Q1.

**ANS:-**

**Flexbox**:

Flexbox is primarily designed for one-dimensional layouts, either in a row or column direction.

It is best suited for laying out items within a single container along a single axis, allowing flexible sizing and alignment of items within that axis.

Flexbox is particularly useful for creating responsive designs, as it automatically adjusts the size and position of items based on available space.

**Grid**:

CSS Grid is designed for two-dimensional layouts, allowing you to define rows and columns to create complex layouts.

It provides precise control over both the rows and columns, allowing you to create more intricate layouts than Flexbox.

Grid is well-suited for layouts where you need to align items in both rows and columns, such as grid-based designs found in many web applications.

It allows you to position items anywhere on the grid, making it ideal for more complex designs with specific placement requirements.

Flexbox is best for one-dimensional layouts along a single axis, providing flexibility and responsiveness, while Grid is better suited for two-dimensional layouts with precise control over rows and columns, allowing for more complex and structured designs. Often, they can be used together within the same project to leverage their respective strengths for different parts of the layout.

Q2

ANS:-

**1.justify-content** :- is a CSS property used within Flexbox layouts to align flex items along the main axis of the flex container. It controls the distribution of space between and around flex items along this axis. Here's a breakdown of its possible values.

Example:-Flex-start, Flex-end, center ,space-between, space-around.

**2.align-items:-** is a CSS property used within Flexbox layouts to control how flex items are aligned along the cross axis of the flex container.

**Example:-** stretch: This is the default value. It stretches the flex items to fill the container along the cross axis. This is often used when you want flex items to expand to match the height of the container.

**3.gap**:- The gap property simplifies the process of creating grid layouts by providing an easy way to define consistent spacing between grid items without needing to specify the gap for rows and columns separately. It enhances readability and maintainability of grid layout code.

4. **flex-direction** :-The flex-direction property is used in CSS Flexbox layouts to define the direction in which flex items are placed inside a flex container. It specifies whether the flex container's main axis runs horizontally or vertically and the direction in which flex items are placed within the container.

5. **flex-wrap** :-The flex-wrap property is used in CSS Flexbox layouts to control whether flex items are forced onto a single line or can be wrapped onto multiple lines within a flex container when there isn't enough space to accommodate them all on one line