AUTOMATION OF RESTAURANT WITH OPTIMIZED PARKING

CONTENT

01 PROBLEM STATEMENT

02 TEAM ROLES AND RESPONSIBILITIES

03 LIST OF INBUILT MODULES

04 PSUEDOCODE

05 OUTPUTS

06 LEARNINGS

07 CONCLUSION

TEAM MEMBERS

- SUDHARSHAN SRINIVASAN
  PES2UG23AM106
- SIOBHAN GHOSAL
  PES2UG23AM099
- SWATHI SHARMA
  PES2UG23EC164
- SHUCHIKA JOY
  PES2UG23CS573

PROBLEM STATEMENT

This project aims to automate a restaurant and optimize its parking.
This program will be run by a restaurant administrator.
If the customer decides to dine in, the restaurant will have a valet parking option for customers with vehicles.
During the time of entry, the program does the work of assisting a valet in finding a parking space to park the vehicle. It records the time and name of the customer and allocates a parking spot to the vehicle which it returns to the user. The user can use the information and park their vehicle in the allotted spot. The system generates a bill for every customer based on the orders made. According to the dishes consumed by the customer, the amount of ingredients will be reduced and the system handles the restocking process when the stock is low, under the supervision of the administrator. While exiting the parking space, the program will compute a parking fee which has to be paid by the customer.

TEAM ROLES

- **SUDARSHAN**

-Give the administrator a monthly report based on collected data in excel. This can be used to view various statistics on the restaurant's sales.

-Integrating the backend with the GUI.

- **SIOBHAN**

-Simulate automated entry generation to plot relevant graphs (for practical purposes).

-GUI

- **SHUCHIKA**

-Provides customers assistance with parking. Guides vehicles directly to available parking spaces, saving their time and preventing unwanted wandering.

-Take vehicle number and time of entry as input

-Allot available parking spot closest to exit

-While exiting, automatically calculate and display fee

-Have an administrator mode to monitor the current state of the

-parking lot

- **SWATHI**

-Takes orders from the user and keeps track of all the orders taken

-Gives a message when an ingredient is running low on stock

OBJECTIVES

□ Suggests required restocking amount of ingredients based on past trends

□ Provides customers assistance with parking.Guides vehicles directly to available parking spaces, saving their time and preventing unwanted wandering

□ Give the administrator a monthly report based on collected data in excel. This can be used to view the various statistics of the restaurant's sales.

□ Takes orders from the user and keeps track of all the orders taken

□ Gives a message when an ingredient is running low on stock.

□ Based on previous orders, it returns a prediction of how much longer each ingredient will last. This can be viewed as a suggestion to the user who can then restock the ingredients to any amount of units.

INBUILT MODULES

CSV

mysql. connector

time

matplotlib.pyplot

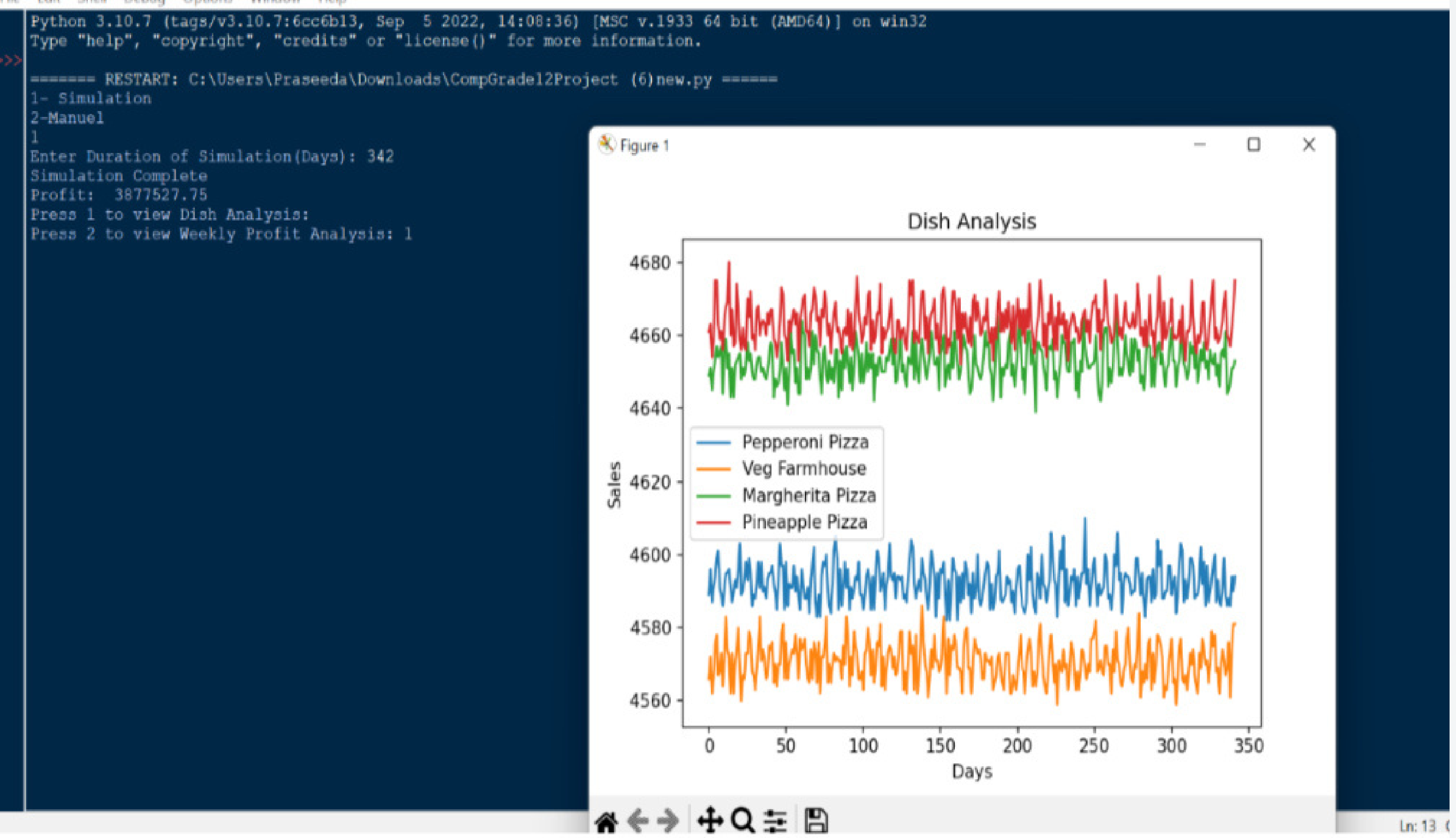
random

streamlIt

PSUEDOCODE

```
1. Take input from user for "Manual" or "Simulation" Modes
2. For "Simulation":
    (a) Initialize all values for ingredients
    (b) Create a table ingredients to insert values of ingredients for specific recipes
    (c) Take input from user for number of days to be simulated

3. For "Manual":
    (a) Take Customer Name as input
    (b) Ask if the customer is a "Walk-In" or "Valet"
    (c) if "Walk-In":
        (i) Ask for Number of people
        (ii) Direct them to required table (Show a message if the restaurant is full)
        (iii) Display the regular bill without any additional parking charges

    (d) if "Valet":
        (i) Direct them to parking spot closest to exit
        (ii) Display the current status of the parking lot
        (iii) At the end of the meal add the parking charges with the price of the meal in the bill

    (e) Take the order from the user
    (f) If the inventory for a specific order is low display a message for restocking the inventory
    (g) Display a restock optioN with suggested amount of ingredients to be restocked
    (h) Display an option to exit the restaurant (Show an error message if the restaurant is currently empty):
        (a) To exit the restaurant take user name as input and check if the customer is a Valet or Walk-In
        (b) Display the bill accordingly
```

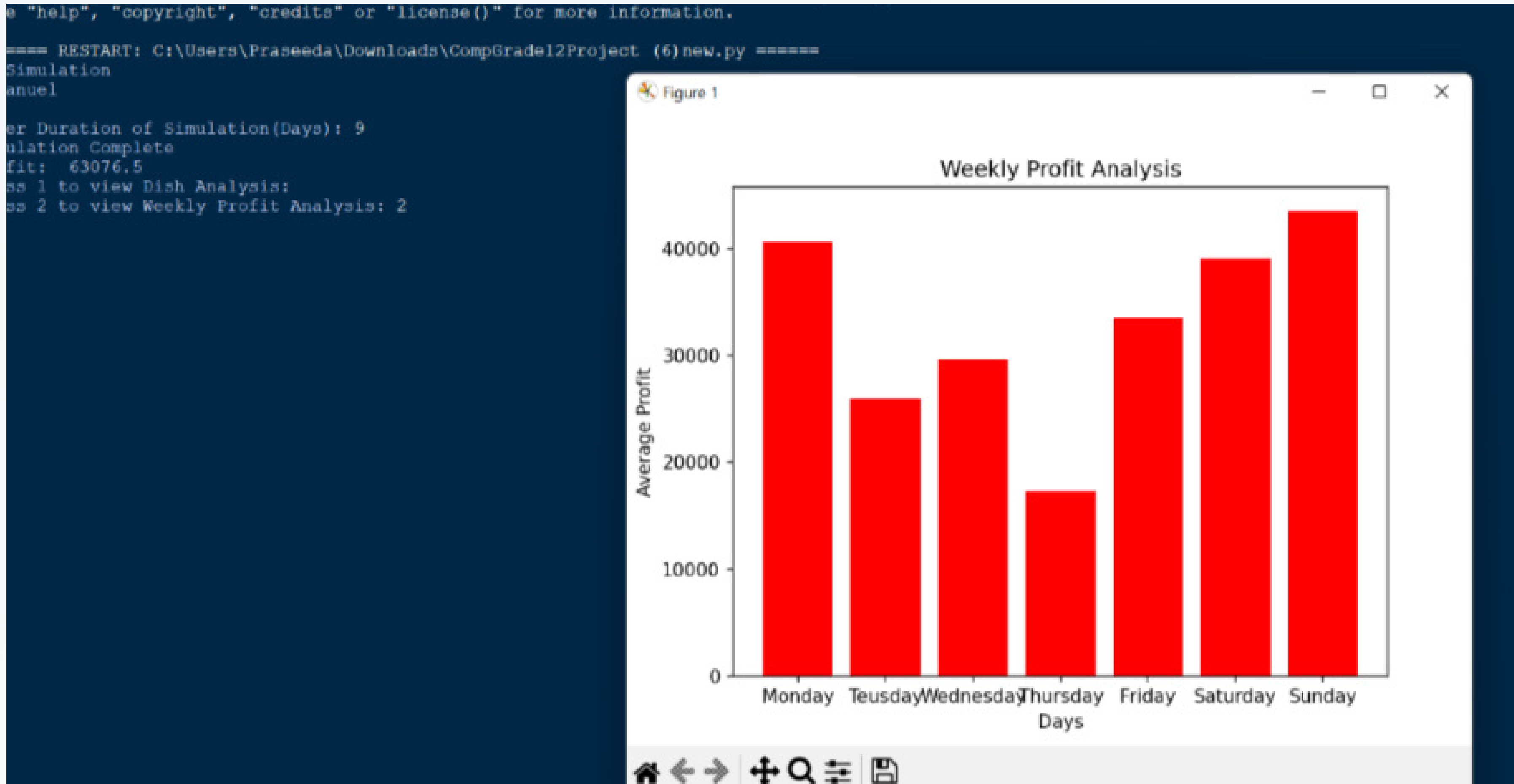
OUTPUTS



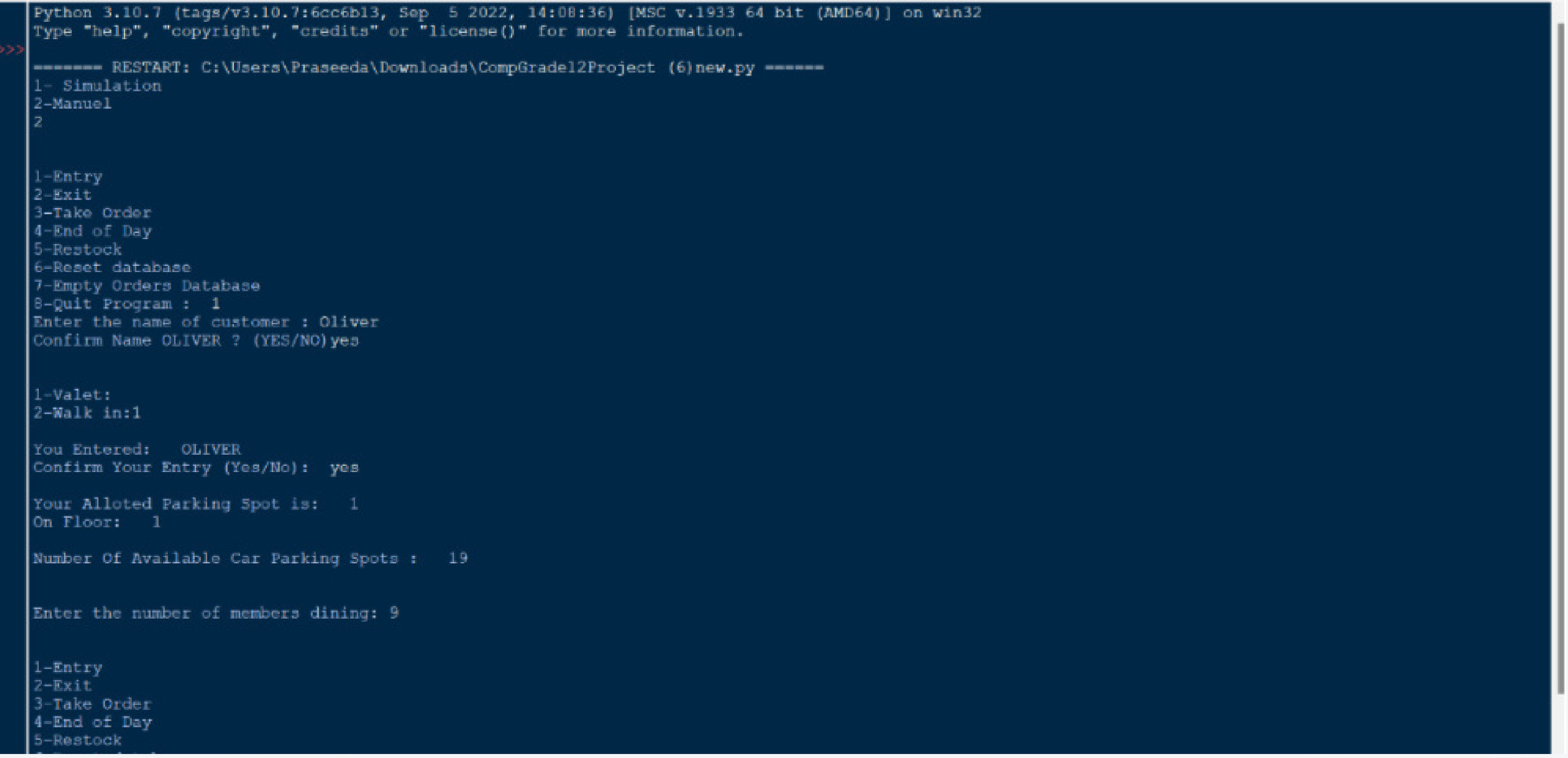
```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

======= RESTART: C:\Users\Praseeda\Downloads\CompGrade12Project (6)new.py ======
1- Simulation
2-Manuel
1
Enter Duration of Simulation(Days): 342
Simulation Complete
Profit:  3877527.75
Press 1 to view Dish Analysis:
Press 2 to view Weekly Profit Analysis: 1
```

The dish analysis graph gives a comparison of frequency of orders per type of pizza.

The weekly analysis chart shows the average profits made on each day of the week while taking into account restocking costs



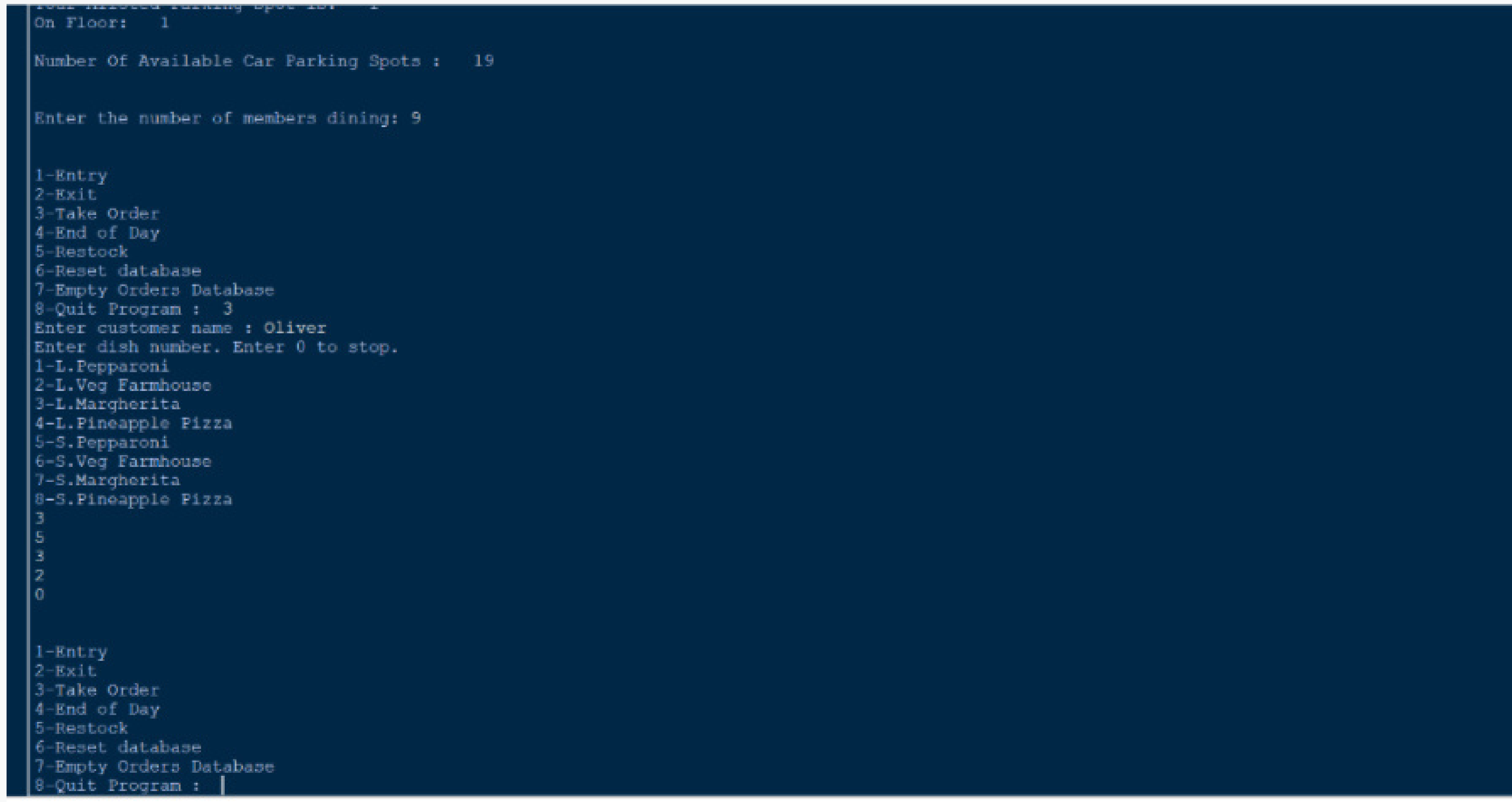
```
e "help", "copyright", "credits" or "license()" for more information.

==== RESTART: C:\Users\Praseeda\Downloads\CompGrade12Project (6)new.py ======
Simulation
anuel

er Duration of Simulation(Days): 9
ulation Complete
fit:  63076.5
ss 1 to view Dish Analysis:
ss 2 to view Weekly Profit Analysis: 2
```



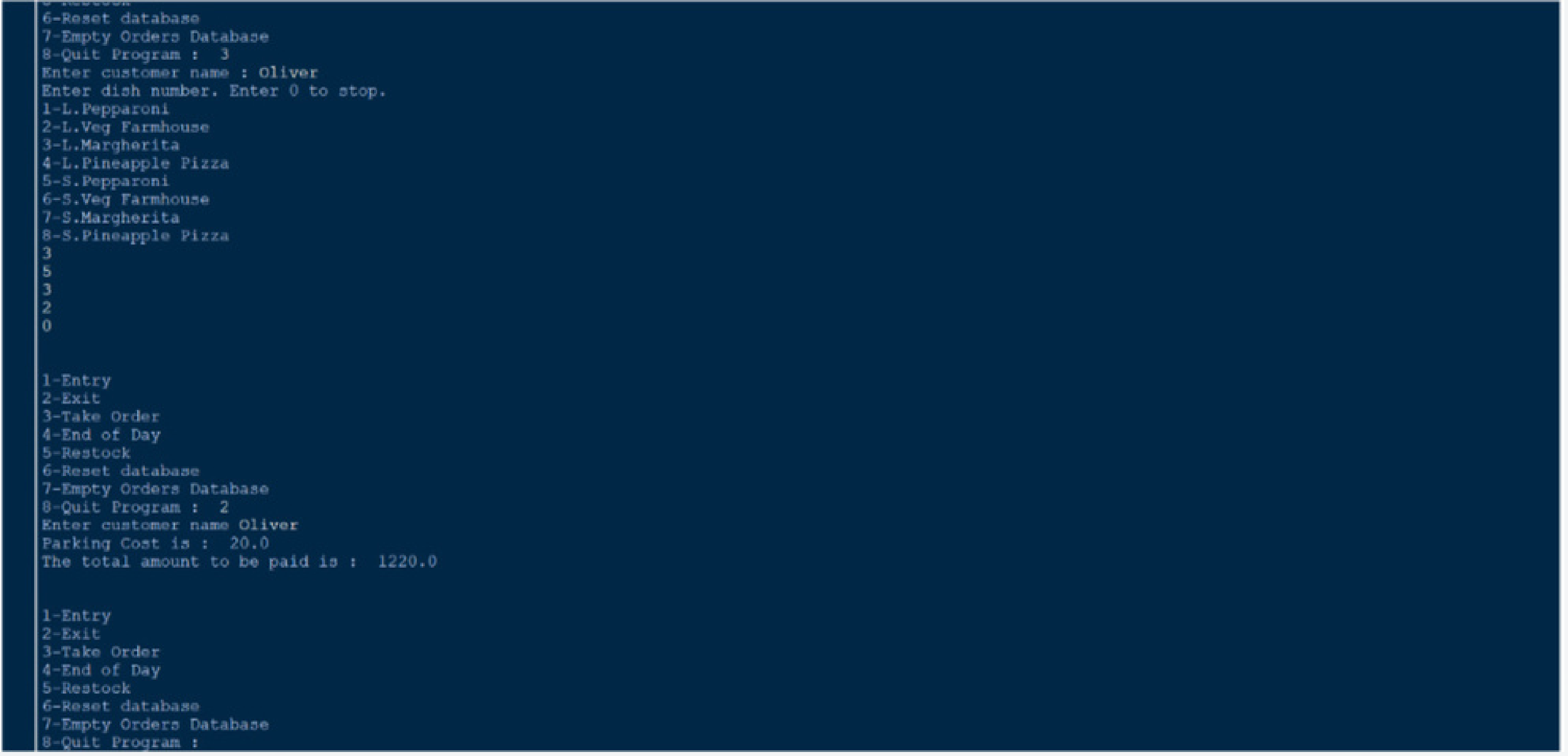
```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:\Users\Praseeda\Downloads\CompGrade12Project (6)new.py ======
1- Simulation
2-Manuel
2



1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :  1
Enter the name of customer : Oliver
Confirm Name OLIVER ? (YES/NO)yes


1-Valet:
2-Walk in:1

You Entered:   OLIVER
Confirm Your Entry (Yes/No):  yes

Your Alloted Parking Spot is:   1
On Floor:   1

Number Of Available Car Parking Spots :   19


Enter the number of members dining: 9


1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
```

This segment of the program automates the entry of a customer. The name is to be entered by the administrator which will then be used to distinguish each customer in later parts of the program. If customer chooses valet parking, the time of his/her entry is noted and a spot in the parking lot is occupied.

This segment automates the ordering of dishes by a customer. The administrator enters the name of customer and enters the dishes by the dish number. Based on the order made by the customer, the program records a reduce in the stock of the ingredients. It also computes a certain cost based on the orders made by the customer



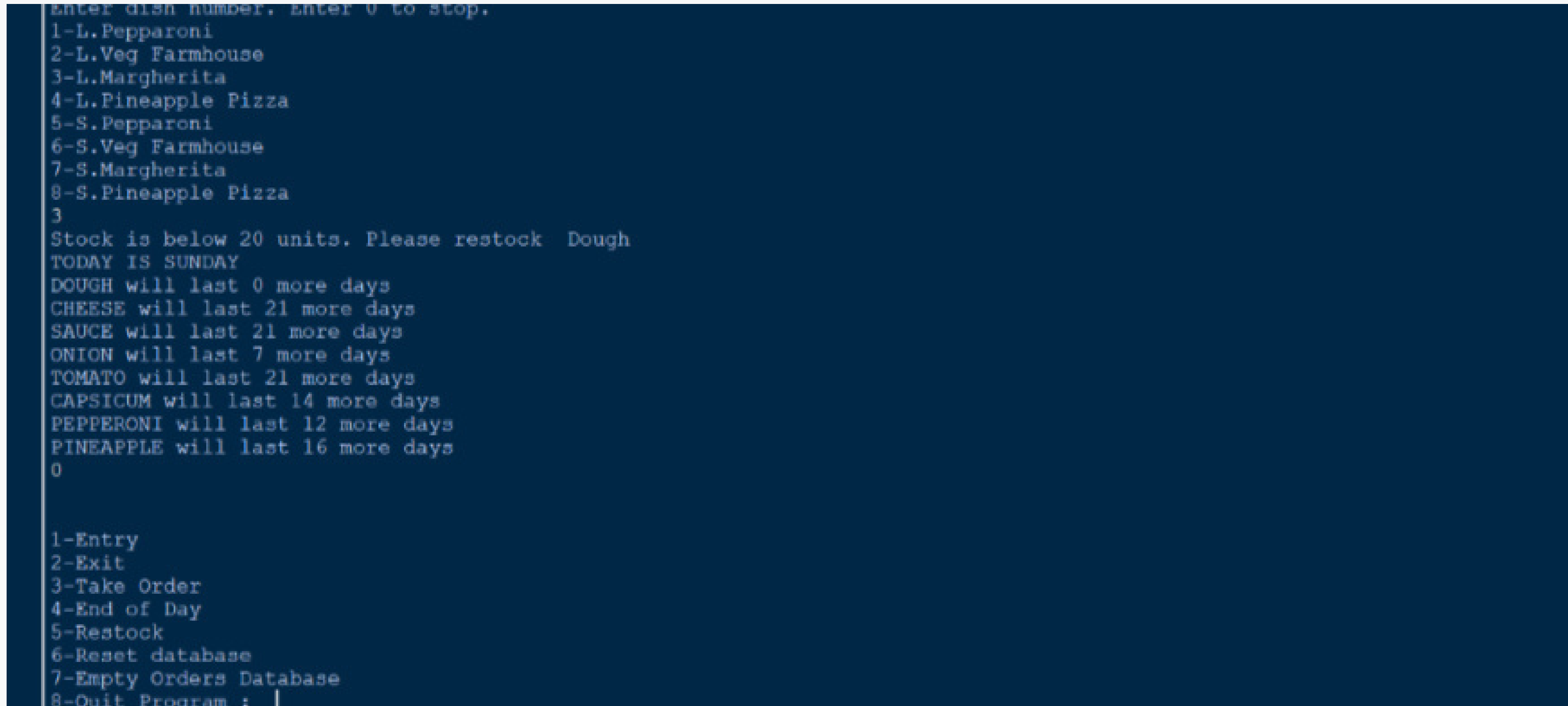
```
On Floor:   1

Number Of Available Car Parking Spots :   19


Enter the number of members dining: 9


1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :  3
Enter customer name : Oliver
Enter dish number. Enter 0 to stop.
1-L.Pepparoni
2-L.Veg Farmhouse
3-L.Margherita
4-L.Pineapple Pizza
5-S.Pepparoni
6-S.Veg Farmhouse
7-S.Margherita
8-S.Pineapple Pizza
3
5
3
2
0



1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :
```



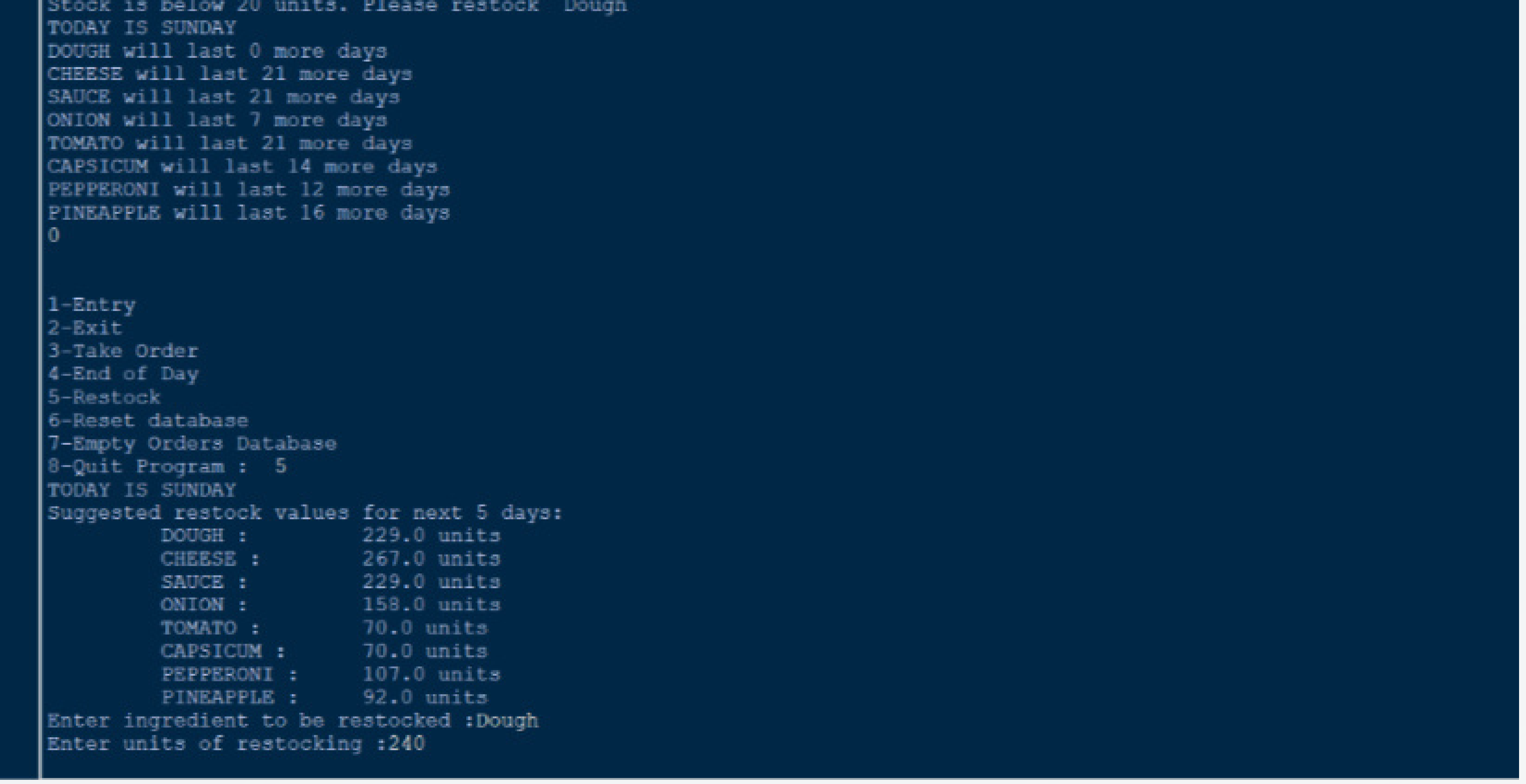
```
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :  3
Enter customer name : Oliver
Enter dish number. Enter 0 to stop.
1-L.Pepparoni
2-L.Veg Farmhouse
3-L.Margherita
4-L.Pineapple Pizza
5-S.Pepparoni
6-S.Veg Farmhouse
7-S.Margherita
8-S.Pineapple Pizza
3
5
3
2
0


1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :  2
Enter customer name Oliver
Parking Cost is :  20.0
The total amount to be paid is :  1220.0


1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :
```

This segment automates the exit of a customer. If the customer has chosen valet parking, the parking charges is added to the dish cost and displayed. If the customer opted for a walk in, then the cost of the dishes is displayed which need to be paid by the customer

While ordering a dish, if the stock of an ingredient falls below a certain predetermined lower limit, then a message will be displayed informing the user of
the low stock of the said ingredient. A prediction is also made based on
previous orders and displays an approximate value of how much longer each
ingredient will last.



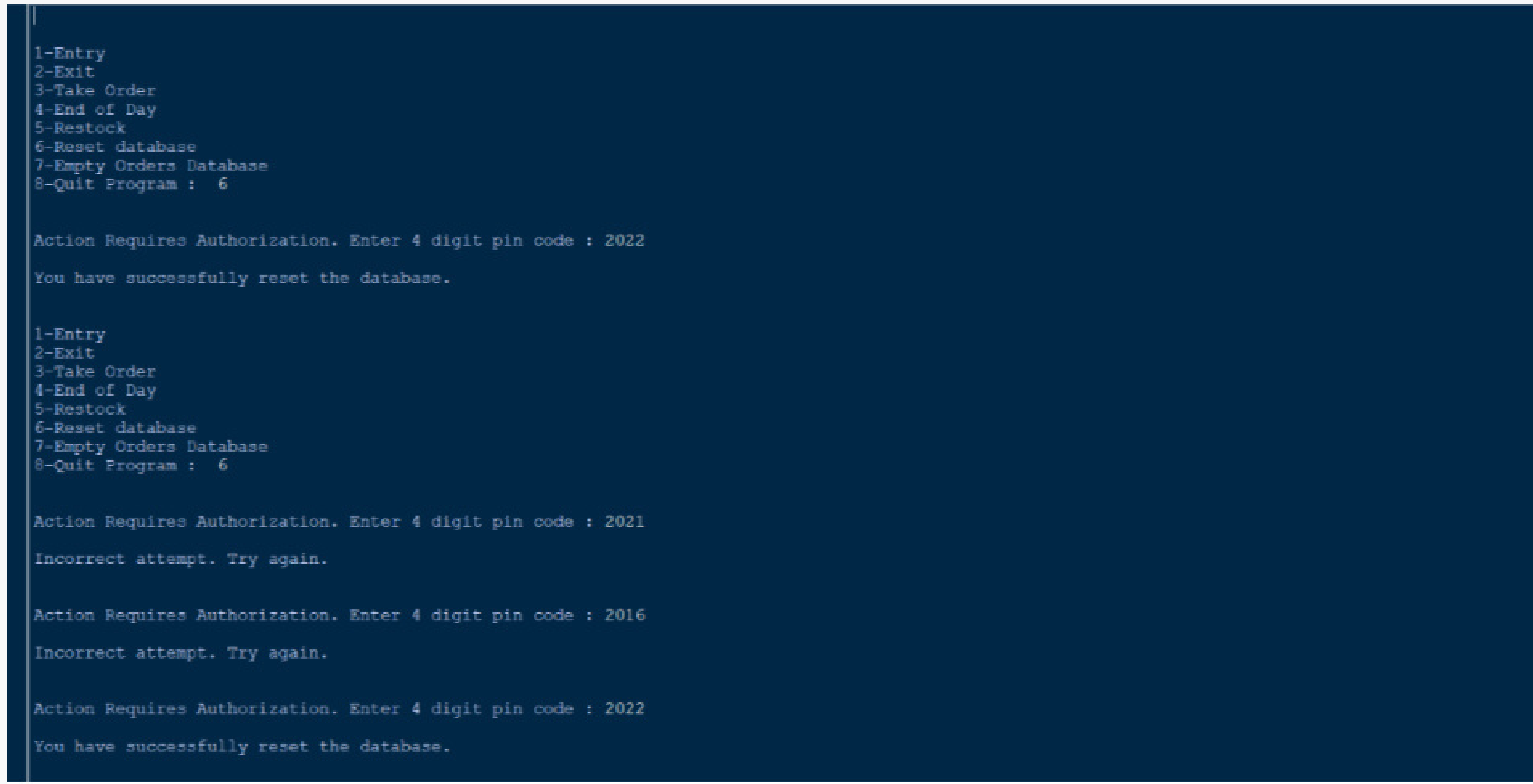
```
Enter dish number. Enter 0 to stop.
1-L.Pepparoni
2-L.Veg Farmhouse
3-L.Margherita
4-L.Pineapple Pizza
5-S.Pepparoni
6-S.Veg Farmhouse
7-S.Margherita
8-S.Pineapple Pizza
3
Stock is below 20 units. Please restock  Dough
TODAY IS SUNDAY
DOUGH will last 0 more days
CHEESE will last 21 more days
SAUCE will last 21 more days
ONION will last 7 more days
TOMATO will last 21 more days
CAPSICUM will last 14 more days
PEPPERONI will last 12 more days
PINEAPPLE will last 16 more days
0


1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :
```



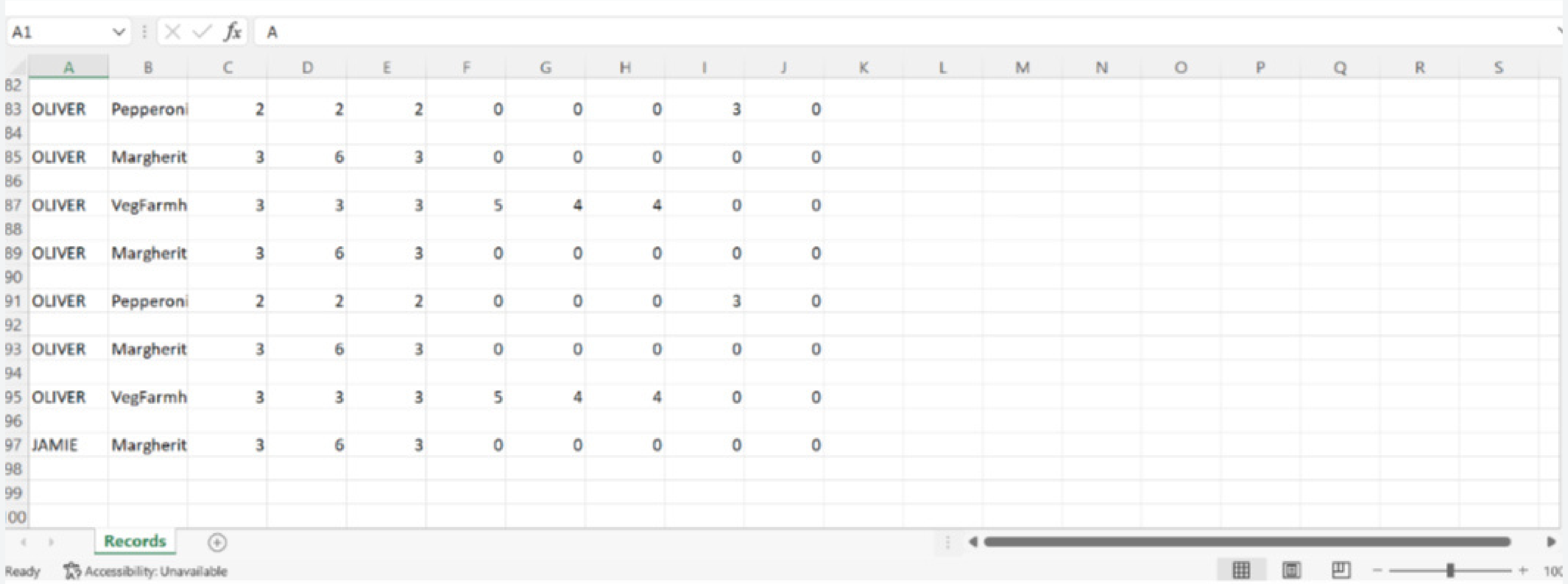
```
Stock is below 20 units. Please restock  Dough
TODAY IS SUNDAY
DOUGH will last 0 more days
CHEESE will last 21 more days
SAUCE will last 21 more days
ONION will last 7 more days
TOMATO will last 21 more days
CAPSICUM will last 14 more days
PEPPERONI will last 12 more days
PINEAPPLE will last 16 more days
0


1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :  5
TODAY IS SUNDAY
Suggested restock values for next 5 days:
         DOUGH :         229.0 units
         CHEESE :        267.0 units
         SAUCE :         229.0 units
         ONION :         158.0 units
         TOMATO :        70.0 units
         CAPSICUM :      70.0 units
         PEPPERONI :     107.0 units
         PINEAPPLE :     92.0 units
Enter ingredient to be restocked :Dough
Enter units of restocking :240
```

The program makes a prediction based on previous orders and displays the units of restocking needed for each and every ingredient to last the next five days.
This prediction can be seen as a suggestion to the user as the user decides which ingredients need to be restocked and by how many units.

Resetting the databases or quitting the program is a task that should be executed only by a select few to prevent loss of data. Thus , a password verification is placed on these tasks.



```
1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :  6


Action Requires Authorization. Enter 4 digit pin code : 2022

You have successfully reset the database.


1-Entry
2-Exit
3-Take Order
4-End of Day
5-Restock
6-Reset database
7-Empty Orders Database
8-Quit Program :  6


Action Requires Authorization. Enter 4 digit pin code : 2021

Incorrect attempt. Try again.


Action Requires Authorization. Enter 4 digit pin code : 2016

Incorrect attempt. Try again.


Action Requires Authorization. Enter 4 digit pin code : 2022

You have successfully reset the database.
```



	A	B	C	D	E	F	G	H	I	J
82										
83	OLIVER	Pepperoni	2	2	2	0	0	0	3	0
84										
85	OLIVER	Margherit	3	6	3	0	0	0	0	0
86										
87	OLIVER	VegFarmh	3	3	3	5	4	4	0	0
88										
89	OLIVER	Margherit	3	6	3	0	0	0	0	0
90										
91	OLIVER	Pepperoni	2	2	2	0	0	0	3	0
92										
93	OLIVER	Margherit	3	6	3	0	0	0	0	0
94										
95	OLIVER	VegFarmh	3	3	3	5	4	4	0	0
96										
97	JAMIE	Margherit	3	6	3	0	0	0	0	0
98										
99										
00										

Displays all the orders taken in the restaurant in a CSV file. Restarting the program will not result in the loss of any information. This information can be used by the administrator to store records of all those who have ever entered the restaurant and the dishes they have ordered.

LIMITATIONS

□ Orders have to be taken one by one in manual mode.

□ The program does not automate the processes happening between the time of entry and exit.

□ This program does not have a very user friendly or visually appealing interface which can make it difficult for the user to operate.

□ Simulation may take some time for large number for entries.

□ Graphs can be viewed only after the program terminates.

CONCLUSION

This System of automated restaurant helps with keeping track of stock for the smooth running of the restaurant. The optimized parking avoids over trafficking and is of better convenience to the customers.