lab-2.4\chart.js

```
// It's best practice to wrap your D3 code in a function that runs once the window has loaded.
window.onload = function () {

    // --- Configuration Variables ---
    const w = 500;
    const h = 150;
    const barPadding = 2;

    // --- D3 Code ---

    // Step 4: Reading in the data using d3.csv()
    // D3 will fetch the file and parse it. The .then() block is executed
    // once the data is successfully loaded.
    d3.csv("Task_2.4_data.csv").then(function (data) {
        // The 'data' variable now holds the parsed CSV data.
        // Let's log it to the console to see its structure.
        console.log(data);

        // Call our function to generate the chart, passing the loaded data.
        generateBarChart(data);
    }).catch(function (error) {
        // Handle any errors that might occur during loading
        console.log("Error loading the CSV file:", error);
    });

    // This function contains the bar chart drawing logic.
    function generateBarChart(wombatSightings) {

        // Create the SVG element inside the #chart div
        const svg = d3.select("#chart")
            .append("svg")
            .attr("width", w)
            .attr("height", h);

        // Create the bars (rectangles)
        svg.selectAll("rect")
            .data(wombatSightings)
            .enter()
            .append("rect")
            .attr("x", function (d, i) {
                // Position bars evenly across the SVG width
                return i * (w / wombatSightings.length);
            })
            .attr("y", function (d) {
                // Position the top of the bar. Note that CSV values are read as strings,
                // so we use '+' to convert d.wombats to a number.
                return h - (+d.wombats * 4); // Added a multiplier for better height
            })
```

```
            .attr("width", w / wombatSightings.length - barPadding)
            .attr("height", function (d) {
                // Set the height based on the 'wombats' column from the CSV.
                return +d.wombats * 4;
            })
            .attr("fill", function (d) {
                // Step 5: Change color based on data value.
                if (+d.wombats > 20) {
                    return "rgb(25, 60, 160)"; // Darker blue for high values
                }
                return "rgb(70, 130, 180)"; // Steel blue for lower values
            });

        // Add labels to the bars
        svg.selectAll("text")
            .data(wombatSightings)
            .enter()
            .append("text")
            .text(function (d) {
                return d.wombats;
            })
            .attr("x", function (d, i) {
                return i * (w / wombatSightings.length) + (w / wombatSightings.length - barPadding) / 2;
            })
            .attr("y", function (d) {
                return h - (+d.wombats * 4) + 14; // Position text inside the bar
            })
            .attr("font-family", "sans-serif")
            .attr("font-size", "11px")
            .attr("fill", "white")
            .attr("text-anchor", "middle");
    }
};

```