

```
@{
    ViewBag.Title = "TeamTaskDetailsReport";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<style>
```

```
    path {
        fill: none;
        stroke: black;
    }
```

```
    body {
        background: #fff;
    }
```

```
</style>
```

```
<style>
```

```
    table {
        font-family: arial, sans-serif;
```

```
font-size: small;
border-collapse: collapse;
width: 100%;
}
```

```
td, th {
border: 1px solid #dddddd;
text-align: left;
padding: 8px;
}
```

```
tr:nth-child(even) {
background-color: #dddddd;
}
```

```
.table > tbody > tr > td, .table > tbody > tr > th, .table > tfoot > tr > td, .table > tfoot > tr > th, .table > thead > tr >
td, .table > thead > tr > th, .table td {
padding: 3px !important;
}
</style>
```

```

<link href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css" rel="stylesheet" />
@*<script src="~/Scripts/jquery-1.10.2.min.js"></script>*@
<script src="~/Scripts/d3.v3.min.js"></script>
@section jTableable{
    <script type="text/javascript" language="javascript" src="https://code.jquery.com/jquery-1.12.4.js"></script>
    <script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
    <!-- couldn't find a CDN'd version of vkBeautify, so include it locally -->

}
<form id="svgform" method="post" action="/Home/ExportChart">
    <httpRuntime requestValidationMode="2.0" />

    <input type="hidden" id="output_format" name="output_format" value="">
    <input type="hidden" id="data" name="data" value="">
</form>
<button type="button" class="btn btn-success" id="save_as_svg" onclick="submit_download_form()" value="">

```

Save as SVG

</button>

<div class="panel panel-white m-t-sm " id="divDiscard">

<div class="panel-body row">

<br />

<br />

<div class="col-md-4">

<div class="form-group no-m">

<label>

Location :

</label>

</div>

</div>

<div class="col-md-4">

<div class="form-group no-m">

<label>

Period :

</label>

```
    </div>
  </div>
</div>
<div class="panel-footer">
  <div class="col-md-6">
    <label class="radio-inline"><input type="radio" value="2" name="TaskType" checked="checked">All
Task</label>
    <label class="radio-inline"><input type="radio" value="0" name="TaskType">Dependent Task</label>
    <label class="radio-inline"><input type="radio" value="1" name="TaskType">Independent Task</label>
  </div>
  <button type="button" id="btnReport" class="btn btn-primary" onclick="updateDate()">Generate
Graph</button>
  @*<button type="button" id="create_pdf" value="Generate PDF" class="btn btn-primary"
onclick="updateDate()">Generate PDF</button>*@
  <input type="button" id="create_pdf" value="Generate PDF">

</div>
<div class="panel-body row">

  <table id="table" class="table table-responsive table-bordered ">
```

```
<thead style="background-color:aquamarine;">
  <tr>
    <th>TaskId</th>
    <th>TaskName</th>
    <th>TaskDuedate</th>
    <th>Dependent TaskId</th>
    <th>Dependent Task Duedate</th>
    <th>Dependent TaskName</th>
  </tr>
</thead>
<tbody></tbody>
</table>

</div>
<div class="panel-body row">
  <div id="wrapper" align="center">
    </div>
  </div>

</div>
```

```
<a href="#" id="download">Download</a>
<button type="button" class="btn btn-success" id="save_as_svg" onclick="submit_download_form()" value="">
  Save as SVG
</button>
<script>

(function () {
  var
    form = $('#divDiscard');
  cache_width = form.width(),
  a2 = [850, 1200]; // for a4 size paper width and height

  $('#create_pdf').on('click', function () {
    $('body').scrollTop(0);
    createPDF();
  });
  //create pdf
  function createPDF() {
    getCanvas().then(function (canvas) {
      var
```

```
    img = canvas.toDataURL("image/png"),
    pdf = new jsPDF('p', 'pt', 'A4');
var options = {
    pagesplit: true
};

//pdf.addHTML($('#divDiscard'), options, function () {
//    pdf.save("Graph-html-to-pdf.pdf");
//});
var width = pdf.internal.pageSize.width;
var height = pdf.internal.pageSize.height;
pdf.drawImage(canvas, 'JPEG', 0, 0, width, height);
pdf.save('test11.pdf');
// img1 = canvas.toDataURL("image/png"),

// doc.drawImage(img, 'JPEG', 20, 20);
// doc.drawImage(img1, 'JPEG', 20, 20);
// doc.save('Graph-html-to-pdf.pdf');
form.width(cache_width);
});
```



```
}
```

```
// create canvas object
```

```
function getCanvas() {
```

```
    form.width((a2[0] * 1.33333) - 80).css('max-width', 'none');
```

```
    return html2canvas(form, {
```

```
        margin: 1,
```

```
        filename: 'myfile.pdf',
```

```
        image: { type: 'jpeg', quality: 1 },
```

```
        html2canvas: { scale: 2, logging: true },
```

```
        jsPDF: { unit: 'in', format: 'a4', orientation: 'l' }
```

```
    });
```

```
}
```

```
    }());
```

```
</script>
```

```
<script>
```

```
    /*
```

```
    * jQuery helper plugin for examples and tests
```

```
    */
```

```
(function ($) {  
  $.fn.html2canvas = function (options) {  
    var date = new Date(),  
        $message = null,  
        timeoutTimer = false,  
        timer = date.getTime();  
    html2canvas.logging = options && options.logging;  
    html2canvas.Preload(this[0], $.extend({  
      complete: function (images) {  
        var queue = html2canvas.Parse(this[0], images, options),  
            $canvas = $(html2canvas.Renderer(queue, options)),  
            finishTime = new Date();  
  
        $canvas.css({ position: 'absolute', left: 0, top: 0 }).appendTo(document.body);  
        $canvas.siblings().toggle();  
  
        $(window).click(function () {  
          if (!$canvas.is(':visible')) {  
            $canvas.toggle().siblings().toggle();  
            throwMessage("Canvas Render visible");  
          }  
        });  
      }  
    }));  
  };  
});
```

```

    } else {
        $canvas.siblings().toggle();
        $canvas.toggle();
        throwMessage("Canvas Render hidden");
    }
});
throwMessage('Screenshot created in ' + ((finishTime.getTime() - timer) / 1000) + " seconds<br />", 4000);
}
}, options));

```

```

function throwMessage(msg, duration) {
    window.clearTimeout(timeoutTimer);
    timeoutTimer = window.setTimeout(function () {
        $message.fadeOut(function () {
            $message.remove();
        });
    }, duration || 2000);
    if ($message)
        $message.remove();
    $message = $('<div ></div>').html(msg).css({

```

```
margin: 0,  
padding: 10,  
background: "#000",  
opacity: 0.0,  
position: "fixed",  
top: 10,  
right: 10,  
fontFamily: 'Tahoma',  
color: '#fff',  
fontSize: 12,  
borderRadius: 12,  
width: 'auto',  
height: 'auto',  
textAlign: 'center',  
textDecoration: 'none'  
}).hide().fadeIn().appendTo('body');  
}  
};  
})(jQuery);
```

</script>

<script type="text/javascript">

function updateDate() {

var datanew = [

{

"TaskTitle": "Team1\_Task1",

"TaskNumber": 1,

"TaskDate": "2018-04-30",

"TaskId": 2376,

"extra": "0",

"DependsOnWorkbasketTaskId": 2376,

"DependantTaskDate": "2018-04-30",

"DependantTaskNumber": 1,

"DependantTaskName": "Team1\_Task1",

"color": "rgb(44, 160, 44)",

"TeamName": "5-jun-Team1",

"IsDependantTask": true

```
},  
{  
  "TaskTitle": "Team1_Task2",  
  "TaskNumber": 1,  
  "TaskDate": "2018-05-01",  
  "TaskId": 2377,  
  "extra": "0",  
  "DependsOnWorkbasketTaskId": 2376,  
  "DependantTaskDate": "2018-04-30",  
  "DependantTaskNumber": 1,  
  "DependantTaskName": "Team1_Task1",  
  "color": "rgb(44, 160, 44)",  
  "TeamName": "5-jun-Team1",  
  "IsDependantTask": false  
},  
{  
  "TaskTitle": "Team1_Task3",  
  "TaskNumber": 1,  
  "TaskDate": "2018-05-02",  
  "TaskId": 2378,
```

```
"extra": "0",
"DependsOnWorkbasketTaskId": 2378,
"DependantTaskDate": "2018-05-02",
"DependantTaskNumber": 1,
"DependantTaskName": "Team1_Task3",
"color": "rgb(44, 160, 44)",
"TeamName": "5-jun-Team1",
"IsDependantTask": true
},
{
  "TaskTitle": "Team1_Task4",
  "TaskNumber": 1,
  "TaskDate": "2018-05-03",
  "TaskId": 2379,
  "extra": "0",
  "DependsOnWorkbasketTaskId": 2378,
  "DependantTaskDate": "2018-05-02",
  "DependantTaskNumber": 1,
  "DependantTaskName": "Team1_Task3",
  "color": "rgb(44, 160, 44)",
```

```
"TeamName": "5-jun-Team1",
"IsDependantTask": false
},
{
  "TaskTitle": "Team1_Task5",
  "TaskNumber": 1,
  "TaskDate": "2018-05-04",
  "TaskId": 2380,
  "extra": "0",
  "DependsOnWorkbasketTaskId": 2380,
  "DependantTaskDate": "2018-05-04",
  "DependantTaskNumber": 1,
  "DependantTaskName": "Team1_Task5",
  "color": "rgb(44, 160, 44)",
  "TeamName": "5-jun-Team1",
  "IsDependantTask": true
},
{
  "TaskTitle": "Team1_Task6",
  "TaskNumber": 1,
```



```
"TaskDate": "2018-05-07",  
"TaskId": 2381,  
"extra": "0",  
"DependsOnWorkbasketTaskId": 2380,  
"DependantTaskDate": "2018-05-04",  
"DependantTaskNumber": 1,  
"DependantTaskName": "Team1_Task5",  
"color": "rgb(44, 160, 44)",  
"TeamName": "5-jun-Team1",  
"IsDependantTask": false  
},  
{  
  "TaskTitle": "Team1_Task6",  
  "TaskNumber": 1,  
  "TaskDate": "2018-05-07",  
  "TaskId": 2381,  
  "extra": "0",  
  "DependsOnWorkbasketTaskId": 2382,  
  "DependantTaskDate": "2018-05-08",  
  "DependantTaskNumber": 1,
```

```
"DependantTaskName": "Team1_Task7",  
"color": "rgb(214, 39, 40)",  
"TeamName": "5-jun-Team1",  
"IsDependantTask": false  
},  
{  
  "TaskTitle": "Team1_Task7",  
  "TaskNumber": 1,  
  "TaskDate": "2018-05-08",  
  "TaskId": 2382,  
  "extra": "0",  
  "DependsOnWorkbasketTaskId": 2382,  
  "DependantTaskDate": "2018-05-08",  
  "DependantTaskNumber": 1,  
  "DependantTaskName": "Team1_Task7",  
  "color": "rgb(44, 160, 44)",  
  "TeamName": "5-jun-Team1",  
  "IsDependantTask": true  
},  
{
```

```
"TaskTitle": "Team2_Task2",  
"TaskNumber": 3,  
"TaskDate": "2018-05-01",  
"TaskId": 2384,  
"extra": "1",  
"DependsOnWorkbasketTaskId": 2383,  
"DependantTaskDate": "2018-05-14",  
"DependantTaskNumber": 3,  
"DependantTaskName": "Team2_Task1",  
"color": "rgb(214, 39, 40)",  
"TeamName": "5-jun-Team2",  
"IsDependantTask": false  
},  
{  
  "TaskTitle": "Team2_Task2",  
  "TaskNumber": 3,  
  "TaskDate": "2018-05-01",  
  "TaskId": 2384,  
  "extra": "1",  
  "DependsOnWorkbasketTaskId": 2384,
```

```
"DependantTaskDate": "2018-05-01",  
"DependantTaskNumber": 3,  
"DependantTaskName": "Team2_Task2",  
"color": "rgb(44, 160, 44)",  
"TeamName": "5-jun-Team2",  
"IsDependantTask": true  
},  
{  
  "TaskTitle": "Team2_Task4",  
  "TaskNumber": 3,  
  "TaskDate": "2018-05-02",  
  "TaskId": 2386,  
  "extra": "1",  
  "DependsOnWorkbasketTaskId": 2384,  
  "DependantTaskDate": "2018-05-01",  
  "DependantTaskNumber": 3,  
  "DependantTaskName": "Team2_Task2",  
  "color": "rgb(44, 160, 44)",  
  "TeamName": "5-jun-Team2",  
  "IsDependantTask": false
```

```
},  
{  
  "TaskTitle": "Team2_Task1",  
  "TaskNumber": 3,  
  "TaskDate": "2018-05-14",  
  "TaskId": 2383,  
  "extra": "1",  
  "DependsOnWorkbasketTaskId": 2383,  
  "DependantTaskDate": "2018-05-14",  
  "DependantTaskNumber": 3,  
  "DependantTaskName": "Team2_Task1",  
  "color": "rgb(44, 160, 44)",  
  "TeamName": "5-jun-Team2",  
  "IsDependantTask": true  
},  
{  
  "TaskTitle": "Team3_Task2",  
  "TaskNumber": 5,  
  "TaskDate": "2018-05-01",  
  "TaskId": 2389,
```

```
"extra": "2",  
"DependsOnWorkbasketTaskId": 2389,  
"DependantTaskDate": "2018-05-01",  
"DependantTaskNumber": 5,  
"DependantTaskName": "Team3_Task2",  
"color": "rgb(44, 160, 44)",  
"TeamName": "5-jun-Team3",  
"IsDependantTask": true  
},  
{  
  "TaskTitle": "Team3_Task1",  
  "TaskNumber": 5,  
  "TaskDate": "2018-05-03",  
  "TaskId": 2388,  
  "extra": "2",  
  "DependsOnWorkbasketTaskId": 2389,  
  "DependantTaskDate": "2018-05-01",  
  "DependantTaskNumber": 5,  
  "DependantTaskName": "Team3_Task2",  
  "color": "rgb(44, 160, 44)",
```

```
"TeamName": "5-jun-Team3",
"IsDependantTask": false
},
{
  "TaskTitle": "Team4_Task2",
  "TaskNumber": 6,
  "TaskDate": "2018-05-10",
  "TaskId": 2391,
  "extra": "3",
  "DependsOnWorkbasketTaskId": 2390,
  "DependantTaskDate": "2018-05-25",
  "DependantTaskNumber": 6,
  "DependantTaskName": "Team4_Task1",
  "color": "rgb(214, 39, 40)",
  "TeamName": "5-jun-Team4",
  "IsDependantTask": false
},
{
  "TaskTitle": "Team4_Task1",
  "TaskNumber": 6,
```

```
"TaskDate": "2018-05-25",  
"TaskId": 2390,  
"extra": "3",  
"DependsOnWorkbasketTaskId": 2390,  
"DependantTaskDate": "2018-05-25",  
"DependantTaskNumber": 6,  
"DependantTaskName": "Team4_Task1",  
"color": "rgb(44, 160, 44)",  
"TeamName": "5-jun-Team4",  
"IsDependantTask": true  
},  
{  
  "TaskTitle": "Team5_Task3",  
  "TaskNumber": 7,  
  "TaskDate": "2018-05-02",  
  "TaskId": 2394,  
  "extra": "4",  
  "DependsOnWorkbasketTaskId": 2394,  
  "DependantTaskDate": "2018-05-02",  
  "DependantTaskNumber": 7,
```



```
"DependantTaskName": "Team5_Task3",  
"color": "rgb(44, 160, 44)",  
"TeamName": "5-jun-Team5",  
"IsDependantTask": true  
},  
{  
  "TaskTitle": "Team5_Task6",  
  "TaskNumber": 7,  
  "TaskDate": "2018-05-18",  
  "TaskId": 2396,  
  "extra": "4",  
  "DependsOnWorkbasketTaskId": 2394,  
  "DependantTaskDate": "2018-05-02",  
  "DependantTaskNumber": 7,  
  "DependantTaskName": "Team5_Task3",  
  "color": "rgb(44, 160, 44)",  
  "TeamName": "5-jun-Team5",  
  "IsDependantTask": false  
},  
{
```

```
"TaskTitle": "Team5_Task8",
"TaskNumber": 7,
"TaskDate": "2018-05-22",
"TaskId": 2398,
"extra": "4",
"DependsOnWorkbasketTaskId": 2394,
"DependantTaskDate": "2018-05-02",
"DependantTaskNumber": 7,
"DependantTaskName": "Team5_Task3",
"color": "rgb(44, 160, 44)",
"TeamName": "5-jun-Team5",
"IsDependantTask": false
},
{
  "TaskTitle": "Team5_Task7",
  "TaskNumber": 7,
  "TaskDate": "2018-05-29",
  "TaskId": 2397,
  "extra": "4",
  "DependsOnWorkbasketTaskId": 2397,
```

```
"DependantTaskDate": "2018-05-29",
"DependantTaskNumber": 7,
"DependantTaskName": "Team5_Task7",
"color": "rgb(44, 160, 44)",
"TeamName": "5-jun-Team5",
"IsDependantTask": true
},
{
  "TaskTitle": "Team5_Task1",
  "TaskNumber": 7,
  "TaskDate": "2018-06-04",
  "TaskId": 2392,
  "extra": "4",
  "DependsOnWorkbasketTaskId": 2397,
  "DependantTaskDate": "2018-05-29",
  "DependantTaskNumber": 7,
  "DependantTaskName": "Team5_Task7",
  "color": "rgb(44, 160, 44)",
  "TeamName": "5-jun-Team5",
  "IsDependantTask": false
```

```
},  
{  
  "TaskTitle": "Team8_Task1",  
  "TaskNumber": 8,  
  "TaskDate": "2018-05-28",  
  "TaskId": 2401,  
  "extra": "5",  
  "DependsOnWorkbasketTaskId": 2401,  
  "DependantTaskDate": "2018-05-28",  
  "DependantTaskNumber": 8,  
  "DependantTaskName": "Team8_Task1",  
  "color": "rgb(44, 160, 44)",  
  "TeamName": "5-jun-Team8",  
  "IsDependantTask": true  
},  
{  
  "TaskTitle": "Team8_Task2",  
  "TaskNumber": 8,  
  "TaskDate": "2018-05-31",  
  "TaskId": 2402,
```

```
"extra": "5",
"DependsOnWorkbasketTaskId": 2401,
"DependantTaskDate": "2018-05-28",
"DependantTaskNumber": 8,
"DependantTaskName": "Team8_Task1",
"color": "rgb(44, 160, 44)",
"TeamName": "5-jun-Team8",
"IsDependantTask": false
},
{
  "TaskTitle": "Team6_Task1",
  "TaskNumber": 9,
  "TaskDate": "2018-05-02",
  "TaskId": 2403,
  "extra": "6",
  "DependsOnWorkbasketTaskId": 2404,
  "DependantTaskDate": "2018-05-02",
  "DependantTaskNumber": 11,
  "DependantTaskName": "Team6_Task2",
  "color": "rgb(44, 160, 44)",
```

```
"TeamName": "5-jun-Team6",
"IsDependantTask": false
},
{
  "TaskTitle": "Team6_Task3",
  "TaskNumber": 10,
  "TaskDate": "2018-05-02",
  "TaskId": 2405,
  "extra": "6",
  "DependsOnWorkbasketTaskId": 2403,
  "DependantTaskDate": "2018-05-02",
  "DependantTaskNumber": 9,
  "DependantTaskName": "Team6_Task1",
  "color": "rgb(214, 39, 40)",
  "TeamName": "5-jun-Team6",
  "IsDependantTask": false
},
{
  "TaskTitle": "Team6_Task1",
  "TaskNumber": 10,
```

```
"TaskDate": "2018-05-02",
"TaskId": 2403,
"extra": "6",
"DependsOnWorkbasketTaskId": 2403,
"DependantTaskDate": "2018-05-02",
"DependantTaskNumber": 9,
"DependantTaskName": "Team6_Task1",
"color": "rgb(44, 160, 44)",
"TeamName": "5-jun-Team6",
"IsDependantTask": true
},
{
  "TaskTitle": "Team6_Task2",
  "TaskNumber": 11,
  "TaskDate": "2018-05-02",
  "TaskId": 2404,
  "extra": "6",
  "DependsOnWorkbasketTaskId": 2404,
  "DependantTaskDate": "2018-05-02",
  "DependantTaskNumber": 11,
```

```
    "DependantTaskName": "Team6_Task2",  
    "color": "rgb(44, 160, 44)",  
    "TeamName": "5-jun-Team6",  
    "IsDependantTask": true  
  }  
]
```

```
var colorData = [];  
//here add a filter  
if (true) {  
  var Period = 0;  
  var locationId = 0;  
  var IndependentTask = "";  
  
  var chart = d3.select("#wrapper");  
  chart.selectAll("svg").remove();  
  debugger;  
  showPleaseWait();  
  // <a href="~/Scripts/data.json">~/Scripts/data.json</a>
```



```

    d3.json('../Home/Graphdata?LocationId=' + locationId + '&PeriodId=' + Period + '&IndependentTask=' +
    $("input[name='TaskType']:checked").val(), function (data) {
        // d3.json('../Scripts/data.json', function (data) {
            hidePleaseWait();
            if (true) {
                //var Data = JSON.parse(data);
                var Data = datanew;
                if (Data.length == 0) {
                    Command: toastr["error"]("Record Not Present related Location and Period ");
                    return false;
                }
            }
            else {
                //legend TeamNames
                var TeamNames = [];
                var TaskEndDates = [];

                var rows = ""
                $.each(Data, function (index, value) {
                    if ((value.TaskTitle == value.DependantTaskName && value.TaskId ==
                    value.DependsOnWorkbasketTaskId) || IndependentTask == true) {

```

```

        value.DependantTaskName = "";
        value.DependsOnWorkbasketTaskId = "";
        value.DependantTaskDate = "";
    }

    if (value.DependsOnWorkbasketTaskId != "") {
        if (value.DependsOnWorkbasketTaskId == null) {
            value.DependantTaskName = "";
            value.DependsOnWorkbasketTaskId = "";
            value.DependantTaskDate = "";
        }
        rows += "<tr><td>" + value.TaskId + "</td><td>" + value.TaskTitle
            + "</td><td>" + value.TaskDate + "</td><td>" + value.DependsOnWorkbasketTaskId
            + "</td><td>" + value.DependantTaskDate + "</td><td>" + value.DependantTaskName +
"</td></tr>";
    }

    if ($.inArray(value.TeamName, TeamNames) == -1) {

```

```

        TeamNames.push(value.TeamName);
    }

    if ($.inArray(value.TaskDate, TaskEndDates) == -1) {
        TaskEndDates.push(value.TaskDate);
    }

});

debugger;
if (rows != "") {
    $("#table tbody").empty();
    $("#table").removeClass("hidden");
    $("#table tbody").append(rows);
}
$('#table').DataTable();
//function sortDates(a, b) {
//    return a.getDate() - b.geteDate();
//}

```

```
var sorted = TaskEndDates.sort();
var minDate = sorted[0];
var maxDate = sorted[sorted.length - 1];
debugger;
var format = d3.time.format("%Y-%m-%d");
minDate = minDate.split('T')[0];
maxDate = maxDate.split('T')[0];
```

```
console.log(minDate);
console.log(maxDate);
```

```
// x axis scal min and maximum Data ...if you to do this dynamaic send the Data for json assign here
//var dateMin = format.parse("20/02/2018");
//var dateMax = format.parse("11/04/2018");
var dateMin = new Date();
var dateMax = new Date();
dateMax = format.parse(maxDate);
dateMin = format.parse(minDate);
var margin = {
  top: 20,
```

```
    right: 20,  
    bottom: 30,  
    left: 30  
  },  
  width = 850 - margin.left - margin.right,  
  height = 950 - margin.top - margin.bottom;
```

```
//add for teamwise color legend  TODO : need to working
```

```
dateMin.setDate(dateMin.getDate() - 1);  
var newMinDate = dateMin.toDateString();  
newMinDate = new Date(Date.parse(newMinDate));
```

```
dateMax.setDate(dateMax.getDate() + 10);  
var newMaxDate = dateMax.toDateString();  
newMaxDate = new Date(Date.parse(newMaxDate));
```

```
///
```

```
var xValue = function (d) {  
    return format.parse(d.TaskDate);  
}, // Data -> value  
  
xScale = d3.time.scale().domain([dateMin, dateMax]).range([0, width], 3.5), // value -> display  
xMap = function (d) {  
    // console.log(xScale.domain()[1].toISOString());  
    return xScale(xValue(d));  
}, // Data -> display  
xAxis = d3.svg.axis().scale(xScale).orient("bottom");  
var yValue = function (d) {  
    return d.TaskNumber;  
}, // Data -> value  
yScale = d3.scale.linear().range([height, 0]), // value -> display  
yMap = function (d) {  
    return yScale(yValue(d));  
}, // Data -> display  
yAxis = d3.svg.axis().scale(yScale).orient("left");  
// setup fill color  
var cValue = function (d) {
```

```
    return d.ownership;
  },
  color = d3.scale.category10();

// add the graph canvas to the body of the webpage
var svg = d3.select("#wrapper").append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
  .append("g")
  .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

svg.append("rect")
  .attr("width", "100%")
  .attr("height", "100%")
  .attr("fill", "#F7F7F7");

// add the tooltip area to the webpage
var tooltip = d3.select("#wrapper").append("div")
  .attr("class", "tooltip")
  .style("opacity", 0);
```

```
// don't want dots overlapping axis, so add in buffer to Data domain
```

```
xScale.domain([dateMin, dateMax]);
```

```
yScale.domain([d3.min(Data, yValue) - 1, d3.max(Data, yValue) + 10]);
```

```
//x-axis name
```

```
svg.append("g")
```

```
  .attr("class", "x axis")
```

```
  .attr("transform", "translate(0," + height + ")")
```

```
  .call(xAxis)
```

```
  .append("text")
```

```
  .attr("class", "label")
```

```
  .attr("x", width)
```

```
  .attr("y", -6)
```

```
  .style("text-anchor", "end")
```

```
  .text("Task End Date");
```

```
// y-axis name
```

```
svg.append("g")
```



```
.attr("class", "y axis")
.call(yAxis)
.append("text")
.attr("class", "label")
.attr("transform", "rotate(-90)")
.attr("y", 6)
.attr("dy", ".71em")
.style("text-anchor", "end");
//.text();

// draw dots
svg.selectAll(".dot")
.append("text")
.data(Data)
.attr("dx", 12)
.attr("dy", ".35em")
.text(function (d) { return d.DependsOnWorkbasketTaskId });

svg.selectAll(".dot")
.data(Data)
```

```

.enter().append("circle")
.attr("class", "dot")
.attr("r", 3.5)
.attr("cx", xMap)
.attr("cy", yMap)
.style("fill", function (d) {
    return color(d.extra);
}).append('text')
.text(function (d) {
    return d.DependsOnWorkbasketTaskId
})
.append("title")
.text(function (d) {
    if (d.TaskId == d.DependsOnWorkbasketTaskId && d.TaskTitle == d.DependantTaskName) {
        d.DependsOnWorkbasketTaskId = "";
        d.DependantTaskName = "";
    }
    return "(" + d.TaskId + ") " + d.TaskTitle + "(" + d.TaskDate + ") DependentTaskID:- " +
d.DependsOnWorkbasketTaskId)
})

```

```

        // .on("mouseover", function (d) {
        //     tooltip.transition()
        //         .duration(200)
        //         .style("opacity", 0.9);
        //     tooltip.html("<strong>Task No:</strong> " + d.TaskId + "<br/>" + "<strong>Task Name:</strong>"
" + d.TaskTitle + "<br/>" + "<strong>Duedate: </strong>" + d.TaskDate)
        //     .style("left", (d3.event.pageX + 5) + "px")
        //     .style("top", (d3.event.pageY - 28) + "px");
        // })
        .on("mouseout", function (d) {
            tooltip.transition()
                .duration(500)
                .style("opacity", 0);
        })
        .attr('data-title', function (e) {
            return e.TaskTitle;
        })
        .attr('data-value', function (e) {
            return e.TaskNumber;
        })
    }

```

```
.attr('data-date', function (e) {  
    return e.TaskDate;  
})  
.attr('data-sqft', function (e) {  
    return e.extra;  
});
```

```
//draw line  
for (var item in Data) {  
    //console.log(Data[item]["TaskNumber"]);  
    if (Data[item]["DependsOnWorkbasketTaskId"] != "") {  
  
        svg.append("line")  
            .attr("x1", xScale(format.parse(Data[item]["TaskDate"])))  
            .attr("y1", yScale(Data[item]["TaskNumber"]))  
            .attr("x2", xScale(format.parse(Data[item]["DependantTaskDate"])))  
            .attr("y2", yScale(Data[item]["DependantTaskNumber"]))  
            .attr("stroke-width", 1)  
            .attr("stroke", Data[item]["color"]);
```

```
}  
}
```

```
// draw legend  
var legend = svg.selectAll(".legend")  
  .data(color.domain())  
  .enter().append("g")  
  .attr("class", "legend")  
  .attr("transform", function (d, i) { return "translate(0," + i * 20 + ")"; });
```

```
// draw legend colored rectangles  
legend.append("rect")  
  .attr("x", width - 18)  
  .attr("width", 18)  
  .attr("height", 18)  
  .style("fill", color);
```

```
// draw legend text  
legend.append("text")  
  .attr("x", width - 24)
```

```

        .attr("y", 9)
        .attr("dy", ".35em")
        .style("text-anchor", "end")
        .text(function (d) {
            return (TeamNames[d]);
        })
        var title = 'Location - ' + '123, ' + 'Period - ' + '2018-2019, ' + 'Task type - ' +
$("input[name='TaskType']:checked").parent().text();
        svg.append("text")
            .attr("x", (width / 2))
            .attr("y", 3 - (margin.top / 2))
            .attr("text-anchor", "middle")
            .style("font-size", "16px")
            .style("text-decoration", "underline")
            .text(title);
    }
}
else {
    Command: toastr["error"](data.Message);
    return false;
}

```

```
    }  
  })  
}  
else {  
  Command: toastr["error"]("Please Select Location and Period");  
  return false;  
}  
};
```

</script>

<script>

```
function submit_download_form(output_format) {  
  // Get the d3js SVG element  
  var tmp = document.getElementById("wrapper");  
  var svg = tmp.getElementsByTagName("svg")[0];  
  // Extract the data as SVG text string  
  var svg_xml = new XMLSerializer().serializeToString(svg);  
  // Submit the <FORM> to the server.  
  // The result will be an attachment file to download.
```

```
var form = document.getElementById("svgform");
form['output_format'].value = output_format;
form['data'].value = svg_xml;
form.submit();
}
```

```
function showPleaseWait() {  
    var modalLoading = '<div class="modal" id="pleaseWaitDialog" data-backdrop="static" data-keyboard="false"  
role="dialog">\n<div class="modal-dialog">\n<div class="modal-content">\n<div class="modal-body">\n<div>\n<h4 class="text-center">Please wait...</h4>\n</div>\n</div>\n</div>\n</div>\n</div>\n'
```



```
</div>';  
    $(document.body).append(modalLoading);  
    $("#pleaseWaitDialog").modal("show");  
}  
  
function hidePleaseWait() {  
    $("#pleaseWaitDialog").modal("hide");  
}  
</script>
```