```
//This is a module I built that is used on the front end to display all the graphs on the dashboard and student pages.

function random_rgb() {
  // Generates random colours...
  var o = Math.round, r = Math.random, s = 255;
  return 'rgb(' + o(r()*s) + ',' + o(r()*s) + ',' + o(r()*s) + ')';
}


function generateGraphColours(num) {
  // Initializing arrays...
  var borderColours = []
  var backgroundColours = []

  // Adding RGB and RGBA (translucent) values for each colour.
  for (i = 0; i < num; i++) {
    var col = random_rgb();
    borderColours.push(col);
    backgroundColours.push(col.replace(')', ', 0.25)').replace('rgb', 'rgba'));
  }

  // Returning values as an object, its syntactically nicer.
  return {
    borderColours: borderColours,
    backgroundColours: backgroundColours
  }
}


function buildChartDataset(data, type) {
  // Initializing "datasets" and generating graph colours.
  var datasets = []

  if (type == "bar") {
    // Generating dataset for all data needed in the graph.
    for (datum of data) {
      console.log(datum.data);
      var cols = generateGraphColours(7);
      var dataset = {
        "label": datum.label,
        "fill": false,
        "borderWidth": 1,
        "borderColor": cols.borderColours,
        "backgroundColor": cols.backgroundColours,
        "data": datum.data
      }
      datasets.push(dataset);
    }
    return datasets;
  } else if (type == "line") {
    console.log("LINE")

    var cols = generateGraphColours(7);
    var colours = cols.borderColours;

    // Generating dataset for all data needed in the graph.
    x = 0
    for (datum of data) {
      console.log(datum.data);
      var dataset = {
        "label": datum.label,
        "fill": false,
        "borderWidth": 1,
        "borderColor": colours[x],
        "backgroundColor": colours[x],
        "data": datum.data
```

```
      }
      datasets.push(dataset);
      x = x + 1
    }
    return datasets;
  }
}


function changeChart(elmtId, type, labels, data) {

  resetCanvas(elmtId);

  // Grabing the elements we need from the HTML.
  var canvas = document.getElementById(elmtId);
  var context = canvas.getContext('2d');

  // Building correctly stuctured datasets out of the data.
  var datasets = buildChartDataset(data, type);

  // Building the chart.
  var chart = new Chart(context,
    {
      "type": type,
      "data":{
        "labels": labels,
        "datasets": datasets
      },
      "options": {
        "scales": {
          "yAxes": [{"ticks": {"beginAtZero": true}}]
        }
      }
    });
}


function resetCanvas(elmtId) {
  elmtParent = $('#'+elmtId).parent().attr('id');
  $('#'+elmtId).remove();
  $('#'+elmtParent).append('<canvas id="'+elmtId+'"><canvas>');
};
```