

Assignment - 2

1. `<!DOCTYPE html>` ~~Play with this~~ `<html lang="en">`

`<head>`

`<title> Document </title>`

`<style>`

~~#computer, #mobilemenu~~

~~display: flex; padding: 10px;~~

~~flex: 1 1 auto;~~

~~box-shadow: 1px 1px 10px;~~

~~gap: 20px;~~

2 ~~#computer & display: none;~~ 3

Section 8

~~display: flex;~~

~~flex-wrap: wrap;~~

~~gap: 20px;~~

~~margin: 20px;~~

~~height: 100%;~~

~~width: 100%;~~

3

~~h2 &~~

~~padding: 0px;~~

~~margin: 0px;~~

3

~~div &~~

~~box-shadow: 2px 1px 10px;~~

3

~~#hidden & display: none;~~

~~flex-direction: column;~~

Name: \_\_\_\_\_

Regd. Number: \_\_\_\_\_

```
#mobilemenu
display: block;
3
@ media only screen and (min-width: 500px) {
#mobilemenu {
display: none;
}
3
#computer {
display: none;
}
3
.showit {
display: block;
}
3
#computer {
display: flex;
padding: 20px;
flex: 1fr 2fr 3fr;
box-shadow: 1px 1px 20px;
gap: 20px;
}
3
@ media only screen and (min-width: 800px) {
#mobilemenu {
display: none;
}
3
#computer {
display: block;
}
3
</style>
<(head)>
<body>
<nav id="computer">
```

```
<h2> book with us </h2>
<select name = "sort" id = "sort">
    <option value = "popularity"> popularity </option>
    <option value = "cost"> cost </option>
    <option value = "rating"> rating </option>
</select>
<input type = "text">
<input type = "button" value = "search">
</nav> </div> </nav>
<section>
    <div>
        <img src = "Image1.jpeg" alt = "First Image">
        <h2> OYO Avenue </h2>
        <p> 5-star rated </p> </div> </div>
        <img src = "Image2.jpeg" alt = "Second Image">
        <h2> New Lodge Lite </h2>
        <p> newly made </p>
    </div>
    <div>
        <img src = "Image3.jpeg" alt = "Third Image">
        <h2> Old Day Lodge </h2>
        <p> with swimming pool </p>
    </div>
    <div>
        <h3> Bookings </h3>
        <script>

```

```
function dropdown () {  
    if (document.getElementById('hidden').style.display  
        = 'none')  
        document.getElementById('hidden').style.display  
        = 'block';  
    else  
        document.getElementById('hidden').style.display  
        = 'none';  
settimeout(() => {document.getElementById('hidden')  
    .style.display = 'none' }, 5000);  
}  
2.
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<title> Document </title> </head>  
<style>  
    .container {  
        display: flex;  
        justify-content: space-between;  
        flex-direction: column;  
        height: 100vh;  
        width: 100%;  
    }  
3
```

• horizontal - box 6

flex: 1;

display: flex;

align-items: center;

justify-content: center;

background-color: #f0f0f0;

border-radius: 20px;

padding: 20px;

3

• vertical - box 6

flex: 1;

height: 100%;

display: flex;

flex-direction: row;

3

#blue box 6

background-color: cyan;

height: 100%;

border-radius: 20px;

flex: 1;

padding: 20px;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

3

```
html { margin: 0; padding: 0; }
body { margin: 0; padding: 0; font-family: sans-serif; }

#blank & #Pic1and3 {
    position: absolute;
    top: 50px;
    left: 50px;
    width: 100px;
    height: 100px;
    border-radius: 50px;
    background-color: #f0f0f0;
    border: 1px solid black;
}

#square & #Pic2 {
    position: absolute;
    top: 150px;
    left: 150px;
    width: 100px;
    height: 100px;
    border-radius: 10px;
    background-color: #f0f0f0;
    border: 1px solid black;
}

img & #horizontal {
    position: absolute;
    top: 250px;
    left: 250px;
    width: 100px;
    height: 100px;
    border-radius: 50px;
    border: 1px solid black;
}

#vertical {
    position: absolute;
    top: 350px;
    left: 350px;
    width: 100px;
    height: 100px;
    border-radius: 10px;
    background-color: #f0f0f0;
    border: 1px solid black;
}

#body {
    position: absolute;
    top: 450px;
    left: 450px;
    width: 100px;
    height: 100px;
    border-radius: 10px;
    background-color: #f0f0f0;
    border: 1px solid black;
}

<html>
    <head>
        <title> CSS Grid Layout Example </title>
    </head>
    <body>
        <div class="container">
            <div class="horizontal-box">
                
                <div class="square"></div>
                
                <div>
                    
                    <div id="bluebox">
                        
                        <div class="square"></div>
                    </div>
                </div>
                
                <div id="blank"></div>
                <div></div>
                <div></div>
            </div>
        </body>
    </html>

```

3    `<!DOCTYPE html>`

`<html lang="en"> <head>`

`<meta charset="UTF-8">`

`<meta name="viewport">`

`<title> Animation </title>`

`<style>`

`• container {`

`position: relative;`

`width: 300px;`

`height: 300px;`

`border: 1px solid black;`

`overflow: hidden;`

3

`• box {`

`position: absolute;`

`width: 50px;`

`height: 50px;`

`background-color: blue;`

3

`• box1 {`

`animation: moveTopRight 2s forwards;`

3

`• box2 {`

~~`animation: moveBottomLeft 2s forwards;`~~

3

`• box3 {`

`animation: moveVertical 2s forwards;`

3

Name: \_\_\_\_\_

Regd. Number: \_\_\_\_\_

• box4 {  
animation: moveTopRight 2s forwards;

3 • circle {  
position: absolute;  
width: 20px;  
height: 20px;  
border-radius: 50%;  
background-color: red;  
transition: all 2.5s ease-in-out;

3 @keyframes moveTopRight {

0% { top: 0; left: 0; }

100% { top: 0; left: 250px; }

3 @keyframes moveBottomLeft {

0% { top: 250px; left: 250px; }

100% { top: 250px; left: 0; }

3 @keyframes moveVertical {

0% { top: 50px; left: 0; }

100% { top: 250px; left: 0; }

3 @keyframes moveTopRight2 {

0% { top: 250px; left: 250px; }

100% { top: 0; left: 250px; }

② Keyframes move(circle {

0% { top: 0; left: 0; }

25% { top: 100px; left: 100px; }

50% { top: 200px; left: 200px; }

75% { top: 150px; left: 50px; }

100% { top: 0; left: 250px; }

3

<style> </head>

<body> <div class = "container">

<div class = "box box1"> </div>

<div class = "box box2"> </div>

<div class = "box box3"> </div>

<div class = "box box4"> </div>

<div class = "circle"> </div>

</div>

<script>

function getRandomPosition () {

var x = Math.floor(Math.random() \* 300);

var y = Math.floor(Math.random() \* 300);

return [x, y];

3 var circles = document.querySelectorAll('.circle');

setInterval(function () {

```
var xy = getRandomPosition();  
circle.style.top = xy[0] + 'px';  
circle.style.left = xy[1] + 'px';  
3; 2000);  
</script> </body> </html>
```

4. 

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<title> Document </title>  
<head>  
<style>  
#bar {  
width: 100px;  
border: 2px solid black;  
height: 50px;  
overflow: hidden;  
position: absolute;  
padding: 0%;}  
#pill {  
animation: fillit 1s ease-in-out forwards;  
height: 50px;  
width: 100px; }  
#pill {  
height: 50px; float: inline-end;  
padding: 0%;}
```

Name: \_\_\_\_\_

Regd. Number: \_\_\_\_\_

② Keyframes (Fill it &

0% {

width: 0; background-color: blue; }

100%:

{ width: 100%;

background: yellow; }

}

#fill { position: fixed;

margin: 0%;

font-size: 40px;

</style> </head>

<body>

<div id = "bar"><div id = "fill"> <p id = "para">0</p><img src = "https://media.tenor.com/ut1Cf9gCxYFABF.gif" alt = ""> </div></div>

<script>

var i = 0;

function countUP() {

if (i <= 100) {

document.getElementById('para').innerHTML

= i++ + "%";

}

}

setInterval(countUP, 100);

</script> </body> </html>

5. CSS Flexbox is a layout model in CSS for creating flexible and responsive arrangements of elements within a container. It simplifies alignment and distribution of space among them, making it ideal for dynamic layouts across various screen sizes than container properties.

display - define a Flex container.

Flex-direction! - specifies direction of the

Flex item.

justify-content: aligns Flex items along the main axis.

align-items: - align Flex items along the cross axis.

Flex-wrap: - controls whether Flex items wrap out to multiple lines.

Flex-item properties:- order: - sets the order of appearance for Flex item.

Flex-grow: - determines the ability of Flex items to grow.

Flex-shrink: - controls the ability of Flex-items to shrink.

Flex-basis: - sets the initial size of Flex items align-self / override alignment set by the container for individual Flex items.