**Task 3 – Explain the use of grid-auto-row and grid-auto-column using code examples.**

**Sol.-**

**grid-auto-row** and **grid-auto-column** are properties in CSS Grid that allow you to control the size of rows and columns that are created automatically when you have more items than explicitly defined rows or columns in your grid.

**grid-auto-row**:

* **grid-auto-row** sets the size of rows that are created automatically in the grid.

.grid-container {

display: grid;

grid-template-columns: 1fr 1fr;

grid-auto-rows: 100px; /\* Set the height of automatically created rows \*/

}

**grid-auto-column**:

* **grid-auto-column** sets the size of columns created automatically in the grid.

.grid-container {

display: grid;

grid-template-rows: 1fr 1fr;

grid-auto-columns: 150px; /\* Set the width of automatically created columns \*/

}

**Task 5 - Explain the difference between justify-items and justify-self using code examples.**

**Sol. –**

1. **justify-items**:

**justify-items** is used to set the horizontal alignment of all the items within a grid or flex container.

.container {

display: grid;

justify-items: center; /\* Center all grid items horizontally \*/

}

1. **justify-self**:

**justify-self** is used to set the horizontal alignment of an individual grid or flex item within a grid or flex container.

.item {

justify-self: start; /\* Align this item to the start (left) within the grid cell \*/

}

The key difference is that **justify-items** applies the horizontal alignment to all items within the container, while **justify-self** allows you to set the alignment for each item individually. This gives you more fine-grained control over the alignment of items within a grid or flex container.