DEMANDEST-AI POWERED FOOD DEMAND FORECASTER

TEAM ID : PNT2022TMID47942

Python coding:

```
print("\n" * 5)


import datetime

import os


list_foods = []

 list_drinks = []

list_services = []


list_item_price = [0] * 100


var_discount_1 = 200

var_discount_2 = 1000

var_discount_3 = 5000

var_discount_1_rate = 0.05

var_discount_2_rate = 0.10

var_discount_3_rate = 0.15




navigator_symbol = "/"

 if os.name == "nt":
```

```
    navigator_symbol = "\\"



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




def def_main():
while True:
        print("" * 28 + "FOOD ORDERING SYSTEM" + "" * 24 + "\n")
        print("" * 31 + "MAIN MENU" + "" * 32 + "\n"
            "\t(O) ORDER\n"
            "\t(R) REPORT\n"
            "\t(P) PAYMENT\n"
            "\t(E) EXIT\n" +
            "_" * 72)


        input_1 = str(input("Please Select Your Operation: ")).upper()
        if (len(input_1) == 1):
            if (input_1 == 'O'):
              print("\n" * 10)
             def_order_menu()
```

```
            break
elif (input_1 == 'R'):
            print("\n" * 10)
            def_report()
             break
elif (input_1 == 'P'):
      print("\n" * 10)
      def_payment()
       break
elif (input_1 == 'E'):
            print("" * 32 + "THANK YOU" + "" * 31 + "\n")
            break
          else:
            print("\n" * 10 + "ERROR: Invalid Input (" + str(input_1) + "). Try again!")
       else:
          print("\n" * 10 + "ERROR: Invalid Input (" + str(input_1) + "). Try again!")
 def def_order_menu():                                                    while True:
       print("" * 31 + "ORDER PAGE" + "" * 31 + "\n"
            "\t(F) FOODS AND DRINKS\n"
            "\t(O) OTHER SERVICES\n"
            "\t(M) MAIN MENU\n"
            "\t(E) EXIT\n" +
            "_" * 72)

       input_1 = str(input("Please Select Your Operation: ")).upper()
```

```
 if len(input_1) == 1:
if (input_1 == 'F'):
print("\n" * 10)
def_food_drink_order()
            break
elif (input_1 == 'O'):
   print("\n" * 10)
   def_other_services()
   break
elif (input_1 == 'M'):
   print("\n" * 10)
   def_main()
   break
 elif (input_1 == 'E'):
   print("" * 32 + "THANK YOU" + "" * 31 + "\n")
            break
else:
            print("\n" * 10 + "ERROR: Invalid Input (" + str(input_1) + "). Try again!")
 else:
         print("\n" * 10 + "ERROR: Invalid Input (" + str(input_1) + "). Try again!")
 def def_full_file_reader():
   file_foods = open('files'+navigator_symbol+'list_foods.fsd', 'r')
for i in file_foods:
 list_foods.append(str(i.strip()))
file_foods.close()
```

```
    file_drinks = open('files'+navigator_symbol+'list_drinks.fsd', 'r')
    for i in file_drinks:
 list_drinks.append(str(i.strip()))
file_drinks.close()


    file_services = open('files'+navigator_symbol+'list_services.fsd', 'r')
for i in file_services:
list_services.append(str(i.strip()))
file_services.close()
    i = 0
while i <= (len(list_foods) - 1):
    if '$' in list_foods[i]:
        list_foods[i] = str(list_foods[i][:list_foods[i].index('$') - 1]) + ' ' * (20 - (list_foods[i].index('$') - 1)) +
str(list_foods[i][list_foods[i].index('$'):])
       i += 1
    i = 0     while i <= (len(list_drinks) - 1):
if '$' in list_drinks[i]:
        list_drinks[i] = str(list_drinks[i][:list_drinks[i].index('$') - 1]) + ' ' * (20 - (list_drinks[i].index('$') - 1))
+ str(list_drinks[i][list_drinks[i].index('$'):])
       i += 1
    i = 0
while i <= (len(list_services) - 1):
if '$' in list_services[i]:
        list_services[i] = str(list_services[i][:list_services[i].index('$') - 1]) + ' ' * (20 -
(list_services[i].index('$') - 1)) + str(list_services[i][list_services[i].index('$'):])
```

```
        i += 1
def_full_file_reader()


def def_file_sorter():
    global list_foods, list_drinks, list_services
list_foods = sorted(list_foods)
list_drinks = sorted(list_drinks)
list_services = sorted(list_services)
    i = 0
    while i < len(list_foods):
        list_item_price[i] = float(list_foods[i][int(list_foods[i].index("$") + 2):])
        i += 1
    i = 0
while i < len(list_drinks):
list_item_price[40 + i] = float(list_drinks[i][int(list_drinks[i].index("$") + 2):])
        i += 1
    i = 0     while i < len(list_services):
        list_item_price[80 + i] = float(list_services[i][int(list_services[i].index("$") + 2):])
        i += 1
def_file_sorter()


def def_food_drink_order():
while True:
            print("" * 26 + "ORDER FOODS & DRINKS" + "" * 26)
print(" |NO| |FOOD NAME|        |PRICE|   |  |NO| |DRINK NAME|       |PRICE|")
```

```
        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(41 + i) + ")" + " " + str(list_drinks[i])
            else:
                drink = ""
print(food, drink)
i += 1

        print("\n (M) MAIN MENU              (P) PAYMENT              (E) EXIT\n" + "_" * 72)

        input_1 = input("Please Select Your Operation: ").upper()
 if (input_1 == 'M'):
print("\n" * 10)
def_main()
break
```

```
if (input_1 == 'E'):
        print("" * 32 + "THANK YOU" + "" * 31 + "\n")
        break
if (input_1 == 'P'):
print("\n" * 10)
def_payment()
 break
 try:
 int(input_1)
        if ((int(input_1) <= len(list_foods) and int(input_1) > 0) or (int(input_1) <= len(list_drinks) + 40
and int(input_1) > 40)):
            try:
              print("\n" + "_" * 72 + "\n" + str(list_foods[int(input_1) - 1]))
            except:
 pass                try:
              print("\n" + "_" * 72 + "\n" + str(list_drinks[int(input_1) - 41]))
            except:
pass
input_2 = input("How Many You Want to Order?: ").upper()
if int(input_2) > 0:
list_item_order[int(input_1) - 1] += int(input_2)
print("\n" * 10)
print("Successfully Ordered!")
def_food_drink_order()
break
else:
print("\n" * 10 + "ERROR: Invalid Input (" + str(input_2) + "). Try again!")
except:
```

```
print("\n" * 10 + "ERROR: Invalid Input (" + str(input_1) + "). Try again!")
```

MODEL OUTPUT: