NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC
Accredited by NAAC with 'A' Grade & Accredited by NBA

Golden Jubilee
YEARS OF EXCELLENCE IN EDUCATION

A MINI PROJECT REPORT

for

Mini Project in Web Frame Works or Operating System (19CSE39)

CLOTHING STORE SYSTEM

Submitted by

B.AJAY
USN: 1NH20CS039
SEM/SEC: 3/A

In partial fulfillment for the award of
the degree of

Bachelor of Engineering

in

COMPUTER SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC
Accredited by NAAC with 'A' Grade & Accredited by NBA

50 Golden Jubilee YEARS OF EXCELLENCE IN EDUCATION

Certificate

This is to certify that the mini project work titled

CLOTHING STORE SYSTEM

Submitted in partial fulfillment of the degree of
Bachelor of Engineering in
Computer Science and Engineering by

B.AJAY
USN: 1NH20CS039

DURING

ODD SEMESTER 2021-2022

for

COURSE CODE: 19CSE39

Signature of Reviewer Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner	*Signature with date*
1. _____	_____
2. _____	_____

ABSTRACT

Clothing store System Project is a simple console application designed to demonstrate the practical use of C programming language with data structure and its features .This mini-project is an effective management system which is very useful for clothing shop owners to manage the stocks of the different clothing items available in the shop. In the olden days, clothing shops used to employ an inventory manager who used to manage all the stock details whenever the shop buys stock from the wholesaler or sell products to the customer. Inventory managers used to manually update the stock numbers by entering them in the register. But now with the help of this program, Clothing shop owners can manage the stocks of clothing items themselves by entering the details of the amount of clothing items bought from the wholesaler and the amount of items sold to the customer. When the user enters these details ,the program will automatically compute the stock details and display it to the user. This makes it very easy for the clothing shop owners to manage the stocks and they also don't need to employ a person for managing the stock. This clothing store system mini-project also has an extra feature where the program displays the daily sales report in which it shows the various clothing items have been sold and the amount of money that has been made from the sales in that particular day. This can be very useful for the clothing shop owners as it can give them a detailed overview of what clothing items are in higher demand so that they can strategically plan which clothing items to buy from the wholesaler and set a price for them so that they can generate a profit. This mini-project helps us understand various concepts of data structures in c programming language. This type of program can be extended for any type of product management system.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for his constant support and encouragement.

I am grateful to **Dr. Amarjeet Singh,** Dean - Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I also express my gratitude to **Dr. Jayashree ,** Department of Computer Science and Engineering, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

B.AJAY

1NH20CS039

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT :

Clothing store system is a mini project that allows the user to enter the amount of clothing items bought/purchased from the wholesaler and also enter the amount of clothing items sold to the customer.when the user enters these details,the program automatically calculates and computes the updated stocks of the various clothing items available in the shop. The whole concept is designed via c language. This project helps us understand various concepts of data structures and structures in c programming language. This type of program can be extended for any type of product management system.This mini-project also displays the daily sales report in which the detailed information about different clothing items sold and the amount of money made from these sales are shown.This mini-project can be of great use to the clothing shop owners as it can save a lot of time and resources in calculating the stock details manually.The clothing shop owners can also benefit from the daily sales report showed by this mini-project as it will show which clothing items are in higher demand so that the clothing shop owner can buy more products from the wholesaler and a fix a higher price in order to make more profit.

1.2 OBJECTIVES :

The main aim of this mini-project is to create an application which can be used in clothing store to make the process of updating the stock items by the shop owner as and when there is a purchase of stock from the wholesaler or a customer purchases items from the shop as easy and hassle free as possible.The main objective of this mini-project is to make life easier for clothing shop owners by minimizing the complexity involved in managing the stock details of the store.this program provides an user friendly interface so that user can enter the stock details and any other information regarding purchase or sale of clothing items easily and get the updated stock details instantly. Clothing shop owner is also provided with the daily sales report so that they know which items are liked by the customers,with the help of which they can plan a new sale to increase profits.

1.3 METHODOLOGY TO BE FOLLOWED :

In this mini-project ,I have used the concept of linked list which is a data structure. I have created a structure ,with the help of which I have created different nodes for different clothing items like shirt,pant,sweater etc.these nodes contain the details for that particular clothing item like cloth number,name,quantity and price. These nodes are connected with help of linked list. Whenever a need arises to update the stock details, I can traverse through this linked list and find the clothing item whose details need to be updated and change the data stored in its members.i have also added another functionality in this clothing store system mini-project to display the daily sales report. This was achieved by collecting the details of all the clothing items and computing the amount of money made by selling those items.

1.4 EXPECTED OUTCOMES :

The main expected outcome is that the process of managing the stock details which is very tedious and time consuming thing when done manually becomes an easy and hassle-free process. Another expected outcome from this mini-project is to provide a user friendly interface to the user so that he/she can easily update their stocks with the help of a few clicks in the computer instead of maintaining a register and writing down the details after each and every purchase or sale. Among other expected outcomes is to provide the clothing shop owner with a detailed and in depth overview of the daily sales report which gives the owner a clear insight as to which clothing items are liked by the customers.

1.5 HARDWARE AND SOFTWARE REQUIREMENTS :

HARDWARE REQUIREMENTS :

1) Processor : 800 MHz Intel Pentium or better.
2) Memory:128 MB or more.
3) Disk space : 512 KB of free disk space or more.

SOFTWARE REQUIREMENTS :

1) Operating system : Microsoft Windows XP/Vista/Windows 7/Windows 10.
2) Software : C compiler.

CHAPTER 2

FUNDAMENTALS OF DATA STRUCTURES IN C

2.1 DATA STRUCTURES

Data Structure is a way to store and organize data so that it can be used efficiently.

The data structure name indicates itself that organizing the data in memory. There are many ways of organizing the data in the memory as we have already seen one of the data structures, i.e., array in C language. Array is a collection of memory elements in which data is stored sequentially, i.e., one after another. In other words, we can say that array stores the elements in a continuous manner. This organization of data is done with the help of an array of data structures. There are also other ways to organize the data in memory. Let's see the different types of data structures.

The data structure is not any programming language like C, C++, java, etc. It is a set of algorithms that we can use in any programming language to structure the data in the memory.

To structure the data in memory, 'n' number of algorithms were proposed, and all these algorithms are known as Abstract data types. These abstract data types are the set of rules.

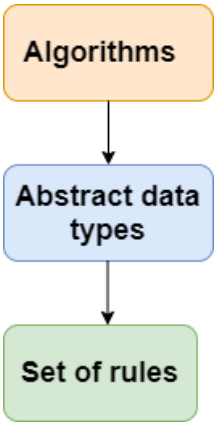

figure 2.1-data strucures

There are two types of data structures:

- Primitive data structure

- Non-primitive data structure

Primitive Data structure :

The primitive data structures are primitive data types. The int, char, float, double, and pointer are the primitive data structures that can hold a single value.

Non-Primitive Data structure :

The non-primitive data structure is divided into two types:

- Linear data structure
- Non-linear data structure

Linear Data Structure :

The arrangement of data in a sequential manner is known as a linear data structure. The data structures used for this purpose are Arrays, Linked list, Stacks, and Queues. In these data structures, one element is connected to only one another element in a linear form.

When one element is connected to the 'n' number of elements known as a non-linear data structure. The best example is trees and graphs. In this case, the elements are arranged in a random manner.

We will discuss the above data structures in brief in the coming topics. Now, we will see the common operations that we can perform on these data structures.

Data structures can also be classified as :

- **Static data structure :** It is a type of data structure where the size is allocated at the compile time. Therefore, the maximum size is fixed.
- **Dynamic data structure :** It is a type of data structure where the size is allocated at the run time. Therefore, the maximum size is flexible.

Advantages of data structures

The following are the advantages of a data structure:

- **Efficiency:** If the choice of a data structure for implementing a particular ADT is proper, it makes the program very efficient in terms of time and space.
- **Reusability:** The data structure provides reusability means that multiple client programs can use the data structure.
- **Abstraction:** The data structure specified by an ADT also provides the level of abstraction. The client cannot see the internal working of the data structure, so it does not have to worry about the implementation part. The client can only see the interface.

2.2 ARRAYS

- Arrays are defined as the collection of similar type of data items stored at contiguous memory locations.
- Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc.
- Array is the simplest data structure where each data element can be randomly accessed by using its index number.
- For example, if we want to store the marks of a student in 6 subjects, then we don't need to define different variable for the marks in different subject. instead of that, we can define an array which can store the marks in each subject at a the contiguous memory locations.

The array **marks[10]** defines the marks of the student in 10 different subjects where each subject marks are located at a particular subscript in the array i.e. **marks[0]** denotes the marks in first subject, **marks[1]** denotes the marks in 2nd subject and so on.

Properties of arrays :

- Each element is of same data type and carries a same size i.e. int = 4 bytes.
- Elements of the array are stored at contiguous memory locations where the first element is stored at the smallest memory location.

- Elements of the array can be randomly accessed since we can calculate the address of each element of the array with the given base address and the size of data element.

for example, in C language, the syntax of declaring an array is like following:

```
int arr[10]; char arr[10]; float arr[5]
```

Need of using arrays :

In computer programming, the most of the cases requires to store the large number of data of similar type. To store such amount of data, we need to define a large number of variables. It would be very difficult to remember names of all the variables while writing the programs. Instead of naming all the variables with a different name, it is better to define an array and store all the elements into it.

Advantages of using arrays

- Array provides the single name for the group of variables of the same type therefore, it is easy to remember the name of all the elements of an array.
- Traversing an array is a very simple process, we just need to increment the base address of the array in order to visit each element one by one.
- Any element in the array can be directly accessed by using the index.

2.3 LINKED LIST

- Linked List can be defined as collection of objects called **nodes** that are randomly stored in the memory.
- A node contains two fields i.e. data stored at that particular address and the pointer which contains the address of the next node in the memory.
- The last node of the list contains pointer to the null.

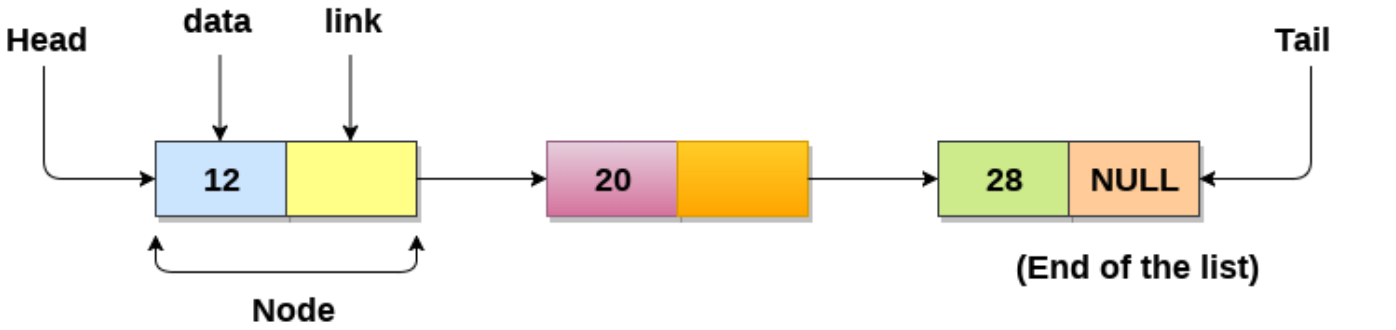

Figure 2.2-linked list

Uses of linked list :

- The list is not required to be contiguously present in the memory. The node can reside any where in the memory and linked together to make a list. This achieves optimized utilization of space.
- list size is limited to the memory size and doesn't need to be declared in advance.
- Empty node can not be present in the linked list.
- We can store values of primitive types or objects in the singly linked list.

Why used linked list over array?

Till now, we were using array data structure to organize the group of elements that are to be stored individually in the memory. However, Array has several advantages and disadvantages which must be known in order to decide the data structure which will be used throughout the program.

Array contains following limitations :

- The size of array must be known in advance before using it in the program.
- Increasing size of the array is a time taking process. It is almost impossible to expand the size of the array at run time.
- All the elements in the array need to be contiguously stored in the memory. Inserting any element in the array needs shifting of all its predecessors.

Linked list is the data structure which can overcome all the limitations of an array. Using linked list is useful because,

- It allocates the memory dynamically. All the nodes of linked list are non-contiguously stored in the memory and linked together with the help of pointers.
- Sizing is no longer a problem since we do not need to define its size at the time of declaration. List grows as per the program's demand and limited to the available memory space.

Singly linked list :

Singly linked list can be defined as the collection of ordered set of elements. The number of elements may vary according to need of the program. A node in the singly linked list consist of two parts: data part and link part. Data part of the node stores actual information that is to be represented by the node while the link part of the node stores the address of its immediate successor.

One way chain or singly linked list can be traversed only in one direction. In other words, we can say that each node contains only next pointer, therefore we can not traverse the list in the reverse direction.

Consider an example where the marks obtained by the student in three subjects are stored in a linked list as shown in the figure.

Figure 2.3-linked list

In the above figure, the arrow represents the links. The data part of every node contains the marks obtained by the student in the different subject. The last node in the list is identified by the null pointer which is present in the address part of the last node. We can have as many elements we require, in the data part of the list.

Doubly linked list :

Doubly linked list is a complex type of linked list in which a node contains a pointer to the previous as well as the next node in the sequence. Therefore, in a doubly linked list, a node consists of three parts: node data, pointer to

the next node in sequence (next pointer) , pointer to the previous node (previous pointer). A sample node in a doubly linked list is shown in the figure.

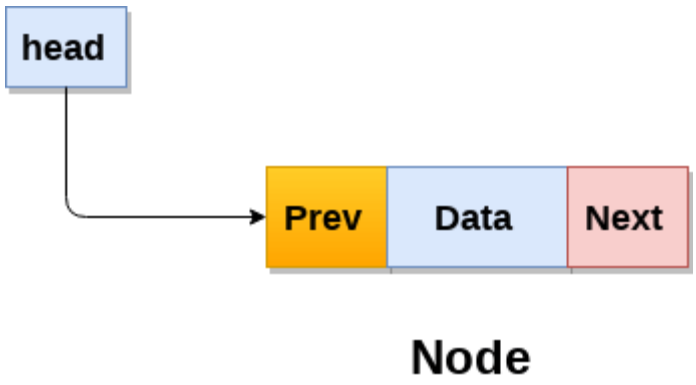

Figure 2.4-linked list

A doubly linked list containing three nodes having numbers from 1 to 3 in their data part, is shown in the following image.

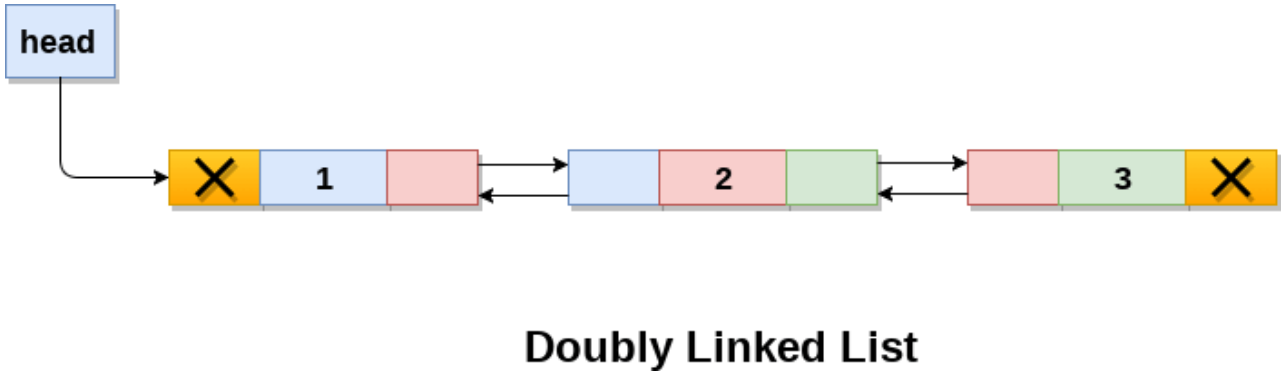

Figure 2.5-linked list

Other types of linked lists :

- Circular singly linked list
- Circular doubly linked list

2.4 STACKS

A Stack is a linear data structure that follows the **LIFO (Last-In-First-Out)** principle. Stack has one end, whereas the Queue has two ends (**front and rear**). It contains only one pointer **top pointer** pointing to the topmost element of the stack. Whenever an element is added in the stack, it is added on the top of the stack, and the element can be deleted only from the stack. In other words, a *stack can be defined as a container in which insertion and deletion can be done from the one end known as the top of the stack.*

- It is called as stack because it behaves like a real-world stack, piles of books, etc.
- A Stack is an abstract data type with a pre-defined capacity, which means that it can store the elements of a limited size.
- It is a data structure that follows some order to insert and delete the elements, and that order can be LIFO or FILO.

Working of a stack

Stack works on the LIFO pattern. As we can observe in the below figure there are five memory blocks in the stack; therefore, the size of the stack is 5.

Suppose we want to store the elements in a stack and let's assume that stack is empty. We have taken the stack of size 5 as shown below in which we are pushing the elements one by one until the stack becomes full.

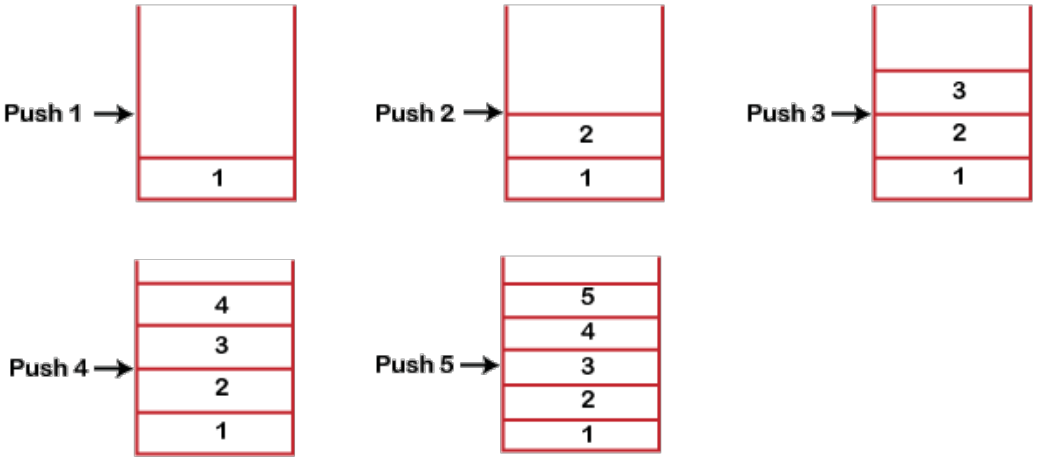

Figure 2.6-stacks

Since our stack is full as the size of the stack is 5. In the above cases, we can observe that it goes from the top to the bottom when we were entering the new element in the stack. The stack gets filled up from the bottom to the top.

When we perform the delete operation on the stack, there is only one way for entry and exit as the other end is closed. It follows the LIFO pattern, which means that the value entered first will be removed last. In the above case, the value 5 is entered first, so it will be removed only after the deletion of all the other elements.

Standard stack operations

The following are some common operations implemented on the stack:

- **push():** When we insert an element in a stack then the operation is known as a push. If the stack is full then the overflow condition occurs.
- **pop():** When we delete an element from the stack, the operation is known as a pop. If the stack is empty means that no element exists in the stack, this state is known as an underflow state.
- **isEmpty():** It determines whether the stack is empty or not.
- **isFull():** It determines whether the stack is full or not.'
- **peek():** It returns the element at the given position.
- **count():** It returns the total number of elements available in a stack.
- **change():** It changes the element at the given position.
- **display():** It prints all the elements available in the stack.

2.5 QUEUES

- A queue can be defined as an ordered list which enables insert operations to be performed at one end called **REAR** and delete operations to be performed at another end called **FRONT.**
- Queue is referred to be as First In First Out list.
- For example, people waiting in line for a rail ticket form a queue.

Figure 2.7-queues

Applicatioons of queue :

Due to the fact that queue performs actions on first in first out basis which is quite fair for the ordering of actions. There are various applications of queues discussed as below.

- Queues are widely used as waiting lists for a single shared resource like printer, disk, CPU.
- Queues are used in asynchronous transfer of data (where data is not being transferred at the same rate between two processes) for eg. pipes, file IO, sockets.
- Queues are used as buffers in most of the applications like MP3 media player, CD player, etc.
- Queue are used to maintain the play list in media players in order to add and remove the songs from the play-list.
- Queues are used in operating systems for handling interrupts.

2.5 TREES

We read the linear data structures like an array, linked list, stack and queue in which all the elements are arranged in a sequential manner. The different data structures are used for different kinds of data.

Some factors are considered for choosing the data structure:

- **What type of data needs to be stored**?: It might be a possibility that a certain data structure can be the best fit for some kind of data.
- **Cost of operations:** If we want to minimize the cost for the operations for the most frequently performed operations. For example, we have a simple list on which we have to perform the search operation; then, we can create an array in which elements are stored in sorted order to perform the *binary search*. The binary search works very fast for the simple list as it divides the search space into half.
- **Memory usage:** Sometimes, we want a data structure that utilizes less memory.

A tree is also one of the data structures that represent hierarchical data. Suppose we want to show the employees and their positions in the hierarchical form then it can be represented as shown below:

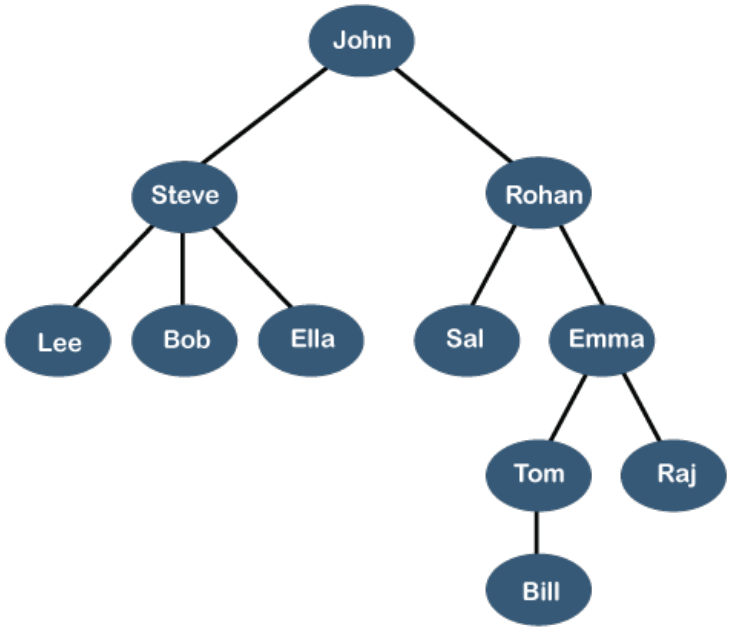

Figure 2.8-trees

The above tree shows the **organization hierarchy** of some company. In the above structure, *john* is the **CEO** of the company, and John has two direct reports named as *Steve* and *Rohan*. Steve has three direct reports named *Lee, Bob, Ella* where *Steve* is a manager. Bob has two direct reports named *Sal* and *Emma***. Emma** has two direct reports named *Tom* and *Raj*. Tom has one direct report named *Bill*. This particular logical structure is known as a *Tree*. Its structure is similar to the real tree, so it is named a *Tree*. In this structure, the *root* is at the top, and its branches are moving in a downward direction. Therefore, we can say that the Tree data structure is an efficient way of storing the data in a hierarchical way.

Let's understand some key points of the Tree data structure.

- A tree data structure is defined as a collection of objects or entities known as nodes that are linked together to represent or simulate hierarchy.
- A tree data structure is a non-linear data structure because it does not store in a sequential manner. It is a hierarchical structure as elements in a Tree are arranged in multiple levels.
- In the Tree data structure, the topmost node is known as a root node. Each node contains some data, and data can be of any type. In the above tree structure, the node contains the name of the employee, so the type of data would be a string.
- Each node contains some data and the link or reference of other nodes that can be called children.

2.6 BINARY TREE

The Binary tree means that the node can have maximum two children. Here, binary name itself suggests that 'two'; therefore, each node can have either 0, 1 or 2 children.

Let's understand the binary tree through an example.

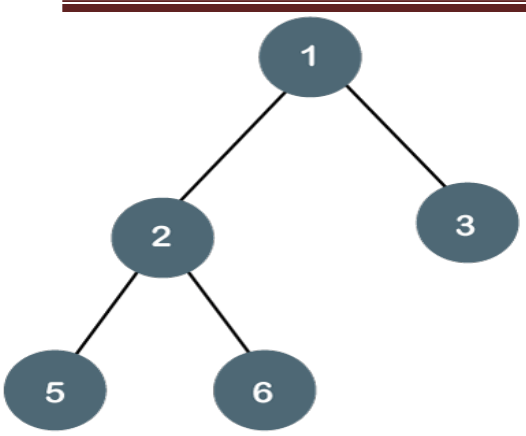

Figure 2.9-binary trees

The above tree is a binary tree because each node contains the utmost two children. The logical representation of the above tree is given below:

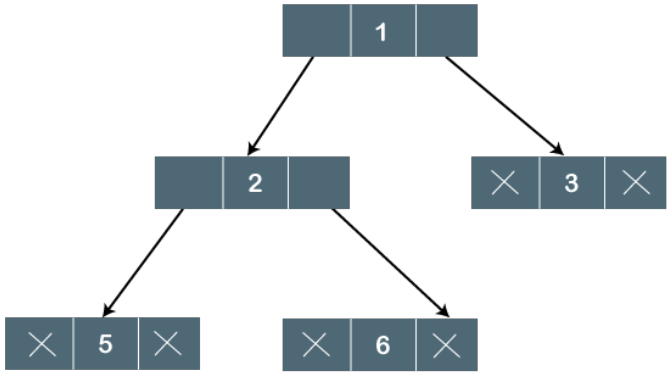

Figure 2.10-binary trees

In the above tree, node 1 contains two pointers, i.e., left and a right pointer pointing to the left and right node respectively. The node 2 contains both the nodes (left and right node); therefore, it has two pointers (left and right). The nodes 3, 5 and 6 are the leaf nodes, so all these nodes contain **NULL** pointer on both left and right parts.

Propeerties of binary trees

- At each level of i, the maximum number of nodes is 2^i.
- The height of the tree is defined as the longest path from the root node to the leaf node. The tree which is shown above has a height equal to 3. Therefore, the maximum number of nodes at height 3 is equal to

(1+2+4+8) = 15. In general, the maximum number of nodes possible at height h is $(2^0 + 2^1 + 2^2 + \ldots 2^h) = 2^{h+1} - 1$.

- The minimum number of nodes possible at height h is equal to **h+1.**
- If the number of nodes is minimum, then the height of the tree would be maximum. Conversely, if the number of nodes is maximum, then the height of the tree would be minimum.

CHAPTER 3

DESIGN

3.1 DESIGN GOALS

Our design goals includes two main things :

1) Reduced complexity in storing and updating stock details of different clothing items
2) Provide detailed information on daily sales report.

3.2 FLOWCHART

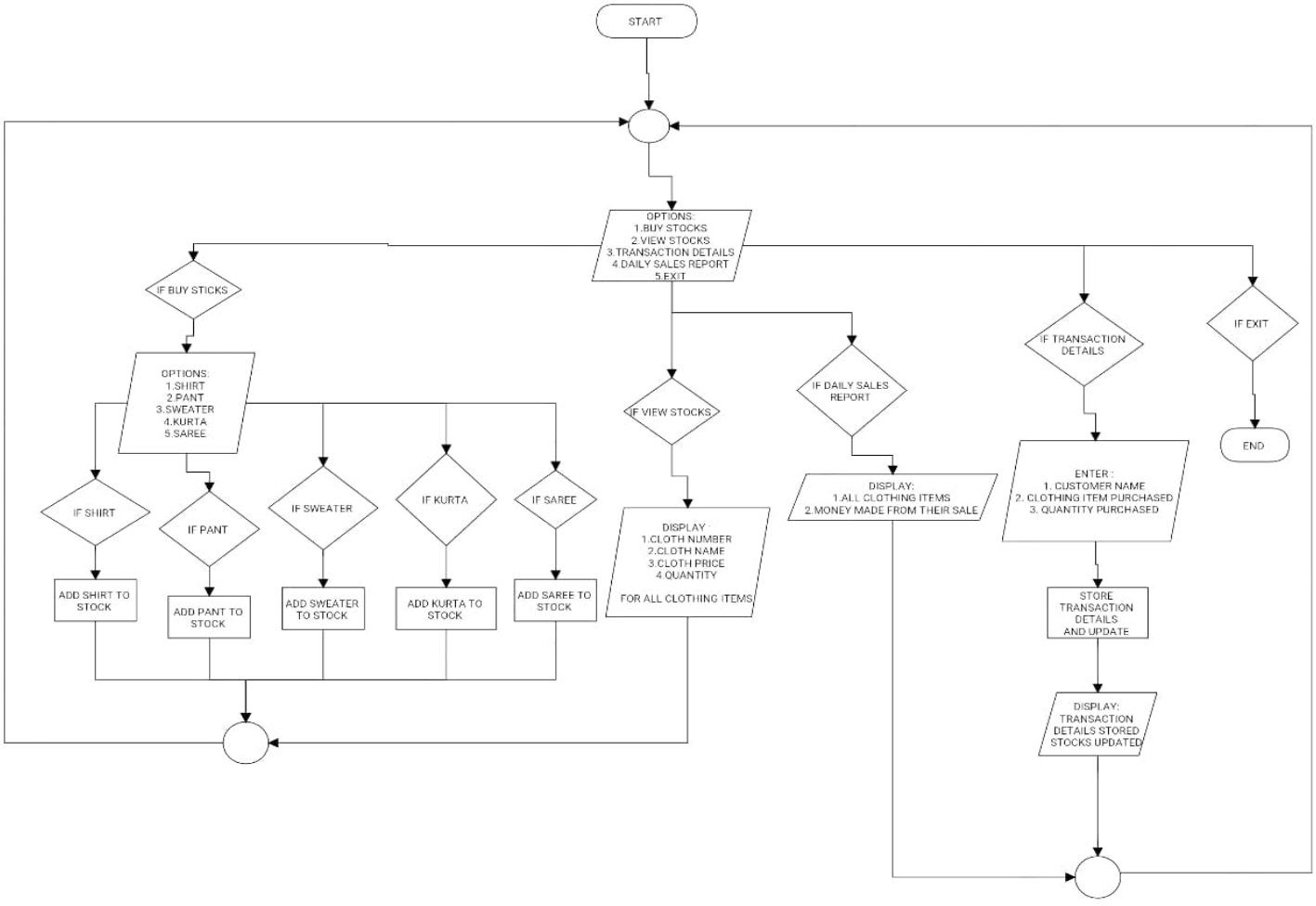

Figure 3.1-flowchart

CHAPTER 4

IMPLEMENTATION

4.1 MODULE 1 FUNCTONALITY

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
struct cloth
{
int clothno;
char clothname[10];
int clothprice;
int qty;
int soldqty;
struct cloth *link;
};
```

This structure is used to implement singly liked list in the program which is later used to store the user data.

It contains 5 fields that is used to store the details such as number of clothing item,name of the clothing item,price of the clothing item,price of the clothing item,quantity of clothing item to b added to to stock and quantity of clothing item sold to customer.

4.2 MODULE 2 FUNCTIONALITY

```
void transdetails()
{
char custname[10],cname[10];
int cqty;
printf("Enter customer name: ");
scanf("%s",custname);
c:
printf("\n\nwhat clothing item did %s purchase? ",custname);
scanf("%s",cname);
if(strcmp(cname,"shirt")==0)
{
printf("\n\nhow many %ss did %s purchase? ",cname,custname);
scanf("%d",&cqty);
shirt->soldqty=shirt->soldqty+cqty;
shirt->qty=shirt->qty-cqty;
printf("\n\nTRANSANCTION DETAILS STORED \nSTOCKS UPDATED\n");
getch();
}
else if(strcmp(cname,"pant")==0)
{
printf("\n\nhow many %ss did %s purchase? ",cname,custname);
scanf("%d",&cqty);
pant->soldqty=pant->soldqty+cqty;
pant->qty=pant->qty-cqty;
printf("\n\nTRANSANCTION DETAILS STORED \nSTOCKS UPDATED\n");
getch();
}
else if(strcmp(cname,"sweater")==0)
{
```

```
printf("\n\nhow many %ss did %s purchase? ",cname,custname);
scanf("%d",&cqty);
sweater->soldqty=sweater->soldqty+cqty;
sweater->qty=sweater->qty-cqty;
printf("\n\nTRANSANCTION DETAILS STORED \nSTOCKS UPDATED\n");
getch();
}
else if(strcmp(cname,"kurta")==0)
{
printf("\n\nhow many %ss did %s purchase? ",cname,custname);
scanf("%d",&cqty);
kurta->soldqty=kurta->soldqty+cqty;
kurta->qty=kurta->qty-cqty;
printf("\n\nTRANSANCTION DETAILS STORED \nSTOCKS UPDATED\n");
getch();
}
else if(strcmp(cname,"saree")==0)
{
printf("\n\nhow many %ss did %s purchase? ",cname,custname);
scanf("%d",&cqty);
saree->soldqty=saree->soldqty+cqty;
saree->qty=saree->qty-cqty;
printf("\n\nTRANSANCTION DETAILS STORED \nSTOCKS UPDATED\n");
getch();
}
else
{
printf("\n\nEnter the cloth name correctly \n");
goto c;
}
}
```

The above module is used to enter the transaction details so that program can receive the amount of clothing items bought by the customer and automatically subtract the value from the stock of the clothing item which is already available.

In this module the program asks for the input for the name of the customer ,name of the clothing item bought by the customer and the quantity of the clothing item bought by the customer.These details are stored and used to compute the new value of the available stock of that particular clothing item.If the user enter the name of the clothing item incorrectly ,the program will ask the user to enter the name correctly again .

4.3 MODULE 3 FUNCTIONALITY

```
void viewstock()
{
struct cloth *temp;
temp=first;
clrscr();
printf("\033[1;34m");
printf("\t\t\t\tSTOCKS\n");
printf("\033[0m");
printf("\033[1;31m");
printf("\nCLOTH NO\t\tCLOTH NAME\t\tCLOTH PRICE\t\tQUANTITY");
printf("\033[0m");
printf("\n\n");
while(temp!=NULL)
{
printf("    %d\t\t\t%s\t\t\t\t%d\t\t  %d\n\n",temp->clothno,temp->clothname,temp->clothprice,temp->qty);
temp=temp->link;
}
getch();
}
```

The above module is used to view the available clothing items and their quantities which are available in stock.

In this module,I have used a while loop with the help of which the linked list has been traversed and as and when a node is accessed,the details stored by its members are displayed.

4.4 MODULE 4 FUNCTIONALITY

```
void buystock()
{
int ch,qty1,i;
printf("\n\n");
printf("\033[1;31m");
printf("\t\t\t   AVAILABLE CLOTHING ITEMS : \n");
printf("\033[0m");
printf("\n\n\t\t\t1.SHIRT\t1 SHIRT=RS %d\n\n\n\t\t\t2.PANT\t1 PANT=RS %d\n\n\n\t\t\t3.SWEATER 1 SWEATER=RS %d\n\n\n\t\t\t4.KURTA\t1 KURTA=RS %d\n\n\n\t\t\t5.SAREE\t1 SAREE=RS %d\n\n\n\t\t\tENTER YOUR CHOICE : ",shirtprice,pantprice,sweaterprice,kurtaprice,sareeprice);
scanf("%d",&ch);
switch(ch)
{
case 1:
printf("\n\t\t\tENTER THE NO OF SHIRTS TO BE ADDED TO STOCK : ");
scanf("%d",&qty1);
shirt->qty=shirt->qty+qty1;
break;

case 2:
printf("\n\t\t\tENTER THE NO OF PANTS TO BE ADDED TO STOCK : ");
scanf("%d",&qty1);
pant->qty=pant->qty+qty1;
break;
```

```
case 3:
printf("\n\t\t\tENTER THE NO OF SWEATERS TO BE ADDED TO STOCK : ");
scanf("%d",&qty1);
sweater->qty=sweater->qty+qty1;
break;

case 4:
printf("\n\t\t\tENTER THE NO OF KURTAS TO BE ADDED TO STOCK : ");
scanf("%d",&qty1);
kurta->qty=kurta->qty+qty1;
break;

case 5:
printf("\n\t\t\tENTER THE NO OF SAREES TO BE ADDED TO STOCK : ");
scanf("%d",&qty1);
saree->qty=saree->qty+qty1;
break;

default:
printf("\n\t\t\tENTER A VALID CHOICE\n");
break;
}
}
```

The above module is used to receive the details of clothing items bought from the wholesaler and add them to the stock of the clothing store.

In this module,the program asks for the input from the user as to which clothing item he/she wishes to purchase from the wholesaler and the amount of clothing items needed to be purchased.with the help of this data I have accessed the linked list and updated the quantity of the clothing items by adding it to the stock.

4.5 MODULE 5 FUNCTIONALITY

```
void dailyreport()
{
clrscr();
printf("\033[1;35m");
printf("\t\t\t DAILY SALES REPORT \n");
printf("\033[0m");
printf("\033[0;32m");
printf("\n\n\tSHIRT SALES : ");
printf("\033[0m");
printf("\nNUMBER OF SHIRTS SOLD : %d\n",shirt->soldqty);
printf("MONEY MADE FROM SHIRT SALES : %d\n",shirtprice*shirt->soldqty);
printf("\033[0;32m");
printf("\n\tPANT SALES : ");
printf("\033[0m");
printf("\nNUMBER OF PANTS SOLD : %d\n",pant->soldqty);
printf("MONEY MADE FROM PANT SALES : %d\n",pantprice*pant->soldqty);
printf("\033[0;32m");
printf("\n\tSWEATER SALES : ");
printf("\033[0m");
printf("\nNUMBER OF SWEATERS SOLD : %d\n",sweater->soldqty);
printf("MONEY MADE FROM SWEATER SALES : %d\n",sweaterprice*sweater->soldqty);
printf("\033[0;32m");
printf("\n\tKURTA SALES : ");
printf("\033[0m");
printf("\nNUMBER OF KURTAS SOLD : %d\n",kurta->soldqty);
printf("MONEY MADE FROM KURTA SALES : %d\n",kurtaprice*kurta->soldqty);
printf("\033[0;32m");
printf("\n\tSAREE SALES : ");
```

```
printf("\033[0m");
printf("\nNUMBER OF SAREES SOLD : %d\n",saree->soldqty);
printf("MONEY MADE FROM SAREE SALES : %d\n",sareeprice*saree->soldqty);
getch();
}
```

The above module is used to display the daily sales report which gives the detailed information on which clothing items are liked by the customers and have had higher sales compared to other clothing items.

In the above module I have a accessed the linked list and taken the value of its members and calculated the amount of money earned from the sale all clothing items and displayed it to give a detailed overview of the daily sales report.

4.6 MODULE 6 FUNCTIONALITY

```
void setvalue()
{
int i;
struct cloth*temp;
for(i=1;i<=5;i++)
{
temp=(struct cloth*) malloc(sizeof(struct cloth));
if(i==1)
{
temp->clothno=1;
strcpy(temp->clothname,"shirt");
temp->clothprice=shirtprice;temp->qty=0;
temp->soldqty=0;
shirt=temp;
}
```

```
if(i==2)
{
temp->clothno=2;
strcpy(temp->clothname,"pant");
temp->clothprice=pantprice;
temp->qty=0;
temp->soldqty=0;
pant=temp;
}
if(i==3)
{
temp->clothno=3;
strcpy(temp->clothname,"sweater");
temp->clothprice=sweaterprice;
temp->qty=0;
temp->soldqty=0;
sweater=temp;
}
if(i==4)
{
temp->clothno=4;
strcpy(temp->clothname,"kurta");
temp->clothprice=kurtaprice;
temp->qty=0;
temp->soldqty=0;
kurta=temp;
}
if(i==5)
{
temp->clothno=5;
strcpy(temp->clothname,"saree");
temp->clothprice=sareeprice;
```

```
temp->qty=0;
temp->soldqty=0;
saree=temp;
}
}
first=shirt;
shirt->link=pant;
pant->link=sweater;
sweater->link=kurta;
kurta->link=saree;
saree->link=NULL;
}
```

The above module is used to create the nodes of all 5 different clothing items ,set the initial values of all the members of nodes as 0 and set the name of that specific clothing item.

In the above module ,I have created a for loop for iterating 5 times and in each iteration I've created a node and initialized all the node members with initial values as 0 and the link part of all nodes are set as NULL.

4.7 MODULE 7 FUNCTIONALITY

```
void main()
{
int ch,i,j,no;
setvalue();

clrscr();
for(j=0;j<11;j++)
{
printf("\n");
}
```

```
printf("\033[1;33m");
printf("\t\t\t    WELCOME TO CLOTH SHOP \n");
printf("\033[0m");
getch();
x:
clrscr();
for(j=0;j<8;j++)
{
printf("\n");
}
printf("\033[1;34m");
printf("\t\t\t\t     MAIN MENU\n\n");
printf("\033[0m");
printf("\t\t\t\t1.BUY   STOCKS\n\t\t\t\t2.VIEW   STOCKS\n\t\t\t\t3.TRANSACTION   DETAILS\n\t\t\t\t4.DAILY
SALES REPORT\n\t\t\t\t5.EXIT\n\n\t\t\t\tEnter your choice : ");
scanf("%d",&ch);
switch(ch)
{
case 1:
clrscr();
buystock();
goto x;

case 2:
getch();
viewstock();
goto x;

case 3:
clrscr();
transdetails();
goto x;
```

```
case 4:
dailyreport();
goto x;

case 5:
exit(0);

default:
printf("Enter a valid choice");
getch();
goto x;
}
getch();
}
```

The above module is used for displaying the main menu which consists of various options that the user can chose to perform a specific operation.This module makes the program user friendly.

In the above module I have used switch case to allow the user to choose between the various options to perform various actions/operations and in each case I have called a function to perform the operation.

CHAPTER 5

RESULTS

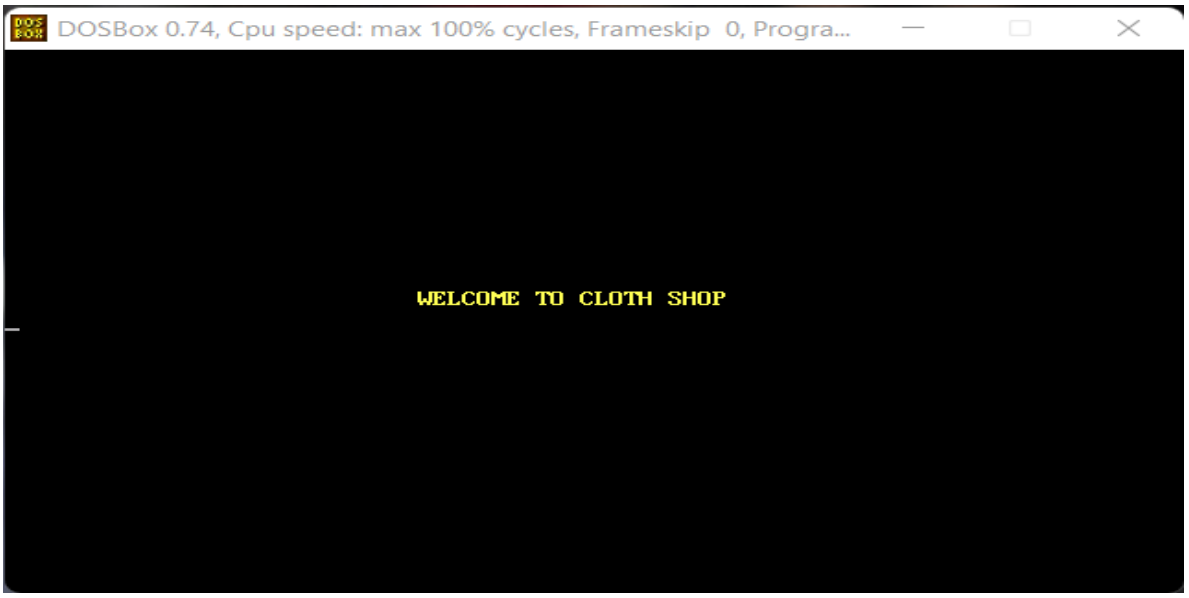

Figure 5.1-output screen 1

This is the the welcome screen.

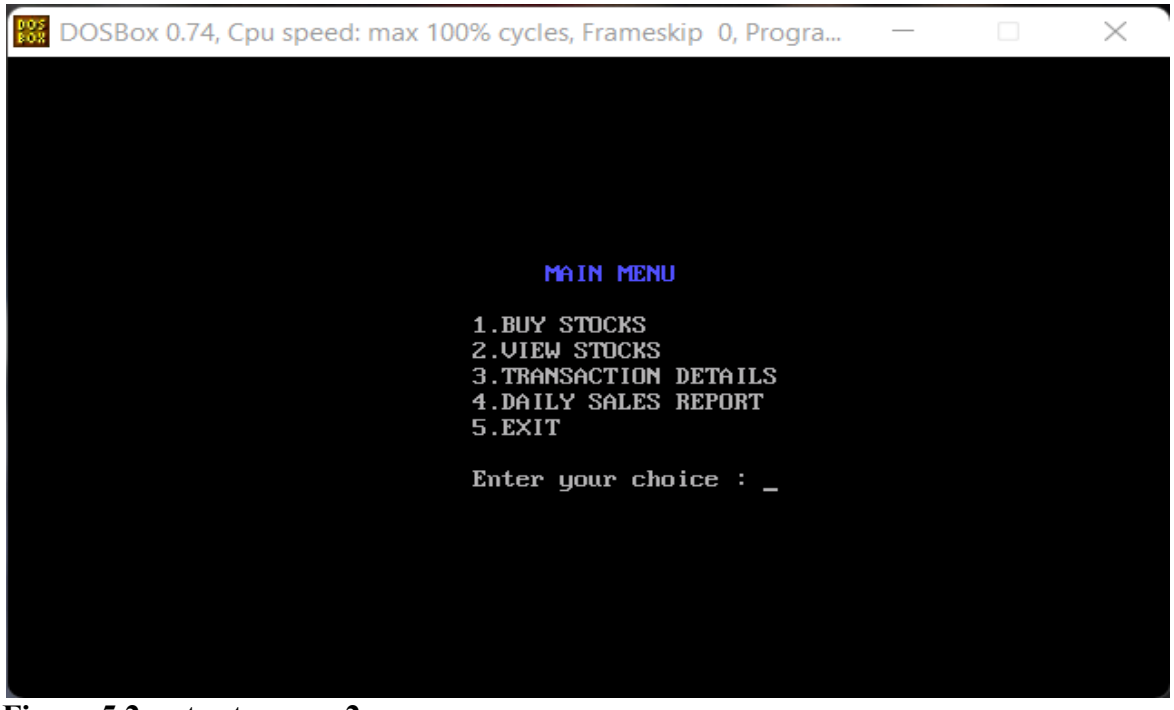

Figure 5.2-output screen 2

This is the Main Menu with the options,buy stocks,view stocks,transaction details,daily sales

report, and exit

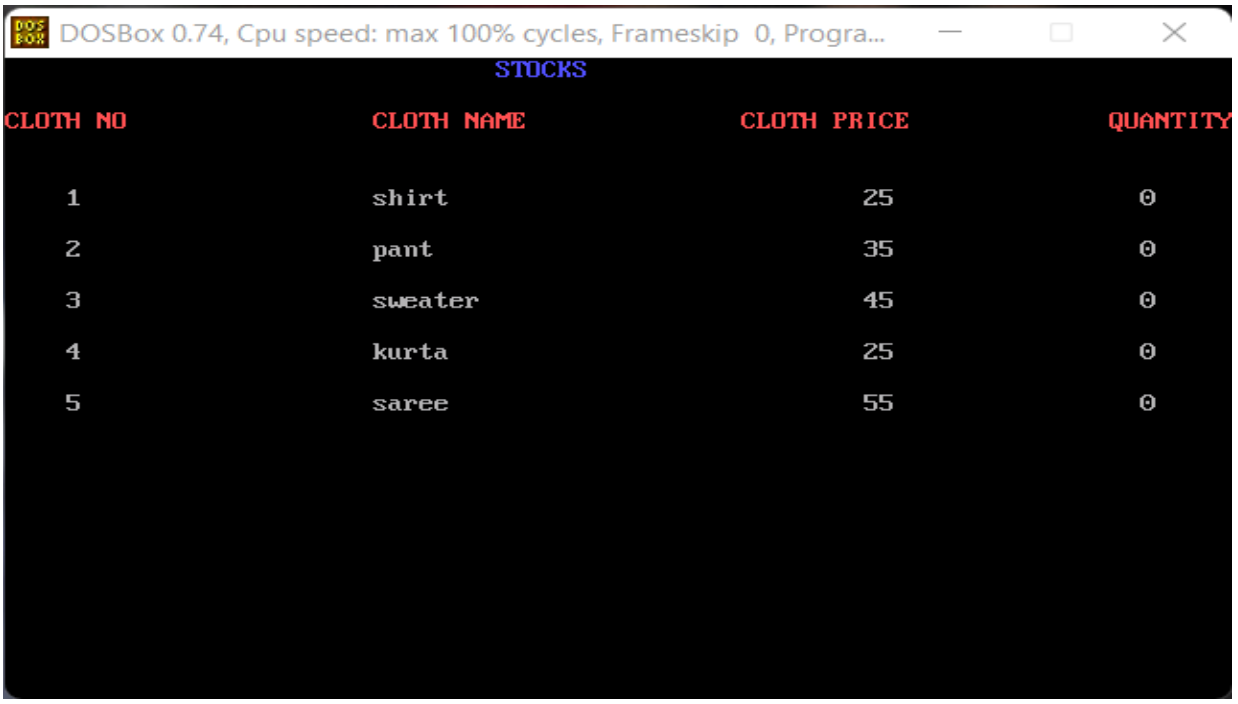

Figure 5.3-output screen 3

Here we have chosen option 2 in main menu which is view stocks.So in the beginning we haven't purchased an stock from the wholesaler so it shows all the items have quantity as 0.

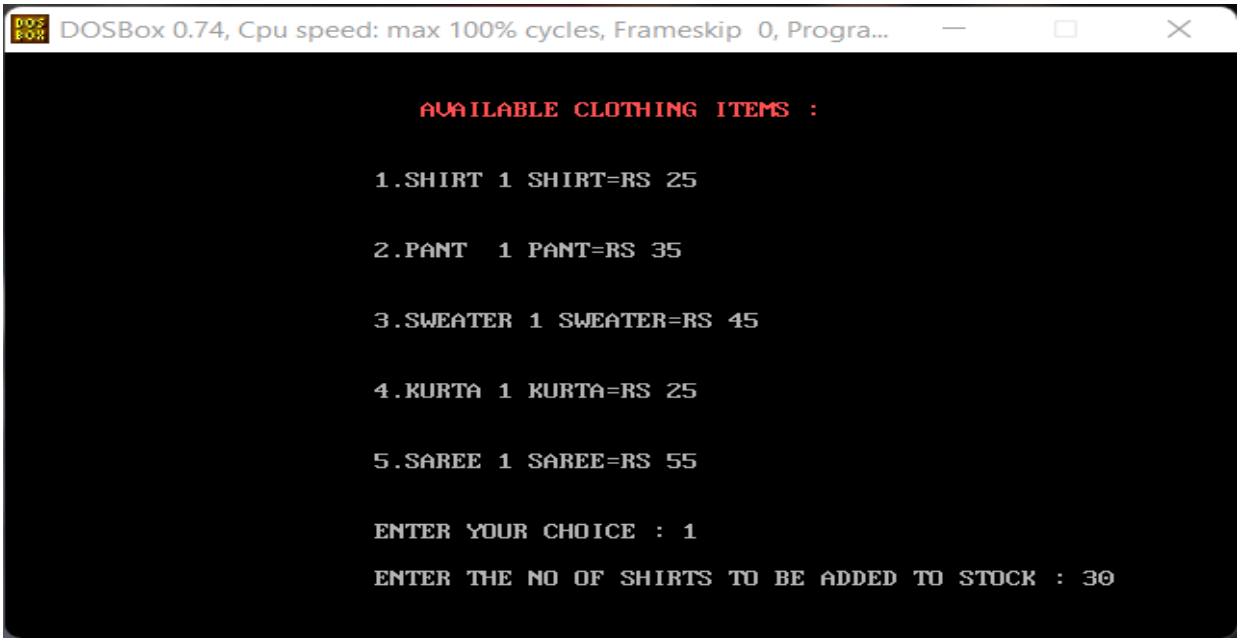

Figure 5.4-output screen 4

Here we have chosen option 1 in the main menu which is buy stocks and we are buying 30

shirts from the wholesaler.so this value will be stored and the stock value of shirts will be changed from 0 to 30.

We repeat the same process for buying 2 more clothing items which are pants and sweaters.We purchase 40 pants and 50 sweaters from the wholesaler for which the output screens are not shown

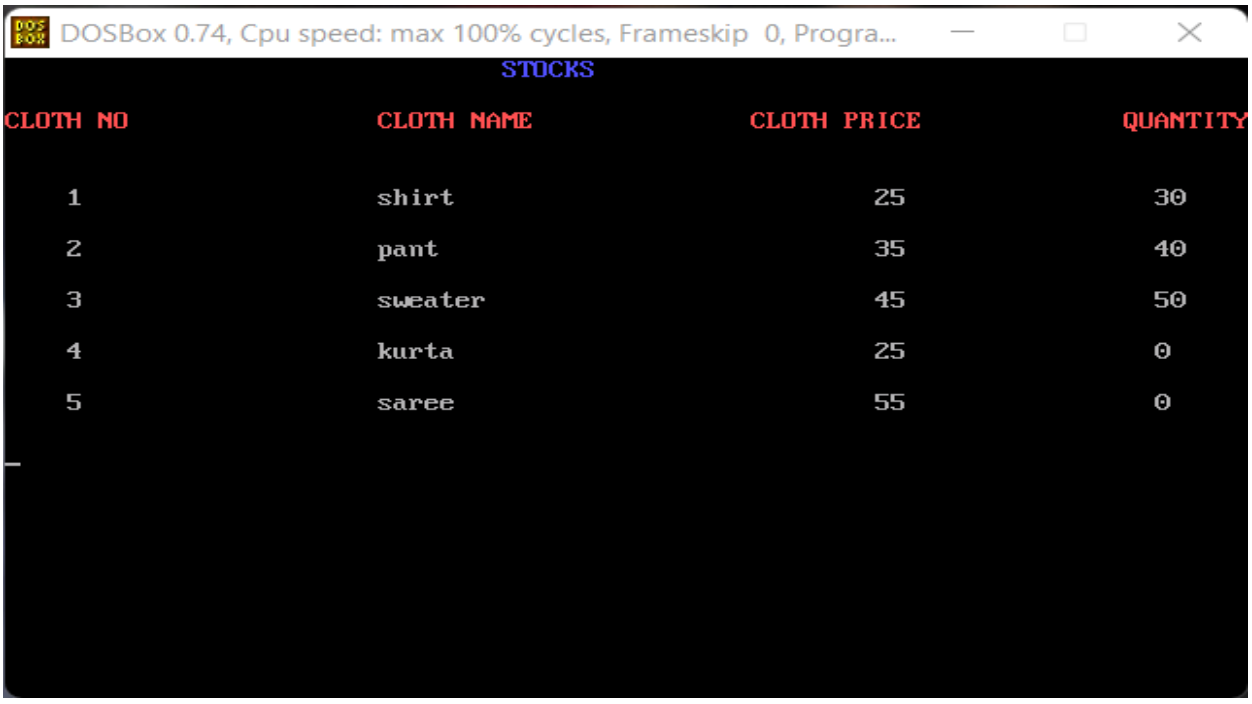

Figure 5.5-output screen 5

Now when we give option 2 in the main menu which is view stocks, we can see that 30 shirts ,40
pants and 50 sweaters have been added to stock.

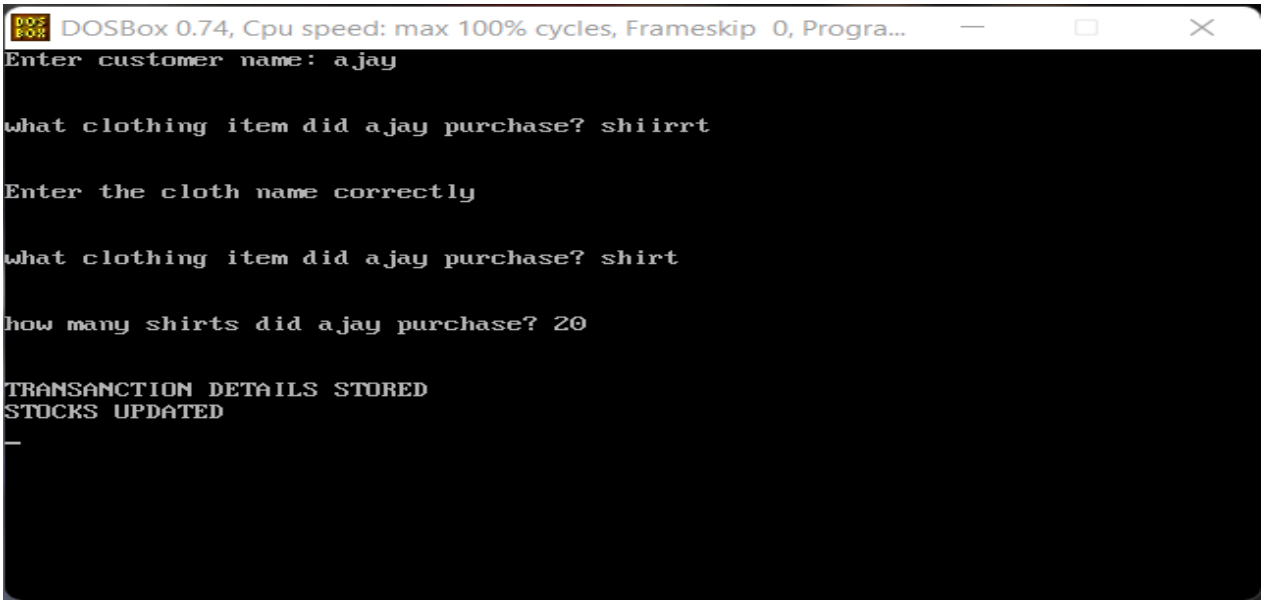

Figure 5.6-output screen 6

Here we have given option 3 in main menu which is transaction details.

we give the customer name as Ajay and when we give an incorrect name for the clothing item purchased , the program says the name is entered incorrectly and asks the user to enter the name again.When we give the correct name of clothing item, the program asks for the quantity of the clothing item purchased .we give the quantity as 20.So when the details are entered, it is stored and the quantity of clothing items purchased are subtracted from the stock.

We perform similar operations and give transaction details of purchasing 15 pants as well as 30 sweaters for which the output screens are not shown

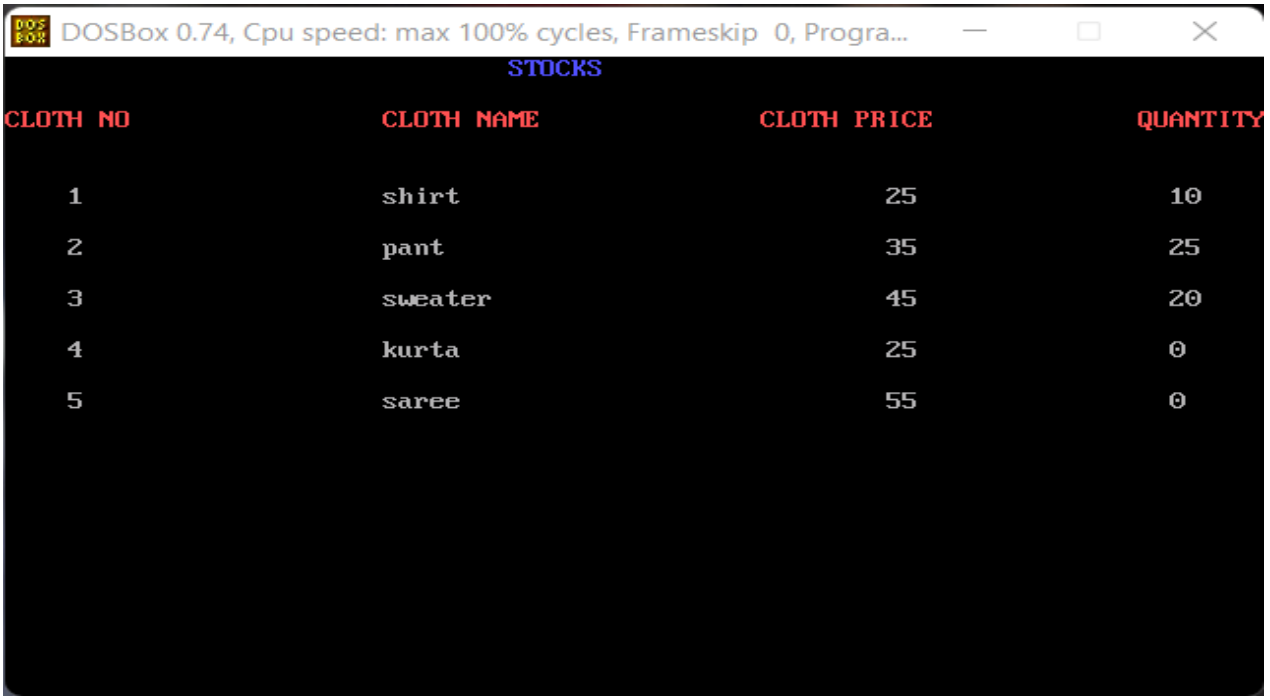

Figure 5.7-output screen 7

Now when we give option 2 in main menu which is view stocks ,we can see that 20 shirts ,15 pants and 30 sweaters have all been subtracted from the stock.

Figure 5.8-output screen 8

Now we have given option 1 which is buy stocks because after the sale of 20 shirts,15 pants and 30 sweaters we want to add some items to stock.we purchase 10 shirts from the wholesaler for which the output screen is given above.

We also purchase 10 more pants and 10 more sweaters from the wholesaler for which the output screen is not shown

Figure 5.9-output screen 9

Now when we give option 2 which is view stocks , we can see that the 10 shirts ,10 pants and 10 sweaters which we purchased from the wholesaler have been stored and updated the in the stocks.

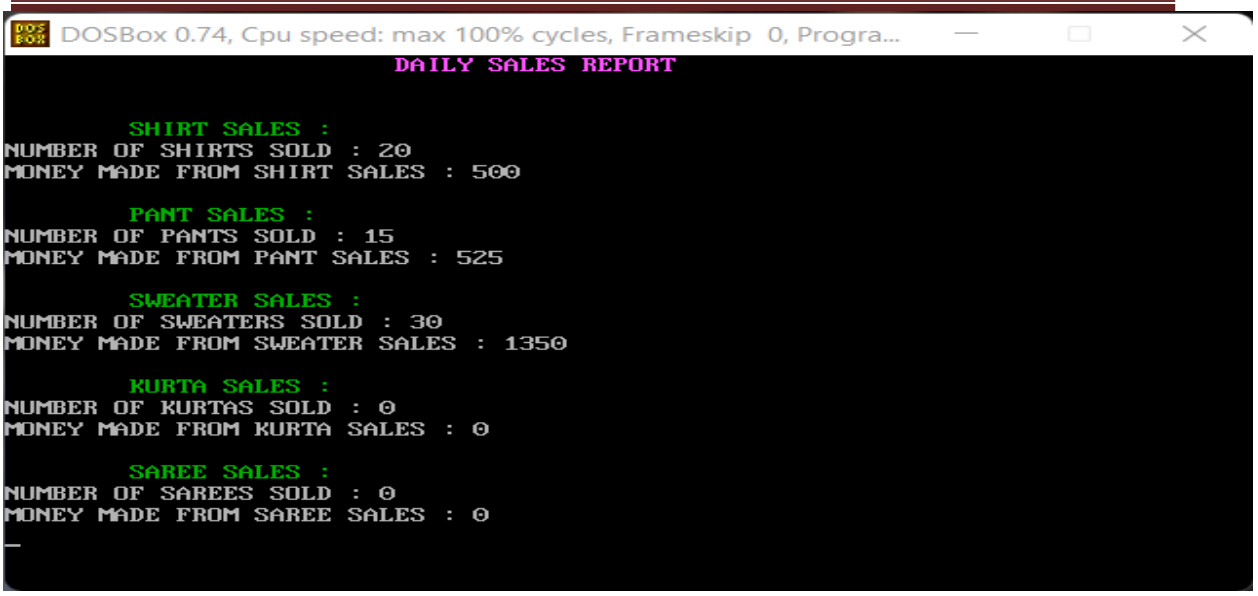

Figure 5.10-output screen 10

Now when we give option 2 in main menu which is Daily sales report, the various clothing items have been displayed along with the quantity of that specific item that have been sold.

The amount of money made from the sales of each clothing item has also been displayed.

After performing this operation we give option 6 in the main menu which is exit and we come out of the program execution.

CONCLUSION

Clothing store System Project is a simple console application designed to demonstrate the practical use of C programming language with data structure and its features . Working with This project helped me get clear understanding various concepts of data structures and structures in c programming language .I also learned how to write and analyze the algorithm for a problem statement.I also gained more experience in working with the c compiler. The main aim of this mini-project is to create an application which can be used in clothing store to make the process of updating the stock items by the shop owner as easy and hassle free as possible and by minimizing the complexity involved in managing the stock details of the store.This mini-project can be of great use to the clothing shop owners as it can save a lot of time and resources in calculating the stock details manually. This program allows the user to enter the amount of clothing items bought/purchased from the wholesaler and also enter the amount of clothing items sold to the customer.when the user enters these details,the program automatically calculates and computes the updated stocks of the various clothing items available in the shop. The clothing shop owners can also benefit from the daily sales report showed by this mini-project as it will show which clothing items are in higher demand so that the clothing shop owner can buy more products from the wholesaler and a fix a higher price in order to make more profit.

REFERENCE

1) https://www.geeksforgeeks.org/

2) https://stackoverflow.com/

3) https://www.tutorialspoint.com/

4) https://www.programiz.com/