**Q.1 –** **Describe the main difference between the CSS Flexbox layout model and the CSS Grid layout model. When would you choose to use one over the other?**

**Sol.-** The main difference between the CSS Flexbox layout model and the CSS Grid layout model is how they control the arrangement of elements on a webpage:

* **Flexbox:**

Flexbox is best for arranging items in a single direction, either horizontally or vertically.

It's like a row of items or a column of items.

You can easily make items in a row or column stretch and fill the available space.

Good for creating navigation bars, lists, and centering items.

* **Grid:**

Grid is ideal for creating two-dimensional layouts with rows and columns.

It's like a grid of items, similar to a table.

You can precisely control the placement of items in both rows and columns.

Great for creating complex layouts like magazine-style pages or responsive designs with multiple sections.

* **When to use each one:**

Use Flexbox when you need to align items in a single direction, like in a navigation bar or a list. It's good for simple, linear layouts.

Use Grid when you want to create more complex layouts with both rows and columns, such as a grid of images or a multi-column webpage layout. It's perfect for creating structured and responsive designs.

**Q.2 – Explain the role of the following key properties in the Flexbox layout model:**

**1. justify-content**

**2. align-items**

**3. gap**

**4. flex-direction**

**5. flex-wrap**

**Sol.-**

1. **justify-content:**
   * This property helps you control the horizontal alignment of items within a flex container.
   * You can use it to make items spread out evenly, align them to the left or right, or even center them along the horizontal axis.
2. **align-items:**
   * It's used to control the vertical alignment of items within a flex container.
   * You can use it to make items align to the top, bottom, center, or stretch to fill the container vertically.
3. **gap:**
   * This property sets the space between items inside a flex container.
   * It's like adding some breathing room between your items, making your layout look clean and organized.
4. **flex-direction:**
   * It determines the direction in which your flex items are laid out within the container.
   * You can make items go from left to right, right to left, top to bottom, or bottom to top, depending on your needs.
5. **flex-wrap:**
   * This property decides whether your flex items should wrap onto a new line when they run out of space in the container.
   * It's like saying whether your items should stay in one row or go to the next row if there's not enough room.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>IRCTC</title>

    <link rel="sylesheet" href="/Flexbox in CSS/Question 5/rail.css">

</head>

<body>

<!-- Container -->

<div class="container">

    <!-- Section1 -->

    <section class="section1">

        <!-- Navigation -->

        <nav>

            <div class="logo">

                <img src="/Flexbox in CSS/Question 5/assets/IRCTCLogo.png">

                <img src="/Flexbox in CSS/Question 5/assets/indianRailwaysLogo.png">

            </div>

            <div class="navigation">

                <ul>

                    <li>Ask Disha</li>

                    <li>Contact us</li>

                    <li>Alerts</li>

                </ul>

                <ul>

                    <li>

                        <button class="selected">Login</button>

                    </li>

                    <li>

                        <button>Register</button>

                    </li>

                    <li>

                        <button>Agent Login</button>

                    </li>

                </ul>

            </div>

        </nav>

        <!-- Main -->

    </section>

</div>

</body>

</html>