CSS Grid

Question 1: Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?

Answer:

CSS Grid is a two-dimensional layout system in CSS. It allows you to arrange elements into rows and columns.

You define a grid container with display: grid, and then set up:

- grid-template-columns - defines how many columns and their widths.
- grid-template-rows - defines how many rows and their heights.
- gap (or grid-gap) - controls spacing between rows and columns.

It's useful when you need a structured, grid-like layout, such as dashboards, image galleries, or page layouts.

Example:

```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* 3 equal columns */
  grid-gap: 20px;
}
```

Grid vs. Flexbox :-

Flexbox is one-dimensional: it arranges items in either a row horizontal or a column vertical. Grid is two-dimensional: it can arrange items across both rows and columns at the same time.

	Flexbox	Grid
Layout Direction	One-dimensional row or column.	Two-dimensional rows and columns.
Best For	Small components, navbars, buttons.	Full-page layouts, dashboards, galleries.

Alignment	Great at distributing space in one direction.	Excellent at aligning items in both directions.

When to use Grid over Flexbox :-

Use Grid when you need a structured, multi-row and multi-column layout.
Use Flexbox when you only need to arrange items in a single line.

Question 2: Describe the grid-template-columns, grid-template-rows, and grid-gap properties. Provide examples of how to use them.

Answer :

1. grid-template-columns :-
   Defines the number and size of columns in the grid.
   You can specify sizes in `px`, `%`, `fr` , or use `auto`.

   Example :

   ```
   .container {
   display: grid;
    grid-template-columns: 200px 1fr 2fr;
   }
   ```

   This creates 3 columns:

   First column = 200px fixed
   Second column = 1 fraction of remaining space
   Third column = 2 fractions of remaining space

2. grid-template-rows :-
   Defines the number and size of rows in the grid.
   Works like grid-template-columns but applies vertically.

   Example :

   ```
   .container {
   display: grid;
   ```

```
 grid-template-rows: 100px auto 50px;
}
```

This creates 3 rows:

First row = 100px fixed
Second row = auto takes content height
Third row = 50px fixed

3. grid-gap :-

   Adds spacing between rows and columns in the grid.
   Can use one value for both rows & columns or two values row-gap, column-gap.

   Example :

   ```
   .container {
     display: grid;
     grid-template-columns: repeat(3, 1fr);
     grid-template-rows: 150px 150px;
     grid-gap: 20px;
   }
   ```

   This creates a 3x2 grid with equal columns, fixed row heights, and 20px spacing between items.