```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <style>
            h1 {color:rgb(160, 12, 12); font-size:2em}
            .center {width:100%; text-align:center}
            #tip {font-size:1.2em; font-style:italic}
        </style>

    </head>
<body>
<h1 class="center">PROTOTYPE FOR AN AUTOMATIC FLOWCHART GENERATION</h1>
<p id="tip" class="center">Insert below a new activity in a JSON format into braces. Do not
insert a comma after the last activity.</p>
<div id="form" class="center">
    <form>
        <textarea id="in_json" rows="5" cols="150"></textarea><br>
        <input type="submit" id="sub" value="Create a flowchart">
    </form>
    <div id="graph" class="chart"></div>
</div>
<script src="http://d3js.org/d3.v3.min.js"></script>
<script>
    /*Form Initialization */
    var flowchart_str = '{"activity":"Notes Taking", "MeanIn":"Books Treatises, erasable
    surfaces...", "MeanOut":"Various written Surfaces" ,"Role":"Listeners, Readers"},\
        {"activity":"Notes discarding", "MeanIn":"Notes", "MeanOut":"Nothing or Integrated
        notes" ,"Role":"Compilers, heirs"},\
        {"activity":"Printing", "MeanIn":"Notes", "MeanOut":"Reference books"
        ,"Role":"Printers"}';
    document.getElementById("in_json").value = flowchart_str;

    /* When submitting, create the flowchart*/
    var form = document.querySelector("form");
    form.addEventListener("submit", function(event) {
        document.getElementById("graph").innerHTML = '';
        var flowchart = [];
        var textarea = document.getElementById("in_json").value;
        flowchart = JSON.parse('['+textarea+']');

        var widthBar            = 360;
        var heigthBar           = 70;
        var xBarPosition        = 360;
        var yBarPosition        = 180;
        var xTextPadding        = 20;
        var yTextPadding        = 30;
        var xTextPosition       = xBarPosition + xTextPadding;
        var Padding             = 5;
        var xMeanPadding        = 80;
        var yMeanInPadding      = -5;
        var yMeanOutPadding     = 5;
        var xRole               = 30;
        var textRoleLength      = 200;
        var xRoleArrow          = xRole + textRoleLength;
        var yRolePadding        = heigthBar/2;
        var xVerticalArrow      = xBarPosition + xBarPosition/2;
        var widthEndArrow       = 6;//6
        var heightEndArrow      = 9;//9
        var BasisEndArrow       = widthEndArrow/2;
        var xEndArrow           = xVerticalArrow-heightEndArrow;
        var strokeArrowActivity = 5;
        var strokeArrowMean     = 2;
        var strokeRoleArrow     = 2;
        var y2ArrowPadding      = - strokeArrowActivity*widthEndArrow; //Take the arrow 
        length into account
        var xRoleArrowPadding   = -strokeRoleArrow*widthEndArrow;

        var svg = d3.select('#graph')
            .append('svg')
                .style('width', 2000)
                .style('height',1000);

        svg.append("svg:defs")
            .append("svg:marker")
                .attr("id", "triangle")
                .attr("viewBox", "0 0 "+widthEndArrow+" "+heightEndArrow)
                .attr("refX", 0)
                .attr("refY", BasisEndArrow)
                .attr("markerWidth", heightEndArrow)
                .attr("markerHeight", widthEndArrow)
                .attr("orient", "auto")
            .append("svg:path")
                .attr("d", "M0,0 L0,"+widthEndArrow+" L"+heightEndArrow+","+BasisEndArrow+" 
                z")
                .attr("fill","#a00c0c");

        /*********************SVG Graphics*************************/
        svg.selectAll('rect.activity')
            .data(flowchart)
            .enter()
            .append('rect')
                .attr('class','activity')
                .attr('x', xBarPosition)
                .attr('y', function (d,i) {return ((i+1)*yBarPosition);})
                .attr({width: widthBar, height: heigthBar, 
                style:"fill:#cb842e;stroke-width:"+strokeArrowActivity+";stroke:#a00c0c"});

        /***************Texts and Labels**********************/
        //Text position variables
        var yMeanInPaddingHalf = yMeanInPadding/2;
        var yMeanOutPaddingHalf = yMeanOutPadding/2;
        var heigthBarHalf = heigthBar/2;
        var xMean = xBarPosition + widthBar + xMeanPadding;
        var xMean2 = xBarPosition + widthBar;
        var yMean = yBarPosition + yMeanInPadding;
        var yMean2 = yBarPosition + heigthBarHalf

        svg.selectAll("text.activity")
            .data(flowchart)
            .enter()
            .append('text')
                .text(function (d) {return d.activity;})
                .attr('class','activity')
                .attr('x', xTextPosition)
                .attr('y', function (d,i) {return ((i+1)*yBarPosition + yTextPadding);})
                .style({"font-size":"20px","fill":"#a00c0c"});

        svg.selectAll("text.role")
            .data(flowchart)
            .enter()
            .append('text')
                .text(function (d) {return d.Role;})
                .attr('class','Role')
                .attr('x', xRole)
                .attr('y', function (d,i) {return ((i+1)*yBarPosition +
                yRolePadding+Padding);})
                .style({"font-size":"20px","fill":"#a00c0c"});

        svg.selectAll("text.MeanIn")
            .data(flowchart)
            .enter()
            .append('text')
                .text(function (d) {return d.MeanIn;})
                .attr('class','MeanIn')
                .attr('x', xMean+Padding)
                .attr('y', function (d,i) {return ((i+1)*yBarPosition +
                yMeanInPadding+Padding);})
                .style({"font-size":"20px","fill":"#a00c0c"});

        svg.selectAll("text.MeanOut")
            .data(flowchart)
            .enter()
            .append('text')
                .text(function (d) {return d.MeanOut;})
                .attr('class','MeanOut')
                .attr('x', xMean+Padding)
                .attr('y', function (d,i) {return ((i+1)*yBarPosition + heigthBar +
                yMeanOutPadding+Padding);})
                .style({"font-size":"20px","fill":"#a00c0c"});

        /**********************Arrows and links***************************/
        //Calculation of the theta angle
        var theta = Math.atan((yMean2 - yMean)/(xMean2 - xMean));
        //Calculation of the increments to add to or substract from the effective
        coordinates of the arrows in order to take into account the arrow head length
        var incX = Math.ceil(heightEndArrow*Math.cos(theta));
        var incY = Math.ceil(heightEndArrow*Math.sin(theta));


        svg.selectAll("line.arrows")
            .data(flowchart, function (d,i) {if (i<flowchart.length-2) return i;})
            .enter()
            .append('line')
                .attr('x1', xVerticalArrow)
                .attr('y1', function (d,i) {return ((i+1)*yBarPosition + heigthBar);})
                .attr('x2', xVerticalArrow)
                .attr('y2', function (d,i) {return ((i+2)*yBarPosition + y2ArrowPadding);})

                .attr({style:"fill:#cb842e;stroke-width:"+strokeArrowActivity+";stroke:#a00c0
                c"})
```

```
                .attr("marker-end", "url(#triangle)" );

        svg.selectAll("line.role")
            .data(flowchart)
            .enter()
            .append('line')
                .attr('x1', xRoleArrow)
                .attr('y1', function (d,i) {return ((i+1)*yBarPosition + yRolePadding);})
                .attr('x2', xBarPosition+xRoleArrowPadding)
                .attr('y2', function (d,i) {return ((i+1)*yBarPosition + yRolePadding);})
                .attr({style:"fill:#cb842e;stroke-width:"+strokeRoleArrow+";stroke:#a00c0c"})
                .attr("marker-end", "url(#triangle)" );

        svg.selectAll("line.meanin")
            .data(flowchart)
            .enter()
            .append('line')
                .attr('x1', xMean)
                .attr('y1', function(d,i){return ((i+1)*yBarPosition + yMeanInPadding);})
                .attr('x2', function() {return (xMean2+incX);})
                .attr('y2', function(d,i){return ((i+1)*yBarPosition + heigthBarHalf + 
                incY);})
                .attr({style:"fill:#cb842e;stroke-width:"+strokeArrowMean+";stroke:#a00c0c"})
                .attr("marker-end", "url(#triangle)" );

        svg.selectAll("line.meanout")
            .data(flowchart)
            .enter()
            .append('line')
                .attr('x1', xMean2)
                .attr('y1', function(d,i){return ((i+1)*yBarPosition + heigthBarHalf);})
                .attr('x2', function(){return(xMean-incX);})
                .attr('y2', function(d,i){return ((i+1)*yBarPosition + heigthBar + 
                yMeanOutPadding + incY);})
                .attr({style:"fill:#cb842e;stroke-width:"+strokeArrowMean+";stroke:#a00c0c"})
                .attr("marker-end", "url(#triangle)" );

        event.preventDefault();
    });
</script>
</body>
</html>
```