EAST WEST UNIVERSITY
excellence in education

Course Code and Name: CSE207 – Data Structures

CSE207 Project

Project name: Shopping club Project using C

Student Name:Noshin Tasnim Khan **Student Id:**2020-1-60-221 **Student Name:**Noushin Pervez **Student Id:**2020-1-60-189 **Student Name:**Sanjida Akter Student Id:2020-1-60-222	**Course Instructor:** Dr. Md. Rezaul Karim Professor, Department of Computer Science and Engineering.

Data structure which we used: Doubly linked list

Advantages of using doubly linked list:

1.Allow us to iterate in both directions

2.We can delete a node easily as we have access to its previous node.

3.Reversing is easy

4.Can grow or shrink in size dynamically.

5.Useful in implementing various other data structures.

We do not use array because

1.It is a consistent set of a fixed number of data items.

2.It has to be specified during declaration.

3.Element location is allocated using compile time.
4. Slow relatively as shifting is required.

Disadvantages of using doubly linked list:

1.Compared to a single linked list, each node store an extra pointer which consumes extra memory.

2.Operations require more time due to the overhead of handling extra pointers as compared to singly-linked lists.

3.No random access of elements.

Though doubly linked list has some disadvantages, We have used doubly linked list because we can easily create list , insert or delete food items from the menu.

Time Complexity of doubly linked list:

There is the best, worst and average case time complexity in doubly linked list. We will discuss time complexity for worse case.

Inserting at beginning or end, time complexity would be O(1).(for constant time)

Deleting at beginning or end, time complexity would be O(1).

For searching and accessing, time complexity O(n). (for linear time)

So,the total complexity would be O(n).

Time complexity of O(1) and O(n) is good because it takes less time to execute. For this, the code would be more efficient. The lesser the time complexity, the faster the execution.