Lab 5_2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="description" content="Data Visualization"/>
    <meta name="keywords" content="HTML, CSS, D3"/>
    <meta name="author" content="Dai Vy"/>
    <title>Task 5.2 D3 Transitions</title>

    <script src="https://d3js.org/d3.v7.min.js"></script>
    <style>

    </style>
</head>
<body>

    <button id="cubic">Transition 1</button>
    <button id="elastic">Transition 2</button>
    <button id="update">Update</button>
    <h1>LAB 5.2 D3 Transitions</h1>

    <script>
        var w = 500;
        var h = 100;
        var maxValue = 25;
        var dataset = [24, 10, 29, 19, 8, 15, 20, 12, 9, 6, 21, 28];

        // Use scaleBand() to create an ordinal scaleable x-axis based on the range of the
data set.
        var xScale = d3.scaleBand()
                .domain(d3.range(dataset.length))
                .rangeRound([0, w])
                .paddingInner(0.05);

        // Use scaleLinear() to create a linear scaleable y-axis based on the range of the
data set.
        var yScale = d3.scaleLinear()
                .domain([0, d3.max(dataset)])
                .range([0, h]);

        var svg = d3.select("body")
                .append("svg")
                .attr("width", w)
                .attr("height", h);

        // Draw initial bars
        svg.selectAll("rect")
            .data(dataset)
```

```
            .enter()
            .append("rect")
            .attr("x", function(d, i) {
                return xScale(i);
            })
            .attr("y", function(d) {
                return h - yScale(d);
            })
            .attr("width", xScale.bandwidth())
            .attr("height", function(d) {
                return yScale(d);
            })
            .attr("fill", "steelblue");

        // Draw initial labels
        svg.selectAll("text")
            .data(dataset)
            .enter()
            .append("text")
            .text(function(d) {
                return d; // This displays the number on each bar
            })
            .attr("x", function(d, i) {
                return xScale(i) + xScale.bandwidth() / 2;
            })
            .attr("y", function(d) {
                return h - yScale(d) + 14;
            })
            .attr("text-anchor", "middle")
            .attr("fill", "white")
            .attr("font-size", "12px");

        var duration = 2000;

        // General update function for bars and labels with transition options
        function updateChart(easeFn) {
            // Generate new random dataset
            dataset = [];
            var numBars = Math.floor(Math.random() * 10) + 8; // Random number of bars
between 8 and 17
            for (var i = 0; i < numBars; i++) {
                dataset.push(Math.floor(Math.random() * maxValue));
            }

            // Update scales
            xScale.domain(d3.range(dataset.length));
            yScale.domain([0, d3.max(dataset)]);

            // Calculate per-bar delay so total time is duration
            var delayStep = duration / dataset.length;
```

```
96
97              // DATA JOIN for bars
98              var bars = svg.selectAll("rect")
99                  .data(dataset);
100
101             // EXIT old elements
102             bars.exit()
103                 .transition()
104                 .duration(duration)
105                 .ease(easeFn)
106                 .attr("y", h)
107                 .attr("height", 0)
108                 .remove();
109
110             // UPDATE existing bars
111             bars.transition()
112                 .duration(duration)
113                 .delay(function(d, i) { return i * delayStep; })
114                 .ease(easeFn)
115                 .attr("x", function(d, i) { return xScale(i); })
116                 .attr("y", function(d) { return h - yScale(d); })
117                 .attr("width", xScale.bandwidth())
118                 .attr("height", function(d) { return yScale(d); });
119
120             // ENTER new bars
121             bars.enter()
122                 .append("rect")
123                 .attr("x", function(d, i) { return xScale(i); })
124                 .attr("y", h)
125                 .attr("width", xScale.bandwidth())
126                 .attr("height", 0)
127                 .attr("fill", "steelblue")
128                 .transition()
129                 .duration(duration)
130                 .delay(function(d, i) { return i * delayStep; })
131                 .ease(easeFn)
132                 .attr("y", function(d) { return h - yScale(d); })
133                 .attr("height", function(d) { return yScale(d); });
134
135             // DATA JOIN for labels
136             var labels = svg.selectAll("text")
137                 .data(dataset);
138
139             // EXIT old labels
140             labels.exit()
141                 .transition()
142                 .duration(duration)
143                 .ease(easeFn)
144                 .attr("y", h)
145                 .remove();
```

```
146
147             // UPDATE existing labels
148             labels.transition()
149                 .duration(duration)
150                 .delay(function(d, i) { return i * delayStep; })
151                 .ease(easeFn)
152                 .text(function(d) { return d; })
153                 .attr("x", function(d, i) { return xScale(i) + xScale.bandwidth() / 2; })
154                 .attr("y", function(d) { return h - yScale(d) + 14; });
155
156             // ENTER new labels
157             labels.enter()
158                 .append("text")
159                 .text(function(d) { return d; })
160                 .attr("x", function(d, i) { return xScale(i) + xScale.bandwidth() / 2; })
161                 .attr("y", h)
162                 .attr("text-anchor", "middle")
163                 .attr("fill", "white")
164                 .attr("font-size", "12px")
165                 .transition()
166                 .duration(duration)
167                 .delay(function(d, i) { return i * delayStep; })
168                 .ease(easeFn)
169                 .attr("y", function(d) { return h - yScale(d) + 14; });
170         }
171
172         // Button event listeners for different transitions
173         d3.select("#cubic")
174             .on("click", function() { updateChart(d3.easeCubicInOut); });
175         d3.select("#elastic")
176             .on("click", function() { updateChart(d3.easeElasticOut); });
177         d3.select("#update")
178             .on("click", function() { updateChart(d3.easeCubicInOut); });
179
180     </script>
181
182     <br>
183
184     <footer style="color: grey">COS30045 Data Visualization<br>
185         Dai Vy
186     </footer>
187 </body>
188 </html>
```