CSS ANIMATION :

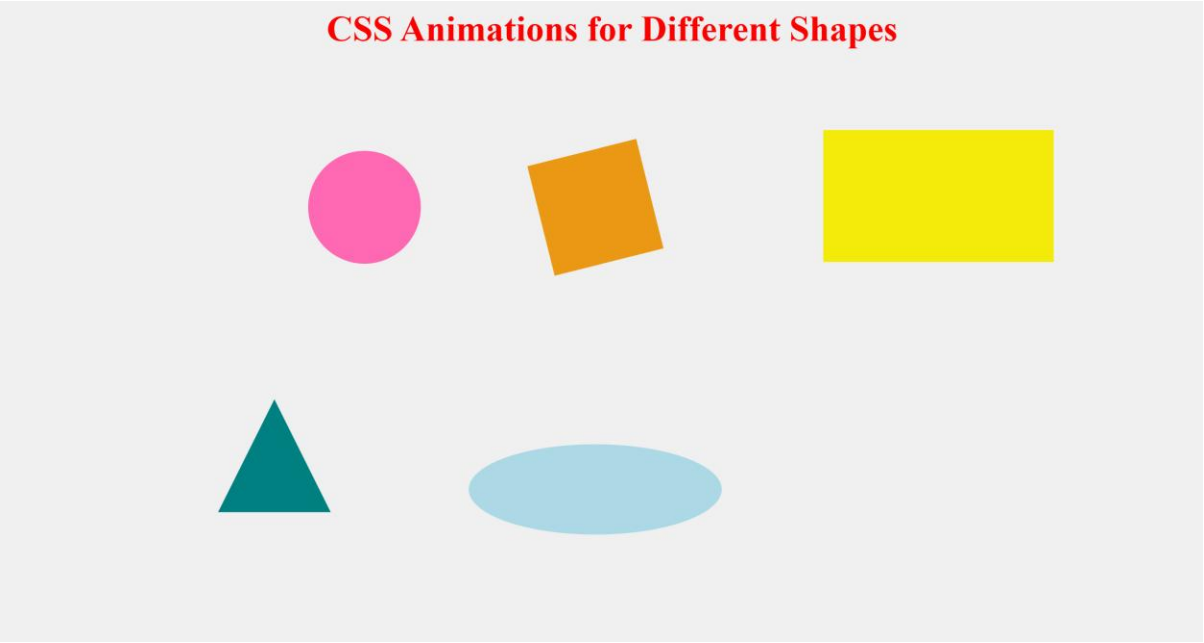
```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Animatons - Shapes</title>
    <style>
        body{
            background-color: #f0f0f0;
        }
        h1{
            text-align: center;
            color: red;
        }
        .container{ displ
            ay: grid;
            grid-template-columns: repeat(3, 1fr);
            gap: 60px;
            justify-items: center;
            margin: 40px auto;
            max-width: 800px;
        }
        .shapes{
            width: 100px;
            height: 100px;

            margin: 50px;
        }
```

```
.circle{
    width: 100px;
    height: 100px;
    border-radius: 100px;
    animation: circleMove 2s infinite;
}
@keyframes circleMove{
    0% { transform: translateX(0); background: red; }
    50% { transform: translateX(80px); background: hotpink; }
    100% { transform: translateX(0); background: green; }
}
.square{
    animation: squareRotate 3s infinite;
}
@keyframes squareRotate{
    0% { transform: rotate(0deg); background: blue; }
    50% { transform: rotate(180deg); background: orange; }
    100% { transform: rotate(360deg); background: blue; }
}
.rectangle{ widt
    h: 140px;
    height: 80px;
    background-color: purple;  animation:
    rectangleScale 3s infinite;
}
@keyframes rectangleScale{
    0% { transform: scale(1); background: purple; }
    50% { transform: scale(1.5); background: yellow; }
    100% { transform: scale(1); background: purple; }
```

```
        }
        .triangle{
            width: 0;
            height: 0;
            border-left: 50px solid transparent;
            border-right: 50px solid transparent;
            border-bottom: 100px solid teal;
            animation: triangleJump 2s infinite;
        }
        @keyframes triangleJump{
            0%, 100% { transform: translateY(0); }
            50% { transform: translateY(-40px); }
        }
        .oval{
            width: 150px;
            height: 80px;
            background-color: coral;
            border-radius: 75px / 40px;
            animation: ovalStretch 2s infinite;
        }
        @keyframes ovalStretch{
            0%, 100% { transform: scaleX(1); background: coral; }
            50% { transform: scaleX(1.5); background: lightblue; }
        }
    </style>
</head>
<body>
    <h1>CSS Animations for Different Shapes</h1>
    <div class="container">
```

```
        <div class="shapes circle"></div>
        <div class="shapes square"></div>
        <div class="shapes rectangle"></div>
        <div class="shapes triangle"></div>
        <div class="shapes oval"></div>
    </div>
</body>
</html>
```

OUTPUT :



CONCLUSION :

This project demonstrates the use of CSS animations to create and animate different geometric shapes such as circle, square, rectangle, triangle, and oval. By applying keyframe animations along with properties like transform, scale, rotate, translate, and background-color, each shape was given unique motion and visual effects.

The case study highlights the power of CSS without JavaScript in creating engaging, interactive, and visually appealing designs. It shows how simple shapes can be transformed dynamically, making them useful for web graphics, educational tools, UI elements, and creative animations