ASSIGNMENT-2

Create a responsive web design for small device (mobile), tablet and desktop using different breakpoint (write media query) theme: Hotels room.

- Mobile and Tablet breakpoint at min-width: 500px.
- Tablet and desktop breakpoint at min-width: 800px.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-
    width; initial-scale=1.0">
<title> Hotels Rooms </title>
<style>
    body {
    font-family: Arial, sans-serif;
    background-color
    margin: 0;
    padding: 0;
}
header{
    background-color: #333;
    color: #fff;
    padding: 20px;
}
header h1 {
    margin: 0;
}
nav ul {
list-style-type: none;
```

```
    padding: 0;
}
nav ul li {
    display: inline;
    margin-right: 20px;
}
nav ul li a {
    color: #fff;
    text-decoration: none;
}
main {
    padding: 20px;
}
.room {
    background-color: #f9f9f9;
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 20px;
    margin-bottom: 20px;
}
.room h2 {
    margin-top: 0;
}
footer {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 10px 0;
}
@media (min-width: 500px) {}
@media (min-width: 800px) {}
```

```
@media(min-width:500px)
        {
        body {
          font-size: 16px;
        }
      }
@media(min-width:800px)
        {
        body {
            font-size: 18px;
        }
      }
</style>
</head>
<body>
    <header>
        <h1> Hotel Rooms </h1>
        <nav> <ul>
            <li> <a href='#'> Home </a> </li>
            <li> <a href='#'> Rooms </a> </li>
            <li> <a href='#'> Contact </a> </li>
        </ul>
    </nav>
</header>
<main>
    <section class = "room">
      <h2> Standard Room </h2>
      <p> Our standard room offers a comfortable
          stay at an affordable price. </p>
    </section>
    <section class = "room">
        <h2> Luxury Suite </h2>
```

```
    <p>Indulge in luxury with our spacious suites,
      featuring breathtaking views</p>
  </section>
</main>
<footer>
  <p>&copy; 2024 Hotel Rooms. All rights reserved.</p>
</footer>
</body>
</html>
```

Q3. Add a 10 second animation for the four square box:
- One going from top left to the top right and stay at top right
- Second going from bottom right to bottom left and stay at bottom left.
- Third going from just below the first box vertically from top left to bottom left and stay at bottom
- fourth going from bottom right to top right and stay at top right.
- In between a small circle moving randomly with different time % value at different location for the infinite time.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
  width, initial-scale=1.0">
  <title>Q3</title>
  <style>
```

```
.box1 {
  width: 100px;
  position: relative;
  height: 100px;
  background-color: dodgerblue;
  border: 5px solid black;
  border-radius: 6%
 animation: am1 5s linear 2s infinite alternate;
}
@keyframes am1 {
    from {
        left: 0px;
        top: 0px;
    }
   to {
     left: 955px;
     top: 0px;
    }
  }
.box2 {
  margin-top: 2px;
  width: 50px;
  position: relative;
  height: 100px;
  background-color: dodgerblue;
  border: 5px solid black;
  border-radius: 6%;
  animation: am2 5s linear 2s infinite alternate;
}
@keyframes am2 {
    from {
```

```
        left: 0;
        top: 0px;
    }
    to {
      left: 0;
      top: 157px;
    }
}

.box3 {
    margin-top: -160px;
    width: 50px;
    position: relative;
    height: 100px;
    background-color: dodgerblue;
    border: 5px solid black;
    border-radius: 6%;
    float: right;
    animation: am3 5s linear 2s infinite alternate;
}

@keyframes am3 {
    from {
      left: 0;
      top: 157px;
    }
    to {
      left: 0;
      top: 0px;
    }
}

.box4 {
    margin-top: 159px;
    margin-left: 60px;
    position: relative;
```

```
    width: 100px;
    height: 100px;
    background-color: dodgerblue;
    border: 5px solid black;
    border-radius: 6%;
    animation: am4 5s linear 0.2s infinite alternate;
}
@keyframes am4 {
    from {
        left: 955px;
        top: 0px;
    }
    to {
        left: 0px;
        top: 0px;
    }
}
.circle {
    margin-top: 100px;
    position: absolute;
    width: 50px;
    height: 50px;
    border: 7px solid black;
    background-color: rgb(255, 38, 0);
    border-radius: 50%;
    animation: circleMovement 10s linear 2s
        infinite alternate;
}
@keyframes circleMovement {
    0% {
        top: 27px;
        left: 100px;
    }
```

```
    25% {
      top: 125px;
      left: 70px;
    }
    50% {
      top: 0px;
      left: 910px;
    }
    75% {
      top: 125px;
      left: 75%;
    }
    100% {
      top: 50px;
      left: 70px;
    }
  }
</style>
</head>
<body>
  <div class = "box1"></div>
  <div class = "box2"></div>
  <div class = "circle"></div>
  <div class = "box3"></div>
  <div class = "box4"></div>
</body>
</html>
```

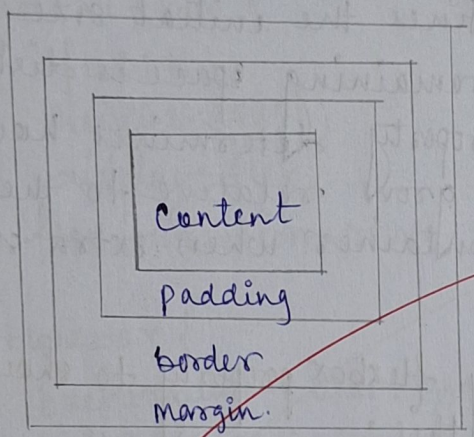
Q4. Create a loading animations effect for 10 seconds which shows at the time of loading of a page.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Loading Animation</title>
<style>
    body {
        margin: 0;
        padding: 0;
        font-family: Arial, sans-serif;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        background-color: #f0f0f0;
    }
    .loader {
        border: 8px solid #f3f3f3;
        border-radius: 50%;
        border-top: 8px solid #3498db;
        width: 50px;
        height: 50px;
        animation: spin 2s linear infinite;
    }
    @keyframes spin {
        0% {
            transform: rotate(0deg);
        }
        100% {
            transform: rotate(360deg);
        }
    }
```

```
    }
    .loading-text{
        margin-top: 20px;
    }
</style>
</head>
<body>
    <div class = "loader"></div>
    <div class = "loading-text">Loading....</div>
<script>
    window.addEventListener('load', function(){
        const loader = document.querySelector('.loader');
        const loadingText = document.querySelector('.loading-
                                                    text');
        setTimeOut(function(){
            loader.style.display = 'none';
            loadingText.style.display = 'none';
        }, 10000);
    });
</script>
</body>
</html>
```

Describe CSS flexbox. Write five properties of flex-container and flex-items.

Ans:- CSS flexbox is a layout model that provides a more efficient way to design and align items within a container. It offer a flexible and dynamic approach to layout, allowing developers to create complex and responsive designs with ease. Flexbox introduces two main components: flexContainer and flex item.



Flex Container Properties:

1. display: This property defines the element as a flex container. The value 'display: flex;' or 'display: inline-flex;' determines whether the flex container is block-level or inline-line, respectively.
2. flex-direction: It specifies the direction in which the flex items are placed within the flex container.
3. justify-content: This property aligns flex items along the main axis of the flex container.
4. align-items: It defines how flex items are aligned along the cross-axis of the flex container.
5. flex-wrap: This property controls whether flex items are forced into a single line or can be wrapped onto multiple lines within the flex container.

Flex Item Properties:

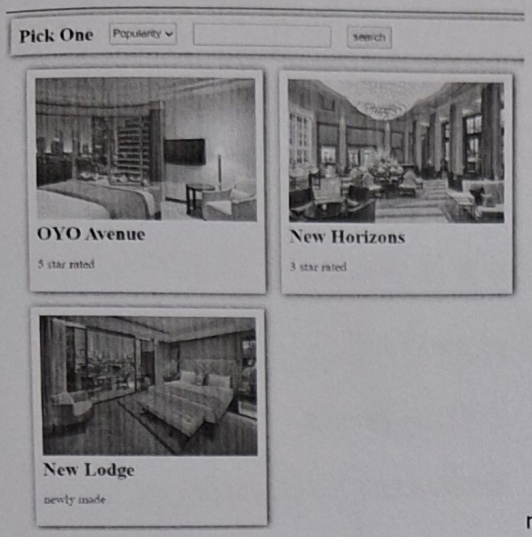
1. flex: The shorthand for three individual properties 'flex-grow', 'flex-shrink' and 'flex-basis'.
2. order: This ~~order~~ property specifies the order in which flex items appear within the flex container.
3. align-self: It overrides the default assignment defined by 'align-items' for a specific flex item.
4. flex-basis: It defines the initial size of a flex item before the remaining space is distributed.
5. flex-grow: This property determines how much a flex item will grow relative to the other flex items in the container when extra space is available.

Q2. Create a page using flexbox property to show page something like this:
Create multiple division and place image accordingly:

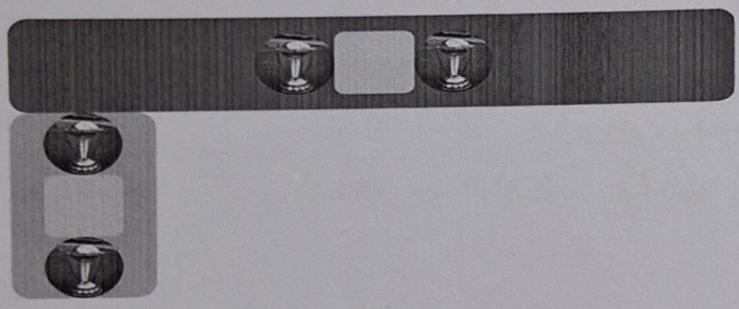
```
<!DOCTYPE html>
<html>
<head>
<title> Document </title>
<style>
    .container{
        display: flex;
        justify-content: space-between;
        flex-direction: column;
        height: 100vh;
        width: 100%;
    }
```

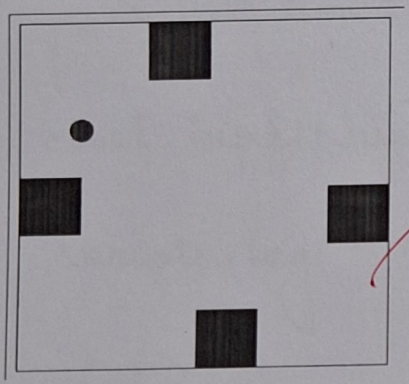
```
.horizontal-box{
    flex:1;
    display:flex;
    align-items:center;
    justify-content:center;
    background-color:rgb(255,0,0);
    border-radius:20px;
    padding:20px;
}
.vertical-box {
        flex:11;
        height:100%;
        display:flex;
      flex-direction:row;
   }
#bluebox {
      background-color:cyan;
      height:100%;
      border-radius:20px;
    flex:1;
    padding:20px;
    display:flex;
    flex-direction:column;
    align-items:center;
    justify-content:center;
   }
#blank {
      flex:11;
   }
.square {
    height:100px;
    width:100px;
```

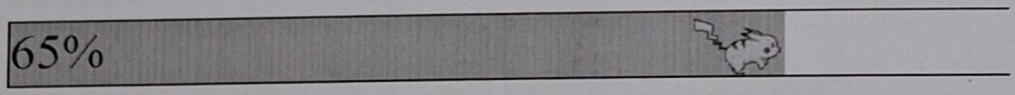
```
background-color : rgb(158, 255, 234);
border-radius: 20px;
}
    img {
        height : 100px;
        width: 100px;
        border-radius: 50%;
    }
</style>
</head>
<body>
    <div class = "container">
        <div class = "horizontal-box">
            <img src = "c:\\hbox.jpeg" alt="">
        </div>
        <div class = "square">
            <img src = "c:\\square.jpeg" alt = "">
        </div>
        <div class = "vertical-box"><div id= "bluebox">
            <img src = "c:\\vbox.jpeg" alt = "">
        </div>
        <div class = "square">
            <img src = "square.jpeg">
            </div>
        <div id = "blank"></div>
    </div>
    </div>
</body>
</html>
```


Pick One
Popularity
search
OYO Avenue
5 star rated
New Horizons
3 star rated
New Lodge
newly made

m






65%

ASSIGNMENT-3.

Q1. Write a program using Javascript to find the sum of cubes of an input five digit number.

```
<script>
    function SumCubes(num){
        if (num < 10000 || num > 99999)
            return "Please enter a 5-digit number.";

        let sum = 0;
        while(number > 0)
            {
            let digit = num%10;
            sum += Math.pow(digit, 3);
            num = Math.floor(num/10);
        }
    return sum;
}
const inputNumber = parseInt(prompt("Enter the number:"));
console.log("Sum of Cubes:", SumCubes(inputNumber));
</script>
```

OUTPUT: Enter the number: 12345

Sum of Cubes: 225.

Q2. Write a program using Javascript to reverse a given number.

```
<script>
    function Reverse(num){
        let reversed = 0;
        while(num != 0)
```

```
        {
        reversed = reversed * 10 + num % 10;
        num = Math.floor(num/10);
      }
    return reversed;
  }
const num = parseInt(prompt("Enter the number:"));
if(isNaN(num))
    {
    document.write("Invalid input.");
  }
else
  {
  document.write("Reverse = ", Reverse(num));
}
</script>
```

OUTPUT: Enter the number : 561

Reverse = 165.

Q3. Write a program using higher order functions of Javascript to generate fibonacci series.

```
<script>
    function fibo(n)
        {
        const f = [0,1];
        for(let i=2; i<n; i++)
            {
            f.push(f[i-1] + f[i-2]);
        }
      return f;
    }
```

```
const n = parseInt(prompt("Enter a number:"));
document.write("Fibonacci Series: ", fibo(n));
</script>
```

OUTPUT: Enter a number: 10

Fibonacci Series: 0,1,1,2,3,5,8,13,21,34

Q4. Write a program using Javascript to find the GCD and LCM of three input numbers.

```
<script>
    function findGCD(num1, num2)
        {
        while(num2 != 0)
            {
            let temp = num2;
            num2 = num1 % num2;
            num1 = temp;
        }
        return num1;
    }
    function findLCM(num1, num2)
        {
        return (num1 * num2) / findGCD(num1, num2);
    }
    function findGCDandLCM(num1, num2, num3)
        {
        let gcdoffirstTwo = findGCD(num1, num2);
        let gcdofALL = findGCD(gcdoffirstTwo, num3);
        let lcmoffirstTwo = findLCM(num1, num2);
        let lcmofALL = findLCM(lcmoffirstTwo, num3);
        return {
            "GCD": gcdofALL;
```

```
    "LCM": lcmOfAll
  };
}

const num1 = parseInt(prompt("Enter the first number:"));
const num2 = parseInt(prompt("Enter the second number:"));
const num3 = parseInt(prompt("Enter the third number:"));
const result = findGCDandLCM(num1, num2, num3);
document.write("GCD: ", result.GCD);
document.write("<br>LCM: ", result.LCM);
</script>
```

OUTPUT: Enter the first number: 24
Enter the second number: 36
Enter the third number: 81
GCD: 3
LCM: 648

Qs. Write a program using Javascript to perform the following:

i) Declare an array of numbers having values 1,2,3,4,5,6,7,8,9 respectively.

ii) Increment each of the array values by 15 using map function and store the results in a new array named as myarray.

iii) Reverse the numbers present in myarray.

iv) Append two elements having values 13 and 12 to the array myarray.

v) Apply reduce method to the updated myarray through operation sum = sum + num, where sum is the accumulator value and take initial accumulator value as 24.

```
<script>
  let arr = [1,2,3,4,5,6,7,8,9];
  let myArr = arr.map(num => num+15);
  myArr.reverse();
  myArr.push(9,10);
  let sum = myArr.reduce((accumulator, currentValue)
            => accumulator + currentValue, 0);
  document.write("Resulting Array: ", myArr);
  document.write("<br> Sum: ", sum);
</script>
```

OUTPUT: Resulting Array: 23,22,21,20,19,18,17,16,9,10
Sum: 196

Q6. Write a program using Javascript declarative programming style to find the factorial of a given number.

```
<script>
  function factorial(n){
      return n<=1 ? 1 : n*factorial(n-1);
  }
  const num = parseInt(prompt("Enter the number:"));
  const result = factorial(num);
  document.write("factorial of "+num+" is "
                 +result);
</script>
```

OUTPUT: Enter the number: 5
factorial of 5 is 120

Q7. Write a program using Javascript declarative programming style to check whether a given number is a prime number or not.

```
<script>
    function isPrime(n) {
        if(n<=1){
            return false;
        }
    for(let i=2; i<=Math.sqrt(n); i++)
            {
            if(n%i==0)
                {
                return false;
            }
        }
    return true;
}
const num=parseInt(prompt("Enter a number:"));
if(isNaN(num))
            {
        document.write("Invalid input. Please enter
                    a valid number.");
    }
    else{
        document.write(num, " is prime => ", isPrime(
                                        num));
    }
</script>
```

OUTPUT: Enter a number: 13

13 is prime => true

Q8. Write a program using Javascript for each loop to print the values of an array color = ["red", "green", "blue", "purple"]

```
<script>
    const colors = ["red","green","blue","purple"];
    for(const color of colors)
        {
            document.write(color," ");
    }
</script>
```

OUTPUT: red green blue purple

99. Write a program using Javascript declarative style of programming to retrieve the animal's name where the species is dog.

```
var animals = [
    {name: 'fuffykins', species: 'rabbit'},
    {name: 'carrot', species: 'dog'},
    {name: 'Hamilton', species: 'dog'},
    {name: 'Harold', species: 'fish'},
    {name: 'Ursula', species: 'cat'},
    {name: 'Jimmy', species: 'fish'},
]
```

```
<script>
    const animals = [
    {name: 'fuffykins', species: 'rabbit'},
    {name: 'carrot', species: 'dog'},
    {name: 'Hamilton', species: 'dog'},
    {name: 'Harold', species: 'fish'},
    {name: 'Ursula', species: 'cat'},
    {name: 'Jimmy', species: 'fish'},
    ];
    const dogNames = animals
```

```
    .filter(animal => animal.species === 'dog')
    .map(dog => dog.name);
document.write("Dog names: ", " ", dogNames);
</script>
```

OUTPUT: Dog names: carro, Hamilton.

Q10. Write a program using Javascript Arrow function to estimate the volume of a sphere having radius 5.5 meters.

```
<script>
  const calculateSphereVolume = (radius) => {
    return (4/3) * Math.PI * Math.pow(radius, 3);
  };
  const radius = 5.5;
  document.write("Volume of the sphere with
                  radius ", radius, " meters is ",
    volume.toFixed(2), " cubic meters.");
</script>
```

OUTPUT: Volume of the sphere with radius 5.5 meters is 696.91 cubic meters.

Q11. Write a program using Javascript iterative programming style to check whether a given number is a strong number or not.

```
<script>
    function factorial(num){
    if(num==0 || num===1)
        return 1;
    let result = 1;
    for(let i=2; i<=num; i++)
```

```
        {
            result *= i;
        }
        return result;
    }
    function isStrongNumber(number)
        {
            let num1 = number;
            let sum = 0;
            while(number > 0)
                {
                let digit = number % 10;
                sum += factorial(digit);
                number = Math.floor(number/10);
            }
            return sum === num1;
    }
const num = parseInt(prompt("Enter the number:"));
document.write(num, " is a Strong Number => ",
                    isStrongNumber(num));
</script>
```

OUTPUT: Enter the number: 145
145 is a Strong Number => true.

Q12. Write a program using Javascript recursive style of programming to check whether a given number is a perfect number or not.

```
<script>
    function findDivisorSum(number, divisor = number-1,
                                sum = 0) {
    if (divisor >= number ) {
        return sum;
    }
```

```
    if (number % divisor == 0)
        {
        sum += divisor;
    }
    return findDivisorSum(number, divisor+1, sum);
}
function isPerfectNumber(n)
        {
        const sum = findDivisorSum(n);
        return sum == n;
    }
const numbers = 28;
const result = isPerfectNumber(numbers);
console.log(numbers + " is a perfect number.",
                                result);
</script>
```

OUTPUT : 28 is a perfect number.

Q13. Write a program using Javascript higher order function to check whether a given number is an armstrong number or not.

```
<script>
const power = (base, exponent) => Math.pow(
                    base, exponent);
const countdigits = number => String(number).length;
const sumofPoweredDigits = (number, numDigits) => {
    let sum = 0;
    let temp = number;
    while (temp > 0)
        {
        const digit = temp % 10;
        sum += power(digit, numDigits);
        temp = Math.floor(temp / 10);
```

```
    }
    return sum;
};
const isArmstrongNumber = number => {
    const numDigits = countDigit(number);
    const sum = sumOfPoweredDigits(number, numDigits);
    return sum === number;
};
const number = 153;
console.log(number + " is an Armstrong Number => ",
                    isArmstrongNumber(number));
```

OUTPUT : 153 is an Armstrong Number => true.

Q14. Write a program to find maximum element.

```
<script>
    function maximum(arr){
        if(arr===0)
            {
            return undefined;
        }
        return Math.max(...arr);
    }
    arr = [6,78,98,45];
    console.log(maximum(arr));
</script>
```

OUTPUT : 98.

Q15. Write a program using Javascript to concatenate two given string.

```
    function concateString(str1, str2)
        {
        return str1.concat(str2);
```

```
}
let str1 = "hello";
let str2 = "world";
console.log("Concate string is ", concateString(str1,
                                                str2));
```

OUTPUT: Concate string is helloworld.

Q16 Write a program using Javascript to check whether any two random number is between 50 to 150 are prime or not using break and continue.

```
<script>
    function factorial(num)
    {
function isPrime(num)
    {
    if(num <= 1) return false;
    if (num <= 3) return true;
    if(num % 2 == 0 || num % 3 == = 0) return false;
for(let i = 5; i * i <= num; i += 6)
    {
    if(num % i == = 0 || num % (i+2) == = 0) return
                                            false;
}
return true;
}
function checkPrimeNumbers() {
    for(let i=0; i<2; i++)
    {
    let randomNumber = Math.floor(Math.random()
                        *(150-50+1)) + 50;
    if(!isPrime(randomNumber))
    {
```

```
        console.log(`${randomNumber} is not a prime number.`);
        continue;
    }
    console.log(`${randomNumber} is a prime number.`);
  }
}
checkPrimeNumbers();
```

OUTPUT: 97 is a prime number
143 is not a prime number.

Q17 Write a program using Javascript to print the grade of a student.

```
function calcGrade(marks)
{
let grade;
if (marks >= 90){
    grade = "O grade";
}
else if (marks >= 80)
    {
    grade = "E grade";
}
else if (marks >= 70)
    {
    grade = "A grade";
}
elseif (marks >= 60)
    grade = "B grade";
elseif (marks >= 50)
    grade = "C grade";
elseif (marks >= 35)
    grade = "D grade";
```

```
    else grade = "F grade";
    return grade;
}
const marks = 83;
console.log(calcgrade(marks));
```

OUTPUT : E grade

8/4/24