Shikshan Maharshi Dr. D. Y. Patil Sanstha's

Dr D. Y. Patil Centre For Management & Research, Chikhali, Pune

<u>DSA & Java Script Lab Workbook</u>

Shikshan Maharshi Dr. D. Y. Patil Sanstha's

Dr D. Y. Patil Centre For Management & Research, Chikhali, Pune

DSA Lab Workbook

Subject: Data-Structure Algorithm Programming Lab

Course: MCA I - [Sem: I]

Prepared By: -

Name:

Roll No:

Seat No:

Program Index

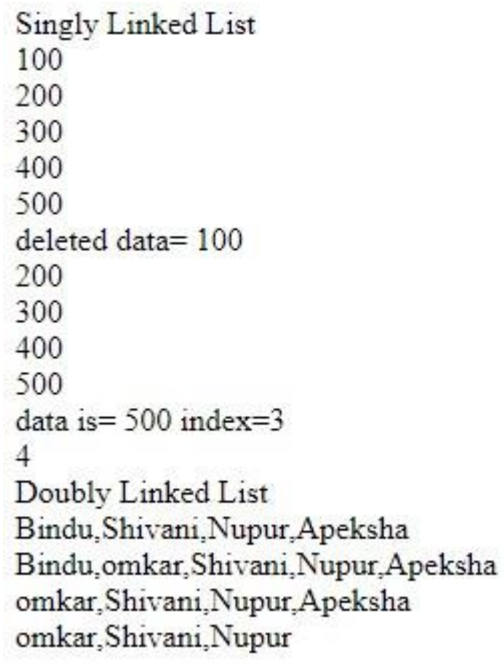
Q1. Demonstrate singly and doubly linked list.

Ans:-

```
<script>
// Edit your script here
class Node {
constructor(data,next=null)
{
 this.data=data;
 this.next=next;
}
}
class Linkedlist
{
  constructor()
  {
    this.head=null;
    this.size=0;
  }
  insertFirst(data)
  {
    this.head=new Node(data,this.head)
    this.size++;
  }
  insertLast(data)
  {
    let node=new Node(data)
    if(!this.head)
    {

      this.head=node;
    }
    else
    {
     let temp=this.head;
      while(temp.next)
      {
        temp=temp.next;
      }
      temp.next=node;
    }
    this.size++;
  }
  insertAtIndex(data,index)
  {
    if(index>0 & index>this.size)
    {
    return;
    }
    if(index===0)
    {
      this.head=new Node(data,this.head)
         return;
    }
   let node=new Node(data)
    let temp1,temp2;

    temp1=this.head;
    let count=0;
    while(count<index)
    {
      temp2=temp1;
      count++;
      temp1=temp1.next;

    }
    node.next=temp1;
    temp2.next=node;
    this.size++;
  }
  getIndex(index)
  {
            var b=false;
    let temp=this.head;
    let count=0;
    while(temp)
    {
      if(count==index)
      {
        document.write("data is= "+temp.data +""+ "
index="+index);
        document.write("</br>");
        b=true;
      }
      count++;
      temp=temp.next;
    }
    if(b==false)
    {
            document.write("null");
    }
  }
  removeat(index)
  {
    let temp1=this.head;
    let temp2;
    let count=0;
    if(index>0 & index>this.size)
    {
      return;
    }

    if(index===0)
    {
      this.head=temp1.next;

    }
    else
    {
      while(count<index)
      {
        count++;
        temp2=temp1;
        temp1=temp1.next;

      }

      temp2.next=temp1.next;
    }
    document.write("deleted data= "+temp1.data);
    document.write("</br>");
    this.size--;
    }
```

```
  clearlist()
  {
    this.head=null;
    this.size=0;
  }
  printdatalist()
  {
    let temp=this.head;
    while(temp)
    {
      document.write(temp.data);
      document.write("</br>");
      temp=temp.next
    }
  }
 }
document.write("Singly Linked List")
document.write("</br>")
const ll=new Linkedlist();
ll.insertFirst(100)
ll.insertAtIndex(200,1)
ll.insertAtIndex(300,2)
ll.insertAtIndex(400,3)
ll.insertLast(500)
ll.printdatalist()
ll.removeat(0)
ll.printdatalist()
ll.getIndex(3)
document.write(ll.size);
document.write("</br>")
document.write("Doubly Linked List")
document.write("</br>")
function doubleLinkedList() {
  let Node = function(element) {
    this.element = element;
    this.next = null;
    this.prev = null;
  }

  let length = 0;
  let head = null;
let tail = null;

this.append = function(element) {
  let node = new Node(element),
        current = head,
        previous;

  if(!head){
        head = node;
        tail = node;
   }else{
     node.prev = tail;
     tail.next = node;
     tail = node;
   }

  length++;
}

this.insert = function(position, element) {

  if(position >= 0 && position <= length){
    let node = new Node(element),
        current = head,
        previous,
        index = 0;

    if(position === 0){
      if(!head){
        head = node;
        tail = node;
      }else{
        node.next = current;
        current.prev = node;
        head = node;
      }
    }else if(position === length){
      current = tail;
      current.next = node;
      node.prev = current;
      tail = node;
    }else{
      while(index++ < position){
        previous = current;
        current = current.next;
      }

      node.next = current;
      previous.next = node;

      current.prev = node;
      node.prev = previous;
    }

    length++;
    return true;
  }else{
    return false;
  }
}

this.removeAt = function(position){
  if(position > -1 && position < length){
    let current = head, previous, index = 0;
          if(position === 0){
      head = current.next;
              if(length === 1){
        tail = null;
      }else{
        head.prev = null;
      }
    }else if(position === length - 1){
      current = tail;
      tail = current.prev;
      tail.next = null;
    }else{
      while(index++ < position){
        previous = current;
        current = current.next;
      }
      previous.next = current.next;
      current.next.prev = previous;
    }

    length--;
    return current.element;
  }else{
    return null;
  }
}
this.indexOf = function(elm){
  let current = head,
  index = -1;
  while(current){
    if(elm === current.element){
       return ++index;
    }

     index++;
     current = current.next;
   }
  return -1;
};

this.isPresent = (elm) => {
  return this.indexOf(elm) !== -1;
};

this.delete = (elm) => {
   return this.removeAt(this.indexOf(elm));
};
this.deleteHead = function(){
  this.removeAt(0);
}
this.deleteTail = function(){
  this.removeAt(length-1);
}
this.toString = function(){
  let current = head,
  string = '';

  while(current){
      string += current.element + (current.next ? '\n'
: '');
      current = current.next;
    }
  return string;
  };
  this.disp = function(){
    let arr = [],
    current = head;
     while(current){
      arr.push(current.element);
      current = current.next;
    }return arr;
  };
  this.isEmpty = function(){
    return length === 0;
  };
  this.size = function(){
    return length;
  }
  this.getHead = function() {
    return head;
  }
  this.getTail = function() {
    return tail;
  }}
let dll = new doubleLinkedList();
dll.append('Bindu');
dll.append('Shivani');
dll.append('Nupur');
dll.append('Apeksha');
document.write(dll.disp());
document.write("</br>");
dll.insert(1, 'omkar');
document.write(dll.disp());
document.write("</br>");
dll.deleteHead();
document.write(dll.disp());
document.write("</br>");
dll.deleteTail();
document.write(dll.disp());
document.write("</br>");
</script>
```

OUTPUT:-

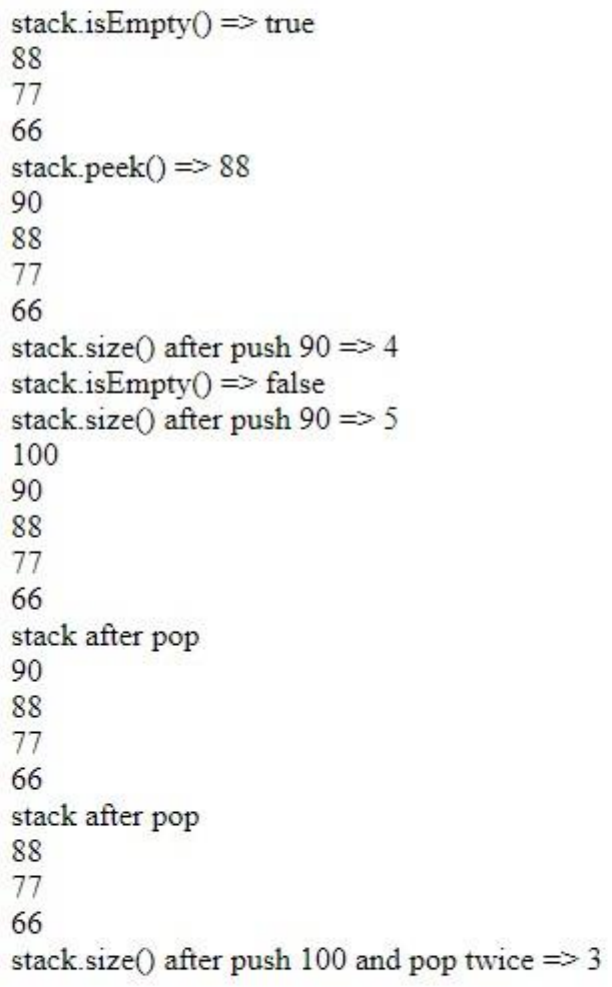


```
Singly Linked List
100
200
300
400
500
deleted data= 100
200
300
400
500
data is= 500 index=3
4
Doubly Linked List
Bindu,Shivani,Nupur,Apeksha
Bindu,omkar,Shivani,Nupur,Apeksha
omkar,Shivani,Nupur,Apeksha
omkar,Shivani,Nupur
```

Q2. STACK implementation using Array with PUSH, POP operations
Ans:-

```
<script>
// Edit your script here class Stack {
constructor() {        this.items = [];
   }

   push(element) {        this.items.push(element);
   }

   pop() {        return this.items.pop();
   }

   peek() {        return this.items[this.items.length - 1];
   }

   isEmpty() {        return this.items.length === 0;     }

   size() {        return this.items.length;
   }

   clear() {        this.items = [];
   }

   toArray() {        return this.items;
   }

   disp(){     for(let i=this.items.length-1;i>=0;i--)
   {
          document.write(this.items[i]);        document.write("</br>");
   }
   }

   toString() {        return this.items.toString();
   }
}
const stack = new Stack(); document.write('stack.isEmpty() => ',
stack.isEmpty());  document.write("<br>"); stack.push(66); stack.push(77);
stack.push(88); stack.disp(); document.write('stack.peek() => ', stack.peek());
document.write("<br>"); stack.push(90); stack.disp();
document.write('stack.size() after push 90 => ', stack.size());
document.write("<br>"); document.write('stack.isEmpty() => ',
stack.isEmpty());  document.write("<br>");
stack.push(100); document.write('stack.size() after push 90 => ', stack.size());
document.write("<br>"); stack.disp(); stack.pop(); document.write("stack after pop </br>");
stack.disp(); stack.pop(); document.write("stack after pop </br>"); stack.disp();
document.write('stack.size() after push 100 and pop twice => ', stack.size());
</script>
<!-- edit your html here -->
```

OUTPUT:-



```
stack.isEmpty() => true
88
77
66
stack.peek() => 88
90
88
77
66
stack.size() after push 90 => 4
stack.isEmpty() => false
stack.size() after push 90 => 5
100
90
88
77
66
stack after pop
90
88
77
66
stack after pop
88
77
66
stack.size() after push 100 and pop twice => 3
```

Q3. Reverse a string using stack .

Ans:-

```
<script> // Edit your script here class Stack
{
            constructor(){
  this.data = [];   this. length = 0;
}
    push(item){      this.length++;      return
this.data.push(item);
    }
    pop (){      if (length <= 0) {        return
null;      } else {        length--;        return
data.pop();
      }
    }
    peek() {      if (length <= 0) {        return null;
} else {        return data[length - 1];
      }
    }

    disp()
    {
            for(let i=this.data.length-1;i>=0;i--)
        {
            document.write(this.data[i]);
        }
    }

    isEmpty(){      return !length;
    }
  }
function reverseString(str) { let result = "";   let
stack = new Stack(); let strArr = str.split("");
strArr.forEach((element) => {
stack.push(element);
  });
stack.disp(); while (!stack.isEmpty()) {
result += stack.pop();
  }
return result;
}
```

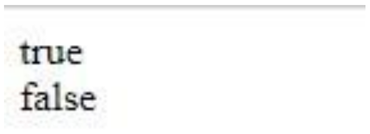
```
let str = "SHIVANI BACHHEWAR"; document.write(reverseString(str));
</script>
<!-- edit your html here -->
```

OUTPUT:- RAWEHHCAB INAVIHS

Q4. Check for balanced parentheses by using Stacks.
Ans:-

```
<script> // Edit your script here isMatchingBrackets =
function(str) {     let stack = [];
   let map = {
      '(': ')',
      '[': ']',
      '{': '}'     }
   for (let i = 0; i < str.length; i++) {

      if (str[i] === '(' || str[i] === '{' || str[i] === '[') {
stack.push(str[i]);
      }               else {
         let last = stack.pop();

         if (str[i] !== map[last]) {
            return false
         };
      }
   }

   if (stack.length !== 0) {        return false
};    return true; }
document.write(isMatchingBrackets("(){}"));  document.write("</br>");
document.write(isMatchingBrackets("({(()))}}"));
document.write("</br>");
</script>
```

OUTPUT:-

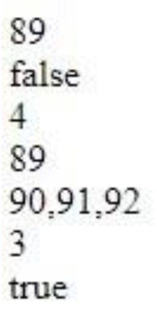


```
true
false
```

Q5. Implement Stack using Linked List.
Ans:-

```
<script>
// Edit your script here function stackUsingLL(){
let Node = function(elm){     this.element = elm;
this.next = null;
  }

  let length = 0;

  let head = null;


  this.push = function(elm){     let node = new
Node(elm),
    current;

    current = head;     node.next = current;     head
= node;

    length++;
  }

  this.pop = function(){     let current = head;
if(current){       let elm = current.element;
current = current.next;       head = current;
length--;       return elm;
    }

    return null;
  }

  this.peek = function(){         if(head){
return head.element;
    }

    return null;
  }

  this.toArray = function(){     let arr = [];     let
current = head;     while(current){
arr.push(current.element);       current =
current.next;
    }
    return arr;   }

  this.isEmpty = function(){
    return length === 0;
  }

  this.size = function(){     return length;
  }

  this.clear = function(){     head = null;
length = 0;
  }

} let stack = new stackUsingLL();     stack.push(92);
stack.push(91);  stack.push(90);  stack.push(89);
document.write(stack.peek());
document.write("<br>");
document.write(stack.isEmpty());
document.write("<br>");
document.write(stack.size());
document.write("<br>");
document.write(stack.pop());
document.write("<br>");
document.write(stack.toArray());
document.write("<br>");
document.write(stack.size());
document.write("<br>");  stack.clear();  //clear the
stack  document.write(stack.isEmpty());
document.write("<br>");
</script>
```

OUTPUT:-

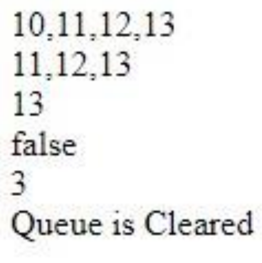


```
89
false
4
89
90,91,92
3
true
```

Q6. Demonstration of Linear Queue.
Ans:-

```
<script> // Edit your script here
class Queue {     constructor() {
this.items = [];
   }

   enqueue(element) {
      return this.items.push(element);
   }

   dequeue() {         if(this.items.length > 0) {
return this.items.shift();
      }
   }

   peek() {
      return this.items[this.items.length - 1];
   }
   isEmpty(){
     return this.items.length == 0;
   }     size(){
      return this.items.length;
   }     clear(){         this.items = [];         var
str="Queue is Cleared";
      return str
   }
} let queue = new Queue(); queue.enqueue(10);
queue.enqueue(11); queue.enqueue(12);
queue.enqueue(13); document.write(queue.items);
document.write("</br>"); queue.dequeue();
document.write(queue.items);
document.write("</br>");
document.write(queue.peek());
document.write("</br>");
document.write(queue.isEmpty());
document.write("</br>");
document.write(queue.size());
document.write("</br>"); var c=queue.clear();
document.write(c); document.write("</br>");
</script>
```

OUTPUT:-

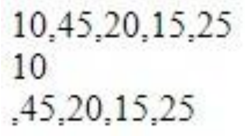


```
10,11,12,13
11,12,13
13
false
3
Queue is Cleared
```

Q7. Demonstration of Circular Queue.
Ans:-

```
 <script>
// Edit your script here class CircularQueue
{  constructor(size) {   this.element = [];
this.size = size   this.length = 0   this.front =
0   this.back = -1
 }   print() {      document.write(this.element);
   }
  isEmpty() {
  return (this.length == 0)
 }
enqueue(element) {
  if (this.length >= this.size) throw (new Error("Maximum length exceeded"))   this.back++
   this.element[this.back % this.size] = element   this.length++
 }
dequeue() {
  if (this.isEmpty()) throw (new Error("No elements in the queue"))   const value =
this.getFront()   this.element[this.front % this.size] = null   this.front++   this.length--
document.write(value);
 } getFront() {
  if (this.isEmpty()) throw (new Error("No elements in the queue"))
  return this.element[this.front % this.size]
 } clear() {
  this.element = new Array()   this.length = 0
this.back = 0   this.front = -1
 } }
let cq=new CircularQueue(5); cq.enqueue(10);
cq.enqueue(45); cq.enqueue(20); cq.enqueue(15);
cq.enqueue(25);
cq.print(); document.write("</br>"); cq.dequeue();
document.write("</br>");
cq.print();
</script>
```

OUTPUT:-



```
10,45,20,15,25
10
,45,20,15,25
```

Q8. Program 8:- Demonstration of Priority Queue.

```
<script>
// Edit your script here class QElement {
constructor(element, priority)     {
      this.element = element;         this.priority = priority;
   }
}

class PriorityQueue {
   constructor()
   {
      this.items = [];
   }
   enqueue(element, priority)
{
   var qElement = new QElement(element, priority);
   var contain = false;

   for (var i = 0; i < this.items.length; i++) {
      if (this.items[i].priority > qElement.priority) {

         this.items.splice(i, 0, qElement);
         contain = true;             break;
      }
   }

   if (!contain) {         this.items.push(qElement);
   }
} dequeue()
{

   if (this.isEmpty())         return "Underflow";
return this.items.shift();
} front() {     if (this.isEmpty())         return "No
elements in Queue";     return this.items[0];
} rear() {
   if (this.isEmpty())         return "No elements in Queue";
return this.items[this.items.length - 1];
} isEmpty()
{

   return this.items.length == 0;
} printPQueue() {     var str = "";     for (var i = 0; i <
this.items.length; i++)         str += this.items[i].element + " ";
   return str;
} } var priorityQueue = new PriorityQueue();
document.write(priorityQueue.isEmpty()); document.write("</br>");
document.write(priorityQueue.front()); document.write("</br>");
priorityQueue.enqueue("Shivani", 2);
priorityQueue.enqueue("Apeskha", 1);
priorityQueue.enqueue("Bindu", 1); priorityQueue.enqueue("Siddhi",
2); priorityQueue.enqueue("Aditi", 3);
document.write(priorityQueue.printPQueue());
document.write("</br>");
document.write(priorityQueue.front().element);
document.write(priorityQueue.rear().element);
document.write("</br>");
document.write(priorityQueue.dequeue().element);
document.write("</br>");   priorityQueue.enqueue("Shweta", 2);
document.write(priorityQueue.printPQueue());
document.write("</br>");
</script>
```

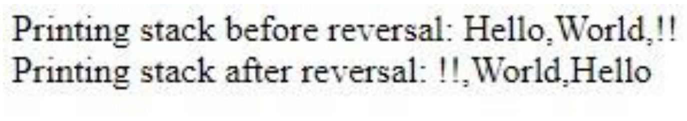
OUTPUT:- true

```
No elements in Queue
Apeskha Bindu Shivani Siddhi Aditi
ApeskhaAditi
Apeskha
Bindu Shivani Siddhi Shweta Aditi
```

Q9. Reverse stack using queue
Ans:-
```
<script> // Edit your script here class Stack
{     constructor(){          this.elements =
[];
   }

   push(element){          this.elements.push(element);
   }

   pop(){
      if(this.isEmpty()) return "Underflow situation";          else return
this.elements.pop();
   }

   isEmpty(){
      return this.elements.length == 0;
   }

   print(){
      return this.elements;
   }

} class Queue {     constructor(){
this.elements = [];
   }

   enqueue(element){
      this.elements.push(element)
   }

   dequeue() {        if(!this.isEmpty()) {              return
this.elements.shift();
      } else {
          return 'Underflow situation';
      }
   }

   isEmpty() {
      return this.elements.length == 0;
   }
} function reverse(stack){     const queue = new
Queue();     while(!stack.isEmpty()){
queue.enqueue(stack.pop());    }
```

```
    while(!queue.isEmpty()){          stack.push(queue.dequeue());    }
}

const stack = new Stack();

stack.push('Hello'); stack.push('World');
stack.push('!!');

document.write('Printing stack before reversal: ', stack.print());
document.write("</br>");

reverse(stack);

 document.write('Printing stack after reversal: ', stack.print());
document.write("</br>");
</script>
```

OUTPUT:-



```
Printing stack before reversal: Hello,World,!!
Printing stack after reversal: !!,World,Hello
```

Q10. Practical based on binary search tree implementation with its operations.
Ans:-

```
<script>
// Edit your script here class Node
{
   constructor(data)
   {        this.data = data;         this.left =
null;         this.right = null;
   }
}
class BinarySearchTree
{    constructor()
   {
      this.root = null;
   }
   insert(data) {
   var newNode = new Node(data);     if(this.root ===
null)         this.root = newNode;
   else
      this.insertNode(this.root, newNode);
}
insertNode(node, newNode)
{
   if(newNode.data < node.data)
   {        if(node.left === null)
         node.left = newNode;         else
         this.insertNode(node.left, newNode);
   }    else
   {        if(node.right === null)            node.right
= newNode;         else
         this.insertNode(node.right,newNode);
   }
} remove(data) {
   this.root = this.removeNode(this.root, data);
}

removeNode(node, key)
{    if(node === null)         return null;
   else if(key < node.data)
   {
      node.left = this.removeNode(node.left, key);
      return node;
   }
   else if(key > node.data)
   {
      node.right = this.removeNode(node.right, key);
      return node;
   }     else
   {
      if(node.left === null && node.right === null)
      {             node = null;
return node;
      }

      if(node.left === null)
      {
         node = node.right;             return node;
      }
      else if(node.right === null)
      {             node = node.left;
return node;
      }

      var aux = this.findMinNode(node.right);
      node.data = aux.data;

      node.right = this.removeNode(node.right, aux.data);
      return node;
   }

} inorder(node) {
   if(node !== null)
   {        this.inorder(node.left);
document.write(node.data);        document.write("
");        this.inorder(node.right);
   }
} preorder(node) {     if(node !== null)
   {
      document.write(node.data);        document.write(" ");
this.preorder(node.left);
      this.preorder(node.right);
   }
} postorder(node) {     if(node !== null)
   {        this.postorder(node.left);
this.postorder(node.right);
document.write(node.data);
      document.write(" ");
   }
```

```
}
findMinNode(node)
{     if(node.left === null)          return node;
    else
        return this.findMinNode(node.left);
} getRootNode()
{     return this.root; } search(node,
data) {     if(node === null)          return
null;     else if(data < node.data)
return this.search(node.left, data);

    else if(data > node.data)
        return this.search(node.right, data);
      else          return node;
}
}
var BST = new BinarySearchTree();

BST.insert(10);
BST.insert(95);
BST.insert(30);
BST.insert(70);
BST.insert(2);
BST.insert(34);
BST.insert(7);
BST.insert(45);
BST.insert(11);
BST.insert(56);

var root = BST.getRootNode();
BST.inorder(root);
BST.remove(5);

var root = BST.getRootNode();
       document.write("</br>");

BST.inorder(root);



BST.remove(7);

var root = BST.getRootNode();
  document.write("</br>");
```

```
BST.inorder(root);

BST.remove(34);

var root = BST.getRootNode(); document.write("</br>");
document.write("inorder traversal"); document.write("</br>");
BST.inorder(root); document.write("</br>");
document.write("postorder traversal"); document.write("</br>");
BST.postorder(root); document.write("</br>"); document.write("preorder
traversal"); document.write("</br>"); BST.preorder(root);
document.write("</br>"); </script>
<!-- edit your html here -->
```

OUTPUT:-

```
2 7 10 11 30 34 45 56 70 95
2 7 10 11 30 34 45 56 70 95 2 10 11 30 34 45
56 70 95 inorder traversal 2 10 11 30 45 56
70 95 postorder traversal 2 11 56 45 70 30
95 10 preorder traversal 10 2 95 30 11 70 45
56
```

Q11. Graph implementation and graph traversals.
Ans:-

```
<script> // Edit your script here class Graph {
constructor() {     this.AdjList = new Map();
console.log('Di-graph');
  }

  addVertex(vertex) {     if (!this.AdjList.has(vertex))
{       this.AdjList.set(vertex, []);
    } else {       throw 'Already Exist!!!'; }
  }

  addEdge(vertex, node) {     if (this.AdjList.has(vertex))
{       if (this.AdjList.has(node)){          let arr =
this.AdjList.get(vertex);         if(!arr.includes(node)){
arr.push(node);         }else{
          throw `Can't add '${node}', it already exists`;
        }
      }else {
        throw `Can't add non-existing vertex ->'${node}'`;
      }
    } else {
      throw `You should add '${vertex}' first`;
    }
  }

  print() {
    document.write(this.AdjList);
document.write("</br>")      for (let [key, value] of
this.AdjList) {       document.write(key, value);
      document.write("</br>")
    }
  }

  createVisitedObject(){     let arr = {};
    for(let key of this.AdjList.keys()){
      arr[key] = false;
    }

    return arr;
  }
  bfs(startingNode){     document.write('\nBFS')
document.write("</br>")      let visited =
this.createVisitedObject();
```

```
    let q = [];     visited[startingNode] = true;
    q.push(startingNode);

    while(q.length){       let current = q.pop()
      document.write(current);

      let arr = this.AdjList.get(current);

      for(let elem of arr){         if(!visited[elem]){
visited[elem] = true;           q.unshift(elem)
        }
      }
          }
  }

  dfs(startingNode){     document.write('\nDFS')
document.write("</br>");     let visited =
this.createVisitedObject();
    this.dfsHelper(startingNode, visited);
  }

  dfsHelper(startingNode, visited){     visited[startingNode] = true;
document.write(startingNode);


    let arr = this.AdjList.get(startingNode);

    for(let elem of arr){       if(!visited[elem]){
        this.dfsHelper(elem, visited);
      }
    }

  }


  doesPathExist(firstNode, secondNode){
     let path = [];     let visited = this.createVisitedObject(); let
q = []; visited[firstNode] = true;
    q.push(firstNode);     while(q.length){       let node =
q.pop();       path.push(node);       let elements =
this.AdjList.get(node);
if(elements.includes(secondNode)){
document.write(path.join('->'))
```

```
        return true;       }else{         for(let elem of
elements){          if(!visited[elem]){
visited[elem] = true;             q.unshift(elem);
          }
        }
      }     }     return false;
  } } function test (arg1, arg2){   if(arg1 ===
arg2){
    document.write(`${arg1} = ${arg2} \t-> passing`)
document.write("</br>");
  }else{     throw 'Not passing';
  } } let g = new Graph();
let arr = ['A', 'B', 'C', 'D', 'E', 'F']; for (let i = 0; i <
arr.length; i++) {    g.addVertex(arr[i]);
}
g.addEdge('A', 'C');
g.addEdge('A', 'B');
g.addEdge('B', 'E');
g.addEdge('B', 'A');
g.addEdge('D', 'E');
g.addEdge('E', 'F');
g.addEdge('E', 'C');
g.addEdge('C', 'F');
g.print();
g.bfs('A'); document.write("</br>"); g.dfs('A');
document.write("</br>"); </script>
<!-- edit your html here -->
```

OUTPUT:- [object Map]

```
AC,B
BE,A
CF
DE
EF,C
F
BFS
ACBFE DFS ACFBE
```

Q12. Implementation of Hashing.

Ans:- <script>

```
// Edit your script here class HashTable {
constructor()  {            this.values  =  {};
this.length =  0;     this.size =  0;
  }
calculateHash(key) {
    return key.toString().length % this.size;
  }
add(key, value) {     const hash = this.calculateHash(key);
     if(!this.values.hasOwnProperty(hash))
    {
      this.values[hash] = {};
    }
    if (!this.values[hash].hasOwnProperty(key)) {        this.length++;
    }
    this.values[hash][key] = value;
  }
search(key) {
     const hash = this.calculateHash(key);
     if (this.values.hasOwnProperty(hash) && this.values[hash].hasOwnProperty(key)) {
document.write(key+" received :")        return this.values[hash][key];
     } else {        return null;
     }}}
const ht = new HashTable(); ht.add("Shivani", "300");
ht.add("Siddhi", "100"); ht.add("Bindu", "50");
document.write(ht.search("Bindu"));
</script>
```

OUTPUT:-

```
Shivani received :300
```

Q13. Implementation of Brute Force technique Linear Search.

Ans:-

```
<script>
function linearSearch(arr, key){
            var b=false;
    for(let i = 0; i < arr.length; i++){         if(arr[i] === key){
         document.write(key+" found at: ")
document.write((i+1)+" position");              b=true;}
    }     if(b==false){
            document.write("Key Not Found");}
}
arr=[7,12,9,5,10] key=5
linearSearch(arr,key)
</script>
```

OUTPUT:-

```
5 found at: 4 position
```

Q14. Implementation of Brute Force technique Rain Terraces.

Ans:-

```
<script>
// Edit your script here function maxWater(arr, n)
   {        let res = 0;
       for(let i = 1; i < n - 1; i++)
       {
          let left = arr[i];             for(let j = 0; j < i;
j++)
          {
             left = Math.max(left, arr[j]);
          }
          let right = arr[i];
          for(let j = i + 1; j < n; j++)
          {
             right = Math.max(right, arr[j]);
          }
          res += Math.min(left, right) - arr[i];
       }
       document.write("Maximum water value will be: ")
       document.write(res);
   }     let arr = [ 1, 0, 4, 2, 1,                  1, 3,
0, 1, 2, 1 ];     let n = arr.length;
maxWater(arr,n);
    </script>
```

OUTPUT:-

```
Maximum water value will be: 9
```

Q15. Implementation of Brute Force technique Travelling Salesman Problem.

Ans:-

```
<script>
function generatePermutations(Arr){
  var permutations = [];   var A = Arr.slice();
function swap(a,b){
    var tmp = A[a];
    A[a] = A[b];
    A[b] = tmp;
  }
function generate(n, A){     if (n == 1){
      permutations.push(A.slice());
    } else {       for(var i = 0; i <= n-1; i++) {
generate(n-1, A);         swap(n % 2 == 0 ? i : 0 ,n-1);
      }
    }
  }
  generate(A.length,  A);    document.write("
")   return permutations;
}function generateCityRoutes(cities){   var pems =
generatePermutations(cities.slice(1));   for (var i = 0; i < pems.length;
i++){     pems[i].unshift(cities[0]);     pems[i].push(cities[0]);
  }
return pems;   document.write("</br>");
}document.write("Permutations are: ")
document.write(generatePermutations([2,1,3]));
document.write("</br>"); document.write("City routes are: ")
document.write(generateCityRoutes([0,2,0]));
document.write("</br>");
</script>
```

OUTPUT:-

```
Permutations are: 2,1,3,1,2,3,3,2,1,2,3,1,1,3,2,3,1,2
City routes are: 0,2,0,0,0,0,2,0
```

Q16. Implementation of Greedy Algorithm-Prim's.
Ans:-

```
<script>
// Edit your script here function createAdjMatrix(V,
G) {   var adjMatrix = [];
  for (var i = 0; i < V; i++) {      adjMatrix.push([]);
    for (var j = 0; j < V; j++) { adjMatrix[i].push(0); }
  }
  for (var i = 0; i < G.length; i++) {      adjMatrix[G[i][0]][G[i][1]] =
G[i][2];     adjMatrix[G[i][1]][G[i][0]] = G[i][2];
  }
  return adjMatrix;
}
function prims(V, G) {
  var adjMatrix = createAdjMatrix(V, G);
  var vertex = 0;   var MST = [];   var
edges = [];   var visited = [];
  var minEdge = [null,null,Infinity];

  while (MST.length !== V-1) {

    visited.push(vertex);

    for (var r = 0; r < V; r++) {       if
(adjMatrix[vertex][r] !== 0) {
        edges.push([vertex,r,adjMatrix[vertex][r]]);
      }
    }
    for (var e = 0; e < edges.length; e++) {
      if (edges[e][2] < minEdge[2] && visited.indexOf(edges[e][1]) === -1) {          minEdge = edges[e];
      }
    }
    edges.splice(edges.indexOf(minEdge), 1);
MST.push(minEdge+"</br>")     vertex = minEdge[1];
minEdge = [null,null,Infinity];
  }
document.write(MST);
  }
var a = 0, b = 1, c = 2, d = 3, e = 4, f = 5;
var graph = [
  [a,b,2],
  [a,c,3],
  [b,d,3],
  [b,c,5],
  [b,e,4],
  [c,e,4],
  [d,e,2],
  [d,f,3],
  [e,f,5]

];
prims(6, graph);
</script>
```

OUTPUT:-

```
0,1,2
,0,2,3
,1,3,3
,3,4,2
,3,5,3
```

Q17. Implementation of Greedy Kruskal's algorithm.

Ans:-

```
<script>
// Edit your script here var kruskal =
function(points) {     const n = points.length;
const dist = [];
   const graph = [...Array(n)].map((_, i) => i);
   let totalDist = 0;

   for(let i = 0; i < n; i++) {         for(let j = i+1; j < n;
j++) {            const [x1, y1] = points[i];            const
[x2, y2] = points[j];
         const distance = Math.abs(x1 - x2) + Math.abs(y1 - y2);
         dist.push([distance, i, j]);
      }
   }

   dist.sort((a, b) => a[0] - b[0]);

   function find(id) {         if(graph[id] === id) return
id;         graph[id] = find(graph[id]);         return
graph[id];
   }

   for(let [d, u, v] of dist) {         const rootU = find(u);
const rootV = find(v);         if(rootU === rootV)
continue;         graph[rootV] = rootU         totalDist +=
d;
   }
   document.write("Minimum Cost is: ")     document.write(totalDist)
};
var points = [[6,5],[12,5],[3,-3]] kruskal(points)
</script>
```

OUTPUT:-

```
Minimum Cost is: 17
```

Q18. Implementation of Divide and Conquer Technique-Binary Search.
Ans:-

```
<script>
// Edit your script here
let recursiveFunction = function (arr, x, start, end) {

   if (start > end) return false;


   let mid=Math.floor((start + end)/2);


   if (arr[mid]===x)
   {
         document.write("Element found at: "+(mid+1))
      document.write("</br>");          return true;
   }
   if(arr[mid] > x)
      return recursiveFunction(arr, x, start, mid-1);     else
      return recursiveFunction(arr, x, mid+1, end);
}
let arr = [1, 10, 5, 6, 8, 9]; let x = 6;
  if (recursiveFunction(arr, x, 0, arr.length-1))     document.write();
else document.write("Element not found!<br>");    x = 10;
if (recursiveFunction(arr, x, 0, arr.length-1))     document.write();
else document.write("Element not found!<br>");
</script>
```

OUTPUT:-

```
Element found at: 4 Element not found!
```

Q19. Implementation of Divide and Conquer Technique-Tower of Hanoi.

Ans:-

```
<script>
function towerOfHanoi(n, from_rod,  to_rod,  aux_rod)
{
        if (n == 1)
        {
            document.write("Move disk 1 from rod " + from_rod + " to rod " + to_rod+"<br/>");              return; }
        towerOfHanoi(n - 1, from_rod, aux_rod, to_rod);
        document.write("Move disk " + n + " from rod " + from_rod +" to rod " + to_rod+"<br/>");
        towerOfHanoi(n - 1, aux_rod, to_rod, from_rod);}      var n = 3; //
Number of disks      towerOfHanoi(n, 'E', 'C', 'A') </script>.
```

OUTPUT:-

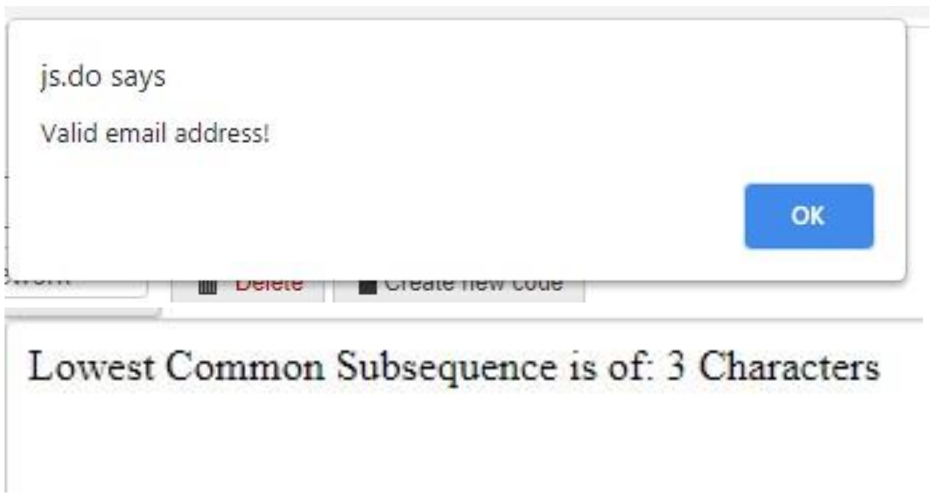
```
Move disk 1 from rod E to rod C
Move disk 2 from rod E to rod A
Move disk 1 from rod C to rod A
Move disk 3 from rod E to rod C
Move disk 1 from rod A to rod E
Move disk 2 from rod A to rod C
Move disk 1 from rod E to rod C
.
```

Q20. Implementation of Dynamic Programming- LCS, Regular Expression Matching.

Ans:-

```
<script>
// Edit your script here  let X, Y;
     function lcs(i, j, count)
    {
        if (i == 0 || j == 0)
            return count;

        if (X[i - 1] == Y[j - 1]) {
            count = lcs(i - 1, j - 1, count + 1);
        }
        count = Math.max(count,
                    Math.max(lcs(i, j - 1, 0),                        lcs(i - 1,
j, 0)));
        return count;
    }

    let n, m;

X = "abcdxyez";
Y = "xyzabcyued";

    n = X.length;     m = Y.length;
            document.write("Lowest Common Subsequence is of: ");
    document.write(lcs(n, m, 0)+" Characters");     function
ValidateEmail(inputText)
{
var mailformat = /^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-
]+)*$/;
if(mailformat.test(inputText))
{
alert("Valid email address!");
return true;
} else {
alert("You have entered an invalid email address!"); return false;
}
}
var s="bindu@yahoo.com";
ValidateEmail(s);
</script>
```

OUTPUT:-



```
js.do says
Valid email address!
```

```
Lowest Common Subsequence is of: 3 Characters
```

Q21. Practical based on backtracking- N Queen's problems.

Ans:-

```
 <script>
// Edit your script here
var iterations = 0

var print_board = function (columns) {   var n =
columns.length, row = 0, col = 0   while (row < n) {     while
(col < n) {
      document.write(columns[row] === col ? 'Q ' : '# ')       col++
    }

    document.write('</br>')
    col = 0     row++
  }
}

var has_conflict = function (columns) {
  var len = columns.length, last = columns[len - 1], previous = len - 2   while (previous >= 0) {
    if (columns[previous] === last) return true     if (last - (len - 1) ===
columns[previous] - previous) return true     if (last + (len - 1) === columns[previous]
+ previous) return true     previous--   }

  return false
}

var place_next_queen = function (total, queens, columns) {   if (queens === 0)
return columns
  columns = columns || []

  for (var column = 0; column < total; column++) {
columns.push(column)     iterations++     if (!has_conflict(columns)
&&
        place_next_queen(total, queens - 1, columns)) {       return columns
    }
    columns.pop(column)
  }

  return null
}

print_board(place_next_queen(17, 12)) document.write('\niterations: ',
iterations)
</script>
```

OUTPUT:-

```
Q # # # # # # # # # # #
# # Q # # # # # # # # #
# # # # Q # # # # # # #
# Q # # # # # # # # # #
# # # Q # # # # # # # #
# # # # # # # # Q # # #
# # # # # # # # # # Q #
# # # # # # # # # # # #
# # # # # # # # # # # #
# # # # # # # # # # # #
# # # # # Q # # # # # # # # # # # # # # Q # #
# #
iterations: 145
```

Shikshan Maharshi Dr. D. Y. Patil Sanstha's

Dr D. Y. Patil Centre For Management & Research, Chikhali, Pune

Java Script Lab Workbook

Subject: Java Script Programming Lab

Course: MCA I - [Sem: I]

Prepared By: -

Name:

Roll No:

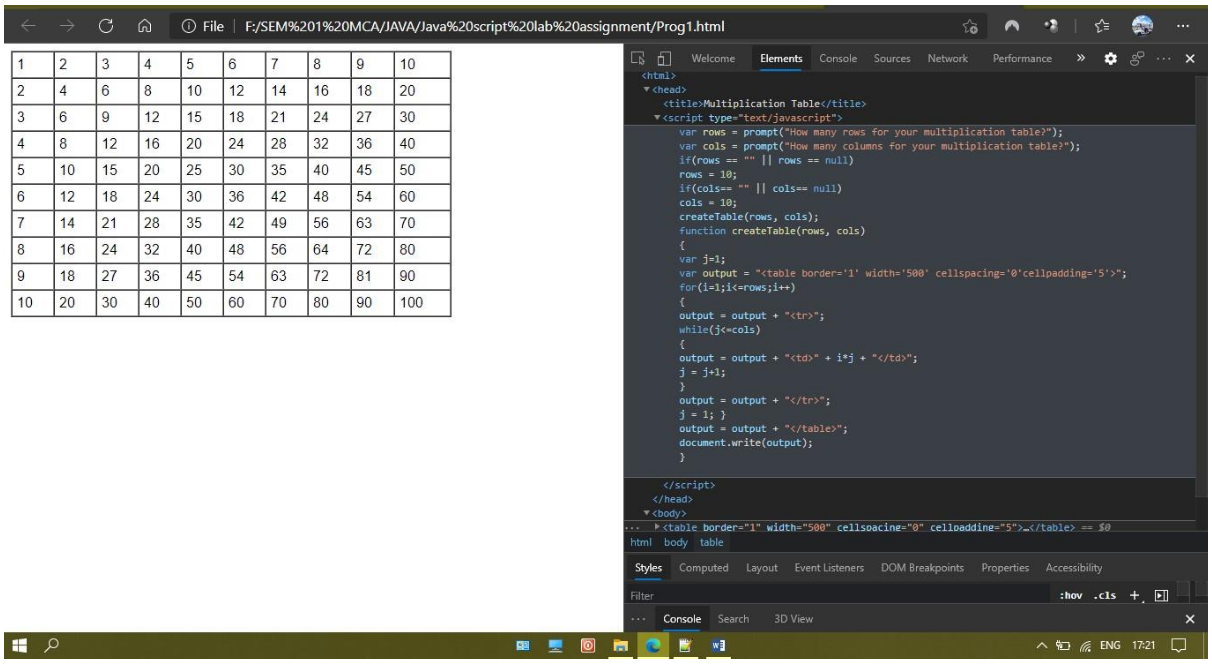
Seat No:

Program Index

Program 1: Write a Javascript program for Multiplication Table.

```
<html>
<head>
 <title>Multiplication Table</title>  <script type="text/javascript">  var rows =
prompt("How many rows for your multiplication table?");  var cols = prompt("How
many columns for your multiplication table?");  if(rows == "" || rows == null)  rows
= 10;  if(cols== "" || cols== null)  cols = 10;  createTable(rows, cols);  function
createTable(rows, cols)
 {
 var j=1;  var output = "<table border='1' width='500' cellspacing='0'cellpadding='5'>";
for(i=1;i<=rows;i++)
 {
 output = output + "<tr>";  while(j<=cols)
 {
 output = output + "<td>" + i*j + "</td>";
 j = j+1;
 }
 output = output + "</tr>";
 j = 1; }
 output = output + "</table>";  document.write(output);
 }
 </script>
</head>
<body>
</body>
</html>
```

Output :



F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/Prog1.html

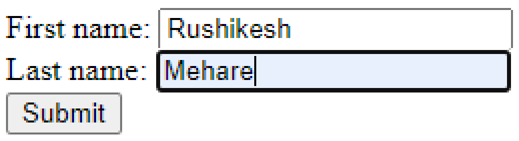
1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

```
<html>
▼<head>
   <title>Multiplication Table</title>
  ▼<script type="text/javascript">
      var rows = prompt("How many rows for your multiplication table?");
      var cols = prompt("How many columns for your multiplication table?");
      if(rows == "" || rows == null)
      rows = 10;
      if(cols== "" || cols== null)
      cols = 10;
      createTable(rows, cols);
      function createTable(rows, cols)
      {
      var j=1;
      var output = "<table border='1' width='500' cellspacing='0'cellpadding='5'>";
      for(i=1;i<=rows;i++)
      {
      output = output + "<tr>";
      while(j<=cols)
      {
      output = output + "<td>" + i*j + "</td>";
      j = j+1;
      }
      output = output + "</tr>";
      j = 1; }
      output = output + "</table>";
      document.write(output);
      }

   </script>
 </head>
▼<body>
... ▶<table border="1" width="500" cellspacing="0" cellpadding="5">…</table> == $0
```

Program 2: Write a JavaScript function to get the values of First and Last name from the form

```
<!DOCTYPE html>
<html><head>
<meta charset=utf-8 />
<title>Return first and last name from a form</title>
<script type="text/javascript">
myFunction = function() {
   var first = document.getElementById("firstname").value;
   var second = document.getElementById("lastname").value;

   document.getElementById("here").innerHTML = first+" "+second;
}
</script>




</head><body>
<form id="here">
<h3>JavaScript function to get the values of First and Last name of the following form.</h3>
First name:<br>
<input type="text" name="firstname" id='firstname'>
<br>
Last name:<br>
<input type="text" name="lastname" id='lastname'>
<br>
<input type="button" value="Send" onclick="myFunction()">
</form>
</body></html>
```

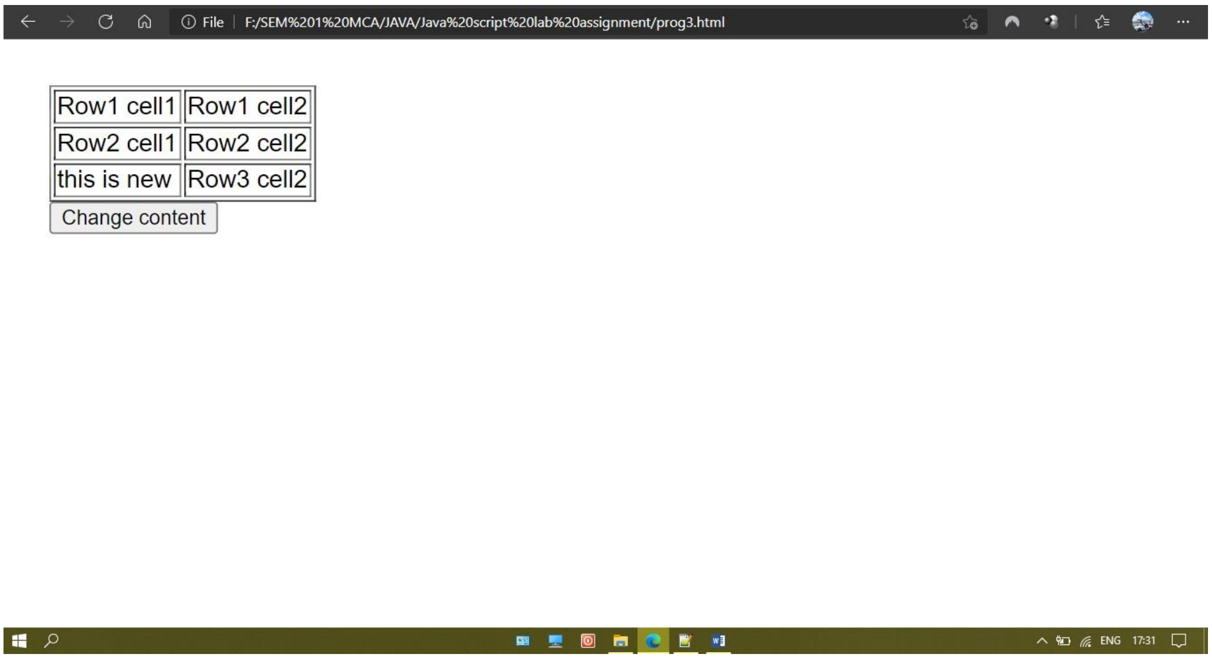
Output:-


First name: Rushikesh
Last name: Mehare

Program 3: Write a JavaScript function that accept row, column, (to identify a particular cell) and a string to update the content of that cell.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>Change the content of a cell</title>
<style type="text/css">  body {margin:
30px;}
</style>
<script type="text/javascript"> function
changeContent()
{
rn = window.prompt("Input the Row number(0,1,2)", "0"); cn =
window.prompt("Input the Column number(0,1)","0"); content =
window.prompt("Input the Cell content");  var
x=document.getElementById('myTable').rows[parseInt(rn,10)].cells;
x[parseInt(cn,10)].innerHTML=content;
}
</script>
</head>
<body>
<table id="myTable" border="1">
<tr><td>Row1 cell1</td>
<td>Row1 cell2</td></tr>
<tr><td>Row2 cell1</td>
<td>Row2 cell2</td></tr>
<tr><td>Row3 cell1</td>
<td>Row3 cell2</td></tr>
</table>
<form>
<input type="button" onclick="changeContent()" value="Change content">
</form>
</body>
</html>
```

Output:-



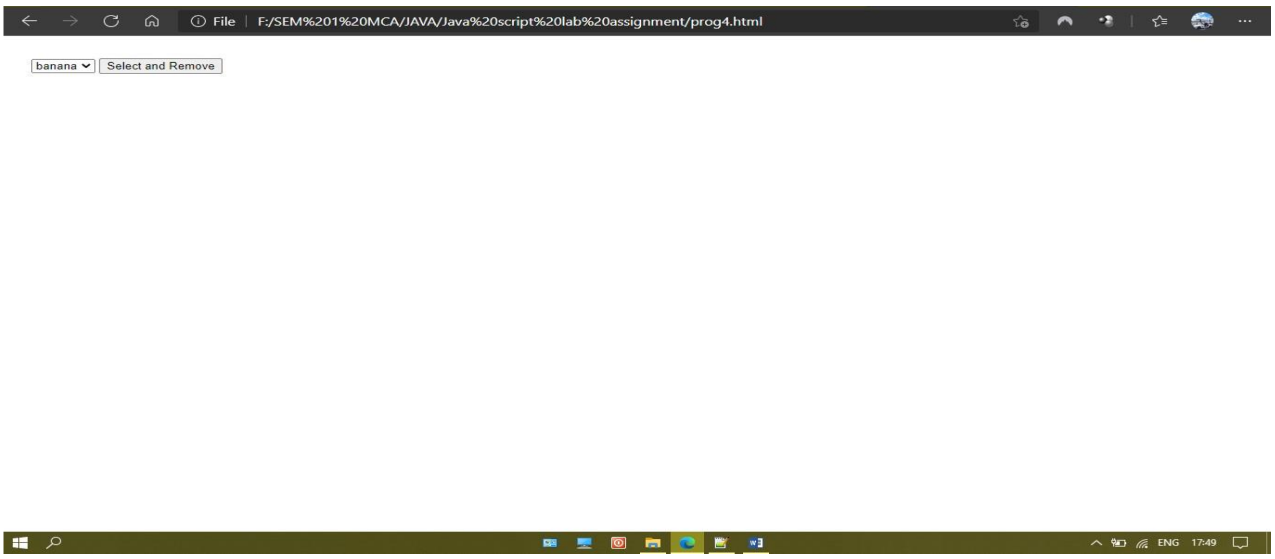
File | F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog3.html

Row1 cell1	Row1 cell2
Row2 cell1	Row2 cell2
this is new	Row3 cell2

Program 4: Write a JavaScript program to remove items from a dropdown list

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">  body {margin:

30px;}

</style>
<meta charset=utf-8 />
<title>Remove items from a dropdown list</title>
<script type="text/javascript"> function removecolor()

{
var x=document.getElementById("colorSelect");
x.remove(x.selectedIndex);
}
</script>
</head><body><form>
<select id="colorSelect">
<option>Red</option>
<option>Green</option> <option>White</option>

<option>Black</option>
</select>
<input type="button" onclick="removecolor()" value="Select and Remove">
</form>
</body>
</html>
```

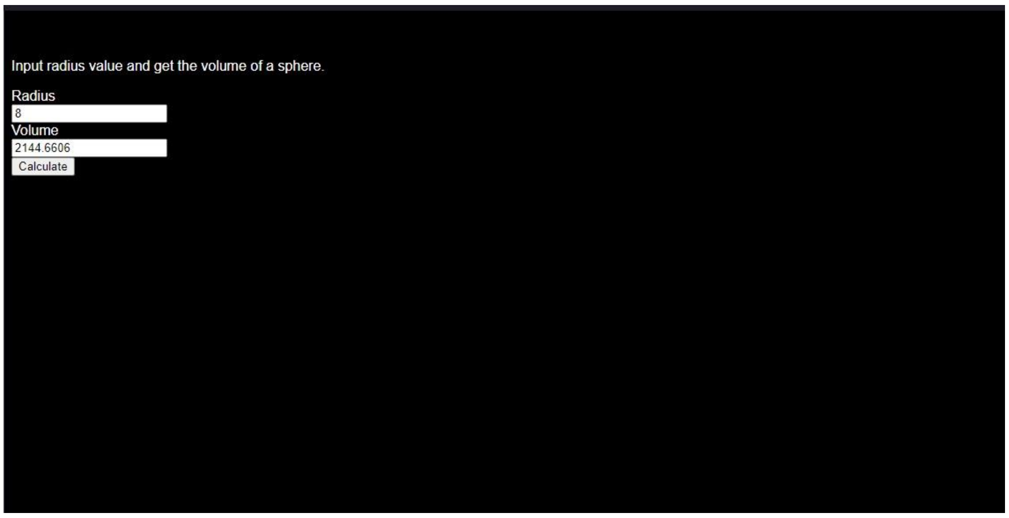
Output:-


File
F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog4.html
banana
Select and Remove
ENG
17:49

Program 5: Write a JavaScript program to calculate the volume of a sphere

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Volume of a Sphere</title>
<style>
body{padding-top:30px;} label,input{display:block;}

</style>
<script>
                function volume_sphere()
 {
  var volume;   var radius =

document.getElementById('radius').value;   radius =

Math.abs(radius);   volume = (4/3) * Math.PI * Math.pow(radius, 3);

volume = volume.toFixed(4);

document.getElementById('volume').value = volume;

  return false;
 }
window.onload = document.getElementById('MyForm').onsubmit = volume_sphere;


</script>
</head>
<body>
<p>Input radius value and get the volume of a sphere.</p>
<form action="" method="post" id="MyForm">
<label for="radius">Radius</label><input type="text" name="radius" id="radius" required>
<label for="volume">Volume</label><input type="text" name="volume" id="volume">
<input type="submit" value="Calculate" id="submit">    </form>
</body>
</html>
```

Output:-



Input radius value and get the volume of a sphere.

Radius

8

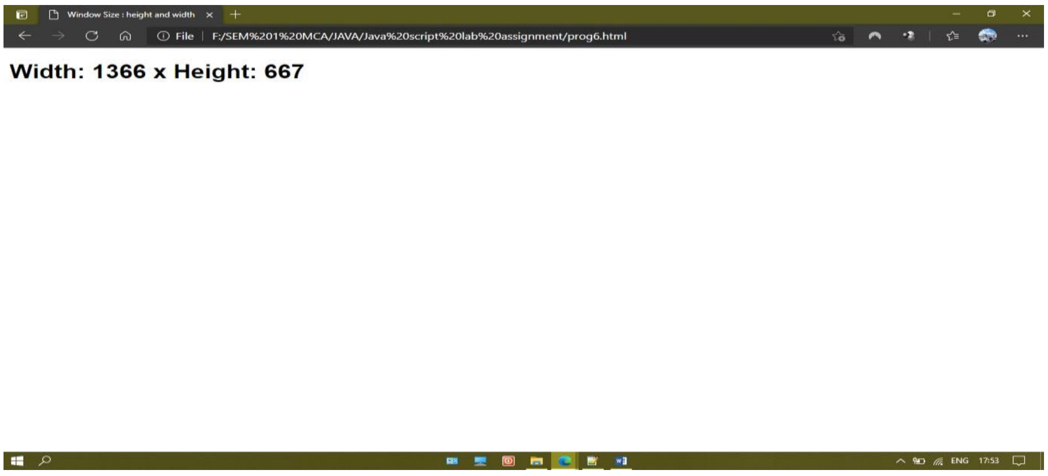
Volume

2144.6606

Program 6: Write a JavaScript program to get the width and height of the window (any time the window is resized).

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>Window Size : height and width</title>
<script type="text/javascript"> function getSize()
{
var w = document.documentElement.clientWidth; var h =
document.documentElement.clientHeight;


// put the result into a h1 tag document.getElementById('wh').innerHTML = "<h1>Width: " + w
+ " x Height: " + h +
"</h1>";
}
</script>
</head>
<!-- Resize the window (here output panel) and see the result !-->
<body onload="getSize()" onresize="getSize()">
<div id="wh">
<!-- Place height and width size here! -->
</div>
<body>
</body>
</html>
```

Output:-



Window Size : height and width

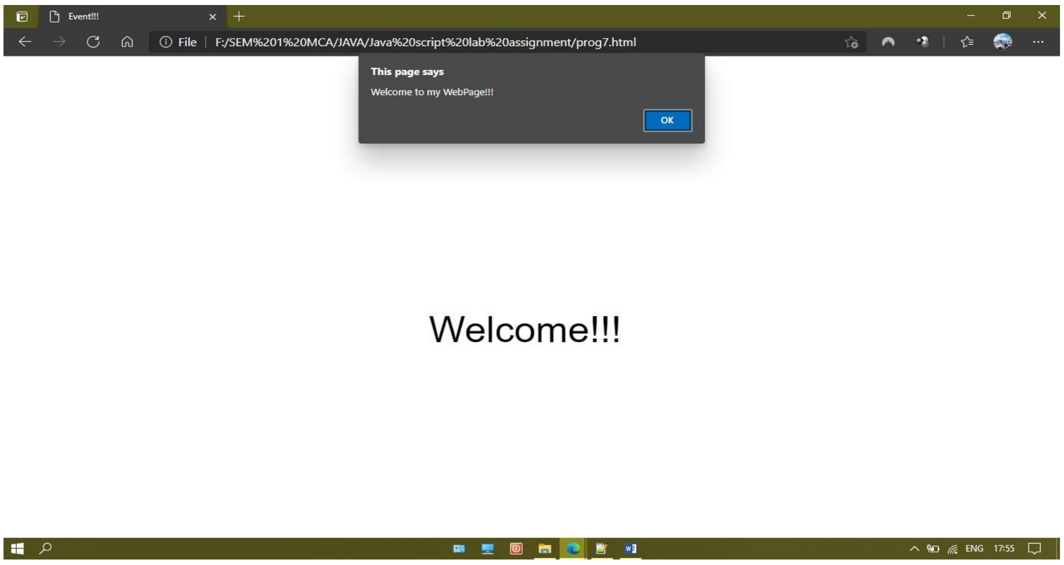
F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog6.html

```
Width: 1366 x Height: 667
```

Program 7: Display a simple message "Welcome!!!" on your demo webpage and when the user hovers over the message, a popup should be displayed with a message "Welcome to my WebPage!!!".

```
<html>
<head>
<title>Event!!!</title> <script
type="text/javascript"> function trigger()
{
document.getElementById("hover").addEventListener("mouseover", popup); function popup()
{
alert("Welcome to my WebPage!!!");
}
}
</script> <style> p
{
 font-size:50px;  position: fixed;
left: 550px;  top: 300px;
}
</style>
</head>
<body onload="trigger();">
<p id="hover">Welcome!!!</p>
</body>
</html>
```

Output:-



Event!!!

File | F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog7.html

This page says

Welcome to my WebPage!!!

OK

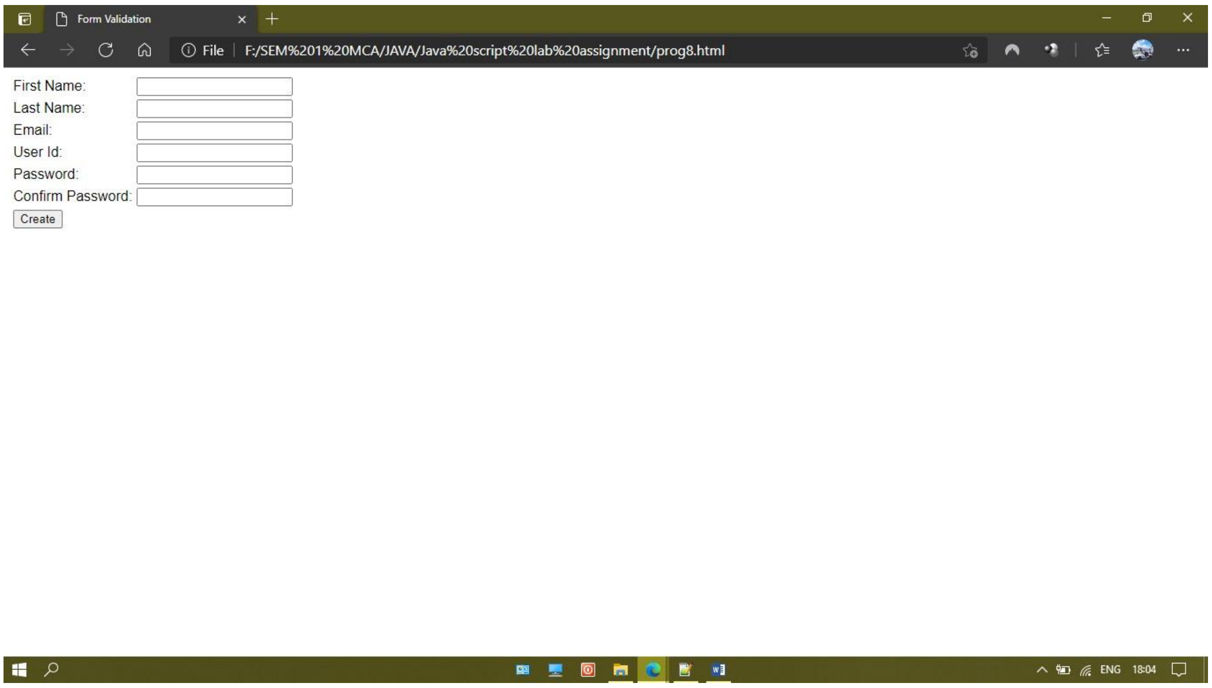
Welcome!!!

Program 8: Create a sample form program that collects the first name, last name, email, user id, password and confirms password from the user. All the inputs are mandatory and email address entered should be in correct format. Also, the values entered in the password and confirm password textboxes should be the same. After validating using JavaScript, In output display proper error messages in red color just next to the textbox where there is an error.

```
<html>
<head>
 <title>Form Validation</title>  <script
type="text/javascript">  var divs = new
Array();  divs[0] = "errFirst";  divs[1] =
"errLast";  divs[2] = "errEmail";  divs[3] =
"errUid";  divs[4] = "errPassword";  divs[5] =
"errConfirm";  function validate()
{
 var inputs = new Array();  inputs[0] = document.getElementById('first').value;  inputs[1]
= document.getElementById('last').value;  inputs[2] =
document.getElementById('email').value;  inputs[3] =
document.getElementById('uid').value;  inputs[4] =
document.getElementById('password').value;  inputs[5] =
document.getElementById('confirm').value;  var errors = new Array();  errors[0] =
"<span style='color:red'>Please enter your first name!</span>";  errors[1] = "<span
style='color:red'>Please enter your last name!</span>";  errors[2] = "<span
style='color:red'>Please enter your email!</span>";  errors[3] = "<span
style='color:red'>Please enter your user id!</span>";  errors[4] = "<span
style='color:red'>Please enter your password!</span>";  errors[5] = "<span
style='color:red'>Please confirm your password!</span>";
 for (i in inputs)
 {
var errMessage = errors[i]; var div =
divs[i]; if (inputs[i] == "")
document.getElementById(div).innerHTML = errMessage;
```

```
 else if (i==2)
 {
 var atpos=inputs[i].indexOf("@");  var

dotpos=inputs[i].lastIndexOf(".");

 if (atpos<1 || dotpos<atpos+2 || dotpos+2>=inputs[i].length)
document.getElementById('errEmail').innerHTML = "<span style='color: red'>Enter a valid email
address!</span>";
 else
 document.getElementById(div).innerHTML = "OK!";
 }
 else if (i==5)
 {
 var first = document.getElementById('password').value;  var second =

document.getElementById('confirm').value;

 if (second != first)
 document.getElementById('errConfirm').innerHTML = "<span style='color: red'>Your passwords don't
match!</span>";
 else
 document.getElementById(div).innerHTML = "OK!";
 }
 else
 document.getElementById(div).innerHTML = "OK!";
 }
 }
 function finalValidate()
 {
 var count = 0;  for(i=0;i<6;i++)  {

 var div = divs[i];
 if(document.getElementById(div).innerHTML == "OK!")  count = count

+ 1;

 }
 if(count == 6)  document.getElementById("errFinal").innerHTML = "All the data you

entered is  correct!!!";

 }
 </script>
</head>
<body>
<table id="table1">
 <tr>
 <td>First Name:</td>
 <td><input type="text" id="first" onkeyup="validate();" /></td>
 <td><div id="errFirst"></div></td>
 </tr>
 <tr>
 <td>Last Name:</td>
 <td><input type="text" id="last" onkeyup="validate();"/></td>
 <td><div id="errLast"></div></td>
 </tr> <tr>
 <td>Email:</td>
 <td><input type="text" id="email" onkeyup="validate();"/></td>
 <td><div id="errEmail"></div></td>
 </tr>
 <tr>
 <td>User Id:</td>
 <td><input type="text" id="uid" onkeyup="validate();"/></td>
 <td><div id="errUid"></div></td>
</tr>
<tr>
<td>Password:</td>
<td><input type="password" id="password" onkeyup="validate();"/></td>
 <td><div id="errPassword"></div></td>
 </tr>
 <tr>
 <td>Confirm Password:</td>
 <td><input type="password" id="confirm" onkeyup="validate();"/></td>
 <td><div id="errConfirm"></div></td>
 </tr>
 <tr>
 <td><input type="submit" id="create" value="Create"  onclick="validate();finalValidate();"/></td>
 <td><div id="errFinal"></div></td>
 </tr>
</table>
</body>
</html>
```

Output:-


Form Validation

F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog8.html

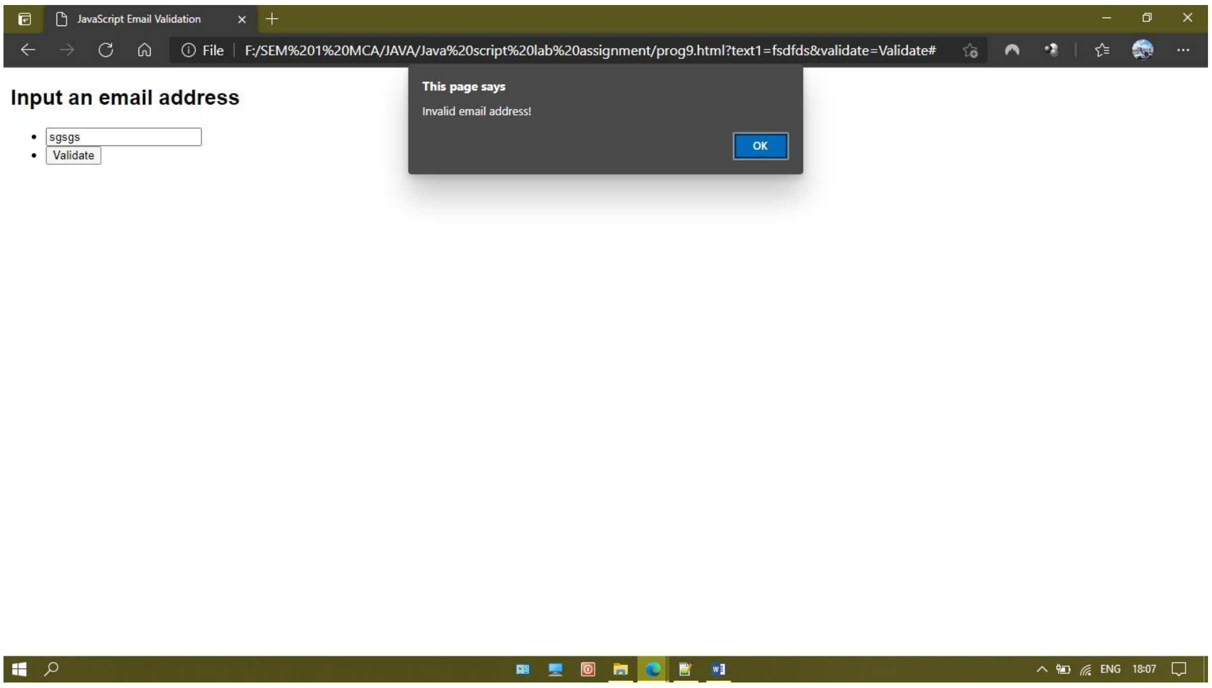
First Name:	
Last Name:	
Email:	
User Id:	
Password:	
Confirm Password:	

Create

Program 9: Write the Java Script Program to validate Email id using Regular Expression.

```
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8" />
 <title>JavaScript Email Validation</title>
 <link rel="stylesheet" href="email.css" type="text/css" />
 </head>
 <body onload="document.form1.text1.focus()">
 <div class="mail">
 <h2>Input an email address</h2>
 <form name="form1" action="#">
 <ul>
 <li><input type="text" name="text1" /></li>

 <li class="validate">
 <input  type="submit"  name="validate"  value="Validate"
onclick="ValidateEmail(document.form1.text1)"
 />
 </li>
 </ul>
 </form>
 </div>  <script> function ValidateEmail(input) {  var validRegex = /^[a-zA-Z0-
9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/; if (input.value.match(validRegex))
{
alert("Valid               email               address!");
document.form1.text1.focus(); return true;
} else {  alert("Invalid email address!");
document.form1.text1.focus();
 return false;
 }
}
</script>
</body>
</html>
```

Output:-


JavaScript Email Validation

F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog9.html?text1=fsdfds&validate=Validate#

Input an email address

- sgsgs
- Validate

This page says

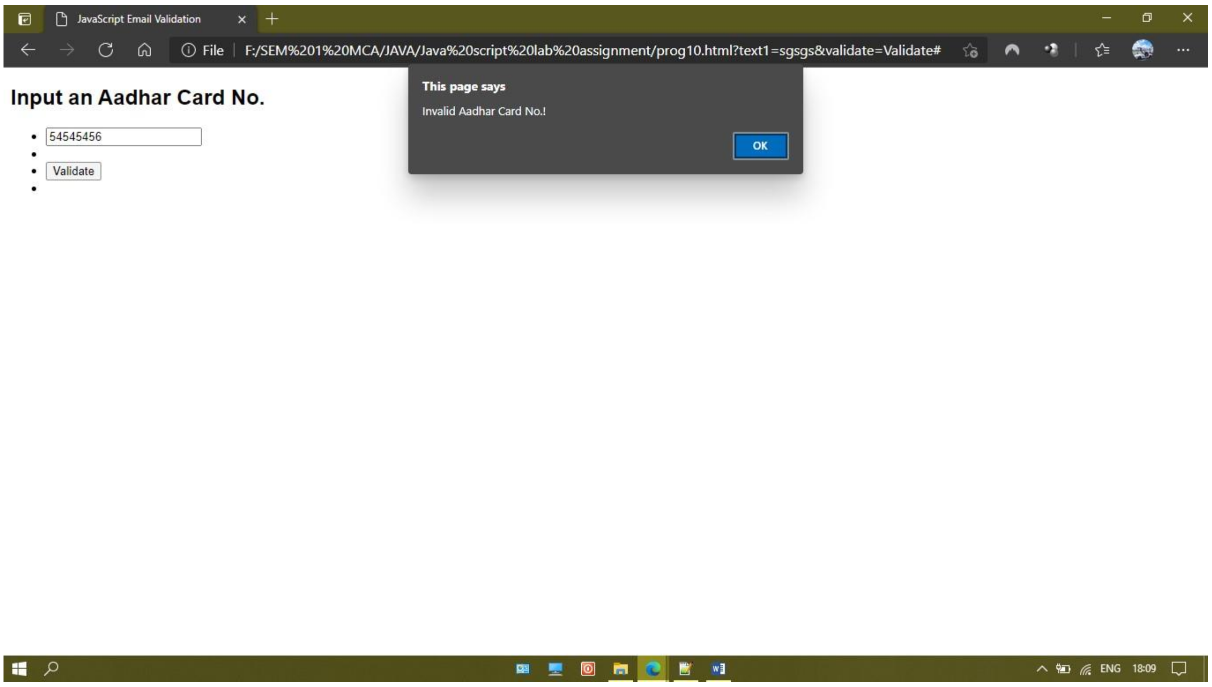
Invalid email address!

OK

Program 10: **Write the Java Script Program to validate Aadhaar Card No. using Regular Expression.**

```
<!DOCTYPE html>
<html lang="en">
 <head>
 <meta charset="utf-8" />
 <title>JavaScript Email Validation</title>
 <link rel="stylesheet" href="email.css" type="text/css" />
 </head>
 <body onload="document.form1.text1.focus()">
 <div class="mail">
 <h2>Input an Aadhar Card No.</h2>
 <form name="form1" action="#">
 <ul>
 <li><input type="text" name="text1" /></li>
 <li>&nbsp;</li>
 <li class="validate">  <input  type="submit"  name="validate"

value="Validate"

onclick="ValidateEmail(document.form1.text1)"

 />
 </li>
 <li>&nbsp;</li>
 </ul>
 </form>
 </div>  <script> function ValidateEmail(input) { var validRegex = /^[2-

9]{1}[0-9]{3}\s{1}[0-9]{4}\s{1}[0-9]{4}$/; if

(input.value.match(validRegex)) { alert("Valid Aadhar Card No.!");

document.form1.text1.focus(); return true;

 } else {  alert("Invalid Aadhar Card No.!");

document.form1.text1.focus();  return false;

 }
}
</script>
</body>
</html>
```

Output:-



JavaScript Email Validation

F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog10.html?text1=sgsgs&validate=Validate#

Input an Aadhar Card No.

- 54545456
-
- Validate
-

This page says

Invalid Aadhar Card No.!

OK

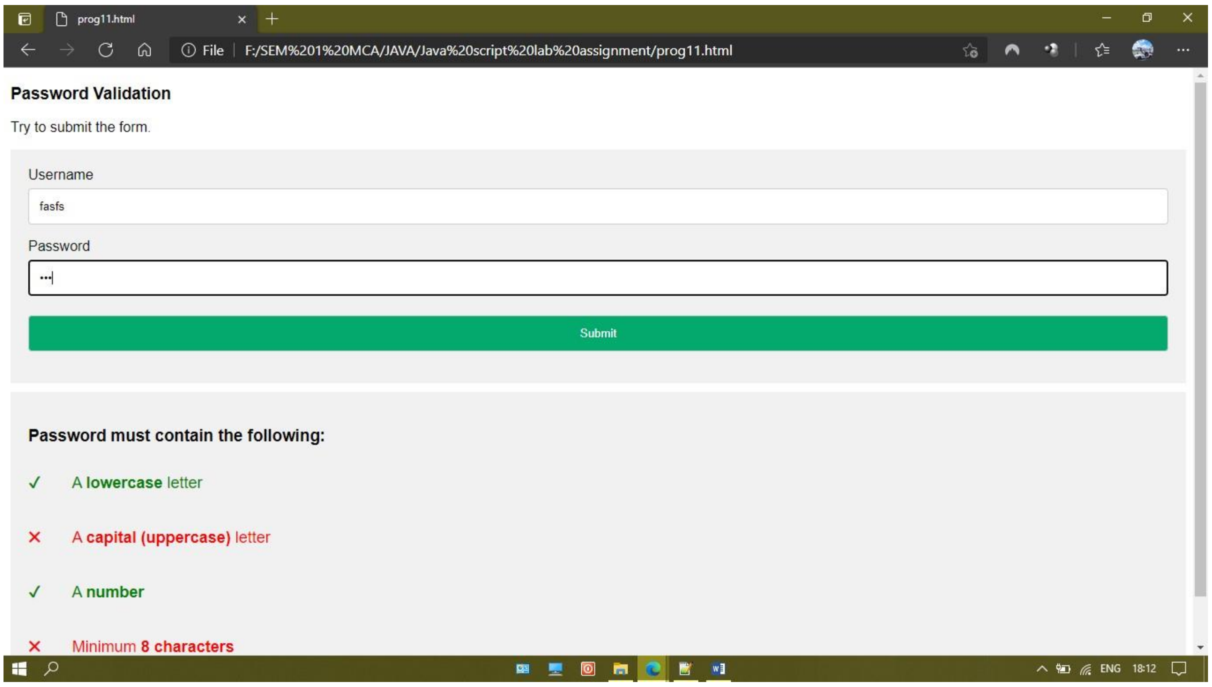
Program 11: Write the JavaScript Program to Validate the Password in the login form.

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
/* Style all input fields */ input {  width:
100%;  padding: 12px;  border: 1px
solid #ccc;  border-radius: 4px;  box-
sizing: border-box;  margin-top: 6px;
margin-bottom: 16px;
}
/* Style the submit button */ input[type=submit] {
background-color: #04AA6D;  color: white;
}
/* Style the container for inputs */
.container {  background-color: #f1f1f1;
padding: 20px;
}
/* The message box is shown when the user clicks on the password field */
#message {  display:none;  background:
#f1f1f1; color: #000;
position: relative; padding: 20px; margin-
top: 10px;
}
#message p {  padding: 10px 35px;
font-size: 18px;
}
/* Add a green text color and a checkmark when the requirements are right */
.valid { color: green;
}
.valid:before {  position: relative;
left: -35px;  content: "✓";
}
```

```
/* Add a red text color and an "x" when the requirements are wrong */
.invalid {  color: red;

}
.invalid:before {  position: relative;

left: -35px;  content: "✖";

}
</style>
</head>
<body>
<h3>Password Validation</h3>
<p>Try to submit the form.</p>
<div class="container">
<form action="/action_page.php">
<label for="usrname">Username</label>
 <input type="text" id="usrname" name="usrname" required>
 <label for="psw">Password</label>
 <input type="password" id="psw" name="psw" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"  title="Must

contain at least one number and one uppercase and lowercase letter, and at least 8 or  more characters"

required>


 <input type="submit" value="Submit">
 </form>
</div>
<div id="message">
 <h3>Password must contain the following:</h3>
 <p id="letter" class="invalid">A <b>lowercase</b> letter</p>
 <p id="capital" class="invalid">A <b>capital (uppercase)</b> letter</p>
 <p id="number" class="invalid">A <b>number</b></p>
 <p id="length" class="invalid">Minimum <b>8 characters</b></p></div>
<script> var myInput = document.getElementById("psw"); var

letter = document.getElementById("letter"); var capital =

document.getElementById("capital"); var number =

document.getElementById("number"); var length =

document.getElementById("length");

// When the user clicks on the password field, show the message box myInput.onfocus =

function() {  document.getElementById("message").style.display = "block";

}
```

```
// When the user clicks outside of the password field, hide the message box myInput.onblur =
function() {  document.getElementById("message").style.display = "none";
}
// When the user starts to type something inside the password field
myInput.onkeyup = function() {  // Validate lowercase letters  var
lowerCaseLetters = /[a-z]/g;  if(myInput.value.match(lowerCaseLetters)) {
letter.classList.remove("invalid");  letter.classList.add("valid");
 } else {  letter.classList.remove("valid");
letter.classList.add("invalid");
 }

 // Validate capital letters  var upperCaseLetters = /[A-Z]/g;
if(myInput.value.match(upperCaseLetters)) {
capital.classList.remove("invalid");
capital.classList.add("valid");
 } else {  capital.classList.remove("valid");
capital.classList.add("invalid");
 }
 // Validate numbers  var numbers = /[0-9]/g;
if(myInput.value.match(numbers)) {
number.classList.remove("invalid");
number.classList.add("valid"); } else {
number.classList.remove("valid");
number.classList.add("invalid");
 }

 // Validate length
if(myInput.value.length >= 8) { length.classList.remove("invalid");
length.classList.add("valid");
 } else {  length.classList.remove("valid");
length.classList.add("invalid");
 }
}
</script>
</body>
</html>
```

Output:-


Password Validation

Try to submit the form.

Username

fasfs

Password

Submit

Password must contain the following:

✓ A **lowercase** letter

✗ A **capital (uppercase)** letter

✓ A **number**

✗ Minimum **8 characters**

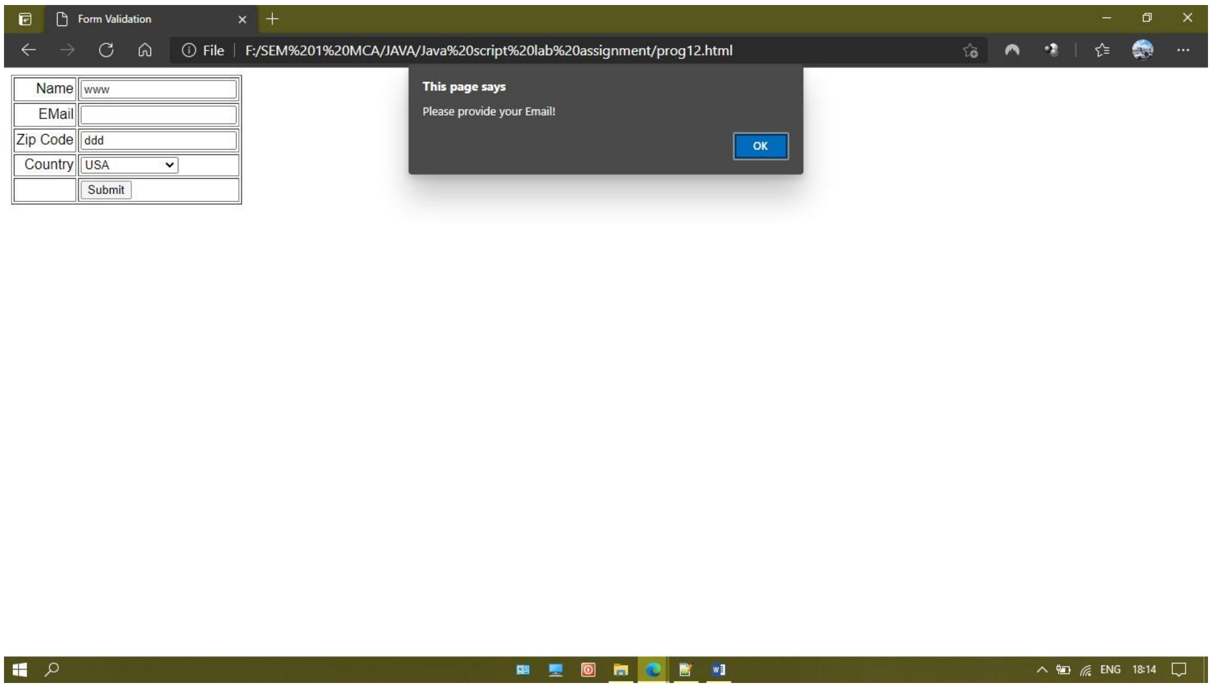
prog11.html
File
F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog11.html
ENG
18:12

Program 12: **Write the JavaScript Program to Validate the Basic Registration form.**

```
<html>
 <head>
 <title>Form Validation</title>
 <script type = "text/javascript">
 <!--  function validate() {


 if( document.myForm.Name.value == "" ) {  alert( "Please
provide your name!" );  document.myForm.Name.focus() ;
 return false;
 }
 if( document.myForm.EMail.value == "" ) {  alert(
"Please provide your Email!" );
document.myForm.EMail.focus() ;  return false;
 }
 if( document.myForm.Zip.value == "" || isNaN( document.myForm.Zip.value ) ||
document.myForm.Zip.value.length != 5 ) {


 alert( "Please provide a zip in the format #####." );
document.myForm.Zip.focus() ;  return false;
 }
 if( document.myForm.Country.value == "-1" ) {  alert(
"Please provide your country!" );  return false;
 }
 return( true );
}
</script>
 </head>

 <body>
 <form action = "/cgi-bin/test.cgi" name = "myForm" onsubmit = "return(validate());">  <table
cellspacing = "2" cellpadding = "2" border = "1">


 <tr>
```

```
<td align = "right">Name</td>
<td><input type = "text" name = "Name" /></td>
</tr>


<tr>
<td align = "right">EMail</td>
<td><input type = "text" name = "EMail" /></td> </tr>


<tr>
<td align = "right">Zip Code</td>
<td><input type = "text" name = "Zip" /></td>
</tr>


<tr>
<td align = "right">Country</td>
<td>
<select name = "Country">
<option value = "-1" selected>[choose yours]</option>
<option value = "1">USA</option>
<option value = "2">UK</option>
<option value = "3">INDIA</option>
</select>
</td>
</tr>


<tr>
<td align = "right"></td>
<td><input type = "submit" value = "Submit" /></td>
</tr>


</table>
</form>
</body>
</html>
```

Output:-



Form Validation

F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog12.html

Name	www
EMail	
Zip Code	ddd
Country	USA
	Submit

This page says

Please provide your Email!

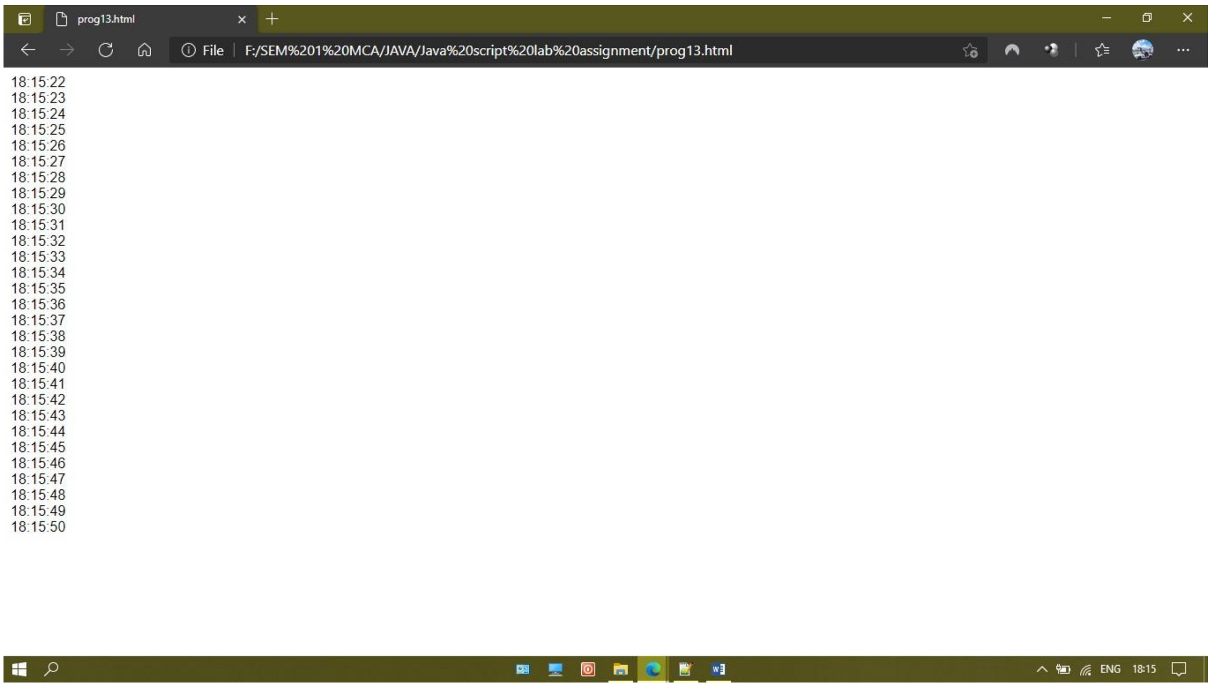
OK

Program 13: Write the Java Script program to create Clock and display the time in each second.

```
<!DOCTYPE html>
 <html>
 <head>
 <meta charset="utf-8">
 <title>Create a clock and display the time in each second </title>
<script type = "text/javascript">
 <!--
function my_Clock()
 {
 this.cur_date = new Date();  this.hours =

this.cur_date.getHours();  this.minutes =

this.cur_date.getMinutes();  this.seconds =

this.cur_date.getSeconds();

 }
my_Clock.prototype.run = function ()
 {
 setInterval(this.update.bind(this), 1000);
 };
my_Clock.prototype.update = function ()
 {
 this.updateTime(1); document.write(this.hours + ":" + this.minutes + ":" +

this.seconds+"<br>");

   };

my_Clock.prototype.updateTime = function (secs)
 {
 this.seconds+= secs; if (this.seconds

>= 60)

 {
 this.minutes++;
 this.seconds= 0;
 }
if (this.minutes >= 60)
 {
 this.hours++;  this.minutes=0;

 }
if (this.hours >= 24)
 {
```

```
 this.hours = 0;
 } };
var clock = new my_Clock();
 clock.run(); </script>
 </head>
 <body>
<form onsubmit="return my_Clock()" method="post">
</body>
 </html>
```

Output:-



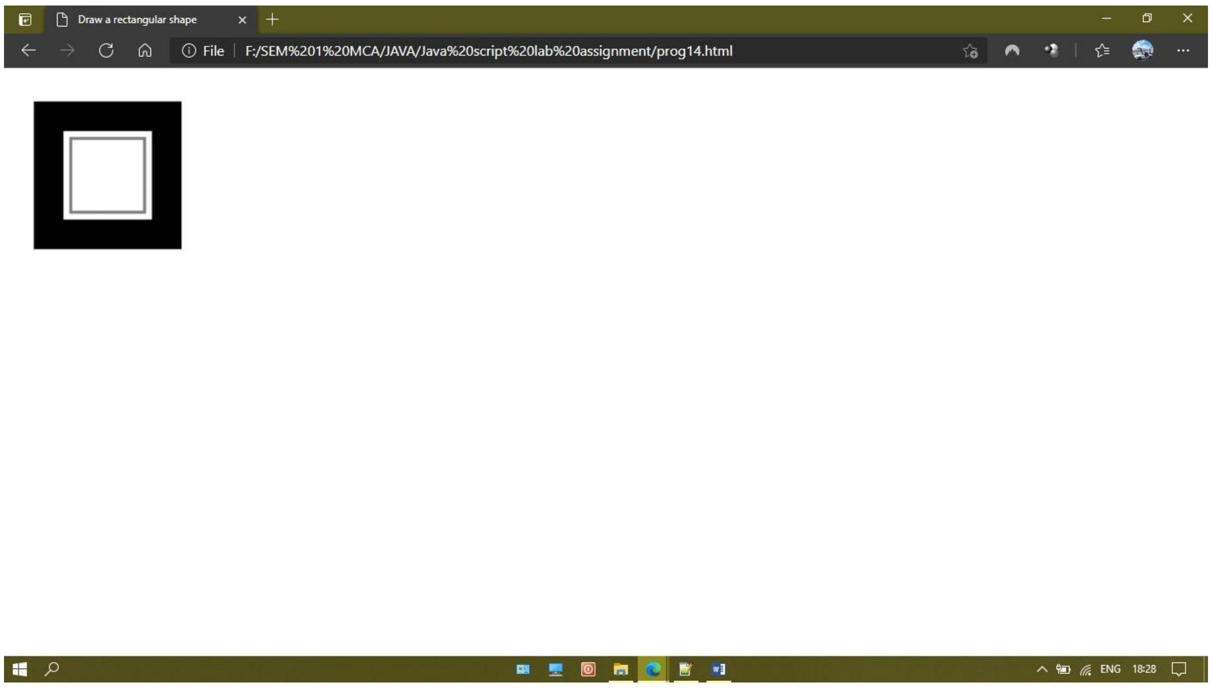
prog13.html
File
F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog13.html
ENG
18:15

```
18:15:22
18:15:23
18:15:24
18:15:25
18:15:26
18:15:27
18:15:28
18:15:29
18:15:30
18:15:31
18:15:32
18:15:33
18:15:34
18:15:35
18:15:36
18:15:37
18:15:38
18:15:39
18:15:40
18:15:41
18:15:42
18:15:43
18:15:44
18:15:45
18:15:46
18:15:47
18:15:48
18:15:49
18:15:50
```

Program 14: Write the Java Script program to create to draw a rectangular shape.

```
<!DOCTYPE html>
 <html>
 <head>
 <meta charset="utf-8">
 <title>Draw a rectangular shape</title>
<script type = "text/javascript">
 <!--
function draw() {  var canvas =

document.getElementById('canvas');  if (canvas.getContext) {

var context = canvas.getContext('2d');

context.fillRect(20,20,100,100);

context.clearRect(40,40,60,60);

context.strokeRect(45,45,50,50);

 }
}
 </script>
 </head>
<body onload="draw();">
<canvas id="canvas" width="150" height="150"></canvas>
</body>
</html>
```

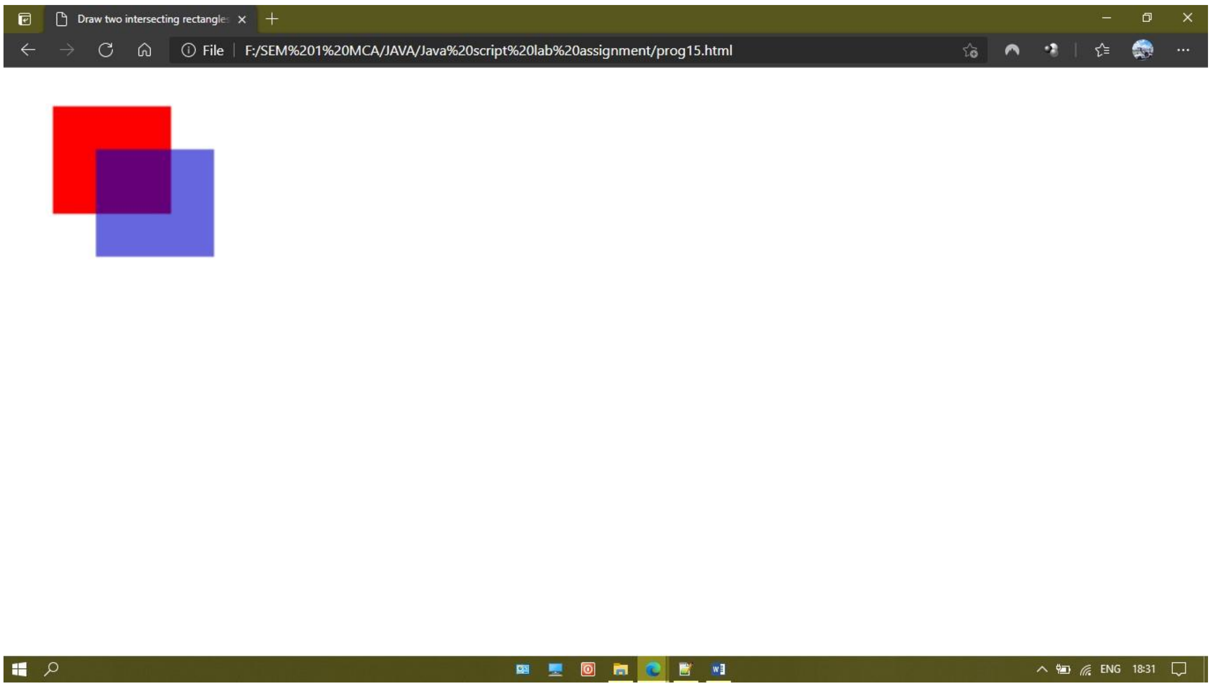
Output:-


Draw a rectangular shape
File | F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog14.html
ENG 18:28

Program 15: Write the JavaScript program draw two intersecting rectangles, one of which has alpha transparency.

```
<!DOCTYPE html>
 <html>
 <head>
 <meta charset="utf-8">
 <title>Draw two intersecting rectangles, one of which has alpha transparency</title>
<script type = "text/javascript">
 <!--
function draw()
 {
 var canvas = document.getElementById("canvas");  if
(canvas.getContext)
 {
 var context = canvas.getContext("2d");  context.fillStyle =
"rgb(256,0,0)";  context.fillRect (15, 10, 55, 50);

 context.fillStyle = "rgba(0, 0, 200, 0.6)";  context.fillRect (35, 30,
55, 50);
 }
 }
 </script>
 </head>
<body onload="draw();">
 <canvas id="canvas" width="150" height="150"></canvas>
 </body>
 </html>
```

Output:-


Draw two intersecting rectangle:
File
F:/SEM%201%20MCA/JAVA/Java%20script%20lab%20assignment/prog15.html
ENG
18:31