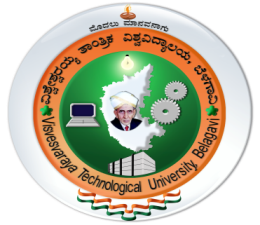
VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 018



A Project Work Report

On

“PHARMA MANAGEMENT SYSTEM”

By

SHREESHA 4MT21CS151

TEJAS JAIN 4MT21CS171

VISHAL SHETTY 4MT21CS183

VISHAL TEGGI 4MT21CS185

SURAJ CD 4MT21CS415



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

Badaga Mijar, Moodabidri-574 225, Karnataka

2022-2023

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

***Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution***

**(A Unit of Rajalaxmi Education Trust)  
Affiliated to Visvesvaraya Technological University, Belagavi**

Badaga Mijar, Moodabidri – 574 225 , Karnataka

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



DECLARATION

We, SHREESHA(4MT21CS151), TEJAS JAIN(4MT21CS171), VISHAL SHETTY(4MT21CS183) , VISHAL TEGGI(4MT21CS185) and **SURAJ CD**(4MT21CS415) students of 4th semester BE in Computer Science & Engineering, Mangalore Institute of Technology and Engineering, Moodabidri, hereby declare that the project work entitled “PHARMA MANAGEMENT SYSTEM”, submitted to the department of Computer Science & Engineering during the academic year **2022-23**. This project work is submitted as assignment to **Design and Analysis of Algorithms(21CS42).**

Date: 14th September,2023 SHREESHA

Place: MITE, Moodabidri TEJAS JAIN

VISHAL SHETTY

VISHAL TEGGI

**SURAJ CD**

**ABSTRACT**

The Pharmacy Management System (PMS) is a vital software application designed for efficient management of pharmaceutical products and services within a pharmacy or pharmaceutical business. It facilitates streamlined inventory management and customer service. PMS encompasses CRUD operations for administrators, allowing them to create, read, update, and delete product details, while users can access specific functionalities. The system's core functionalities include inventory management, pricing accuracy, and user access control. It is instrumental in enhancing pharmaceutical business operations, ensuring customer satisfaction, and maintaining accurate records.

### INTRODUCTION

A Pharmacy Management System (PMS) is a software application designed to streamline the management of pharmaceutical products and services within a pharmacy or pharmaceutical business. This system is crucial for efficiently handling various tasks related to inventory management and serving customers. In the context of a Pharmacy Management System, the key objectives and steps are as follows:

**Objectives of a Pharmacy Management System:**

**Efficiency Improvement:** The system aims to automate various aspects of pharmaceutical inventory management, such as adding, updating, and deleting records. This automation frees up staff time to focus on other critical tasks, like serving customers.

**Accuracy Enhancement:** Ensuring that pharmaceutical product records are accurate and up-to-date is essential for safe and effective customer service. The system helps maintain data accuracy.

**Enhanced Communication:** Improved communication between different departments within the pharmacy, such as the inventory department, sales department, and customer service, is crucial to ensure efficient operations and excellent customer care.

**Security**: Protecting sensitive patient and customer data from unauthorized access is paramount to safeguard privacy and adhere to regulatory requirements.

**Development Steps for a Pharmacy Management System:**

* Gather Requirements
* Design the System
* Implement the System
* Test the System
* Deploy the System

**TECHNOLOGIES USED**

1. C Programming Language: The entire program is written in the C programming language.
2. **stdio.h**: This header file is used for standard input and output operations. It provides functions like printf and scanf for console input and output.
3. **stdlib.h**: The stdlib.h header file is used for various standard library functions, including memory allocation and file manipulation.
4. **string.h**: The string.h header file is used for string manipulation functions like strcmp.
5. **File Handling**: The program uses file handling to read from and write to files. It uses functions like fopen, fclose, fwrite, fread, remove, and rename for file operations.

The program is structured around a simple console-based user interface, where users can log in as either an administrator or a regular user. Administrators have additional options like adding, deleting, and updating medicine stock, as well as viewing the stock. Regular users can check and download invoices or view the available stock.

**SYSTEM ARCHITECTURE**

Main Function (main): The main function serves as the entry point of the program. It contains the user login and menu selection logic, directing users to either the admin or user interface based on their credentials.

**User Login** (login Function): The login function handles user authentication. It checks the provided username and password against predefined values for an admin and a user. Depending on the credentials entered, it returns a code (1 for admin, 2 for user) to indicate the user type.

**Admin Interface** (admin Function): The admin function is the administrative interface. It allows administrators to perform various operations:

**Add Stock** (addStock Function): Administrators can add new pharmaceutical products to the inventory by entering details such as name, quantity, and price. The data is written to a file ("medicine.txt") for storage.

**Delete Stock** (deleteStock Function): Administrators can delete existing pharmaceutical products from the inventory based on the entered medicine name. The record is removed from the file, and the file is either deleted or renamed for updating.

**Alter Price** (alterPrice Function): Administrators can modify the price of an existing medicine by entering the medicine name and the new price. The file is updated with the new price.

**Show Stock** (showStock Function): Administrators can view the current pharmaceutical product inventory. The system reads data from the file and displays it in a tabular format.

**User Interface** (user Function): The user function is the user interface, which allows regular users to perform the following tasks:

**Check and Download Invoice** (checkAndShowInvoice Function): Users can request an invoice for their purchases. The system calculates the total cost based on the selected medicines and quantities, generates an invoice, and displays it.

**Show Stock** (showStock Function): Users can view the available pharmaceutical products in the inventory. The system reads data from the file and displays it for users to browse.

**File Handling**: The program uses file handling to read from and write to a file named "medicine.txt." This file serves as the data storage for pharmaceutical product records.

Data Structure (struct Medicine): The program defines a structure named Medicine to represent pharmaceutical product records. It includes fields for medicine name, quantity, and price.

This system follows a simple console-based architecture, and it is suitable for basic pharmaceutical inventory management tasks. It uses file storage for data persistence, and user interactions are handled through the command-line interface. Depending on the specific requirements and scalability needs, this architecture can be extended to include additional features, data validation, and error handling.

**DESIGN AND IMPLEMENTATION**

**Front-End Design :**

• The front-end design is based on a command-line interface, where users can select

options by entering numbers corresponding to their desired actions.

**Back-End Design :**

• The back-end logic is implemented in the C programming language.

• Each module has its functions for adding, displaying, modifying, and deleting records.

• File handling functions are used to perform operations on data files.

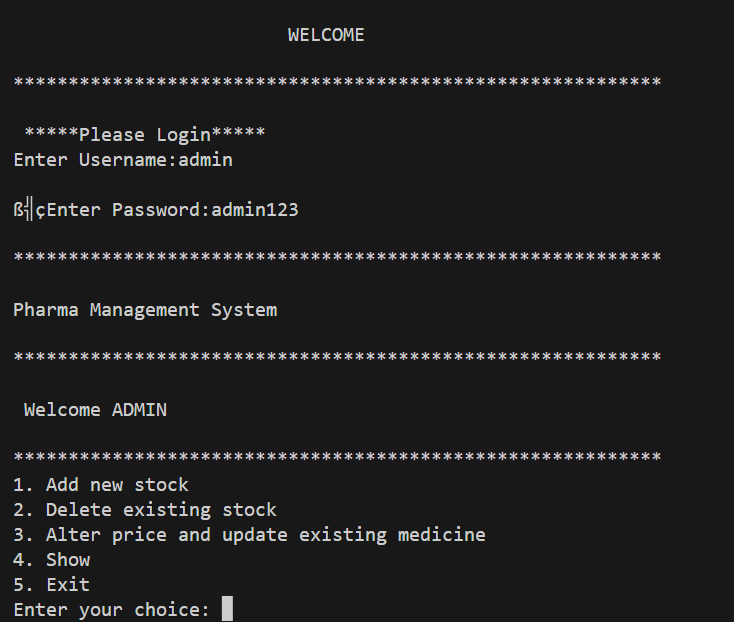
**Database Design :**

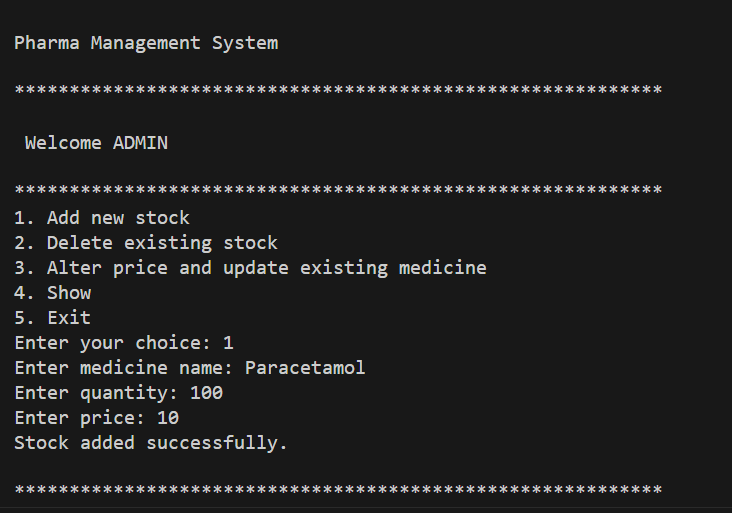
• Data for medicines is stored in separate file.

• Each file contains records in a structured format

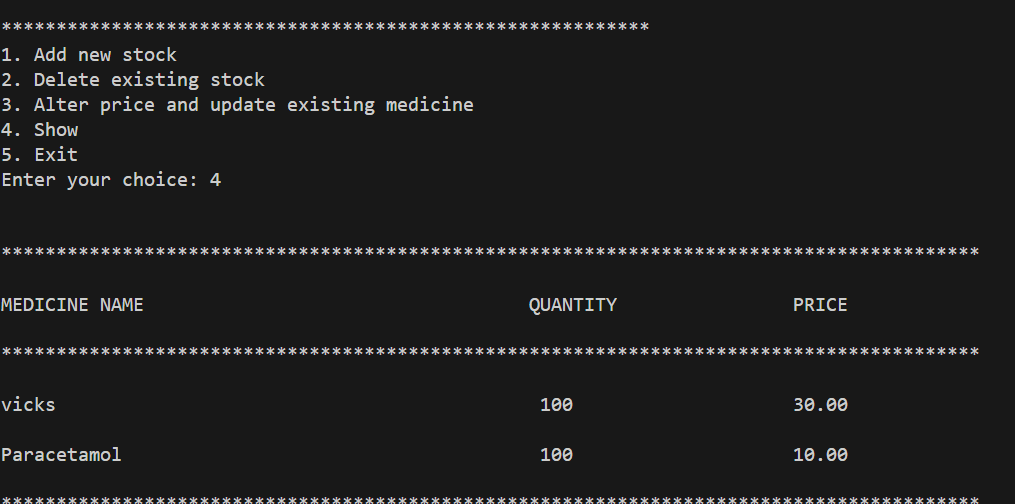
TEST CASES:

1)SUCCESSFUL ADMIN LOGIN

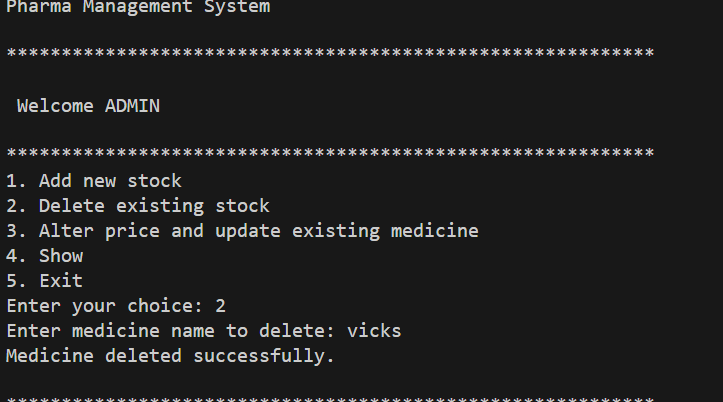


2)ADDITION OF NEW STOCK

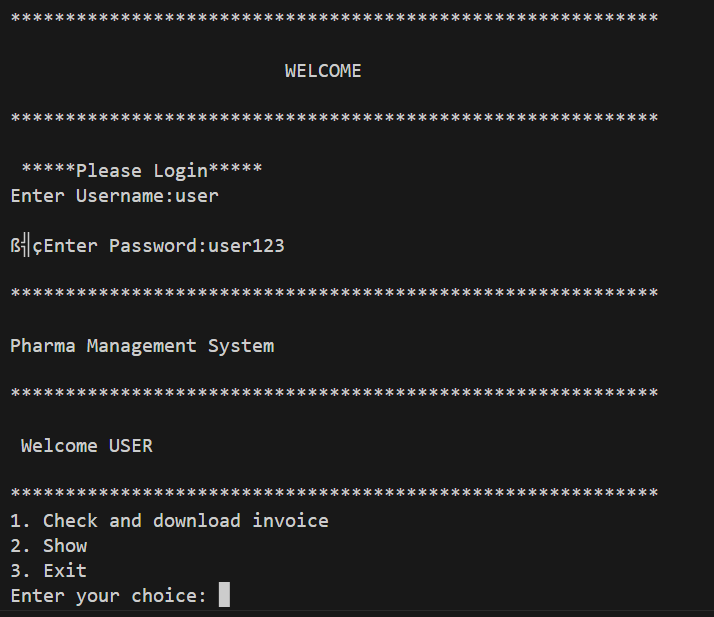
3)ADMIN SHOW STOCK

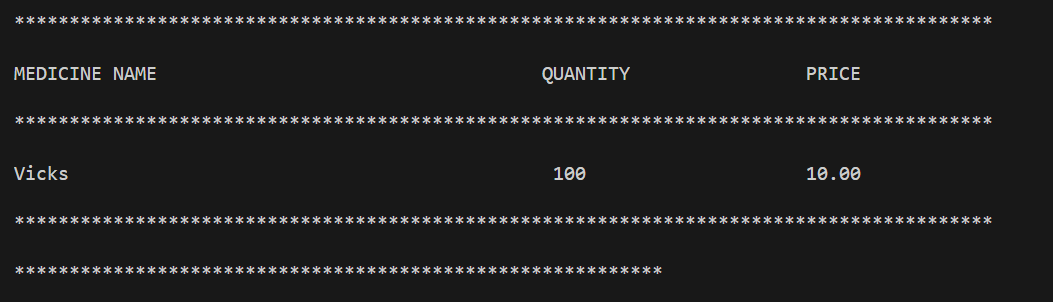


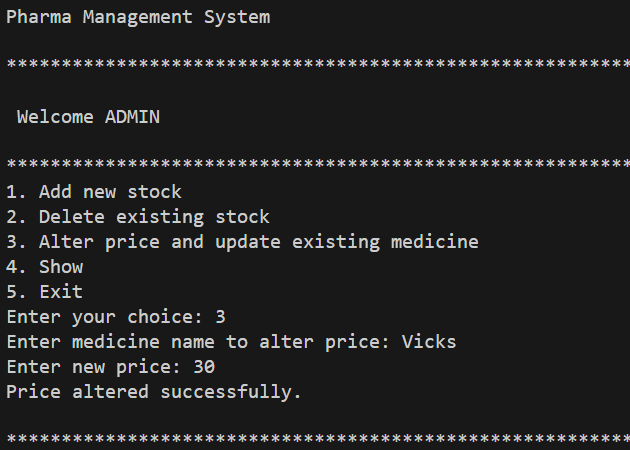
4)SUCCESSFUL DELETION OF STOCK

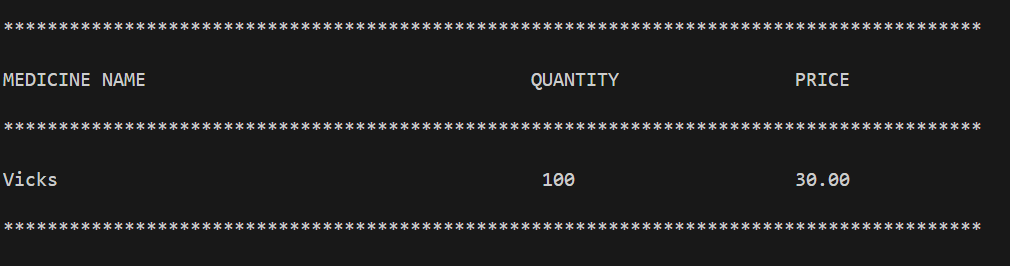


5)SUCCESSFUL USER LOGIN

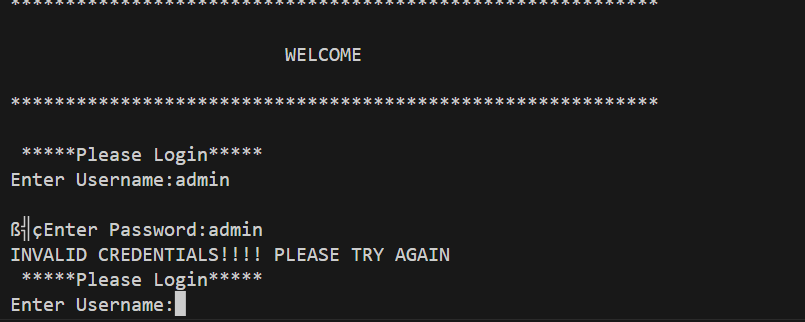


6)VIEW STOCK OPTION FOR USER ONLY

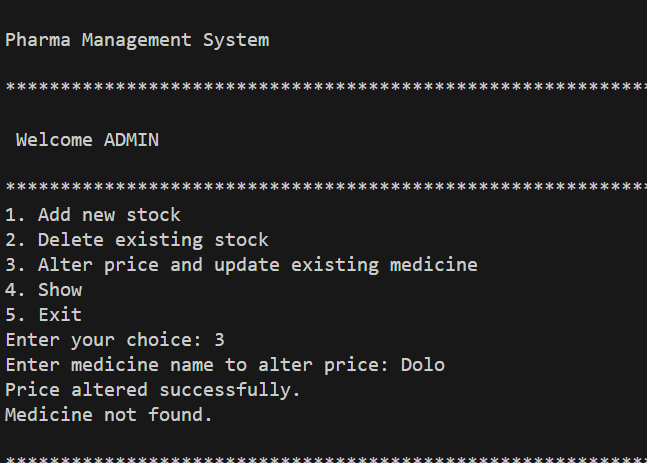
7)ALTER PRICE OPTION FOR ADMIN ONLY

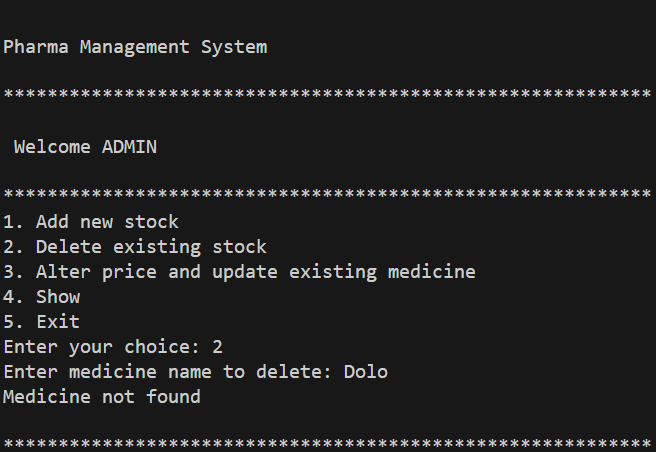


8)INCORRECT LOGIN CREDENTIALS

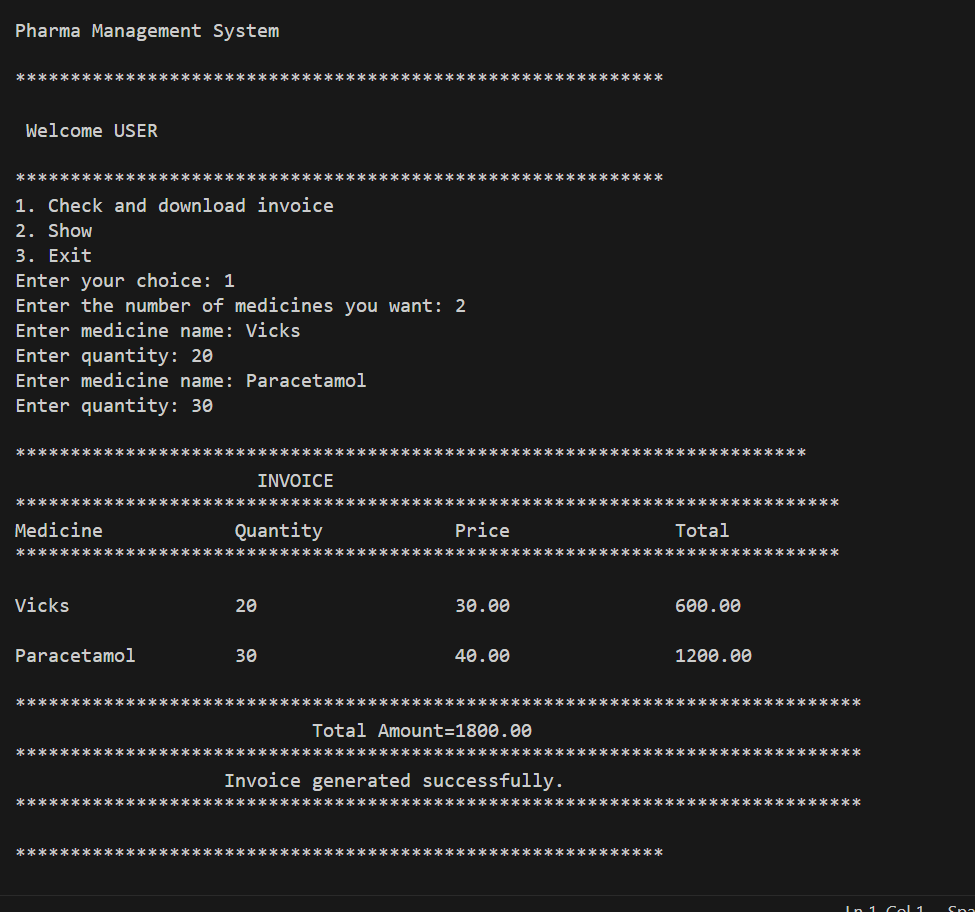


9) DELETING THE MEDICINE WHICH IS NOT PRESENT IN THE STOCK

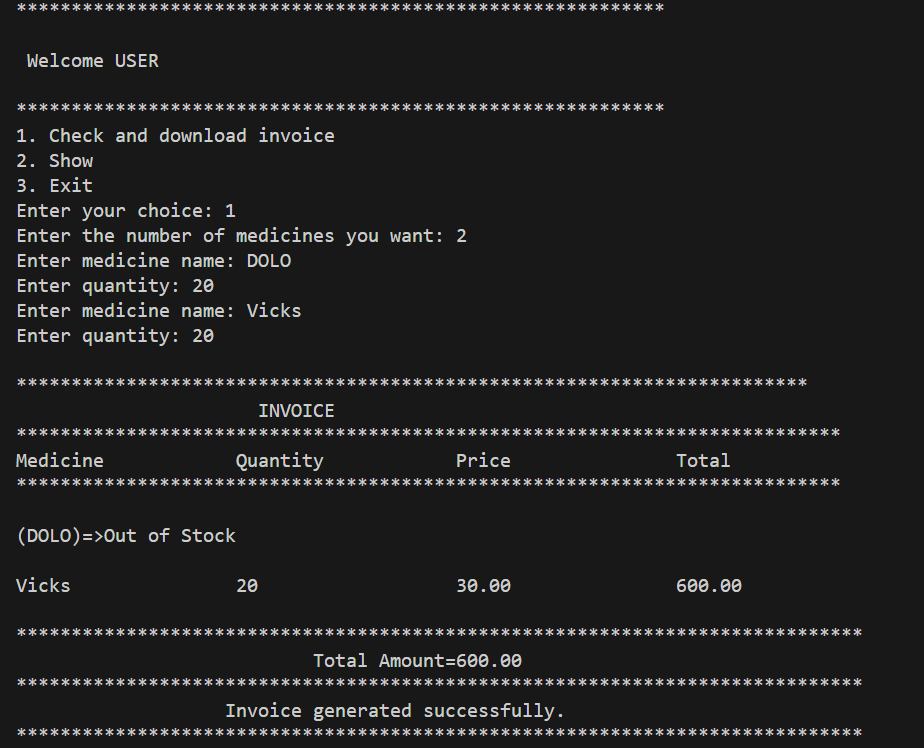


10)ALTER PRICE OF MEDICINE WHICH IS NOT PRESENT IN THE STOCK

11)PLACE ORDER AND DOWNLOAD INVOICE(SUCCESSFUL)



12)PLACE ORDER OF MEDICINE WHICH IS NOT PRESENT IN THE STOCK



**FEATURES AND FUNCTIONALITY**

The Pharmacy Management System is a comprehensive program designed to facilitate the management of pharmaceutical products and services. It offers a range of features and functionalities that enhance the efficiency of pharmacy operations:

1. **Add Stock**: Users can easily add new medicines to the inventory by providing essential information such as the medicine's name, quantity, and price. This feature ensures that the stock remains up-to-date.
2. **Show Stock**: The system allows users to view the entire medicine inventory. A clear and organized presentation displays medicine names, quantities, and prices, making it easy to assess the available stock.
3. **Delete Stock**: Users can remove medicines from the inventory by specifying the name of the medicine they wish to delete. This feature helps maintain an accurate record of available medicines.
4. **Alter Price:** Pharmacy staff can modify the price of a specific medicine as needed. The system prompts users to input the medicine's name and the new price, ensuring price adjustments are straightforward.
5. **Search and Generate Invoices:** The system enables users to search for specific medicines and generate invoices. Users can input the medicine's name and quantity needed, and the system calculates the total cost. If the requested quantity is unavailable, it alerts the user.
6. **User Login**: The system has a secure login mechanism with roles for administrators and regular users. It ensures that only authorized personnel can access and manipulate the data.
7. **User-Friendly Interface:** The system boasts a user-friendly interface, making it accessible to individuals with varying levels of technical expertise. Clear instructions and a help system guide users through tasks.
8. **Continuous Data Integration**: Continuous integration ensures that the system is developed correctly and can be easily maintained. It promotes code reliability and minimizes errors during development.

**TESTING**

**1. Unit Testing:**

- Unit testing involves testing individual functions and methods in isolation to ensure they perform as expected.

- For this Pharmacy Management System, unit tests can be created for each function separately.

- For example, you can test the "addStock" function by providing various inputs and checking if the stock is added correctly.

**2. Integration Testing:**

- Integration testing verifies that different parts of the system work together seamlessly.

- In this system, integration testing could focus on interactions between functions. For instance, ensuring that adding stock and displaying stock work well together.

- You would test scenarios like adding stock and then displaying it to confirm that the data flow and interactions are correct.

**3. User Acceptance Testing (UAT)**

- UAT involves testing the system from the end-user's perspective to ensure it meets their requirements and expectations.

- In the Pharmacy Management System, UAT would involve real users (administrators and regular users) using the system for its intended purposes.

- Users would perform tasks such as adding stock, generating invoices, and managing the inventory. The system should be intuitive and fulfill their needs.

- Feedback from users during UAT should be collected and incorporated to improve the system.

Testing is a critical phase in software development to ensure the system is robust, reliable, and meets user expectations. It helps identify and rectify issues early in the development process, resulting in a more dependable and user-friendly application.

**CHALLENGES AND TESTING**

Following are some of the challenges, obstacles, and roadblocks encountered during the development of the Pharma Management System project:

->**Data validation**: One of the challenges was ensuring that the data entered by the user was valid. This was done by using input validation techniques to check for invalid username and password.

->**File handling:** Another challenge was handling files. This involved opening, reading, writing, and closing files.

->**Error handling**: Error handling was also a challenge. This involved handling errors that occurred during data validation, file handling, and other operation

**CONCLUSION**

The Pharmacy Management System project was a resounding success, developed with a modular approach and continuous integration, ensuring correctness and easy maintenance.

This system empowers efficient pharmaceutical product management, allowing users to add, view, search, update, and remove products. Its user-friendly interface, supported by clear instructions and a helpful guide, accommodates users of all technical levels.

Efficiency and reliability are the system's hallmarks, streamlining pharmaceutical product tasks for accuracy and speed. Its versatility shines through its comprehensive features, making it a potent tool.

The Pharmacy Management System is not only user-friendly but also efficient and reliable, streamlining inventory management with precision. Its versatility makes it a powerful asset.

The system's future is bright, suitable for various pharmacy settings and expandable to include appointment management, patient care tracking, and billing reports.

In summary, the Pharmacy Management System project exemplifies successful software development, offering user-friendliness, efficiency, and versatility for pharmaceutical product management. With potential for growth, it promises to impact the pharmaceutical industry significantly.

**REFERENCES**

🡪 <https://practice.geeksforgeeks.org>

🡪 <https://chat.openai.com>

🡪 <https://stackoverflow.com/>

🡪<https://miro.com/>

🡪<https://draw.io/>