<script>

// Define a function to format numbers with commas

*function* comma(*number*) {

  return *number*.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",");

}

*let* chartWidth = 533; // The width of the chart from the image

*let* chartHeight = 277; // The height of the chart from the image

*let* financialYears = [

      { year: 'FY23', RMG: 16428, NonRMG: 13596 },

      { year: 'FY24E', RMG: 19974, NonRMG: 16520 },

      { year: 'FY25E', RMG: 22421, NonRMG: 18171 },

      { year: 'FY26E', RMG: 25440, NonRMG: 20170 },

      { year: 'FY27E', RMG: 28923, NonRMG: 22389 },

      { year: 'FY28', RMG: 33243, NonRMG: 25076 }

    ];

          // Define the maximum value for scaling

*const* maxValue = Math.max(...financialYears.flatMap(*fy* *=>* [*fy*.RMG, *fy*.NonRMG]));

// Define tooltip state

*let* tooltip = { display: 'none', data: null, x: 0, y: 0 };

// Calculate scaled height for bars

*function* getScaledHeight(*value*) {

  return (*value* / maxValue) \* chartHeight;

}

      // Functionality for tooltips on hover

*function* showTooltip(*data*, *event*) {

      tooltip = {

        display: 'block',

        data: *data*,

        x: *event*.clientX,

        y: *event*.clientY

      };

    }

*function* hideTooltip() {

      tooltip.display = 'none';

    }

</script>

<div class="chart-container" bind:clientWidth={chartWidth}>

    <svg width="{chartWidth}" height="{chartHeight}">

    {#each financialYears as data, i}

        <!-- RMG Bars -->

        <rect class="bar-rmg"

            x={i \* (chartWidth / financialYears.length)}

            y={chartHeight - getScaledHeight(data.RMG)}

            width={(chartWidth / financialYears.length) / 1.5}

            height={getScaledHeight(data.RMG)}

            on:mouseover="{*event* *=>* showTooltip(data, *event*)}"

            on:mouseout="{hideTooltip}"

            fill="#5B6DCD" />

        <!-- Non-RMG Bars -->

        <rect class="bar-nonrmg"

            x={i \* (chartWidth / financialYears.length) + (chartWidth / financialYears.length) / 180}

            y={chartHeight - getScaledHeight(data.NonRMG)}

            width={(chartWidth / financialYears.length) / 1.5}

            height={getScaledHeight(data.NonRMG)}

            on:mouseover="{*event* *=>* showTooltip(data, *event*)}"

            on:mouseout="{hideTooltip}"

            fill="#A27DDE" />

        <!-- Year Labels -->

        <text x={i \* (chartWidth / financialYears.length)}

            y={chartHeight + 15}

            fill="black">{data.year}</text>

    {/each}

    <!-- Tooltip -->

    {#if tooltip.display === 'block'}

    <foreignObject x={tooltip.x} y={tooltip.y} width="100" height="50" style="overflow:visible;">

        <div xmlns="http://www.w3.org/1999/xhtml" class={`tooltip ${tooltip.display === 'block' ? 'tooltip-visible' : ''}`}>

        {tooltip.data.year}<br/>

        RMG: {comma(tooltip.data.RMG)}<br/>

        Non-RMG: {comma(tooltip.data.NonRMG)}

        </div>

    </foreignObject>

    {/if}

    </svg>

</div>

<style>

.tooltip {

*background-color*: #fff;

*border*: 1px solid #ddd;

*padding*: 10px;

*position*: absolute;

*text-align*: left;

*display*: none; /\* initially hidden \*/

*pointer-events*: none; /\* makes it so that mouse events pass through to elements below \*/

}

.tooltip-visible {

*display*: block; /\* shown when this class is added \*/

}

</style>