**Experiment No:** 08

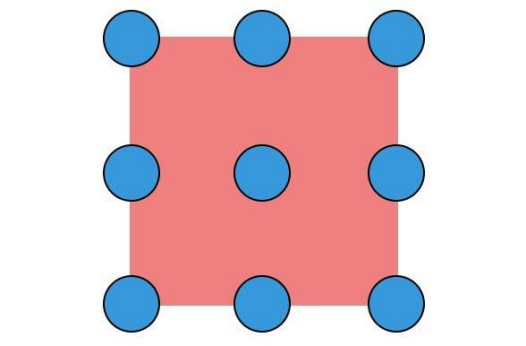
**Experiment Name:** Introduction to CSS Positioning.

**Theory:** CSS positioning is used to control the layout of elements on a web page. It allows developers to place elements precisely where they want using properties like ‘*static*’, ‘*relative*’, ‘*absolute*’, ‘*fixed*’, and ‘*sticky*’. Each type of positioning behaves differently, offering flexibility for organizing content. For example, ‘*relative*’ moves an element to its normal position, while ‘*absolute*’ places it based on its closest positioned ancestor. This helps create dynamic and responsive layouts.

**1. Static Positioning:** It is the default position behavior. If the position property isn't declared in the style of a specific element, the element will revert to the default position: static.

**2. Relative Positioning:** Positioning relative means, the element will move relatively to its normal/actual position. It allows the use of top, right, bottom, and left properties for adjustment.

**Lab Task:** Design similar to this picture using the CSS "position" property:



**Source Code:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>CSS Position</title>

    <link rel="stylesheet" href="position.css".css">

  </head>

  <body>

    <div class="container">

      <div class="square"></div>

      <div class="circle top-left"></div>

      <div class="circle top-center"></div>

      <div class="circle top-right"></div>

      <div class="circle middle-left"></div>

      <div class="circle middle-center"></div>

      <div class="circle middle-right"></div>

      <div class="circle bottom-left"></div>

      <div class="circle bottom-center"></div>

      <div class="circle bottom-right"></div>

    </div>

  </body>

</html>

Inside position.css:

.container {

  position: relative;

  width: 200px;

  height: 200px;

  margin: 50px auto;

}

.square {

  width: 170px;

  height: 170px;

  background-color: #faa2e0;

  position: absolute;

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

}

.circle {

  position: absolute;

  width: 40px;

  height: 40px;

  background-color: rgb(111, 83, 238);

  border-radius: 50%;

}

.circle.top-left {

  top: 0;

  left: 0;

}

.circle.top-center {

  top: 0;

  left: 50%;

  transform: translateX(-50%);

}

.circle.top-right {

  top: 0;

  right: 0;

}

.circle.middle-left {

  top: 50%;

  left: 0;

  transform: translateY(-50%);

}

.circle.middle-center {

  top: 50%;

  left: 50%;

  transform: translate(-50%, -50%);

}

.circle.middle-right {

  top: 50%;

  right: 0;

  transform: translateY(-50%);

}

.circle.bottom-left {

  bottom: 0;

  left: 0;

}

.circle.bottom-center {

  bottom: 0;

  left: 50%;

  transform: translateX(-50%);

}

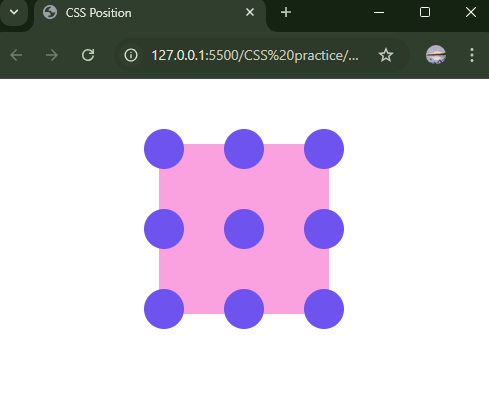
.circle.bottom-right {

  bottom: 0;

  right: 0;

}

**Output:**



**Discussion:** In today’s lab, we got knew about the positioning of CSS.CSS positioning is used to control the layout of elements on a web page. It allows developers to place elements precisely where they want using properties like ‘*static*’, ‘*relative*’, ‘*absolute*’, ‘*fixed*’, and ‘*sticky*’. Each type of positioning behaves differently, offering flexibility for organizing content. For example, ‘*relative*’ moves an element to its normal position, while ‘*absolute*’ places it based on its closest positioned ancestor. This helps create dynamic and responsive layouts.By using properties like ‘*static*’, ‘*relative*’, ‘*absolute*’, ‘*fixed*’, and ‘*sticky*’ ,we completed our lab task.