**Abstract of the project**

The main idea revolves around a restaurant that deals with three diﬀerent kinds of customers and serves them in diﬀerent ways. The Project is built to assist that small business. It also stores information about the customers served and the total earnings of the shop.

Searching for the served Customers can be done. The delivery charges are calculated based on the shortest distance on the map and 100 RS\_/ per KM is charged as delivery charges.

Types of Customers

* Walk-In Customers

They will be served on the basis of their age. The older customer will be served ﬁrst.

* Home Delivery Customers

They will be served on the basis of LIFO

* Dine-In Customers

They will be served on the basis of First come and ﬁrst served basis.

**Problem Statement**

Design a restaurant management system that efficiently serves walk-in, home delivery, and dine-in customers. The system should prioritize service based on the age of walk-in customers, LIFO for home delivery customers, and first come, first serve basis for dine-in customers. For home delivery customers, the system should also calculate the shortest path between the restaurant and the customer's location using Dijkstra's algorithm to ensure that the minimum path is taken.

**Topics used**

**1. Linked List** - The whole implementation of Customer's Enqueue and dequeue is based on the Linked List.

**2. Stack** - It is used in placing orders and serving Home Delivery Customers. The customers who places order at last will be served first (LIFO).

**3.Dynamic Queue** - It is used in placing orders and serving Dine-In Customers i.e. the customers are served on first cum first serve basis.

**4. Dynamic Priority Queue** - It is used in placing orders and serving Walk-In Customers i.e. they are served on their basis of age, older person is served first.

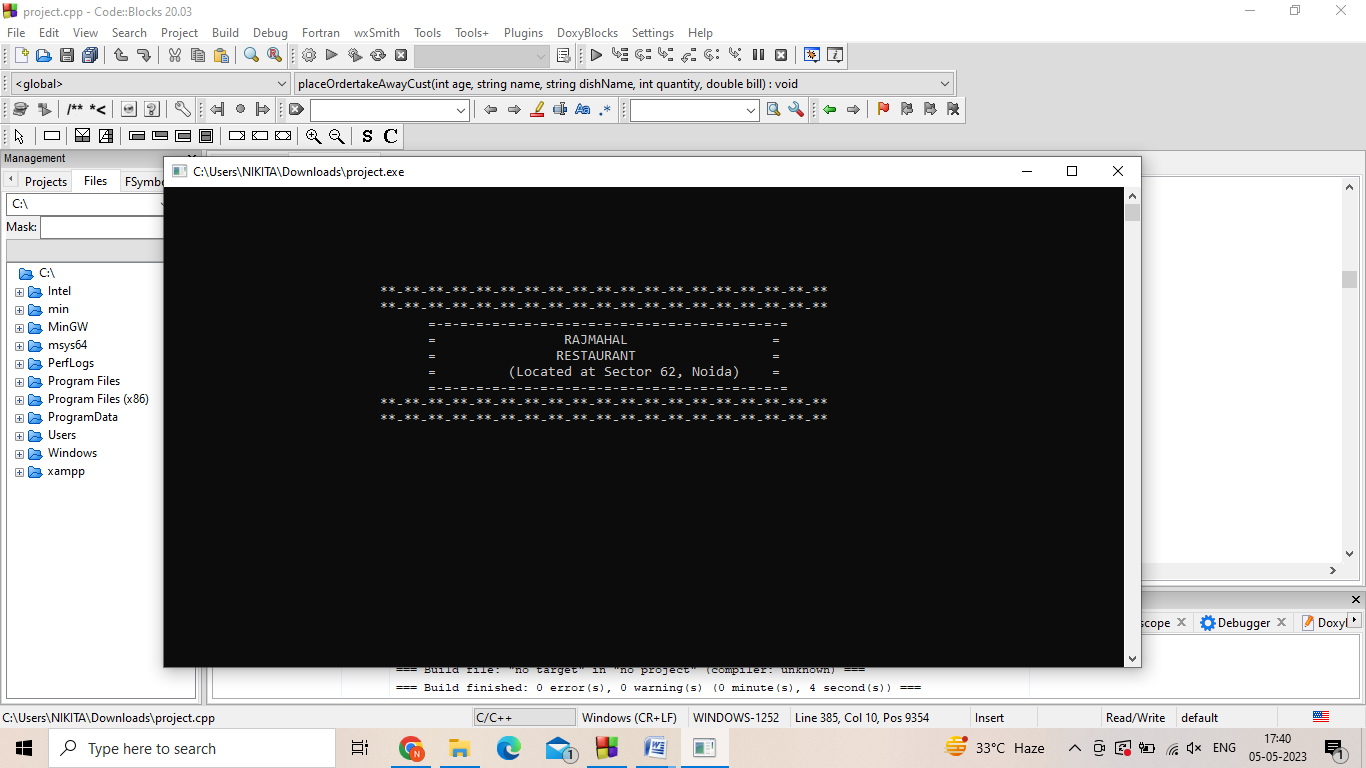
**5.Graph (Adjacency List Representation)** - It is used for the Map for available home delivery options.

**6. AVL Tree** - The Served Customers are saved in it for eﬃcient insertion, deletion, and searching. The implementation is based on the lexicographical comparison of strings.

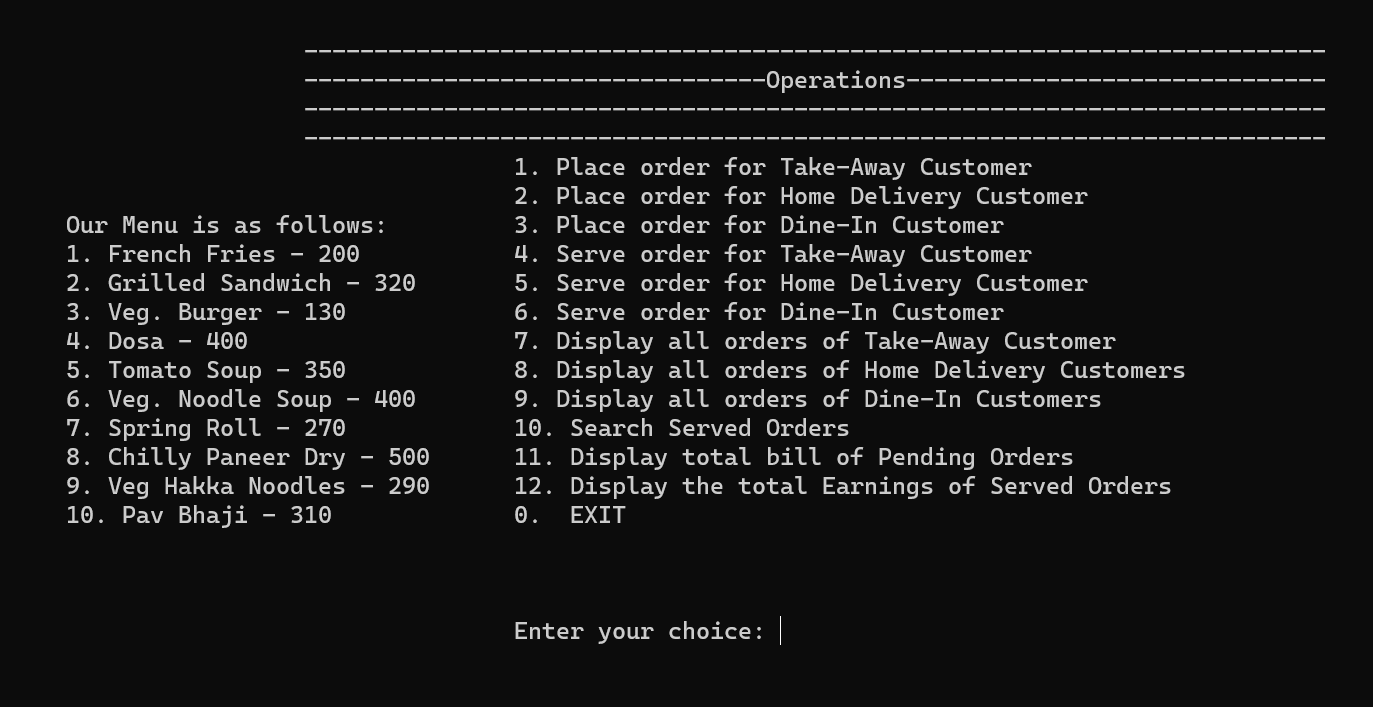
**7. Dijkstra Algorithm** - It is used to ﬁnd the shortest possible distance from the restaurant to each delivery location each having a certain distance from the restaurant and on this basis, the Delivery charges are calculated (delivery charges being Rs. 100/km).

**Design of the project**

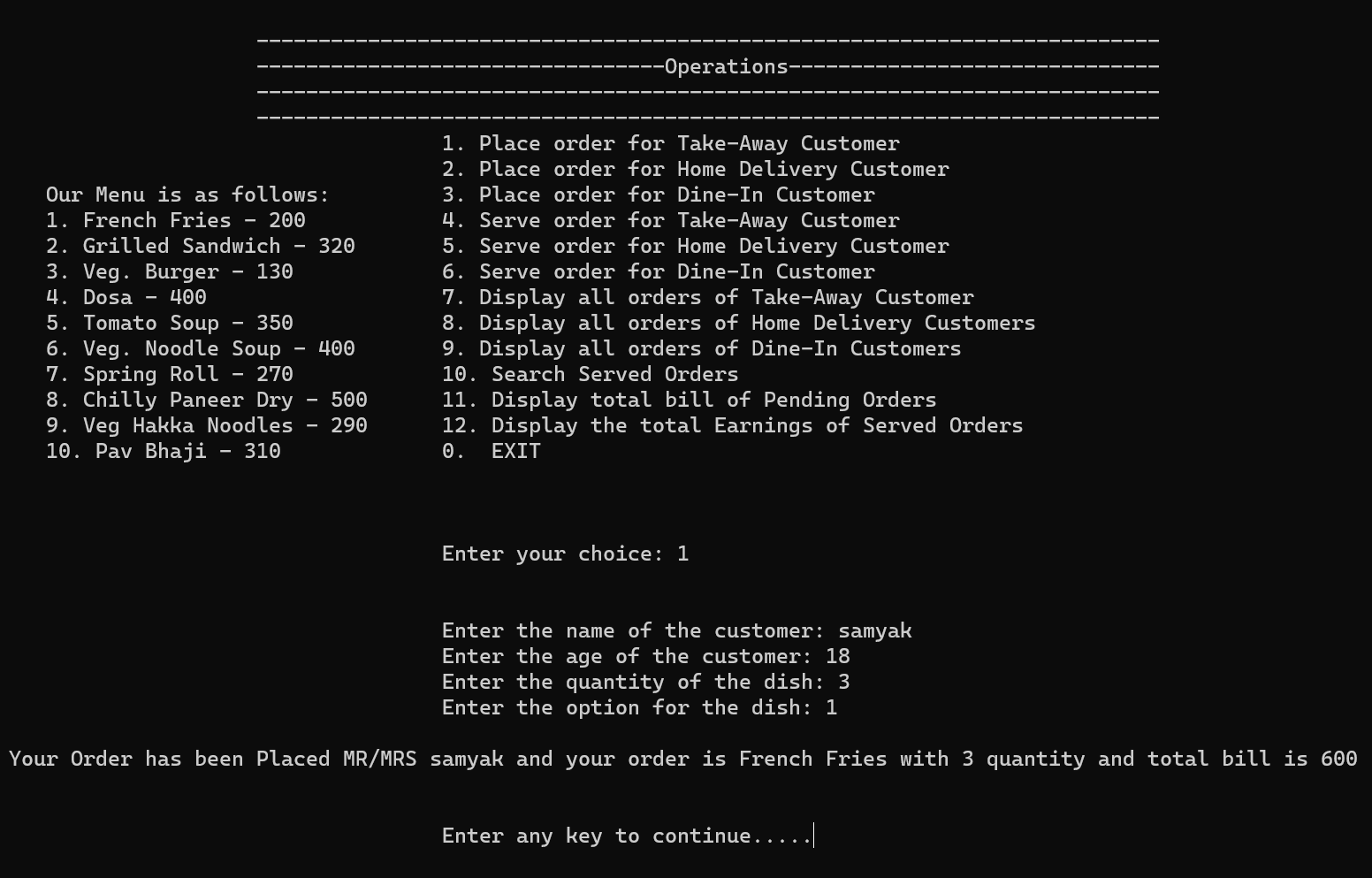
Welcome page



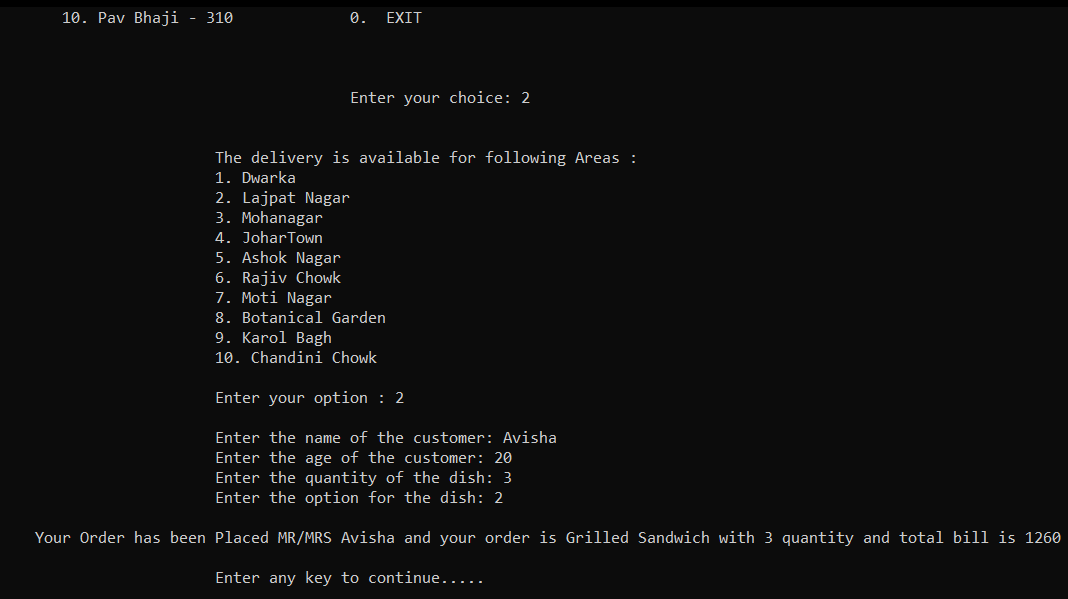
Overview



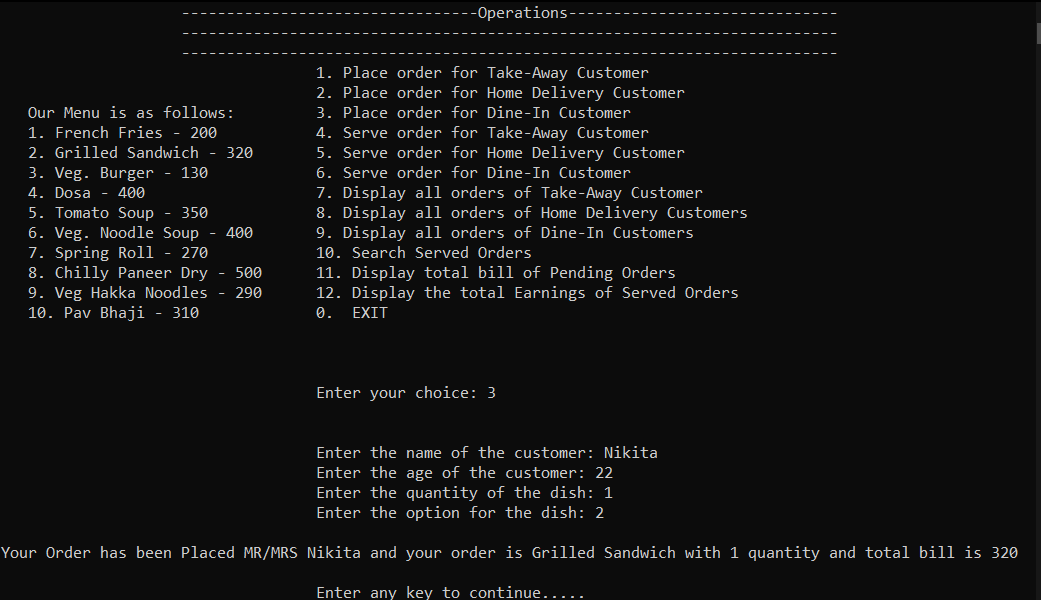
Take-away customer



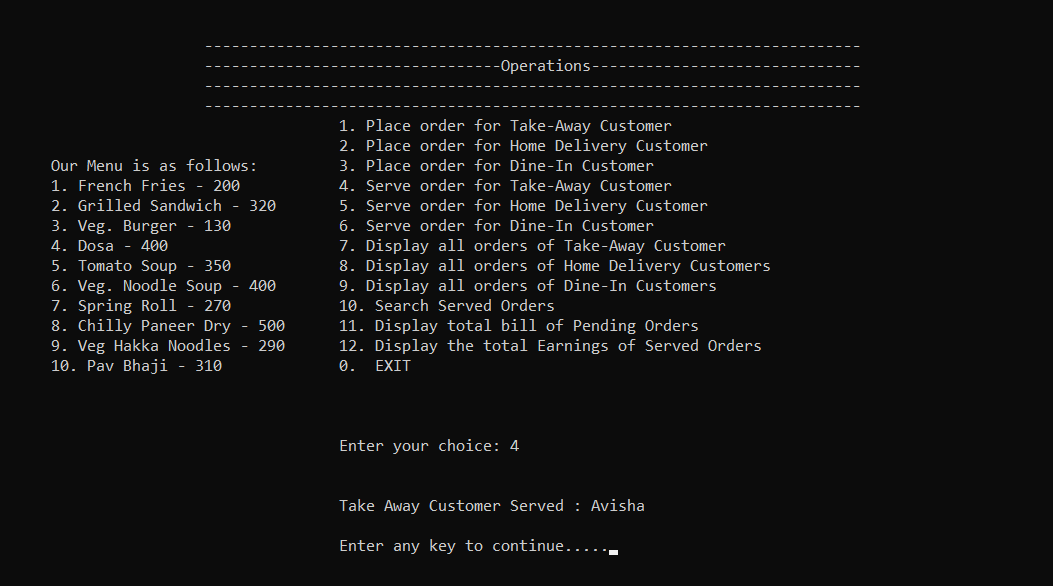
Home-Delivery customer



Dine-In Customer



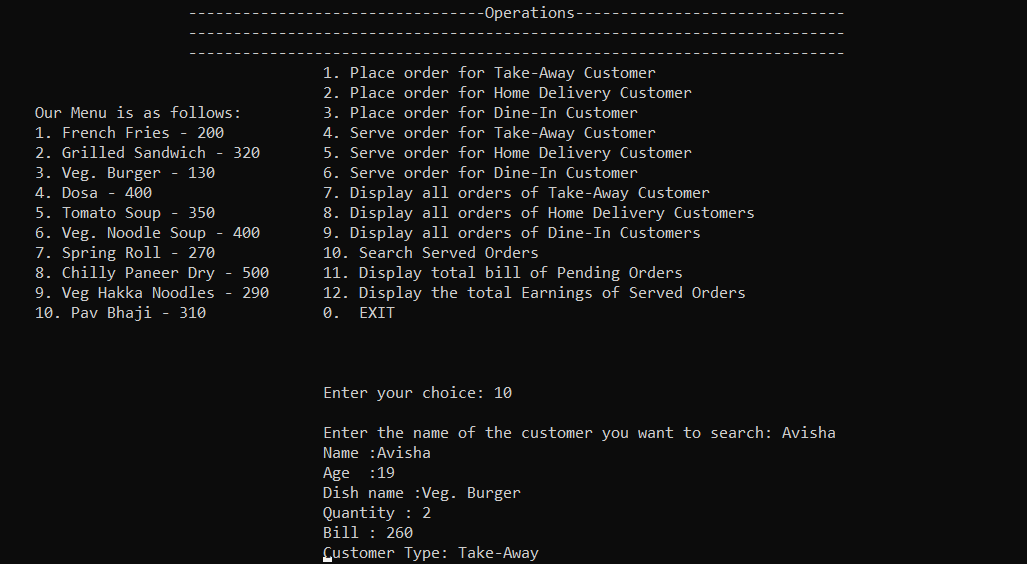
Serve for Take-Away customer



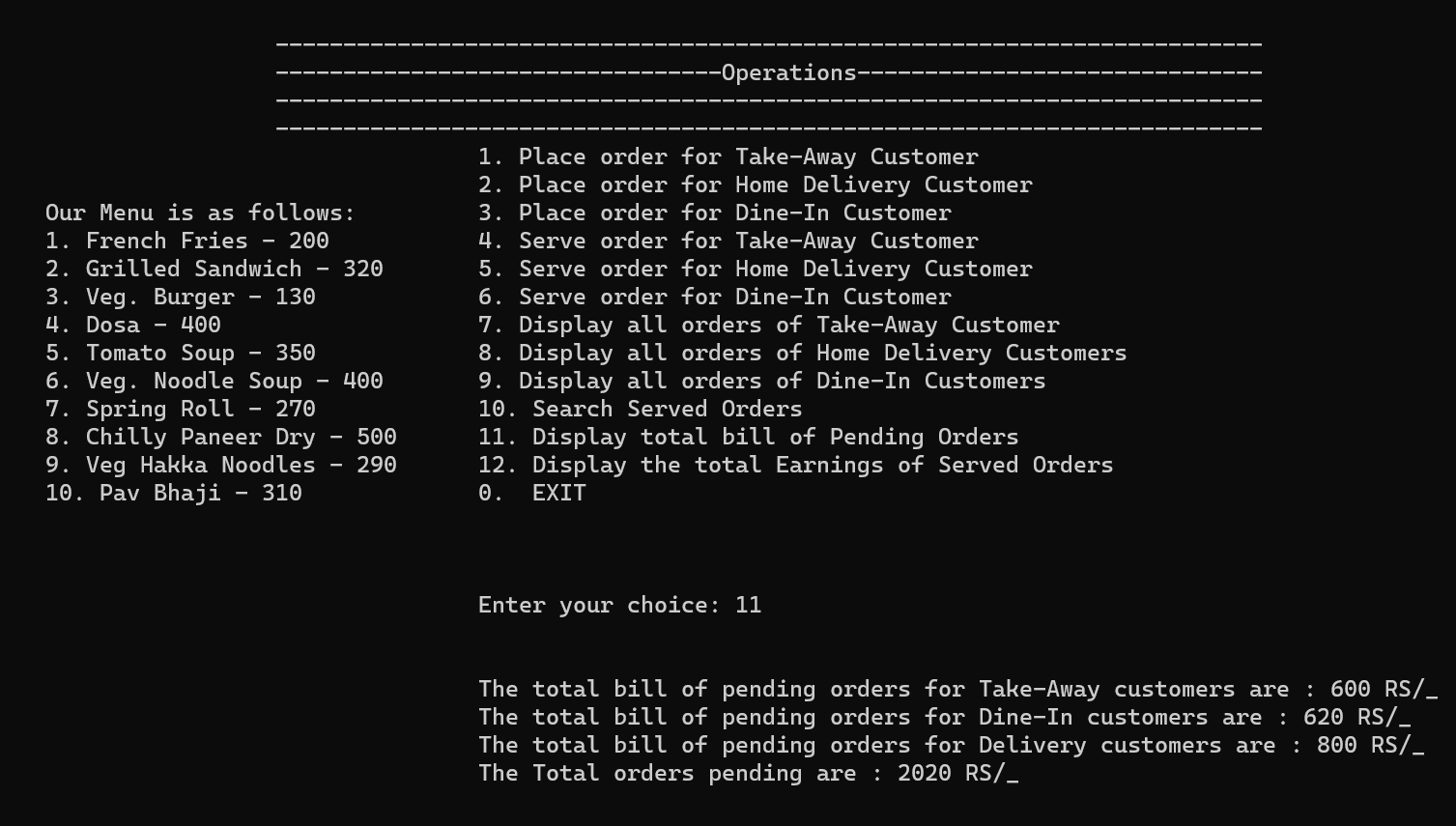
Display all orders of Take-away



Search served orders



Display total bill of orders which are not served



**Conclusion –**

Designing a restaurant management system that efficiently serves walk-in, home delivery, and dine-in customers. The system should include several key features such as order processing, inventory management, customer management, and delivery management system should calculate the shortest path between the restaurant and the customer's location using Dijkstra's algorithm to optimize delivery time. For dine-in customers, the system should implement on a first-come, first-serve basis. Overall, a well-designed restaurant management system that efficiently serves walk-in, home delivery, and dine-in customers with different service priorities can help restaurant owners and managers streamline their operations, improve customer satisfaction, and increase revenue.

**Acknowledgment -**

We have learned a lot and really enjoyed working on this project. We did not have much-advanced knowledge of data structures and algorithms in C++ before starting this project but across this project, we learned how to implement our theoretical knowledge and make a successful project. We literally want to thank all who have helped up throughout this project. The Internet played a main role where we got to know about many new things and we were able to implement that in our project.