**Data Structure and Algorithm Practicals**

1. Demonstrate doubly linked list

<!DOCTYPE html>

<html>

<head>

    <title>Doubly Linked List</title>

    <script type="text/javascript">

        class node{

            constructor(){

                this.info=document.getElementById("intvalue").value;

                this.prev=null;

                this.next=null;

            }

        }

        class doublylist{

            constructor()

            {

                this.head=null;

                this.tail=null;

            }

            create()

            {

                var temp = new node();

                if(this.head==null)

                {

                    temp.prev=null;

                    temp.next=null;

                    this.head=temp;

                    this.tail=temp;

                }

                else

                {

                    temp.prev=this.tail;

                    temp.next=null;

                    this.tail.next=temp;

                    this.tail=temp;

                }

                alert("Element Added");

                document.getElementById("intvalue").value="";

            }

            insertfirst()

            {

                var temp = new node();

                if(this.head==null)

                {

                    temp.prev=null;

                    temp.next=null;

                    this.head=temp;

                    this.tail=temp;

                }

                else

                {

                    temp.prev=null;

                    temp.next=this.head;

                    this.head.prev=temp;

                    this.head=temp;

                }

                alert("Element Added At First Position");

                document.getElementById("intvalue").value="";

            }

            insertlast()

            {

                var temp = new node();

                if(this.head==null)

                {

                    temp.prev=null;

                    temp.next=null;

                    this.head=temp;

                    this.tail=temp;

                }

                else

                {

                    temp.prev=this.tail;

                    temp.next=null;

                    this.tail.next=temp;

                    this.tail=temp;

                }

                alert("Element Added At Last Position");

                document.getElementById("intvalue").value="";

            }

            deletefirst()

            {

                if(this.head==null)

                {

                    alert("List Is Empty.");

                }

                else

                {

                    var deletedelement = this.head.info;

                    this.head=this.head.next;

                    alert("Element Deleted From First Position. Deleted Element is "+deletedelement);

                }

            }

            deletelast()

            {

                var deletedelement;

                if(this.head==null)

                {

                    alert("List Is Empty.");

                }

                if(this.head.next==null)

                {

                    deletedelement=this.tail.info;

                    this.head=null;

                    this.tail=null;

                    alert("Element Deleted From First Position. Deleted Element is "+deletedelement);

                }

                else

                {

                    deletedelement=this.tail.info;

                    this.tail.prev.next=null;

                    this.tail=this.tail.prev;

                    alert("Element Deleted From First Position. Deleted Element is "+deletedelement);

                }

            }

            display()

            {

                var curr=this.head;

                if(this.head==null)

                {

                    document.getElementById("printhere").innerHTML="";

                    alert("List is Empty.");

                }

                else

                {

                    var str="Representation of Linked List is <br>Head <--> ";

                    while(curr)

                    {

                        str=str+curr.info+" <--> ";

                        curr=curr.next;

                    }

                    document.getElementById("printhere").innerHTML=str+"Tail";

                }

            }

            reversedisplay()

            {

                var curr=this.tail;

                if(curr!=null)

                {

                    var str="Representation of Linked List is <br>Tail <--> ";

                    while(curr)

                    {

                        str=str+curr.info+" <--> ";

                        curr=curr.prev;

                    }

                    document.getElementById("printhere").innerHTML=str+"Head";

                }

                else

                {

                    document.getElementById("printhere").innerHTML="";

                    alert("List is Empty.");

                }

            }

        }

        var obj =new doublylist();

    </script>

</head>

<body>

    <label>Doubly Linked List</label><br><br>

    <label>Enter Value : </label>

    <input type="text" id="intvalue"><br>

    <button id="btninsert" onclick="obj.insertposition.call(obj)" style="display: none;">Insert</button>

    <button id="btndelete" onclick="obj.deleteposition.call(obj)" style="display: none;">Delete</button><br><br>

    <button onclick="obj.create.call(obj)">Create</button>

    <button onclick="obj.insertfirst.call(obj)">Insert At First</button>

    <button onclick="obj.insertlast.call(obj)">Insert At Last</button>

    <button onclick="obj.deletefirst.call(obj)">Delete At First</button>

    <button onclick="obj.deletelast.call(obj)">Delete At Last</button>

    <button onclick="obj.display.call(obj)">Display</button><br><br>

    <button onclick="obj.reversedisplay.call(obj)">Reverse Display</button><br><br>

    <h3 id="printhere" style="font-family:Comic Sans MS;"></h3>

</body>

</html>