**Restaurant Management System**

GitHub : https://github.com/Mustufa-Ismail/Resturant\_Management\_System.git

# Abstract:

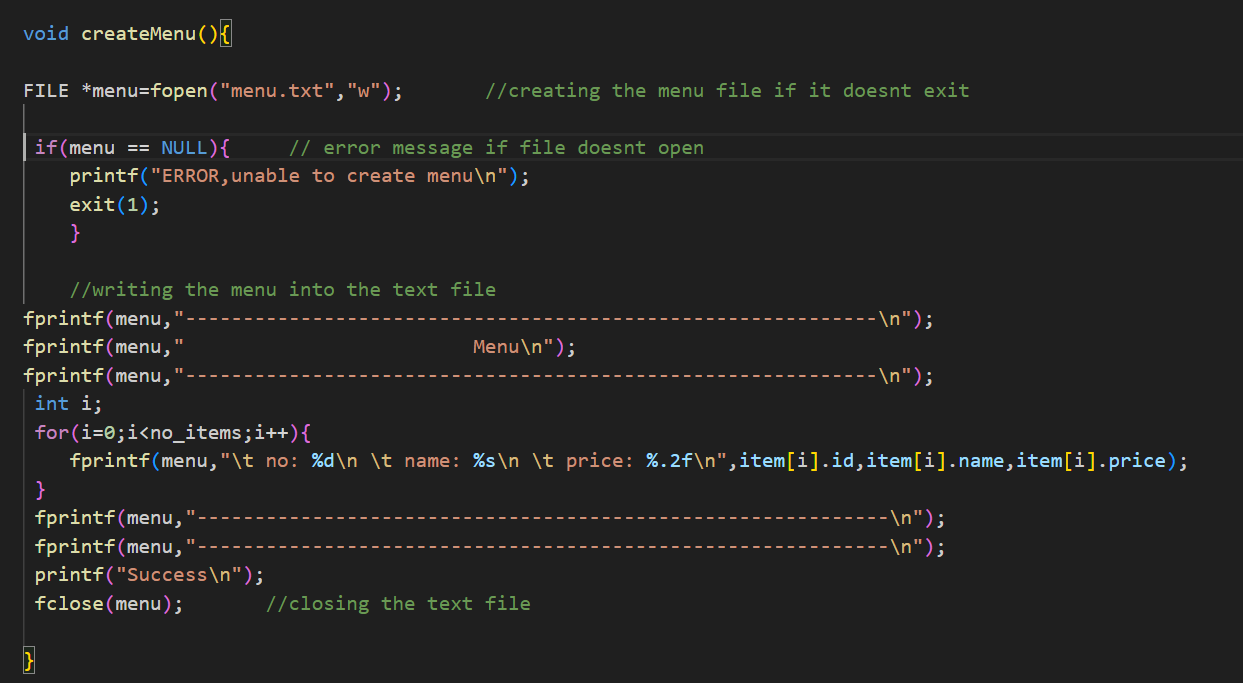
*This report presents the design and implementation of a menu management system for a restaurant, aimed at providing an intuitive user interface for both management and customer interaction. The system is structured into two main modules: Management and Customer, each offering different functionalities to enhance the restaurant's operations. The Management module allows authorized users to view, add, update, and remove menu items, as well as manage customer feedback. The Customer module provides users with the ability to place orders, review the menu, and give feedback on their experience.*

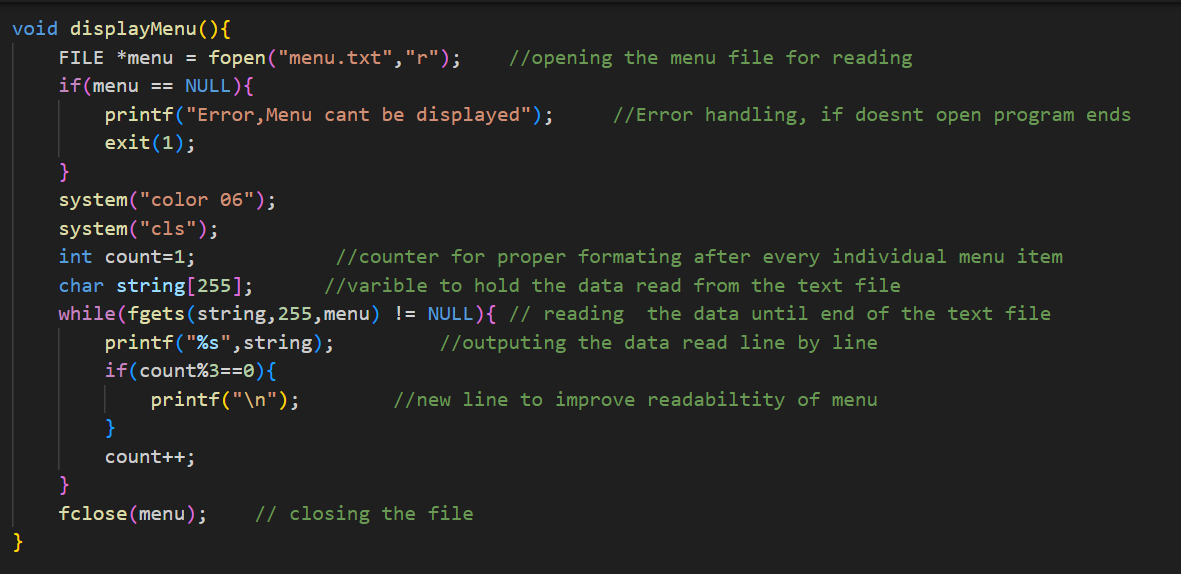
*Key features of the system include a tax-inclusive payment system, customer feedback management, and options for order customization. The system integrates payment options, including card and cash payments, with built-in discounts based on order value and card type. Additionally, the software uses a file-based storage mechanism for saving and displaying menu items, customer orders, and feedback.*

*The system employs robust input validation, error handling, and dynamic menu adjustments to accommodate changing menu items. A well-structured design and modular approach ensure scalability, with support for up to 50 menu items and efficient order processing.*

*This report details the functionalities implemented, the system design, and challenges encountered during the development, along with suggestions for future improvements. The system is intended to streamline restaurant operations, improve customer satisfaction, and provide real-time feedback for better service quality.*

## Main Functionality of Code:

Create Menu Function: It is called at the start of the menu function to create the menu text file and to write the initial menu.

Display menu function: This function is used for displaying the data written in the menu text file, it also clears the screen and changes the color of the text on the screen to yellow.

A computer screen shot of text

Description automatically generated

Save menu function: This writes the data menu item structure array to menu file

A computer screen shot of text

Description automatically generated

A computer code on a black background

Description automatically generated

Add item function: This asks for no of items to be updated, then runs a loop that many times. At each iteration, it asks name and price of the item to be added and adds it to at the end of the menu item’s structure array. No of items are limited to 50. There are checks in place which will not let the user enter more than 50 items in the menu.

A screenshot of a computer program

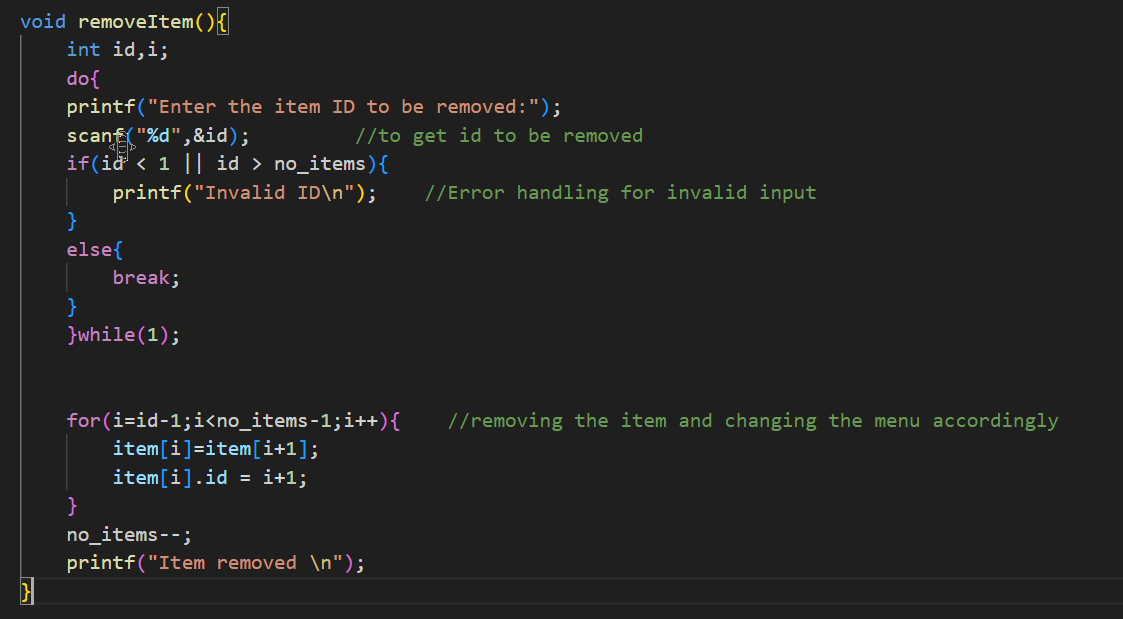
Description automatically generated

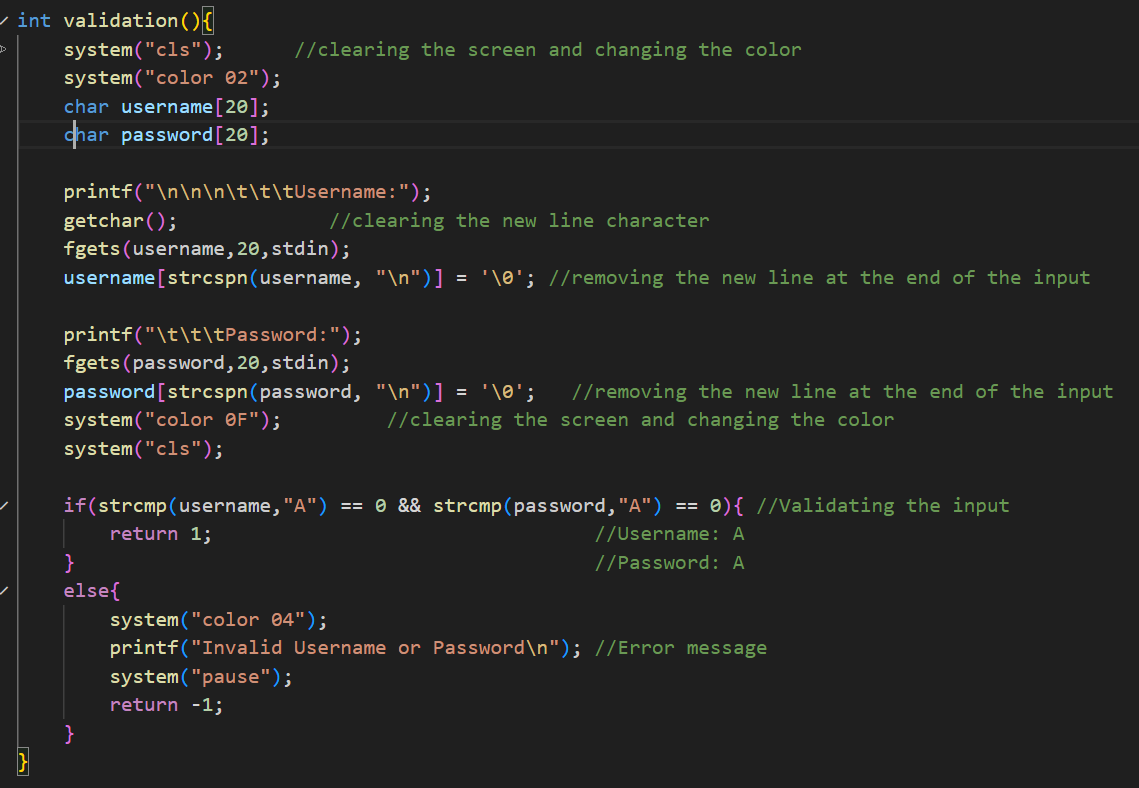
A screen shot of a computer program

Description automatically generated

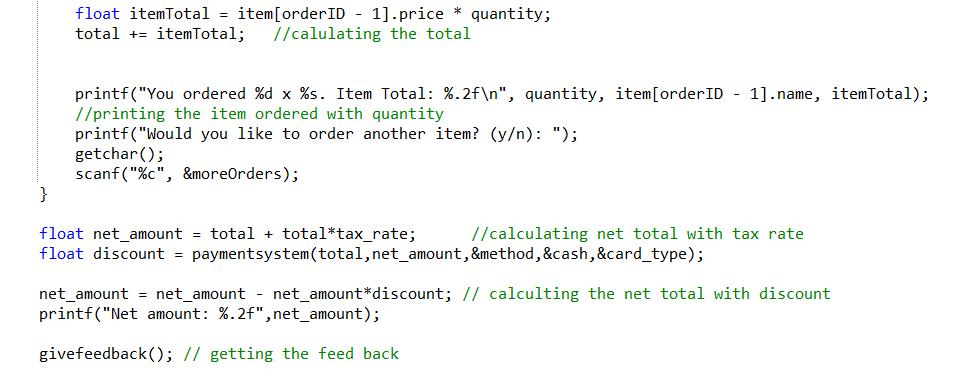
Update item function: This function asks for the id of the item to be updated, it also checks if it is valid or not. Then it presents a menu to ask what does the user wants to update. The menu has a option to exit it.

Remove item function: it asks for the id of the item to be removed and validates it. Then it deletes the items and adjusts the rest of the menu accordingly.

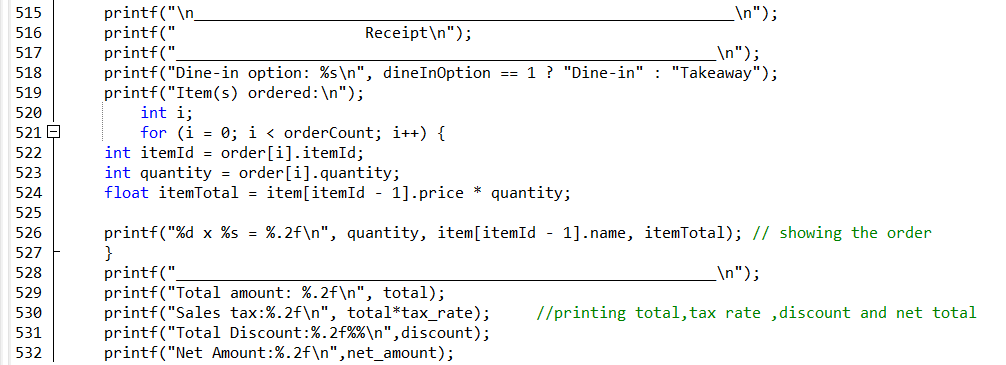




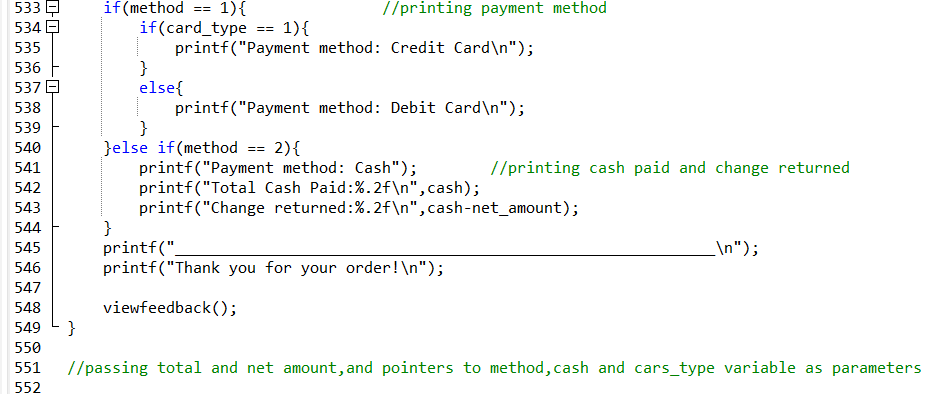
Validation Function: This function is called when the user accesses the management menu, it prompts the user to enter the username and password, if it matches the username and password set in the code then the access is granted else the access is denied. The text turns green when the function is called and turns red when invalid username or Password it entered.

**

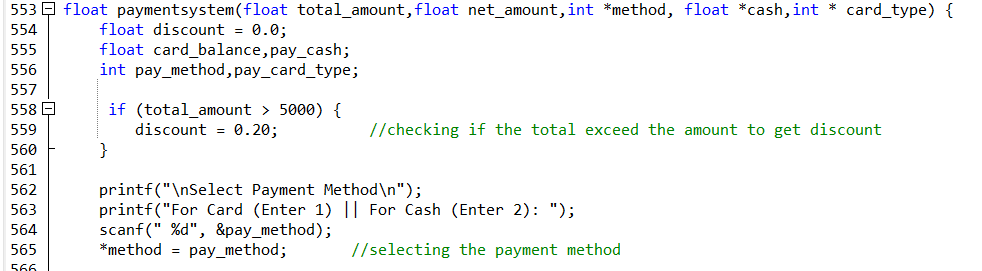
* *Have used* ***Float*** *(as the price of item can be in decimal so the* ***itemTotal*** *same for the* ***net\_amount*** *and* ***discount*** *as well).*
* *Used* ***Scanf*** *to ask from the user.*
* *Using* ***Printf*** *for printing the Net amount.*
* *Used* ***getchar*** *for buffer handling.*
* ***Givefeedback()*** *is already explained. Used it to in a function to connect within* ***takeorder*** *function  
  attach Ss of this function and explain.*



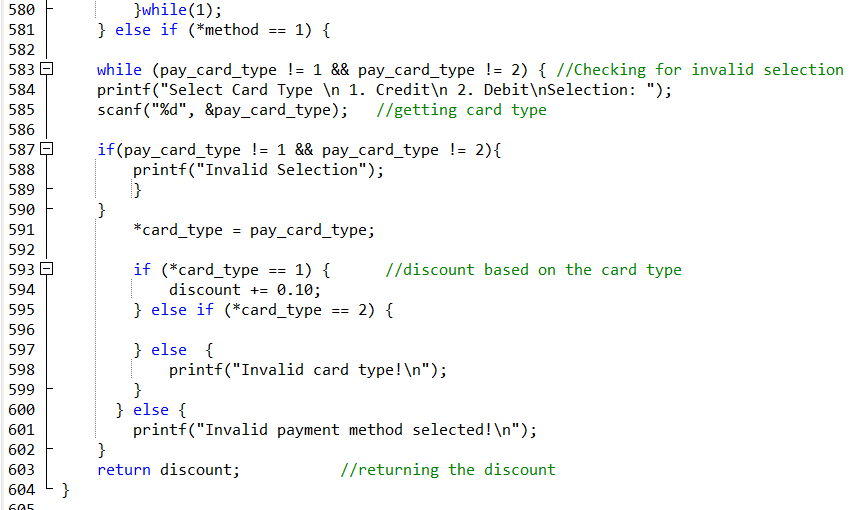
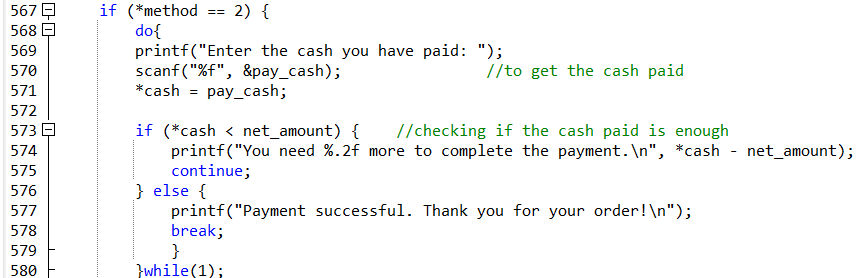
* *Printing the receipt of the order*



* *Using* ***If-else-if*** *for payment method.*
* *Used* ***nested if-else*** *in method 1 for further asking for card type. Credit or Debit.*

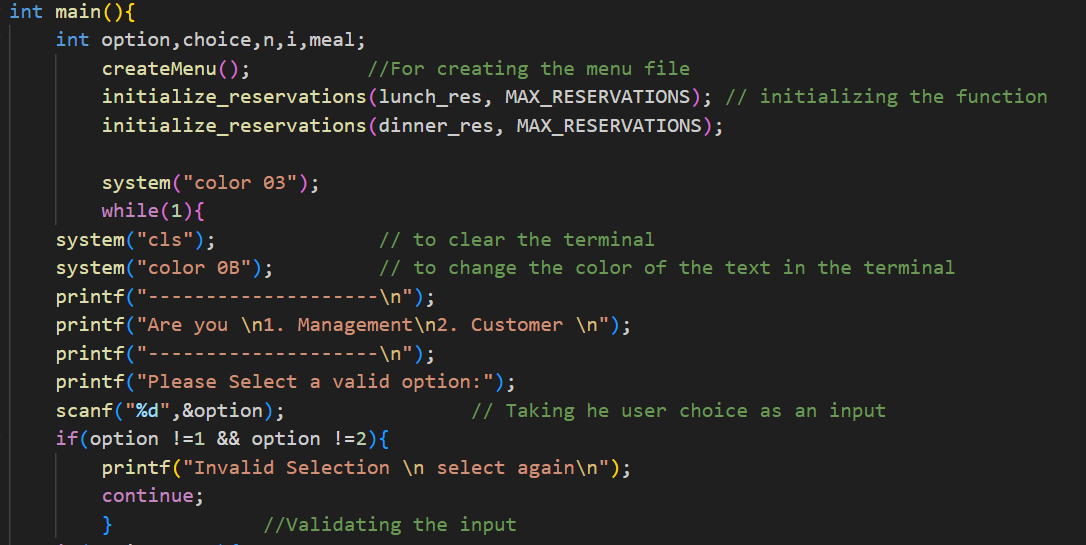
**

* *Using* ***pointers*** *to connect the* ***\*method,\*cash,\*card\_type*** *from a different function. In my case I am connection* ***method,cash,card\_type*** *from function* ***takeorder*** *to function* ***paymentsystem*** *so the total\_amount can be connected with the other function.*

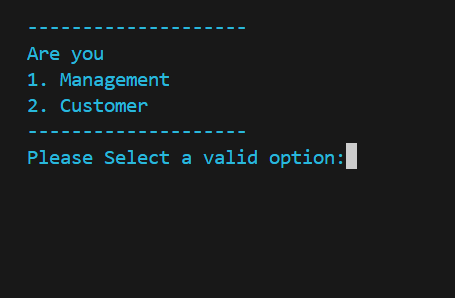
**

* *Using* ***nested if-else*** *with* ***pointer*** *to connect the order with the* ***cash system****. And giving discounts which will be applied to the order’s bill.*

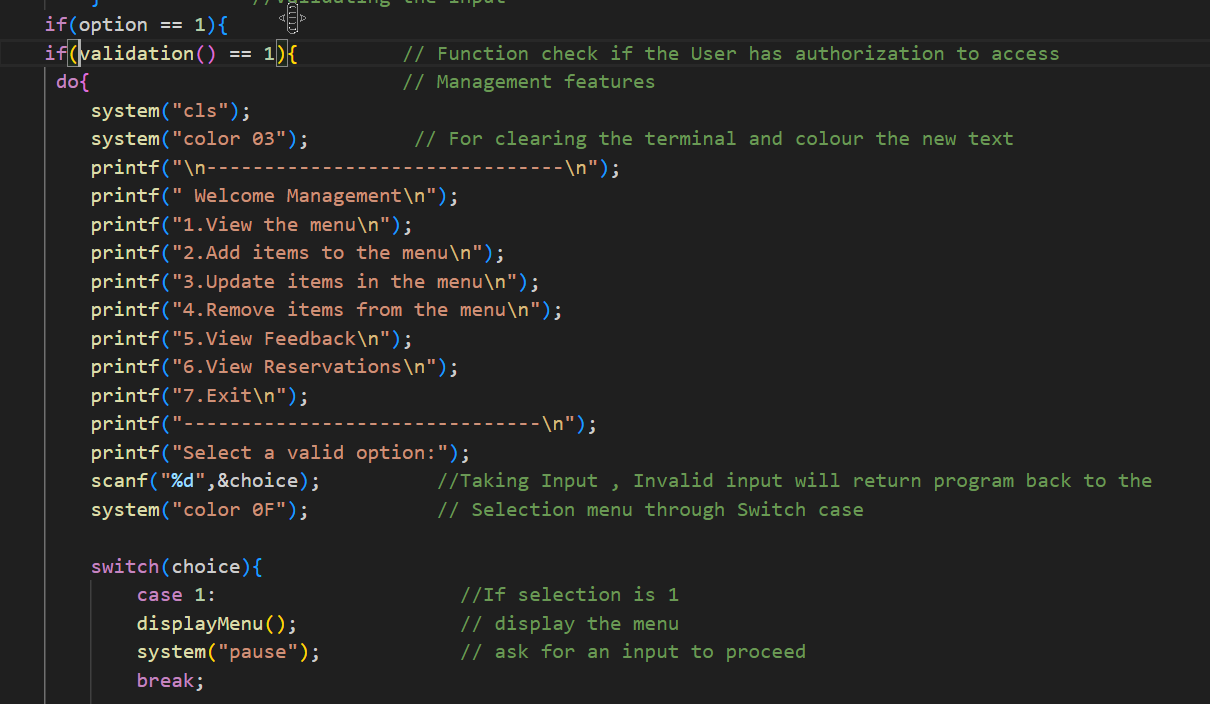
**Main Function:**

****

In the main function at the start few variables are declared. Them create menu function and initialize reservation functions are called. After that a user asked whether he/she wants to access management menu or customer menu. If an invalid input is given, user is asked to enter a valid selection again, until a valid input is given.



This screen will be shown until a valid output is selected.



If user selects the management option, then validation function is called, if validation fails user is send to the first menu. If validation is successful user is shown the management menu and prompted to select an option. If an invalid selection is made, user is asked again. The menu loop ends when the user selects the exit option.

Option 1 calls the display menu function and waits for and input to proceed again with the program

A screenshot of a computer

Description automatically generated

To access management menu, User will be asked to enter username and password.

A black screen with red text

Description automatically generated

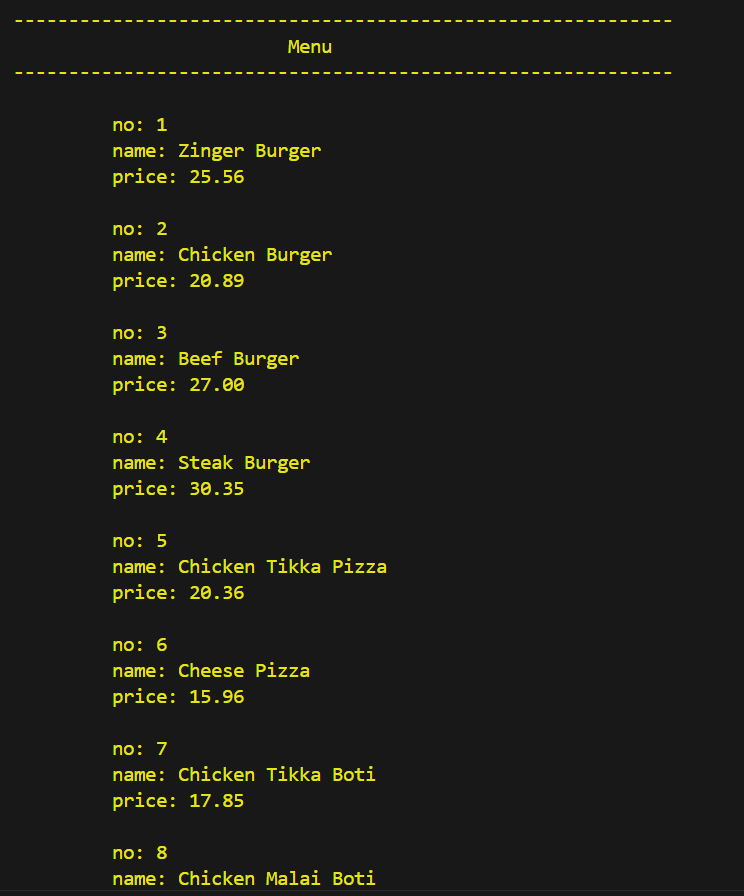
In case of incorrect username or password, this screen will be displayed.

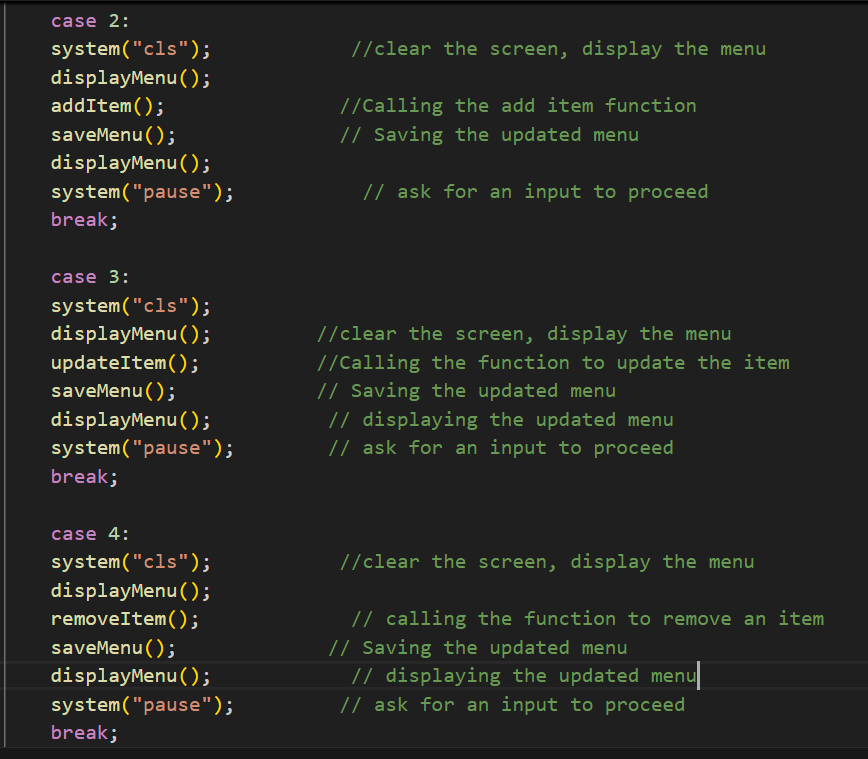
A screenshot of a computer program

Description automatically generated

In case of correct username and password, This menu will be displayed. This will continue to be displayed until a valid selection is made.

Option 1 will display the menu like this.

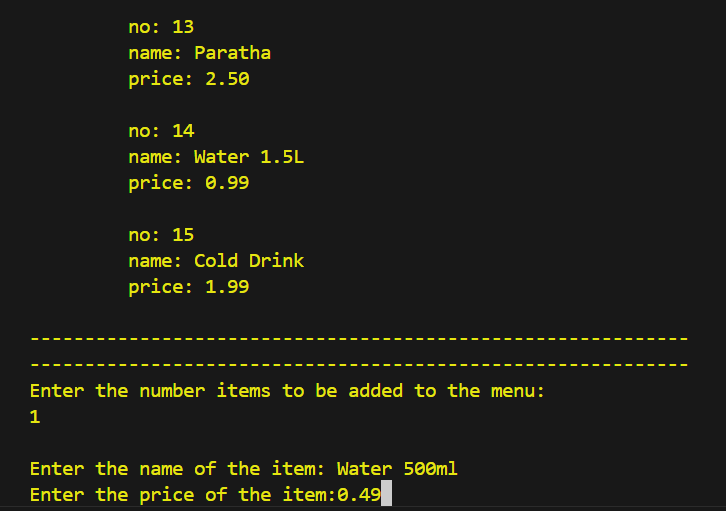




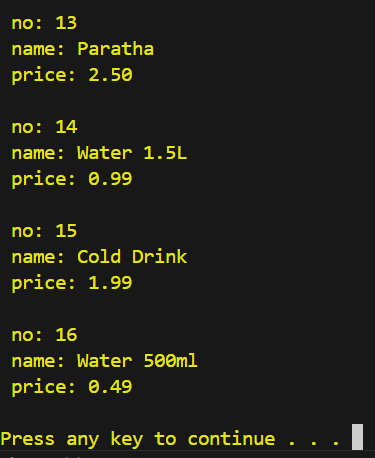
Option 2 clears the screen, calls the display menu function and calls the add item function. Then it saves the menu and calls the display menu function and waits for the user to enter to proceed.

Option 3 clears the screen, calls the display menu function and calls the update item function. Then it saves the menu and calls the display menu function and waits for the user to enter to proceed.

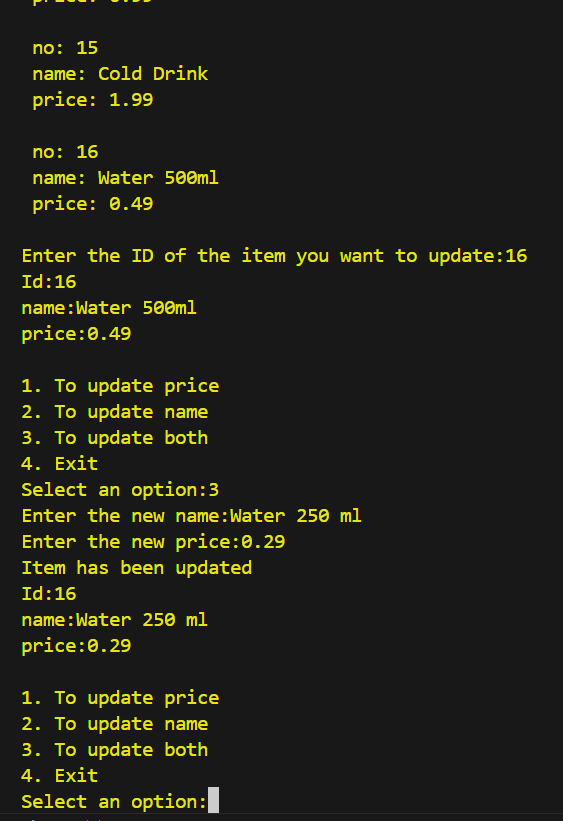
Option 4 clears the screen, calls the display menu function and calls the remove item function. Then it saves the menu and calls the display menu function and waits for the user to enter to proceed.



Selecting option 2 will display this screen and ask users to enter the required data. After which an updated menu is shown



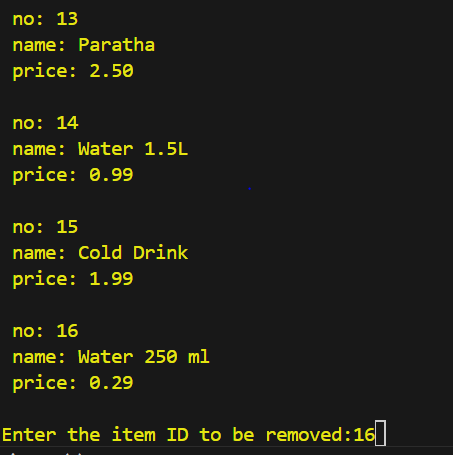
Selecting option 3 will display the following screen , asking user to enter the required data, in case of invalid selection it will prompt again



After exiting the updated menu is shownA screenshot of a computer

Description automatically generated

On selecting option 4, user will be asked to enter the id of the item to be removed, then display the updated menu

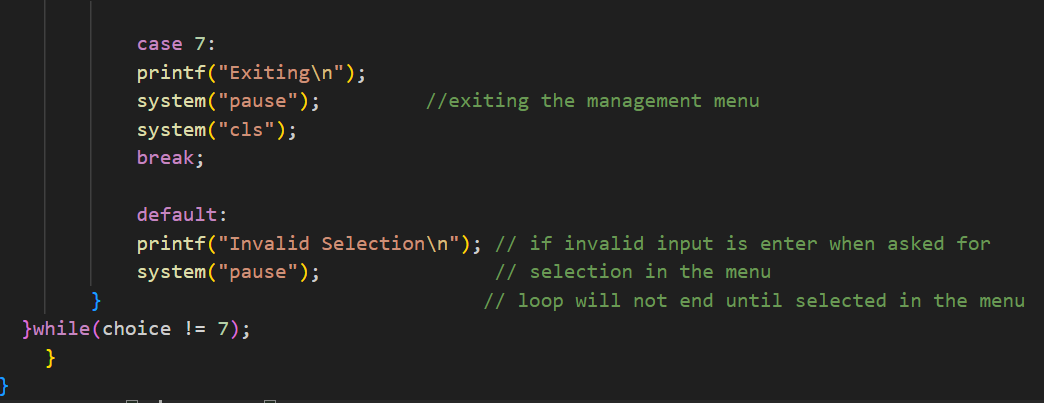


A screenshot of a computer

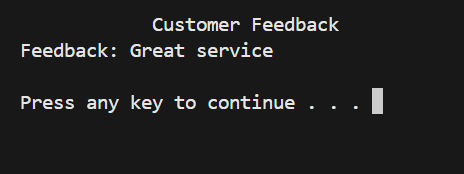
Description automatically generated

A screenshot of a computer program

Description automatically generated



Option 5 clears the screen and calls the view feedback function which shows the last feedback



Option 6 clears the screen and changes the colour, then it prompts the user to select which reservation list user wants to see and then it calls the view reservation function according to the input

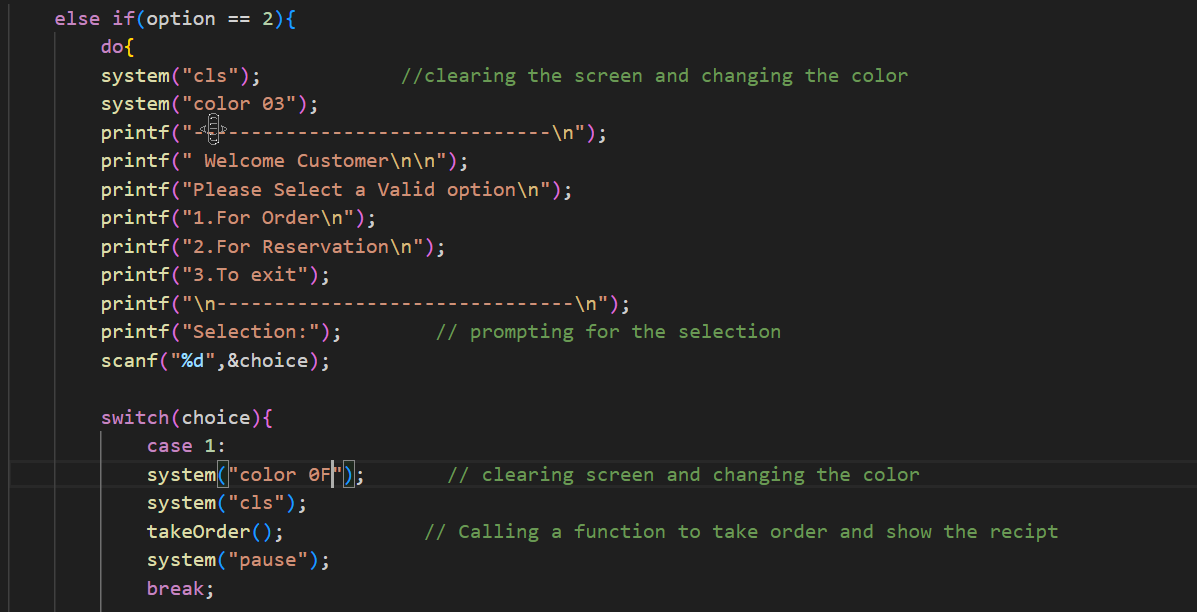
A screenshot of a menu

Description automatically generated

Option 7 exits the menu

A screenshot of a computer program

Description automatically generated

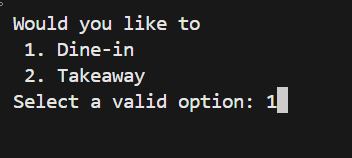


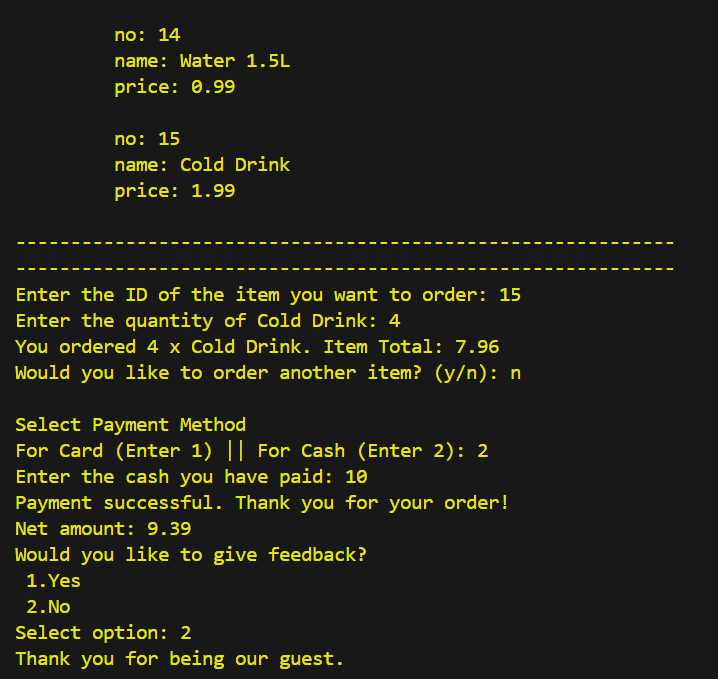
If customer option is selected then the customer menu is shown to the user, it clears the screen and changes the colour and prompts the user for the selection. If an invalid selection is made, user is asked to select again until a valid selection is made.

A screenshot of a computer program

Description automatically generated

Option 1 clears the screen and changes the colour. Then take order function is called after which the programs pause before proceeding.

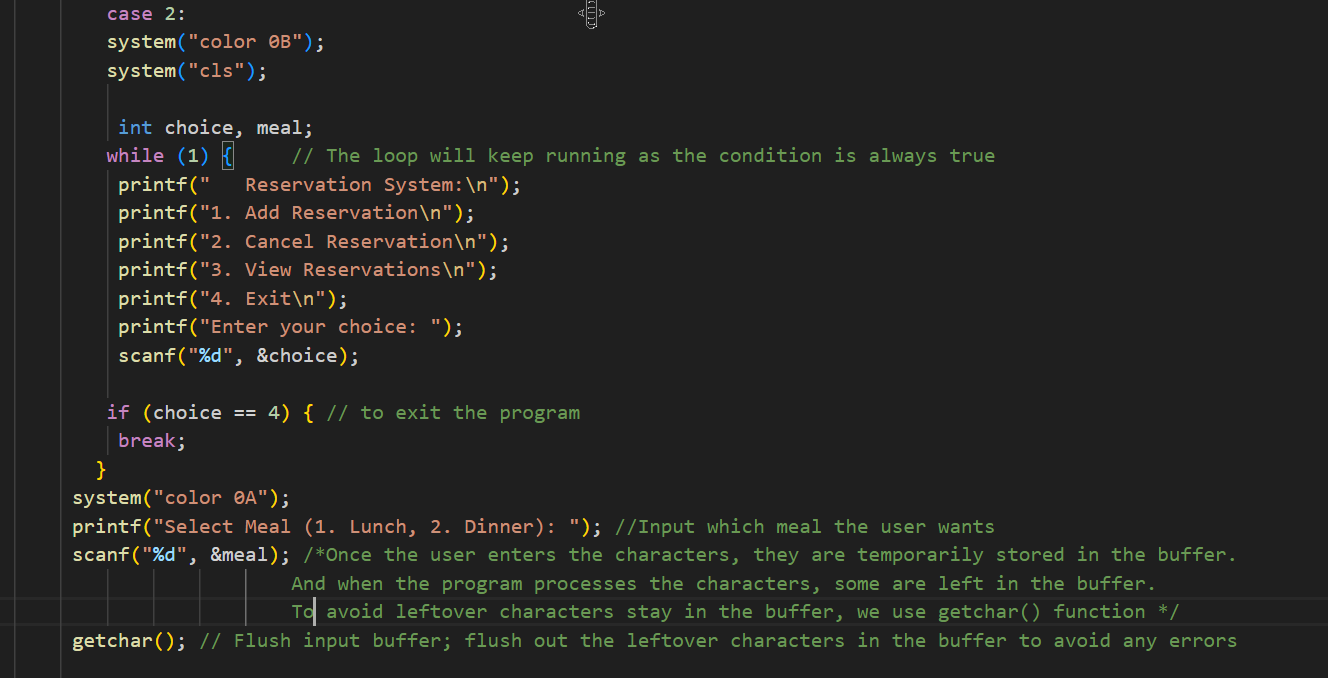




A screenshot of a computer screen

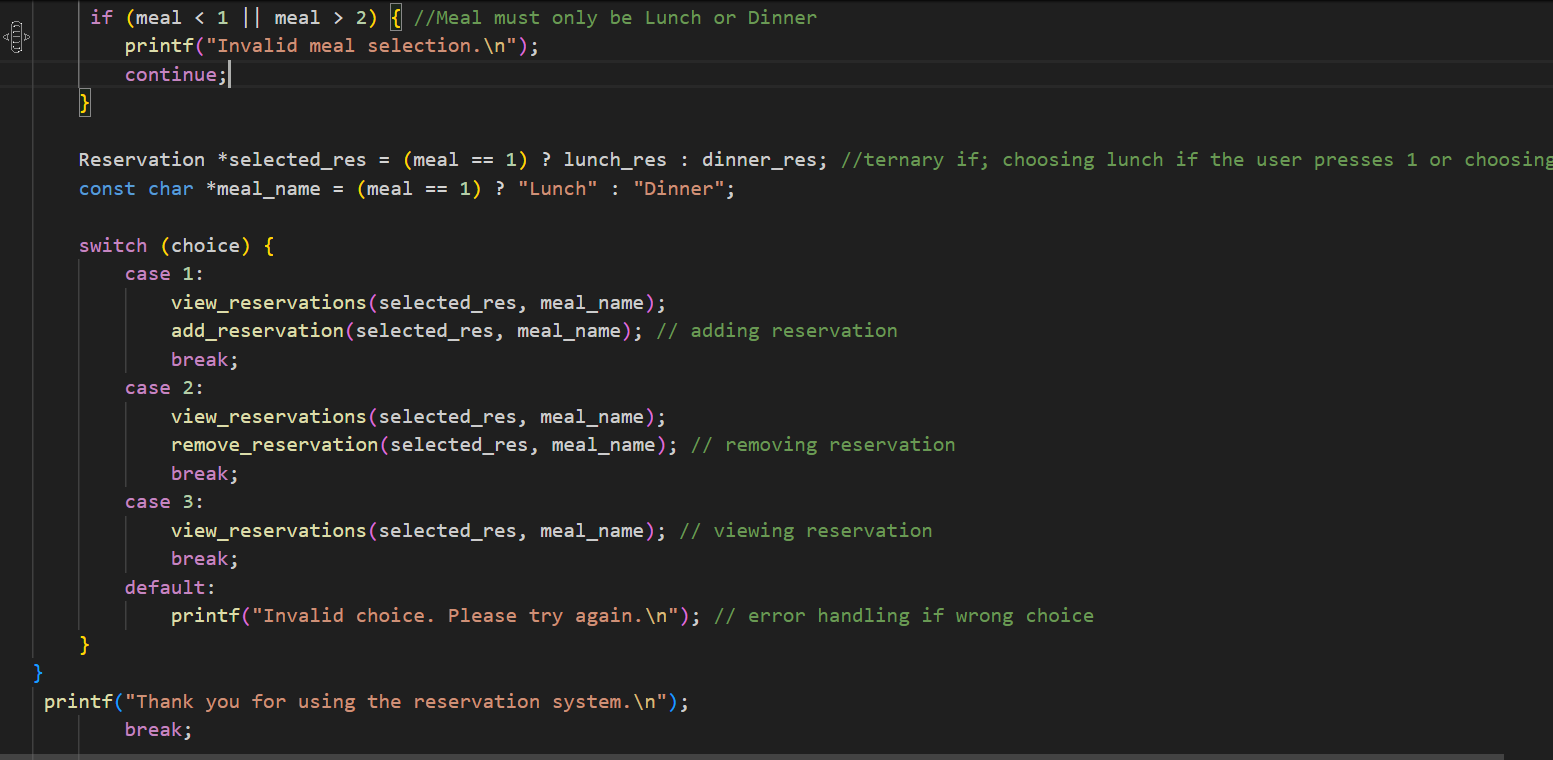
Description automatically generated

After placing the order and paying, the above receipt is generated.



Option 2 clears the screen and changes the colour. An option to add, cancel, view reservation and exit is shown, and user is asked for an input. A loop runs until user enters a valid selection. Then system colour is changes

These are output of different selections made

 User is then prompted to select the meal for the reservation, in case option 1 in the reservation view reservation and add reservation functions are called.

In case of option 2 view reservation and remove reservation functions are called.

In case of option 3 view reservation function is called

in case of option 4 the menu loop is exited

A black screen with green text

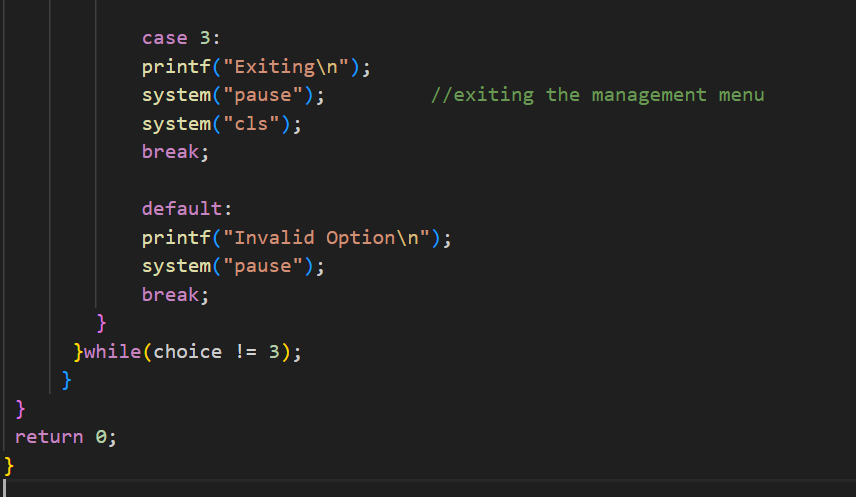
Description automatically generated

A screenshot of a computer program

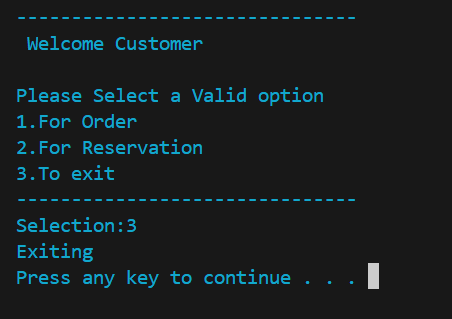
Description automatically generated

A screenshot of a computer program

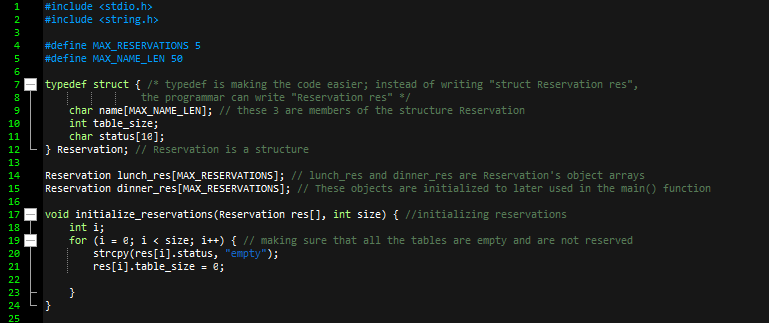
Description automatically generated



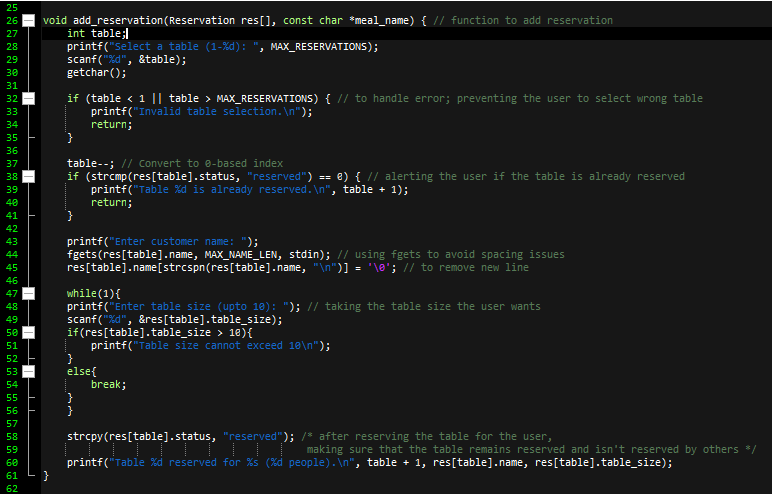
When option 3 on the customer menu is selected the program ends.



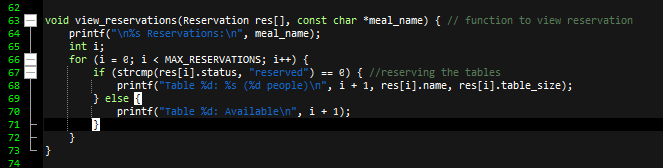
## Reservation System



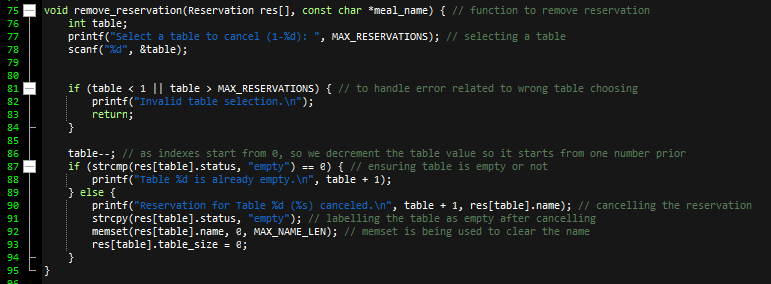
1. Typedef structure is made to prevent the user from writing the structure name again and again
2. Lunch\_res and dinner\_res are object arrays that are later used in the main() function
3. Initialize\_reservation function is initializing the reservations and ensuring that all the tables are empty for reservations
   1. I have used strcpy in place of scanf as scanf is inefficient when used with a sting



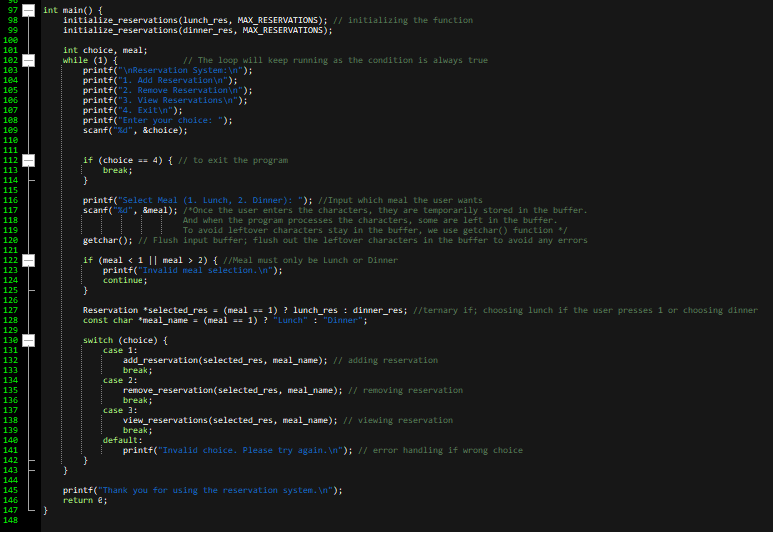
1. Add\_reservation function is mainly used to add the reservations
   1. The code is asking user to reserve the table
      1. I have used **getchar()** to flush out the leftover characters in the buffer to avoid any errors
   2. The if statement is validating the table input
   3. The following if statement is telling the user that the table is already reserved; setting the status to reserved
      1. **table--** is being used, so we decrement the table value so it starts from one number prior because indexes start from 0
   4. The customer name is being input
      1. Using **fgets** to avoid spacing issues
   5. While loop (that is already true) is taking the table size
      1. The if statement is making sure that the customer choose the table size within the range
   6. Making the table reserved
      1. Printing the table reserved, the customer name, and the table size to display to the user



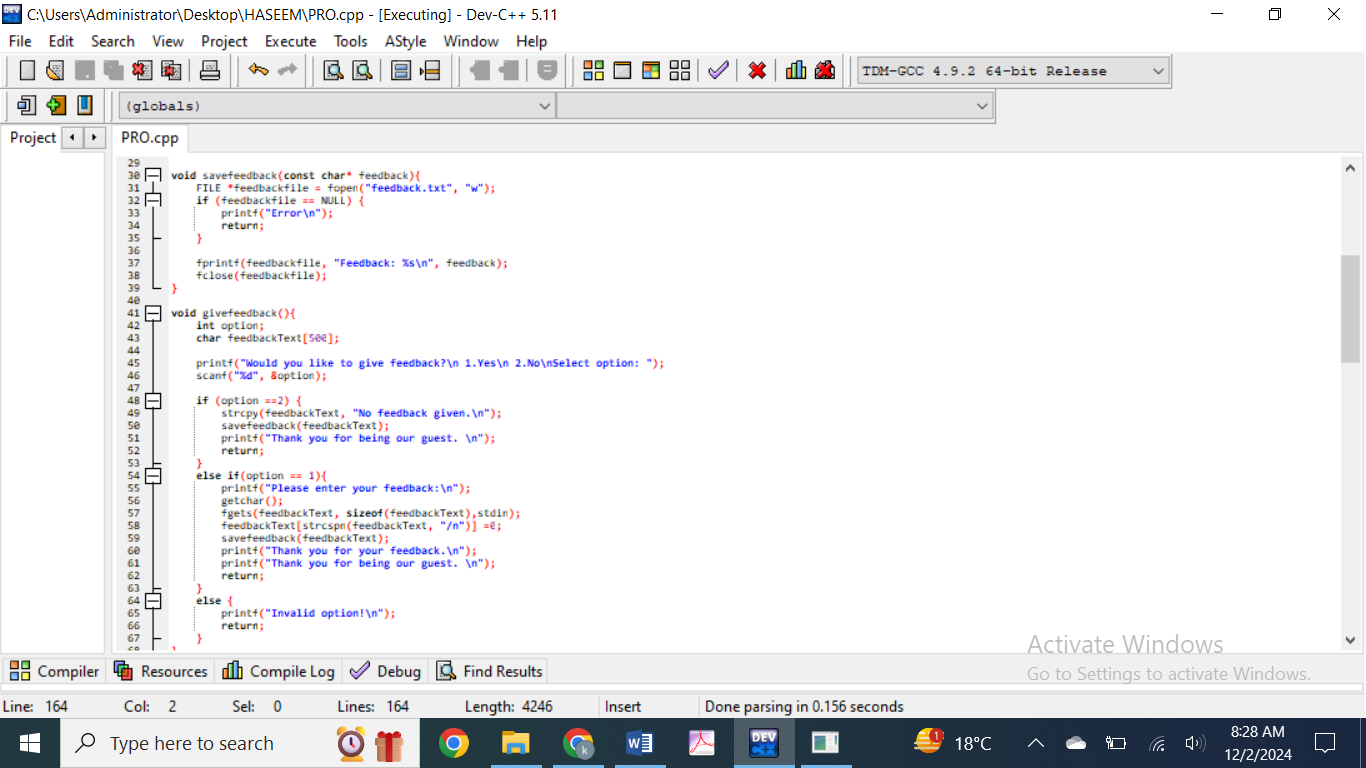
1. View\_reservation function is being used to view reservations
   1. For loop is displaying the tables as reserved or not reserved; printing available if the table is not reserved



1. Remove\_reservation() function is being used to remove reservations
   1. Asking the user for the table he want to cancel
   2. Using if statement to disallow invalid table selection
   3. The following if statement is determining either the table is empty or not; if it is not empty, then using methods to make it empty

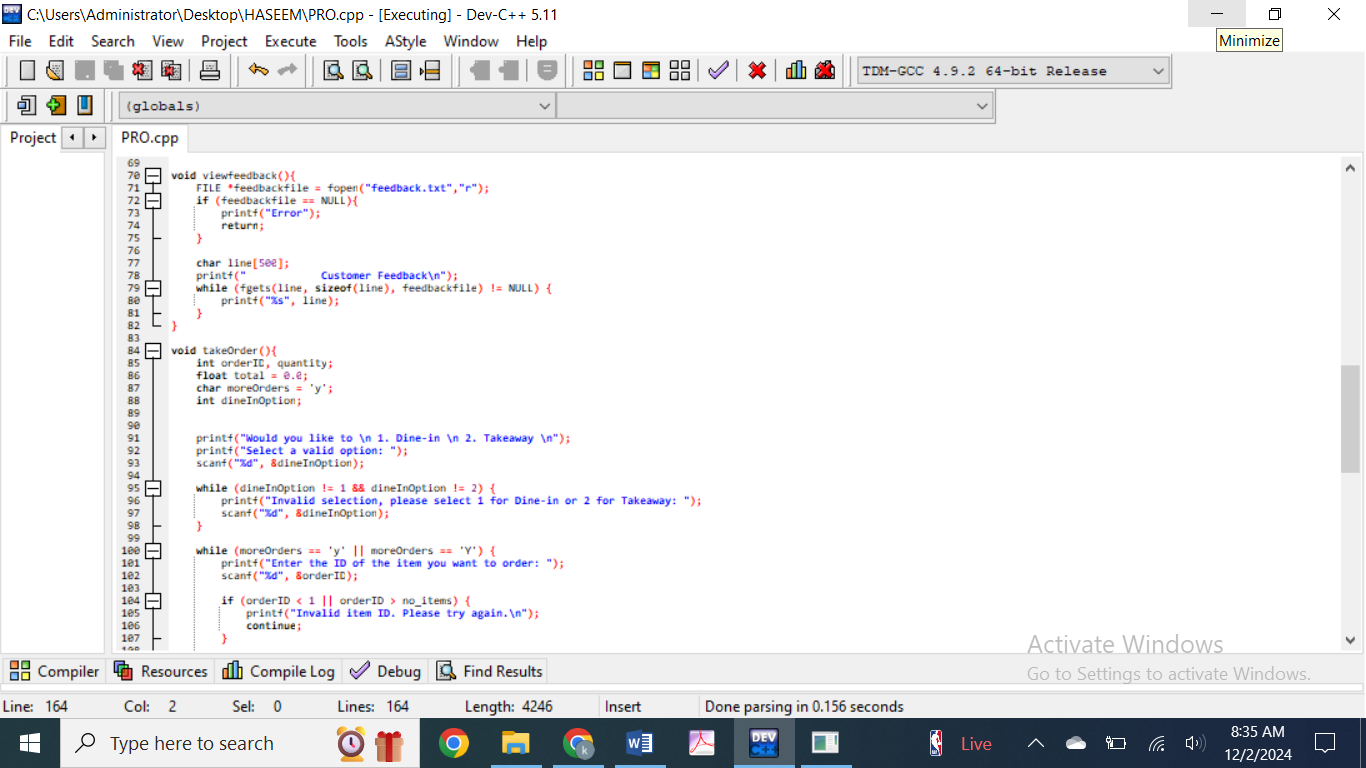


1. Main function is being used to run the code. It’s the main spine of the code
   1. Lunch\_res and dinner\_res are initialized
   2. Using while (always true) to ask user input
   3. Using if statement to exit the program upon wrong choice input
   4. Asking user to select meal (lunch or dinner)
   5. Using if statement to verify the meal chosen is valid
   6. Ternary if to choose lunch or dinner
   7. Switch case to redirect user according to his choice
   8. Thanking user for using our top notch reservation system
   9. Return 0 to ensure that everything has ran smoothly with no errors



The function **savefeedback** saves the customer feedback to the text file. It only works if the user chooses to give feedback after placing the order.

The function **givefeedback** asks the user if they want to give the feedback and then calls savefeedback function to store in the file. It gives option to the user to whether he has to give feedback or not. If he selects 1 it reads feedback and stores in savefeedback using fgets.



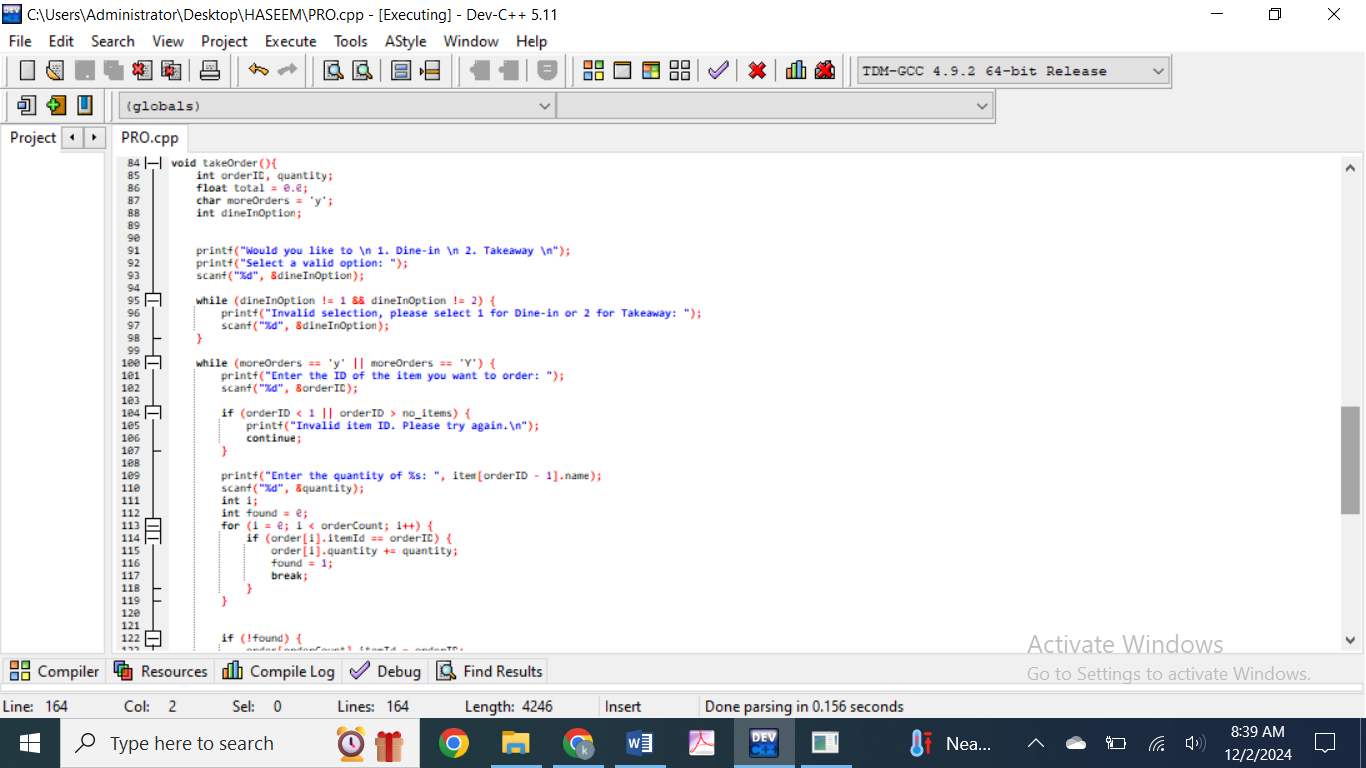
Viewfeedback() function displays the function saved in the text file.

It is printed on the receipt that is issued to the user.

The takeorder function handles the entire process of taking a customer's order.

It allows users to choose between dine-in or takeaway options, select items from the menu, specify quantities, and see the final receipt with the total cost.

The dine in and take away option first asks the user whether they would like to dine-in or take the order as takeaway. It ensures the user provides a valid choice.

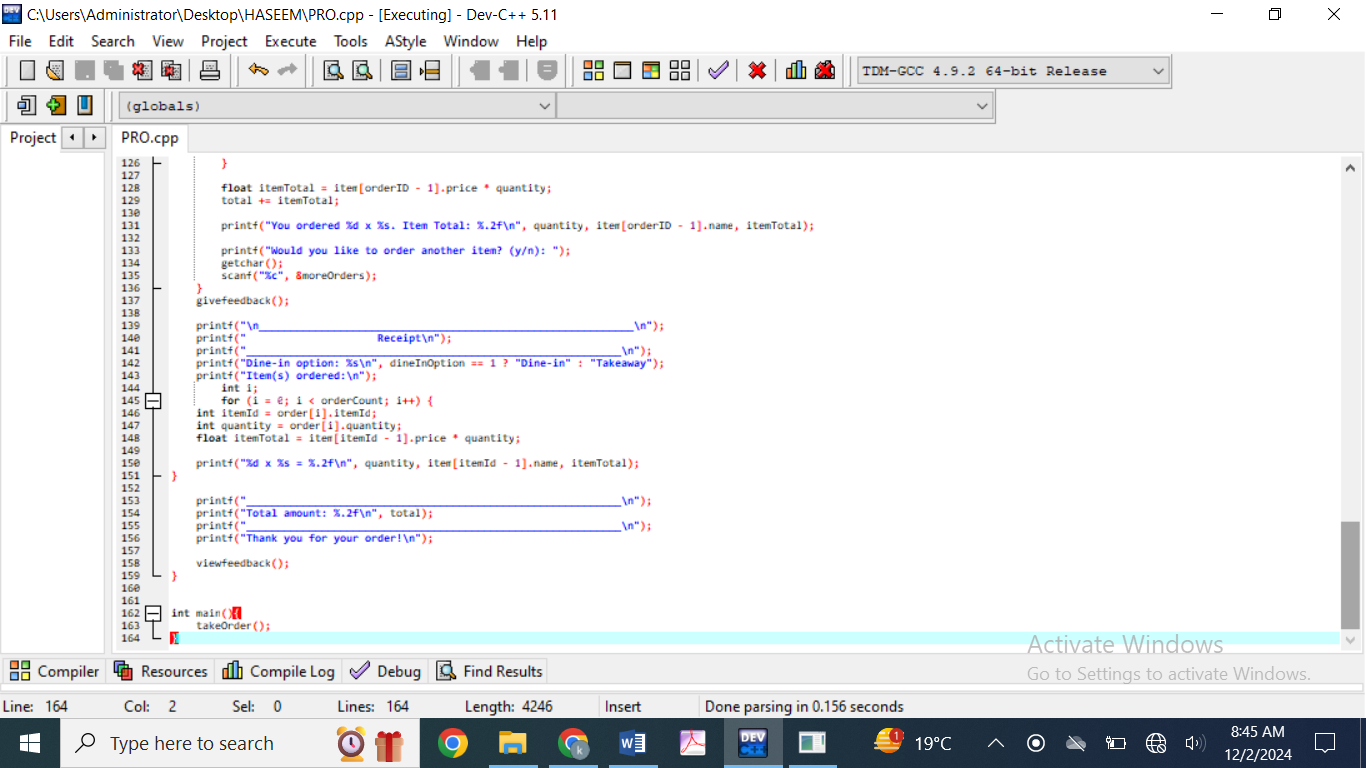


After wards the program enters a loop where the user can order multiple items. The user is prompted to input the ID of the item they wish to order along with the desired quantity.

If the entered item ID is invalid (i.e., outside the valid range of menu items), the program will prompt the user to try again.

If the item ID is valid, the program checks whether the item has already been ordered by searching the order array.

If the item is already in the list, it updates the quantity of that item; otherwise, it adds the item to the list as a new order.



For each item ordered, the total price is updated, and the details (quantity, item name, and item total) are displayed to the user.

After each item order, the user is asked if they would like to order another item. The loop continues until the user answers n, signaling that they no longer wish to order more items.