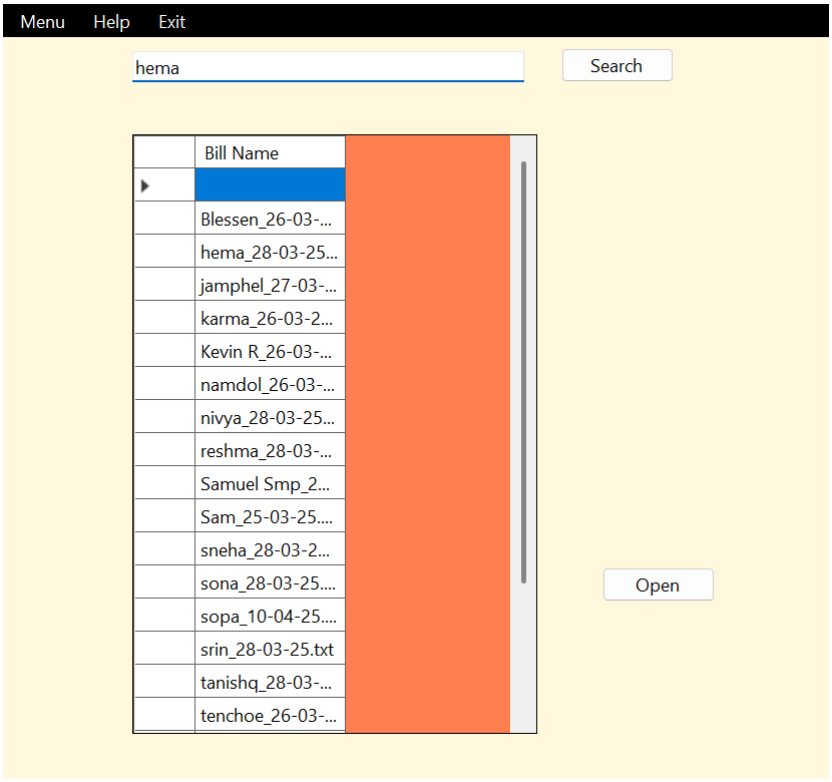
Supplier_id	MName	Contact	Address	Click to Add
1	Sun Pharmace	+91-22-4324 43	Sun House, CT:	
2	Dr. Reddy's Lat	+91-40-4900 29	8-2-337, Road	
3	Cipla Ltd	+91-22-2308 28	Cipla House, Pe	
4	Lupin Limited	+91-22-6640 31	Lupin Ltd., 3rd	
5	Aurobindo Pha	+91-40-6672 5(	Aurobindo Pha	
(New)				

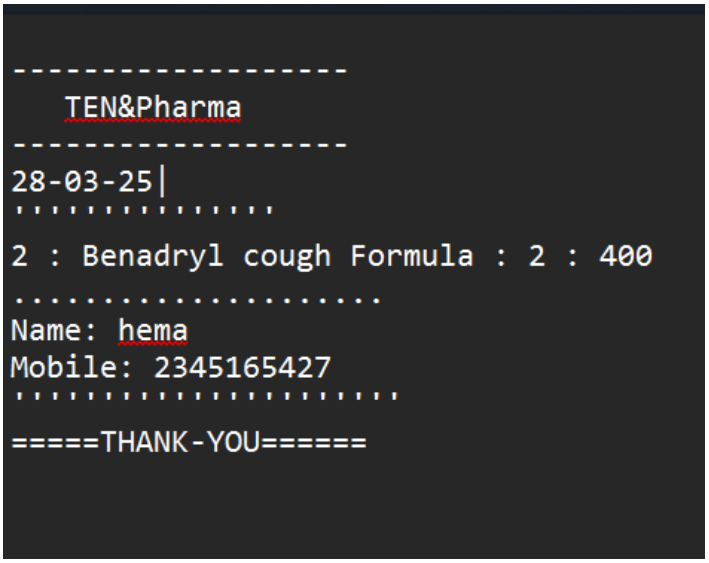


	Drug_Id	Name	Manufacture_id	Price	Expiry_date	Stock_quantity
▶	1	MetroGyl	1102	120	30-07-2025	482
	2	Benadryl cough...	1103	200	31-10-2025	30
	3	Gaviscon Liquid	1104	250	31-10-2025	483
	4	Rabvix S Vaccine	1105	300	01-01-2025	488
	6	Asthalin Pump	1106	310	31-03-2025	30
	25	paracethamol	11	23	10-06-2025	-1
*						

Drug_Id	Name	Manufacture	Price	Expiry_date	Stock_quant	*Click to Add*
1	MetroGyl	1102	120	30-07-2025	492	
2	Benadryl cough	1103	200	31-10-2025	86	
3	Gaviscon Liquic	1104	250	31-10-2025	498	
4	Rabvix S Vaccin	1105	300	01-01-2025	493	
6	Asthalin Pump	1106	307	31-03-2025	30	
(New)						

Bills:


Menu
Help
Exit
hema
Search
Bill Name
Blessen_26-03-...
hema_28-03-25...
jamphel_27-03-...
karma_26-03-2...
Kevin R_26-03-...
namdol_26-03-...
nivya_28-03-25...
reshma_28-03-...
Samuel Smp_2...
Sam_25-03-25....
sneha_28-03-2...
sona_28-03-25....
sopa_10-04-25....
srin_28-03-25.txt
tanishq_28-03-...
tenchoe_26-03-...
Open


TEN&Pharma
28-03-25
2 : Benadryl cough Formula : 2 : 400
Name: hema
Mobile: 2345165427
=====THANK-YOU======

4 SYSTEM CONFIGURATION

4.6 Hardware requirements

RAM	Recommended 8.00 GB
Hard disk	260 GB
Processor	Intel(R) Core (TM)i3-1005G1
Processing speed	CPU @ 1.20 GHz, 1190 MHz

4.7 Software Requirements:

Front end	ASP.NET
Back end	MS ACCESS
Tools	Visual Studio
Operating System	Microsoft Windows 11

5 DETAILS OF SOFTWARE

A development process consists of various phases, each phase ending with a defined output. The phases are performed in an order specified by the process model being followed. The main reason for having a phased process is that it breaks the problem of developing software into successfully performing a set of phases, each handling a different concern of software development.

This ensures that the cost of development is lower than what it would have been if the whole problem were tackled together. A phased development process is central to the software engineering approach for solving the software crisis.

5.1 Overview of Front End

VB.NET (Visual Basic .NET) is a programming language developed by Microsoft, primarily used for building Windows applications with a graphical user interface (GUI) using Windows Forms (WinForms). It provides an easy drag-and-drop interface in Visual Studio for designing UI component like TextBox, Button, Label, ComboBox, DataGridView, and PictureBox. Event-driven programming is a key feature where actions such as button clicks or text changes trigger specific functions. VB.NET supports database connectivity using ADO.NET, allowing seamless interaction with MySQL SQL Server, or Access databases through SQL queries. File handling is another essential feature, enabling reading and writing of text files using System.IO.The language also supports UI customization, such as changing themes dynamically with ColorDialog, and printing functionality using PrintDocument for generating invoices or reports. Once the application is complete, it can be deployed as an executable (.exe) through Visual Studio's Publish feature, making VB.NET a robust choice for Windows desktop application development.

5.2 Overview of Back End

The back-end of the Pharmacy Management System is developed using MS Access (PharmacyDB.accdb) to efficiently handle login authentication, medicine inventory, billing, and customer tracking. The Users Table stores login credentials with fields like UserId, UserName, and Password, ensuring secure system access. The MedicineTable manages inventory with fields such as MedId, MedName, MedCategory, MedQuantity, Path, and Price. Customer transactions are recorded in the CustomerDetail Table using fields like CustId, MobileNo, Date, and Quantity, while financial records are maintained in the Income Table with Id and Amount. The system utilizes ADO.NET in VB.NET to connect with the database and execute SQL operations (INSERT, UPDATE, DELETE, SELECT), with DataGridView components dynamically displaying records, ensuring smooth pharmacy operations and accurate data management.

5.3 About the Platform

Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. It was unveiled in November 2007, with the first commercial Android device launched in September 2008.

It is free and open-source software; its source code is known as Android Open-Source Project (AOSP), which is primarily licensed under the Apache License. However most Android devices ship with additional proprietary software pre-installed, most notably Google Mobile Services (GMS) which includes core apps such as Google Chrome, the digital distribution platform Google Play and associated Google Play Services development platform. About 70 percent of Android smartphones run Google's ecosystem; competing Android ecosystems and forks include Fire OS (developed by Amazon) or LineageOS. However, the "Android" name and logo are trademarks of Google which impose standards to restrict "uncertified" devices outside their ecosystem to use Android branding.

The source code has been used to develop variants of Android on a range of other electronics, such as game consoles, digital cameras, portable media players, PCs and others, each with a specialized user interface. Some well-known derivatives include Android TV for televisions and Wear OS for wearables, both developed by Google. Software packages on Android, which use the APK format, are generally distributed through proprietary application stores like Google Play Store, Samsung Galaxy Store, Huawei AppGallery, Cafe Bazaar, and GetJar, or open-source platforms like Aptoide or F-Droid.

Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthly active users, the largest installed base of any operating system, and as of January 2021, the Google Play Store features over 3 million apps.[16] The current stable version is Android 11, released on September 8, 2020.

TESTING

6 TESTING

Testing is a vital part of software development, and it is important to start it as early as possible, and to make testing a part of the process of deciding requirements. To get the most useful perspective on your development project, it is worthwhile devoting some thought to the entire lifecycle including how feedback from users will influence the future of the application. The tools and techniques we've discussed in this book should help your team to be more responsive to changes without extra cost, despite the necessarily wide variety of different development processes. Nevertheless, new tools and process improvements should be adopted gradually, assessing the results after each step.

Testing is part of a lifecycle. The software development lifecycle is one in which you hear of a need, you write some code to fulfil it, and then you check to see whether you have pleased the stakeholders—the users, owners, and other people who have an interest in what the software does. Hopefully they like it, but would also like some additions or changes, so you update or augment your code; and so the cycle continues, or every few years,

SOFTWARE DEVELOPMENT LIFE CYCLE

Testing is a proxy for the customer. You could conceivably do your testing by releasing it into the wild and waiting for the complaints and compliments to come back. Some companies have been accused of having such a strategy as their business model even before it became fashionable. But on the whole, the books are better balanced by trying to make sure that the software will satisfy the customer before we hand it over. This portal **"TRAVEL WITH US"** is developed using Incremental Model and Waterfall Model.

SOFTWARE TESTING TYPES:

1. FUNTIONAL TESTING

This type of testing ignores the internal parts and focus on the output is as per requirement or not.
They are:

Black box testing – Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

White box testing – This testing is based on knowledge of the internal logic of an application's code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

Unit testing – Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. May require

developing test drive modules or test harnesses.

System testing – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

Acceptance testing -Normally this type of testing is done to verify if system meets the customer specified requirements. User or customers do this testing to determine whether to accept application. **Alpha testing** – In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

2. NON-FUNCTIONAL TESTING

Security testing – Can system be penetrated by any hacking way. Testing how well the system protects against unauthorized internal or external access. Checked if system, database is safe from external attacks.

Usability testing – User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point, basically system navigation is checked in this testing.

7 CONCLUSION AND FUTURE ENHANCEMENT

The Pharmacy Management System provides a streamlined solution for managing key pharmacy operations such as secure login, medicine inventory control, customer transaction tracking, and income monitoring. By using MS Access as the database and ADO.NET with VB.NET for connectivity, the system ensures reliable performance and data integrity. With the integration of DataGridView, users can view and interact with records dynamically, enhancing usability. Overall, the system improves efficiency, reduces manual errors, and offers a user-friendly interface tailored for pharmacy workflows.

Future Enhancements:

In the future, the system can be enhanced by integrating features such as barcode scanning for faster medicine entry, automated low-stock alerts, SMS/email notifications for order updates, and cloud-based database support for remote access. Additionally, implementing role-based access control, sales analytics dashboards, and multi-user support can make the system more scalable and suitable for larger pharmacy chains.

8 BIBILIOGRAPHY

1. MS Access Database Documentation – Microsoft Learn. Retrieved from: https://learn.microsoft.com/en-us/office/client-developer/access/

2. ADO.NET Overview – Microsoft Docs. Retrieved from: https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/

3. VB.NET Programming Guide – Microsoft Learn. Retrieved from: https://learn.microsoft.com/en-us/dotnet/visual-basic/

4. Pharmacy Management System Concepts – ResearchGate, Academic Articles

5. SQL Query Language Basics – W3Schools. Retrieved from: https://www.w3schools.com/sql/

6. User Interface Design for Desktop Applications – Nielsen Norman Group. Retrieved from: https://www.nngroup.com/

7. Software Engineering: A Practitioner's Approach – Roger S. Pressman, McGrawHill Education.

8. Designing Data-Intensive Applications – Martin Kleppmann, O'Reilly Media.

9. Pharmaceutical Information Systems: Design and Implementation – WHO Publications, Health Information Systems Resources.

10. Visual Basic .NET Programming for Beginners – Bryan Newsome, Wiley Publishing

11. Fundamentals of Database Systems – Elmasri & Navathe, Pearson Education.

12. GitHub Documentation – Guide to version control and collaboration. Retrieved from: https://docs.github.com/

9 APPENDICES A – Table Structure

1. Login Table

Table Name: Login
Description: Stores login credentials for admin**.**

UserId	UserName	Password	Click to Add
admin	karma	1234	
Admin001	Tenzin	1234	
Admin002	Sam Saheb	1234	
Admin003	abhi	1234	
*			

2.DrugTable

Table Name: Drug Table
Description: Stored Drug detail available in pharmacist, expiry date and price

Drug_Id	Name	Manufacture	Price	Expiry_date	Stock_quant	Click to Add
1	MetroGyl	1102	120	30-07-2025	492	
2	Benadryl cough	1103	200	31-10-2025	86	
3	Gaviscon Liquic	1104	250	31-10-2025	498	
4	Rabvix S Vaccin	1105	300	01-01-2025	493	
6	Asthalin Pump	1106	307	31-03-2025	30	
(New)						

3.Patient Detail

Table Name: Patient Table

Description: Stored Patient Detail Including Name, Phone no and DOB

Patient

Patient_id	PName	ContactNo	DOB
1	Sam	9879861684	31-03-2005
2	Tenzin	9879861684	12-03-2005
3	Samuel Smp	9879861684	31-3-2005
4	Kevin R	8000643164	31-03-2006
5	Blessen	9879861684	31-03-2005
6	tenchoe	1234567890	23-4-2004
7	karma	12345432432	12-02-2005
8	namdol	9876543456	12-09-2005
9	jamphel	3353567547	02-12-2004
10	tenzinchoenzoı	6573452678	12-06-2005
11	hema	2345165427	12-03-2004
12	sneha	2341543267	12-5-2005
13	tanishq	2345146782	01-01-2005
14	srin	1235265142	12-07-2006
15	bhumi	2463751890	12-01-2006
16	sona	1234132454	12-09-2005
17	nivya	2431562476	12-09-2008
18	reshma	4563234565	23-09-2007
19	tsewang	1234315625	16-09-2004
20	zenha	2343157898	09-03-2005
21	sopa	3452671652	09-08-2006
(New)			

5.Sales Detail

Table Name: Sales Table
Description: Stored sales detail

Sales_id	SalesDate	PatientName	DrugName	TotalAmoun	Quantity
1	25-03-25	Sam	1 : MetroGyl : 2	890	
2	26-03-25	Tenzin	1 : MetroGyl : 2	840	
3	26-03-25	Samuel Smp	4 : Rabvix S Vac	970	
4	26-03-25	Kevin R	7 : Asthaline Ta	440	
5	26-03-25	Blessen	2 : Benadryl cou	2320	
6	26-03-25	tenchoe	1 : MetroGyl : 2	640	
7	26-03-25	karma	4 : Rabvix S Vac	3300	
(New)					

4.Supplier Detail

Table Name: Supplier Table
Description: Stored supplier of contact and address who supply the medicine

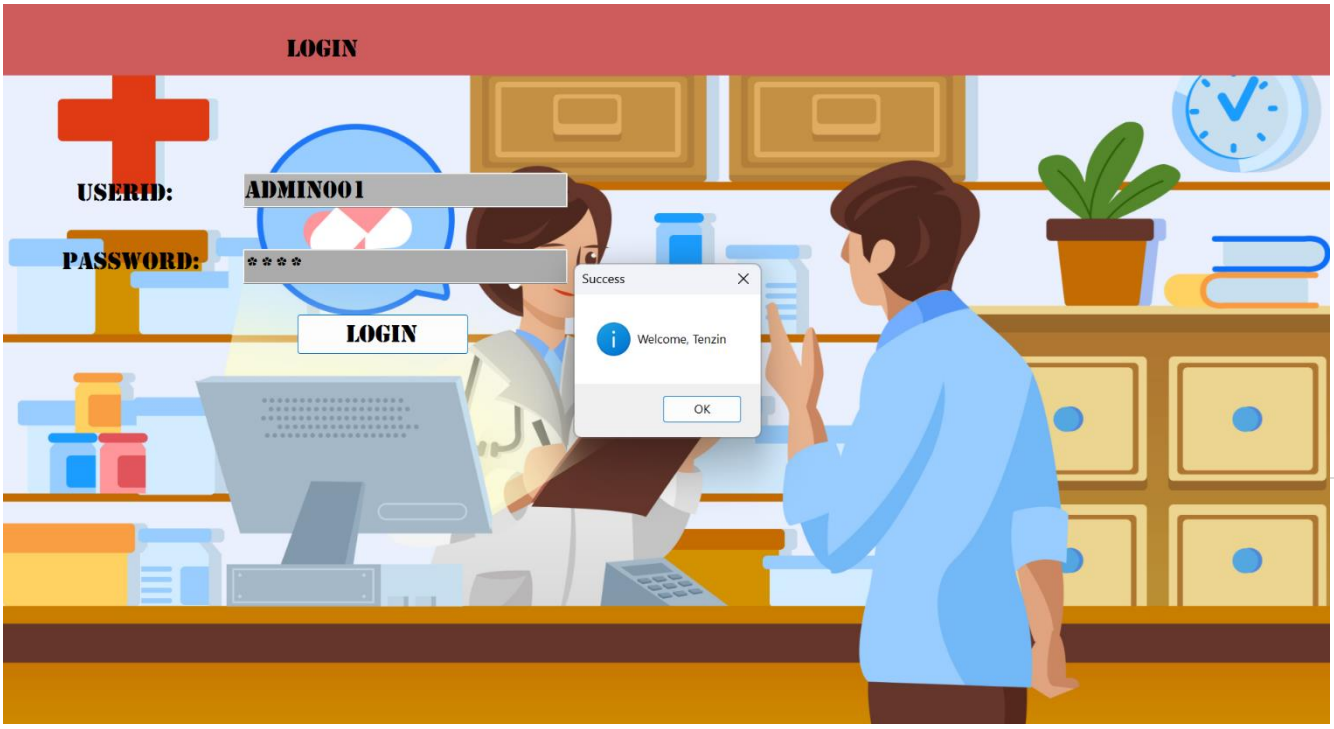
Supplier_id	MName	Contact	Address	Click to Add
1	Sun Pharmace	+91-22-4324 43	Sun House, CT	
2	Dr. Reddy's Lab	+91-40-4900 29	8-2-337, Road	
3	Cipla Ltd	+91-22-2308 28	Cipla House, Pe	
4	Lupin Limited	+91-22-6640 31	Lupin Ltd., 3rd	
5	Aurobindo Pha	+91-40-6672 50	Aurobindo Pha	
(New)				

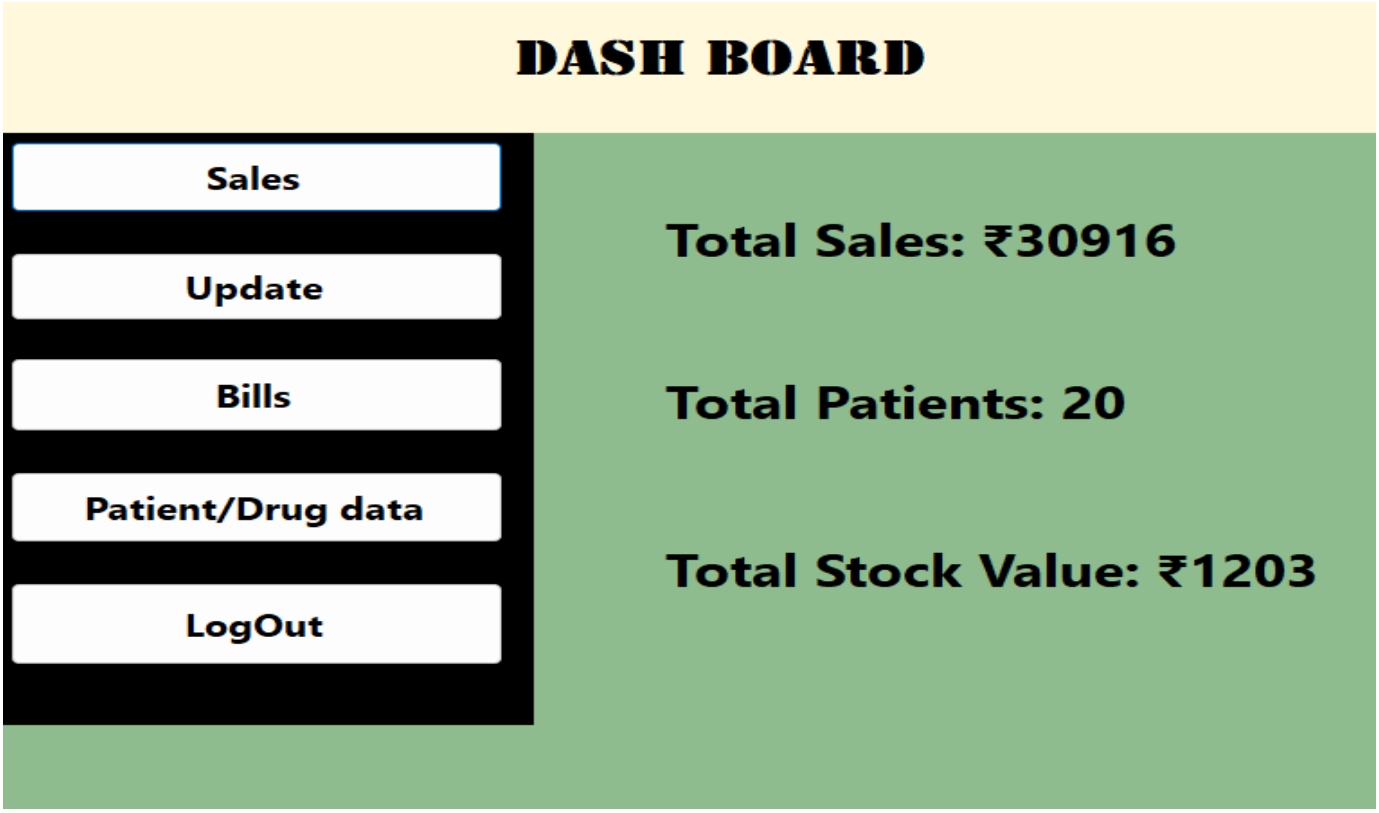
5.Billimg Record

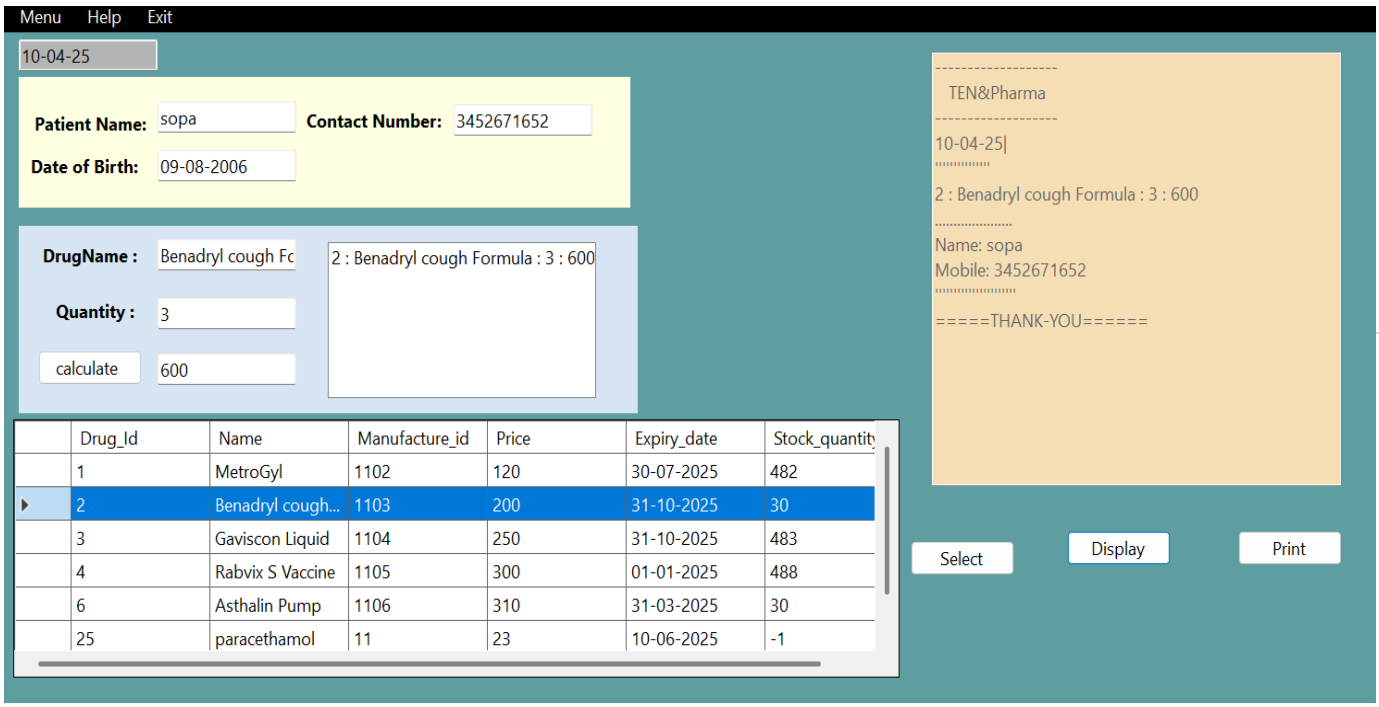
bhumi_28-03-25	28-03-2025 12:53	Text Document
Blessen_26-03-25	26-03-2025 19:40	Text Document
hema_28-03-25	28-03-2025 11:11	Text Document
jamphel_27-03-25	27-03-2025 15:44	Text Document
karma_26-03-25	26-03-2025 20:00	Text Document
Kevin R_26-03-25	26-03-2025 19:22	Text Document
namdol_26-03-25	26-03-2025 20:44	Text Document
nivya_28-03-25	28-03-2025 14:15	Text Document
reshma_28-03-25	28-03-2025 14:54	Text Document
Sam_25-03-25	25-03-2025 20:06	Text Document
Samuel Smp_26-03-25	26-03-2025 12:57	Text Document
sneha_28-03-25	28-03-2025 11:22	Text Document
sona_28-03-25	28-03-2025 14:05	Text Document
sopa_10-04-25	10-04-2025 22:23	Text Document
srin_28-03-25	28-03-2025 12:47	Text Document
tanishq_28-03-25	28-03-2025 12:11	Text Document
tenchoe_26-03-25	26-03-2025 19:42	Text Document

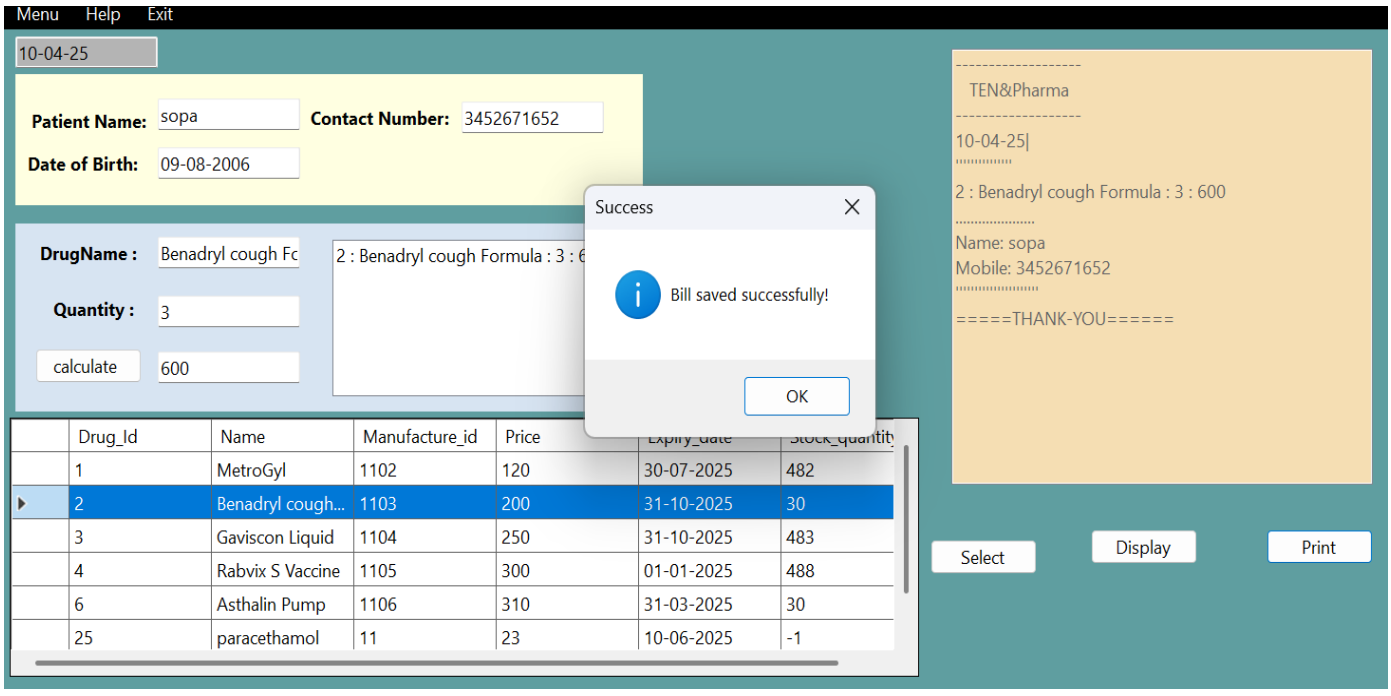
10 APPENDICES B - SCREENSHOTS

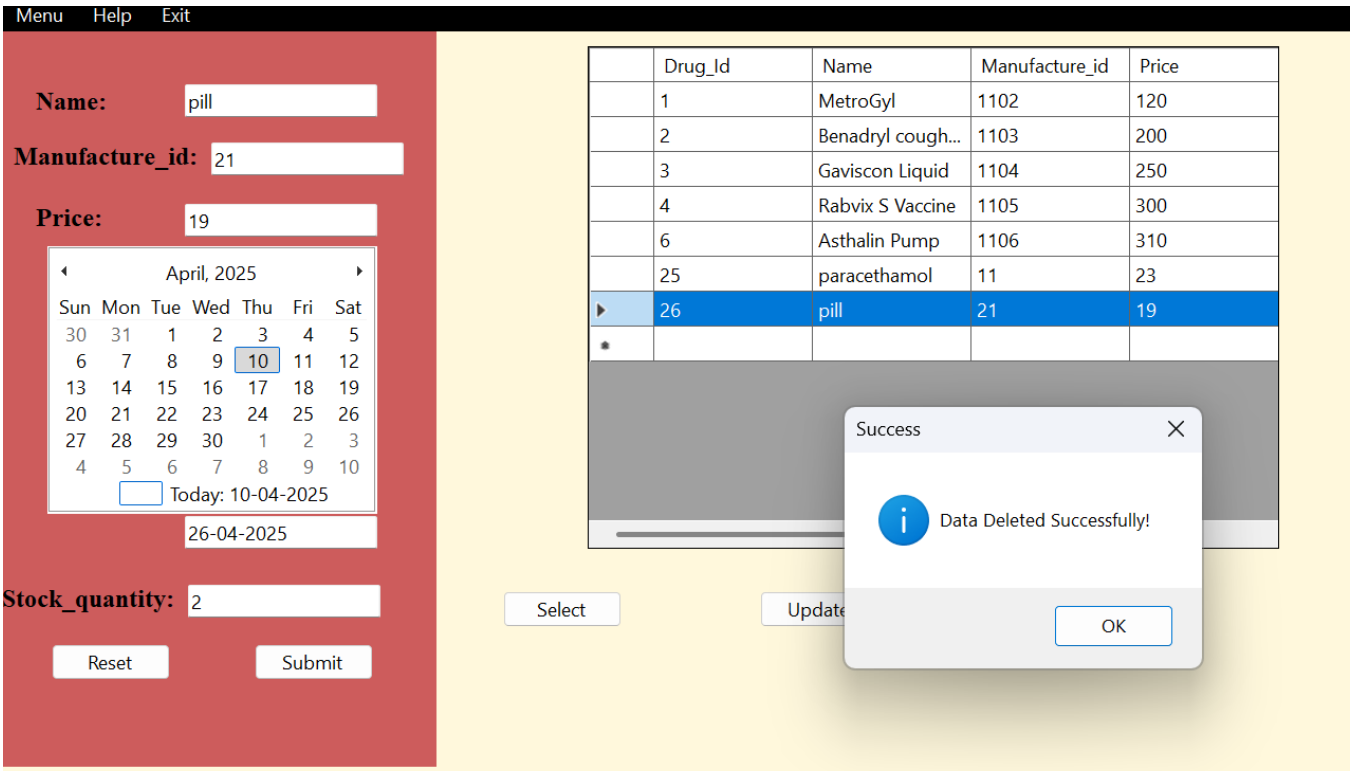
Login:

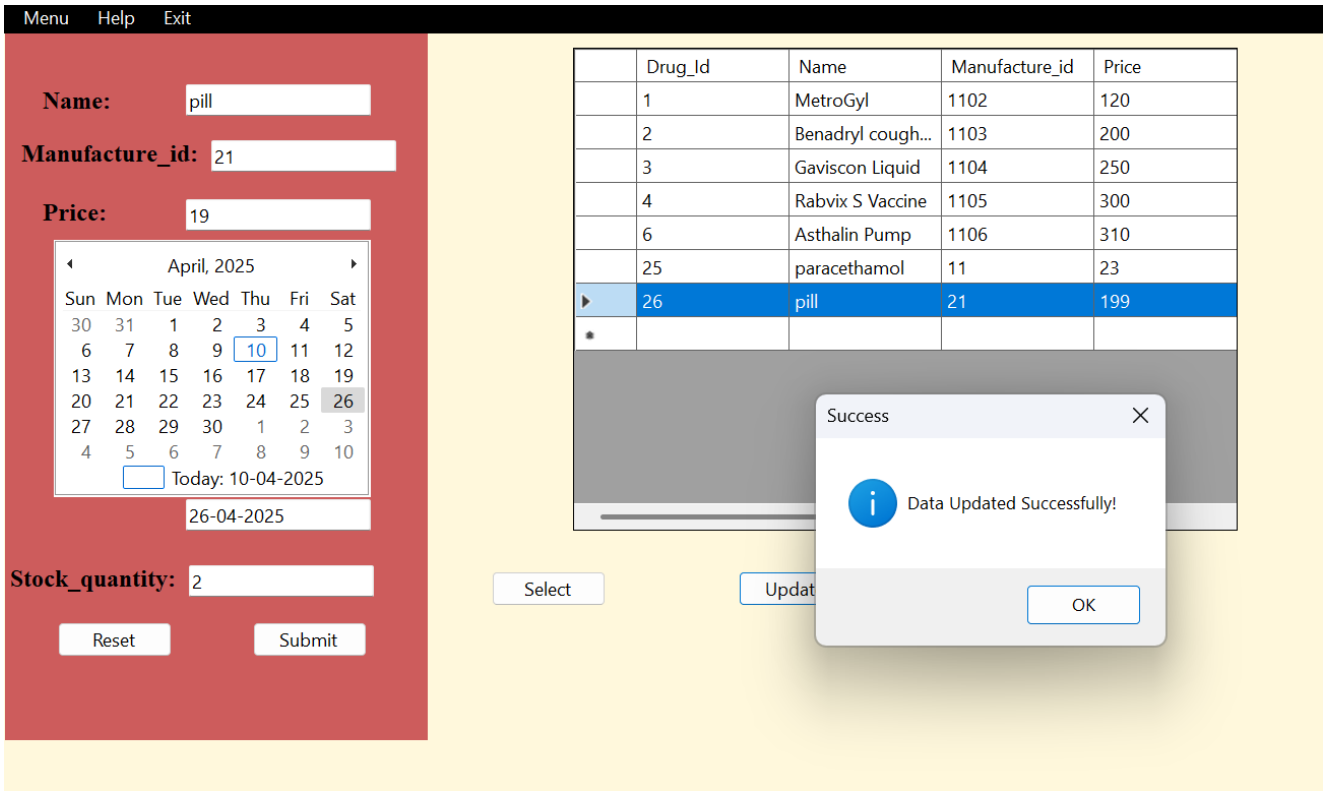


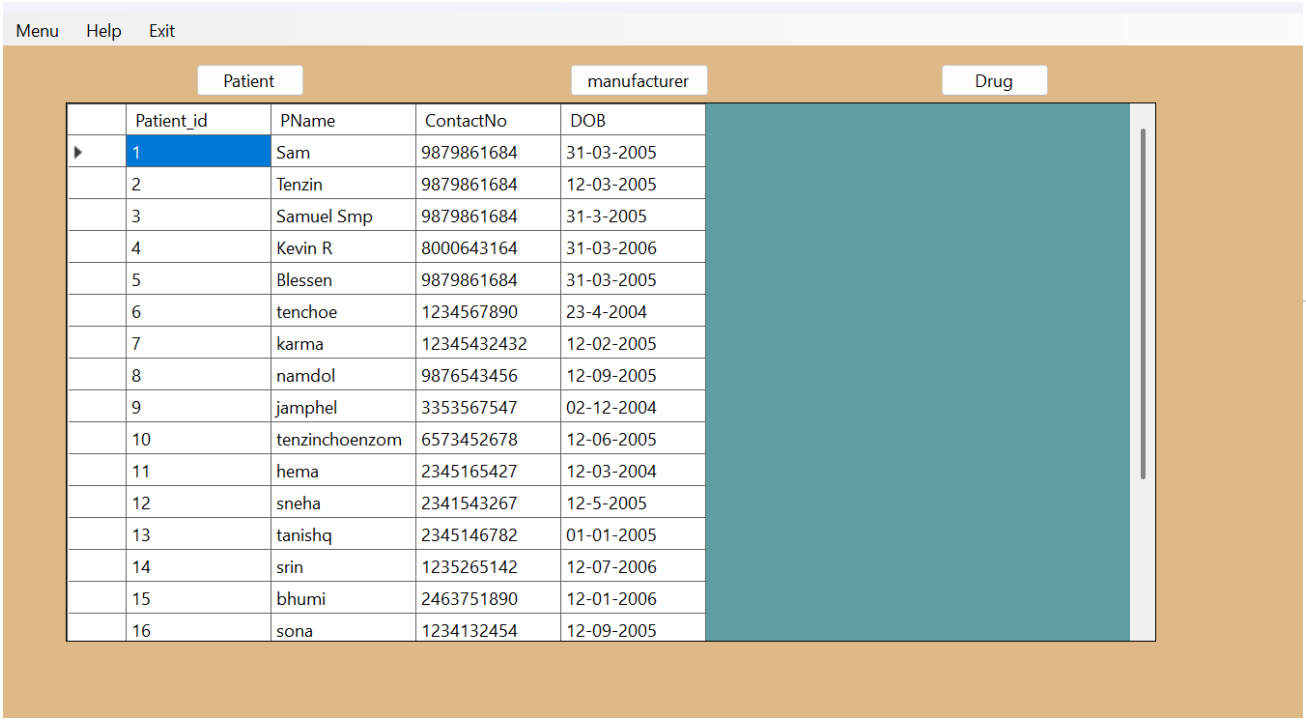



Menu Help Exit
10-04-25
Patient Name: sopa
Contact Number: 3452671652
Date of Birth: 09-08-2006
DrugName : Benadryl cough Fc
2 : Benadryl cough Formula : 3 : 600
Quantity : 3
calculate 600
TEN&Pharma
10-04-25
2 : Benadryl cough Formula : 3 : 600
Name: sopa
Mobile: 3452671652
=====THANK-YOU======
Drug_Id Name Manufacture_id Price Expiry_date Stock_quantity
1 MetroGyl 1102 120 30-07-2025 482
2 Benadryl cough... 1103 200 31-10-2025 30
3 Gaviscon Liquid 1104 250 31-10-2025 483
4 Rabvix S Vaccine 1105 300 01-01-2025 488
6 Asthalin Pump 1106 310 31-03-2025 30
25 paracethamol 11 23 10-06-2025 -1
Select
Display
Print

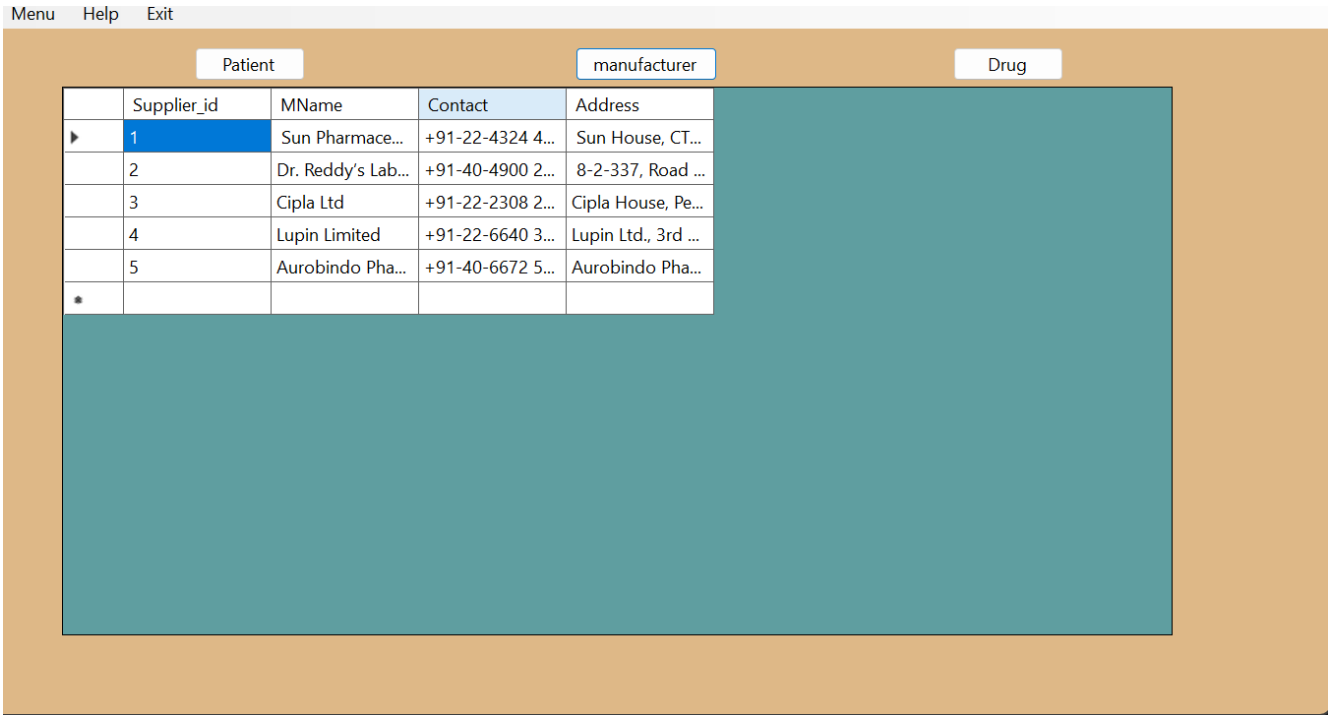

Menu Help Exit
10-04-25
Patient Name: sopa
Contact Number: 3452671652
Date of Birth: 09-08-2006
DrugName : Benadryl cough Fc
2 : Benadryl cough Formula : 3 : 6
Quantity : 3
calculate 600
Success
Bill saved successfully!
OK
TEN&Pharma
10-04-25
2 : Benadryl cough Formula : 3 : 600
Name: sopa
Mobile: 3452671652
=====THANK-YOU======
Drug_Id Name Manufacture_id Price
1 MetroGyl 1102 120 30-07-2025 482
2 Benadryl cough... 1103 200 31-10-2025 30
3 Gaviscon Liquid 1104 250 31-10-2025 483
4 Rabvix S Vaccine 1105 300 01-01-2025 488
6 Asthalin Pump 1106 310 31-03-2025 30
25 paracethamol 11 23 10-06-2025 -1
Select
Display
Print


Menu Help Exit
Name: pill
Manufacture_id: 21
Price: 19
April, 2025
Sun Mon Tue Wed Thu Fri Sat
30 31 1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 1 2 3
4 5 6 7 8 9 10
Today: 10-04-2025
26-04-2025
Stock_quantity: 2
Reset
Submit
Drug_Id Name Manufacture_id Price
1 MetroGyl 1102 120
2 Benadryl cough... 1103 200
3 Gaviscon Liquid 1104 250
4 Rabvix S Vaccine 1105 300
6 Asthalin Pump 1106 310
25 paracethamol 11 23
26 pill 21 19
Success
Data Deleted Successfully!
OK
Select
Update

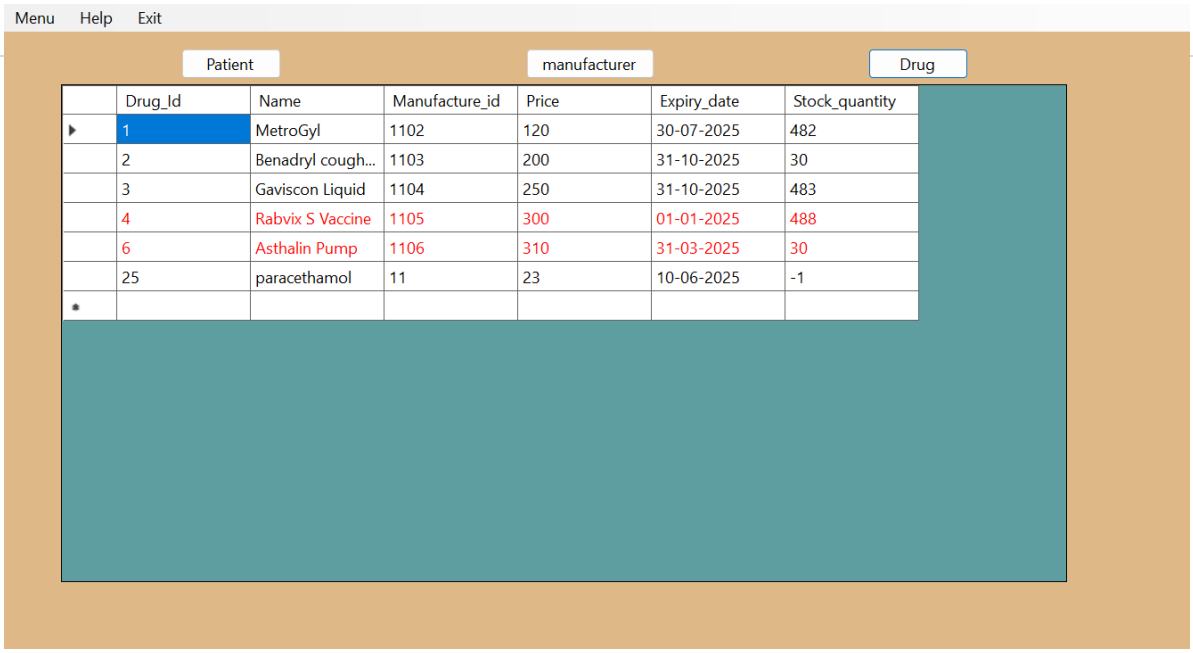

Menu Help Exit
Name: pill
Manufacture_id: 21
Price: 19
April, 2025
Sun Mon Tue Wed Thu Fri Sat
30 31 1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 1 2 3
4 5 6 7 8 9 10
Today: 10-04-2025
26-04-2025
Stock_quantity: 2
Reset
Submit
Drug_Id Name Manufacture_id Price
1 MetroGyl 1102 120
2 Benadryl cough... 1103 200
3 Gaviscon Liquid 1104 250
4 Rabvix S Vaccine 1105 300
6 Asthalin Pump 1106 310
25 paracethamol 11 23
26 pill 21 199
Success
Data Updated Successfully!
OK
Select
Updat



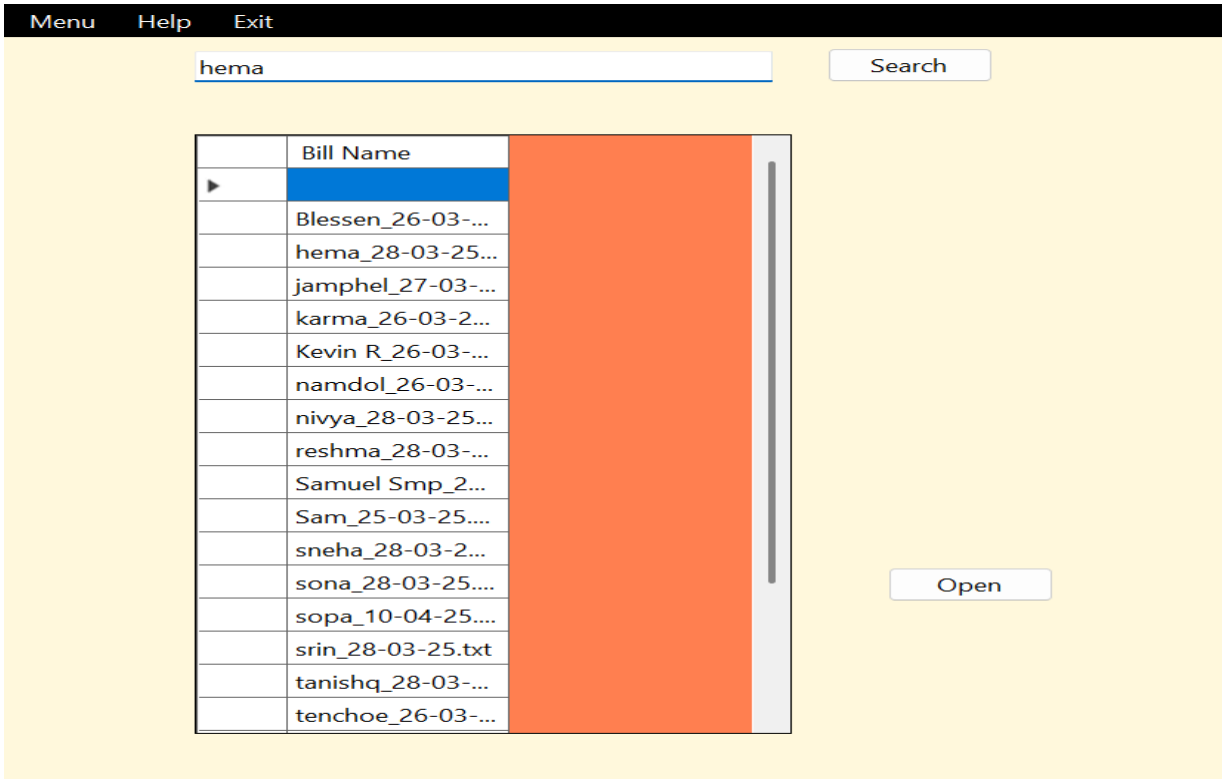
	Patient_id	PName	ContactNo	DOB
▶	1	Sam	9879861684	31-03-2005
	2	Tenzin	9879861684	12-03-2005
	3	Samuel Smp	9879861684	31-3-2005
	4	Kevin R	8000643164	31-03-2006
	5	Blessen	9879861684	31-03-2005
	6	tenchoe	1234567890	23-4-2004
	7	karma	12345432432	12-02-2005
	8	namdol	9876543456	12-09-2005
	9	jamphel	3353567547	02-12-2004
	10	tenzinchoenzom	6573452678	12-06-2005
	11	hema	2345165427	12-03-2004
	12	sneha	2341543267	12-5-2005
	13	tanishq	2345146782	01-01-2005
	14	srin	1235265142	12-07-2006
	15	bhumi	2463751890	12-01-2006
	16	sona	1234132454	12-09-2005

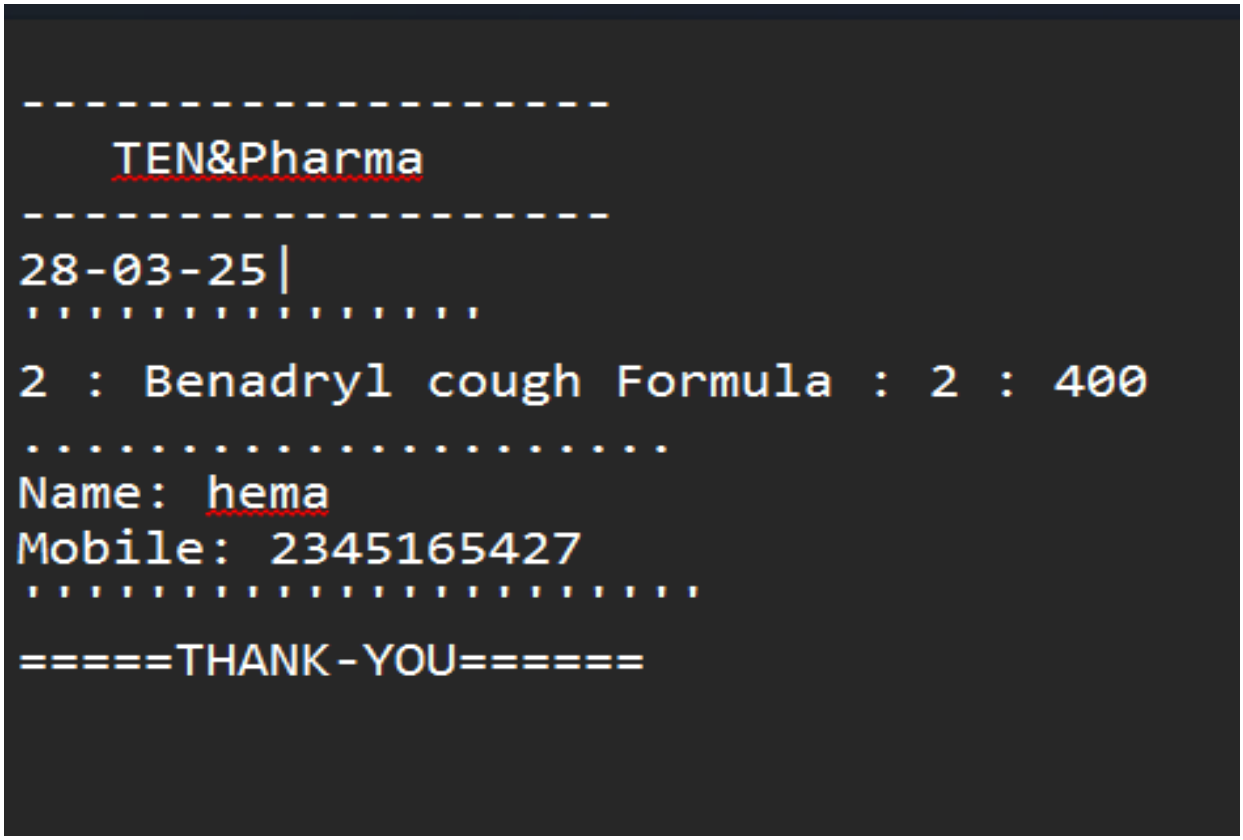


	Supplier_id	MName	Contact	Address
▶	1	Sun Pharmace...	+91-22-4324 4...	Sun House, CT...
	2	Dr. Reddy's Lab...	+91-40-4900 2...	8-2-337, Road ...
	3	Cipla Ltd	+91-22-2308 2...	Cipla House, Pe...
	4	Lupin Limited	+91-22-6640 3...	Lupin Ltd., 3rd ...
	5	Aurobindo Pha...	+91-40-6672 5...	Aurobindo Pha...
*				



	Drug_Id	Name	Manufacture_id	Price	Expiry_date	Stock_quantity
▶	1	MetroGyl	1102	120	30-07-2025	482
	2	Benadryl cough...	1103	200	31-10-2025	30
	3	Gaviscon Liquid	1104	250	31-10-2025	483
	4	Rabvix S Vaccine	1105	300	01-01-2025	488
	6	Asthalin Pump	1106	310	31-03-2025	30
	25	paracethamol	11	23	10-06-2025	-1
*						




TEN&Pharma
28-03-25
2 : Benadryl cough Formula : 2 : 400
Name: hema
Mobile: 2345165427
=====THANK-YOU======

11 APPENDICES C – TEST CASES

Test ID	Test Scenario Name	Test Data	Expected Result	Actual Result	Test Status
T1	User Login	UserId: Invalid Password: Invalid	Error Message: Incorrect credentials	Error Message: Incorrect credentials	Pass
T2	User Login	UserId: Valid Password: Invalid	Error Message: Incorrect credentials	Error Message: Incorrect credentials	Pass
T3	User Login	UserId: Valid Password: Valid	Dashboard opens	Dashboard opens	Pass
T4	Add Medicine	MedName: Paracetamol Quantity: 50 Price: 20	Medicine added successfully	Medicine added successfully	Pass
T5	Add Medicine	MedName: "" Quantity: 10 Price: 15	Error: Medicine name required	Error: Medicine name required	Pass
T6	Update Medicine Stock	MedId: 101 Quantity: 100	Stock updated successfully	Stock updated successfully	Pass
T7	Delete Medicine	MedId: Invalid ID	Error: Medicine not found	Error: Medicine not found	Pass
T8	Customer Billing	Selected medicines with valid quantities	Bill generated with total amount	Bill generated correctly	Pass
T9	Customer Billing	Selected medicines with stock = 0	Error: Out of stock	Error: Out of stock	Pass
T10	Income Report	After multiple bills generated	Updated total income shown	Updated total income shown	Pass
T11	Search Medicine	MedName: "Paracetamol"	Medicine details displayed	Medicine details displayed	Pass
T12	User Registration	New UserID with strong password	User account created successfully	User account created successfully	Pass
T13	User Registration	Existing UserID	Error: User already exists	Error: User already exists	Pass

12 APPENDICES D - SOURCE CODE

Login:

```
Imports System.Data.OleDb

Public Class Form1
' Connection String
Dim con As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\tsewang\source\repos\pharmacy management\Database1.accdb")

Private Sub btnLogin_Click(sender As Object, e As EventArgs) Handles btnLogin.Click
Try
' Open connection
con.Open()

' Query to check UserId and Password
Dim query As String = "SELECT UserName FROM Admin WHERE UserId = @UserId AND Password =
@Password"
Dim cmd As New OleDbCommand(query, con)

' Add parameters
cmd.Parameters.AddWithValue("@UserId", txtUserId.Text)
cmd.Parameters.AddWithValue("@Password", txtPassword.Text)

' Execute Query
Dim userName As Object = cmd.ExecuteScalar()

' Check if user exists
If userName IsNot Nothing Then
MessageBox.Show("Welcome, " & userName.ToString(), "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information)
```

```
 ' Open Form2 (Main Dashboard) after successful login
 Dim dashboard As New Form4()
 dashboard.Show()
 Me.Hide() ' Hide the login form
 Else
MessageBox.Show("Incorrect UserId or Password!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
End If

 Catch ex As Exception
MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
Finally
con.Close()
End Try

txtPassword.Text = ""
txtUserId.Text = ""

End Sub

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

Private Sub Label3_Click(sender As Object, e As EventArgs) Handles Label3.Click

End Sub

Private Sub txtUserId_TextChanged(sender As Object, e As EventArgs) Handles txtUserId.TextChanged

End Sub
End Class
```

Add and Update of drug:

```
Imports System.Data.OleDb

Public Class Form2
' Database Connection
Dim con As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\tsewang\source\repos\pharmacy management\Database1.accdb")

' Load Data into DataGridView
Private Sub LoadData()
Try
con.Open()
Dim query As String = "SELECT * FROM DrugTable"
Dim adapter As New OleDbDataAdapter(query, con)
Dim dt As New DataTable()
adapter.Fill(dt)
DataGridView1.DataSource = dt
Catch ex As Exception
MessageBox.Show("Error loading data: " & ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
Finally
con.Close()
End Try
End Sub

' Display Selected Date in txtExpiray_date
Private Sub MonthCalendar1_DateSelected(sender As Object, e As DateRangeEventArgs) Handles
MonthCalendar1.DateSelected
txtExpiray_date.Text = MonthCalendar1.SelectionRange.Start.ToShortDateString()
End Sub
```

```
' Submit Data to DrugTable
Private Sub btnSubmit_Click(sender As Object, e As EventArgs) Handles btnSubmit.Click
Try
, Stock_quantity) VALUES (@Name, @Manufacture_id, @Price, @Expiry_date, @Stock_quantity)"
 Dim cmd As New OleDbCommand(query, con)

' Add Parameters
cmd.Parameters.AddWithValue("@Name", txtName.Text)
cmd.Parameters.AddWithValue("@Manufacture_id", Convert.ToInt32(txtManufacture_id.Text))
cmd.Parameters.AddWithValue("@Price", Convert.ToDouble(txtPrice.Text))
cmd.Parameters.AddWithValue("@Expiry_date", txtExpiray_date.Text)
cmd.Parameters.AddWithValue("@Stock_quantity", Convert.ToInt32(txtStock_quantity.Text))

' Execute Query
cmd.ExecuteNonQuery()
MessageBox.Show("Data Added Successfully!", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information)
 Catch ex As Exception
MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
Finally
con.Close()
End Try

' Refresh DataGridView
LoadData()
End Sub

' Select Row Data from DataGridView1
Private Sub btnSelect_Click(sender As Object, e As EventArgs) Handles btnSelect.Click
If DataGridView1.SelectedRows.Count > 0 Then
Dim row As DataGridViewRow = DataGridView1.SelectedRows(0)
```

```
txtName.Text = row.Cells("Name").Value.ToString()
txtManufacture_id.Text = row.Cells("Manufacture_id").Value.ToString()
txtPrice.Text = row.Cells("Price").Value.ToString()
txtExpiray_date.Text = row.Cells("Expiry_date").Value.ToString()
txtStock_quantity.Text = row.Cells("Stock_quantity").Value.ToString()
Else
MessageBox.Show("Please select a row to display details.", "Warning", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
End If
End Sub

' Update Selected Data in DrugTable
Private Sub btnUpdate_Click(sender As Object, e As EventArgs) Handles btnUpdate.Click
Try
con.Open()
Dim query As String = "UPDATE DrugTable SET Name=@Name, Manufacture_id=@Manufacture_id,
Price=@Price, Expiry_date=@Expiry_date, Stock_quantity=@Stock_quantity WHERE
Name=@OldName"
Dim cmd As New OleDbCommand(query, con)

' Add Parameters
cmd.Parameters.AddWithValue("@Name", txtName.Text)
cmd.Parameters.AddWithValue("@Manufacture_id", Convert.ToInt32(txtManufacture_id.Text))
cmd.Parameters.AddWithValue("@Price", Convert.ToDouble(txtPrice.Text))
cmd.Parameters.AddWithValue("@Expiry_date", txtExpiray_date.Text)
cmd.Parameters.AddWithValue("@Stock_quantity", Convert.ToInt32(txtStock_quantity.Text))
cmd.Parameters.AddWithValue("@OldName", txtName.Text) ' Assuming Name is unique

' Execute Query
cmd.ExecuteNonQuery()
MessageBox.Show("Data Updated Successfully!", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information)
```

```
Catch ex As Exception
MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
Finally
con.Close()
End Try

' Refresh DataGridView
LoadData()
End Sub

' Delete Selected Row
Private Sub btnDelete_Click(sender As Object, e As EventArgs) Handles btnDelete.Click
If DataGridView1.SelectedRows.Count > 0 Then
Dim row As DataGridViewRow = DataGridView1.SelectedRows(0)
Dim drugName As String = row.Cells("Name").Value.ToString()

Try
con.Open()
Dim query As String = "DELETE FROM DrugTable WHERE Name=@Name"
Dim cmd As New OleDbCommand(query, con)
cmd.Parameters.AddWithValue("@Name", drugName)
cmd.ExecuteNonQuery()
MessageBox.Show("Data Deleted Successfully!", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information)
Catch ex As Exception
MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
Finally
con.Close()
End Try

' Refresh DataGridView
LoadData()
```

```
Else
MessageBox.Show("Please select a row to delete.", "Warning", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
End If
End Sub

' Form Load - Display DrugTable
Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
LoadData()
End Sub

 ' Reset Form Fields
 Private Sub btnReset_Click(sender As Object, e As EventArgs) Handles btnReset.Click
 txtName.Text = ""
 txtManufacture_id.Text = ""
 txtPrice.Text = ""
 txtExpiray_date.Text = ""
 txtStock_quantity.Text = ""
 End Sub

 ' Navigation Menu Events
 Private Sub SellToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
 SellToolStripMenuItem.Click
 Me.Hide()
 Form3.Show()
 End Sub

 Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
 ExitToolStripMenuItem.Click
 Application.Exit()
 End Sub
```

```
Private Sub LogOutToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
LogOutToolStripMenuItem.Click
Me.Hide()
Form1.Show()
End Sub

Private Sub BillsToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
BillsToolStripMenuItem.Click
Me.Hide()
Form5.Show()
End Sub

Private Sub DashBoardToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
DashBoardToolStripMenuItem.Click
Me.Hide()
Form4.Show()
End Sub

Private Sub PatientDrugToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
PatientDrugToolStripMenuItem.Click
Me.Hide()
Form6.Show()
End Sub


Private Sub UpdateDataToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
UpdateDataToolStripMenuItem.Click
Me.Hide()
Form3.Show()
End Sub
End Class
```

Sales:

```
Imports System.Data.OleDb
Imports System.IO

Public Class Form3
' Database Connection
Dim con As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\tsewang\source\repos\pharmacy management\Database1.accdb")

' Function to Load DrugTable Data into DataGridView
Private Sub LoadDrugTable()
Try
con.Open()
Dim query As String = "SELECT * FROM DrugTable"
Dim adapter As New OleDbDataAdapter(query, con)
Dim dt As New DataTable()
adapter.Fill(dt)
DataGridView1.DataSource = dt
Catch ex As Exception
MessageBox.Show("Error loading DrugTable: " & ex.Message)
Finally
con.Close()
End Try
End Sub

' Form Load Event
Private Sub Form3_Load(sender As Object, e As EventArgs) Handles MyBase.Load
txtSalesDate.Text = DateTime.Now.ToString("dd-MM-yy")
LoadDrugTable()
End Sub
```

```
' btnSelect: Select Drug from DataGridView
Private Sub btnSelect_Click(sender As Object, e As EventArgs) Handles btnSelect.Click
If DataGridView1.SelectedRows.Count > 0 Then
txtDrugName.Text = DataGridView1.SelectedRows(0).Cells("Name").Value.ToString()
Else
MessageBox.Show("Please select a drug!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)
End If
End Sub

' btnAdd: Add Drug to ListBox1 and Calculate Total
Private Sub btnAdd_Click(sender As Object, e As EventArgs) Handles btnAdd.Click
If txtDrugName.Text = "" Or txtQuantity.Text = "" Then
MessageBox.Show("Please enter a valid quantity and select a drug.", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
Return
End If

Dim selectedRow As DataGridViewRow = DataGridView1.SelectedRows(0)
Dim drugId As String = selectedRow.Cells("Drug_Id").Value.ToString()
Dim drugName As String = selectedRow.Cells("Name").Value.ToString()
Dim price As Double = Convert.ToDouble(selectedRow.Cells("Price").Value)
Dim quantity As Integer = Convert.ToInt32(txtQuantity.Text)
Dim totalCost As Double = price * quantity

' Add to ListBox1
Dim item As String = $"{drugId} : {drugName} : {quantity} : {totalCost}"
ListBox1.Items.Add(item)

' Update Total Amount
Dim totalAmount As Double = If(txtTotalAmount.Text = "", 0, Convert.ToDouble(txtTotalAmount.Text))
totalAmount += totalCost
```

```
txtTotalAmount.Text = totalAmount.ToString()

' Update Drug Stock
Dim newStock As Integer = Convert.ToInt32(selectedRow.Cells("Stock_quantity").Value) - quantity
Try
con.Open()
Dim updateQuery As String = "UPDATE DrugTable SET Stock_quantity = @Stock WHERE Drug_Id = @DrugId"
Dim cmd As New OleDbCommand(updateQuery, con)
cmd.Parameters.AddWithValue("@Stock", newStock)
cmd.Parameters.AddWithValue("@DrugId", drugId)
cmd.ExecuteNonQuery()
Catch ex As Exception
MessageBox.Show("Error updating stock: " & ex.Message)
Finally
con.Close()
End Try

' Refresh DrugTable
LoadDrugTable()
End Sub

' btnDisplay: Show Bill and Save to Database
Private Sub btnDisplay_Click(sender As Object, e As EventArgs) Handles btnDisplay.Click
' Format Bill in ListBox2
ListBox2.Items.Clear()
ListBox2.Items.Add("--------------------")
ListBox2.Items.Add("   TEN&Pharma")
ListBox2.Items.Add("--------------------")
ListBox2.Items.Add(txtSalesDate.Text & "|")
ListBox2.Items.Add("""""""""""""")
For Each item In ListBox1.Items
```

```
ListBox2.Items.Add(item)
Next
ListBox2.Items.Add(".....")
ListBox2.Items.Add("Name: " & txtPatientName.Text)
ListBox2.Items.Add("Mobile: " & txtContactNo.Text)
ListBox2.Items.Add("""""""""""""""""")
ListBox2.Items.Add("=====THANK-YOU======")

' Insert into Patient Table
Try
con.Open()
Dim patientQuery As String = "INSERT INTO Patient (PName, ContactNo, DOB) VALUES (@PName,
@ContactNo, @DOB)"
Dim cmdPatient As New OleDbCommand(patientQuery, con)
cmdPatient.Parameters.AddWithValue("@PName", txtPatientName.Text)
cmdPatient.Parameters.AddWithValue("@ContactNo", txtContactNo.Text)
cmdPatient.Parameters.AddWithValue("@DOB", txtDOB.Text)
cmdPatient.ExecuteNonQuery()
Catch ex As Exception
MessageBox.Show("Error inserting patient data: " & ex.Message)
Finally
con.Close()
End Try

' Insert into Sales Table
Try
con.Open()
Dim salesQuery As String = "INSERT INTO Sales (SalesDate, PatientName, DrugName, TotalAmount)
VALUES (@SalesDate, @PatientName, @DrugName, @TotalAmount)"
Dim cmdSales As New OleDbCommand(salesQuery, con)
cmdSales.Parameters.AddWithValue("@SalesDate", txtSalesDate.Text)
cmdSales.Parameters.AddWithValue("@PatientName", txtPatientName.Text)
```

```
cmdSales.Parameters.AddWithValue("@DrugName", String.Join(", ", ListBox1.Items.Cast(Of
String).ToArray()))
cmdSales.Parameters.AddWithValue("@TotalAmount", txtTotalAmount.Text)
cmdSales.ExecuteNonQuery()
Catch ex As Exception
MessageBox.Show("Error inserting sales data: " & ex.Message)
Finally
con.Close()
End Try
End Sub

' btnPrint: Save Bill to File
Private Sub btnPrint_Click(sender As Object, e As EventArgs) Handles btnPrint.Click
Try
Dim billFolder As String = "C:\Users\tsewang\source\repos\pharmacy management\Bill"
If Not Directory.Exists(billFolder) Then
Directory.CreateDirectory(billFolder)
End If

Dim billFile As String = Path.Combine(billFolder, $"{txtPatientName.Text}_{txtSalesDate.Text}.txt")
Using writer As New StreamWriter(billFile)
For Each item In ListBox2.Items
writer.WriteLine(item.ToString())
Next
End Using

MessageBox.Show("Bill saved successfully!", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information)
Catch ex As Exception
MessageBox.Show("Error saving bill: " & ex.Message)
End Try
```

```
txtPatientName.Text = ""
txtDOB.Text = ""
txtContactNo.Text = ""
txtTotalAmount.Text = ""
txtQuantity.Text = ""
txtDrugName.Text = ""
ListBox1.Items.Clear()
ListBox2.Items.Clear()
End Sub

Private Sub SellToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
SellToolStripMenuItem.Click

End Sub

Private Sub UpdateDataToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
UpdateDataToolStripMenuItem.Click
Me.Hide()
Form2.Show()
End Sub

Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExitToolStripMenuItem.Click
Application.Exit()
End Sub

Private Sub LogOutToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
LogOutToolStripMenuItem.Click
Me.Hide()
Form1.Show()
End Sub
```

```
Private Sub BillsToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
BillsToolStripMenuItem.Click
Me.Hide()
Form5.Show()

End Sub

Private Sub DashBoardToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
DashBoardToolStripMenuItem.Click
Me.Hide()
Form4.Show()
End Sub

Private Sub PatientDrugToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
PatientDrugToolStripMenuItem.Click
Me.Hide()
Form6.Show()
End Sub

Private Sub Label1_Click(sender As Object, e As EventArgs) Handles Label1.Click

End Sub
End Class
```

Dash Board:

```
Imports System.Data.OleDb

Public Class Form4
' Database Connection
```

```
Dim con As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\tsewang\source\repos\pharmacy management\Database1.accdb")

' Form Load Event - Load Data
Private Sub Form4_Load(sender As Object, e As EventArgs) Handles MyBase.Load
LoadSalesTotal()
LoadPatientTotal()
LoadStockTotal()
End Sub

' Get Total Sales from Sales Table
Private Sub LoadSalesTotal()
Try
con.Open()
Dim query As String = "SELECT SUM(TotalAmount) FROM Sales"
Dim cmd As New OleDbCommand(query, con)
Dim result = cmd.ExecuteScalar()

lblSales.Text = "Total Sales: ₹" & If(IsDBNull(result), 0, result)
Catch ex As Exception
MessageBox.Show("Error fetching sales data: " & ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
Finally
con.Close()
End Try
End Sub

' Get Total Patients from Patient Table
Private Sub LoadPatientTotal()
Try
con.Open()
Dim query As String = "SELECT COUNT(*) FROM Patient"
```

```
Dim cmd As New OleDbCommand(query, con)
Dim result = cmd.ExecuteScalar()

lblPatient.Text = "Total Patients: " & If(IsDBNull(result), 0, result)
Catch ex As Exception
MessageBox.Show("Error fetching patient data: " & ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
Finally
con.Close()
End Try
End Sub

' Get Total Stock Value from DrugTable
Private Sub LoadStockTotal()
Try
con.Open()
Dim query As String = "SELECT SUM(Price) FROM DrugTable"
Dim cmd As New OleDbCommand(query, con)
Dim result = cmd.ExecuteScalar()

lblStock.Text = "Total Stock Value: ₹" & If(IsDBNull(result), 0, result)
Catch ex As Exception
MessageBox.Show("Error fetching stock data: " & ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
Finally
con.Close()
End Try
End Sub

Private Sub btnUpdate_Click(sender As Object, e As EventArgs) Handles btnUpdate.Click
Me.Hide()
Form2.Show()
```

```
End Sub

Private Sub btnSales_Click(sender As Object, e As EventArgs) Handles btnSales.Click
Me.Hide()
Form3.Show()
End Sub

Private Sub btlBills_Click(sender As Object, e As EventArgs) Handles btlBills.Click
Hide
Form5.Show
End Sub

Private Sub btnLogOut_Click(sender As Object, e As EventArgs) Handles btnLogOut.Click
Me.Hide()
Form1.Show()
End Sub

Private Sub lblStock_Click(sender As Object, e As EventArgs) Handles lblStock.Click

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Me.Hide()
Form6.Show()
End Sub
End Class
```

Billing:

```
Imports System.IO

Public Class Form5
```

```
Dim billFolderPath As String = "C:\Users\tsewang\source\repos\pharmacy management\Bill"

' Form Load - Load Bills into DataGridView1
Private Sub Form5_Load(sender As Object, e As EventArgs) Handles MyBase.Load
LoadBillFiles()
End Sub

' Load Bill Files into DataGridView1
Private Sub LoadBillFiles()
If Directory.Exists(billFolderPath) Then
Dim files As String() = Directory.GetFiles(billFolderPath, "*.txt")
Dim dt As New DataTable()
dt.Columns.Add("Bill Name")

For Each file As String In files
dt.Rows.Add(Path.GetFileName(file))
Next

DataGridView1.DataSource = dt
Else
MessageBox.Show("Bill folder not found!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If
End Sub

' Search Bill by Name
Private Sub btnSearch_Click(sender As Object, e As EventArgs) Handles btnSearch.Click
Dim searchText As String = txtSearch.Text.Trim().ToLower()
If searchText = "" Then
LoadBillFiles()
Return
End If
```

```
Dim dt As New DataTable()
dt.Columns.Add("Bill Name")

For Each row As DataGridViewRow In DataGridView1.Rows
If Not row.IsNewRow Then
Dim billName As String = row.Cells("Bill Name").Value.ToString().ToLower()
If billName.Contains(searchText) Then
dt.Rows.Add(row.Cells("Bill Name").Value)
End If
End If
Next

DataGridView1.DataSource = dt
End Sub

' Open Selected Bill
Private Sub btnOpen_Click(sender As Object, e As EventArgs) Handles btnOpen.Click
If DataGridView1.SelectedRows.Count > 0 Then
Dim selectedBill As String = DataGridView1.SelectedRows(0).Cells("Bill Name").Value.ToString()
Dim filePath As String = Path.Combine(billFolderPath, selectedBill)

If File.Exists(filePath) Then
Process.Start("notepad.exe", filePath) ' Opens in Notepad
Else
MessageBox.Show("Bill file not found!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If
Else
MessageBox.Show("Please select a bill from the list.", "Info", MessageBoxButtons.OK,
MessageBoxIcon.Information)
End If
End Sub
```

```
Private Sub SellToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
SellToolStripMenuItem.Click
Me.Hide()
Form3.Show()
End Sub

Private Sub UpdateDataToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
UpdateDataToolStripMenuItem.Click
Me.Hide()
Form2.Show()
End Sub

Private Sub LogOutToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
LogOutToolStripMenuItem.Click
Me.Hide()
Form1.Show()
End Sub

Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExitToolStripMenuItem.Click
Application.Exit()
End Sub

Private Sub DashBoardToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
DashBoardToolStripMenuItem.Click
Me.Hide()
Form4.Show()
End Sub

Private Sub PatientDrugToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
PatientDrugToolStripMenuItem.Click
Me.Hide()
```

```
Form6.Show()
End Sub
End Class
```

Patient, Manufacture and Drug Detail:

```
Imports System.Data.OleDb

Public Class Form6
' Database Connection
Dim con As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\tsewang\source\repos\pharmacy management\Database1.accdb")

' Load Form6 and show Patient details by default
Private Sub Form6_Load(sender As Object, e As EventArgs) Handles MyBase.Load
LoadPatientData()
End Sub

' Function to Load Data into DataGridView
Private Sub LoadData(query As String)
Try
con.Open()
Dim adapter As New OleDbDataAdapter(query, con)
Dim dt As New DataTable()
adapter.Fill(dt)
DataGridView1.DataSource = dt
Catch ex As Exception
MessageBox.Show("Error loading data: " & ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
Finally
con.Close()
```

```
End Try
End Sub

' Load Patient Data
Private Sub LoadPatientData()
Dim query As String = "SELECT * FROM Patient"
LoadData(query)
End Sub

' Load Manufacturer Data
Private Sub LoadManufacturerData()
Dim query As String = "SELECT * FROM Manufacture"
LoadData(query)
End Sub

' Load Drug Data & Highlight Expired Drugs
Private Sub LoadDrugData()
Dim query As String = "SELECT * FROM DrugTable"
LoadData(query)
HighlightExpiredDrugs()
End Sub

' Highlight Expired Drugs in Red
Private Sub HighlightExpiredDrugs()
For Each row As DataGridViewRow In DataGridView1.Rows
If row.Cells("Expiry_date").Value IsNot DBNull.Value Then
Dim expiryDate As Date = Convert.ToDateTime(row.Cells("Expiry_date").Value)
If expiryDate < Date.Today Then
row.DefaultCellStyle.ForeColor = Color.Red ' Expired drugs in red
End If
End If
Next
```

```
End Sub

' Button Click Events
Private Sub btnPatient_Click(sender As Object, e As EventArgs) Handles btnPatient.Click
LoadPatientData()
End Sub

Private Sub btnManufacturer_Click(sender As Object, e As EventArgs) Handles btnmanufacturer.Click
LoadManufacturerData()
End Sub

Private Sub btnDrug_Click(sender As Object, e As EventArgs) Handles btnDrug.Click
LoadDrugData()
End Sub

Private Sub SellToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
SellToolStripMenuItem.Click
Me.Hide()
Form3.Show()
End Sub

Private Sub UpdateDataToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
UpdateDataToolStripMenuItem.Click
Me.Hide()
Form3.Show()
End Sub

Private Sub LogOutToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
LogOutToolStripMenuItem.Click
Me.Hide()
Form1.Show()
End Sub
```

```
Private Sub BillsToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
BillsToolStripMenuItem.Click
Me.Hide()
Form5.Show()
End Sub

Private Sub DashBoardToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
DashBoardToolStripMenuItem.Click
Me.Hide()
Form4.Show()
End Sub

Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExitToolStripMenuItem.Click
Application.Exit()
End Sub

Private Sub DataGridView1_CellContentClick(sender As Object, e As DataGridViewCellEventArgs)
 Handles DataGridView1.CellContentClick

End Sub
End Class
```