DOUBLY LINKED LIST

A Mini Project Report Submitted by

Adithya Holla K (4NM19CS007)
Aditya Murugan(4NM19CS010)

UNDER THE GUIDANCE OF

MR.PUNEETH RP
ASSISTANT PROFESSOR GD-11
Department of Computer Science and Engineering

in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Engineering in
Computer Science & Engineering

from

Visvesvaraya Technological University, Belagavi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belagavi) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified) NITTE -574 110, Udupi District, KARNATAKA.

(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

TABLE OF CONTENTS :

ABSTRACT AND INTRODUCTION

A Doubly Linked List (DLL) contains an extra pointer, typically called previous pointer, together with next pointer and data which are there in singly linked list. Doubly linked list is a complex type of linked list in which a node contains a pointer to the previous as well as the next node in the sequence. Therefore, in a doubly linked list, a node consists of three parts: node data, pointer to the next node in sequence (next pointer) ,pointer to the previous node (previous pointer). A doubly linked list containing three nodes. Memory Representation of a doubly linked list is shown in the following image. Generally, doubly linked list consumes more space for every node and therefore, causes more expansive basic operations such as insertion and deletion. However, we can easily manipulate the elements of the list since the list maintains pointers in both the directions. Doubly Linked List is a variation of Linked list in which navigation is possible in both ways, either forward and backward easily as compared to Single Linked List. Following are the important terms to understand the concept of doubly linked list.

Some of the OpenGl functions we have used are :

OpenGL Utility Library (GLU): is a computer graphics library for OpenGL. It consists of a number of functions that use the base OpenGL library to provide higher-level drawing routines from the more primitive routines that OpenGL provides. It is usually distributed with the base OpenGL package. When using GLU library you will need to use #include in your program.GLU routines begin with the prefix glu.

OpenGL Utility Toolkit (GLUT) :is a library of utilities for OpenGL programs, which primarily perform system-level I/O with the host operating system. Functions performed include window definition, window control, and monitoring of keyboard and mouse input. Routines for drawing a number of geometric primitives (both in solid and wireframe mode) are also provided, including cubes, spheres and the Utah teapot. GLUT also has some limited support for creating pop-up menus. Here we need to use #include in your program. All GLUT functions start with the prefix glut.

glutKetboardFunc Registers callback handler for keyboard event.

Handling mouse inputs with Glut We can register callback function to handle mouse-click and mouse-motion. glutMouseFunc Registers callback handler for mouse click.

IMPORTANCE

It is used in the navigation systems where front and back navigation is required.

It is used by the browser to implement backward and forward navigation of visited web pages that is a back and forward button. It is also used to represent a classic game deck of cards.

It is also used by various applications to implement undo and redo functionality.

Doubly Linked List is also used in constructing MRU/LRU (Most/least recently used) cache.

Other data structures like stacks, Hash Tables, Binary trees can also be constructed or programmed using a doubly-linked list. Also in many operating systems, the thread scheduler(the thing that chooses what process needs to run at which time) maintains a doubly-linked list of all processes running at that time

OBJECTIVE

Our project has objective mainly where the user can insert node by inserting front or rear . user also can delete the node by using front or rear operations. The user can also specify the node . Our project has a feature to find the maximum and minimum node. There is also feature that it show the maximum number of node

HARDWARE / SOFTWARE REQUIREMENTS

Software Requirements:

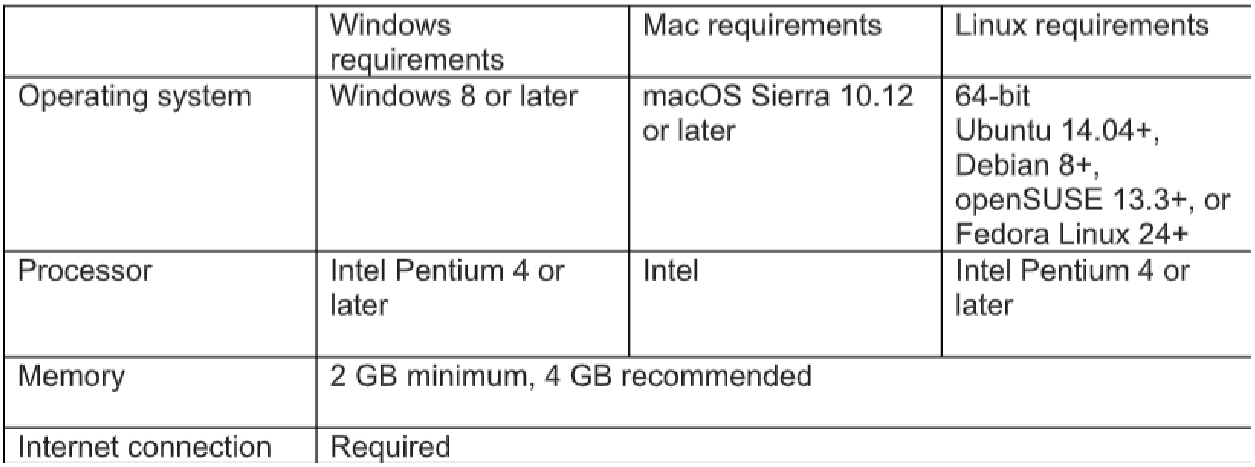
➢ IDE: Code Block

HARDWARE SPECIFICATION:

Processor: Pentium-3 or Higher

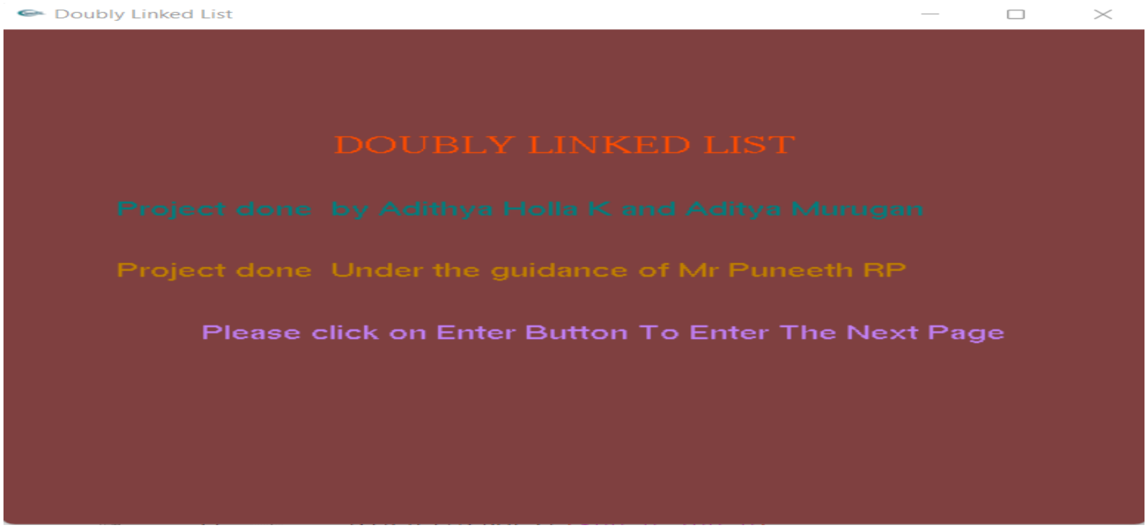
➢ RAM: 64MB or Higher

➢ Hard Disk: 80GB or Higher



	Windows requirements	Mac requirements	Linux requirements
Operating system	Windows 8 or later	macOS Sierra 10.12 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
Processor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Memory	2 GB minimum, 4 GB recommended		
Internet connection	Required		

SCREENSHOTS OF THE PROJECT

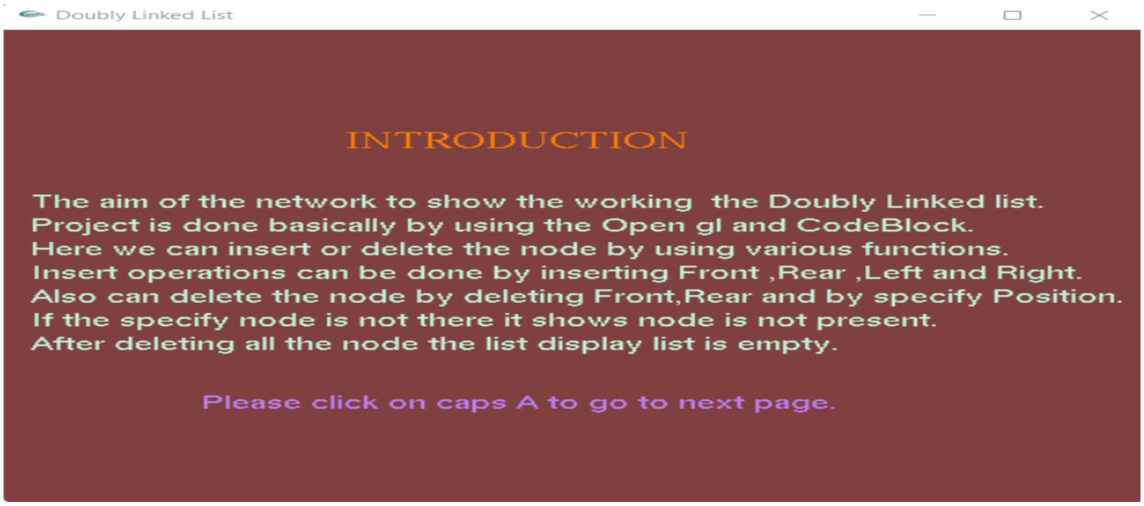


DOUBLY LINKED LIST

Project done by Adithya Holla K and Aditya Murugan

Project done Under the guidance of Mr Puneeth RP

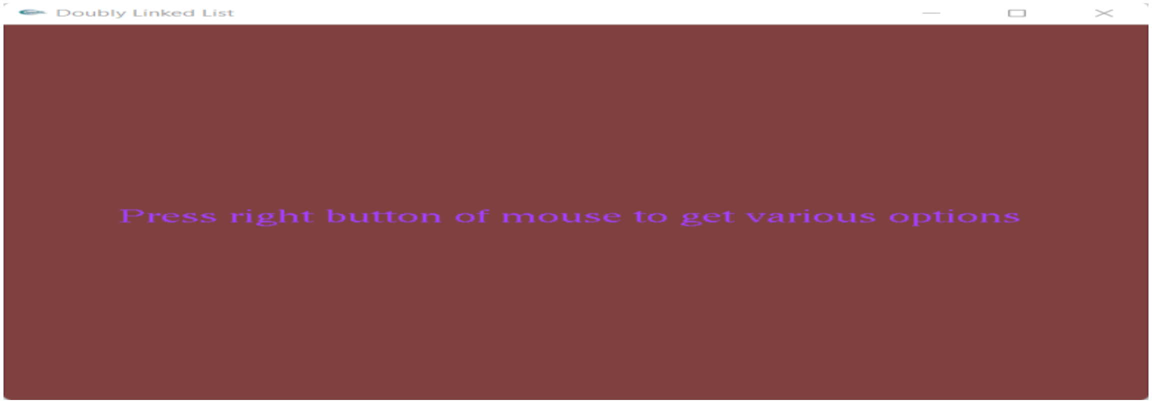
Please click on Enter Button To Enter The Next Page



INTRODUCTION

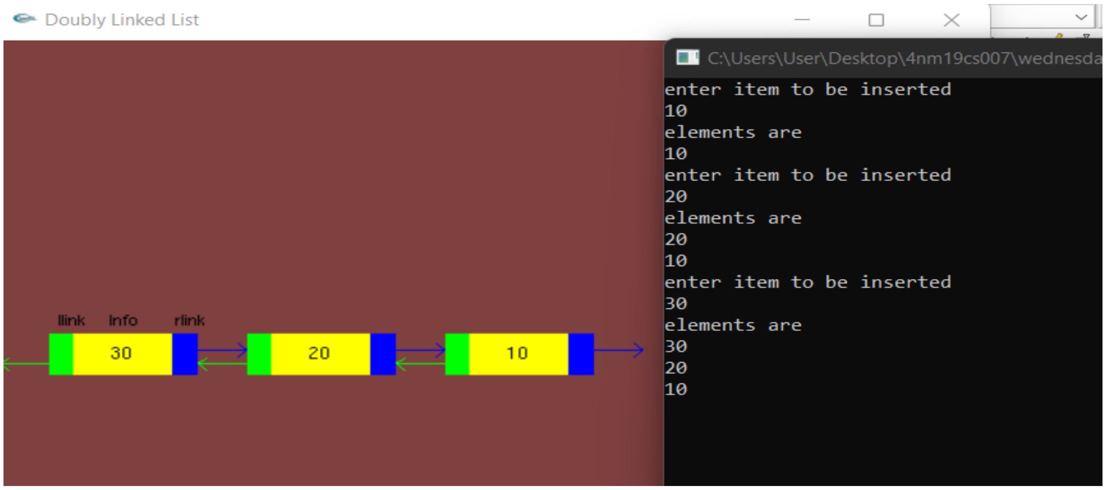
The aim of the network to show the working the Doubly Linked list.
Project is done basically by using the Open gl and CodeBlock.
Here we can insert or delete the node by using various functions.
Insert operations can be done by inserting Front ,Rear ,Left and Right.
Also can delete the node by deleting Front,Rear and by specify Position.
If the specify node is not there it shows node is not present.
After deleting all the node the list display list is empty.

Please click on caps A to go to next page.

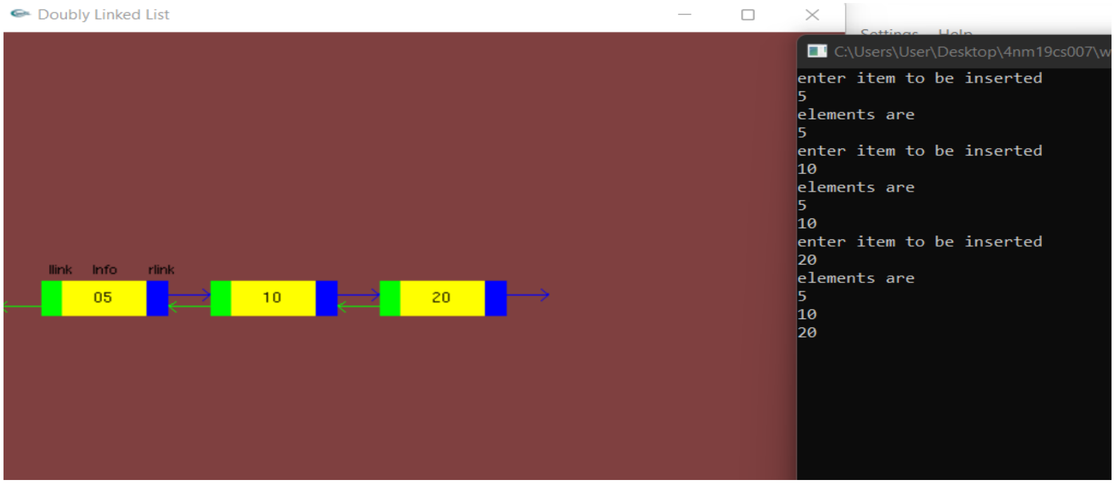


Press right button of mouse to get various options

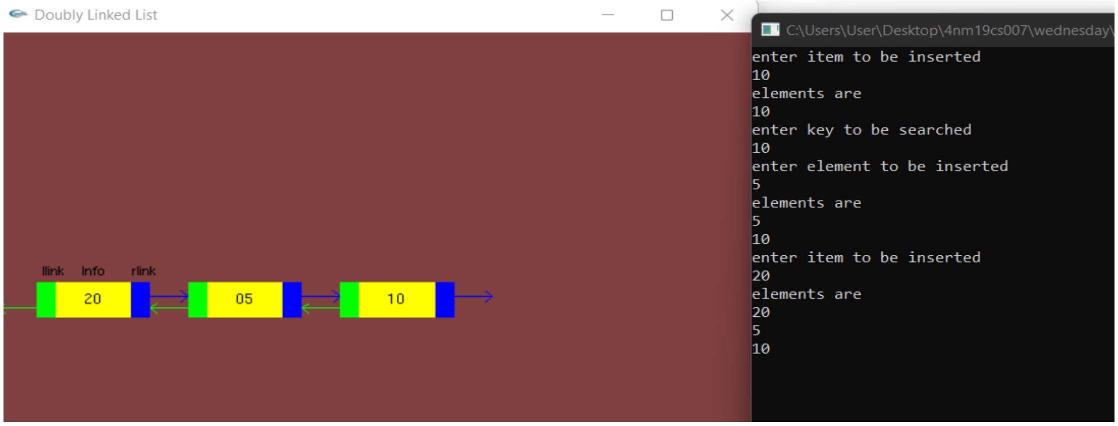
INSERTING NODE AT FRONT , REAR , LEFT , RIGHT



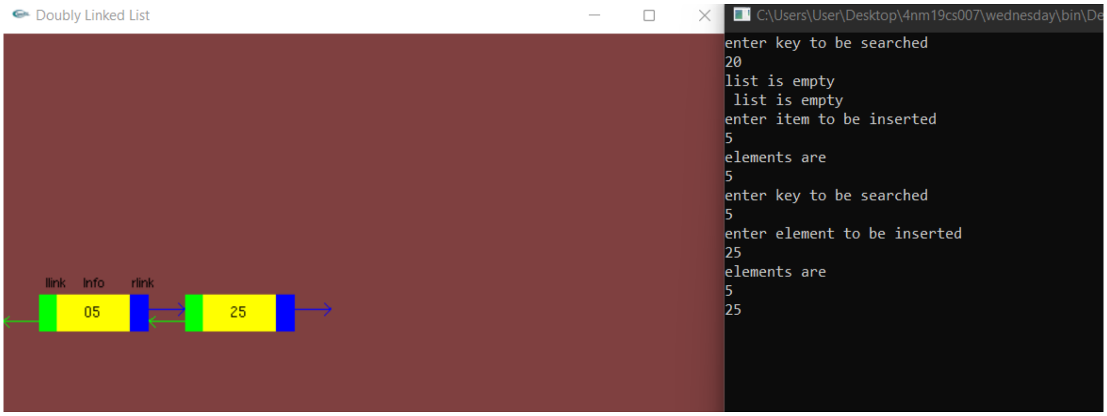
```
enter item to be inserted
10
elements are
10
enter item to be inserted
20
elements are
20
10
enter item to be inserted
30
elements are
30
20
10
```



```
enter item to be inserted
5
elements are
5
enter item to be inserted
10
elements are
5
10
enter item to be inserted
20
elements are
5
10
20
```

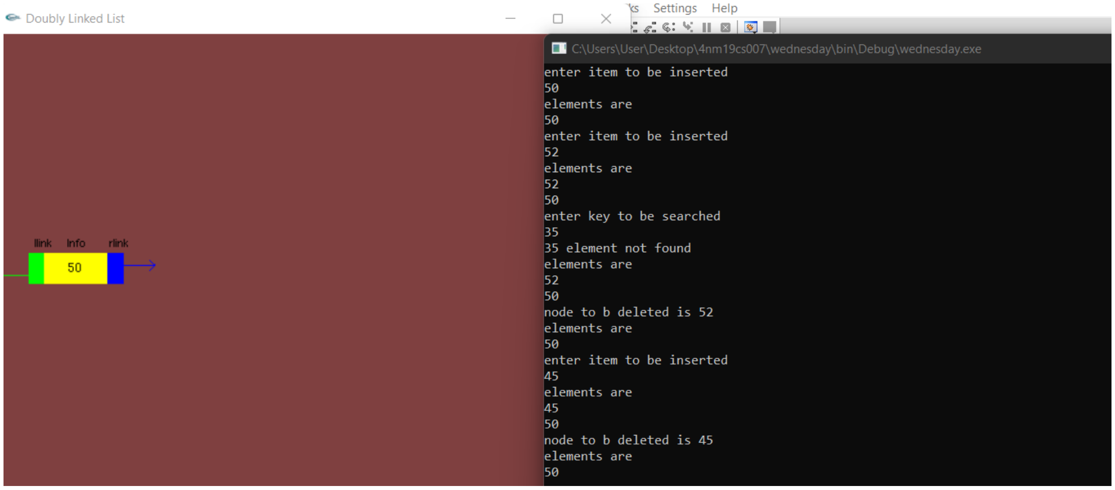


```
enter item to be inserted
10
elements are
10
enter key to be searched
10
enter element to be inserted
5
elements are
5
10
enter item to be inserted
20
elements are
20
5
10
```

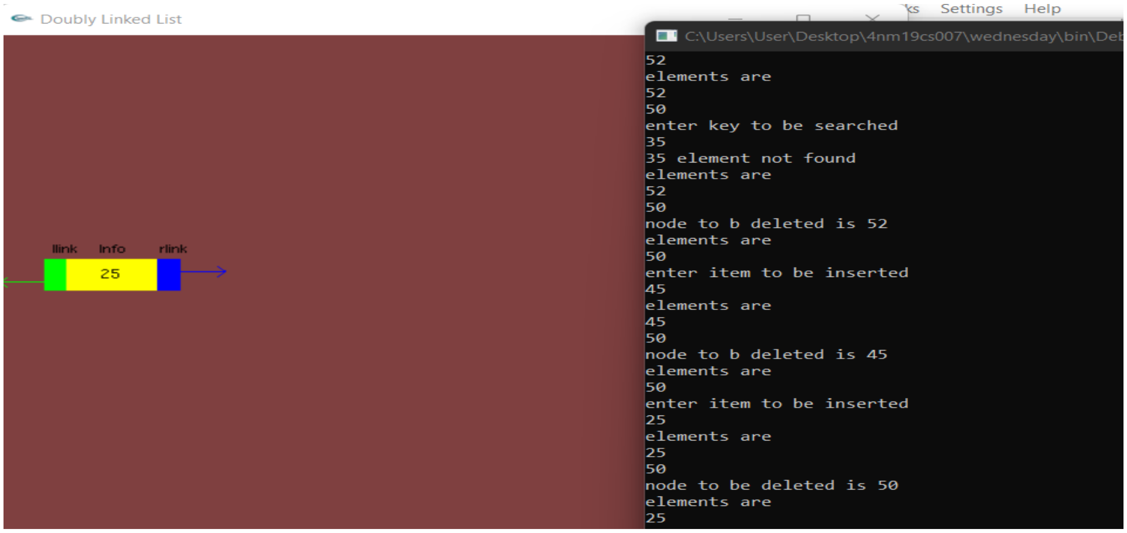


```
enter key to be searched
20
list is empty
 list is empty
enter item to be inserted
5
elements are
5
enter key to be searched
5
enter element to be inserted
25
elements are
5
25
```

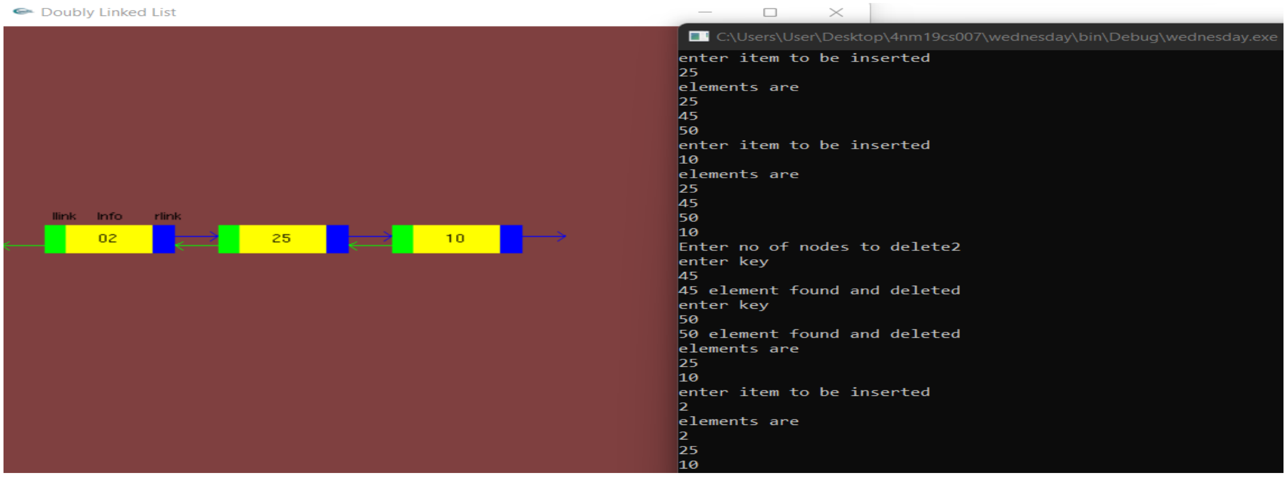
<u>DELETING NODE AT FRONT , REAR , SPECIFY THE NODE</u>:


Doubly Linked List
ks Settings Help
C:\Users\User\Desktop\4nm19cs007\wednesday\bin\Debug\wednesday.exe
llink
info
rlink
50

```
enter item to be inserted
50
elements are
50
enter item to be inserted
52
elements are
52
50
enter key to be searched
35
35 element not found
elements are
52
50
node to b deleted is 52
elements are
50
enter item to be inserted
45
elements are
45
50
node to b deleted is 45
elements are
50
```

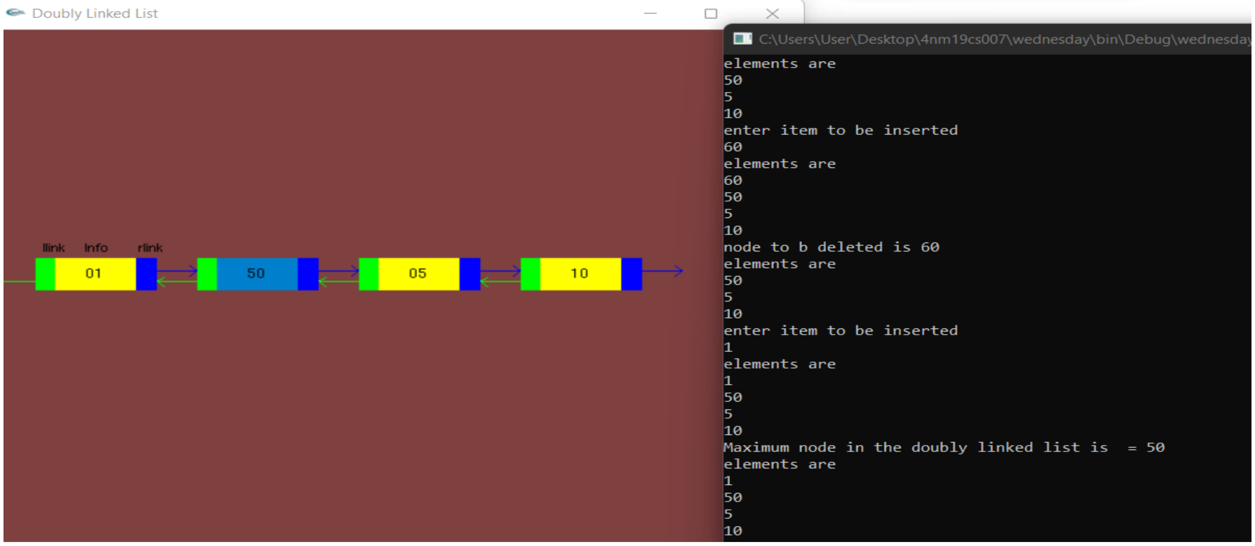


```
52
elements are
52
50
enter key to be searched
35
35 element not found
elements are
52
50
node to b deleted is 52
elements are
50
enter item to be inserted
45
elements are
45
50
node to b deleted is 45
elements are
50
enter item to be inserted
25
elements are
25
50
node to be deleted is 50
elements are
25
```

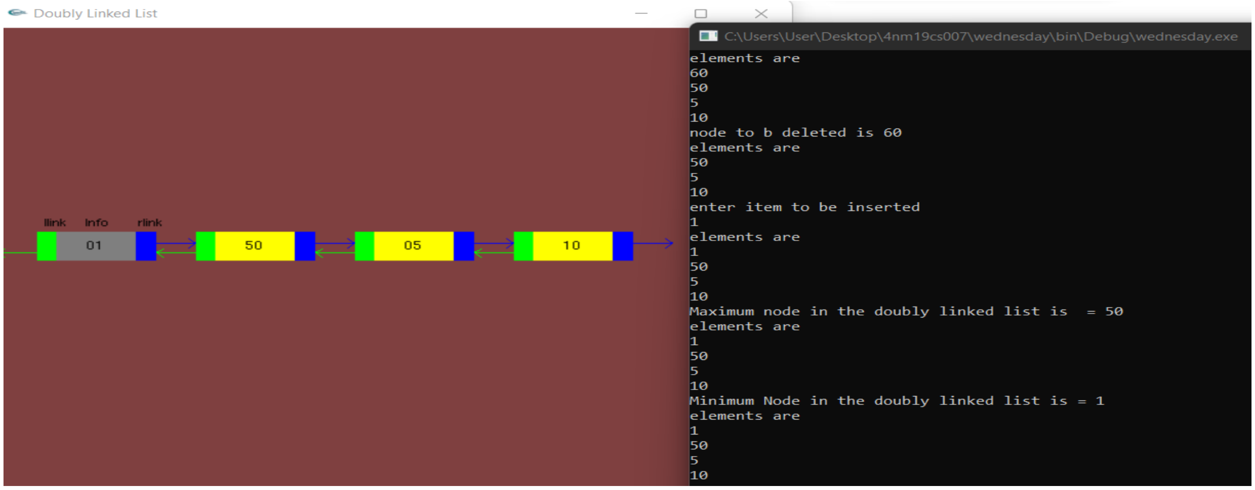

Doubly Linked List
C:\Users\User\Desktop\4nm19cs007\wednesday\bin\Debug\wednesday.exe
llink
info
rlink
02
25
10

```
enter item to be inserted
25
elements are
25
45
50
enter item to be inserted
10
elements are
25
45
50
10
Enter no of nodes to delete2
enter key
45
45 element found and deleted
enter key
50
50 element found and deleted
elements are
25
10
enter item to be inserted
2
elements are
2
25
10
```

MAXIMUM AND MINIMUM NODE:

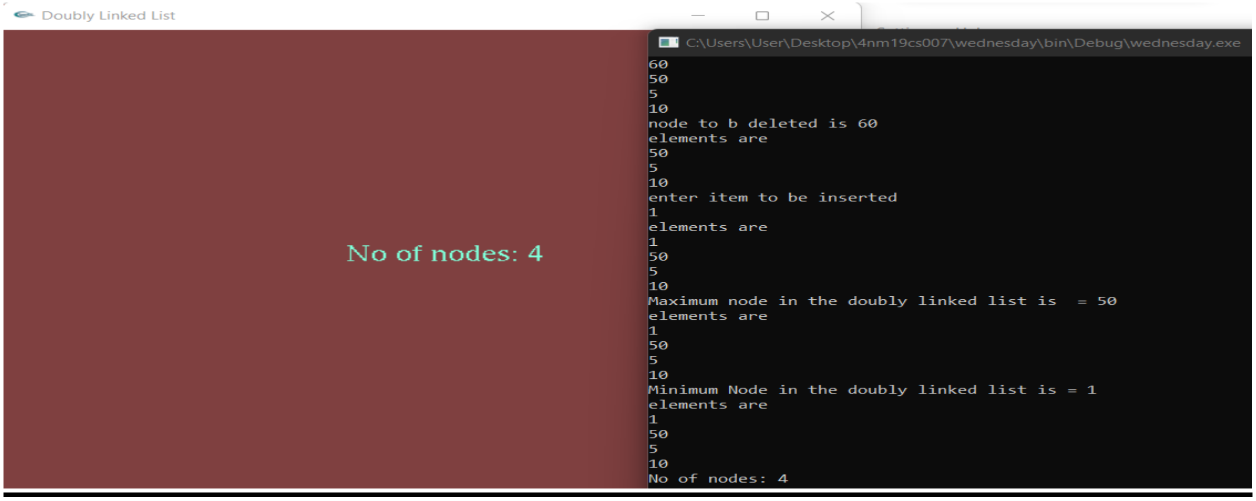

Doubly Linked List
C:\Users\User\Desktop\4nm19cs007\wednesday\bin\Debug\wednesday
llink
info
rlink
01
50
05
10

```
elements are
50
5
10
enter item to be inserted
60
elements are
60
50
5
10
node to b deleted is 60
elements are
50
5
10
enter item to be inserted
1
elements are
1
50
5
10
Maximum node in the doubly linked list is  = 50
elements are
1
50
5
10
```

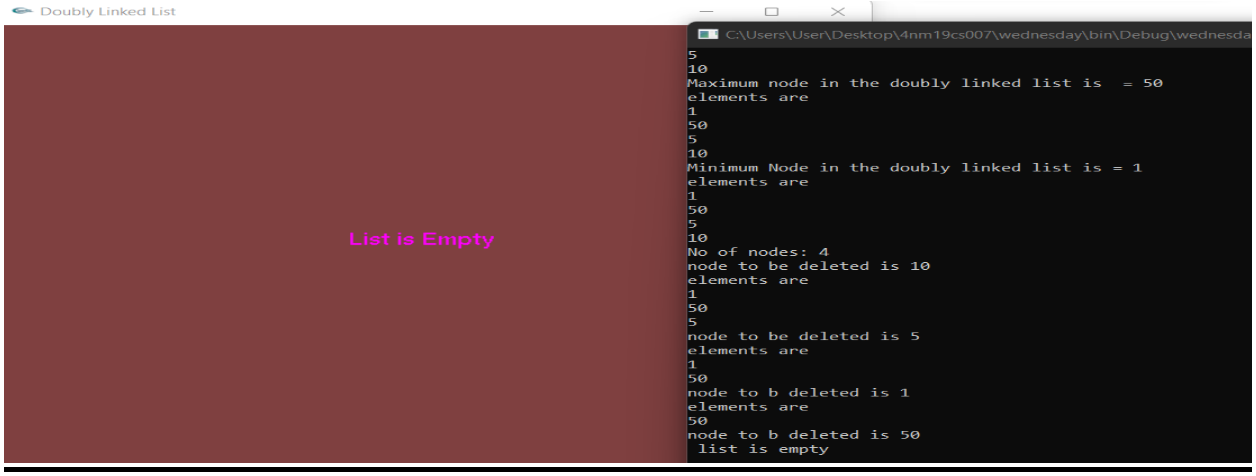

Doubly Linked List
C:\Users\User\Desktop\4nm19cs007\wednesday\bin\Debug\wednesday.exe
llink
info
rlink
01
50
05
10

```
elements are
60
50
5
10
node to b deleted is 60
elements are
50
5
10
enter item to be inserted
1
elements are
1
50
5
10
Maximum node in the doubly linked list is  = 50
elements are
1
50
5
10
Minimum Node in the doubly linked list is = 1
elements are
1
50
5
10
```

TOTAL NO OF NODE AND DISPLAYING LIST IS EMPTY



```
C:\Users\User\Desktop\4nm19cs007\wednesday\bin\Debug\wednesday.exe
60
50
5
10
node to b deleted is 60
elements are
50
5
10
enter item to be inserted
1
elements are
1
50
5
10
Maximum node in the doubly linked list is  = 50
elements are
1
50
5
10
Minimum Node in the doubly linked list is = 1
elements are
1
50
5
10
No of nodes: 4
```



```
C:\Users\User\Desktop\4nm19cs007\wednesday\bin\Debug\wednesda
5
10
Maximum node in the doubly linked list is  = 50
elements are
1
50
5
10
Minimum Node in the doubly linked list is = 1
elements are
1
50
5
10
No of nodes: 4
node to be deleted is 10
elements are
1
50
5
node to be deleted is 5
elements are
1
50
node to b deleted is 1
elements are
50
node to b deleted is 50
 list is empty
```

CONCLUSION :

We successfully implemented this project using OpenGL functionalities and learnt a lot about its implementation and used it thereby satisfying the proper requirements.

FUTURE SCOPE :

- Can be implemented by adding many more operations.

REFERENCES

- “Overview Guides Reference Samples Libraries Support Go to console” Documentation Firebase, https://firebase.google.com/docs/
- Stack Overflow, https://stackoverflow.com
- YouTube, https://www.youtube.com