# **INTRODUCTION:**

This C++ program simulates a shopping experience in a store called "Mehfil-E-Shopping”. The project aims to create a user-friendly console-based shopping application that allows users to select products from various sections, including bakery, dairy, cosmetics, fruits & vegetables, and spices, and add items to a shopping cart, and generate a receipt. It includes interactive features like voice narration using Microsoft's Speech API (SAPI), and manages the shopping process, including item selection, cart management, tax calculation, and discounts.

# **OBJECTIVE:**

The main objectives of this program are:

**User Interaction:** To create a console-based application that allows user to select items from different sections and specify the quantity.

**Voice Feedback:** To implement a voice feedback system which can make interaction with user stronger and provide guidance to the user through voice and this is done by using Microsoft’s SAPI.

**Receipt Generation:** In order to generate a detailed receipt for the user which includes purchased items name, their quantity, prices, and total amount. Including applicable taxes and discounts

**Tax and Discount Calculation:** To calculate the GST on the total purchase amount & apply discounts based on predefined criteria.

**Data Management:** To manage and store transaction data in a structured format for easy retrieval & display.

# **SCOPE:**

Our console-based shopping app lets users to buy from 5 categories: Bakery, Dairy, Cosmetics, Fruits & Vegetables and Spices

**Key Features:**

* Advance input validation.
* Voice feedback (Microsoft Speech API).
* Detailed receipt generation (summary, taxes, discounts).
* Efficient transaction data management.

**Future Enhancements:** GUI & online payment integration.

# **PROGRAM**

#include <iostream>

#include <string>

#include <iomanip>

#include <windows.h>

#include <ctime>

#include <sapi.h>

using namespace std;

//global variables

string Receipt[100][4];

int a = 0; //iterator for generate receipt

void voice(wstring input) //block of code for voiceover

{

    {

        ISpVoice\* pVoice = NULL;

        HRESULT hr;

        // Initialize COM library

        hr = CoInitializeEx(NULL, COINIT\_APARTMENTTHREADED);

        if (FAILED(hr))

        {

            cout << "ERROR: Failed to initialize COM library.\n";

        }

        // Create SAPI voice instance

        hr = CoCreateInstance(CLSID\_SpVoice, NULL, CLSCTX\_ALL, IID\_ISpVoice, (void\*\*)&pVoice);

        if (SUCCEEDED(hr))

        {

            // Predefined text to speak

            hr = pVoice->Speak(input.c\_str(), 0, NULL);

            pVoice->Release();

            pVoice = NULL;

        }

        else

        {

            cout << "ERROR: Failed to create voice instance.\n";

        }

        CoUninitialize();

    }

}

void welcome()

{

    system("Color F4"); // F is for background, 4 is for text according to chart

    cout << " "; // space in the start

    for (int I = 1; I <= 65; I++)

    {

        cout << '\*';

        Sleep(20); // 20 milliseconds gap between each asteric

    }

    cout << endl;

    cout << " ";

    cout << " !!!--------";

    Sleep(13);

    cout << 'W';

    Sleep(13);

    cout << 'E';

    Sleep(13);

    cout << 'L';

    Sleep(13);

    cout << 'C';

    Sleep(13);

    cout << 'O';

    Sleep(13);

    cout << 'M';

    Sleep(13);

    cout << 'E';

    Sleep(13);

    cout << ' ';

    Sleep(13);

    cout << 'T';

    Sleep(13);

    cout << 'O';

    Sleep(13);

    cout << ' ';

    Sleep(13);

    cout << 'M';

    Sleep(13);

    cout << 'E';

    Sleep(13);

    cout << 'H';

    Sleep(13);

    cout << 'F';

    Sleep(13);

    cout << 'I';

    Sleep(13);

    cout << 'L';

    Sleep(13);

    cout << '-';

    Sleep(13);

    cout << 'E';

    Sleep(13);

    cout << '-';

    Sleep(13);

    cout << 'S';

    Sleep(13);

    cout << 'H';

    Sleep(13);

    cout << 'O';

    Sleep(13);

    cout << 'P';

    Sleep(13);

    cout << 'P';

    Sleep(13);

    cout << 'I';

    Sleep(13);

    cout << 'N';

    Sleep(13);

    cout << 'G';

    Sleep(13);

    cout << ' ';

    Sleep(13);

    cout << "--------!!!";

    cout << endl;

    cout << " ";

    for (int I = 1; I <= 65; I++)

    {

        cout << '\*';

        Sleep(20);

    }

    cout << endl;

    wstring I = L"Welcome to Mehfil-a-Shopping"; // L is prespecified protocol for voiceover as we used wstring, without using L, error occurs

    voice(I); // function call for voiceover

}

void GenerateReceipt(string name, int price, double quantity, double total)

{

    Receipt[a][0] = name;

    Receipt[a][1] = to\_string(price); // to\_string is a built-in function to convert any data type to string, our array is in string therefore we converted each data type into string

    Receipt[a][2] = to\_string(quantity);

    Receipt[a][3] = to\_string(total);

    a++;

}

void DisplayReceipt()

{

    cout << left << setw(20) << "Item Name" << setw(15) << "Price" << setw(15) << "Quantity" << setw(15) << "Total" << endl;

    cout << "------------------------------------------------------------" << endl;

    string st;

    for (int I = 0; I < a; I++) // a which is used for storing elements

    {

        for (int j = 0; j < 4; j++)

        {

            if (j <= 1)

            {

                cout << Receipt[I][j]; // displaying the receipt elements

                cout << " ";

            }

            else if (j == 2) {

                st = Receipt[I][j]; // variable to display receipt elements as the conversion of double to string causes unwanted decimal points

                for (int k = 0; k < 4; k++)

                {

                    cout << st[k]; // quantity display

                }

                cout << " ";

            }

            else if (j == 3) {

                st = Receipt[I][j];

                for (int l = 0; l < 7; l++)

                {

                    cout << st[l]; // total price display

                }

            }

        }

        cout << endl;

    }

    cout << "------------------------------------------------------------" << endl;

}

void Bakery(double& T)

{

    string bakeryItems[8] = { "Bread", "Biscuits", "Buns", "Rusk", "Patties", "Fruit Cake", "Deal A", "Deal B" };

    int pricesB[8] = { 120,500,50,200,80,250,500,450 };

    string weightage[8] = { "pack","kg","piece","pack","piece","pack","amount","amount" };

    int choice;

    double quantity;

    double calculation;

    char ch;

    do {

        cout << "Select Bakery Items (Press 1 to 8 to select item) : " << endl;

        for (int I = 0; I < 6; I++)

        {

            cout << I + 1 << ". " << bakeryItems[I] << " - Rs." << pricesB[I] << " per " << weightage[I] << endl; // I is for number display as original array value of I is 0

        }

        cout << endl;

        cout << "7. Deal A : (Bread + Fruit Cake + Rusk) - Rs.500 \"Save Rs.70\":" << endl;

        cout << "8. Deal B : (4 Buns + 4 Patties) - Rs.450 \"Save Rs.70\":" << endl;

        do { //check loop

            cout << endl;

            cin >> choice;

            if (choice <= 0 || choice > 8) // for displaying error

            {

                cout << "Enter the valid input for items between 1 to 8 !!" << endl;

            }

        } while (choice <= 0 || choice > 8); // for actual detection and carrying out of error

        choice--; // for array value matching

        cout << endl;

        cout << "Kindly Enter the Quantity of " << bakeryItems[choice] << " in (" << weightage[choice] << ") : ";

        cin >> quantity;

        calculation = quantity \* pricesB[choice];

        T += calculation; // T is passed by reference variable to store total amount so we can see the change in main function as well

        cout << "Item successfully added to cart!!" << endl;

        GenerateReceipt(bakeryItems[choice], pricesB[choice], quantity, calculation);

        cout << endl;

        cout << "Do you want to buy another item from this section. Press (y) for yes and (n) for No.: ";

        cin >> ch;

        cout << endl;

        system("cls");

    } while (ch == 'y' || ch == 'Y');

}

void Dairy(double& T)

{

    string dairyItems[8] = { "Eggs", "Milk", "Yogurt", "Cheese", "Desi Ghee", "Butter" ,"Deal A", "Deal B" };

    int pricesD[8] = { 31,200,220,500,1000,1000,510,1800 };

    string weightage[8] = { "piece","kg","kg","pack","kg","kg","amount","amount" };

    int choice;

    double quantity;

    double calculation;

    char ch;

    do {

        cout << "Select Dairy Items (Press 1 to 8 to select item) : " << endl;

        for (int I = 0; I < 6; I++)

        {

            cout << I + 1 << ". " << dairyItems[I] << " - Rs." << pricesD[I] << " per " << weightage[I] << endl;

        }

        cout << endl;

        cout << "7. Deal A : (12 Eggs + 1kg Milk) - Rs.510 \"Save Rs.62\":" << endl;

        cout << "8. Deal B : (1kg Desi Ghee + 1kg Butter) - Rs.1800 \"Save Rs.200\":" << endl;

        do { //check loop

            cout << endl;

            cin >> choice;

            if (choice <= 0 || choice > 8)

            {

                cout << "Enter the valid input for items between 1 to 8 !!" << endl;

            }

        } while (choice <= 0 || choice > 8);

        choice--;

        cout << endl;

        cout << "Kindly Enter the Quantity of " << dairyItems[choice] << " in (" << weightage[choice] << ") : ";

        cin >> quantity;

        calculation = quantity \* pricesD[choice];

        T += calculation;

        cout << "Item successfully added to cart!!" << endl;

        GenerateReceipt(dairyItems[choice], pricesD[choice], quantity, calculation);

        cout << endl;

        cout << "Do you want to buy another item from this section. Press (y) for yes and (n) for No.: ";

        cin >> ch;

        cout << endl;

        system("cls");

    } while (ch == 'y' || ch == 'Y');

}

void Cosmetics(double& T)

{

    string cosmeticsItems[6][4] = { {"Shampoo","Pantene","Sunsilk","Meclay"},

                                    {"Face Wash","Ponds","Garnier","Loreal"},

                                    {"Moisturizer","Vaseline","Johnsons","Nivea"},

                                    {"Conditioner","Pantene","Sunsilk","Meclay"},

                                    {"Body Spray","Axe","Fogg","Bold"},

                                    {"Soaps","Dove","Safeguard","Harmony"}

    }; // 6 is the total items and 4 is including the main item and brands.

    int pricesC[18] = { 320,300,350,500,550,700,450,750,800,350,330,380,500,480,450,350,175,140 };

    string weightage[6] = { "bottle","piece","bottle","bottle","spray","piece" };

    int choice;

    int brandChoice;

    double quantity;

    double calculation;

    char ch;

    int helper;

    int selector;

    do {

        cout << "Select Cosmetic Items (Press 1 to 6 to select item): " << endl;

        for (int I = 0; I < 6; I++)

        {

            cout << I + 1 << ". " << cosmeticsItems[I][0] << endl; // first basic elements

        }

        do { //check loop

            cout << endl;

            cin >> choice;

            if (choice <= 0 || choice > 6)

            {

                cout << "Enter the valid input for items between 1 to 6 !!" << endl;

            }

        } while (choice <= 0 || choice > 6);

        choice--;

        selector = choice \* 3; // jump of 3 to reach array value and to store the price

        helper = choice \* 3; // to display price of brand

        // SELECTOR AND HELPER ARE USED TO ACCESS PRICE OF THE ITEMS IN PRICE ARRAY

        cout << endl;

        system("cls");

        cout << "Now Select the Brand of your Item : " << endl;

        for (int I = 1; I < 4; I++) // to print 3 brand names

        {

            cout << I << ". " << cosmeticsItems[choice][I] << " - Rs." << pricesC[helper] << " per " << weightage[choice] << endl;

            helper++;

        }

        do { //check loop

            cout << endl;

            cin >> brandChoice;

            if (brandChoice <= 0 || brandChoice > 3)

            {

                cout << "Enter the valid input for items between 1 to 3 !!" << endl;

            }

        } while (brandChoice <= 0 || brandChoice > 3);

        cout << endl;

        cout << "Kindly Enter the Quantity of " << cosmeticsItems[choice][brandChoice] << " in (" << weightage[choice] << ") : ";

        cin >> quantity;

        brandChoice--;

        selector = selector + brandChoice; // to access the exact price in array of the elements

        calculation = quantity \* pricesC[selector];

        T += calculation;

        cout << "Item successfully added to cart!!" << endl;

        cout << endl;

        GenerateReceipt(cosmeticsItems[choice][brandChoice], pricesC[selector], quantity, calculation); // choice and brandchoice for items access and selector for price

        cout << "Do you want to buy another item from this section. Press (y) for yes and (n) for No.: ";

        cin >> ch;

        cout << endl;

        system("cls");

    } while (ch == 'y' || ch == 'Y');

}

void Fruits\_Veges(double& T)

{

    string fruitsVegItems[6] = { "Tomato", "Potato", "Onion", "Apple", "Bananas", "Oranges" };

    int pricesFV[6] = { 120,100,200,210,15,20 };

    string weightage[6] = { "kg","kg","kg","kg","piece","piece" };

    int choice;

    double quantity;

    double calculation;

    char ch;

    do {

        cout << "Select Fruits/Vegetables (Press 1 to 6 to select item): " << endl;

        for (int I = 0; I < 6; I++)

        {

            cout << I + 1 << ". " << fruitsVegItems[I] << " - Rs." << pricesFV[I] << " per " << weightage[I] << endl;

        }

        do { //check loop

            cout << endl;

            cin >> choice;

            if (choice <= 0 || choice > 6)

            {

                cout << "Enter the valid input for items between 1 to 6 !!" << endl;

            }

        } while (choice <= 0 || choice > 6);

        choice--;

        cout << endl;

        cout << "Kindly Enter the Quantity of " << fruitsVegItems[choice] << " in (" << weightage[choice] << ") : ";

        cin >> quantity;

        calculation = quantity \* pricesFV[choice];

        T += calculation;

        cout << "Item successfully added to cart!!" << endl;

        cout << endl;

        GenerateReceipt(fruitsVegItems[choice], pricesFV[choice], quantity, calculation);

        cout << "Do you want to buy another item from this section. Press (y) for yes and (n) for No.: ";

        cin >> ch;

        cout << endl;

        system("cls");

    } while (ch == 'y' || ch == 'Y');

}

void Spices(double& T)

{

    string spicesItems[6] = { "Sugar", "Tea", "Salt" ,"Red Chilli Powder", "Turmeric", "Garam Masala" };

    int pricesS[6] = { 120,1700,100,250,250,250 };

    string weightage[6] = { "kg","kg","kg","pack","pack","pack" };

    int choice;

    double quantity;

    double calculation;

    char ch;

    do {

        cout << "Select Spices Items (Press 1 to 6 to select item) : " << endl;

        for (int I = 0; I < 6; I++)

        {

            cout << I + 1 << ". " << spicesItems[I] << " - Rs." << pricesS[I] << " per " << weightage[I] << endl;

        }

        do { //check loop

            cout << endl;

            cin >> choice;

            if (choice <= 0 || choice > 6)

            {

                cout << "Enter the valid input for items between 1 to 6 !!" << endl;

            }

        } while (choice <= 0 || choice > 6);

        choice--;

        cout << endl;

        cout << "Kindly Enter the Quantity of " << spicesItems[choice] << " in (" << weightage[choice] << ") : ";

        cin >> quantity;

        calculation = quantity \* pricesS[choice];

        T += calculation;

        cout << "Item successfully added to cart!!" << endl;

        cout << endl;

        GenerateReceipt(spicesItems[choice], pricesS[choice], quantity, calculation);

        cout << "Do you want to buy another item from this section. Press (y) for yes and (n) for No.: ";

        cin >> ch;

        cout << endl;

        system("cls");

    } while (ch == 'y' || ch == 'Y');

}

int main() {

    welcome();

    wstring username; // customer name and voiceover is accepted by only wstring

    cout << "\n\nKindly Enter Your Name: ";

    getline(wcin, username); // as we want the name to be used in voiceover, we use wcin instead of normal cin

    system("cls");

    system("Color F1"); // F is bright white background, 1 is blue text

    char cont;

    string sections[5] = { "Bakery", "Dairy", "Cosmetics", "Fruits & Vegs", "Spices" };

    double total = 0.0; // this variable is passed by reference in every function of the section

    int sectionChoice;

    cout << "\n Salaam! ";

    wcout << username << endl << endl;

    wstring I = L"Salaam!"; // L prefix indicates that the its a wide string

    voice(I);

    voice(username);

    I = L"Everything you need is available here";

    voice(I);

    do {

        cout << "Please select a section:\n";

        for (int I = 0; I < 5; I++) {

            cout << I + 1 << ". " << sections[I] << endl;

        }

        do { //check loop

            cout << endl;

            cin >> sectionChoice;

            if (sectionChoice <= 0 || sectionChoice > 5)

            {

                cout << "Enter the valid input for Section between 1 to 5 !!" << endl;

            }

        } while (sectionChoice <= 0 || sectionChoice > 5);

        system("cls");

        switch (sectionChoice)

        {

        case 1:

            cout << endl;

            Bakery(total);

            break;

        case 2:

            cout << endl;

            Dairy(total);

            break;

        case 3:

            cout << endl;

            Cosmetics(total);

            break;

        case 4:

            cout << endl;

            Fruits\_Veges(total);

            break;

        case 5:

            cout << endl;

            Spices(total);

            break;

        }

        cout << "Do you want to continue shopping? (y/n): ";

        cin >> cont;

        system("cls");

    } while (cont == 'y' || cont == 'Y');

    double gst = total \* 0.18;

    double finalTotal = total + gst;

    double discount;

    time\_t now = time(NULL); // time t is a data type which is used to display date and time, null is a function

    char str[26] = {}; // to display date and time in characters

    ctime\_s(str, 26, &now); // converter from hexadecimals into characters

    system("Color 75"); // 7 is white, 5 is purple

    cout << "\n MEHFIL-E-SHOPPING\n";

    cout << " -----------";

    cout << "\n RECEIPT\n";

    cout << " -----------" << endl << endl;

    cout << " Customer Name: ";

    wcout << username << endl;

    cout << " Time: " << str << endl;

    DisplayReceipt();

    cout << " Total: Rs." << total << endl;

    cout << " GST (18%): Rs." << gst << endl;

    cout << " -----------------------" << endl;

    cout << " Total after tax:Rs." << finalTotal << endl;

    if (finalTotal >= 5000)

    {

        cout << " You availed 5 percent discount!!" << endl;

        discount = finalTotal \* 0.05;

        finalTotal -= discount;

        cout << " -----------------------" << endl;

        cout << " Total Amount after Discount : Rs." << finalTotal << endl;

    }

    cout << endl << " THANKS FOR SHOPPING!!" << endl;

    I = L"Thanks for shopping! ALLAH HAFIZ";

    voice(I);

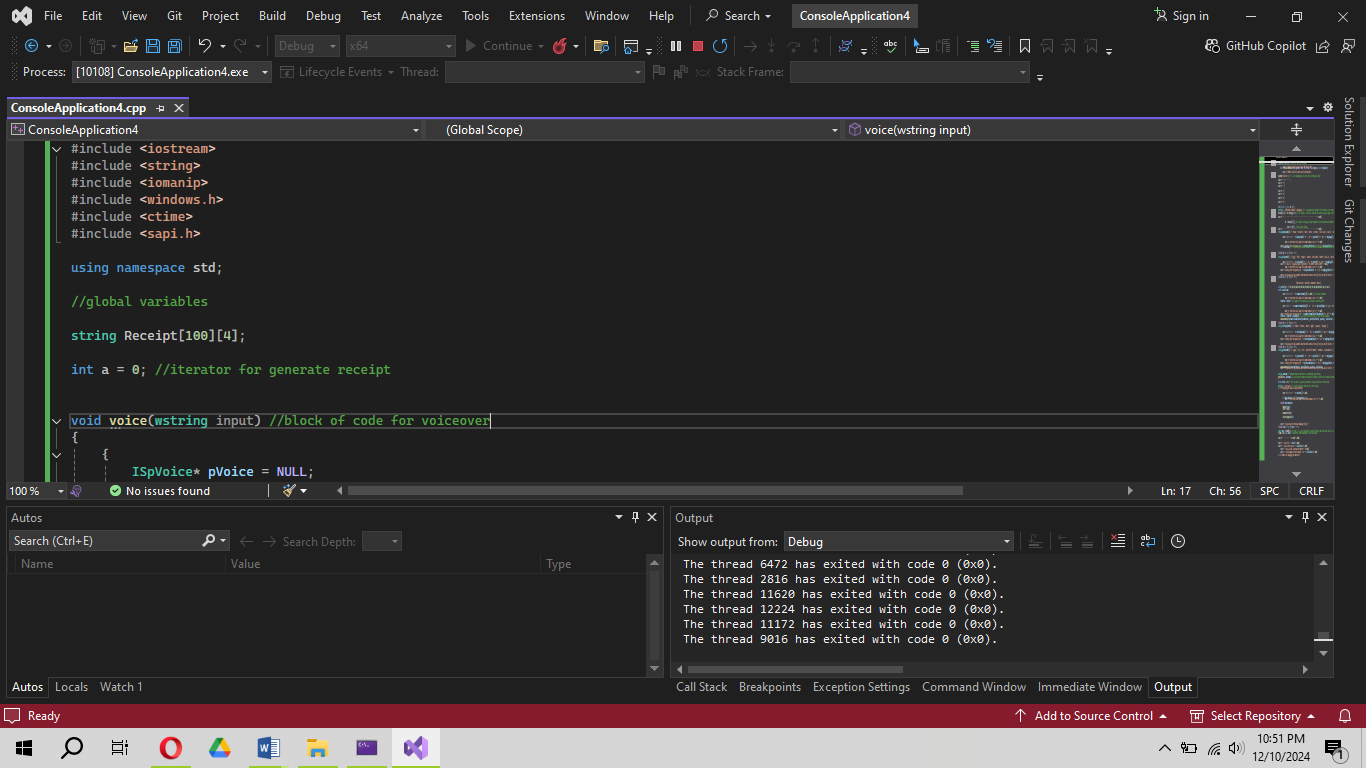
    return 0;

}

# **The Foundation: Global Variables and Functions**

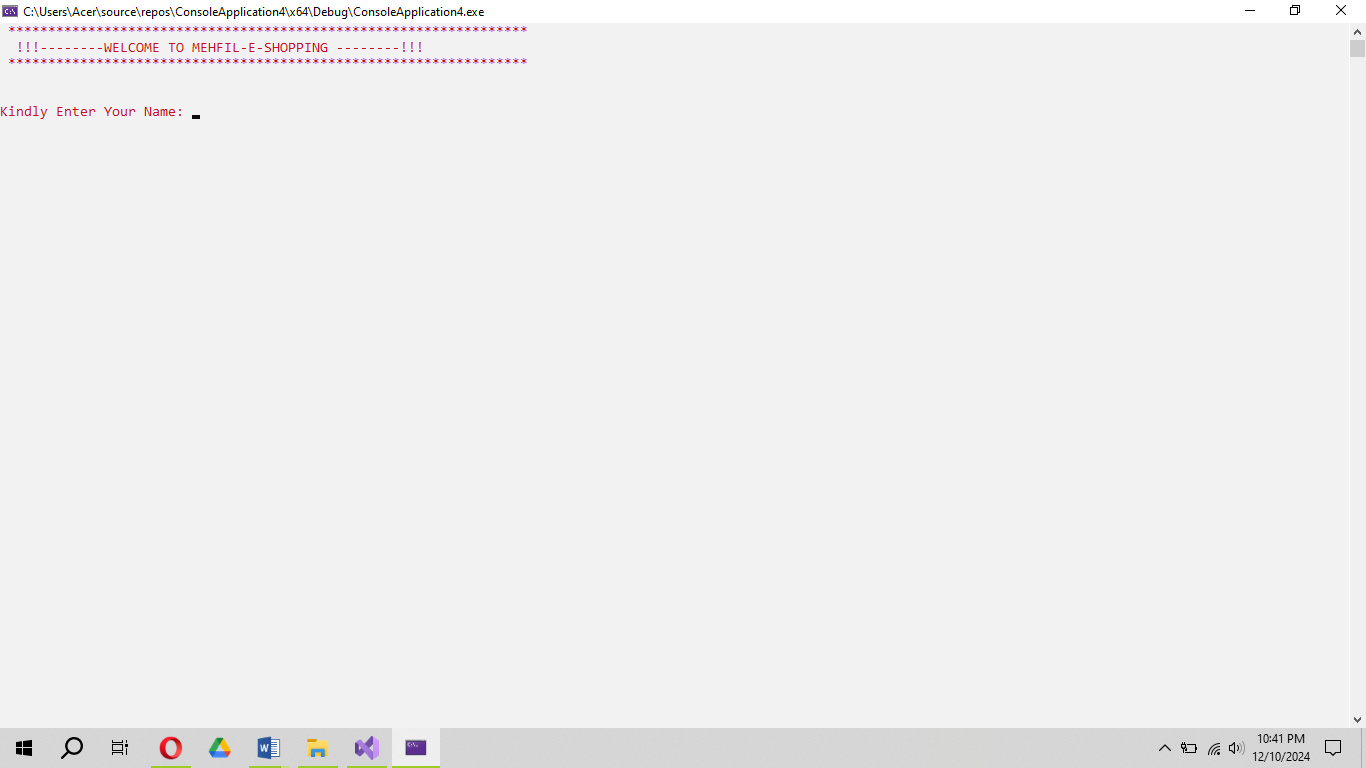
# **Global Variables:**

The program utilizes a global 2D array “Receipt”, to store details of selected items. Each entry in the array represents an item, holding information such as name, price, quantity and total cost. And index counter “a” keeps track of the number of items added to the shopping cart.



# **Welcome Message:**

“voice(wstring input)”: This function leverages Microsoft’s SAPI to convert text into spoken output. The welcome() function displays a visually engaging welcome message with animation and uses “voice()” to narrate a greeting.



# **Generate and Display Receipt Function:**

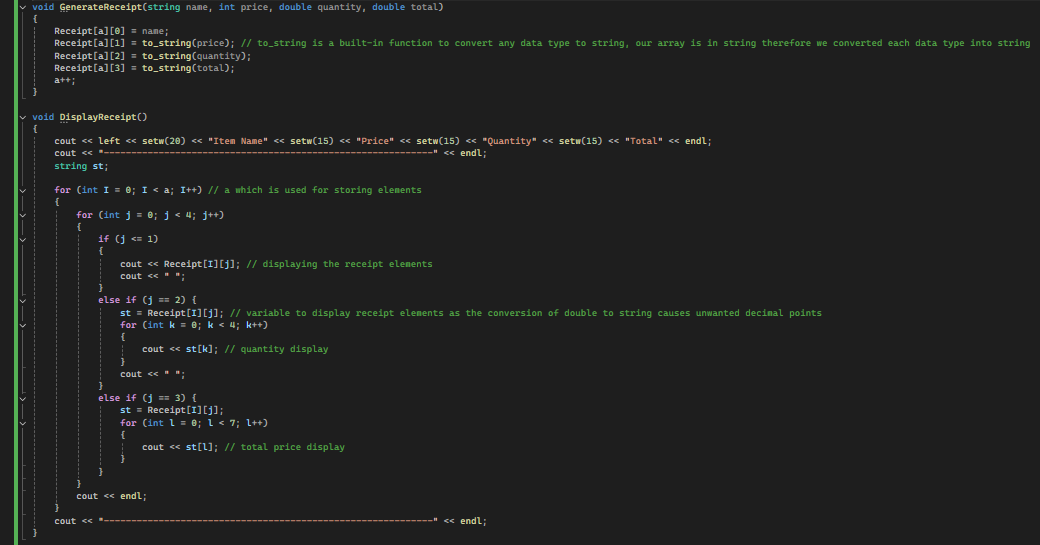
The “generateReceipt” function is responsible for giving complete summary of user purchase from all the 5 sections and their individual prices, quantity and total price, including any applicable taxes.

**How function works:**

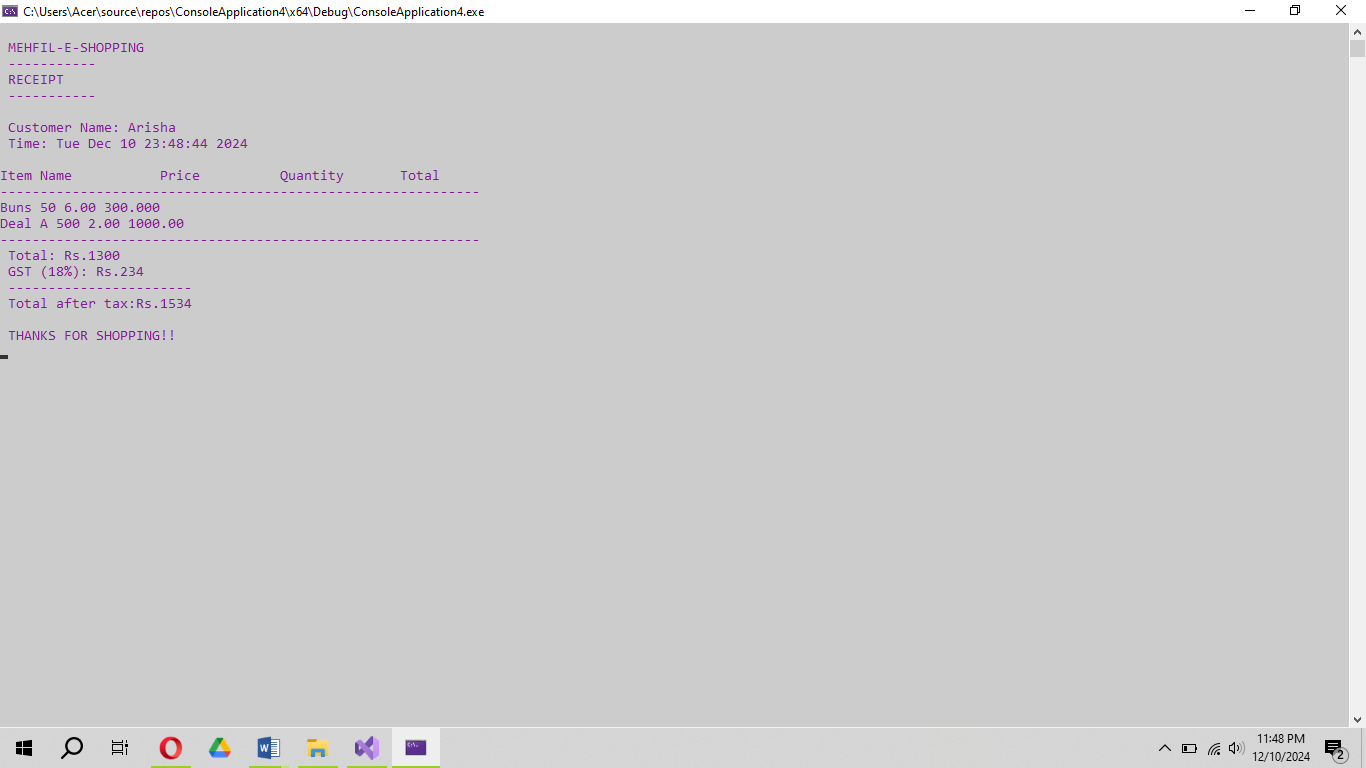
1. The function begins by accessing the shopping cart, which is selected by the user along with their quantity and prices.
2. It calculated the subtotal of the items and apply any applicable taxes (e.g., GST ) and generate the total amount due.
3. The function formats the receipt for Readability
4. The output is formatted using setw to align the columns.

**Display Receipt:** Finally, the formatted receipt is printed to the console, providing the user clear summary of their purchase.

**Input:**



**Output:**



# **Product Sections (Bakery, Dairy, Cosmetics, Fruits & Veges, Spices):**

Each section corresponds to a product category, where the user can select items and add them to the cart.

Each function (Bakery, Dairy, Cosmetics, Fruits & Veges, Spices) presents a list of products, and the user selects items from the list.

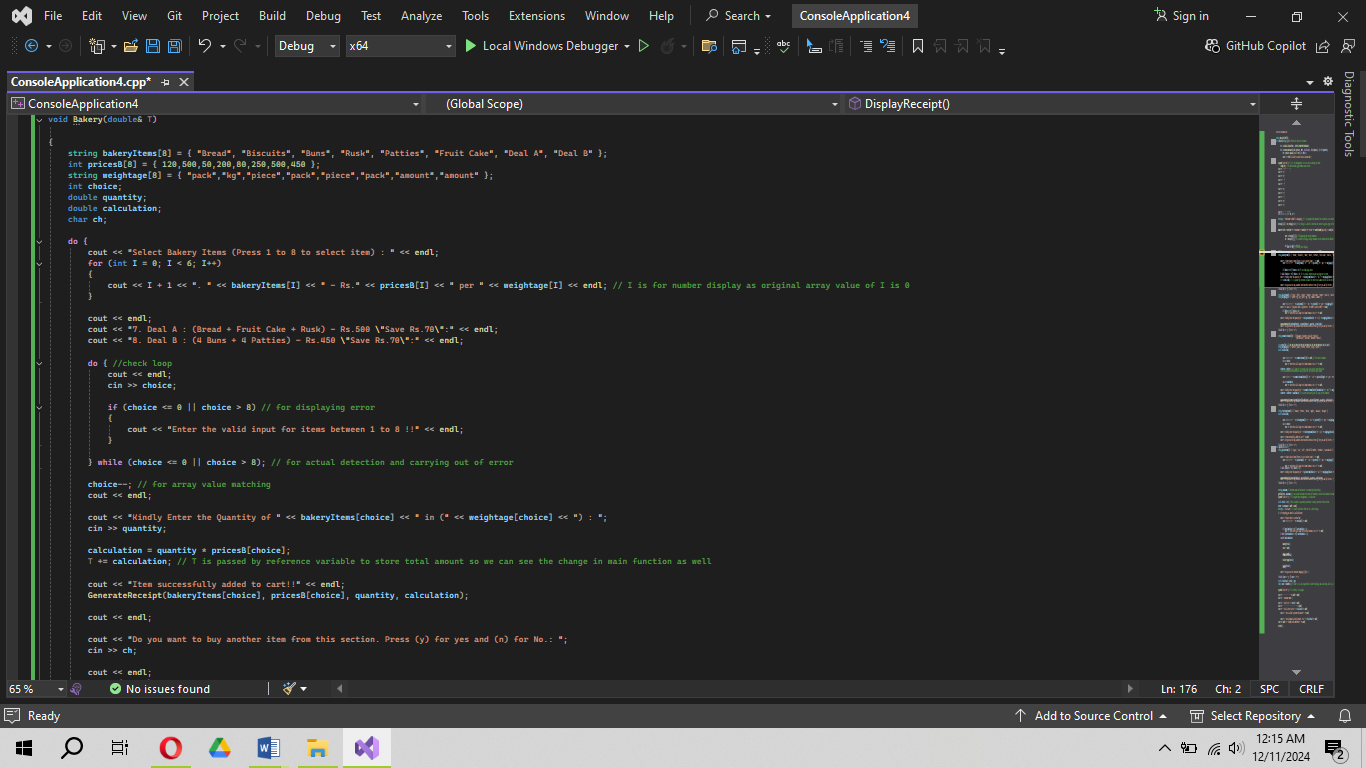
The user is prompted to enter the quantity of the selected item, and the total price is calculated.

# **Bakery Function:**

It allows user to select item from the given list and enter their quantity and tell if they want to continue shopping from this section.

# **How it works:**

1. The function begins by showing the list of available bakery items. Each item is displayed with its price and brief description to help user.
2. After displaying the list the user is asked to select an item by entering its corresponding number. Check loop is used to ensure the user selection is valid.
3. After selecting item, user have to select its quantity.
4. The function adds the chosen item into the shopping cart.
5. Then user is asked if they want to continue shopping from this section or not.

**Input:**

**Output:**

# **Main Function:**

The “main()” function is the entry point of the code. It initializes the program, handles user interactions, and manages the flow of shopping experience.

# **Detailed Breakdown of the Main Function:**

***Welcome Message:*** The program starts by calling welcome(), which displays the welcome message and plays the voice greeting.

***User Name Input:*** The program prompts the user to enter their name, which is then spoken aloud using the voice() function.

***Section Selection:*** The user is presented with a menu to select a shopping section (Bakery, Dairy, Cosmetics, Fruits & Vegs, Spices). The program validates the user's input to ensure it's within the range of available sections.

***Loop for Shopping:*** The user can continue shopping by selecting items from various sections. The shopping loop runs until the user decides to stop (by entering 'n' or 'N' when asked if they want to continue shopping).