```
 <script  type="text/javascript">
    var tasks = <?php echo json_encode($taskArray); ?>;
    var coordinates = [];
    //check to see if input is not started yet or not done yet
    function outsideDateRange(date) {
        var today = new Date();
        var result = false;
        if (date <= today)
            result = true;

        return result;
    }
    //create point sub-arrays to add to coordinates array
    function pushPoint(task, startDate, dueDate, x, y) {
        var startYet = outsideDateRange(startDate);
        var doneYet = outsideDateRange(dueDate);
        var today = new Date();

        point = {};
        point["title"] = task;
        if(!doneYet) {
            if(startYet && x <= today) {
                point["date"] = today.getFullYear() + "-" + ("0" +
                (today.getMonth()+1)).slice(-2) + "-" + ("0" + today.getDate()).slice(-2);
                point["hours"] = y;
            }
            else {
                point["date"] = x.getFullYear() + "-" + ("0" + (x.getMonth()+1)).slice(-2)
                + "-" + ("0" + x.getDate()).slice(-2);
                point["hours"] = y;
            }
            point["due"] = dueDate.getFullYear() + "-" + ("0" +
            (dueDate.getMonth()+1)).slice(-2) + "-" + ("0" + dueDate.getDate()).slice(-2);
            coordinates.push(point);
        }
    }
    //create array with calculated values; create coordinate array
    for (var i=0; i<tasks.length; i++) {
        //perform date calculations and set variables
        var dueDate = new Date(tasks[i].due);
        var startDate = new Date(tasks[i].start);
        var timePerDay = tasks[i].hrsperday;
        var title = tasks[i].title;
        var today = new Date();
        var plus = startDate;
        //set plus to today plus 1 if startDate is in the past
        while (plus < today) {
            plus = d3.timeDay.offset(plus,1);
        }
        //send x,y combinations for range testing and addition to coordinates array
        //pushPoint(title, startDate, dueDate, startDate, timePerDay);  //x = startDate
        //add x,y values for intervals from current date or startDate until dueDate or 30
        days is reached
        while ((plus >= today) && (plus <= dueDate) && (plus <=
        d3.timeDay.offset(today,30))) {
            pushPoint(title, startDate, dueDate, plus, timePerDay);
            plus = d3.timeDay.offset(plus,1);
        }
    }

    //console.log(JSON.stringify(coordinates));

if(coordinates.length != 0) {
    //STACKED BAR CHART
    var svg = dimple.newSvg("#graph", 650, 450);
    var myChart = new dimple.chart(svg, coordinates);
```

```
        myChart.setBounds(65, 50, 505, 310)
        var x = myChart.addCategoryAxis("x", "date");
        //x.tickFormat = "%m-%d-%y";
        x.addOrderRule("date");
        var y = myChart.addMeasureAxis("y", "hours");
        y.tickFormat = ",.2f";
        var s = myChart.addSeries(["due","title"], dimple.plot.bar);
        s.interpolation = "cardinal";
        myChart.addLegend(60, 10, 600, 100, "right");
        myChart.draw();
    }
    </script>
```