STORE GOODS RECORD KEEPING AND BILLING UTILITY

PROJECT ON C PROGRAMMING

SUBMITTED BY

ABHIJEET KUMAR

MCA SEC 2

ENROLL NO. -
00435304419

INTRODUCTION

The "store goods record keeping and billing utility" aims at automating the goods management and billing for any small scale departmental store. This utility provides the user to add/edit/delete/view records of goods as well as maintain and publish billing records for the customer at the time of shopping on a day to day basis.

OBJECTIVE

This project will serve the following objectives:

1. Add and maintain records of available products.
2. Add and maintain description of new products.
3. Add and maintain new entered category of products
4. Provides a convenient solution of billing pattern.
5. Make an easy to use environment for users and customers.
6. Make the billing procedures efficient

PROBLEM OF EXISTING SYSTEM

The existing system here refers to the manual billing system operated by the individual. The existing system has the following anomalies:

1. Inability of modification of data: The managing of huge data effectively and efficiently for efficient results, storing the details of the consumers etc. in such a way that the database can be modified as not possible in the current system.

2. Not user friendly: The existing system is not user friendly because the retrieval and storing of data is slow and data is not maintained

efficiently.

3. Difficulty in reports generating: Either no reports generating in a current system or they are generated with great difficulty reports take time to generate in the current system.

4. Manual operator control: Manual operator control is there and lead to a lot of chaos and errors.

5. Lot of paperwork: Existing system requires lot of paper work and even a small transaction requires many papers fill. Moreover, any unnatural cause (such as fire in the organization) can destroy all data of the organization. Loss of even a single paper led to difficult situation because all the papers are interrelated.

6. Inability of sharing the data: Data cannot be shared in the existing system. This means that no two persons can use the same data in existing system. Also the two departments in an organization cannot interact with each other without the actual movement of data.

7. No support in decision-making: Existing system does not support managerial decision-making.

8. No support in strategic competitive advantage: Existing system do not support strategic competitive advantages.

CHARACTERISTICS OF PROPOSED SYSTEM

The automated store records keeping and billing system tackles the problems of existing system in following ways:

1. Easiness in modification of data: The proposed system provides managing of huge data effectively and efficiently for efficient results, storing the details of the customers, employees etc. in such a way that the database can be modified.

2. User friendly: The proposed system is user friendly because the retrieval and storing of data is fast and data is maintained efficiently. Moreover, the graphical user interface is provided in the proposed

system, which provides user to deal with the system very easily.

3. Reports are easily generated: Reports can be easily generated in a proposed system. So any type of reports can be generated in a proposed system, which helps the managers in a decisions-making activity.

4. Sharing the data is possible: Data can be shared in proposed system. This means that two or more persons can use the same data in existing system provided that they have right to access that data. Also the two or more departments in an organization can easily interact with each other without the actual movement of data.

5. No or very few paperwork: The proposed system either does not require paper work or very few paper work is required. All the data is feted into the computer immediately and various bills and reports can be generated through computers. Since all the data is kept in a database no data of the organization can be destroyed. Moreover, work becomes very easy because there is no need to keep data on papers.

6. Support strategic competitive advantage: Proposed system supports strategic competitive advantages. Since the proposed systems provide easiness in reports generating it will provide strategic advantages among competitors.

7. Computer operator control: Computer operator control will be there no errors. Moreover, storing and retrieving of information is easy. So work can be done speedily and in time.

MODULE DESCRIPTION

1. **DISPLAY FUNCTIONS:**
   This module consists of functions which are used for the purpose of display manipulations.

   a. **void curser(int);**
      This is a function for cursor movement. It is called whenever we wish to navigate through the menu. It takes one argument of integer type.

   b. **void dbill();**
      It is a function which displays the bill window where the details of the customer's purchase are shown in the format of an invoice.

   c. **void highlight(int,int);**
      It is a subsidiary function for the function curser(), it helps in navigating in between different menus. It takes two arguments of integer type namely no and count.

   d. **void d_mainmenu();**
      This function displays the first window. It contains the primary main menu options for the user to choose from.
      It includes: Calculate Bill, Add Goods, Edit Goods, Display All, Search, Delete Goods, Exit.

   e. **void display(rec*,int,int);**
      It is a function to display in screen items for printing the things purchased. It takes three arguments of type structure rec, integer and integer respectively.

   f. **void window(int,int,int,int);**
      It is a display function which the displays the window design for the main page of the system. It generates the designing pattern for the design box.

g. **void dis_con();**
   It is a function to display window for item display (SN, Item Name, Item Code, Rate, Quantity).

h. **void d_search();**
   This functions displays the window for the search query of the user. The search is based on the unique item code for the product.

2. **CHECKING FUNCTIONS (Validating the data):**
   These functions are used to check the validity of data before performing any operations that affect the database. It checks for correct input values before altering the database.

   a. **void c_code(char[]);**
      It is a function to check for the availability of unique code id for the specific item. If two or more inputs share the same code for the item, it prompts the user for new input.

   b. **int check(char[]);**
      It is a function to check validity of unique code id input by the user while editing and deleting the item.

3. **MAIN MENU FUNCTIONS:**
   This module describes about the main operational functions for the billing system.

   a. **void bill();**
      This function is used to calculate the bill by accepting the required inputs. It is used in adjacent with the function dbill().

   b. **void edit();**

This function is used to edit the records of the items in the database. The edit() function can be used to edit name of the item, code of the item, rate of the item and quantity of the item.

c. **void add();**
This function is used to add new records to the database. The new record info includes name, code, rate and quantity of the new item.

d. **void del();**
This function is used to delete records from the database. It asks for the unique code id of the item to be deleted.

4. **DISPLAY SUBMENU FUNCTIONS:**

These functions are used in association with the main menu functions.

a. **void d_code();**
This function is used to display the records in the database based on unique id of the code.

b. **void d_rate();**
This function is used to display the records based on the lower range and upper range of the rate of the product.

c. **void d_quan();**
This function is used to display the records based on the weightage of quantity of the particular product.

d. **void d_all();**
This function is used to display all the records stored in the database.

CODE

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>
#include<windows.h>

#define ANS 15
#define ACS 4
 COORD coord={0,0}; // this is global variable
                          //center of axis is set to the top left cornor of the screen
void gotoxy(int x,int y)
{
   coord.X=x;
   coord.Y=y;
```

```
   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),coord);
}
/*declaration of checking functions*/
void c_code(char[]);
int check(char[]);

/*structure declaration*/
typedef struct{
char name[ANS],code[ACS];
float rate;
int quantity;
}rec;
rec item;

/*declaration of display functions*/
void curser(int);
void dbill();
void d_mainmenu();
void display(rec *,int,int);
void window(int,int,int,int);
void dis_con();
void d_search();
void highlight(int,int);

/*declaration of main menu functions*/
```

```
void bill() ;
void edit();
void add();
void del();
void exit();

/*declaration of display submenu functions*/
void d_code();
void d_rate();
void d_quan();
void d_all();

/*start of main*/
int main()
{
 d_mainmenu();
 return 0;
}



void d_mainmenu()
{
 int i;
 char ch;
```

```
 const char *menu[]={"   Calculate Bill","   Add Goods","   Edit Goods","   Display All ","   Search", "   Delete Goods","   Exit"};
 system("cls");
 //textbackground(11);
 //textcolor(0);
 //_setcursortype(_NOCURSOR);
 window(25,50,20,32);
 gotoxy(33,18);printf("MAIN MENU");
 for (i=0;i<=6;i++){
  gotoxy(30,22+i+1);printf("%s\n\n\n",menu[i]);}
  curser(7);
 }

void d_search()
{
 char ch;
 int i;
    const char *menu[]={"   By Code","   By Rate","   By Quantity","   Back to main menu"};
 system("cls");
 //textbackground(11);
 //textcolor(0);
 window(25,50,20,32);
 gotoxy(33,18);printf("SEARCH MENU");
 for (i=0;i<=3;i++){
 gotoxy(30,22+i+1);printf("%s\n\n\n",menu[i]);}
```

```
 curser(4);
}

/*function for cursor movement*/
void curser(int no)
{
    int count=1;
    char ch='0';
    gotoxy(30,23);
 while(1){
  switch(ch){
   case 80:
    count++;
    if (count==no+1) count=1;
    break;
   case 72:
    count--;
    if(count==0) count=no;
    break;
  }
  highlight(no,count);
  ch=getch();
  if(ch=='\r'){
   if(no==7){
    if (count==1) bill() ;
```

```
    else if(count==2) add();
    else if(count==3) edit();
    else if (count==4) d_all();
    else if (count==5) d_search();
    else if (count==6) del();
    else   exit(0);
   }
   if(no==4){
    if (count==1) d_code();
    else if (count==2)d_rate();
    else if (count==3) d_quan();
    else d_mainmenu();
   }
  }
 }
}

void highlight(int no,int count)
{
 if (no==4){
  //textbackground(11);
  //textcolor(0);
  gotoxy(30,23);printf("   By Code          ");
  gotoxy(30,24);printf("   By Rate          ");
  gotoxy(30,25);printf("   By Quantity      ");
```

```
 gotoxy(30,26);printf("   Back to main menu");
 //textcolor(0);
 //textbackground(2);
 switch (count)
 {
  case 1:
   gotoxy(30,23);
   printf(" - By Code          ");
   break;
  case 2:
   gotoxy(30,24);
   printf(" - By Rate          ");
   break;
  case 3:
   gotoxy(30,25);
   printf(" - By Quantity      ");
   break;
  case 4:
   gotoxy(30,26);
   printf(" - Back to main menu");
   break;
 }
}

if(no==7){
```

```
//textbackground(11);
//textcolor(0);
gotoxy (30,23);printf("   Calculate Bill ");
gotoxy (30,24);printf("   Add Goods      ");
gotoxy (30,25);printf("   Edit Goods     ");
gotoxy (30,26);printf("   Display All    ");
gotoxy (30,27);printf("   Search         ");
gotoxy (30,28);printf("   Delete Goods   ");
gotoxy (30,29);printf("   Exit           ");
//textcolor(0);
//textbackground(2);
switch(count){
 case 1:
  gotoxy (30,23);
  printf(" - Calculate Bill ");
  break;
 case 2:
  gotoxy (30,24);
  printf(" - Add Goods      ");
  break;
 case 3:
  gotoxy (30,25);
  printf(" - Edit Goods     ");
  break;
 case 4:
```

```
    gotoxy (30,26);
    printf(" - Display All    ");
    break;
   case 5:
    gotoxy (30,27);
    printf(" - Search         ");
    break;
   case 6:
    gotoxy (30,28);
    printf(" - Delete Goods   ");
    break;
   case 7:
    gotoxy (30,29);
    printf(" - Exit           ");
    break;
  }
 }
}

void bill()
{
  char x[4]={0};
  int j=29,q=0,size=0,i=1;
  float total=0,gtotal=0;
  FILE *file;
```

```
file=fopen("record.txt","r+b");
rewind(file);
system("cls");
dbill();
gotoxy(26,15);printf("enter  \"end\" to finish input");
while(1){
gotoxy(25,18);printf("                    ");
gotoxy(25,19);printf("                    ");
gotoxy(25,18);printf("enter item code:");
scanf("%s",x);
if(strcmp(x,"end")==0)
 break;
gotoxy(25,19);printf("enter quantity:");
scanf("%d",&q);
rewind(file);
while(fread(&item,sizeof(item),1,file)){
 if((strcmp(item.code,x)==0)){
  total=item.rate*q;
  gotoxy(11,j);printf("%4d",i);
  printf("%9s",item.name);
  printf("%13d",q);
  printf("%15.2f",item.rate);
  printf("%13.2f",total);
  gtotal=gtotal+total;
  size=sizeof(item);
```

```
    item.quantity=item.quantity-q;
    j+=2; i++;
    fseek(file,-size,SEEK_CUR);fwrite(&item,sizeof(item),1,file);
    break;
   }
    }
 }
 if(gtotal!=0){
  gotoxy(30,j+5);
  printf("TOTAL AMOUNT = NRs. %6.2f",gtotal);
 }
 fclose(file);
 getch();
 d_mainmenu();
}
 /*function to display bill window*/
void dbill()
{
 int i;
 gotoxy(20,10);
 //;
 for (i=1;i<=10;i++)
  printf("*");
 printf(" * FASHION WEAR * ");
 for (i=1;i<=10;i++)
```

```
        printf("*");
 printf("\n\n");
 gotoxy(30,11);printf("Departmental Store");
 //textcolor(1);
 gotoxy(32,25);printf("CUSTOMER'S BILL") ;
 //textcolor(8);
 gotoxy(13,27);printf("SN.   Item Name     Quantity     Rate          Total");

}
/*function to add records*/
void add ()
{
 FILE *file;
 char y[ACS],x[12];
 system("cls");
 //textbackground(11);
 //textcolor(0);
 gotoxy(25,25);printf("Enter new record(Y/N)?");
 while(toupper(getche())=='Y'){
  system("cls");
  file=fopen("record.txt","ab");
  c_code(y);
  strcpy(item.code,y);
  gotoxy(22,28);printf("Enter rate of the item:");
  scanf("%f",&item.rate);
```

```
  gotoxy(22,30);printf("Enter quantity of the item:");
  scanf("%d",&item.quantity);
  gotoxy(22,32);printf("Enter name of the item:");
  scanf("%s",item.name);
  fseek(file,0,SEEK_END);fwrite(&item,sizeof(item),1,file);
  fclose(file);
  gotoxy(22,34);printf("Enter new record(Y/N)?");

   }
   d_mainmenu();
}

 /*function to check availability of code*/
void c_code(char y[])
{
 int flag;
 FILE *file;
 file=fopen("record.txt","rb");
 while(1){
  system("cls");
  window(20,58,23,36);
  gotoxy(32,18);printf(" ADD ARTICLES ")  ;
  flag=1;
  rewind(file);
  gotoxy(22,25);printf("Enter new code of the article:");
```

```
 scanf(" %[^\n]",y);
 while(fread(&item,sizeof(item),1,file)==1){
  if (strcmp(y,item.code)==0){
   flag=0;
   gotoxy(26,30);printf("code already exists");
   gotoxy(29,32);printf("enter again");getch();
   break;
  }
 }
 if (flag==1)
 break;
}
}

/*function for editing*/
void edit()
{
 int flag=0,choice;
 char x[ACS],y[ACS];
 FILE *file;
 int size;
 system("cls");
 //textcolor(0);
 //textbackground(11);
 window(20,63,20,46);
```

```
gotoxy(35,18);printf("EDIT RECORDS");
;
gotoxy(25,23);printf("enter item code: ");
scanf("%s",x);
flag=check(x);
if(flag==0){
 file=fopen("record.txt","r+b");
 rewind(file);
 while (fread(&item,sizeof (item),1,file)){
  if(strcmp(item.code,x)==0){
   //textcolor(0);
   gotoxy(25,27);printf("name       = %s",item.name);
   gotoxy(25,28);printf("code       = %s",item.code);
   gotoxy(25,29);printf("rate       = %g",item.rate);
   gotoxy(25,30);printf("quantity   = %d",item.quantity);
   gotoxy(25,32); printf("do you want to edit this record(y/n):");
   fflush(file);
   if(toupper(getche())=='Y'){
    //textcolor(0);
    gotoxy(25,34); printf("1- edit name ");
    gotoxy(25,35); printf("2- edit code ");
    gotoxy(25,36); printf("3- edit rate ");
    gotoxy(25,37); printf("4- edit quantity ");
    gotoxy(25,39);  ; printf(" enter your choice(1, 2, 3, 4) ");
    scanf("%d",&choice);
```

```
switch(choice){
 case 1:
  system("cls");
  window(23,48,20,40);
  gotoxy(35,18);printf("EDIT RECORDS");
  gotoxy(25,24); printf(" enter new name: ");
  scanf("%s",item.name);
  size=sizeof(item);
  fseek(file,-size,SEEK_CUR);fwrite(&item,sizeof(item),1,file);
  break;
 case 2:
  system("cls");
  window(23,65,20,40);
  gotoxy(35,18);printf("EDIT RECORDS");
  gotoxy(25,24);printf("enter new code:");
  c_code(y);
  strcpy(item.code,y);
  size=sizeof(item);
  fseek(file,-size,SEEK_CUR);fwrite(&item,sizeof(item),1,file);
  break;
 case 3:
  system("cls");
  window(23,65,20,40);
  gotoxy(35,18);printf("EDIT RECORDS");
  gotoxy(25,24);
```

```
      printf(" enter new rate: ");
      scanf("%f",&item.rate);
      size=sizeof(item);
      fseek(file,-size,SEEK_CUR);fwrite(&item,sizeof(item),1,file);
      break;
     case 4:
      system("cls");
      window(23,65,20,40);
      gotoxy(35,18);printf("EDIT RECORDS");
      gotoxy(25,24);
      printf(" enter new quantity: ");
      scanf("%d",&item.quantity);
      size=sizeof(item);
      fseek(file,-size,1);fwrite(&item,sizeof(item),1,file);
      break;
    }
    gotoxy(27,30);printf("--- item edited---");
    break;
   }
  }
 }
}
if (flag==1){
 gotoxy(32,30);printf("item does not exist");
 gotoxy(36,32);printf("TRY ABGAIN");
```

```
 }
 getch();
 fclose(file);
 d_mainmenu();
}

/*function to display all records*/
void d_all()
{
 int i,j=1;
 FILE *file;
 dis_con();
 file=fopen("record.txt","rb");
 rewind(file);
 i=26;
 fflush(file);
 while(fread(&item,sizeof(item),1,file)){
  display(&item,i,j);
  i++;
  j++;
  if ((j%20)==0){
        gotoxy(27,47);/*textcolor(0)*/;printf("press any key to see more.....");
        getch();
        system("cls");
        dis_con();
```

```
        i=26;
        continue;
  }
  }
  getch();
  if (i==26) {
  gotoxy(24,30); printf("-- no articles found --");
  }
 getch();
 fclose(file);
 d_mainmenu();
}

/*function to display by quantity*/
void d_quan()
{
  int i,j=1;
  int a,b;
  FILE *file;
  dis_con();
  file=fopen("record.txt","rb");
  rewind(file);
  i=26;
  gotoxy(16,20);;printf("enter lower range: ");
  scanf("%d",&a);
```

```
 gotoxy(16,21);printf("enter upper range:");
 scanf("%d",&b);
 fflush(file);
 while(fread(&item,sizeof(item),1,file)){
 if((item.quantity>=a)&&(item.quantity<=b)){
  display(&item,i,j);
  i++;
  j++;
  if ((j%20)==0){
   gotoxy(27,47);printf("press any key to see more.....");
   getch();
   system("cls");
   dis_con();
   i=26;
   continue;
  }
    }
 }
 getch();
 if (i==26){
 gotoxy(28,30); printf(" no item found ");
}
getch();
d_search();
fclose(file);
```

```
}

/*function to display by rate*/
void d_rate(){
  int i,j=1;
  float a,b;
  FILE *file;
  dis_con();
  file=fopen("record.txt","rb");
  rewind(file);
  i=26;
  gotoxy(16,20);;printf("enter lower range: ");
  scanf("%f",&a);
  gotoxy(16,21);printf("enter upper range: ");
  scanf("%f",&b);
  fflush(file);
  while(fread(&item,sizeof(item),1,file)){
  if((item.rate>=a)&&(item.rate<=b)){
   display(&item,i,j);
   i++;
   j++;
   if ((j%20)==0){
    gotoxy(27,47);printf("press any key to see more.....");
    getch();
    system("cls");
```

```
    dis_con();
    i=26;
    continue;
   }
  }
  }
 getch();
 if (i==26){
  gotoxy(28,30); printf(" no item found ");
 }
 getch();
 fclose(file);
 d_search();
}

/*function to display by code*/
void d_code()
{
  int i,j=1;
  char x[4]={0};
  FILE *file;
  dis_con();
  file=fopen("record.txt","rb");
  rewind(file);
  i=26;
```

```
 gotoxy(16,20);;printf("enter item code: ");
 scanf("%s",x);
 fflush(file);
 while(fread(&item,sizeof(item),1,file)){
 if((strcmp(item.code,x)==0)){
  display(&item,i,j);
  i++;
  j++;
  break;
 }
 }
 if (i==26){
 gotoxy(28,30); printf("no item found");
 }
 getch();
 fclose(file);
 d_search();
}

/*function to display window for item display*/
void dis_con()
{
 int i;
 system("cls");
 gotoxy(20,10);
```

```
 ;
 for (i=1;i<=10;i++)
  printf("*");
 printf(" * FASHION WEAR * ");
 for (i=1;i<=10;i++)
       printf("*");
 printf("\n\n");
 gotoxy(30,11);printf("Departmental Store");
 //textcolor(1);
 gotoxy(32,17);printf("RECORDS") ;
 //textcolor(8);
 gotoxy(18,23);printf ("SN   Item Name   Item Code      Rate     Quantity");
}

/*function to display in screen*/
void display(rec *item,int i,int j)
{
 gotoxy(16,i);//textcolor(13);
 printf("%4d",j);
 printf("%9s",item->name);
 printf("%12s",item->code);
 printf("%14.2f",item->rate);
 printf("%11d",item->quantity);
}
```

```
/*function to delete records*/
void del()
{
 int flag;
 char x[ANS];
 FILE *file,*file1;
 system("cls");
 //textbackground(11);
 //textcolor(0);
 window(23,51,25,34);
 gotoxy(29,18);printf("DELETE ARTICLES");
 gotoxy(27,27);printf("enter item code: ");
 scanf("%s",x);
 flag=check(x);
 if(flag==0){
  file1=fopen("record1.txt","ab");
  file=fopen("record.txt","rb");
  rewind(file);
  while (fread(&item,sizeof (item),1,file)){
   if(strcmp(item.code,x)!=0)
    fwrite(&item,sizeof(item),1,file1);
  }
  gotoxy(27,29);printf("---item deleted---");
  remove("record.txt");
  rename("record1.txt","record.txt");
```

```
 }
 if (flag==1){
  gotoxy(25,29);printf("---item does not exist---");
  gotoxy(30,31);printf("TRY AGAIN");
 }
 fclose(file1);
 fclose(file);
 getch();
 d_mainmenu();
}

/*function to check validity of code while editing and deleting*/
int check(char x[ANS])
{
 FILE *file;
 int flag=1;
 file=fopen("record.txt","rb");
 rewind(file);
 while (fread(&item,sizeof (item),1,file)){
  if(strcmp(item.code,x)==0){
   flag=0;
   break;
  }
 }
 fclose(file);
```

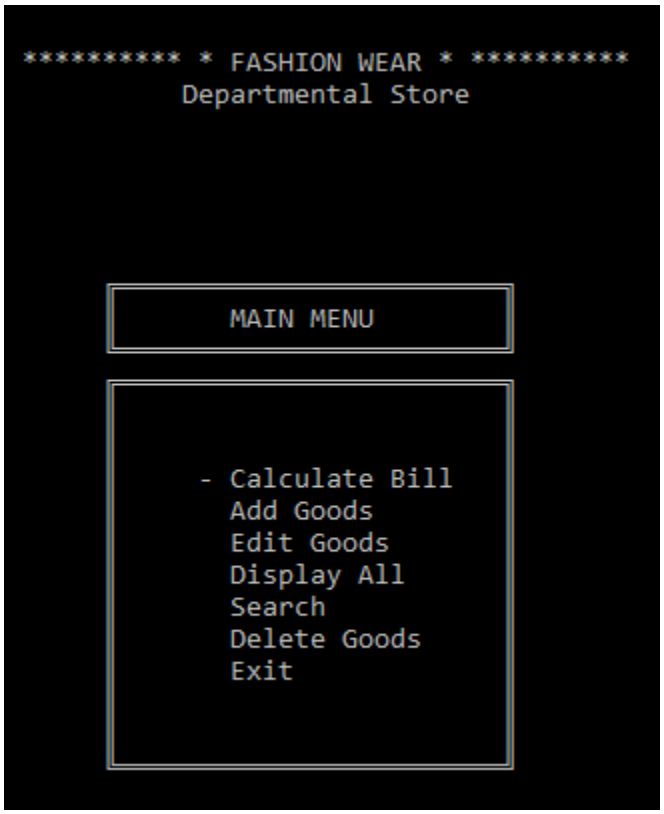
```
 return flag;
 }

/*function to display box*/
void window(int a,int b,int c,int d)
{
 int i;
 system("cls");
 gotoxy(20,10);
 //textcolor(1);
 for (i=1;i<=10;i++)
  printf("*");
 printf(" * FASHION WEAR * ");
 for (i=1;i<=10;i++)
       printf("*");
 printf("\n\n");
 gotoxy(30,11);printf("Departmental Store");
 //textcolor(4);
 for (i=a;i<=b;i++){
       gotoxy(i,17);printf("\xcd");
       gotoxy(i,19);printf("\xcd");
       gotoxy(i,c);printf("\xcd");
       gotoxy(i,d);printf("\xcd");
       }
```

```
 gotoxy(a,17);printf("\xc9");
 gotoxy(a,18);printf("\xba");
 gotoxy(a,19);printf("\xc8");
 gotoxy(b,17);printf("\xbb");
 gotoxy(b,18);printf("\xba");
 gotoxy(b,19);printf("\xbc");
 //textcolor(4);
 for(i=c;i<=d;i++){
  gotoxy(a,i);printf("\xba");
  gotoxy(b,i);printf("\xba");
 }
 gotoxy(a,c);printf("\xc9");
 gotoxy(a,d);printf("\xc8");
 gotoxy(b,c);printf("\xbb");
 gotoxy(b,d);printf("\xbc");
 //textbackground(11);
 //textcolor(0);
}
```

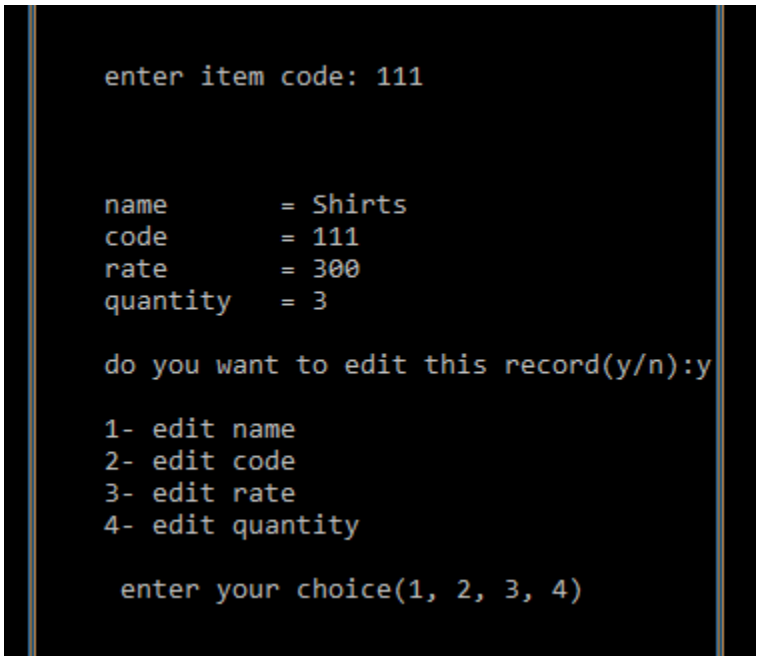
OUTPUT SCREENSHOTS

a. **Main Menu**



b. **Add Goods**

c. **Edit Goods:**



d. **Display all records**

```
********** * FASHION WEAR * **********
          Departmental Store



                RECORDS



  SN   Item Name   Item Code      Rate     Quantity

   1   shirts         111        340.00         10
   2    Jeans         222        300.00         30
   3 T-shirts         333        300.00         18
```

e. **Search records**

```
        SEARCH MENU

      - By Code
        By Rate
        By Quantity
        Back to main menu
```

```
                RECORDS

enter item code: 111

  SN   Item Name   Item Code      Rate     Quantity

   1   shirts         111        340.00         10
```

```
                RECORDS


enter lower range: 100
enter upper range: 300

  SN   Item Name   Item Code      Rate     Quantity


   1    Jeans         222        300.00         30
   2 T-shirts         333        300.00         18
```

```
                RECORDS


enter lower range: 10
enter upper range:40

  SN   Item Name   Item Code      Rate     Quantity


   1   shirts         111        340.00         10
   2    Jeans         222        300.00         30
   3 T-shirts         333        300.00         18
```

f. **Delete records:**

```
                RECORDS




 SN   Item Name   Item Code      Rate     Quantity


  1   Shirts         111        300.00         12
  2    Jeans         222        350.00         20
```

```
   enter item code: 111

   ---item deleted---
```

```
********** * FASHION WEAR * **********
          Departmental Store



              RECORDS



 SN   Item Name   Item Code      Rate     Quantity

  1    Jeans         222        350.00         20
```

g. **Calculate bill**

```
********** * FASHION WEAR * **********
          Departmental Store


            enter  "end" to finish input

           enter item code:end




                 CUSTOMER'S BILL

 SN.   Item Name     Quantity     Rate          Total

  1   shirts           10         340.00      3400.00

  2 T-shirts           15         300.00      4500.00



                TOTAL AMOUNT = NRs. 7900.00
```

FUTURE SCOPE

1. The utility currently aims at operating a small scale, but its operation can be widened by adding some more functionalities to it so that it can be used at a large scale.
2. The scope of server centered system for this utility can also be explored.
3. A proper database of all activities performed can be maintained at a larger scale such as malls, or chain of retail shops, etc.
4. The system can implement security functions for users to access the system.

REFERENCES

1. https://www.geeksforgeeks.org/
2. https://en.wikipedia.org/wiki/Department_store
3. https://en.wikibooks.org/wiki/Windows_Programming/windows.h
4. https://www.learn-c.org/