INTERNSHIP REPORT

WEEK 3 DAY 3

Submitted to:	**Ali Hyder**	Submission Date:	**10TH July, 2025**
Internship Domain:	**Front Development**	Internship Name:	**ProSensia**
Student Name:	**Yasal Qamar**	Roll No.	**S25031**

CSS Transitions and Animations

Objective

The objective of this task was to explore and implement CSS transitions and animations to enhance the interactivity and visual appeal of user interface elements. This exercise aimed to deepen understanding of how **CSS can control animation timing, movement,** and **transitions in modern web development.**

Introduction:

CSS provides two main techniques to add motion to web elements: **transitions** and **animations**.

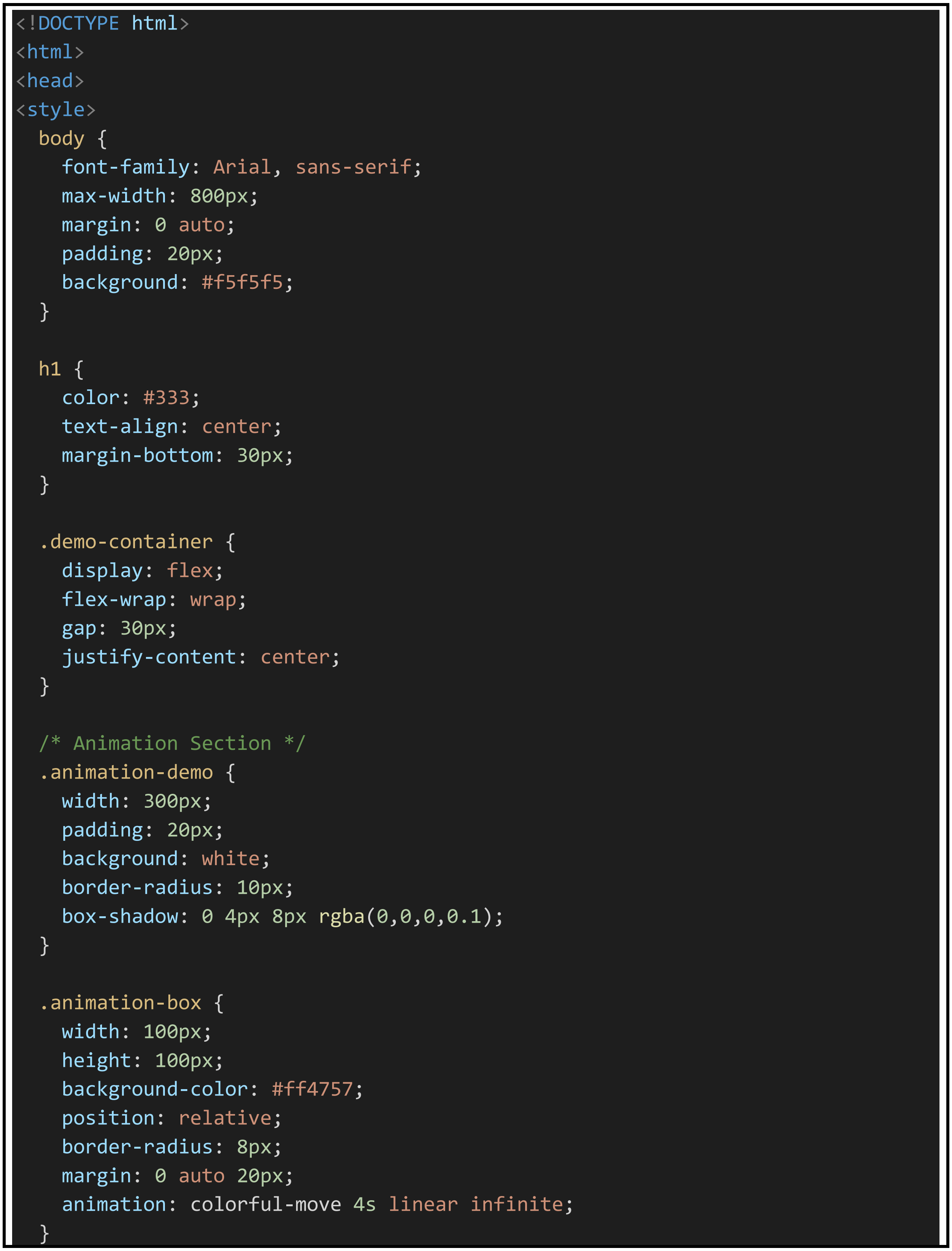
- **Transitions** allow elements to change values over a specified duration when triggered by a user interaction, such as hovering.
- **Animations**, on the other hand, use **@keyframes** to define intermediate steps in a style change and can run automatically and infinitely.

These tools are essential for creating visually engaging, responsive, and modern web designs.

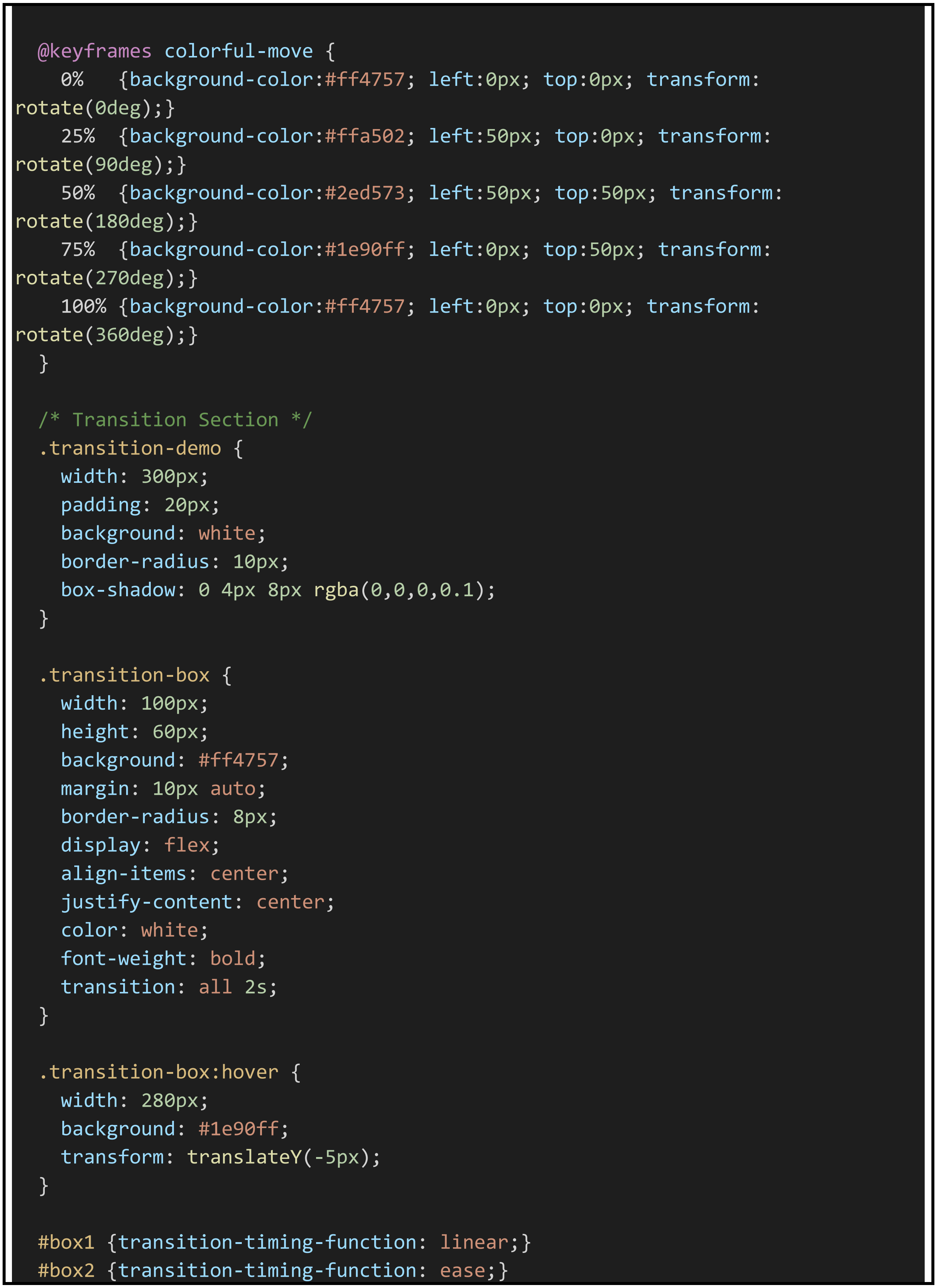
Key Learning:

- How to use **@keyframes** to create complex animations.
- Difference between animation and transition in CSS.
- Use of different timing functions like ease, linear, ease-in-out, and cubic-bezier.
- Practical implementation of hover-based transition effects.
- Creating reusable, styled components using CSS Flexbox and box-shadow.

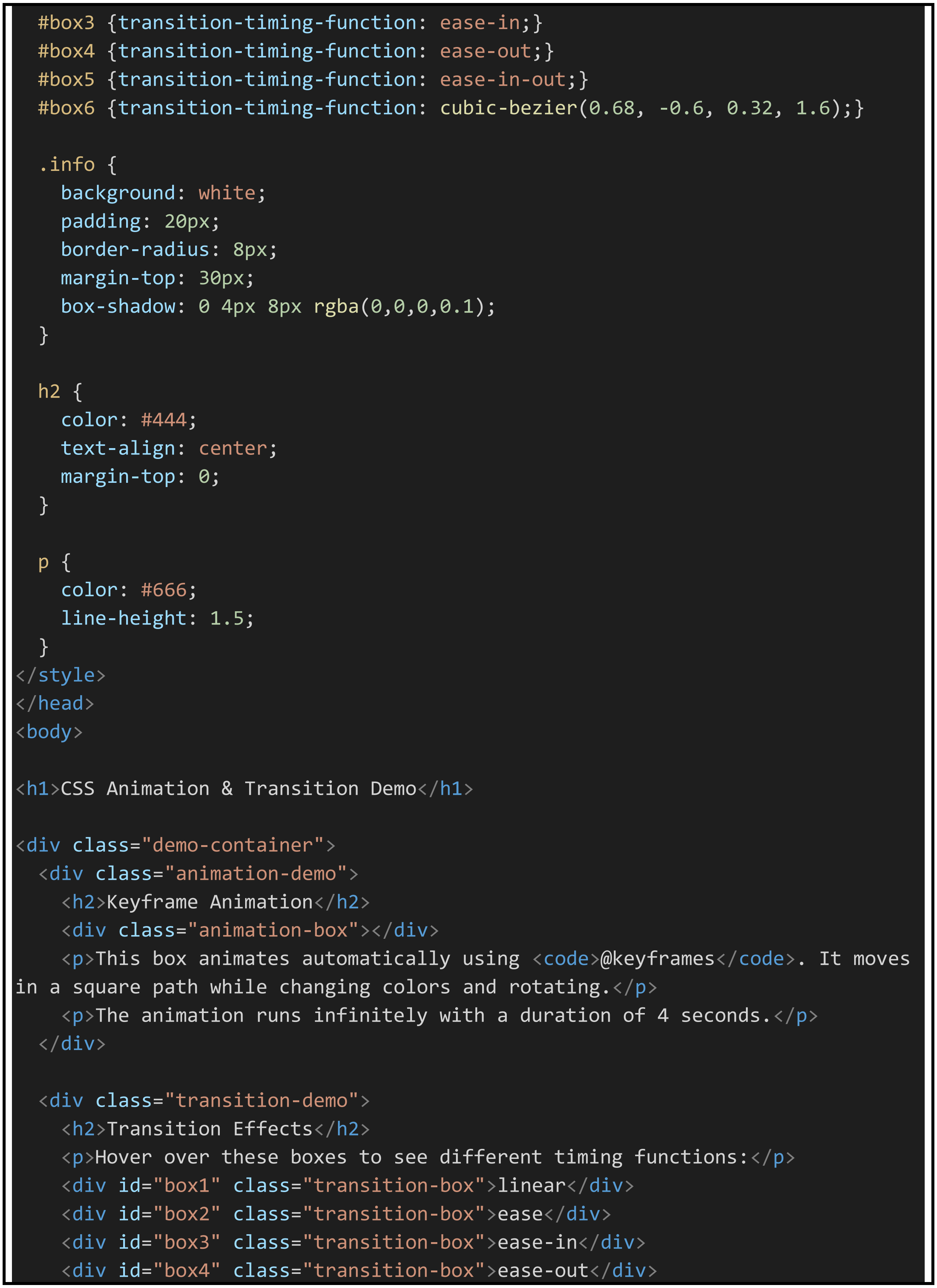
Coding:



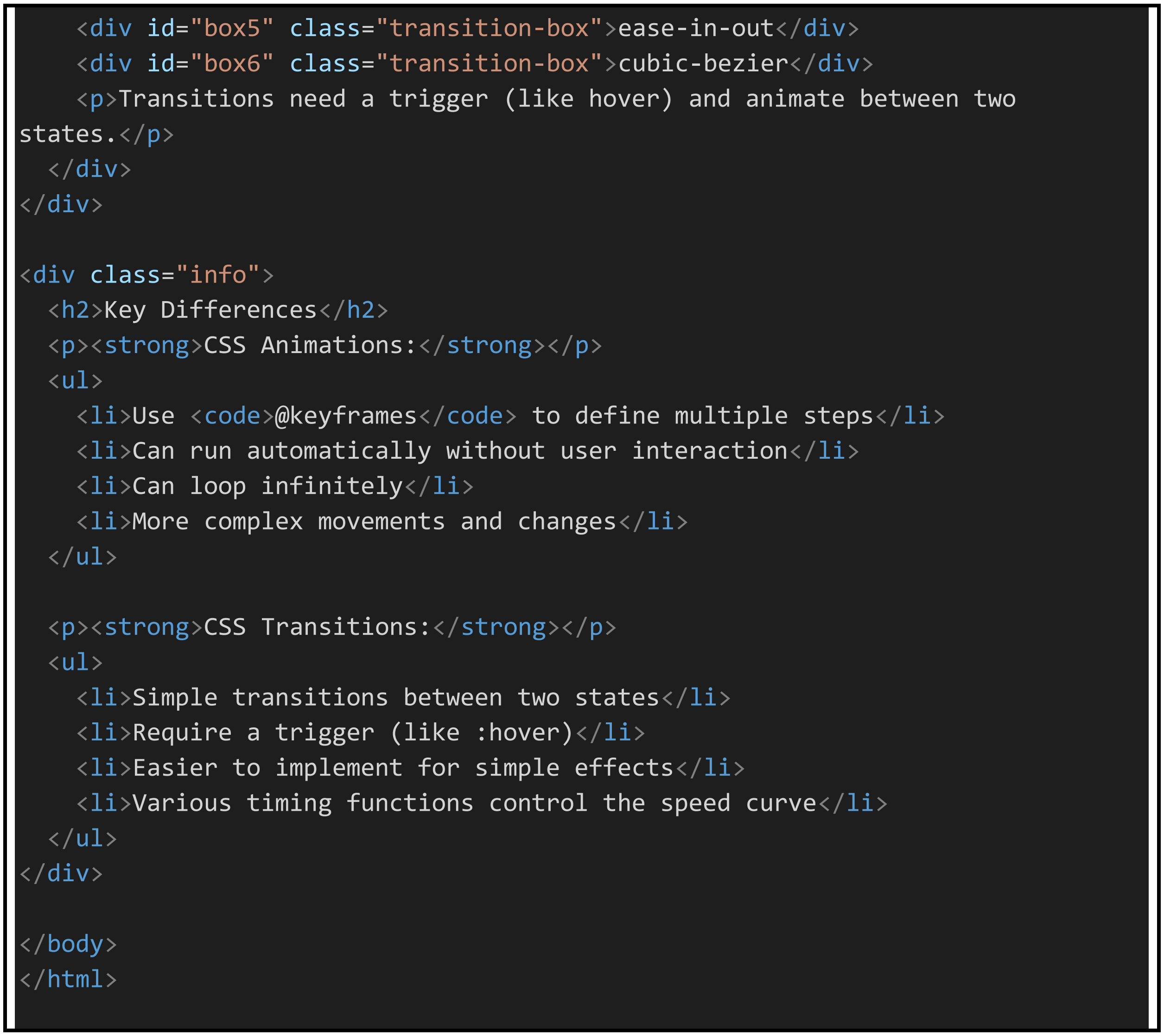
```
<!DOCTYPE html>
<html>
<head>
<style>
  body {
    font-family: Arial, sans-serif;
    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
    background: #f5f5f5;
  }

  h1 {
    color: #333;
    text-align: center;
    margin-bottom: 30px;
  }

  .demo-container {
    display: flex;
    flex-wrap: wrap;
    gap: 30px;
    justify-content: center;
  }

  /* Animation Section */
  .animation-demo {
    width: 300px;
    padding: 20px;
    background: white;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0,0,0,0.1);
  }

  .animation-box {
    width: 100px;
    height: 100px;
    background-color: #ff4757;
    position: relative;
    border-radius: 8px;
    margin: 0 auto 20px;
    animation: colorful-move 4s linear infinite;
  }
```



```css
  @keyframes colorful-move {
    0%   {background-color:#ff4757; left:0px; top:0px; transform:
rotate(0deg);}
    25%  {background-color:#ffa502; left:50px; top:0px; transform:
rotate(90deg);}
    50%  {background-color:#2ed573; left:50px; top:50px; transform:
rotate(180deg);}
    75%  {background-color:#1e90ff; left:0px; top:50px; transform:
rotate(270deg);}
    100% {background-color:#ff4757; left:0px; top:0px; transform:
rotate(360deg);}
  }

  /* Transition Section */
  .transition-demo {
    width: 300px;
    padding: 20px;
    background: white;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0,0,0,0.1);
  }

  .transition-box {
    width: 100px;
    height: 60px;
    background: #ff4757;
    margin: 10px auto;
    border-radius: 8px;
    display: flex;
    align-items: center;
    justify-content: center;
    color: white;
    font-weight: bold;
    transition: all 2s;
  }

  .transition-box:hover {
    width: 280px;
    background: #1e90ff;
    transform: translateY(-5px);
  }

  #box1 {transition-timing-function: linear;}
  #box2 {transition-timing-function: ease;}
```

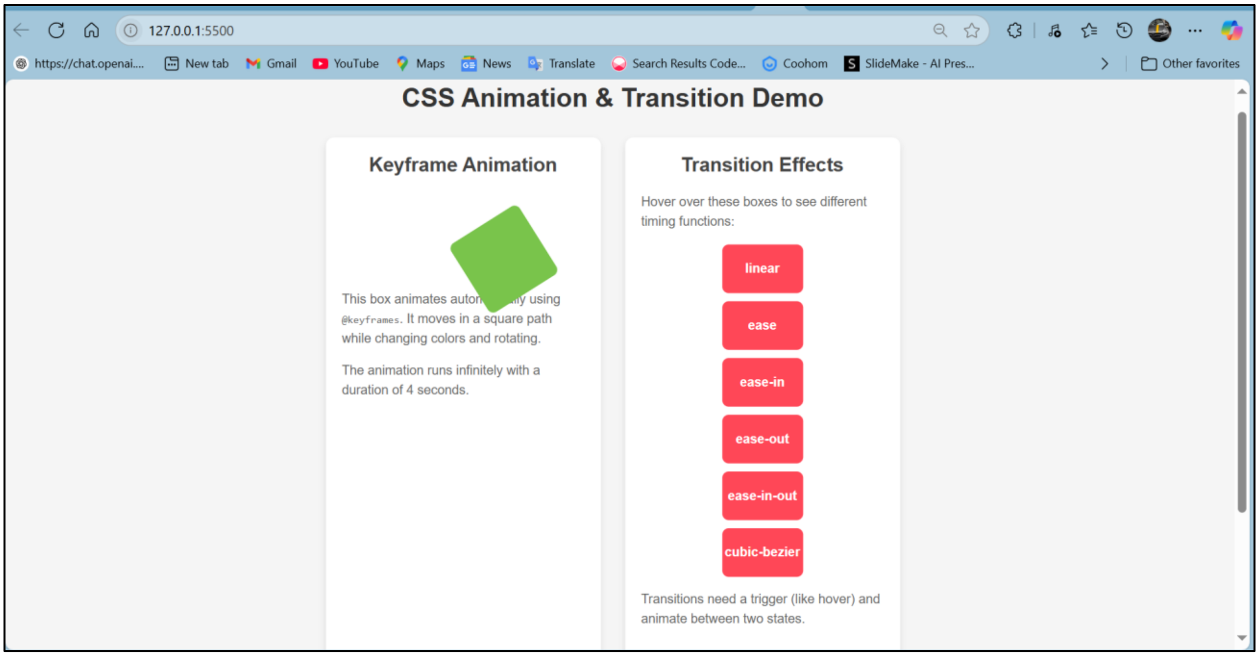


```
  #box3 {transition-timing-function: ease-in;}
  #box4 {transition-timing-function: ease-out;}
  #box5 {transition-timing-function: ease-in-out;}
  #box6 {transition-timing-function: cubic-bezier(0.68, -0.6, 0.32, 1.6);}

  .info {
    background: white;
    padding: 20px;
    border-radius: 8px;
    margin-top: 30px;
    box-shadow: 0 4px 8px rgba(0,0,0,0.1);
  }

  h2 {
    color: #444;
    text-align: center;
    margin-top: 0;
  }

  p {
    color: #666;
    line-height: 1.5;
  }
</style>
</head>
<body>

<h1>CSS Animation & Transition Demo</h1>

<div class="demo-container">
  <div class="animation-demo">
    <h2>Keyframe Animation</h2>
    <div class="animation-box"></div>
    <p>This box animates automatically using <code>@keyframes</code>. It moves
in a square path while changing colors and rotating.</p>
    <p>The animation runs infinitely with a duration of 4 seconds.</p>
  </div>

  <div class="transition-demo">
    <h2>Transition Effects</h2>
    <p>Hover over these boxes to see different timing functions:</p>
    <div id="box1" class="transition-box">linear</div>
    <div id="box2" class="transition-box">ease</div>
    <div id="box3" class="transition-box">ease-in</div>
    <div id="box4" class="transition-box">ease-out</div>
```



```
    <div id="box5" class="transition-box">ease-in-out</div>
    <div id="box6" class="transition-box">cubic-bezier</div>
    <p>Transitions need a trigger (like hover) and animate between two
states.</p>
  </div>
</div>

<div class="info">
  <h2>Key Differences</h2>
  <p><strong>CSS Animations:</strong></p>
  <ul>
    <li>Use <code>@keyframes</code> to define multiple steps</li>
    <li>Can run automatically without user interaction</li>
    <li>Can loop infinitely</li>
    <li>More complex movements and changes</li>
  </ul>

  <p><strong>CSS Transitions:</strong></p>
  <ul>
    <li>Simple transitions between two states</li>
    <li>Require a trigger (like :hover)</li>
    <li>Easier to implement for simple effects</li>
    <li>Various timing functions control the speed curve</li>
  </ul>
</div>

</body>
</html>
```

Output:



CSS Animation & Transition Demo

Keyframe Animation

This box animates auton ly using `@keyframes`. It moves in a square path while changing colors and rotating.

The animation runs infinitely with a duration of 4 seconds.

Transition Effects

Hover over these boxes to see different timing functions:

Transitions need a trigger (like hover) and animate between two states.

Explanation:

Animation Section

- **.animation-box** is styled to be a red square initially.
- It uses the **@keyframes** colorful-move to:
  - Move along a square path.
  - Change colors at each corner.
  - Rotate 360 degrees.
- Animation is set to loop infinitely with animation: **colorful-move 4s linear infinite;.**

3. Transition Section

- **Multiple .transition-box** elements are created, each with a different transition-timing-function like ease-in, ease-out, etc.
- **On :hover,** the boxes expand in width and change color, demonstrating the visual difference each timing function creates.
- Transitions are triggered by hover and **last 2 seconds.**

Conclusion:

This task provided valuable hands-on experience in implementing CSS animations and transitions. Animations using **@keyframes** are ideal for continuous and complex movement, while transitions are perfect for smooth changes on **user interaction**. By combining both, developers can significantly enhance user engagement and interface appeal. Mastering these concepts is essential for modern front-end development and builds the foundation for advanced UI/UX design.