



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

A MINI PROJECT REPORT
ON

SHOPPING
“BUYMEE”

*Submitted in partial fulfillment for the award of the degree of
Bachelor of Engineering
In*

COMPUTER SCIENCE AND ENGINEERING

Submitted by

PREKSHA SHRIDHAR

1NH17CS102

'B' SECTION

Reviewed by

MR. MURALIDHARA

Asst. Professor in computer science dep.



NEW HORIZON
COLLEGE OF ENGINEERING

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

CERTIFICATE

This is to certify that the mini project work titled

SHOPPING

*Submitted in partial fulfillment for the award of the degree of
Bachelor of Engineering*

PREKSHA SHRIDHAR
1NH17CS102

During the academic year
2018-2019

Signature of Reviewer

Signature of HOD

Semester End Examination

Name of the Examiner

Signature with date

1.....

.....

2.....

.....

ACKNOWLEDGEMENT

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

I thank the management, Dr. Mohan Manghnani, Chairman of NEW HORIZON EDUCATIONAL INSTITUTIONS for providing necessary infrastructure and creating good environment.

I also record here the constant encouragement and facilities extended to me by Dr. Manjunatha, Principal, NHCE, Dr. Prashanth.C.S.R, Dean Academics, Dr. B. Rajalakshmi, Head of the Department of Computer Science and Engineering. I extend my sincere gratitude to them.

I express my gratitude to Mr.Muralidhara, my project reviewer for constantly monitoring the development of the project and setting up precise deadlines. His valuable suggestions were the motivating factors in completing the work.

Finally a note of thanks to all the teaching and non-teaching staff of Computer Science and Engineering Department for their cooperation extended to me and my friends, who helped me directly or indirectly in the course of the project work.

PREKSHA SHRIDHAR

1NH17CS102

ABSTRACT

Main object of the project is to implement Data structure.

Linked list is the type of data structure which is mainly used to implement shopping cart.

Admin should input all the available item names with their quantity and price.

It's stored in the files. The shopping main page opens.

User must log-in to proceed further. If the user hasn't registered then user must register with their name, email id, contact number and address.

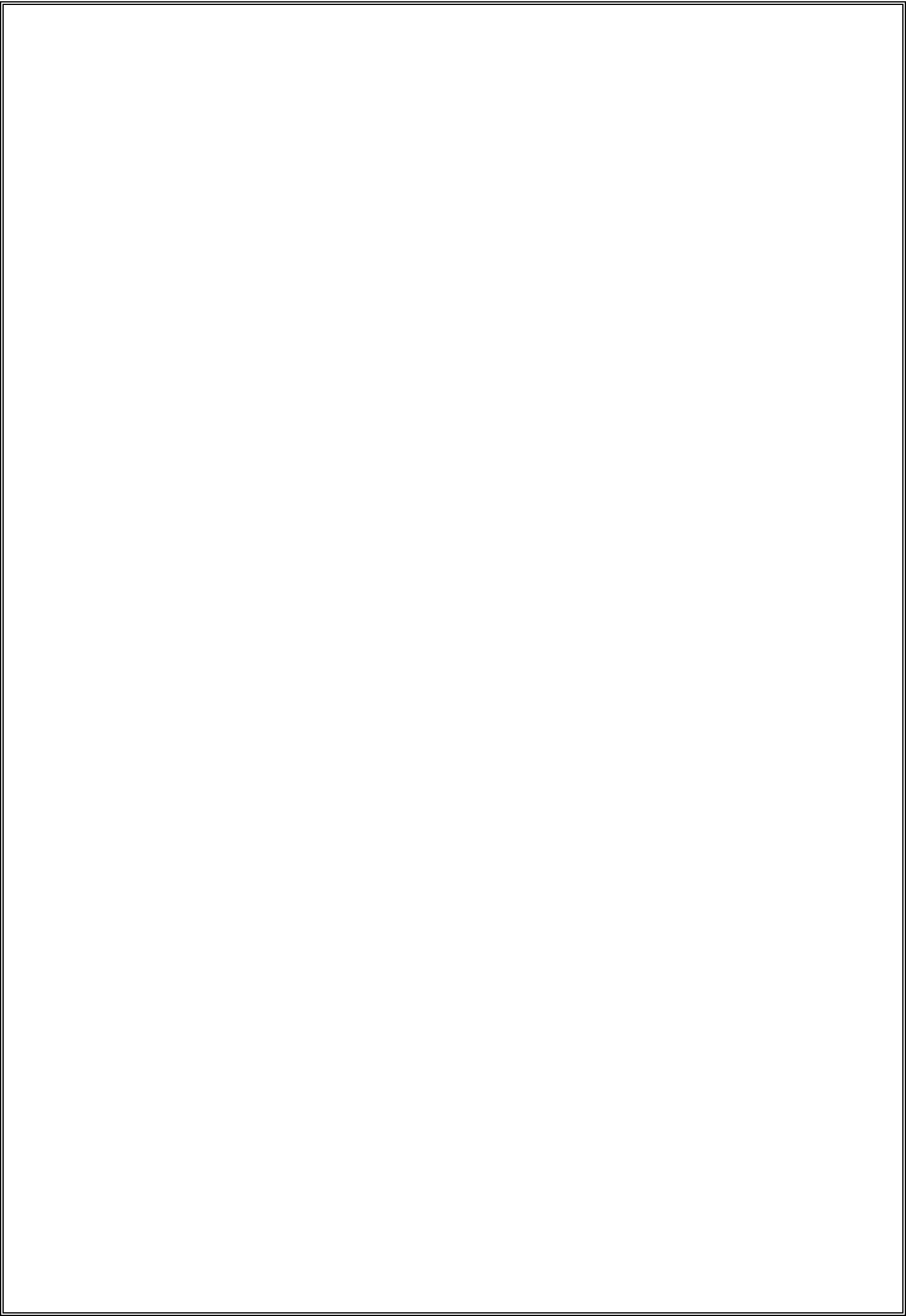
User can search for needed items, select and add to cart.

1. Collection of items each containing details for cart information such as quantity and price.
2. Items are able to be added and removed.
3. The cart can be emptied in a single action.
4. Certain parts of the data should be updated when items are added/removed such as subtotal, quantity etc.
5. Displays the item in the cart and simultaneously calculates the total amount and prints the bill.

TABLE OF CONTENTS

CHAPTERS

CHAPTER NO.	TITLE	PAGE NO.
1	Introduction	1
2	Analysis	2
2.1	Objectives of the project	2
2.2	Requirements Specifications	2
3	Design	3
3.1	Algorithm	3
3.2	Data structures	3-6
4	Implementation	7-13
5	Sample Output	14-18
6	Conclusion	19
7	Bibliography	20



CHAPTER 1

INTRODUCTION

This project is all about the shopping procedure. The purpose of this project is that it provides an easy and convenient way of calculating the total amount of the items taken. This project not only contains the details of items but also takes the details of the users. This project not only contains the billing but also takes in new items and also the user can delete the items from the cart. Admin has the access to insert the items to store. Users can choose the items from the displayed items. Admin has the access to insert the items to the store. Users can choose the items from the displayed items in the store according to their needs.

This project is made easy for users to buy the items whenever and wherever they want to buy from.

It's one of the conveniences and safest method of shopping. The greatest advantage of this method is that the user doesn't have to wait in queue and purchase items fast.

CHAPTER 2

ANALYSIS

2.1 OBJECTIVES OF THE PROJECT

The main object is to implement data structure.

The linked list is the type of data structure mainly used to implement a shopping cart.

1. Collection of items each containing properties for cart information such as quantity and price.
2. Items should be able to be added and removed.
3. The cart should be able to be emptied in a single action.
4. Certain parts of data should be updated when items are added/removed such as subtotal, quantity etc.

Admin should input all the available item names with their quantity and price. The shopping man page opens. The user must log-in to proceed further. If the user hasn't registered then the user must register with their phone number. The user can search for needed items, select and add to cart. After adding items, the user can either place an order or can cancel the items. After placing the order total price will be calculated and shown on the screen. Then the user must choose an option of payment. The user can log-out of the page in the end.

2.2 REQUIREMENT SPECIFICATION

2.2.1. Hardware Requirement

- 1.PIV 2.8 GHz Processor and Above
- 2.RAM 512MB and Above
- 3.HDD 20 GB Hard Disk Space and Above

2.2.2. Software Requirements

- 1.WINDOWS OS (XP/10)
- 2.Turbo c++.

CHAPTER 3

DESIGN

3.1 ALGORITHM

Step 1: Start.

Step 2: User page.

Step 3: Login using id or phone number.

Step 4: If not registered, register and then login.

Step 5: Select items and add to cart.

Step 6: Place the ordering or cancel the order.

Step 7: After placing order go for payment.

Step 8: Feedback (not mandatory).

Step 9: Logout of the page.

Step 10: Stop.

3.2 DATA STRUCTURES

Files

The operations using C program are done on a terminal which are not stored anywhere. Most of the programs are written to store the information that are fetched. One such way is to store that fetched information is in the form of file. Different operations that can be performed on a file are:

1. Creation of a new file (fopen with attributes as “a” or “a+” or “w” or “w++”)
 2. Opening an existing file (fopen)
 3. Reading from file (fscanf or fgetc)
 4. Writing to a file (fprintf or fputs)
 5. Moving to a specific location in a file (fseek, rewind)
 6. Closing a file (fclose)
- a. When a program is terminal, the entire data is lost. Storing in a file will preserve your data even if the program terminates.
 - b. If you must enter a large number of data, it will take a lot of time to enter them all.
 - c. However, if you have a file containing all the data, you can easily access the contents of the file using few commands in C.
 - d. You can easily move your data from one computer to another without any changes.

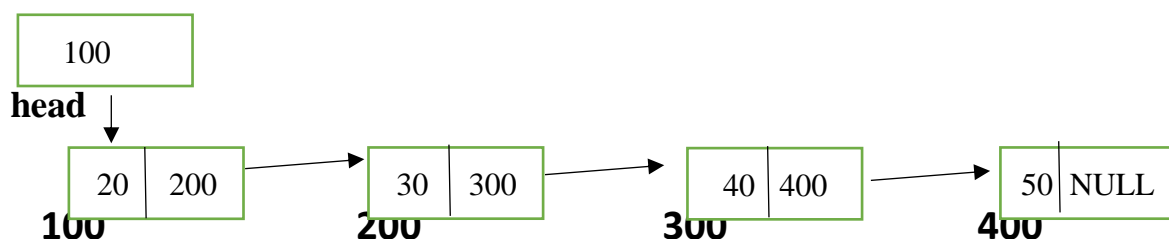
Linked List Data Structure

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. In simple words, a linked list consists of nodes where each node contains a data field and a reference (link) to the next node in the list. A linked list is a set of dynamically allocated nodes, arranged in such a way that each node contain one value and one pointer.

The pointer always points to the next member of the list. If the pointer is NULL, then it is the last node in the list.

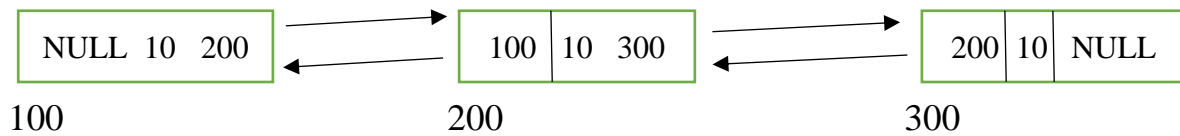
A linked list is held using a local pointer variable while points to the first item of the list. Is that pointer is also NULL, then the list is considered to be empty.

The elements in a linked list are linked using pointers as shown in the below images:

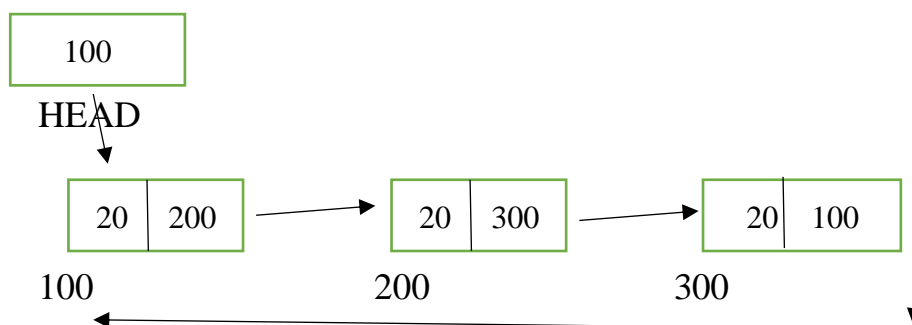


Types of linked lists are:

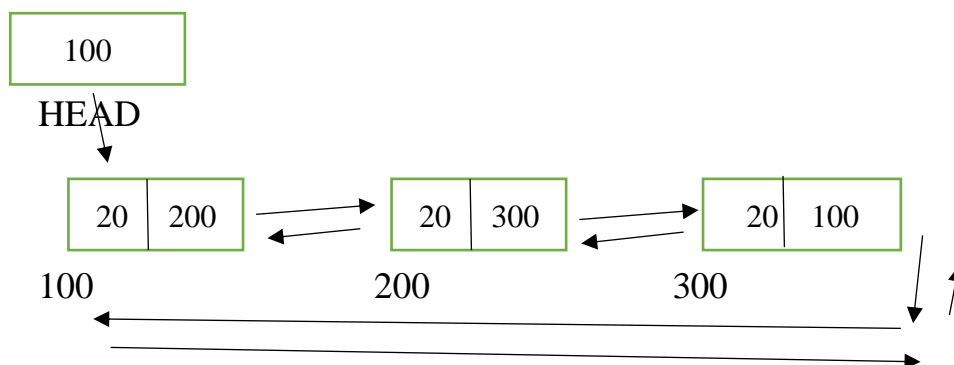
1. Singly Linked lists.
2. Doubly Linked lists.



3. Circular Linked lists.



4. Doubly Circular Linked lists.



Advantages of linked list over arrays are:

1. Items can be added or removed from the middle of the list
2. There is no need to define an initial size

linked lists also have a few disadvantages also:

1. There is no "random" access. It is impossible to reach the nth item in the array without first iterating over all items up until that item. This means we have to start from the beginning of the list and count how many times we advance in the list until we get to the desired item.
2. Dynamic memory allocation and pointers are required, which complicates the code and increases the risk of memory leaks and segment faults.
3. Linked lists have a much larger overhead over arrays, since linked list items are dynamically allocated (which is less efficient in memory usage) and each item in the list also must store an additional pointer.

CHAPTER 4

IMPLEMENTATION

```
int itemId()
{ FILE *fp;
  int val=0;
  fp=fopen("itemID.txt","r");
  if(fp==NULL)
  { fp=fopen("itemID.txt","w");
    fprintf(fp,"%d",0);
    fclose(fp);
    fp=fopen("itemID.txt","r");
  }
  fscanf(fp,"%d",&val);
  fclose(fp);
  fp=fopen("itemID.txt","w");
  fprintf(fp,"%d",val+1);
  fclose(fp);
  return val+1;
}
```

```
void newItem()
{ FILE *fp;
  struct items t1;
  fp=fopen("items","a");
  t1.id=itemId();
  printf("Enter the items name\n");
  scanf("%s",&t1.name);
  printf("Enter the price\n");
  scanf("%d",&t1.price);
}
```

```

printf("Enter the quantity\n");
scanf("%d",&t1.qty);
fwrite(&t1,sizeof(t1),1,fp);
fclose(fp);
fflush(stdin);
}

```

```

void displayitems()
{ FILE *fp;
  struct items t;
  int id;
  clrscr();
  fp=fopen(fitems,"r");
  printf("|||-----|||\n");
  printf("///-----ITEMS IN MARKET-----\\n");
  printf("id\t NAME\t qty\t PRICE\n");
  printf("--\t ----\t ---\t ----\n");
  while(1)
  { fread(&t,sizeof(t),1,fp);
    if(feof(fp))
      break;
    else
    { printf("%d\t",t.id);
      printf("%s\t",t.name);
      printf("%d\t",t.qty);
      printf("%d\t\t",t.price);
    }
  }printf("_____\n");
  fclose(fp);
}

void additems()
{

```

```

struct node *temp,*p;
int id,z,y,price;
char name;
while(1)
{ clrscr();
printf("do you wanna purchase?[y=1/n=0]\n");
scanf("%d",&z);
if(z==1)
{ FILE *fp,*fp1;
struct bill t1;
struct items t2;
fp=fopen(fbill,"a");
clrscr();
displayitems();
p=(struct node*)malloc(sizeof(struct node));
fp1=fopen("items.txt","r");
printf("Enter the item-ID you need\n");
scanf("%d",&id);
t2=finditems(id);
t1.iid=t2.id;
p->id=t1.iid;
scanf("%d",&p->id);
scanf("%s",&(p->name));
printf("Enter the quantity\n");
scanf("%d",&(p->qty));
t1.iprice=t2.price;
p->price=t1.iprice;
p->link=NULL;
if(first==NULL)
{ first=p;
printf("\t--Product added--");
}
else

```

```

    { temp=first;
      while(temp->link!=NULL)
        { temp=temp->link;
          }
      temp->link=p;
      printf("\t--Product added--");
    }
    fclose(fp1);
    fclose(fp);
  }clrscr();
  printf("wanna continue?[y=1/n=0]\n");
  scanf("%d",&y);
  if(y==0)
    { break;
    }
  }
}

```

```

int deleteitems()
{ struct node *temp=first,*prev=NULL;
  int Did;
  bill();
  printf("\n--Enter the item-ID to be removed\n");
  scanf("%d",&Did);
  { if(temp->id==Did)
    {
      prev=temp;
      first=prev->link;
      temp=NULL;
    }
  else
    {
      while(temp->id!=Did)
        {

```



```

        prev = temp;
        temp = temp->link;
    }
    prev->link = temp->link;
}
free(temp);
printf("\t---Item removed  %s ---",temp->name);
}
return 0;
}

```

```

int bill()
{ struct node *temp;
  temp=first;
  if(temp==NULL)
  { printf("\t-----List is empty-----\n");
  }
  else
  {
    printf("\t-----items in the list are-----\n");
    printf("-item - quantityXprice - cost\n");
    while(temp!=NULL)
    { printf("\n%d.%s\t %dX%d\n",temp->id,temp->name,temp->qty,temp->price);
      temp->bill=(temp->qty)*(temp->price);
      printf("\t\t=%d",temp->bill);
      temp=temp->link;
    }
    printf("\nTotal cost::\t%d",total(first));
  } return 0;
}

```

```
int total(struct node *first)
{ int total=0;
  while(first!=NULL)
  { total+=first->bill;
    first=first->link;
  }
  return total;
}
```

```
struct items finditems(int id)
{ FILE *fp;
  struct items t1;
  fp=fopen(fitems,"r");
  while(1)
  { fread(&t1,sizeof(t1),1,fp);
    if(!feof(fp))
    { break;
    }
    if(id==t1.id)
    { insertcart(id);
      break;
    }
  }
  fclose(fp);
  return t1;
}
```

```
void insertcart(int id)
{ FILE *fp;
  struct items t;
  int found=0;
  fp=fopen(fitems,"r");
  while(1)
  { fread(&t,sizeof(t),1,fp);
```

```
    if(feof(fp))
        break;
    if(t.id==id)
        { found=1;
          break;
        }
    }
    if(found==0)
        { printf("out of stock");
        }
    else
        { printf("enter the item name\n");
        }
    fclose(fp);
}
```

CHAPTER 5

SAMPLE OUTPUT

~THE SCREEN WHEN ADMIN'S ACCESSING

```

||-----||
||-----WELCOME TO THE STORE-----||
||-----||

--ENTER YOUR CHOICE--
1.admin
2.user
3.Exit
1
-----
----ONLY BY ADMINS----
-----
-----

1.Newitems 2.Displayitems 3.Back
-----
---SELECT OPTION---

```

~DISPLAYING THE ITEMS STORED BY ADMIN

```

|||-----|||
///-----ITEMS IN MARKET-----\
id      NAME      qty      PRICE
--      -
1       pen       50       10
2       books     50       30
3       dress     40       1000
4       shoes     40       1500
-----

```

~WHEN THE USER ENTERS. USER HAS TO ENTER THEIR PHONE NUMBER.

```

||-----||
||-----WELCOME TO THE STORE-----||
||-----||

--ENTER YOUR CHOICE--
1.admin
2.user
3.Exit
2
Enter your phone no.

```

~USER SELECTS THE ITEM HE NEEDS AND ALSO GIVES THE QUANTITY OR THE ITEM HE NEED

```

|||-----|||
///-----ITEMS IN MARKET-----\
id      NAME      qty      PRICE
--      -
1       pen       50       10
2       books     50       30
3       dress     20       1000

Enter the item-ID you need
2
added
books
Enter the quantity
4_

```

~BILL DISPLAYS THE TOTAL AMOUNT USER HAS TAKEN

```
-----items in the list are-----  
-item - quantityXprice - cost  
  
2.books  4X30  
                =120  
3.dress  1X1000  
                =1000  
Total cost::  1120
```

~ASKS THE USER IF HE WANT TO PURCHASE

```
do you wanna purchase?[y=1/n=0]  
-
```

~TO DELETE THE ITEM

```
-----items in the list are-----  
-item - quantityXprice - cost  
  
2.books  4X30  
                =120  
3.dress  1X1000  
                =1000  
Total cost::  1120  
--Enter the item-ID to be removed  
-
```

~RESULT AFTER DELETING

```
-----items in the list are-----  
-item - quantityXprice - cost  
  
2.books  4X30  
                =120  
3.dress  1X1000  
                =1000  
Total cost::  1120  
--Enter the item-ID to be removed  
3  
        ---Item dress removed---
```

~BILL AFTER DELETING ITEMS

```
-----items in the list are-----  
-item - quantityXprice - cost  
  
2.books  4X30  
                =120  
Total cost::    120_
```


CHAPTER 6

CONCLUSION

This project make purchase in traditional manner. This project has increased and changed the way people shop. There are many goals of this project which are determined by the users. Users can save time and energy by using this kind of access.

User can easily login while at home. User can reach the number of products with different varieties and can order them quick. This project also gives the benefits of comparing products.

This project has great future to be successful. It is important to spread the awareness about this projects benefit.

I am thankful for all the support and guidance of the reviewer and other faculty members.

CHAPTER 7

BIBLIOGRAPHY

~Google.com

~geek for geek

~Stackoverflow.com