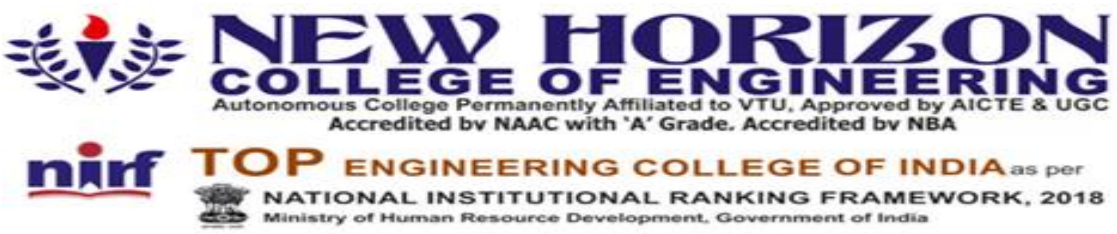

A

MINI PROJECT REPORT

ON

"MEDICAL STORE MANAGEMENT SYSTEM USING C"

Submitted in the partial fulfillment of the requirements in the 3rd semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

SHALINI R S

(1NH17IS095)

Under the guidance of

Prof. Karthiyayini

Sr. Assistant Professor

Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

(Autonomous College Permanently Affiliated to VTU, Approved by AICTE,
Accredited by NAAC with 'A' Grade & NBA)
Ring Road, Bellandur Post, Near Marathahalli,
Bengaluru-560103, INDIA

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade. Accredited by NBA
nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

CERTIFICATE

Certified that the mini project entitled "Medical Store Management System using C" carried out by Shalini R S(1NH17IS095), a bonafied student of New Horizon College of Engineering, Bengaluru, in partial fulfillment of the requirements in the III semester of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2018-2019. The project report has been approved as it satisfies the academic requirement in respect of mini project work.

Project Guide **Head of the Department**

ABSTRACT

The project titled `Medical Store Management System` is an application developed to monitor and control the activities performed in a medical store such as add medicine details, delete medicine details, view the available medicines, search for medicines, edit medicine details, purchase medicines and many more. Through the above proposed medical store management system it is possible to overcome the problems faced by the existing manual medical store management system. It helps in better utilization of resources and allows the staff to provide the services in a more systematic and efficient manner. Managing a medical store manually can be a tedious task. Therefore to make the process of managing a medical store easier and effective the above proposed medical store management system can be adapted.

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of the intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out on this mini project. I would like to take this opportunity to thank them all.

I thank the management, **Dr. Mohan Manghnani**, chairman, New Horizon Educational Institutions for providing the necessary infrastructure and creating conducive environment for effective learning.

I also record here the constant encouragement, support and facilities extended to us by **Dr. Manjunatha**, Principal, New Horizon College of Engineering, Bengaluru.

I extend sincere gratitude for the constant encouragement, support and facilities provided to us by **Dr. R J Anandhi**, Professor and Head of the Department, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru.

I sincerely acknowledge the encouragement, timely help and guidance to me by **Prof. Karthiyayini**, Sr. Assistant Professor, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru, to complete the mini project within the stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of Information Science and Engineering Department for their cooperation extended to us and our friends, who helped me directly or indirectly in the successful completion of this mini project.

TABLE OF CONTENTS

Chapter 01

INTRODUCTION

The mini project `Medical Store Management System` is an application developed for managing a medical store. The project aims at developing an application using C Language that enables a pharmacist to maintain his medicine records efficiently. The Medical Store Management System uses basic C functions and concepts to generate menus, display messages, print texts on the screen and many more. The above proposed system implements the concept of structures using C to define the medical store items along with various other concepts of C such as files and its operations, looping and branching constructs and strings and string manipulation functions.

1.1 Motivation of project

Medical store management system is an application developed for medical shops. Even in the digital era that we are living in, many pharmacies are managed manually. The existing manual medical store management system has its own set of disadvantages which includes data loss, requirement of manual labor, consumes more time and many more.

To overcome the disadvantages discussed above the application `Medical Store Management System` is developed. This system is concerned with obtaining and offering medicines, maintaining the stock records, producing receipts, creating indication about the expiry dates of the drugs and many more. It allows the pharmacist to perform various tasks such as entering the records of new medicines and retrieving their details. Through the above proposed system, the pharmacist can list all the medicines available in the medical store, issue them, maintain the records of the issued medicines and other tasks which makes the medical store management simpler and effective.

1.2 Problem Statement

To design and implement different functions in a medical store management system to overcome the disadvantages of the existing manual medical store management system.

Chapter 02

SYSTEM REQUIREMENT SPECIFICATION

Purpose: The main purpose of the medical store management system is that it reduces the manual work involved in managing a medical store. Through the above proposed system the medical store can be maintained efficiently.

The purposes of developing of a library management system are:

1. To implement different constructs of C language like loops, branching constructs, switch cases and many more.
2. To be able to solve different problem statements in C.
3. To be able to work in a group and hence develop team management skills.
4. To understand different concept of C.

The objectives of the library management system are:

1. To eliminate the errors involved in the existing manual medical store management system.
2. To simplify different processes in a pharmacy and make it efficient.
3. To reduce the paper work involved in the manual medical store management system.
4. To ensure that no data is lost by saving all the details.

Scope:

1. Ensures effective management of a medical store.
2. Any medicine details can be viewed and searched easily.
3. Minimizes the wastage of resources.
4. Helps in stock management.
5. Provides indication about the expiry date of the drugs.
6. Receipts can be generated easily.

3.1 Hardware System Configuration

Processor	-Intel Core i5
Speed	-1.8 GHz
RAM	-256 MB (min)
Hard disk	-10 GB

3.2 Software System Specifications

Operating System	-Windows 8
Programming Language	-C language
Compiler	-Code : : blocks

Chapter 03

METHODOLOGY

3.1 Algorithm

The above proposed medical store management system enables the user to perform various functions such as adding new medicine details to the medical store, deleting the older medicines, purchasing the medicines, searching and viewing medicine details and editing the medicine details. As soon as the user opens the application, it displays a menu including all the functions that can be performed in a medical store.

Based on the user's choice the particular function will be called and the task can be performed. The algorithm for the above function is as follows-

STEPS:

1) The main menu is displayed:
    a. 1 - Add Medicine
    b. 2 - Delete a Medicine
    c. 3 - Search Medicine Information
    d. 4 - Stock of Medicine in Store
    e. 5 - Purchase Medicine
    f. 6 - Change Quantity of Medicine
2) The user is asked to enter a choice.
    a. Choice 1: Add_Med function is called.
    b. Choice 2: Delete _Med function is called.
    c. Choice 3: Search_Med function is called.
    d. Choice 4: View_Med function is called.
    e. Choice 5: buy_Med function is called.
    f. Choice 6: edit_Med function is called.

ADD MEDICINE FUNCTION

In a medical store we will need to add new medicine details regularly so as to keep the store updated. When the user enters the choice 1 the add_medicines() function is called. This function asks the user for the medicine details. The algorithm for the above function is as follows:

STEPS

1) system(“cls”)
2) ask the user to enter the medicine details like medicine id, medicine name, company name, manufactured and expired date, and the quantity and price.
3) After the user enters all the details display the message medicine details saved successfully and ask if he wants to continue.
4) If yes display the mainmenu.
5) If no exit.

DELETE MEDICINE FUNCTION

The medicines that are expired are supposed to be deleted from the medical store so as to make space for the new medicines. Therefore when the user enters the choice 2 the delete_Med() function is called.

STEPS

1) system (“cls”)
2) ask the user to enter the medicine id to be deleted.
3) Check if the entered medicine is present in the store.
    a. If yes: delete the medicine and display the message medicine deleted successfully.
    b. Else display the medicine id not found.
4) Ask the user if he wants to continue, if yes display the mainmenu and if no exit.

SEARCH MEDICINE FUNCTION

If the users wants to search a particular medicine in the medical store then this function can be called by entering the choice 3.

STEPS

1) system(“cls”)
2) ask the user to enter to enter the medicine name.
    a. if the medicine name is present then display its details.
    b. Else display an error message.
3) Ask the user if he wants to continue.
    a. If yes display the mainmenu.
    b. Else exit.
4) Stop.

VIEW MEDICINE FUNCTION

If the user wants to view the medicine details f all the medicines available in a medical store the he can opt for choice 4 and the view_Med() function is called.

STEPS

1) system("cls")
2) display the details of all the medicine added to the store until the number becomes equal to 0.
3) If there are no medicines added to the store then display an error message.
4) Ask the user if he wants to continue.
    a. If yes display the mainmenu.
    b. Else exit.

BUY MEDICINE FUNCTION

If the user wants t buy medicines from the medical store then this function can be called by opting for the choice 5. Here the user can either buy the medicines using the medicine id or the medicine name. After entering the quantity of purchase the total cost will be calculated and will be displayed.

STEPS

1) system("cls")
2) ask the user if he knows the medicine id to be id.
    a. If yes enter 1:
        i. Ask the user to enter the medicine id.
        ii. If the medicine id is available then display its details.
        iii. If the medicine id is not present display an error message.
        iv. Ask the user if he wants to purchase the purchase the medicine.
        v. If yes ask the user to enter the quantity of medicine.t be purchased.
        vi. Calculate the total cost and display it to the user.
    b. If no enter any other number except 1.
    c. Ask the user to enter the medicine name and repeat the steps `i` to `vi`.

EDIT MEDICINE DETAILS.

Sometimes there are chances that the details of the medicine added are incorrect and the user might want to edit the details. Then he can opt the choice 6 and edit the details. The choice 6 calls the edit_Med function.

STEPS

1) System("cls")
2) Ask the user to enter the medicine id to be edited.
3) If the entered medicine id is available then display the options:
    a. 1 - Change Quantity

b. 2 - Change Price
c. 3 - Change Name
d. 4 - Change Company
e. 5 - Change Manufaturing Date
f. 6 - Change Expiry Date
g. 7 - Change Info

4) Ask the user to enter the choice.
    a. If choice is 1 ask the user to enter the quantity to edit.
        i. Copy the quantity entered to the structure variable m[j].quantity and display the message quantity changed successfully.
    b. If choice is 2 ask the user to enter the price to edit.
        i. Copy the price entered to the structure variable m[j].price and display the message price changed successfully.
    c. If choice is 3 ask the user to enter the name to edit.
        i. Copy the name entered to the structure variable m[j].medicneName and display the message name changed successfully.
    d. If choice is 4 ask the user to enter the company name to edit.
        i. Copy the company name entered to the structure variable m[j].Company and display the message company name changed successfully.
    e. If choice is 5 ask the user to enter the manufactured date to edit.
        i. Copy the manufactured date entered to the structure variable m[j].Mfg_Date and display the message manufactured date changed successfully.
    f. If choice is 6 ask the user to enter the expiry date to edit.
        i. Copy the expiry date entered to the structure variable m[j].Exp_Date and display the message expiry date changed successfully.
    g. If choice is 7 ask the user to enter the information to edit.
        i. Copy the information entered to the structure variable m[j].info and display the message information changed successfully.

3.3 Code and implementation

```
#include<stdio.h>
 #include<conio.h>
 #include<string.h>
 #include<ctype.h>
 struct Medicine{
   int id,price,quantity;
   char medicneName[100],Company[100],Mfg_Date[11],Exp_Date[11],info[5000];
  }m[100];
 void buy_Med(int number);
  void view_Med(int number);
 void search_Med(int number);
 void Add_Med(int number,struct Medicine m[]);
 void Delete_Med(int number);
 void edit_Med(int number);
 main()
 {
   int i,j,choice,number=0,c;
   for(i=0;i<100;i++)
   {
    m[i].id=0;
   m[i].price=0;
   m[i].quantity=0;
   strcpy(m[i].Mfg_Date,"");
  strcpy(m[i].Exp_Date,"");
   strcpy(m[i].medicneName,"");
   strcpy(m[i].Company,"");
   strcpy(m[i].info,"");
   }
   m[0].id=1;
   m[0].price=120;
   m[0].quantity=30;
   strcpy(m[0].Mfg_Date,"23-04-2016");
  strcpy(m[0].Exp_Date,"24-04-2020");
```

```
  strcpy(m[0].medicneName,"Paracetmol");
  strcpy(m[0].Company,"ABCD");
  strcpy(m[0].info,"Good Medicine for Fever");
  do{
      system("cls");
  printf("Enter\n1 - Add Medicine\n2 - Delete a Medicine\n3 - Search Medicine Information\n4 - Stock of Medicine in Store\n5 - Purchase Medicine\n6 - Change Quantity of Medicine\n");
  scanf("%d",&choice);
  switch(choice)
  {
   case 1:
   {
    ++number;
      Add_Med(number,m);
    break;
   }
    case 2:
   {
       Delete_Med(number+1);
     break;
    }
     case 3:
     {
      search_Med(number+1);
      break;
     }
     case 4:
              {
        view_Med(number+1);
      break;
     }
     case 5:
     {
```

```
          buy_Med(number+1);
       break;
      }
      case 6:
      {
       edit_Med(number+1);
       break;
      }
   }
    printf("To Continue with other Options Enter 1 Else any other number\n");
    scanf("%d",&c);
   }while(c==1);

}
void buy_Med(int number)
{
    system("cls");
 int id,check,j,quantity,flag=0;
 char name[1000];
 printf("Enter 1 if you know ID\nelse any other number to enter Name of Medicine\n");
 scanf("%d",&check);
 if(check==1)
 {
  printf("Enter Id of Medicine to be purchased\n");
  scanf("%d",&id);
  for(j=0;j<number;j++)
  {
   if(m[j].id==id)
   {
    flag=1;
    int c;
    printf("Details of Medicine are\n");
    printf("Name%s\nPrice=%d\nAvailable                Quantity=%d\nCompany=%s\nMfg
Date=%s\nExp
```

```
Date=%s\n",m[j].medicneName,m[j].price,m[j].quantity,m[j].Company,m[j].Mfg_Date,m
[j].Exp_Date);
      if(strcmp(m[j].info,"")==0)
      {
       printf("Medicine Not Available\n");
      }
      else
      {
       printf("Medicine Info=%s\n",m[j].info);
      }
    printf("Do you want to purchase %s \nEnter 1 if yes else any other
number\n",m[j].medicneName);
     scanf("%d",&c);
     if(c==1)
     {
      printf("Enter Quantity\n");
      scanf("%d",&quantity);
      if(m[j].quantity>quantity)
      {
       printf("Total Price=%d\n",quantity*m[j].price);
      }
      else{
       printf("Please Enter quantity below Available Quantity\n");
      }
     }
     break;
    }
   }
   if(flag==0)
   {
    printf("Entered medicine Id Not Found\n");
   }
  }
```

```
 Else

 {
  printf("Enter the Name to be searched and Purchased\n");
  gets(name);
  for(j=0;j<number;j++)
  {
   if(strcmp(m[j].medicneName,name)==0)
   {
    flag=1;
    int c;
    printf("the details of Medicine are\n");
    printf("Name=%s\nPrice=%d\nAvailable Quantity=%d\nCompany=%s\nMfg
Date=%s\nExp
Date=%s\n",m[j].medicneName,m[j].price,m[j].quantity,m[j].Company,m[j].Mfg_Date,m
[j].Exp_Date);
     if(strcmp(m[j].info,"")==0)
     {
      printf("Medicine Info Not Available\n");
     }
     else
     {
      printf("Medicine Info=%s\n",m[j].info);
     }
    printf("Do you want to purchase %s \nEnter 1 if yes else any other
number\n",m[j].medicneName);
    scanf("%d",&c);
    if(c==1)
    {
     printf("Enter Quantity\n");
     scanf("%d",&quantity);
     if(m[j].quantity>quantity)
     {
      printf("Total Price=%d\n",quantity*m[j].price);
```

```
      }
      else{

      printf("Please Enter quantity below Available Quantity\n");
      }
     }
     break;
    }
   }
  if(flag==0)
  {
   printf("Entered medicine Name Not Found\n");
  }
 }
}
 void search_Med(int number)
 {
    system("cls");
  int j,flag=0;
 char name[100];
 printf("Enter Name of the medicine you want to search\n");
 scanf("%s",name);
 for(j=0;j<number;j++)
 {
  if(strcmp(m[j].medicneName,name)==0)
  {
   flag=1;
   printf("the details of Medicine are\n");
   printf("Name=%s\nPrice=%d\nAvailable Quantity=%d\nCompany=%s\nMfg
Date=%s\nExp
Date=%s\n",m[j].medicneName,m[j].price,m[j].quantity,m[j].Company,m[j].Mfg_Date,m
[j].Exp_Date);
   if(strcmp(m[j].info,"")!=0)
   {
```

```
     printf("Review=%s\n",m[j].info);
    }
    else

    {
     printf("Info Not Available\n");
    }
   }
  }
  if(flag==0)
  {
   printf("Entered medicine Name Not Found\n");
  }
 }
 void view_Med(int number)
 {
     system("cls");
  int j;
  if(number!=0)
  {
      printf("All Available medicines are\n");
   for(j=0;j<number;j++)
   {
    if(m[j].id!=0){
    printf("Id=%d\nName=%s\t\tPrice=%d\tAvailable Quantity=%d\nCompany=%s\t\tMfg
Date=%s\tExp
Date=%s\n",m[j].id,m[j].medicneName,m[j].price,m[j].quantity,m[j].Company,m[j].Mfg
_Date,m[j].Exp_Date);
    if(strcmp(m[j].info,"")!=0)
    {
     printf("Review=%s\n",m[j].info);
    }
    else
    {
```

```
    printf("Review Not Available\n");
   }
    }
  }

 }
 else{
  printf("No Medicines Available\n");
 }
}
void Add_Med(int number,struct Medicine m[])
{
   system("cls");
 char name[100];
 printf("Enter Medicine Id\n");
 scanf("%d",&(m[number].id));
 printf("Enter Medicine name\n");
 scanf("%s",name);
 strcpy(m[number].medicneName,name);
 printf("Enter the Company Name\n");
 scanf("%s",(m[number].Company));
 printf("Enter Manufactured Date\n");
 scanf("%s",(m[number].Mfg_Date));
 printf("Enter Expiry Date\n");
 scanf("%s",(m[number].Exp_Date));
 printf("Enter Quantity\n");
 scanf("%d",&(m[number].quantity));
 printf("Enter Price\n");
 scanf("%d",&(m[number].price));
 strcpy(m[number].info,"");
 printf("Medicine with id %d Added Successfully\n",m[number].id);
}
void Delete_Med(int number)
{
```

```
    system("cls");
 int id,j,flag=0,num;
 printf("Enter Id to be deleted\n");
 scanf("%d",&id);
 for(j=0;j<number;j++)

 {
  if(m[j].id==id)
  {
   flag=1;
   m[j].id=0;
   m[j].price=0;
   m[j].quantity=0;
   strcpy(m[j].medicneName,"");
   strcpy(m[j].Company,"");
   strcpy(m[j].Mfg_Date,"");
   strcpy(m[j].Exp_Date,"");
   strcpy(m[j].info,"");
   num=j;
   break;
  }
 }
 if(flag==1)
 {
  printf("Medicine with %d is Deleted Successfully\n",id);
 }
}

void edit_Med(int number)
{
    system("cls");
 int id,quantity,choice,c,j;
 printf("Enter id to edit\n");
 scanf("%d",&id);
```

```
 for(j=0;j<number;j++)
 {
  if(m[j].id==id && m[j].id!=0)
  {
   do
   {

      printf("Enter\n1 - Change Quantity\n2 - Change Price\n3 - Change Name\n4 - Change Company\n5 - Change Manufaturing Date\n6 - Change Expiry Date\n7 - Change Info\n");
      scanf("%d",&choice);
      if(choice==1)
      {
       int quantity;
       printf("Enter Quantity to edit\n");
       scanf("%d",&quantity);
       m[j].quantity=quantity;
       printf("Quantity changed Successfully\n");
      }
      if(choice==2)
      {
       int price;
       printf("Enter Price to be editted\n");
       scanf("%d",&price);
       m[j].price=price;
       printf("Price changed Successfully\n");
      }
      if(choice==3)
      {
       char name[100];
       printf("Enter Name to be edited\n");
       scanf("%s",name);
       strcpy(m[j].medicneName,name);
       printf("Medicine Name changed Successfully\n");
```

```
}
if(choice==4)
{
 char company[100];
 printf("Enter company to be changed\n");
 scanf("%s",company);
 strcpy(m[j].Company,company);

 printf("Company changed Successfully\n");
}
if(choice==5)
{
 char mfg[11];
 printf("Enter Manufacturing date to be editted\n");
 scanf("%s",mfg);
 strcpy(m[j].Mfg_Date,mfg);
 printf("Manufacturing Date changed Successfully\n");
}
if(choice==6)
{
 char exp[11];
 printf("Enter Expiry date to be changed\n");
 scanf("%s",exp);
 strcpy(m[j].Exp_Date,exp);
 printf("Expiry Date changed Successfully\n");
}
if(choice==7)
{
 char info[100];
 printf("Enter Info to be changed(Less than 100 Characters)\n");
 scanf("%s",info);
 strcpy(m[j].info,info);
 printf("Info changed Successfully\n");
}
```

```
    if(choice<=0 && choice>7)
    {
     printf("Enter valid Choice\n");
    }
    printf("Enter 1 to Change other Details Else any other number\n");
    scanf("%d",&c);
  }while(c==1);
  break; }}}
```

Chapter 04

RESULTS AND DISCUSSION

4.1 Output Snapshots

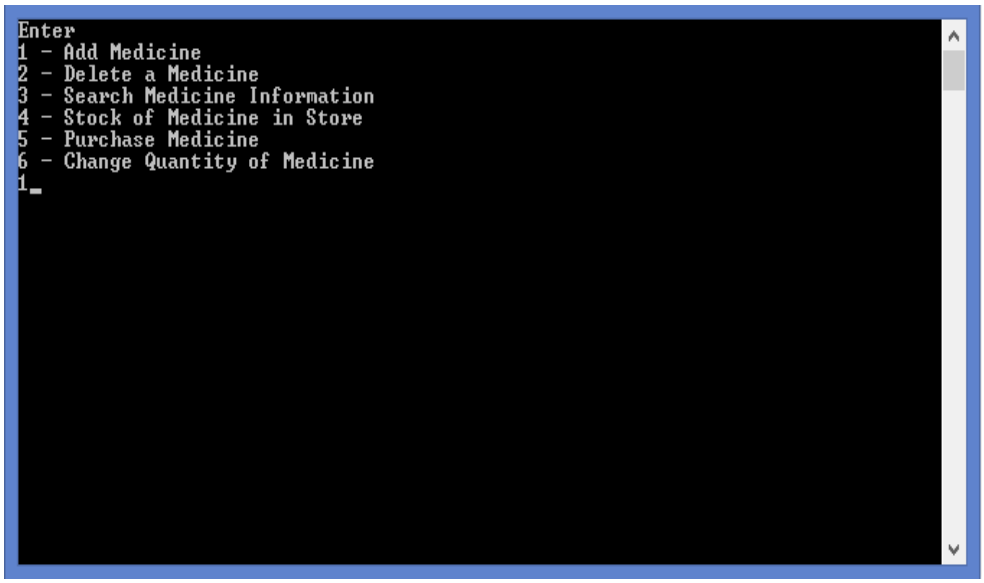

Fig 4.1 mainmenu displayed and the user's choice is 1

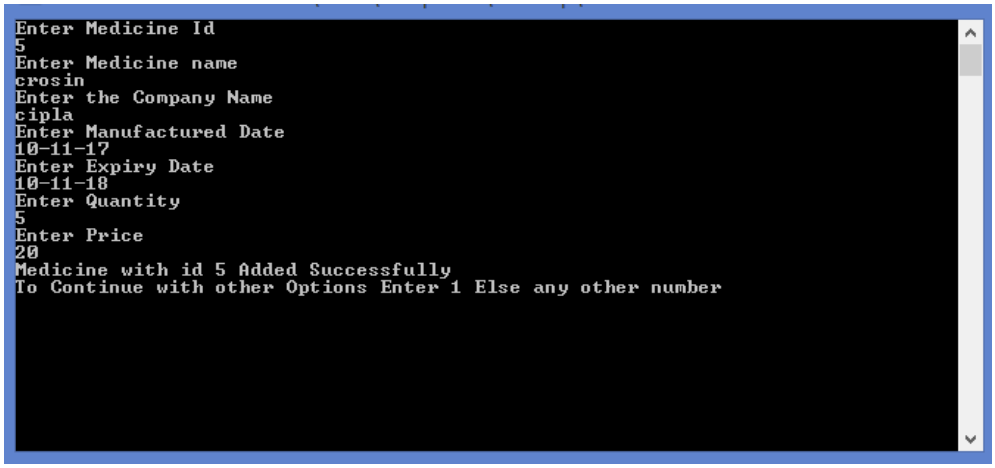

Fig 4.2 the medicine details are entered by the user

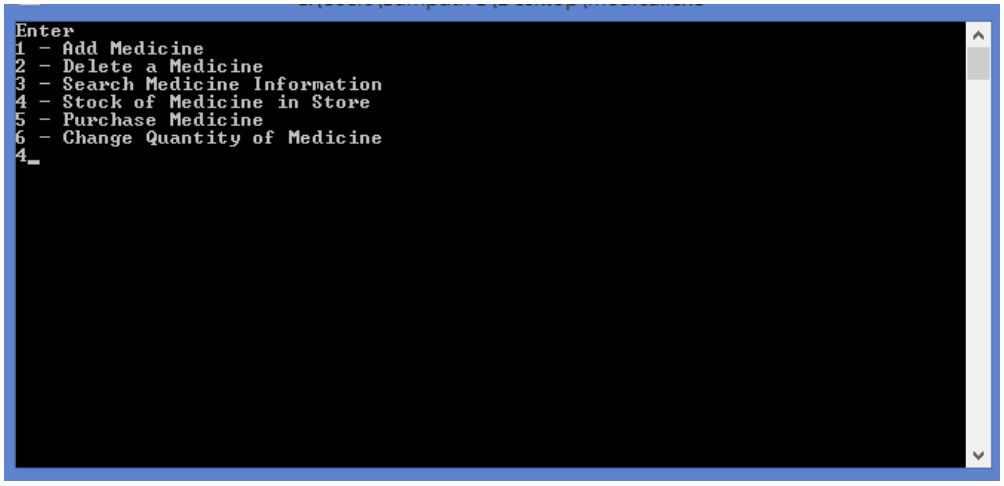

Fig 4.3 user chooses the option 4

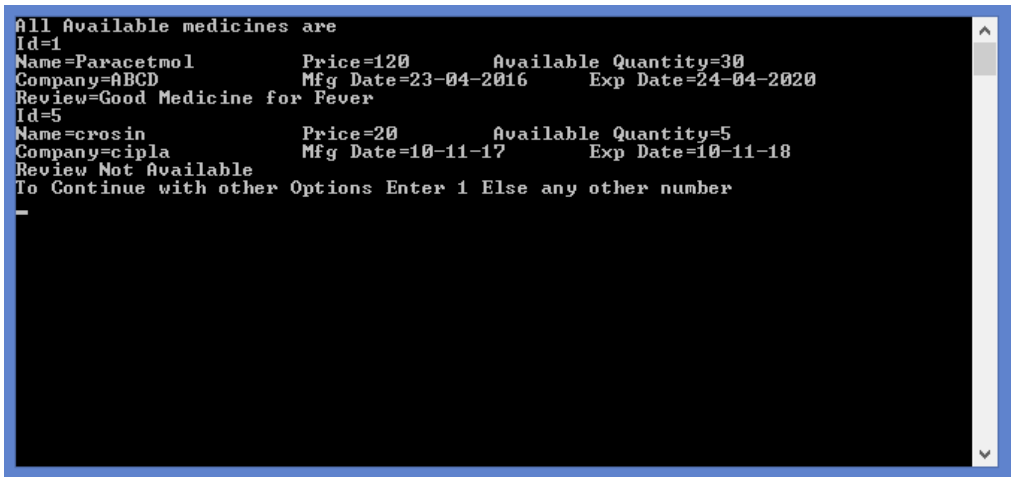

Fig 4.4 the medicine details are added

Fig 4.5 the user gives the option 2 and delete medicine is called

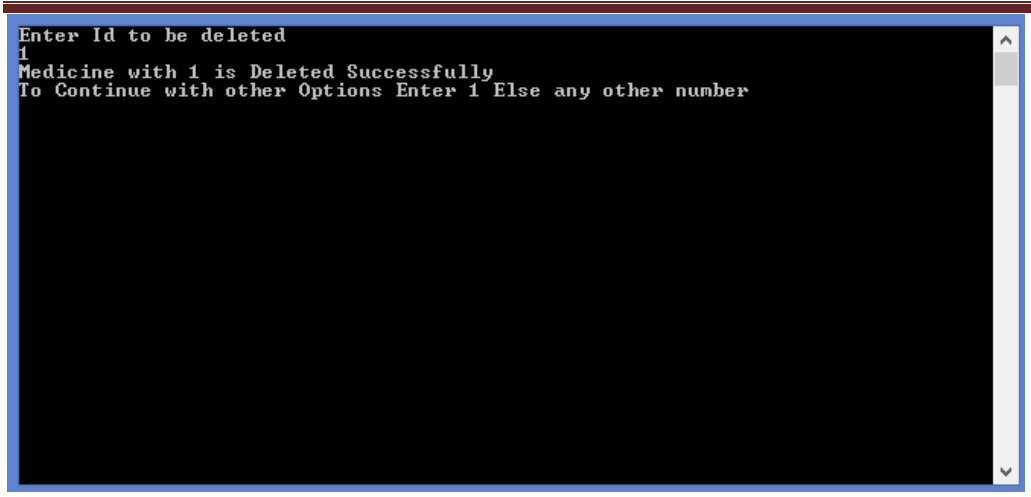

Fig 4.6 the medicine id to be deleted is given

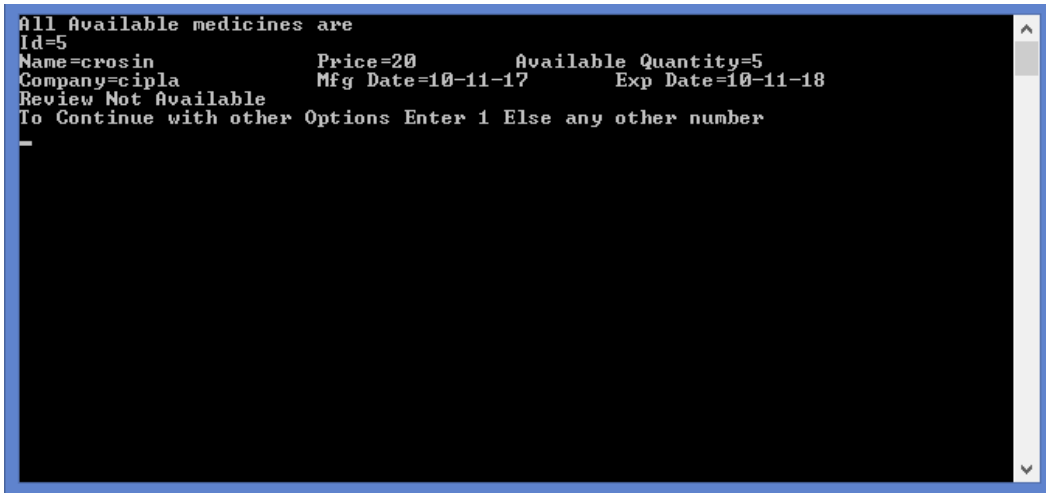

Fig 4.7 the medicines with the given id is deleted

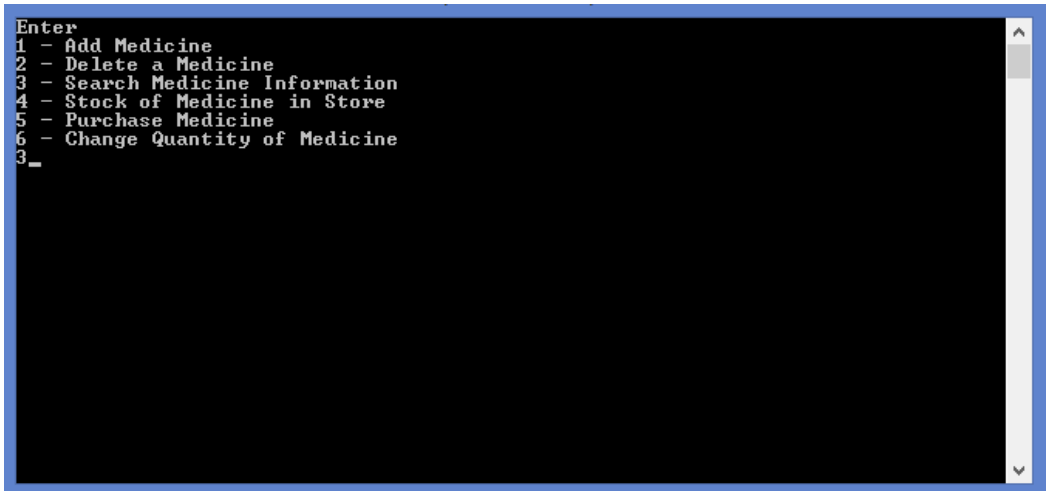

Fig 4.8 the user chooses the option 3

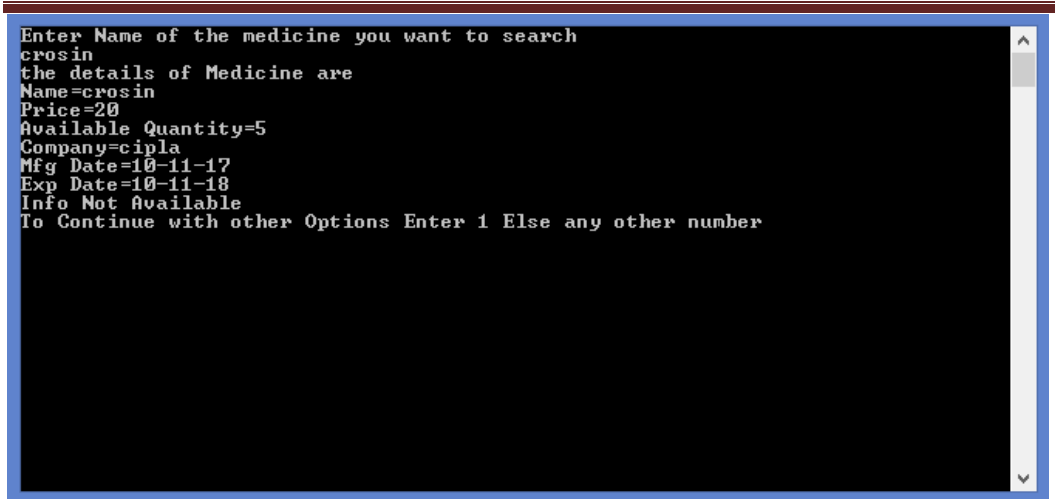

Fig 4.9 the medicine with the given name is searched

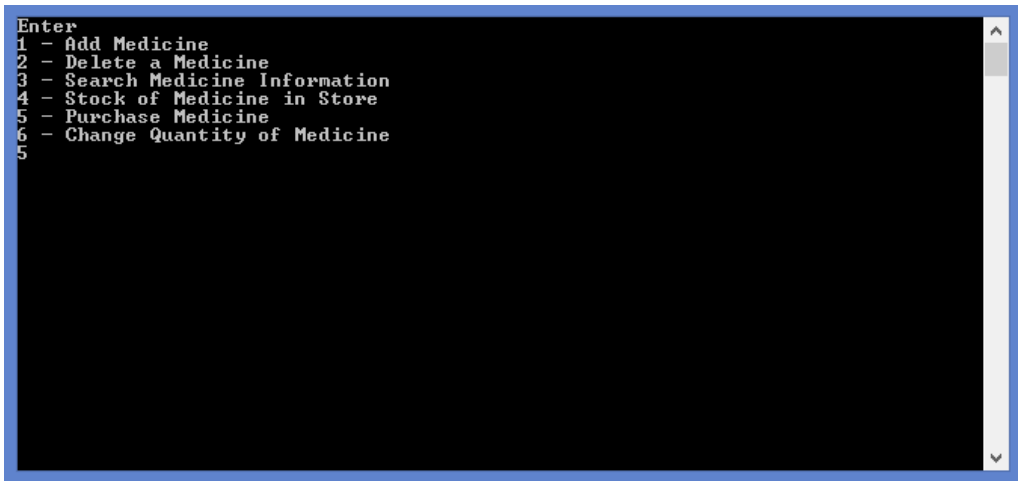

Fig 4.10 the user chooses the option 5

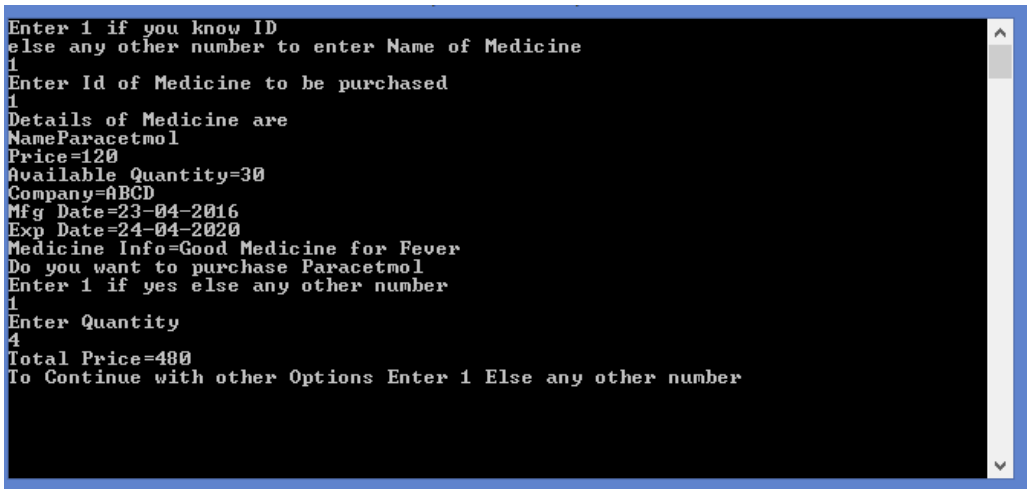

Fig 4.11 the medicine with the given id is purchased

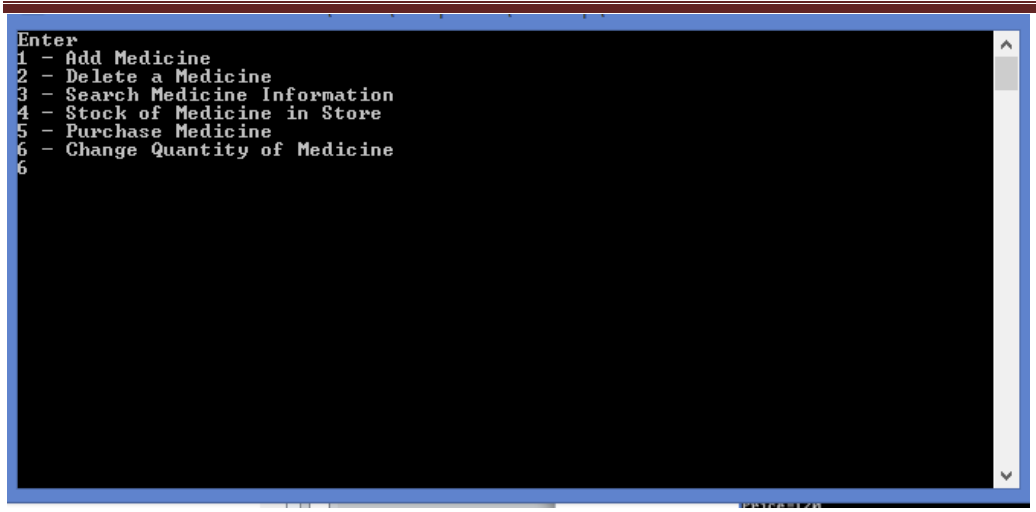

Fig 4.12 the user chooses the option 6

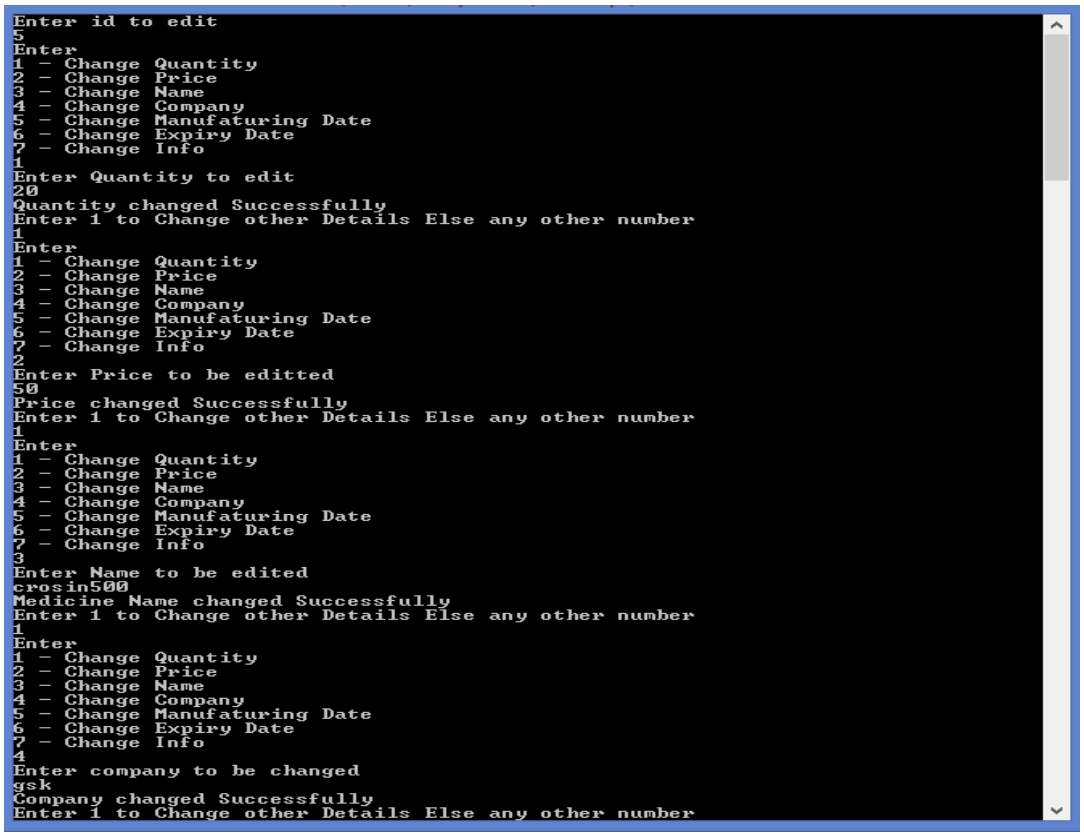

Fig 4.13 the quqntity, price, name, company of the medicine with the given id is changed

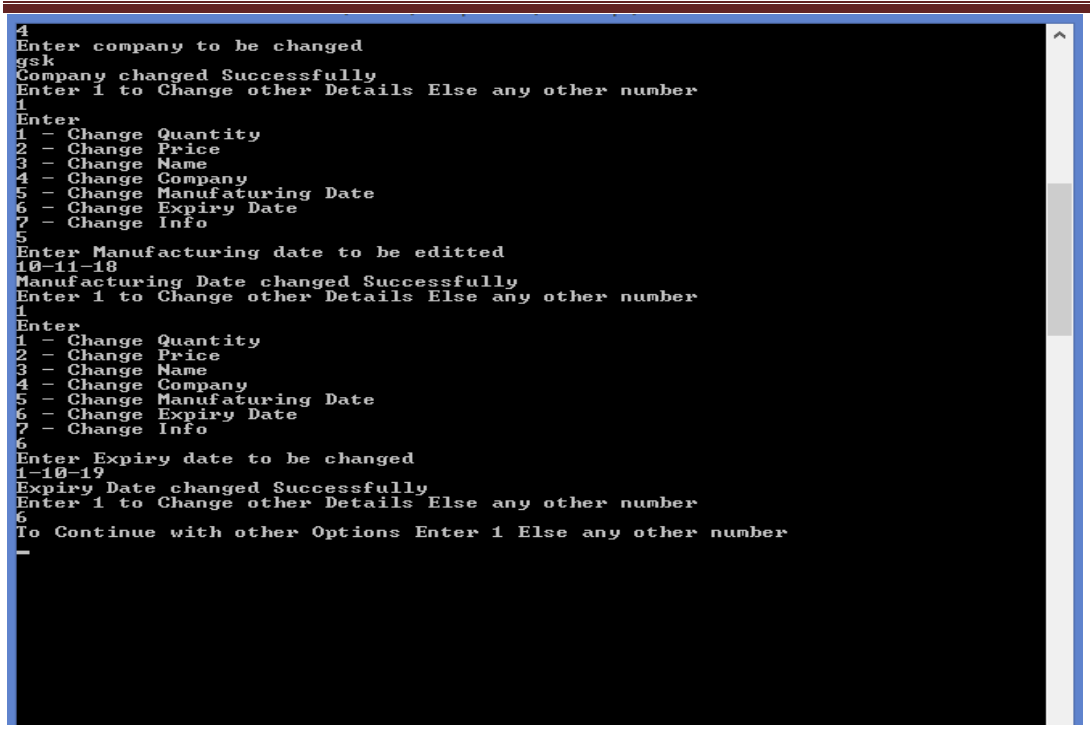

Fig 4.14 the manufacturing date and the expiry date of the medicine with the given id is changed

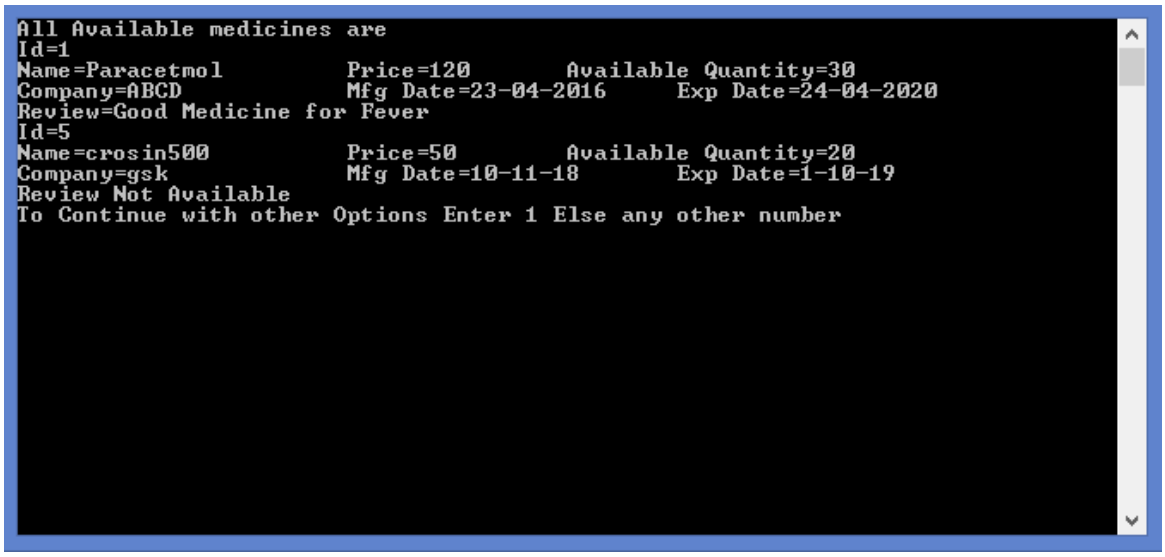

Fig 4.15 the list of all the available medicines is displayed

Chapter 05

CONCLUSION

As the existing manual medical store management system has its own set of disadvantages including lower efficiency, higher time consumption and requirement of higher manual labor. The above mentioned disadvantages can be eliminated by using the above proposed automated medical store management system. The medical store management system proves to be an user friendly application and has various advantages over the manual medical store management system which includes quick, easy and flexible generation of the medicine details, modification of the database is quick and effective, provides backup and restore facilities, safe storage of the information and many more.