

STOCK MANAGEMENT SYSTEM MINOR PROJECT REPORT

A MINI PROJECT REPORT

Submitted by

RA2211026010016 – Karan Sood

RA2211026010027 – Aditya Nair

Under the guidance of

Dr. Pitchaimanickam B

Assistant Professor, Department of Computing Intelligence

In partial fulfilment for the Course

of

21CSC201J – DATA STRUCTURES AND ALGORITHMS

in

DEPARTMENT OF COMPUTING INTELLIGENCE



FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC201J – DATA STRUCTURES AND ALGORITHMS** entitled in "**STOCK MANAGEMENT SYSTEM**" is the bonafide work of **KARAN SOOD (RA2211026010016) , ADITYA NAIR (RA2211026010027) and** who carried out the work under my supervision.

SIGNATURE

Dr. Pitchaimanickam B
Assistant Professor,
Department of Computing Intelligence
SRM Institute of Science and Technology
Kattankulathur

SIGNATURE

Dr.R.Annie Uthra
Professor & Head,
Department of Computing Intelligence
SRM Institute of Science and Technology
Kattankulathur

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems and Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Dr. Pitchaimanickam B Assistant Professor, Computing Intelligence** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD, Dr.R.Annie Uthra & Head, Computing Intelligence** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

ABSTRACT

A stock management system is a software solution designed to efficiently oversee and control the flow of inventory within a business or organization. Its primary goal is to optimize inventory levels, minimize stockouts, reduce excess stock, and streamline the overall inventory management process.

Key components of a stock management system typically include:

1. ****Inventory Tracking****: Monitoring and recording the quantity, location, and status of all items in stock. This often involves barcode scanning, RFID technology, or manual input to update inventory levels.
2. ****Inventory Analysis****: Utilizing historical data and analytics to forecast demand, identify trends, and make informed decisions about stock replenishment, purchasing, and stocking levels.
3. ****Order Management****: Automating the process of generating purchase orders, managing supplier relationships, and ensuring timely restocking of inventory to prevent shortages.
4. ****Warehouse Management****: Optimizing the physical storage of inventory within warehouses or distribution centers to maximize space utilization and facilitate efficient order fulfillment.
5. ****Real-time Updates****: Providing up-to-date information on inventory levels, sales, and stock movement to enable quick decision-making and prevent overstocking or stockouts.
6. ****Reporting and Insights****: Generating reports and analytics that offer insights into inventory turnover, stock aging, profitability, and other key performance indicators to improve overall inventory management strategies.

Implementing a robust stock management system can result in various benefits for businesses, including improved inventory accuracy, reduced carrying costs, increased efficiency in order fulfillment, enhanced customer satisfaction, and better financial management through optimized stock levels.

The specifics of a stock management system can vary based on the size and nature of the business, industry requirements, and the chosen software or technology solution.

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
1	INTRODUCTION	
	1.1 Motivation	10
	1.2 Objective	10
	1.3 Problem Statement	11
	1.4 Challenges	11
2	LITERATURE SURVEY	12
3	REQUIREMENT ANALYSIS	13
4	ARCHITECTURE & DESIGN	14
5	IMPLEMENTATION	15
6	EXPERIMENT RESULTS & ANALYSIS	21
7	CONCLUSION	24
8	REFERENCES	25

1. INTRODUCTION

1.1 The motivation

Stock management systems are motivated by several key factors, all of which contribute to the efficiency, profitability, and sustainability of a business:

1. ****Optimized Inventory Levels****: Effective stock management systems help maintain optimal inventory levels. By avoiding excess stock and preventing stockouts, businesses can minimize carrying costs while ensuring they have the right products available to meet customer demand.
2. ****Improved Efficiency and Productivity****: Automation and streamlining of inventory processes lead to increased efficiency. By automating tasks such as reordering, tracking, and managing stock, businesses can save time and resources, allowing employees to focus on more value-added activities.
3. ****Enhanced Customer Satisfaction****: Having the right products available when customers want them is crucial for customer satisfaction. A well-managed stock system ensures timely order fulfillment, reducing the likelihood of backorders or long waiting periods, which in turn leads to happier customers.
4. ****Cost Reduction****: Overstocking ties up capital and incurs storage costs, while stockouts can lead to missed sales opportunities and dissatisfied customers. Effective stock management helps reduce these costs by optimizing inventory levels and preventing unnecessary holding or emergency ordering expenses.

1.2 PROBLEM STATEMENTS

The problem statement for a stock management system typically outlines the challenges or inefficiencies present in the current inventory control processes that necessitate the implementation of a more effective system. Here is an example problem statement:

"Inefficient inventory management practices lead to recurring issues such as inaccurate stock levels, frequent stockouts, overstocking of certain items, and a lack of real-time visibility into inventory movement. The current manual tracking system is prone to errors, resulting in discrepancies between recorded and actual stock quantities. This leads to increased carrying

costs, operational inefficiencies, decreased customer satisfaction due to delayed or unfulfilled orders, and missed sales opportunities. Lack of forecasting and data-driven insights further compounds the problem, leading to suboptimal inventory replenishment and procurement decisions. To address these challenges and improve overall operational efficiency, there is a critical need for an advanced stock management system that offers real-time tracking, accurate inventory data, robust reporting and analysis capabilities, and seamless integration across the supply chain.

CHALLENGES

Stock management systems come with several challenges, particularly in balancing inventory levels, operational efficiency, and customer satisfaction. Some of these challenges include:

- 1. ****Inventory Accuracy****:** Maintaining accurate inventory counts across multiple locations or warehouses can be difficult. Discrepancies due to theft, errors in data entry, or technical issues can lead to inefficiencies and incorrect decision-making.
- 2. ****Demand Forecasting****:** Predicting future demand accurately is challenging. Fluctuations in customer behavior, market trends, and seasonality can make it difficult to forecast demand effectively, leading to overstocking or stockouts.
- 3. ****Supplier Management****:** Coordinating with suppliers to ensure timely deliveries and manage lead times can be a challenge. Delays in receiving stock can disrupt the supply chain and impact customer satisfaction.
- 4. ****Cost Management****:** Balancing the cost of carrying inventory against the risk of stockouts is crucial. Carrying excess inventory ties up capital and incurs holding costs, while insufficient stock can lead to lost sales and dissatisfied customers.
- 5. ****Integration and Technology****:** Implementing and integrating stock management systems with existing software and hardware can be complex. Compatibility issues, data migration challenges, and training employees on new systems can pose obstacles.
- 6. ****Seasonal and Trend Variations****:** Industries prone to seasonal fluctuations or those heavily influenced by trends face the challenge of managing inventory levels to meet **varying**

demand without excess stock remaining after the peak period.

7. ****Stock Visibility****: Maintaining real-time visibility across all inventory locations can be difficult, especially in multi-channel retail environments or businesses with disparate systems, leading to discrepancies and inefficient stock allocation.

8. ****Supply Chain Disruptions****: External factors such as natural disasters, geopolitical events, or supplier issues can disrupt the supply chain, causing delays in inventory replenishment and affecting stock availability.

Addressing these challenges often involves a combination of improved data accuracy, leveraging technology for better forecasting, optimizing supply chain processes, employing inventory management best practices, and fostering better collaboration among various stakeholders involved in the inventory management process.

REQUIREMENTS

The requirements for a stock management system typically revolve around various functionalities and features needed to effectively manage inventory within a business. Here's an outline of essential requirements:

1. **Inventory Tracking and Monitoring:**

- Barcode/QR Code Scanning or RFID Technology: To track items entering and leaving the inventory.
- Real-time Updates: Continuous monitoring and updating of inventory levels to reflect accurate stock counts.

2. **User-Friendly Interface:**

- Intuitive and easy-to-use interface for employees to input data, manage inventory, and generate reports.

3. **Inventory Classification and Organization:**

- Categorization and tagging of items for easy identification and retrieval.

- Grouping items based on various criteria (e.g., SKU, category, location) for efficient management.

4. **Stock Replenishment and Purchase Order Management:**

- Automated stock reordering based on preset thresholds or demand forecasting.
- Generation and management of purchase orders for restocking inventory from suppliers.

5. **Integration and Compatibility:**

- Compatibility with existing systems (e.g., accounting software, ERPs) for seamless data exchange.
- Integration with sales systems to synchronize inventory levels with sales data.

6. **Reporting and Analytics:**

- Generation of detailed reports on inventory turnover, stock levels, trends, and performance metrics.
- Analytics capabilities to forecast demand, identify slow-moving items, and optimize stock levels.

7. **Security and Access Control:**

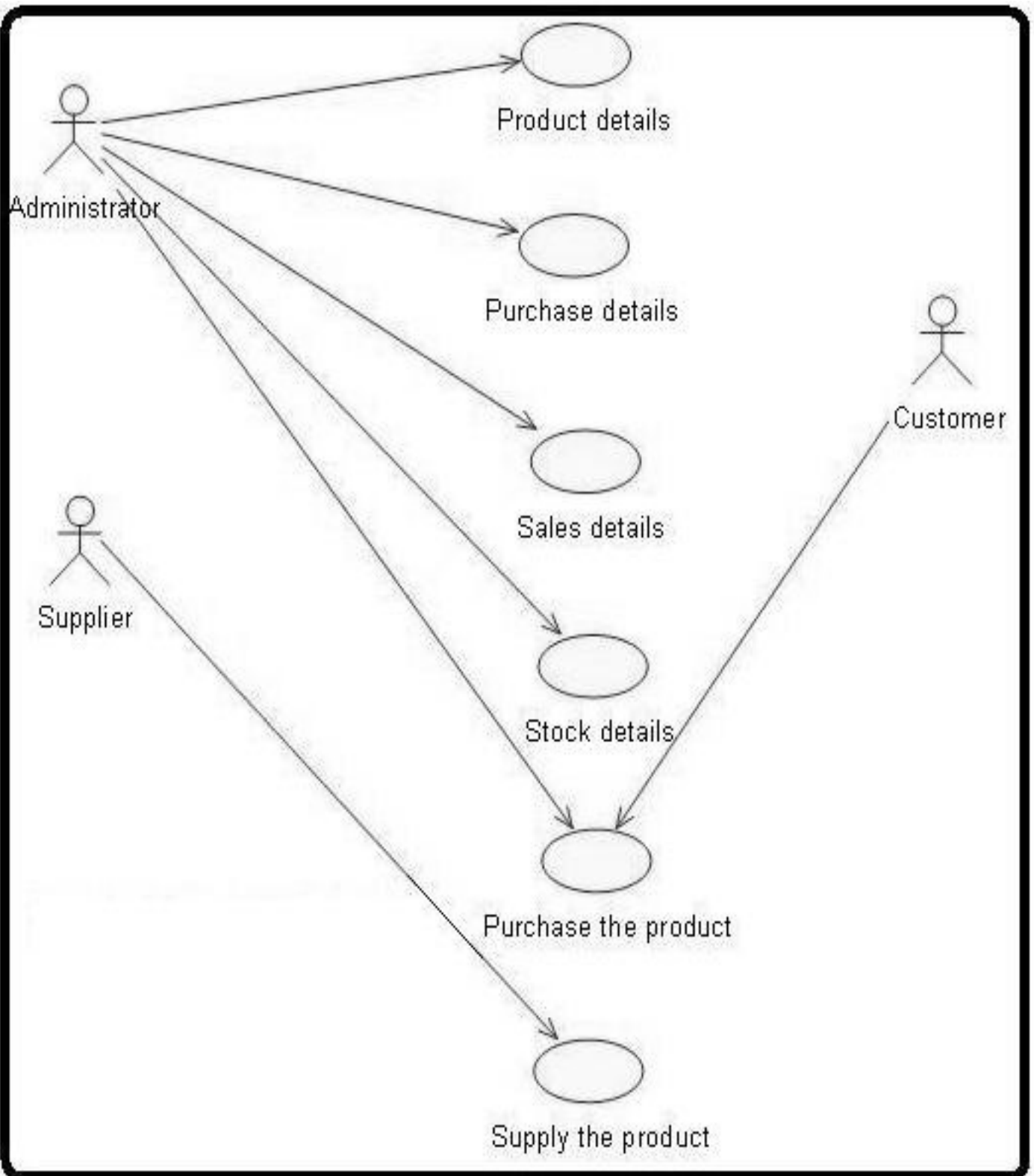
- User authentication and authorization protocols to restrict access based on roles and responsibilities.
- Data encryption and regular backups to ensure the security and integrity of inventory information.

8. **Mobile Accessibility:**

- Mobile app or responsive design for accessibility on various devices to enable on-the-go inventory management.

These requirements serve as a foundation for developing or selecting a stock management system that aligns with the specific needs and goals of a business, enabling efficient control and optimization of inventory operations.

4.ARCHITECTURE AND DESIGN



5.IMPLEMENTATION :

5.1 C IMPLEMENTATION

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
#include<windows.h>
#include<time.h>
#define ENTER 13
#define BKSP 8h
#define SPACE 32
#define TAB 9
void setcolor(int ForgC)
{ WORD wColor;
HANDLE hStdOut=GetStdHandle(STD_OUTPUT_HANDLE);
CONSOLE_SCREEN_BUFFER_INFO csbi;
if(GetConsoleScreenBufferInfo(hStdOut,&csbi))
{
    wColor=(csbi.wAttributes & 0xF0)+(ForgC & 0x0F);
    SetConsoleTextAttribute(hStdOut,wColor);
}
}
struct item
{
    char productname[40],productcomp[40],c;
    int productid;
    int price;
    int Qnt;
}st;
void wel_come(void);
void title(void);
void login();
void menu(void);
void title(void);
void deleteproduct(void);
void gotoxy(short x, short y)
{
    COORD pos ={x,y};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),pos);
```

[illegible]

```

int a=0,i=0;
char uname[10],c=' ';
char pword[10],code[10];
char user[10]="user";
char pass[10]="pass";
do
{
    printf("\n ===== LOGIN ===== ");
    printf(" \n          USERNAME:-");
    scanf("%s", &uname);
    printf(" \n          PASSWORD:-");
    while(i<10)
    {
        pword[i]=getch();
        c=pword[i];
        if(c==13) break;
        else printf("*");
        i++;
    }
    pword[i]='\0';
    i=0;
    if(strcmp(uname,"user")==0 && strcmp(pword,"pass")==0)
    {
        printf(" \n\n\n          WELCOME TO PRODUCT MANAGEMENT SYSTEM !!!! LOGIN IS
SUCCESSFUL");
        printf("\n\n\n\t\t\t\t\tPress any key to continue...");
        getch();//holds the screen
        break;
    }
    else
    {
        printf("\n          SORRY !!!! LOGIN IS UNSUCCESSFUL");
        a++;
        getch();//holds the screen
    }
}

while(a<=2);
if (a>2)
{
    printf("\nSorry you have entered the wrong username and password for four
times!!!");
    getch();
}
system("cls");
menu();
}

void menu(void)
{
    int choice;
    system("cls");

```

```

main:
printf("\n===== Product Management System
=====");
printf("
");
printf("
");
printf("\n\t\tPress <1> Add Products");
printf("\n\t\tPress <2> Delete Products");
printf("\n\t\tPress <3> Search Products");
printf("\n\t\tPress <4> Read Items");
printf("\n\t\tPress <5> Edit Items");
printf("\n\t\tPress <6> Exit!");
printf("\n\n\t\tEnter your choice[1-6]");
scanf("%i", &choice);
system("cls");
switch(choice)
{
    case 1:
        add_item();
        break;
    case 2:
        deleteproduct();
        break;
    case 3:
        search_item();
        break;
    case 4:
        read_item();
        break;
    case 5:
        edit_item();
        break;
    case 6:
        printf("System Exit");
        exit(0);
        break;
    default:
        printf("Invalid Choice! System Exit\n");
        getch();
}
}
void add_item()
{
    int index, valid;
    char c;
    int a=0;
    FILE *fp;
    do
    {

```

```

system("cls");
printf("===== Enter Product Detail =====");
int ID;
fp = fopen("NextFile.dat","a");//opening file and creating a staff.txt
file to append or write

if((fp = fopen("NextFile.dat","a"))!=NULL)//if condition to check file
is NULL or not
{
    I:
    printf("\nProduct Code\t :");
    scanf("%i",&ID);
    while(fscanf(fp,"%s %s %i %i %i", st.productname, st.productcomp,
&st.price, &st.productid,&st.Qnt)!=EOF)
    {
        if(ID == st.productid)
        {
            printf("\n\tTHE PRODUCT CODE ALREADY EXIST.\n");
            goto I;
        }
    }
    st.productid = ID;
}
else//runs if sfile is empty
{
    printf("\nProduct Code\t :");
    scanf("%i",&st.productid);
}
do
{
    fflush(stdin);
    printf("\nProduct Name\t :");
    gets(st.productname); // get input string
    st.productname[0]=toupper(st.productname[0]);
    for (index=0; index<strlen(st.productname); ++index)
    { //check if character is valid or not
        if(isalpha(st.productname[index]))
            valid = 1;
        else
        {
            valid = 0;
            break;
        }
    }
    if (!valid)
    {
        printf("\n Name contain invalid character. Please 'Enter'
again");
        getch();
    }
}

```

```

    }while(!valid);
    do
    {
        char productcomp[40];
        fflush(stdin);
        printf("\nProduct Company\t :");
        gets(st.productcomp); // get input string
        st.productcomp[0]=toupper(st.productcomp[0]);
        for (index=0; index<strlen(st.productcomp); ++index)
        { //check if character is valid or not
            if(isalpha(st.productcomp[index]))
                valid = 1;
            else
            {
                valid = 0;
                break;
            }
        }
        if (!valid)
        {
            printf("\n Name contain invalid character. Please 'Enter'
again");
            getch();
        }
    }while(!valid);
    do
    {
        printf("\nPrice [10-5000]Rupees:");
        scanf("%i",&st.price);
        if(st.price<10 || st.price>5000)
        {
            printf("\n\tYou Cannot Enter the price limit [10-5000].Re-
Enter.");
        }
    }while(st.price<10 || st.price>5000);

    do
    {
        printf("\nQuantity [1-500]\t:");
        scanf("%i",&st.Qnt);
        if(st.Qnt<1 || st.Qnt>500)
        {
            printf("\n\tEnter Quantity[1-500] only.Re-Enter.");
        }
    }while(st.Qnt<1 || st.Qnt>500);
    fp=fopen("NextFile.dat","a");
    fprintf(fp,"\n%s %s %i %i %i", st.productname, st.productcomp,st.price,
st.productid,st.Qnt);
    fclose(fp);

```



```

        printf("\nPress 'Enter' to add more item and any other key to go to main
menu");

    }
    while((c = getch()) == '\r');
    menu();
}
void search_item()
{
    char target[40];
    int found=0;
    FILE *sfile;
    sfile=fopen("NextFile.dat","r");
    printf("\nEnter name to search:");
    fflush(stdin);
    gets(target);
    target[0]=toupper(target[0]);
    while (!feof(sfile) && found==0)
    {
        fscanf(sfile,"%s %s %i %i %i", st.productname, st.productcomp, &st.price,
&st.productid,&st.Qnt);
        if(strcmp(target, st.productname)==0)
        {
            found=1;
        }
    }
    if(found)
    {
        printf("\n Record found");
        printf("\nProduct Name\t\t:%s \nProduct Company\t\t:%s \nProduct
Price\t\t:%i \nProduct ID\t\t:%i \nProduct Quantity\t\t:%i", st.productname,
st.productcomp, st.price, st.productid, st.Qnt);
    }
    else
    {
        printf("Noo Record found");
        fclose(sfile);
        printf("\nPress any key to go to Main Menu!");
        while((st.c = getch()) == '\r');
        menu();
    }
}
void deleteproduct(void)
{
    char target[40];
    int found=0;
    FILE *sfile, *tfile;
    sfile=fopen("NextFile.dat","r");
    tfile=fopen("TempFile.dat","w+");
    printf("\n Enter name to Delete: ");
    fflush(stdin);
    scanf("%s",target);

```

```

    target[0]=toupper(target[0]);
    while (fscanf(sfile,"%s %s %i %i %i\n",st.productname,st.productcomp,
&st.price,&st.productid,&st.Qnt)!=EOF)
    {
        if(strcmp(target,st.productname)==0)
        {
            found=1;
        }
        else
        {
            fprintf(tfile,"%s %s %i %i %i\n", st.productname,st.productcomp,
st.price,st.productid,st.Qnt);
        }
    }

    if(!found)
    {
        printf("\n Record not Found");
        getch();
        menu();
    }
    else
    {
        printf("\n Record deleted");
    }
    fclose(sfile);
    fclose(tfile);
    remove("NextFile.dat");
    rename("TempFile.dat","NextFile.dat");
    printf("\nPress any key to go to Main Menu!");
    while((st.c = getch()) !='\r');
    menu();
}

void read_item()
{
    FILE *f;
    int i, q;
    if((f=fopen("NextFile.dat","r"))==NULL)
    {
        gotoxy(10,3);
        printf("NO RECORDS");
        printf("\n\t\tPress any key to go back to Menu.");
        getch();
        menu();
    }
    else
    {
        gotoxy(0,5);
        for(i=0;i<100;i++)
        {
            printf("-");

```

```

    }
    gotoxy(5,6);
    printf("Product Name");
    gotoxy(25,6);
    printf("Product Price");
    gotoxy(40,6);
    printf("Product Company");
    gotoxy(60,6);
    printf("Product CODE");
    gotoxy(80,6);
    printf("Product Quantity\n");
    for(i=0;i<100;i++)
    {
        printf("-");
    }
    q=8;
    while(fscanf(f,"%s %s %i %i %i\n", st.productname,st.productcomp,
&st.price, &st.productid,&st.Qnt)!=EOF)
    {
        gotoxy(5,q);
        printf("%s",st.productname);
        gotoxy(25,q);
        printf("%i",st.price);
        gotoxy(40,q);
        printf("%s",st.productcomp);
        gotoxy(60,q);
        printf("%i",st.productid);
        gotoxy(80,q);
        printf("%i",st.Qnt);
        q++;
    }
    printf("\n");
    for(i=0;i<100;i++)
        printf("-");
}
fclose(f);
printf("\nPress any key to go to Main Menu!");
getch();
menu();
}

void edit_item()
{
    int index, valid;
    char target[40];
    FILE *fp, *rp;
    int a=0;
    int id;
    char edit;
    long int size=sizeof(st);

```

```

if((fp=fopen("NextFile.dat","r+"))==NULL)
{
    printf("NO RECORD ADDED.");
    menu();
}
else
{
    rp = fopen("TempFile.dat","a");
    system("cls");
    printf("Enter Product Code for edit:");
    scanf("%i",&id);
    fflush(stdin);
    while(fscanf(fp,"%s %s %i %i %i\n", st.productname,st.productcomp,
&st.price, &st.productid,&st.Qnt)!=EOF)
    {
        if(id==st.productid)
        {
            a=1;
            printf("\n\t***** Record Found *****");
            printf("\nProduct Name\t\t: %s",st.productname);
            printf("\nProduct Company\t\t: %s",st.productcomp);
            printf("\nPrice\t\t\t: %i",st.price);
            printf("\nProduct Code\t\t: %i",st.productid);
            printf("\nProduct Quantity\t: %i",st.Qnt);
            printf("\n\n\t*** New Record ***");

            do
            {
                fflush(stdin);
                printf("\nNew Product Name\t\t: ");
                gets(st.productname); // get input string
                st.productname[0]=toupper(st.productname[0]);
                for (index=0; index<strlen(st.productname); ++index)
                { //check if character is valid or not
                    if(isalpha(st.productname[index]))
                        valid = 1;
                    else
                    {
                        valid = 0;
                        break;
                    }
                }
                if (!valid)
                {
                    printf("\n Name contain invalid character. Please 'Enter'
again");

                    getch();
                }
            }while(!valid);
            do
            {

```

```

        char productcomp[40];
        fflush(stdin);
        printf("\nNew Product Company\t\t:");
        gets(st.productcomp); // get input string
        st.productcomp[0]=toupper(st.productcomp[0]);
        for (index=0; index<strlen(st.productcomp); ++index)
        { //check if character is valid or not
            if(isalpha(st.productcomp[index]))
                valid = 1;
            else
            {
                valid = 0;
                break;
            }
        }
        if (!valid)
        {
            printf("\n Name contain invalid character. Please 'Enter'
again");
            getch();
        }
    }while(!valid);
    do
    {
        printf("\nNew Price [10-5000]Rupees:");
        scanf("%i",&st.price);
        if(st.price<10 || st.price>5000)
        {
            printf("\n\tYou Cannot Enter the price limit [10-5000].Re-
Enter.");
        }
    }while(st.price<10 || st.price>5000);

    printf("\nEnter New Product Code\t\t:");
    scanf("%i",&st.productid);
    do
    {
        printf("\nNew Quantity [1-500]\t:");
        scanf("%i",&st.Qnt);
        if(st.Qnt<1 || st.Qnt>500)
        {
            printf("\n\tEnter New Quantity[1-500] only.Re-Enter.");
        }
    }while(st.Qnt<1 || st.Qnt>500);
    printf("Press 'y' to edit the existing record or any key to
cancel...");
    edit=getche();
    if(edit=='y' || edit=='Y')
    {

```

```

        fprintf(rp,"%s %s %i %i %i\n", st.productname,
st.productcomp, st.price, st.productid,st.Qnt);
        fflush(stdin);
        printf("\n\n\t\tYOUR RECORD IS SUCCESSFULLY EDITED!!!");
    }
}
else
{
    fprintf(rp,"%s %s %i %i %i\n", st.productname, st.productcomp,
st.price, st.productid,st.Qnt);
    fflush(stdin);
}
}
if(!a)
{
    printf("\n\nTHIS PRODUCT DOESN'T EXIST!!!");
}
fclose(rp);
fclose(fp);
remove("NextFile.dat");
rename("TempFile.dat","NextFile.dat");
getch();
}
menu();
}

```

6.RESULTS AND DISCUSSION

LOGIN MODULE

```
=====
Tue Nov 14 11:38:15 2023
=====
|                                     |
| WELCOME TO                         |
| PRODUCT MGMT SYSTEM               |
|                                     |
| Address                            |
| Number                            |
| "WE BELIEVE IN QUALITY"          |
|                                     |
|                                     |
| MENTOR:KARAN SOOD,ADITYA NAIR....|
| Press any key to continue.....  |
|                                     |
=====
```

LOGIN MODULE

```
===== LOGIN =====
USERNAME:-user

PASSWORD:-****

WELCOME TO PRODUCT MANAGEMENT SYSTEM !!!! LOGIN IS SUCCESSFUL

Press any key to continue...█
```

MENU BAR

```
===== Product Management System =====

Press <1> Add Products
Press <2> Delete Products
Press <3> Search Products
Press <4> Read Items
Press <5> Edit Items
Press <6> Exit!

Enter your choice[1-6]█
```

ADD PRODUCT

===== Enter Product Detail =====

Product Code :1234567

Product Name :mobilecharger

Product Company :apple

Price [10-5000]Rupees:1800

Quantity [1-500] :15

Press 'Enter' to add more item and any other key to go to main menu

SEARCH PRODUCT

Enter name to search:mobilecharger

Record found

Product Name :Mobilecharger

Product Company :Apple

Product Price :1800

Product ID :1234567

Product Quantity :15

Press any key to go to Main Menu!

READ ITEM

Product Name	Product Price	Product Company	Product CODE	Product Quantity
Karan	2056	Adhukaf	102354	2
Mobilecharger	1800	Apple	1234567	15

Press any key to go to Main Menu!

7.CONCLUSION

In conclusion, a well-implemented stock management system stands as a fundamental asset for businesses seeking efficient control and optimization of their inventory. By integrating advanced functionalities and addressing specific operational needs, this system becomes pivotal in streamlining processes, enhancing accuracy, and boosting overall productivity.

The core benefits of a stock management system include:

- 1. **Optimized Inventory Control:** Real-time tracking and monitoring facilitate accurate inventory counts, reducing instances of stockouts or overstocking.**
- 2. **Improved Operational Efficiency:** Automation of tasks like stock replenishment and purchase order generation streamlines workflows, saving time and resources.**
- 3. **Enhanced Decision-Making:** Data-driven insights and analytics empower businesses to make informed decisions regarding inventory levels, purchasing, and overall strategy.**
- 4. **Enhanced Customer Satisfaction:** Maintaining optimal stock levels ensures timely order fulfillment, leading to satisfied customers and increased loyalty.**
- 5. **Cost Reduction:** Efficient inventory management helps minimize carrying costs associated with excess stock and mitigates losses due to stockouts.**
- 6. **Adaptability and Scalability:** The flexibility of a robust stock management system allows for adaptation to changing business needs and future growth.**

However, success in implementing such a system hinges on aligning it with the specific requirements and workflows of the business. Moreover, ongoing maintenance, training, and updates are crucial to ensuring the system continues to meet evolving demands and technological advancements.

Ultimately, a well-designed stock management system is not just a tool; it's a strategic investment that can significantly impact a business's profitability, customer satisfaction, and long-term success in a competitive marketplace.

8.REFERENCES

8.1 Geeks For Geeks

<https://www.geeksforgeeks.org>

8.2 ChatGPT

<https://chat.openai.com/>

8.3 Coding Ninjas

[https://www.codingninjas.com/landing/cnsat/?utm_source=google&utm_medium=\[search\]&utm_campaign=20131922872_149942135238_e_coding%20ninjas__658412799015_m____9300909&gad_source=1&gclid=Cj0KCQjw4vKpBhCZARIsAOKHoWSpFlBO_5rgdFtxoMTToX9WlF_bG-QwVxjyCt8PqUq_076jV0dUOnY0aAtEEEALw_wcB](https://www.codingninjas.com/landing/cnsat/?utm_source=google&utm_medium=[search]&utm_campaign=20131922872_149942135238_e_coding%20ninjas__658412799015_m____9300909&gad_source=1&gclid=Cj0KCQjw4vKpBhCZARIsAOKHoWSpFlBO_5rgdFtxoMTToX9WlF_bG-QwVxjyCt8PqUq_076jV0dUOnY0aAtEEEALw_wcB)

8.4 Quora

<https://www.quora.com>