VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-**590018**

“File Structures Laboratory with Mini project”

(Subject Code: 18ISL67)

“DEPARTMENTAL STORE MANAGEMENT SYSTEM”

Submitted in partial fulfillment for 6th semester for the Award of Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

ANANYA R (1EP19IS003)

ASHA.K (1EP19IS010)

ASHWAQULLA BAIG (1EP19IS011)

BAL KISHAN REDDY (1EP19IS013)

UNDER THE GUIDANCE OF

Prof.INDUMATHI S

Prof.SWETHA R

Assistant. Professor

Dept. of ISE, EPCET

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY

Department of Information Science and Engineering
Jnana Prabha Campus, Bidarahalli,
Bangalore – 560 049
2021 -2022

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

Jnana Prabha Campus, Bidarahalli,

Bangalore – 560 049

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the **File Structures Laboratory with Mini project (18ISL67)** entitled **"DEPARTMENTAL STORE MANAGEMENT SYSTEM "** is a bonafide work carried out by ANANYA R, **bearing USN** 1EP19IS003, ASHA K, **bearing USN** 1EP19IS010, ASHWAQULLA BAIG, **bearing USN** 1EP19IS011, BAL KISHAN REDDY, **bearing USN** 1EP19IS013 in partial fulfilment of 6th semester for the award of, **Bachelor of Engineering in Information Science and Engineering** under **Visvesvaraya Technological University, Belagavi** during the year **2021- 2022**. This report has been approved as it satisfies the academic requirements in respect of DBMS Mini Project work prescribed for the award of the said degree.

GUIDE	**HOD**	**PRINCIPAL**
Prof. Indumathi S	**Dr. Lingaraju G M**	**Dr. T.K. Sateesh**
Prof. Swetha R	Head of the Department	Principal
Asst. Professor	Dept of ISE	EPCET, Bangalore
Dept of ISE	EPCET, Bangalore	
EPCET, Bangalore		

External examiner's **Date and sign**

1)

2)

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. We would like to take this opportunity to thank them all.

First and foremost, we would like to express our sincere regards and thanks to **Mr.Promod Gowda** and **Mr. Rajiv Gowda**, CEO's, East Point Group of Institutions, Bangalore, for providing necessary infrastructure and creating good environment.

We express our gratitude to **Dr.T.K. Sateesh,** Principal, EPCET who has always been a great source of inspiration.

We express our sincere regards and thanks to **Dr Lingaraju G M,** Professor and Head of Department of Information Science and Engineering, EPCET, Bangalore, for his encouragement and support.

We are grateful to acknowledge the guidance and encouragement given to us by **Prof.Indumathi S,** Assistant Professor and **Prof.swetha R,** Assistant Professor Department of Information Science and Engineering, EPCET, Bangalore, as the project coordinator and guides who have rendered a valuable assistance.

We also extend our thanks to the entire faculty of the **Department of Information Science and Engineering,** EPCET, Bangalore, who have encouraged us throughout the course of the project work .

Last, but not the least, we would like to thank our family and friends for their inputs to improve the project.

ANANYA R :1EP19IS003
ASHA.K :1EP19IS010
ASHWAQULLA BAIG:1EP19IS011
BAL KISHAN REDDY:1EP19IS013

ABSTRACT

“Departmental Store Management System” is designed to make the existing manual system, automatic with the help of computerized equipment and full-edged computer program module, fulfilling all their requirements of managing products, accurate bill calculation, so that their valuable data and information of customers can be stored for a longer period with easy access and manipulation of the same. The required software is easily available and easy to work with. This program module can maintain and view computerized records without getting redundant entries. The project describes how to manage user data for better performance and accurate calculation of bill using the data stored. To overcome the difficultiesof the activities of the retail store, the present system is automated to perform all the activities of the store.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Problem definition

The proposed project “Departmental Store Management System” has been developed to overcome the problems faced in the practicing of manual system. This program module is built to eliminate and, in some cases, reduce the hardships faced by the existing system. Moreover this system is designed for particular need of the retail stores to carry out their operations in a smooth and effective manner.

Many customer delays are faced by every retail store, which must be overcome by these stores.Every store has different bill calculation and managing data needs. Therefore this project have designed exclusively to adapt to the store’s managerial requirements.

1.2 Purpose

The purpose of this project document is to describe the functionality and specifications of the design of a program module for managing customer data and their bill calculations. At last, we can say that this module system will not only automate the process, will also save the valuable time of the manager or the bill receptionist, which can be well utilized by every retail store. This will be an additional advantage for every retail store to organize and process the products effectively.

CHAPTER 2

REQUIREMENT SPECIFICATIONS

2.1 Hardware Requirements:

Processor Brand	: Intel
Processor Type	: Intel Core i5
Processor Speed	: 4 GHz
Processor Count	: 1
RAM Size	: 8 GB
Memory Technology	: DDR3
Computer Memory Type	: DDR3 SDRAM
Hard Drive Size	: 1 GB

2.2 Software Requirements:

Operating system	: Windows 11
Front end	: C
Browser	: Google chrome
Connectivity	: Internet

CHAPTER 3

TOOL DESCRIPTION

3.1 OVERVIEW OF FRONT END APPLICATION:

C is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs. C was originally first implemented on the DEC PDP-11 computer in 1972.

In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, now known as the K&R standard.

The UNIX operating system, the C compiler, and essentially all UNIX application programs have been written in C. C has now become a widely used professional language for various reasons −

- Easy to learn
- Structured language
- It produces efficient programs
- It can handle low-level activities
- It can be compiled on a variety of computer platforms

Facts about C:

- C was invented to write an operating system called UNIX.
- C is a successor of B language which was introduced around the early 1970s.
- The language was formalized in 1988 by the American National Standard Institute (ANSI).
- The UNIX OS was totally written in C.

Use of C:

C was initially used for system development work, particularly the programs that make-up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as the code written in assembly language Operating Systems
Language Compilers, Assemblers, Text Editors, Print Spoolers, Network Drivers, Modern, Programs etc.

CHAPTER 4

IMPLEMENTATION

4.1 Source code

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

#define CLEAR "cls"
#define MAX 20

typedef struct items
{
    char itemCode[MAX];
    char itemName[MAX];
    int quantity;
    float rate;
    char description[MAX];
}Item;

Item item;



void options();
int isCodeAvailable();
void calculateBill();
void addProd();
void deleteProd();
void editProd();
void display();

int main()
{
   options();
   system(CLEAR);
   return 0;
}



void options()
{
   int choice;

   do
   {
      printf("\n\n\n\n\n\t\t\t\t\t*** WELCOME TO ALL-IN ONE DEPARTMENTAL STORE ***");
      printf("\n\t\t\t\t\t-----------------------------------------------");
      printf("\n\t\t\t\t\t\t\t1. CALCULATE BILL");
      printf("\n\t\t\t\t\t\t\t2. ADD A PRODUCT");
      printf("\n\t\t\t\t\t\t\t3. DELETE A PRODUCT");
      printf("\n\t\t\t\t\t\t\t4. EDIT A PRODUCT");
      printf("\n\t\t\t\t\t\t\t5. DISPLAY STORE");
      printf("\n\t\t\t\t\t\t\t6. EXIT STORE");
      printf("\n\t\t\t\t\t-----------------------------------------------\n");
      printf("\t\t\t\t\tEnter your choice: ");
      scanf("%d", &choice);

      switch(choice)
      {
```

```
case 1:
    system(CLEAR);
    calculateBill();
    break;
case 2:
    system(CLEAR);
    addProd();
    break;
case 3:
    system(CLEAR);
    deleteProd();
    break;
case 4:
    system(CLEAR);
    editProd();
    break;
case 5:
    system(CLEAR);
    display();
    break;
case 6:
    system(CLEAR);
    printf("\n\n\n\n\n\n\n\n\n\n");
    printf("\n\n\n\t\t\t\t\tThanks! for Shopping with us\n");
    printf("\t\t\t\t      ==================================\n");
    printf("\t\t\t\t\tHope to see you soon again\n\n\n");
    exit(0);
    break;
default:
    printf("\n\t\t\t\t\tInvalid! entry. Make a choice between 1 & 6\n\n\n");
    break;
```

```
        }
    } while (choice != 6);
}

int isCodeAvailable(char code[])
{
    FILE *file;
    file = fopen("Record.txt", "r");
    if(file == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file!\n");
        exit(1);
    }

    while (fread(&item, sizeof(item), 1, file))
    {
        if(strcmp(code, item.itemCode) == 0)
        {
            fclose(file);
            return 1;
        }
    }
    fclose(file);
    return 0;
}

void addProd()
{
    char code[MAX];
    FILE *file;
    file = fopen("Record.txt", "a");
```

```
if(file == NULL)
{
    printf("\n\n\t\t\t\tError occurred while opening the file!\n");
    exit(1);
}

display();
jump:
printf("\t\t\t\t\t    *****ADDING A PRODUCT*****\n\n");
printf("\t\t\t\t   Enter \"end\" if you want to go back to main menu\n\n");
printf("\t\t\t\t\t\tProduct Code:");
scanf("%s", code);

if (strcmp(code, "end") == 0)
{
    system(CLEAR);
    options();
}

if(isCodeAvailable(code) == 1)
{
    printf("\n\n\t\t\t\t\tProduct already exists!\n\n\n");
    goto jump;
}

strcpy(item.itemCode, code);

printf("\t\t\t\t\t\tProduct Name:");
scanf("%s", item.itemName);

printf("\t\t\t\t\t\tProduct Rate:");
```

```
    scanf("%f", &item.rate);

    printf("\t\t\t\t\t\tProduct Quantity:");
    scanf("%d", &item.quantity);

    printf("\t\t\t\t\t\tDescription:");
    scanf("%s", item.description);

    if(fwrite(&item, sizeof(item), 1, file))
    {
        printf("\n\t\t\t\t\t  Data has been stored successfully!\n\n\n");
    }
    fclose(file);
}

void deleteProd()
{
    FILE *file1, *file2;
    char code[MAX];

    file1 = fopen("Record.txt", "r");
    file2 = fopen("temp.txt", "w");
    if(file1 == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file 1!\n");
        exit(1);
    }

    if(file2 == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file 2!\n");
        fclose(file1);
        exit(1);
    }

    display();
    jump:
    printf("\t\t\t\t\t    *****DELETING A PRODUCT*****\n\n");
    printf("\t\t\t\t   Enter \"end\" if you want to go back to main menu\n\n");
    printf("\t\t\t\t   Enter the code of the product you want to delete: ");
    scanf("%s", code);

    if (strcmp(code, "end") == 0)
    {
        system(CLEAR);
        options();
    }

    if(isCodeAvailable(code) == 0)
    {
        printf("\n\n\t\t\t\t\t\tProduct doesn't exist!\n\n\n");

        goto jump;
    }

    while (fread(&item, sizeof(item), 1, file1))
    {
        if (strcmp(code, item.itemCode) != 0)
        {
            fwrite(&item, sizeof(item), 1, file2);
        }
    }
    fclose(file1);
    fclose(file2);

    file1 = fopen("Record.txt", "w");
    file2 = fopen("temp.txt", "r");

    if(file1 == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file 1!\n");
        exit(1);
    }

    if(file2 == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file 2!\n");
        fclose(file1);
        exit(1);
    }

    while (fread(&item, sizeof(item), 1, file2))
    {
        fwrite(&item, sizeof(item), 1, file1);
    }

    printf("\n\v\t\t\t\t\t   Product deleted sucessfully!\n\n\n\n");

    fclose(file1);
    fclose(file2);
}

void editProd()
```

```
{
    char code[MAX];
    FILE *file1, *file2;

    display();
    jump:
    printf("\t\t\t\t\t    *****UPDATING A PRODUCT*****\n\n");
    printf("\t\t\t\t   Enter \"end\" if you want to go back to main menu\n\n");
    printf("\t\t\t\t   Enter the code of the product you want to edit: ");
    scanf("%s", code);

    if (strcmp(code, "end") == 0)
    {
        system(CLEAR);
        options();
    }

    if(isCodeAvailable(code) == 0)
    {
        printf("\n\n\t\t\t\t\t\tProduct doesn't exist!\n\n\n");

        goto jump;
    }

    file1 = fopen("Record.txt", "r");
    file2 = fopen("temp.txt", "w");
    if(file1 == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file 1!\n");
        exit(1);
    }
```

```
if(file2 == NULL)
{
    printf("\n\n\t\t\t\tError occurred while opening the file 2!\n");
    fclose(file1);
    exit(1);
}

while(fread(&item, sizeof(item), 1, file1))
{

    if(strcmp(code, item.itemCode) != 0)
    {
        fwrite(&item, sizeof(item), 1, file2);
    }
    else
    {
        printf("\t\t\t\t\t\tProduct Name:");
        scanf("%s", item.itemName);

        printf("\t\t\t\t\t\tProduct Rate:");
        scanf("%f", &item.rate);

        printf("\t\t\t\t\t\tProduct Quantity:");
        scanf("%d", &item.quantity);

        printf("\t\t\t\t\t\tDescription:");
        scanf("%s", item.description);

        fwrite(&item, sizeof(item), 1, file2);
    }
    }
    fclose(file1);
    fclose(file2);

    file1 = fopen("Record.txt", "w");
    file2 = fopen("temp.txt", "r");
    if(file1 == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file 1!\n");
        exit(1);
    }

    if(file2 == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file 2!\n");
        fclose(file1);
        exit(1);
    }

    while(fread(&item, sizeof(item), 1, file2))
    {
        fwrite(&item, sizeof(item), 1, file1);
    }

    printf("\n\n\t\t\t\t\tData has been updated successfully\n\n\n");

    fclose(file1);
    fclose(file2);
}

void display()
```

```
{
    int count = 0;
    FILE *file;
    file = fopen("Record.txt", "r");
    if(file == NULL)
    {
        printf("\n\n\t\t\t\tError occurred while opening the file!\n");
        exit(1);
    }

    printf("\t\t\t\t\t\t    AVAILABLE PRODUCTS\n");
    printf("\t\t\t\t\t\t***************************\n");
    printf("\t\t\t-----------------------------------------------------------------------\n");
    printf("\t\t\t CODE\t||    NAME\t||     RATE\t||   QUANTITY   ||  DESCRIPTION \n");
    printf("\t\t\t-----------------------------------------------------------------------\n");

    while (fread(&item, sizeof(item), 1, file))
    {
        printf("\t\t\t  %s\t||   %s\t||     %.2f\t||     %d\t||  %s\t\n", item.itemCode, item.itemName, item.rate,
item.quantity, item.description);
        count++;
    }
    printf("\t\t\t-----------------------------------------------------------------------\n\n\n");

    if (count == 0)
    {
        system(CLEAR);
        printf("\n\t\t\t\t\t   *No products available in store\n\n\n");

    }
    fclose(file);
```

```
}

void calculateBill()
{
   FILE *file1, *file2;
   COORD c;
   char code[MAX];
   int quantity = 0, itemCounter = 1, coordY = 16;
   float total = 0, totalBill = 0;

   c.X = 4;
   c.Y = 13;
   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
   printf("\t\t\t---------------------------------------------------------------------------------------");

   c.X = 4;
   c.Y = 14;
   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
   printf("\t\t\t S.N\t||    CODE\t||    NAME\t||     RATE\t||   QUANTITY   ||  TOTAL ");

   c.X = 4;
   c.Y = 15;
   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
   printf("\t\t\t---------------------------------------------------------------------------------------");



   while(1)
   {
      c.X = 50;
      c.Y = 7;
      SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
```

```
printf("Enter \"end\" if you want to exit.");

c.X = 50;
c.Y = 8;
SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
printf("Enter product code:");
scanf("%s", code);

if (strcmp(code, "end") == 0)
{
   c.X = 50;
   c.Y = coordY + 3;
   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
   printf("Total bill: Rs. %.2f", totalBill);
   break;
}

c.X = 50;
c.Y = 9;
SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
printf("Enter Quantity:");
scanf("%d", &quantity);

file1 = fopen("Record.txt", "r");
file2 = fopen("Update.txt", "w");
while(fread(&item, sizeof(item), 1, file1))
{
   if((strcmp (code, item.itemCode)) == 0)
   {
      total = quantity * item.rate;
      totalBill += total;
```

```
            item.quantity -= quantity;

            c.X = 4;
            c.Y = coordY;
            SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
            printf("\t\t\t   %d\t||     %s\t||   %s\t||     %.2f\t||      %d\t||   %.2f\n", itemCounter, item.itemCode,
item.itemName, item.rate, quantity, total);
            coordY++;
            itemCounter++;
            fwrite(&item, sizeof(item), 1, file2);
        }
        else
        {
            fwrite(&item, sizeof(item), 1, file2);
        }
    }
    fclose(file1);
    fclose(file2);

    file1 = fopen("Record.txt", "w");
    file2 = fopen("Update.txt", "r");
    while(fread(&item, sizeof(item), 1, file2))
    {
        fwrite(&item, sizeof(item), 1, file1);
    }
    fclose(file1);
    fclose(file2);
  }
}
```

CHAPTER 5

TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

5.1 Unit Testing

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by python. The various controls are tested to ensure that each performs its action as required.

5.2 Integration Testing

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the Server module and Client module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

5.3 User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

5.4 TEST CASES

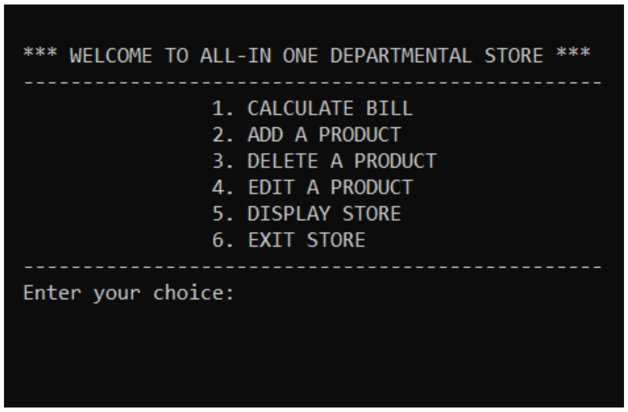
Table 5.4: Test Cases

Test no	**Test name**	**Input**	**Actual output**	**Expected output**	**Status**
1	Menu System	Run source code and compile.	Menu System gets displayed.	Menu System gets displayed.	Pass
2	Add Product	Enter the choice 2.	Adding of product details.	Product details gets added successfully.	Pass
3	Update Product	Enter the choice 4.	Updating of product details.	Product details gets updated successfully.	Pass
4	Delete Product	Enter the choice 3.	Deleting of product details.	Product details gets deleted successfully.	Pass
5	Bill Calculation	Enter the choice 1.	Calculating of product bill.	Product bill gets calculated and displayed successfully.	Pass
5	Exit	Enter the choice 6.	Termination of the program.	The program gets terminated successfully.	Pass
6	Invalid Choice	Enter the choice 7.	Invalid Choice	Invalid Choice	Fail

CHAPTER 6

SNAPSHOTS

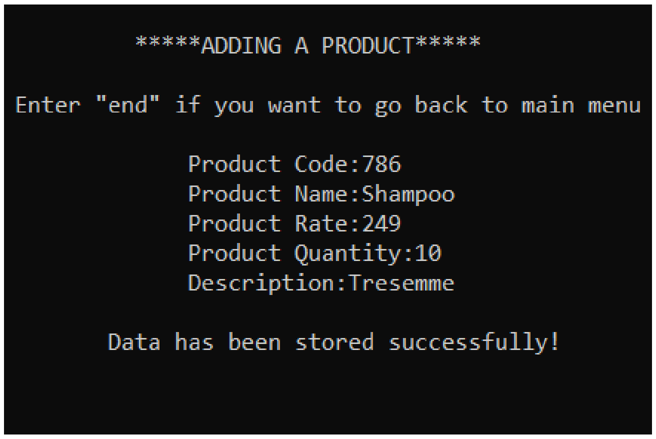
Menu System: This is the main interface in the terminal, which allows the user to choose from multiple choices i.e., from 1-6 for specific operations like adding, updating, deleting, displaying, bill calculating and exiting the program.



```
*** WELCOME TO ALL-IN ONE DEPARTMENTAL STORE ***
------------------------------------------------
                1. CALCULATE BILL
                2. ADD A PRODUCT
                3. DELETE A PRODUCT
                4. EDIT A PRODUCT
                5. DISPLAY STORE
                6. EXIT STORE
------------------------------------------------
Enter your choice:
```

Figure 6.1: Menu System

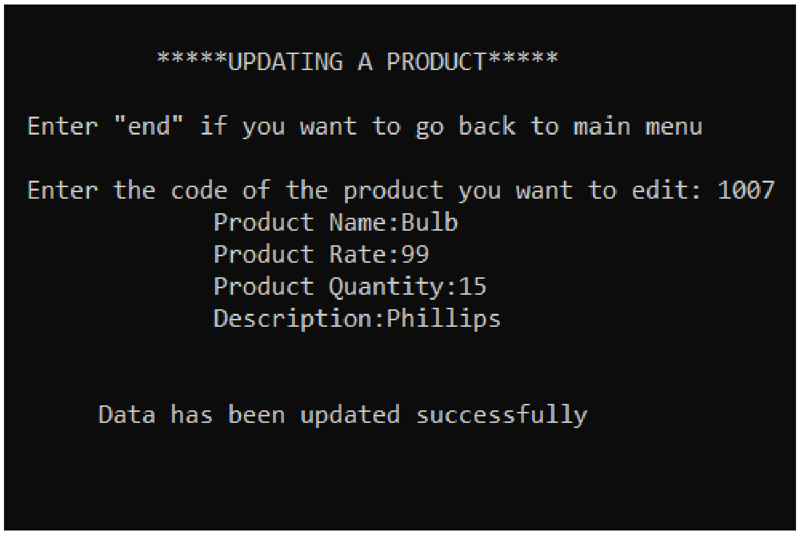
Insert Product: If a user opts for choice 2, adding a product section displays, then the user is allowed to add any product by entering the product details: code, name, rate, quantity, description. After entering these details the message “Data has been stored successfully!” gets displayed.



```
           *****ADDING A PRODUCT*****

Enter "end" if you want to go back to main menu

               Product Code:786
               Product Name:Shampoo
               Product Rate:249
               Product Quantity:10
               Description:Tresemme

        Data has been stored successfully!
```

Figure 6.2: Insert Product

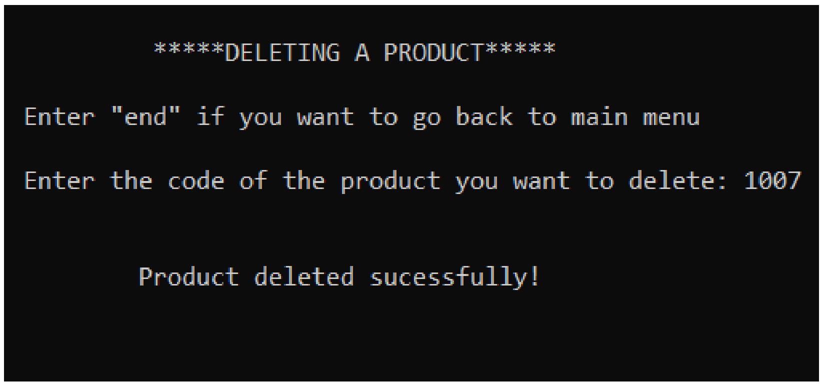
Update Product: If a user opts for choice 4, updating a product section displays, then the user is allowed to update any product by entering the product details: code, name, rate, quantity, description. After entering these details the message “Data has been updated successfully!” gets displayed.



```
          *****UPDATING A PRODUCT*****

 Enter "end" if you want to go back to main menu

 Enter the code of the product you want to edit: 1007
             Product Name:Bulb
             Product Rate:99
             Product Quantity:15
             Description:Phillips


      Data has been updated successfully
```

Figure 6.3: Update Product

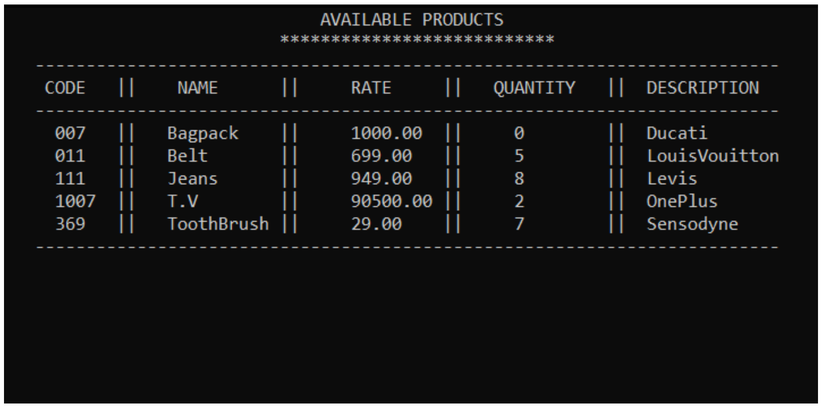
Delete Product: If a user opts for choice 3, deleting a product section displays, then the user is allowed to delete any product by entering the product code. After entering the product code, the message “Product deleted successfully!” gets displayed.



```
          *****DELETING A PRODUCT*****

 Enter "end" if you want to go back to main menu

 Enter the code of the product you want to delete: 1007


        Product deleted sucessfully!
```

Figure 6.4: Delete Product

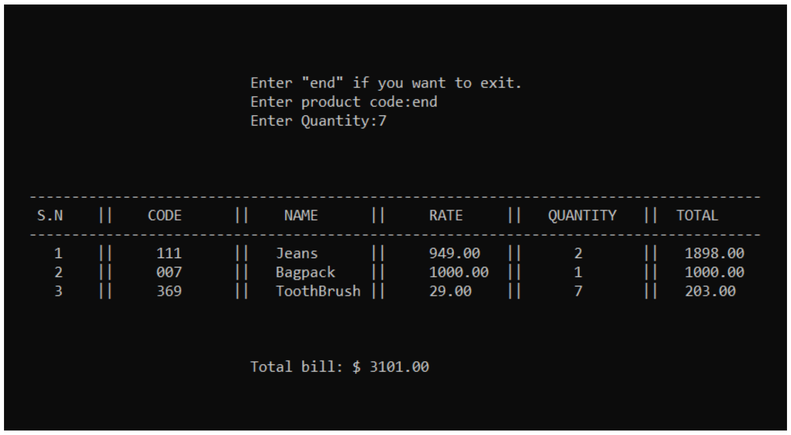
Display Product: If a user opts for choice 5, available products section displays, then the user can see the listed or available product details: code, name, rate, quantity, description, after any Inserting, Deleting and Updating of a product is done.



```
                         AVAILABLE PRODUCTS
                     **************************
 ----------------------------------------------------------------------
 CODE   ||    NAME      ||     RATE     ||   QUANTITY   ||  DESCRIPTION
 ----------------------------------------------------------------------
  007   ||   Bagpack    ||    1000.00   ||      0       ||  Ducati
  011   ||   Belt       ||    699.00    ||      5       ||  LouisVouitton
  111   ||   Jeans      ||    949.00    ||      8       ||  Levis
  1007  ||   T.V        ||    90500.00  ||      2       ||  OnePlus
  369   ||   ToothBrush ||    29.00     ||      7       ||  Sensodyne
 ----------------------------------------------------------------------
```

Figure 6.5: Display Product

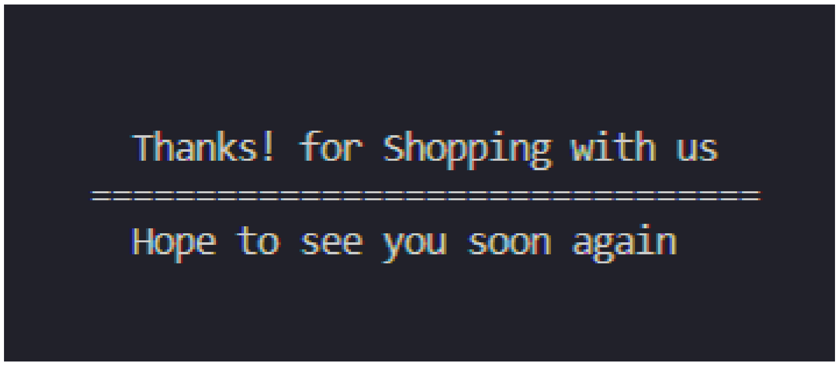
Bill Calculation: If a user opts for choice 1, calculation a product section displays, then the user is allowed to enter the product code and quantity for the product a user wants to buy. After entering the product code and quantity, each product gets listed along with its total. After finalizing, type" end" to close bill calculation, then the final bill gets displayed at the end.



```
                    Enter "end" if you want to exit.
                    Enter product code:end
                    Enter Quantity:7


 -------------------------------------------------------------------------------
 S.N   ||   CODE    ||   NAME     ||    RATE    ||  QUANTITY  ||  TOTAL
 -------------------------------------------------------------------------------
   1   ||    111    ||  Jeans     ||    949.00  ||     2      ||   1898.00
   2   ||    007    ||  Bagpack   ||    1000.00 ||     1      ||   1000.00
   3   ||    369    ||  ToothBrush||    29.00   ||     7      ||   203.00


                    Total bill: $ 3101.00
```

Figure 6.6: Bill Calculation

Exit: If a user opts for choice 6, then the menu system gets terminated and the final message “Thanks! For Shopping with us, Hope to see you soon again” gets displayed.



```
  Thanks! for Shopping with us
================================
  Hope to see you soon again
```

Figure 6.7: Exit

CONCLUSION

Departmental Store Management System is a program module developed for a company has been designed to achieve maximum efficiency and reduce the time taken to handle the bill calculating activity. It is designed to replace an existing manual bill counting record system thereby reducing time taken for calculation and for storing data.

REFERENCES

- https://www.w3schools.com
- https://www.programiz.com
- https://www.visualstudio.microsoft.com