STUDENT NAME	**R.ROJA**	
STUDENT REGISTRATION NUMBER	**251U1R2064**	**CLASS: AIML-IC**
PROGRAM	**UG**	**YEAR and TERM: 1st year & 1st term**
SUBJECT NAME	**HTML**	
NAME OF THE ASSESSMENT	**Reflective lab journal-7**	
DATE OF SUBMISSION	**9.11.25**	

Write a program in css3 colors:

```
<html>
<head>
<style type=”text/css”>
div {
width: 200px;
height: 100px;
background-color: rgba(0, 128, 255, 0.4);
border: 2px solid black;
color: rgba(0, 0, 0, 0.8);
text-align: center;
line-height: 100px;
}
</style>
</head>
<body>
<div>RGBA Example</div>
</body>
</html>
```

Output:

Explanation:

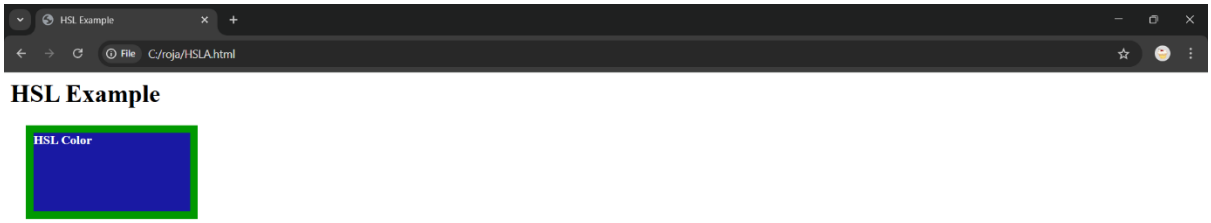
1. This program demonstrates how to apply different **CSS3 colour formats** to web elements.
2. The <style> tag in the <head> section defines colour styles for the body, headings, and paragraphs.
3. It uses **named colours** (like light blue), **RGB values**, **HEX codes**, and **HSL** formats.
4. These colour systems give designers flexibility in choosing and adjusting shades precisely.
5. CSS3 expanded colour support, allowing for **transparency, gradients, and advanced colour effects**.

Write a program on HSL colour:

```
<html>
<head>
<title>HSL Example</title>
<style type="text/css">
.box {
width: 200px;
height: 100px;
margin: 20px;
color: white;
font-weight: bold;
}
.myhsl {
background-color: hsla(240, 100%, 30%, 0.9);
border: 10px solid hsl(120, 100%, 30%);
}
</style>
</head>
```

```
<body>
<h1>HSL Example</h1>
<div class="box myhsl">HSL Color</div>
</body>
</html>
```

Output:



Explanation:

1. This program shows how to use HSL (Hue, Saturation, Lightness) colours in CSS.

2. Hue defines the type of colour (0°–360° on the colour wheel), saturation controls intensity, and lightness adjusts brightness.

3. Each HTML element (like <h1> or <p>) is styled with different HSL values.

4. HSL makes it easier to create lighter or darker shades of a colour by adjusting the lightness value.

5. This method gives designers more control and flexibility than RGB or HEX colour formats.

Write a program on Gradient colours:

```
<html>
<head>
<title>Gradient Examples</title>
<style type=”text/css”>
.box {
width: 250px;
height: 120px;
```

```
margin: 20px;
color: white;
font-weight: bold;
}
.linear {
background: linear-gradient(to right, red, yellow);
}
.radial {
background: radial-gradient(circle, purple, pink);
}
</style>
</head>
<body>
<h2>CSS3 Gradient Examples</h2>
<div class="box linear">Linear Gradient</div>
<div class="box radial">Radial Gradient</div>
</body>
</html>
```

Output:

Explanation:

1. Gradient colours in CSS3 create a smooth transition between two or more colours.

2. The property linear-gradient() makes colours blend in a straight line (e.g., left to right or top to bottom).

3. The property radial-gradient() creates a circular colour blend from the center outward.

4. Gradients don't need image files — they are generated directly with CSS.

5. CSS3 gradients make web pages look more attractive and modern by adding colourful backgrounds or effects.

Write a program on shadows:

```
<html>
<head>
<style type="text/css">
h1 {
text-align: center;
font-size: 50px;
color: #333;
text-shadow: 10px 10px 10px rgba(0,0,0,0.5);
}
div{
background-color:green;
box-shadow: 3px 3px 1px rgba(172, 9, 9, 0.5);
}
</style>
</head>
<body>
<div>
<h1>Text Shadow Example</h1>
</div>
</body>
</html>
```

Output:

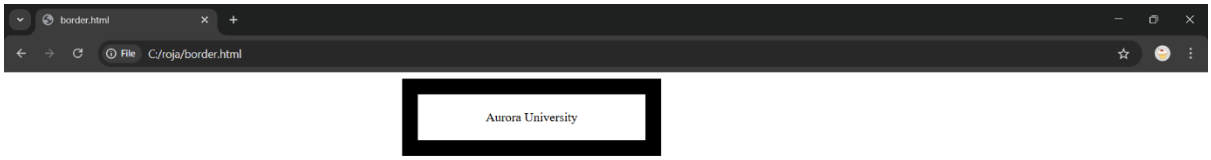
Explanation:

1. This program shows how to create shadows for text and boxes using CSS3.
2. The property text-shadow adds shadow effects behind text (here, gray behind the heading).
3. The property box-shadow creates shadows around elements like div boxes.
4. You can adjust horizontal offset, vertical offset, blur radius, and colour for different shadow styles.
5. CSS3 shadows make web elements look 3D and visually appealing without using images.

Write a program on border:

```
<html>
<head>
<style>
.myborder {
border: 20px solid;
border-image: url('./images/border.png') 30 round;
padding: 20px;
text-align: center;
width: 250px;
margin-left: 500px;
}
</style>
</head>
<body>
<div class="myborder">
Aurora University
</div>
</body>
</html>
```

Output:



Explanation:

1. This program demonstrates how to style elements using CSS border properties.

2. The border property controls the thickness, style, and color of the border.

3. Common border styles include solid, dotted, and dashed.

4. The border-radius property is used to round the corners of borders.

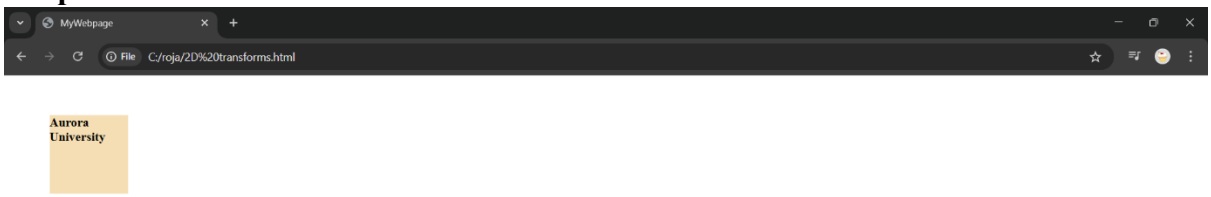
5. CSS3 borders make page elements stand out and look more attractive visually.

Write a program on 2D transforms:

```
<html>
<head>
<title>MyWebpage</title>
<style type="text/css">
.box {
width: 100px;
height: 100px;
background-color: wheat;
margin: 50px;
transition: transform 0.5s;
}
.box:hover {
transform: rotate(45deg) scale(1.5) translate(50px, 0);
}
</style>
</head>
<body>
```

```
<div class="box">
<h4>Aurora University</h4>
</div>
</body>
</html>
```

Output:



Explanation:

1. 2D transforms in CSS3 allow you to move, rotate, scale, or skew elements on a flat (2D) plane.
2. The transform property is used to apply these effects to HTML elements.
3. Common 2D functions include translate() (move), rotate() (turn), scale() (resize), and skew() (slant).
4. In this program, hovering the mouse rotates, enlarges, and shifts the box.
5. 2D transforms make web pages interactive and visually dynamic without needing JavaScript.

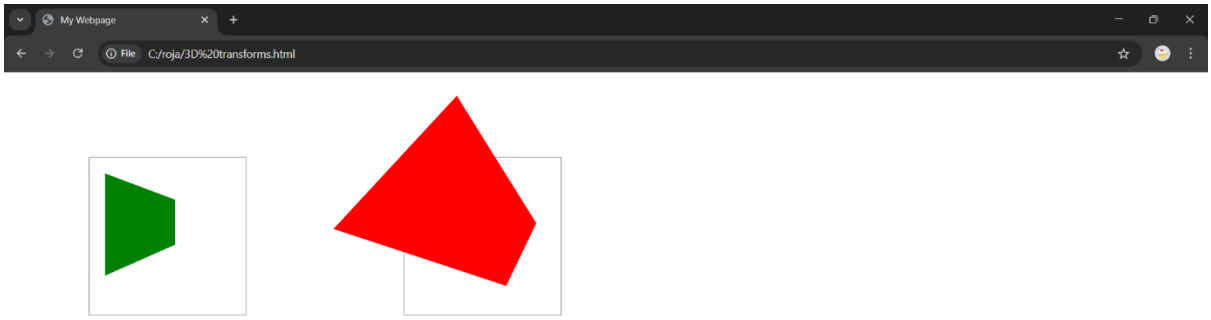
Write a program on 3D transfoms:

```
<html>
<head>
<title>My Webpage</title>
<style type="text/css">
#main1 {
width:200px;
height:200px;
border:1px solid grey;
perspective:100px;
```

```
perspective-origin:40% 40%;
margin:100px;
float:left;
}
#main2 {
width:200px;
height:200px;
border:1px solid grey;
margin:100px;
float:left;
perspective:100px;
}
#main1 #D1 {
width:180px;
height:130px;
margin:20px;
background-color: green;
transform:rotateY(45deg);
transform-origin:left;
}
#main2 #D2 {
width:180px;
height:130px;
margin:20px;
background-color: red;
transform:rotate3d(0,1,1,45deg);
}
</style>
</head>
<body>

<div id="main1">
<div id="D1"></div>
</div>
<div id="main2">
<div id="D2"></div>
</div>
</body>
</html>
```

Output:



Output:

1. 3D transforms in CSS3 allow elements to be rotated, moved, or scaled 3D in three-dimensional space (X, Y, and Z axes).

2. The transform property with functions like rotateX(), rotateY(), and translateZ() is used to create depth effects.

3. The perspective property gives a sense of distance and depth, making the element appear truly 3D.

4. In this example, hovering over the box makes it rotate on both X and Y axes, creating a 3D flip effect.

5. CSS3 3D transforms make web elements look realistic, dynamic, and engaging without using JavaScript or 3D graphics tools.

Write a program on transitions:

```
<html>
<head>
<title>My Webpage</title>
<style type="text/css">
#main {
width:800px;
height:300px;
border:1px solid blue;
background-color: yellow;
}
#D1 {
width:20px;
height:20px;
float:left;
margin:100px;
background-color: lightskyblue;
```

```
transition:width 2s,heights 2s,ease 2s;
}
#D1:hover
{
background-color: red;
width:200px;
height:200px;
}
#D2 {
width:20px;
height:20px;
float:left;
margin: 100px;
background-color: green;
transition:linear 3s;
transition-delay: 1s;
}
#D2:hover
{
background-color: blue;
width:200px;
height:200px;
}

#D3
{
width:20px;
height:20px;
float:left;
margin:100px;
background-color: maroon;
transition-timing-function: linear;
transition-duration: 3s;
transition-delay: 1s;
}
#D3:hover
{
background-color: orange;
width:200px;
height:200px;
}
</style>
</head>
<body>
<div id="main">
<div id="D1"></div>
<div id="D2"></div>
<div id="D3"></div>
```

```
</div>
</body>
</html>
```

Output:

Explanation:

1. CSS3 transitions allow smooth, animated changes between different property values over a set time.

2. The transition property defines which properties to animate, how long, and what speed curve to use.

3. In this program, when you hover over the box, its color, size, and shape change smoothly.

4. The ease keyword makes the transition start slow, speed up, then slow down again for a natural effect.

5. Transitions make web pages look interactive and visually appealing without using JavaScript or extra animation tools.1.

Write a program on animations:

```
<html>
<head>
<title>mywebpage</title>
<style type="text/css">
.box {
width: 100px;
height: 100px;
background: coral;
position: relative;
animation-name: moveBox;
animation-duration: 4s;
```

```
animation-timing-function: ease-in-out;
animation-delay: 1s;
animation-iteration-count: infinite;
animation-direction: alternate;
}
@keyframes moveBox {
0% { left: 0; background: coral; }
100% { left: 300px; background: lightgreen; }
}
</style>
</head>
<body>
<div class="box">
<h3 align="center">Aurora University</h3>
</div>
</body>
</html>
```

Output:

Explanation:

1. CSS3 animations let you create moving and changing effects without using JavaScript.

2. The @keyframes rule defines the steps or stages of the animation (like start, middle, end).

3. The animation property applies the animation name, duration, and behavior to an element.

4. In this program, the box moves sideways and changes colour repeatedly using the move Box animation.
5. CSS3 animations make web pages dynamic, engaging, and visually interactive with smooth motion effects.