The Setup

Two sets of data are attached (these data sets were auto-generated, they may not make sense in the real world). These data sets describe 30 days of impressions, conversions and revenue associate with user accounts.

`users.json`

An array of user objects. Each user has an id, name, avatar and occupation.

`logs.json`

Event information about clicks and impressions. Each item has a type (either 'conversion' or 'impression'), date and time of the event (YYYY-MM-DD HH:MM:SS), user_id of the account this event is related to and revenue collected (will always be 0 for impressions).

Task

- Write a client side application that implements the attached mockup.
- Each card should have the user's avatar, name and occupation. The sum of all conversions, impressions and revenue associated with that user are displayed on the right side of the card. A simple chart of conversions per day is displayed on the left side of the card.
- Some users have an empty avatar url, display the first letter of their first name in place of their avatar.

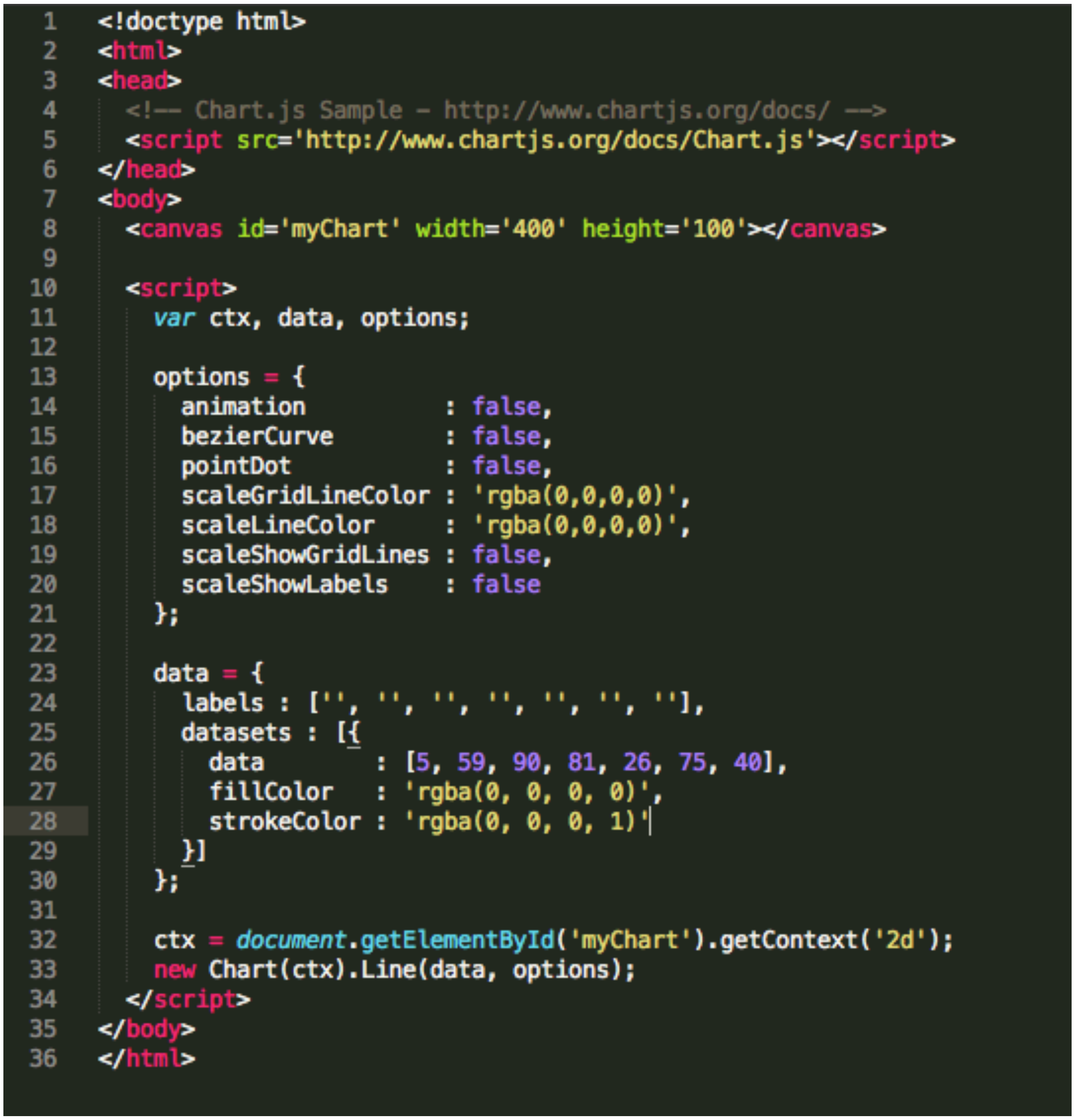
Bonus Items

- Make your app responsive so it looks great across multiple device sizes.
- Use a CSS meta-language such as SASS.
- Use an Javascript MV* framework such as AngularJS
- Write unit tests for testable portions of your code

Notes

- The provided mockup is not set in stone and your implementation doesn't have to be pixel perfect.
- Use libraries where appropriate, you're not expected to reinvent the wheel.
- The next page contains a code sample for charting using Chart.js - you don't have to use Chart.js, this is only provided as a quick-start sample.

Chart.js Sample Implementation



```
<!doctype html>
<html>
<head>
  <!-- Chart.js Sample - http://www.chartjs.org/docs/ -->
  <script src='http://www.chartjs.org/docs/Chart.js'></script>
</head>
<body>
  <canvas id='myChart' width='400' height='100'></canvas>

  <script>
    var ctx, data, options;

    options = {
      animation          : false,
      bezierCurve        : false,
      pointDot           : false,
      scaleGridLineColor : 'rgba(0,0,0,0)',
      scaleLineColor     : 'rgba(0,0,0,0)',
      scaleShowGridLines : false,
      scaleShowLabels    : false
    };

    data = {
      labels : ['', '', '', '', '', '', ''],
      datasets : [{
        data        : [5, 59, 90, 81, 26, 75, 40],
        fillColor   : 'rgba(0, 0, 0, 0)',
        strokeColor : 'rgba(0, 0, 0, 1)'
      }]
    };

    ctx = document.getElementById('myChart').getContext('2d');
    new Chart(ctx).Line(data, options);
  </script>
</body>
</html>
```