

Name = Om Kumar

Email ID = [omk5938@gmail.com](mailto:omk5938@gmail.com)

Assignment Name = Flexbox in CSS

GitHub = [Link](#)

Q1. Describe the main differences between the CSS Flexbox layout model and the CSS Grid layout model. When would you choose to use one over the other?

The **CSS Flexbox** and **CSS Grid** layout models are both powerful tools for creating responsive web layouts, but they serve different purposes and are best suited for different use cases.

---

## Main Differences Between Flexbox and Grid

Feature	Flexbox	Grid
<b>Layout Direction</b>	One-dimensional (either row or column)	Two-dimensional (both rows and columns)
<b>Use Case</b>	Aligning items along a single axis	Building full page or complex grid-based layouts
<b>Child Placement</b>	Based on content flow; order matters	Explicit placement via grid lines, rows, and columns
<b>Control</b>	Better for content-driven layouts	Better for layout-driven (design-first) structures
<b>Alignment</b>	Powerful alignment features (justify, align)	Also supports alignment, with more precise control
<b>Responsiveness</b>	Easy to create flexible layouts with wrapping	Supports responsive design using <code>auto-fit</code> , <code>minmax</code>
<b>Browser Support</b>	Very well supported	Well supported in modern browsers

---

## When to Use Flexbox

Use Flexbox when:

- You are aligning items in a row or column.

- You need to distribute space dynamically between items.
- The layout is simple (e.g., navbar, card components, toolbars).
- You want elements to wrap based on container size.

**Example:** A navigation bar with horizontal links that wrap on small screens.

---

## When to Use Grid

Use Grid when:

- You need a full-page layout with rows and columns.
- You're designing a layout-first structure (e.g., dashboard, portfolio page).
- You want precise control over item positioning.
- You have overlapping or layered elements.

**Example:** A complex landing page with a sidebar, header, content area, and footer.

---

## Pro Tip:

You can **combine both** — use Grid for the overall page layout, and Flexbox inside components for internal item alignment.

Q2. Explain the role of the following key properties in the Flexbox layout model.

1. Justify-content
  2. Align-item
  3. Gap
  4. flex-direction
  5. Flex-wrap
-

## 1. **justify-content**

**Purpose:** Aligns items **horizontally** (main axis).

**Used on:** Flex **container**

**Values:**

- **flex-start** – items align to the start of the main axis (default)
- **flex-end** – items align to the end
- **center** – items are centered
- **space-between** – equal space between items
- **space-around** – equal space around items
- **space-evenly** – equal space between and around items

**Example:**

```
.container {  
  display: flex;  
  justify-content: center;  
}
```

---

## 2. **align-items**

**Purpose:** Aligns items **vertically** (cross axis).

**Used on:** Flex **container**

**Values:**

- **stretch** – items stretch to fill container (default)

- **flex-start** – items align to the start of the cross axis
- **flex-end** – items align to the end
- **center** – items are centered
- **baseline** – aligns items based on text baseline

**Example:**

```
.container {  
  display: flex;  
  align-items: flex-start;  
}
```

---

### 3. **gap**

**Purpose:** Adds **space between flex items** (no need for margins!).

**Used on:** Flex **container** (also works with Grid)

**Example:**

```
.container {  
  display: flex;  
  gap: 20px;  
}
```

---

### 4. **flex-direction**

**Purpose:** Defines the **main axis direction**.

**Used on:** Flex container

**Values:**

- `row` – left to right (default)
- `row-reverse` – right to left
- `column` – top to bottom
- `column-reverse` – bottom to top

**Example:**

```
.container {  
  display: flex;  
  flex-direction: column;  
}
```

---

## 5. `flex-wrap`

**Purpose:** Allows items to **wrap** onto multiple lines if they don't fit in one row/column.

**Used on:** Flex container

**Values:**

- `nowrap` – all items stay on one line (default)
- `wrap` – items wrap onto the next line
- `wrap-reverse` – items wrap in the opposite direction

**Example:**

```
.container {
```

```
display: flex;

flex-wrap: wrap;

}
```

Q3. Write the code to center a div using CSS Flexbox.

HTML CODE:-

```
<!DOCTYPE html>

<html>

  <head>

    <title>Hello, World!</title>

    <link rel="stylesheet" href="styles.css" />

  </head>

  <body>

    <div>

      <div class="container">

        <div class="box"></div>

      </div>

    </div>

  </body>

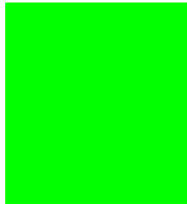
</html>
```

CSS :-

```
.container {
  display: flex;
  justify-content: center; /* Horizontal center */
  align-items: center;    /* Vertical center */
  height: 100vh;         /* Full viewport height */
  background: white;
```

```
}  
  
.box {  
  width: 200px;  
  height: 200px;  
  background-color: lime; /* Same as bright green */  
}
```

Output:-



Q4. A client of yours wants to add a pricing section on their website to showcase their newly introduced premium plans.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Pricing Section</title>  
  <style>  
    body {  
      margin: 0;  
      padding: 40px;  
      font-family: Arial, sans-serif;  
      background-color: #f4f1ff;  
    }  
  
    .pricing-section {  
      display: flex;
```

```
justify-content: center;
gap: 40px;
flex-wrap: wrap;
}
```

```
.card {
width: 200px;
padding: 50px;
background-color: white;
border-radius: 20px;
box-shadow: 0 0 15px rgba(0,0,0,0.1);
text-align: center;
}
```

```
.card.dark {
background-color: #1e1e3f;
color: white;
}
```

```
.card h2 {
font-size: 22px;
margin-bottom: 5px;
}
```

```
.card p {
font-size: 14px;
color: inherit;
opacity: 0.7;
margin-bottom: 20px;
}
```

```
.card h3 {
font-size: 28px;
margin: 10px 0 20px;
}
```

```
.card ul {
list-style: none;
padding: 0;
margin: 0 0 20px;
text-align: left;
}
```

```
.card ul li {
```



```
margin: 10px 0;
font-size: 14px;
}
```

```
.card ul li::before {
  content: "✓ ";
  color: #000;
}
```

```
.card.dark ul li::before {
  color: white;
}
```

```
.card button {
  padding: 12px 20px;
  border: none;
  border-radius: 25px;
  font-size: 14px;
  font-weight: bold;
  cursor: pointer;
  background-color: #1e1e3f;
  color: white;
}
```

```
.card.dark button {
  background-color: white;
  color: #1e1e3f;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="pricing-section">
```

```
<!-- Free Plan -->
```

```
<div class="card">
```

```
<h2>Free Plan</h2>
```

```
<p>For personal</p>
```

```
<h3>$0</h3>
```

```
<ul>
```

```
<li>1 users</li>
```

```
<li>4 Web mails</li>
```

```
<li>Responsive Website</li>
```

```
<li>Free SSL</li>
```

```
</ul>
```

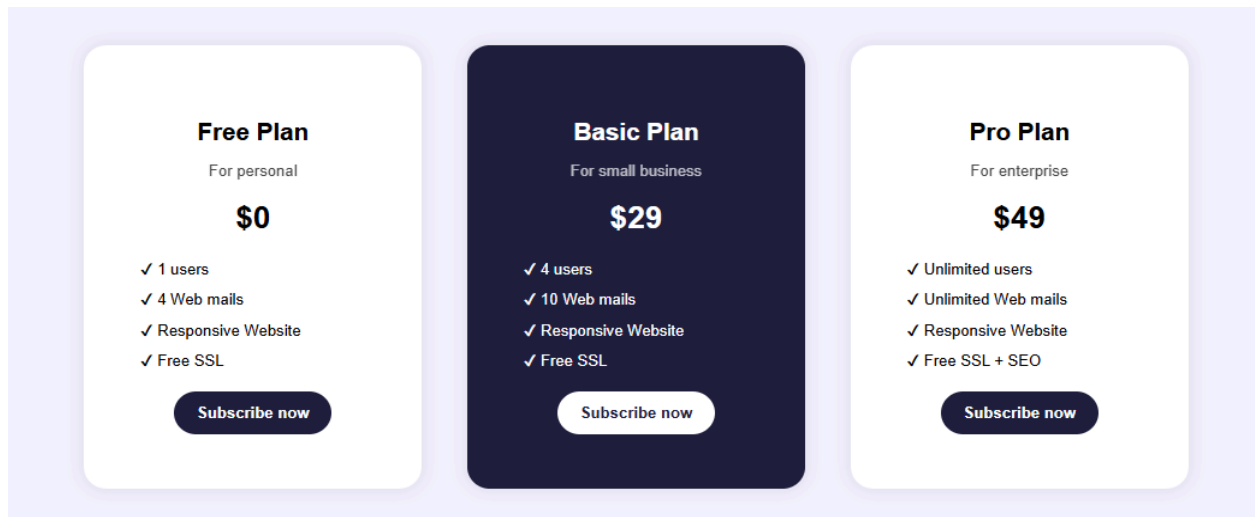
```
<button>Subscribe now</button>
</div>
```

```
<!-- Basic Plan -->
<div class="card dark">
  <h2>Basic Plan</h2>
  <p>For small business</p>
  <h3>$29</h3>
  <ul>
    <li>4 users</li>
    <li>10 Web mails</li>
    <li>Responsive Website</li>
    <li>Free SSL</li>
  </ul>
  <button>Subscribe now</button>
</div>
```

```
<!-- Pro Plan -->
<div class="card">
  <h2>Pro Plan</h2>
  <p>For enterprise</p>
  <h3>$49</h3>
  <ul>
    <li>Unlimited users</li>
    <li>Unlimited Web mails</li>
    <li>Responsive Website</li>
    <li>Free SSL + SEO</li>
  </ul>
  <button>Subscribe now</button>
</div>
</div>
```

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

Output:-



Q5. build a clone of the IRCTC Ticket booking page.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>IRCTC Ticket Booking (Clone)</title>
  <style>
    body {
      font-family: 'Roboto', sans-serif; /* Using a common, clean font */
      margin: 0;
      padding: 0;
      background-color: #f0f2f5; /* Light grey background */
      color: #333;
    }

    header {
      background-color: #ca2127; /* Typical IRCTC red */
      color: white;
      padding: 15px 20px;
      display: flex;
      justify-content: space-between;
      align-items: center;
    }
```

```
.logo {
  display: flex;
  align-items: center;
}

.logo img {
  height: 40px;
  margin-right: 15px;
}

.logo h1 {
  margin: 0;
  font-size: 1.4em;
  font-weight: 500;
}

nav ul {
  list-style: none;
  padding: 0;
  margin: 0;
  display: flex;
  align-items: center;
}

nav ul li {
  margin-left: 15px;
}

nav ul li a {
  color: white;
  text-decoration: none;
  padding: 8px 12px;
  border-radius: 5px;
  font-size: 0.9em;
}

nav ul li a:hover {
  background-color: rgba(255, 255, 255, 0.15);
}
```

```

}

.booking-container {
  background-color: #fff;
  margin: 20px auto;
  padding: 20px;
  max-width: 960px; /* Wider container */
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

.booking-tabs {
  display: flex;
  margin-bottom: 20px;
  border-bottom: 2px solid #ddd;
}

.booking-tabs button {
  padding: 12px 18px;
  border: none;
  background-color: transparent;
  color: #555;
  cursor: pointer;
  font-weight: bold;
  border-bottom: 2px solid transparent;
  margin-right: 20px;
  font-size: 1em;
}

.booking-tabs button.active {
  color: #ca2127;
  border-bottom: 2px solid #ca2127;
}

.booking-form {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); /*
Responsive columns */
  gap: 15px;

```

```
padding: 20px 0;
align-items: center;
}

.input-group {
display: flex;
flex-direction: column;
}

.input-group label {
font-size: 0.9em;
margin-bottom: 5px;
color: #333;
font-weight: bold;
}

.input-group input[type="text"],
.input-group select {
padding: 10px;
border: 1px solid #ccc;
border-radius: 4px;
font-size: 0.9em;
}

.input-group .station-code {
font-size: 0.8em;
color: #777;
margin-top: 3px;
}

.input-group.swap-icon {
display: flex;
justify-content: center;
align-items: center;
}

.input-group.swap-icon img {
height: 25px;
cursor: pointer;
}
```

```
}

.search-button {
  background-color: #28a745; /* Green search button */
  color: white;
  padding: 12px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  font-size: 1.1em;
  font-weight: bold;
  transition: background-color 0.2s ease;
  grid-column: 1 / -1; /* Make it full width on smaller screens */
}

.search-button:hover {
  background-color: #1e7e34;
}

.image-section {
  display: flex;
  justify-content: space-around;
  margin: 20px auto;
  max-width: 960px;
}

.image-section img {
  max-width: 48%;
  height: auto;
  border-radius: 8px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

footer {
  text-align: center;
  padding: 15px;
  background-color: #333;
  color: white;
  font-size: 0.9em;
}
```

```

    }

    /* Basic styling for swap icon */
    .swap-icon {
        font-size: 1.2em;
        color: #555;
        cursor: pointer;
        display: flex;
        align-items: center;
        justify-content: center;
    }
</style>
</head>
<body>
    <header>
        <div class="logo">
            
            <h1>Indian Railways</h1>
        </div>
        <nav>
            <ul>
                <li><a href="#">Ask Disha</a></li>
                <li><a href="#">Contact Us</a></li>
                <li><a href="#">Alerts</a></li>
                <li><a href="#" class="login">Login</a></li>
                <li><a href="#" class="register">Register</a></li>
            </ul>
        </nav>
    </header>

    <main class="booking-container">
        <div class="booking-tabs">
            <button class="tab active">Book Ticket</button>
            <button class="tab">PNR Status</button>
            <button class="tab">Charts / Vacancy</button>
        </div>

```



```

<div class="booking-form">
  <div class="input-group">
    <label for="from">From</label>
    <input type="text" id="from" placeholder="Enter Source Station">
    <p class="station-code">Station Code</p>
  </div>
  <div class="swap-icon">
    &#8644; </div>
  <div class="input-group">
    <label for="to">To</label>
    <input type="text" id="to" placeholder="Enter Destination
Station">
    <p class="station-code">Station Code</p>
  </div>
  <div class="input-group">
    <label for="date">Date</label>
    <input type="text" id="date" placeholder="DD-MM-YYYY">
  </div>
  <div class="input-group">
    <label for="class">Class</label>
    <select id="class">
      <option>All Classes</option>
      <option>AC First Class (1A)</option>
      <option>AC 2 Tier (2A)</option>
      <option>AC 3 Tier (3A)</option>
      <option>Sleeper (SL)</option>
      <option>... (more classes)</option>
    </select>
  </div>
  <div class="input-group">
    <label for="quota">Quota</label>
    <select id="quota">
      <option>General</option>
      <option>Tatkal</option>
      <option>Premium Tatkal</option>
      <option>Ladies</option>
      <option>... (more quotas)</option>
    </select>
  </div>
</div>

```

```
        <button class="search-button">Search Train</button>
    </div>
</main>

<div class="image-section">
    
    
</div>

<footer>
    <p>&copy; 2025 Indian Railways (Clone)</p>
</footer>

<script>
    // Basic tab switching functionality (no content change)
    const tabs = document.querySelectorAll('.booking-tabs button');
    tabs.forEach(tab => {
        tab.addEventListener('click', function() {
            tabs.forEach(t => t.classList.remove('active'));
            this.classList.add('active');
        });
    });
</script>
</body>
</html>
```

Output:-

