# **Shopping Management System**

***NAME: KAMALNATH S***

***REG NO: 17BCA0108***

**Introduction:**

Shopping Cart System is the Simple shopping Solution. It’s a full-featured website and shopping cart system that bends over backwards to give you the flexibility you need to online store.

The basic concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store.

The information pertaining to the products are stores on an RDBMS at the server side (store). The Server process the customers and the items are shipped to the address submitted by them. The details of the items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction.

**Problem:**

#include<iostream>

#include<string.h>

#include<conio.h>

#pragma warning(disable:4996)

#pragma warning(disable:4700)

using namespace std;

#define MAX    100

class itemEntry {

public:

       float unit\_price;

       int copies;

       char name[30];

       char company[30];

};

class Store {

public:

       int numItem;

       itemEntry database[MAX];

       Store() {

              numItem = 0;

       }

       void insertItem(char itemName[], char company[], intc, float p);

       void deleteItem(char itemName[]);

       itemEntry \*search(char itemName[]);

       void updateItem(char itemName[], int total, floatprice);

};

void Store::insertItem(char itemName[], char company[],int c, float p) {

       strcpy(database[numItem].name, itemName);

       strcpy(database[numItem].company, company);

       database[numItem].copies = c;

       database[numItem].unit\_price = p;

       cout << "Item Inserted Successfully.\n";

       ++numItem;

}

void Store::deleteItem(char itemName[]) {

       int i;

       for (i = 0; i < numItem; i++) {

              if (strcmp(itemName, database[i].name) == 0) {

                     database[i].copies--;

                     cout << "Item Deleted Successfully.\n";

                     return;

              }

       }

       cout << "Item not found.\n";

}

itemEntry \*Store::search(char itemName[]) {

       int i;

       for (i = 0; i < numItem; i++) {

              if (strcmp(itemName, database[i].name) == 0)

                     return &database[i];

       }

       return NULL;

}

void Store::updateItem(char itemName[], int total, floatprice)

{

       itemEntry \*item = search(itemName);

       if (item == NULL)

       {

              cout << "Item not found.\n";

              return;

       }

       item->copies += total;

       item->unit\_price = price;

}

int main() {

       Store sto;

       system("cls");

       char option='f', name[30], company[30], copies[10], unit\_price[30];

       while (option != 'e') {

              cout << "\n----------->Enter your choice:<------------\n";

              cout << "I for insert\n";

              cout << "D for delete\n";

              cout << "S for search\n";

              cout << "U for update\n";

              cout << "E for exit\n";

              cin.getline(name, 80);

              option = name[0];

              switch (option) {

              case 'i':

                     cout << "Enter Name of Item, Company and no of copies,Unit price  per line:\n";

                     cin.getline(name, 80);

                     cin.getline(company, 80);

                     cin.getline(copies, 80);

                     cin.getline(unit\_price, 80);

                     sto.insertItem(name, company, atoi(copies), atof(unit\_price));

                     break;

              case 'd':

                     cout << "Enter Name of Item:\n";

                     cin.getline(name, 80);

                     sto.deleteItem(name);

                     break;

              case 's':

                     cout << "Enter Name of Item:\n";

                     cin.getline(name, 80);

                     itemEntry \*test;

                     test = sto.search(name);

                     if (test != NULL) {

                           cout << "---------------->Searching Result<---------------------" << endl;

                           cout << "Item found\n" << "Name of the Item:" << test->name << endl << "Company name:" << test->company << endl << "Number of copies available:" <<test->copies << endl << "Unit price:" << test->unit\_price <<endl;

                     }

                     else

                           cout << "Item not found\n";

                     break;

              case 'u':

                     cout << "Enter details for update...\n";

                     cout << "Enter name: "; cin.getline(name, 80);

                     cout << "Enter total new entry: "; cin.getline(copies, 80);

                     cout << "Enter new price: "; cin.getline(unit\_price, 80);

                     sto.updateItem(name, atoi(copies), atof(unit\_price));

                     break;

                     /\*        case 'e':

                     exit(0);

                     break;\*/

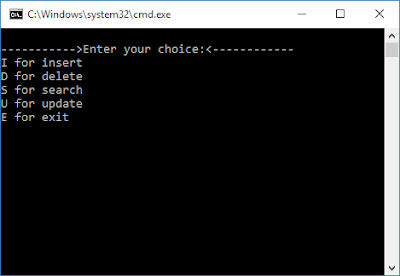
              }

       }

       return 0;

}

**Output:**

[](https://4.bp.blogspot.com/-AFHjrfvQxw0/V1vHlBl_PAI/AAAAAAAAAvk/gG6v8ZWJMec_pSYbgKLQ8Hjl55O7GUReQCLcB/s1600/Shopping+Management+System+Project+in+C++.PNG)