

FOOD ORDERING SYSTEM

Objective:

The "Food Ordering System" has been developed to override the problems prevailing in the current manual system . This program is supported to eliminate and in some cases reduce the hardships faced by the current system . The interface provided is user-friendly and easy to work upon . The system can lead to error free, secure , reliable and fast management system .

Functionalities provided are as follows :

- **Getting user information such as Name , Contact Details and Address .**
- **Sign-up and Log-in functionality implementation in C++ Program.**
- **Run-time OTP Verification in Python Program .**
- **Displaying Food and Drinks Menu and giving user the rights to choose the items .**
- **Bill Displaying Functionality using proper indentation .**
- **Finalizing the Payment Details and placing the order .**

Programming Languages used in the Project :

- **Python**
- **C++**

Function Description

Sign-up Function :

The sign-up function allows the user to set the contact number as well as a password that can be used to login to further program . This function further calls a password verifier function .

Password Verifier :

The Password Verifier function checks whether the password enter by user is valid or not .It has to be atleast 8 characters , one numeric character , smallcase character , uppercase character and special symbol are mandatory . This is a function with return type bool which returns 0 if password entered by user doesn't meets the required conditions and returns 1 if the password is valid .

Login function :

The login function creates an interface where user has to enter the authentication credentials that should match those which were entered during signing up . If credentials match , it will allow to proceed further or else it will pop-up again to re-enter the correct login credentials .

OTP function :

The OTP function is a special function created using third party Twilio API that helps to send text messages to a phone number . It has two parts :

- 1) OTP Sender : It sends an actual text message containing OTP to the given mobile number by creating a random otp passcode using random function .
- 2) OTP Verifier : In this function , the user has to enter the correct otp to proceed further into the program . If user enters wrong OTP , command shifts back to otp sender function that once again sends an OTP to the registered mobile number and again OTP verifier checks it .

Order Menu function :

The order menu function converts all the food and drinks item to respective datatypes . And also it provides 3 options to the user .

- 1) Open Food and Drinks Menu
- 2) Go to the main menu
- 3) Exit the program

Food and Drinks Menu function:

Food and Drinks menu option user slicer and sorter functions to display the menu of the cafe . It shows the item name as well as its price . The user can select the food or drinks of choice any number of times .

Slicer function :

The slicer function slices the string from the food list as well as drinks list and converts into two different operands : Item and its Price .

Sorter function :

Sorter function sorts the food and drinks menu list according to the price using the sorter functionality using the string to int datatype conversion and using Slicer function that divided a string into two parts .

Show Bill Function :

The show bill function creates a bill and displays it to the user . It is implemented to give proper alignment of item name , price and its quantity . It gathers the data user selected while ordering the items and finally uses it and performs operations to it to show the final bill to the customer .

// C++ Code :

```
#include <bits/stdc++.h>
#include <windows.h>
using namespace std;

void welcome(){
    cout << endl << endl ;
    cout << "\t\t\t----- Welcome to LA'OPALA CAFE ----- \n" <<
}

// Wait function
void wait_func()
{
    cout<<"Please Wait ";
    for(int i = 0 ; i < 3 ; i++){
        cout<<". ";
        Sleep(1000);
    }
    cout << endl ;
}

bool verify_password(char *p){
    // check if the length is insufficient
    int length = strlen(p);
    if (length < 8) return false;

    bool has_upper = false;
    bool has_lower = false;
    bool has_digit = false;
    bool has_symbol = false;

    // check for each of the required character classes
    for (int i = 0; i < length; i++){
        if (islower(p[i])) has_lower = true;
        if (isdigit(p[i])) has_digit = true;
        if (isupper(p[i])) has_upper = true;
        if (ispunct(p[i])) has_symbol = true;
    }

    // return false if any required character class is not present
    if (!has_upper) return false;
    if (!has_lower) return false;
    if (!has_digit) return false;
    if (!has_symbol) return false;
```

```

    // if we couldn't invalidate the password it must be valid
    return true;
}

void sign_up(string &password, string &contact_num, string &name){

    cout << "\t\t\t\t\t ----- SIGN UP YOUR NEW ACCOUNT ----- \n" << endl ;
    cout << "Enter your Name : ";
    cin >> name ;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cout << "Enter your contact number : " ;
    cin >> contact_num ;
    cout << endl ;
    label:
    cout << "Create a password(must be minimum 8 characters ,must include
    special character ,upper case and lower case alphabets and number):\n" ;
    cin >> password ;
    cout << endl ;
    char p[50];
    for(int i = 0 ; i < password.length() ; i++ )
    {
        p[i] = password[i] ;
    }
    bool result = verify_password(p);
    if (result) printf("Verified password!\n\n");
    else {
        printf("Invalid password! \n");
        goto label;
    }
}

void log_in(string &password, string &contact_num,
            string &address, string &name){

    string contact_verify , password_verify ;
    cout << "\t\t\t\t\t----- LOG IN TO YOUR ACCOUNT ----- \n" ;
    cout << endl ;
    label:
    cout << "Enter contact number : " ;
    cin >> contact_verify ;
    cout << "Enter Password : " ;
    cin >> password_verify ;
    cout << endl ;
    cout << endl ;
    if((contact_num==contact_verify)&&(password==password_verify)){
        cout << "Account Verified. Logging in ...";
        cout << endl ;
        wait_func();
        cout << "Welcome " << name << endl << endl ;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
}

```

```

        cout << "Enter your delivery address : " ;
        cin >> address ;
    }
    else{
        cout << "Try Again " << endl ;
        goto label;
    }
}

void menu_card(vector<string> &fmenu, vector<int> &price ,
    vector<string> &cart_items , vector <int> &cart_prices ,
    vector <int> &cart_quantity , int quantity , int order){

    while(true)
    {
        int choice ;

        label:
        cout << endl << endl << " ----- Food Menu ----- \n" ;
        cout << endl ;

        for (int i = 0; i < 10; i++)
        {

            // Showing food menu to customer
            cout << i + 1 << ". " << fmenu[i] << endl;
        }

        cout << "\nSelect the food you want to order : ";
        // Taking order from customer
        cin >> order;

        cout << "\nYou have selected " << fmenu[order - 1] << endl;

        cout << "\nHow many you want to order ? : ";
        // Taking quantity of food
        cin >> quantity;

        cout<< endl ;

        cart_items.push_back(fmenu[order - 1]) ;
        cart_prices.push_back(price[order - 1]) ;
        cart_quantity.push_back(quantity) ;

        cout << "Would you like to order anything else ? \n\n" ;
        cout << "Press 1 for Yes or Press 0 for No.\n" ;
        cin>>choice ;
        cout<<endl;
    }
}

```

```

        if(choice){
            goto label;
        }
        else{
            break;
        }
    }

}

}

void payment(int &total_price){
    string coupon ;
    cout<< "Enter FIRST15 to get 15% off on your first order .\n\n";
    cin >> coupon ;
    if(coupon == "FIRST15" || coupon == "first15"){
        total_price = total_price - 0.15*total_price ;
    }
    cout << "You need to pay Rs. " << total_price ;
}

void show_payment_bill(vector<string> &cart_items ,
    vector <int> &cart_prices , vector <int> &cart_quantity){

    int total_price = 0 ;
    cout << "----- YOUR BILL -----";
    cout << endl ;
    cout << endl ;
    cout << "Ordered Items\t" << "          " << " Price x Quantity" ;
    cout << endl ;
    cout << endl ;

    for(int i = 0 ; i < cart_items.size() ; i++ )
    {
        total_price = total_price + cart_prices[i]*cart_quantity[i] ;
        cout << cart_items[i] ;

        for(int j=0;j<=(20-cart_items[i].size());j++){
            cout<<" ";
        }
        cout<<"|          " <<"Rs. " << cart_prices[i] << " x " <<
            cart_quantity[i] << endl;
    }
    cout << "-----\n";

    cout << "TOTAL          :          " <<"Rs. " << total_price ;
}

```

```

        cout << endl ;
        cout << endl ;
        options :
        cout << "(C) APPLY COUPON\t(P) MAKE PAYMENT\t (E) EXIT" ;
        cout << endl ;

        char inp2 ;
        cin >> inp2 ;

        options_3 :
        if(toupper(inp2) == 'C'){
            payment(total_price) ;
        }
        else if(toupper(inp2) == 'P'){
            cout << "You need to pay Rs. " << total_price ;
        }
        else if(toupper(inp2) == 'E'){
            exit(0);
        }
        else {
            cout << "Invalid Input . Try Again !\n" ;
            goto options_3 ;
        }
    }

int main()
{
    string password ;
    string contact_num ;
    vector <string> cart_items;
    vector <int> cart_prices;
    vector <int> cart_quantity;
    int order , quantity , cpayment;
    string name , delivery , paymentnum , transid , address ;
    // Storing list of foods in string type array "fmenu"
    vector <string> fmenu =
    {
        "Burger",
        "Coke",
        "Cold Coffee",
        "Pizza",
        "Crispy Corn",
        "French Fries",
        "Americano Coffee",
        "Ice Cream",
        "Latte",
        "Momos"
    }
}

```



```

};
// Storing prices of foods in integer type array "price"
vector<int> price =
{
    100,
    40,
    99,
    249,
    149,
    80,
    150,
    100,
    149,
    179
};

char inp ;

welcome ();

sign_up (password , contact_num , name );
wait_func ();
log_in (password , contact_num , address , name );

menu_card (fmenu , price , cart_items , cart_prices , cart_quantity ,
            quantity , order );

options :
cout << "(E) EXIT \t\t (P) PAYMENT" << endl ;

cin >> inp ;

if (toupper(inp) == 'E'){
    exit(0);
}
else if (toupper(inp) == 'P'){
    show_payment_bill (cart_items , cart_prices , cart_quantity );
}
else {
    cout << "Invalid Input . Try Again !\n" ;
    goto options ;
}

cout << endl ;
cout << "Reaching you in 27 minutes !";
cout << endl ;

```

```
cout << "\nThank you for choosing us. \nENJOY YOUR SNACK! :) \n";  
cout << endl ;  
cout << endl ;  
  
return 0;  
}
```

// C++ Code Output :

```
notpujan@notebook-Virt

----- Welcome to LA'OPALA CAFE -----
----- SIGN UP YOUR NEW ACCOUNT -----

Enter your Name : Pujan
Enter your contact number : 95222342XX

Create a password(must be minimum 8 characters , must include special character , up
PujaN@123

Verified password!

----- LOG IN TO YOUR ACCOUNT -----

Enter contact number : 95222342XX
Enter Password : PujaN@123

Account Verified. Logging in ...
Welcome Pujan

Enter your delivery address : Indore

----- Food Menu -----
1. Burger
2. Coke
3. Cold Coffee
4. Pizza
5. Crispy Corn
6. French Fries
7. Americano Coffee
8. Ice Cream
9. Latte
10. Momos

Select the food you want to order : 3

You have selected Cold Coffee

How many you want to order ? : 2

Would you like to order anything else ?

Press 1 for Yes or Press 0 for No.
```



Would you like to order anything else ?

Press 1 for Yes or Press 0 for No.

1

----- Food Menu -----

1. Burger
2. Coke
3. Cold Coffee
4. Pizza
5. Crispy Corn
6. French Fries
7. Americano Coffee
8. Ice Cream
9. Latte
10. Momos

Select the food you want to order : 5

You have selected Crispy Corn

How many you want to order ? : 4

Would you like to order anything else ?

Press 1 for Yes or Press 0 for No.

1

----- Food Menu -----

1. Burger
2. Coke
3. Cold Coffee
4. Pizza
5. Crispy Corn
6. French Fries
7. Americano Coffee
8. Ice Cream
9. Latte
10. Momos

Select the food you want to order : 7

```
1. Burger
2. Coke
3. Cold Coffee
4. Pizza
5. Crispy Corn
6. French Fries
7. Americano Coffee
8. Ice Cream
9. Latte
10. Momos

Select the food you want to order : 7

You have selected Americano Coffee

How many you want to order ? : 1

Would you like to order anything else ?

Press 1 for Yes or Press 0 for No.
0

(E) EXIT                (P) PAYMENT
p
----- YOUR BILL -----

Ordered Items          |          Price x Quantity
Cold Coffee            |          Rs. 99 x 2
Crispy Corn            |          Rs. 149 x 4
Americano Coffee       |          Rs. 150 x 1
-----
TOTAL                  :          Rs. 944

(C) APPLY COUPON        (P) MAKE PAYMENT        (E) EXIT
C
Enter FIRST15 to get 15% off on your first order .

first15
You need to pay Rs. 802
Reaching you in 27 minutes !

Thank you for choosing us.
ENJOY YOUR SNACK! :)

notpujan@notebook-VirtualBox:~/Documents/CPP_PROJECT$ |
```

// C++ Code Debugging :

```
notpujan@notebook-VirtualBox:~/Documents/CPP_PROJECT$ g++ -g minicpp.cpp
notpujan@notebook-VirtualBox:~/Documents/CPP_PROJECT$ gdb a.out
GNU gdb (Ubuntu 12.0.90-0ubuntu1) 12.0.90
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) break main
Breakpoint 1 at 0x359a: file minicpp.cpp, line 222.
(gdb) n
The program is not being run.
(gdb) run
Starting program: /home/notpujan/Documents/CPP_PROJECT/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at minicpp.cpp:222
222     {
(gdb) n
223     string password ;
(gdb) n
224     string contact_num ;
(gdb) n
225     vector <string> cart_items;
(gdb) n
226     vector <int> cart_prices;
(gdb) n
227     vector <int> cart_quantity;
(gdb) n
229     string name, delivery, paymentnum, transid, address ;
(gdb) n
243     };
(gdb) n
250     ..
```

```

notpujan@notebook-VirtualBox: ~/Documents/CPP_PROJE
265     welcome();
(gdb) n

----- Welcome to LA'OPALA CAFE -----

269     sign_up(password,contact_num,name);
(gdb) n

----- SIGN UP YOUR NEW ACCOUNT -----

Enter your Name : Pujan
Enter your contact number : 9522234000

Create a password(must be minimum 8 characters , must include special character , upper case and lower
Pujan@123

Verified password!

271     log_in(password,contact_num,address,name);
(gdb) n

----- LOG IN TO YOUR ACCOUNT -----

Enter contact number : 9522234000
Enter Password : Pujan@123

Account Verified. Logging in ...
Welcome Pujan

Enter your delivery address : indore
273     menu_card(fmenu,price,card_items,card_prices,card_quantity,quantity,order);
(gdb) n

----- Food Menu -----

1. Burger
2. Coke
3. Cold Coffee
4. Pizza
5. Crispy Corn
6. French Fries
7. Americano Coffee
8. Ice Cream
9. Latte
10. Momos

Select the food you want to order : 7

```



10. Momos

Select the food you want to order : 7

You have selected Americano Coffee

How many you want to order ? : 2

Would you like to order anything else ?

Press 1 for Yes or Press 0 for No.

1

----- Food Menu -----

1. Burger
2. Coke
3. Cold Coffee
4. Pizza
5. Crispy Corn
6. French Fries
7. Americano Coffee
8. Ice Cream
9. Latte
10. Momos

Select the food you want to order : 1

You have selected Burger

How many you want to order ? : 5

Would you like to order anything else ?

Press 1 for Yes or Press 0 for No.

1

----- Food Menu -----

1. Burger
2. Coke
3. Cold Coffee
4. Pizza
5. Crispy Corn


```
notpujan@notebook-VirtualBox: ~/Documents/CPP_PROJECT

----- Food Menu -----
1. Burger
2. Coke
3. Cold Coffee
4. Pizza
5. Crispy Corn
6. French Fries
7. Americano Coffee
8. Ice Cream
9. Latte
10. Momos

Select the food you want to order : 5

You have selected Crispy Corn

How many you want to order ? : 4

Would you like to order anything else ?

Press 1 for Yes or Press 0 for No.
0

276      cout<< "(E) EXIT \t\t (P) PAYMENT" << endl ;
(gdb) info local
password = "Pujan@123"
contact_num = "9522234000"
cart_items = std::vector of length 3, capacity 4 = {"Americano Coffee", "Burger", "Crispy Corn"}
cart_prices = std::vector of length 3, capacity 4 = {150, 100, 149}
cart_quantity = std::vector of length 3, capacity 4 = {2, 5, 4}
order = 21845
quantity = 1431694336
cpayment = <optimized out>
name = "Pujan"
delivery = ""
paymentnum = ""
transid = ""
address = "indore"
fmenu = std::vector of length 10, capacity 10 = {"Burger", "Coke", "Cold Coffee", "Pizza", "Crispy Corn", "French
    "Momos"}
price = std::vector of length 10, capacity 10 = {100, 40, 99, 249, 149, 80, 150, 100, 149, 179}
inp = 0 '\000'
(gdb) n
(E) EXIT                                (P) PAYMENT
278      cin >> inp ;
(gdb) n
```

```

notpujan@notebook-
(E) EXIT (P) PAYMENT
278 cin >> inp ;
(gdb) n
p
282 if(toupper(inp) == 'E'){
(gdb) n
285 else if (toupper(inp) == 'P'){
(gdb) n
286 show_payment_bill(cart_items, cart_prices, cart_quantity);
(gdb) n
----- YOUR BILL -----

Ordered Items      | Price x Quantity
Americano Coffee   | Rs. 150 x 2
Burger             | Rs. 100 x 5
Crispy Corn        | Rs. 149 x 4
-----
TOTAL              : Rs. 1396

(C) APPLY COUPON (P) MAKE PAYMENT (E) EXIT
c
Enter FIRST15 to get 15% off on your first order .

first15
293 cout<< endl ;
(gdb) n
You need to pay Rs. 1186
294 cout << "Reaching you in 27 minutes !";
(gdb) n
295 cout << endl ;
(gdb) n
Reaching you in 27 minutes !
297 cout << "\nThank you for choosing us. \nENJOY YOUR SNACK! :) \n";
(gdb) n

Thank you for choosing us.
ENJOY YOUR SNACK! :)
298 cout << endl ;
(gdb) n

299 cout << endl ;
(gdb) n

302 return 0;
(gdb) n
303 }
(gdb) info local

```

```

295     cout << endl ;
(gdb) n
Reaching you in 27 minutes !
297     cout << "\nThank you for choosing us. \nENJOY YOUR SNACK! :) \n";
(gdb) n

Thank you for choosing us.
ENJOY YOUR SNACK! :)
298     cout << endl ;
(gdb) n

299     cout << endl ;
(gdb) n

302     return 0;
(gdb) n
303 }
(gdb) info local
password = "Pujan@123"
contact_num = "9522234000"
cart_items = std::vector of length 3, capacity 4 = {"Americano Coffee", "Burger", "Crispy Corn"}
cart_prices = std::vector of length 3, capacity 4 = {150, 100, 149}
cart_quantity = std::vector of length 3, capacity 4 = {2, 5, 4}
order = 21845
quantity = 1431694336
cpayment = <optimized out>
name = "Pujan"
delivery = ""
paymentnum = ""
transid = ""
address = "indore"
fmenu = std::vector of length 10, capacity 10 = {"Burger", "Coke", "Cold Coffee", "Pizza", "Crispy Corn", "French Fries", "Americano Coffee", "Ice Cr
    "Momos"}
price = std::vector of length 10, capacity 10 = {100, 40, 99, 249, 149, 80, 150, 100, 149, 179}
inp = 112 'p'
(gdb) n
__libc_start_call_main (main=main@entry=0x555555557582 <main()>, argc=argc@entry=1, argv=argv@entry=0x7fffffffe0e8) at ../sysdeps/nptl/libc_start_cal
74  ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb) break main

```

Python Code :

```
import time
import random
import datetime
import os
from twilio.rest import Client

# greets the user
def greeting():
    print()
    print("WELCOME TO LA'OPALA CAFE !")
    print()

# wait.func() adds a delay
def wait.func():
    print("Please wait ", end="")
    for i in range(0, 3):
        print(".", end="")
        time.sleep(1)
    print()

# taking user_details as input from the user :
def user.info():
    print("Please enter your login details")
    user.name = input("Enter your name : ")
    password = input("Enter the password : ")
    contact.no = input("Enter your Contact number : ")
    wait.func()
    print(f"Hey, {user.name}")
    locality = input("Address : ")
    flat.number = input("Flat No. : ")
    wait.func()
    print(f"Your address is Flat No. {flat.number} , {locality}")

# otp_function() sends and verifies the otp using 3rd party Twilio API
def otp_function():
    # OTP SENDER
    otp = random.randint(1000, 9999)
    account.sid = "AC228071cb024f37dad6b2970f83a99c74"
    auth.token = 'c1fa18f3fe71bb1cfc47ada1b2994625'
    client = Client(account.sid, auth.token)
    msg = client.messages.create(
        body=f"Your OTP is {otp}",
        from_="+15139603290",
        to="+919522234236"
    )
    # OTP VERIFICATION
    user.otp = int(input("Enter OTP : "))
    time.sleep(1)
    if otp == user.otp:
        print("Successfully Verified !")
    else:
        print("Invalid OTP !")
        print("Press 1 to retry .")
        retry = int(input())
        if retry == 1:
            otp_function()
        else:
            exit()

# food.menu
list_foods = ["Aloo Tikki Burger Rs 100.00",
              "Chatpata Naan Aloo Rs 100.00",
              "Pizza Rs 120.00",
              "Chocolate Brownie Rs 165.00",
              "Spicy Paazta Rs 100.00",
              "White Sauce Paazta Rs 129.00",
              "Masala Chaap Rs 109.00",
              "Veg Noodles Rs 110.00",
              "Spring Roll Rs 149.00",
              "Crispy Corn Rs 119.00",
              "Manchurian Rs 119.00",
              "Masala Dosa Rs 100.00",
              "Spicy Veg Momos Rs 119.00",
              "Tandoor Momos Rs 129.00",
              "Steamed Momos Rs 109.00",
              "Masala Sandwich Rs 149.00",
              "Club Sandwich Rs 180.00",
              "Cheesy Corn Rs 149.00",
              "Bombay Bhel Rs 115.00",
              "Paneer Rolls Rs 200.00"]

# drinks.menu
list_drinks = ["Coca-Cola Rs 35.00",
               "Fanta Rs 35.00",
               "Black Tea Rs 20.00",
               "Sprite Rs 35.00",
               "Limca Rs 35.00",
               "Milk Tea Rs 15.00",
               "Lemon Tea Rs 30.50",
               "Kesariya Milk Rs 45.00",
               "Water Rs 20.00",
               "Virgin Mojito Rs 99.00",
               "Mojito Rs 89.00",
               "Black Coffee Rs 39.00",
               "Cappuccino Rs 69.00"]
```

```

        "Americano Rs 79.00",
        "Espresso Rs 89.00",
        "Latte Rs 99.00",
        "Milkshake Rs 99.00",
        "Mixed Fruit Juice Rs 75.00",
        "Coconut Crush Rs 89.00",
        "Cold Coffee Rs 89.00"]

list_item_price = [0] * 100

def def_default():
    global list_drinks, list_foods, cart_items, list_item_price
    cart_items = [0] * 100

# main func()
def main():
    while True:
        inp = str(input("Please enter M to open Menu : ").upper())
        if len(inp) == 1:
            if inp == 'M':
                print("\n" * 2)
                order_menu()
                break
            else:
                print("\n" * 2 + "ERROR: Invalid Input (" + str(inp) + "). Try again!")
        else:
            print("\n" * 2 + "ERROR: Invalid Input (" + str(inp) + "). Try again!")

#order_menu() function provides 3 options to choose from:
# 1) FOODS AND DRINKS MENU 2) MAIN MENU 3) EXIT
def order_menu():
    while True:
        print("-" * 20 + "ORDER PAGE" + "-" * 20 + "\n\n"
              "\t(F) FOODS AND DRINKS\n"
              "\t(M) MAIN MENU\n"
              "\t(E) EXIT\n" +
              "-" * 72)
        inp = str(input("Please Select Your Option: ").upper())
        if len(inp) == 1:
            if inp == 'F':
                print("\n" * 2)
                food_and_drinks()
                break
            elif inp == 'M':
                print("\n" * 2)
                main()
                break
            elif inp == 'E':
                print("*" * 32 + "THANK YOU" + "*" * 31 + "\n")
                break
            else:
                print("\n" * 2 + "ERROR: Invalid Input (" + str(inp) + ").")
        else:
            print("\n" * 2 + "ERROR: Invalid Input (" + str(inp) + ").")

# slicer_func() uses string slicing method to differentiate
# item from its price using index of 'Rs'
def slicer_func():
    i = 0
    while i <= (len(list_foods) - 1):
        if 'Rs' in list_foods[i]:
            list_foods[i] = str(list_foods[i][:list_foods[i].index('Rs') - 1]) + ' ' * (
                20 - (list_foods[i].index('Rs') - 1)) + str(list_foods[i][list_foods[i].index('Rs'):])
        i += 1

    i = 0
    while i <= (len(list_drinks) - 1):
        if 'Rs' in list_drinks[i]:
            list_drinks[i] = str(list_drinks[i][:list_drinks[i].index('Rs') - 1]) + ' ' * (
                20 - (list_drinks[i].index('Rs') - 1)) + str(list_drinks[i][list_drinks[i].index('Rs'):])
        i += 1

slicer_func()

# sorter_func() sorts the food and drinks menu list according to the price
def menu_sorter():
    global list_foods, list_drinks
    list_foods = sorted(list_foods)
    list_drinks = sorted(list_drinks)

# converting prices to float from string using slicing
i = 0
while i < len(list_foods):
    list_item_price[i] = float(list_foods[i][int(list_foods[i].index("Rs") + 2):])
    i += 1

i = 0
while i < len(list_drinks):
    list_item_price[20 + i] = float(list_drinks[i][int(list_drinks[i].index("Rs") + 2):])
    i += 1

```

```

menu_sorter()

# displays food and drinks menu with proper indentation
# from where user can choose items of his choice any number of times
def food_and_drinks():
    while True:
        print("-" * 26 + "ORDER FOODS & DRINKS" + "-" * 26)
        print("\n")
        print(" [NO] |FOOD NAME|           |PRICE|      | [NO] |DRINK NAME|           |PRICE|\n")

        i = 0
        while i < len(list_foods) or i < len(list_drinks):
            var_space = 1
            if i <= 8:
                var_space = 2

            if i < len(list_foods):
                food = " (" + str(i + 1) + ") " + " " * var_space + str(list_foods[i]) + " | "
            else:
                food = " " * 36 + "| "
            if i < len(list_drinks):
                drink = "(" + str(21 + i) + ") " + " " * var_space + str(list_drinks[i])
            else:
                drink = ""
            print(food, drink)
            i += 1

    # Three option to the user to choose from :
    print("\n (M) MAIN MENU                (P) PAYMENT                (E) EXIT\n" + "-" * 72)

    inp = input("Please Select Your Option: ").upper()
    if inp == 'M':
        print("\n" * 2)
        main()
        break
    if inp == 'E':
        print("-" * 32 + "THANK YOU" + "-" * 31 + "\n")
        break
    if inp == 'P':
        print("\n" * 2)
        show_bill()
        break

    try:
        int(inp)
        if (int(inp) <= len(list_foods) and int(inp) > 0) or (int(inp) <= len(list_drinks) + 20 and int(inp) > 20):
            try:
                print("\n" + "-" * 72 + "\n" + str(list_foods[int(inp) - 1]))
            except:
                pass
            try:
                print("\n" + "-" * 72 + "\n" + str(list_drinks[int(inp) - 21]))
            except:
                pass

            quantity = input("How Many You Want to Order?: ").upper()
            if int(quantity) > 0:
                cart_items[int(inp) - 1] += int(quantity)
                print("\n" * 2)
                print("Successfully Ordered!")
                food_and_drinks()
                break
            else:
                print("\n" * 2 + "ERROR: Invalid Input (" + str(quantity) + ").")
        except:
            print("\n" * 2 + "ERROR: Invalid Input (" + str(inp) + ").")

# shows the bill to the user
def show_bill():
    print("Generating your bill . . . ")
    print()
    wait_func()
    while True:
        print("-" * 30 + "YOUR BILL" + "-" * 30 + "\n")
        total_price = 0

        date_time = "\n\n" + " " * 17 + "*" * 35 + "\n" + " " * 17 + "DATE: " + str(datetime.datetime.now())[19] + "\n" + " " * 17 + "-" * 35
        i = 0
        print(date_time)
        while i < len(cart_items):
            if cart_items[i] != 0:
                if (i >= 0) and (i < 20):
                    date_time += "\n" + " " * 17 + str(list_foods[i]) + " x " + str(cart_items[i])
                    print(" " * 17 + str(list_foods[i]) + " x " + str(cart_items[i]))
                    total_price += list_item_price[i] * cart_items[i]
                if (i >= 20) and (i < 40):
                    date_time = date_time + "\n" + " " * 17 + str(list_drinks[i - 20]) + " x " + str(cart_items[i])
                    print(" " * 17 + str(list_drinks[i - 20]) + " x " + str(cart_items[i]))
                    total_price += list_item_price[i] * cart_items[i]
                i += 1
            else:
                i += 1

        date_time = date_time + "\n" + " " * 17 + "-" * 35 + "\n" + " " * 17 + "TOTAL PRICES:      Rs " + str(
            round(total_price, 2)) + "\n" + " " * 17 + "*" * 35
        print(" " * 17 + "-" * 35 + "\n" + " " * 17 + "TOTAL PRICES:      Rs " + str(round(total_price, 2)))

        print("\n (P) PAY                (M) MAIN MENU                (E) EXIT\n" + "-" * 72)
        inp = str(input("Please Select Your Option : ")).upper()
        if inp == 'P':
            print("\n" * 2)

```

```

        print(" Successfully Paid!")

    def_default()
elif inp == 'M':
    print("\n" * 2)
    main()
    break
elif ('E' in inp) or ('e' in inp):
    print("-" * 31 + "THANK YOU" + "-" * 31 + "\n")
    break

### Calling the functions according to precedence
def_default()
greeting()
user_info()

# OTP-function() is commented because to get the OTP , number should be registered
# on the website of third-party Twilio API

#otp_function()

main()

```

Python Code Output :

```
"C:\Program Files\Python310\python.exe" "D:\PP MINI PROJECT\miniproject.py"
```

WELCOME TO LA'OPALA CAFE !

Please enter your login details

Enter your name : *Pujan*

Enter the password : *Pujan@111*

Enter your Contact number : *9522234236*

Please wait ...

Hey, Pujan

Address : *Sgsits , Indore*

Flat No. : *12*

Please wait ...

Your address is Flat No. 12 , Sgsits , Indore

Please enter M to open Menu : *M*

-----ORDER PAGE-----

(F) FOODS AND DRINKS

(M) MAIN MENU

(E) EXIT

Please Select Your Option: *F*

-----ORDER FOODS & DRINKS-----

NO	FOOD NAME	PRICE		NO	DRINK NAME	PRICE
(1)	Aloo Tikki Burger	Rs 100.00		(21)	Americano	Rs 79.00
(2)	Bombay Bhe!l	Rs 115.00		(22)	Black Coffee	Rs 39.00
(3)	Chatpata Naan Aloo	Rs 100.00		(23)	Black Tea	Rs 20.00
(4)	Cheesy Corn	Rs 149.00		(24)	Cappuccino	Rs 69.00
(5)	Chocolate Brownie	Rs 165.00		(25)	Coca-Cola	Rs 35.00
(6)	Club Sandwich	Rs 180.00		(26)	Coconut Crush	Rs 89.00
(7)	Crispy Corn	Rs 119.00		(27)	Cold Coffee	Rs 89.00
(8)	Manchurian	Rs 119.00		(28)	Espresso	Rs 89.00
(9)	Masala Chaap	Rs 109.00		(29)	Fanta	Rs 35.00
(10)	Masala Dosa	Rs 100.00		(30)	Kesariya Milk	Rs 45.00
(11)	Masala Sandwich	Rs 149.00		(31)	Latte	Rs 99.00
(12)	Paneer Rolls	Rs 200.00		(32)	Lemon Tea	Rs 30.50
(13)	Pizza	Rs 120.00		(33)	Limca	Rs 35.00
(14)	Spicy Paazta	Rs 100.00		(34)	Milk Tea	Rs 15.00
(15)	Spicy Veg Momos	Rs 119.00		(35)	Milkshake	Rs 99.00
(16)	Spring Roll	Rs 149.00		(36)	Mixed Fruit Juice	Rs 75.00
(17)	Steamed Momos	Rs 109.00		(37)	Mojito	Rs 89.00
(18)	Tandoor Momos	Rs 129.00		(38)	Sprite	Rs 35.00
(19)	Veg Noodles	Rs 110.00		(39)	Virgin Mojito	Rs 99.00
(20)	White Sauce Paazta	Rs 129.00		(40)	Water	Rs 20.00

(M) MAIN MENU

(P) PAYMENT

(E) EXIT

Please Select Your Option: 4

Cheesy Corn Rs 149.00

Cappuccino Rs 69.00

How Many You Want to Order?: 2

Successfully Ordered!

-----ORDER FOODS & DRINKS-----

NO	FOOD NAME	PRICE		NO	DRINK NAME	PRICE
(1)	Aloo Tikki Burger	Rs 100.00		(21)	Americano	Rs 79.00
(2)	Bombay Bhe!l	Rs 115.00		(22)	Black Coffee	Rs 39.00
(3)	Chatpata Naan Aloo	Rs 100.00		(23)	Black Tea	Rs 20.00
(4)	Cheesy Corn	Rs 149.00		(24)	Cappuccino	Rs 69.00
(5)	Chocolate Brownie	Rs 165.00		(25)	Coca-Cola	Rs 35.00
(6)	Club Sandwich	Rs 180.00		(26)	Coconut Crush	Rs 89.00
(7)	Crispy Corn	Rs 119.00		(27)	Cold Coffee	Rs 89.00
(8)	Manchurian	Rs 119.00		(28)	Espresso	Rs 89.00
(9)	Masala Chaap	Rs 109.00		(29)	Fanta	Rs 35.00
(10)	Masala Dosa	Rs 100.00		(30)	Kesariya Milk	Rs 45.00
(11)	Masala Sandwich	Rs 149.00		(31)	Latte	Rs 99.00
(12)	Paneer Rolls	Rs 200.00		(32)	Lemon Tea	Rs 30.50
(13)	Pizza	Rs 120.00		(33)	Limca	Rs 35.00
(14)	Spicy Paazta	Rs 100.00		(34)	Milk Tea	Rs 15.00
(15)	Spicy Veg Momos	Rs 119.00		(35)	Milkshake	Rs 99.00
(16)	Spicy Veg Sandwich	Rs 100.00		(36)	Milkshake	Rs 99.00

(10) Masala Dosa	Rs 100.00		(30) Kesariya Milk	Rs 45.00
(11) Masala Sandwich	Rs 149.00		(31) Latte	Rs 99.00
(12) Paneer Rolls	Rs 200.00		(32) Lemon Tea	Rs 30.50
(13) Pizza	Rs 120.00		(33) Limca	Rs 35.00
(14) Spicy Paazta	Rs 100.00		(34) Milk Tea	Rs 15.00
(15) Spicy Veg Momos	Rs 119.00		(35) Milkshake	Rs 99.00
(16) Spring Roll	Rs 149.00		(36) Mixed Fruit Juice	Rs 75.00
(17) Steamed Momos	Rs 109.00		(37) Mojito	Rs 89.00
(18) Tandoor Momos	Rs 129.00		(38) Sprite	Rs 35.00
(19) Veg Noodles	Rs 110.00		(39) Virgin Mojito	Rs 99.00
(20) White Sauce Paazta	Rs 129.00		(40) Water	Rs 20.00

(M) MAIN MENU

(P) PAYMENT

(E) EXIT

Please Select Your Option: 20

White Sauce Paazta Rs 129.00

Water Rs 20.00

How Many You Want to Order?: 3

Successfully Ordered!

-----ORDER FOODS & DRINKS-----

Generating your bill . . .

Please wait .

..

-----YOUR BILL-----

DATE: 2022-11-16 15:22:53

Cheesy Corn Rs 149.00 x 2
Club Sandwich Rs 180.00 x 5
Veg Noodles Rs 110.00 x 10
White Sauce Paazta Rs 129.00 x 3
Americano Rs 79.00 x 12
Cold Coffee Rs 89.00 x 4
Latte Rs 99.00 x 5

TOTAL PRICES: Rs 4484.0

(P) PAY (M) MAIN MENU (E) EXIT

Please Select Your Option : -----YOUR BILL-----

Successfully Paid!

-----YOUR BILL-----

DATE: 2022-11-16 15:23:01

TOTAL PRICES: Rs 0

(P) PAY (M) MAIN MENU (E) EXIT

Please Select Your Option : **E**

-----THANK YOU-----

Process finished with exit code 0