i) I would use 3 divs surrounded by a wrapper div to instantiate the html framework of this design. Then I would designate each div with a class of “grid” the wrapper div would have the class names “div wrapper”. Inside the wrapper div I would use the class names of “grid col1” “grid col2” and “grid col1 final” respectively. This is to designate what each type of column is and to divide the grid columns into subtypes.

Inside the css I would use display:grid for the wrapper grid so that all children divs inherit this class designation. Next I would set the “grid-template-column” attribute to “1fr 2fr 1fr” in order to create a layout that is 1-2-1 in designation with the 2fr column being half the size of the screen. This works because the fractional units of 1fr 2fr 1fr designate the amount of fr units each column takes up, with 2 being double the size of 1.

For the height, since each column is different I would have to give the heights in either percentages or rem units. Columns with the designation “col1” would be given a height of “50%”, “col2” would be given “25%”, and “col1 final” would be given a height of “100%” this gives each column a different height based on the percentage of screen-y it takes up.

When applying the color attributes you can give them the designated colors by giving the attribute “background-color” and either specifying the values as “red, green blue” respective for each column or you can go “rgb(200,50,50)”,“rgb(50,200,50)”,“rgb(50,50,200)”. This colors the divs.

ii) Use 3 divs wrapped in 1 div. Call the wrapper div class “row” and the 3 column divs class “column col1”,“column col2”,“column col1 final” respectively. This is to make the base html for the column design.

In CSS create a styling guide based on the names provided namely creating a * class that encapsulated “box-sizing:border-box” to ensure everything follows the border box model of css standards. For the 1st and 3rd column establish a col1 class , inside give the attributes float:left, width25% attributes. These give it the attributes of 25% of the screen and put height:50%, we will later define the height in their specialized classes, also put background color:red. This works because we are directly defining the width and height of each column, and since the 1st and 3rd column have the same 2nd class we can define them at the same time.

For the middle column, give the same float attribute but designate the width as 50%, this ensures that the width of the column is double the size of the other columns, and go ahead and define the background color for this one as “background-color: green”. For the final column, we havent defined the height or the color yet, so create a class named “final” in the css and give it a height of “100%” and a background color of blue. This works because the final class has higher priority of col1, overriding it when it renders.