* **44-599: Introduction to Data Visualization**
* **Worksheet**

**Worksheet for visualizing Enclosure Diagram using D3.js.**

**Objective:** The main objective of this worksheet is to generate **Enclosure Diagram** using **D3.js**.

**Bitbucket Link:**

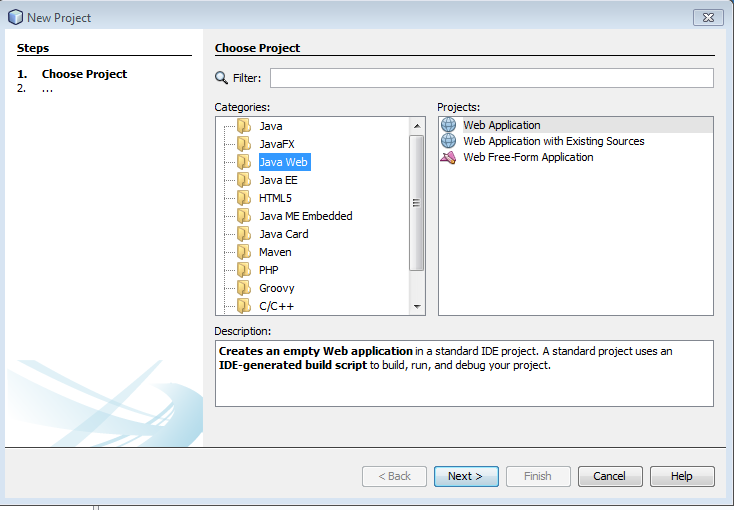
**Data set:**

****

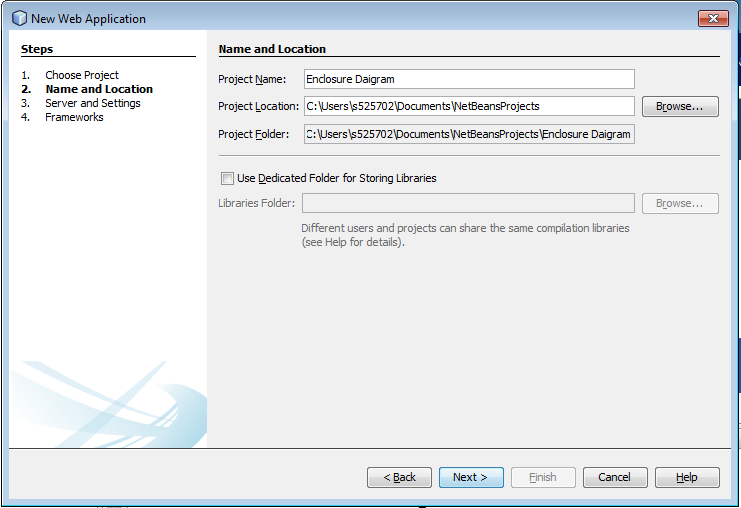
****

**Visualizing data using D3.js:**

1. Download NetBeans from <https://netbeans.org/downloads/>
2. Create a web application project in NetBeans



Select Web Application from the right pan. Click Next



Select “Project Name” as EnclosureDiagram. Click Next and Finish

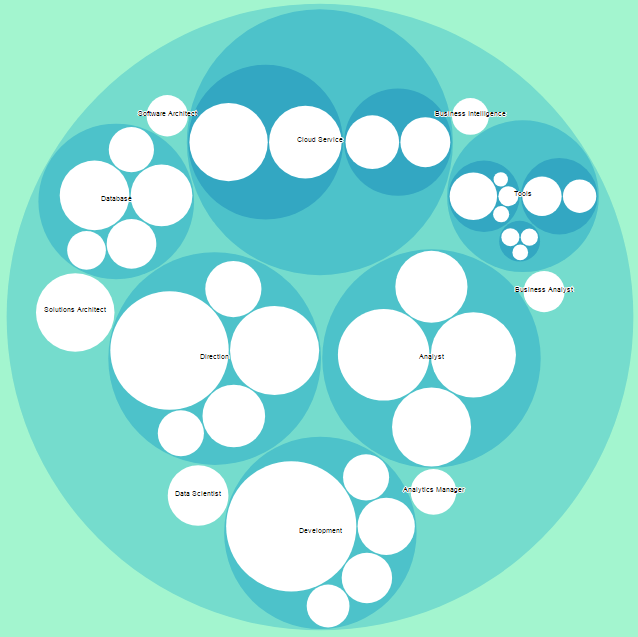
1. Include following code in index.html file

This code snippet is for goal 1

|  |
| --- |
| <html>  <head>  <title>eNCLOSURE dAIGRAM</title>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <style>  .node {  cursor: pointer;  }  .node:hover {  stroke: #000;  stroke-width: 1.5px;  }  .node--leaf {  fill: white;  }  .label {  font: 11px "Helvetica Neue", Helvetica, Arial, sans-serif;  text-anchor: middle;  text-shadow: 0 1px 0 #fff, 1px 0 0 #fff, -1px 0 0 #fff, 0 -1px 0 #fff;  }  .label,  .node--root,  .node--leaf {  pointer-events: none;  }    </style>  </head>  <body>  <svg width="960" height="960"></svg>  <script src="https://d3js.org/d3.v4.min.js"></script>    <script>  var svg = d3.select("svg"),  margin = 20,  diameter = +svg.attr("width"),  g = svg.append("g").attr("transform", "translate(" + diameter / 2 + "," + diameter / 2 + ")");  var color = d3.scaleLinear()  .domain([-1, 5])  .range(["hsl(152,80%,80%)", "hsl(228,30%,40%)"])  .interpolate(d3.interpolateHcl);  var pack = d3.pack()  .size([diameter - margin, diameter - margin])  .padding(2);  /\*Creates a tip variable\*/  // var tip = d3.tip()  // .attr('class', 'd3-tip')  // .offset([-10, 0])  // .html(function (d) {  // return +d.name + " " + d.size + " count";  // })  d3.json("presentation.json", function (error, root) {  if (error)  throw error;  root = d3.hierarchy(root)  .sum(function (d) {  return d.size;  })  .sort(function (a, b) {  return b.value - a.value;  });  var focus = root,  nodes = pack(root).descendants(),  view;  var circle = g.selectAll("circle")  .data(nodes)  .enter().append("circle")  .attr("class", function (d) {  return d.parent ? d.children ? "node" : "node node--leaf" : "node node--root";  })  .style("fill", function (d) {  return d.children ? color(d.depth) : null;  })  .on("click", function (d) {  if (focus !== d)  zoom(d), d3.event.stopPropagation();  });  var text = g.selectAll("text")  .data(nodes)  .enter().append("text")  .attr("class", "label")  .style("fill-opacity", function (d) {  return d.parent === root ? 1 : 0;  })  .style("display", function (d) {  return d.parent === root ? "inline" : "none";  })  .text(function (d) {  return d.data.name ;  });  var node = g.selectAll("circle,text");  svg  .style("background", color(-1))    .on("click", function () {  zoom(root);  });  zoomTo([root.x, root.y, root.r \* 2 + margin]);    function zoom(d) {  var focus0 = focus;  focus = d;  var transition = d3.transition()  .duration(d3.event.altKey ? 7500 : 750)  .tween("zoom", function (d) {  var i = d3.interpolateZoom(view, [focus.x, focus.y, focus.r \* 2 + margin]);  return function (t) {  zoomTo(i(t));  };  });  transition.selectAll("text")  .filter(function (d) {  return d.parent === focus || this.style.display === "inline";  })  .style("fill-opacity", function (d) {  return d.parent === focus ? 1 : 0;  })  .on("start", function (d) {  if (d.parent === focus)  this.style.display = "inline";  })  .on("end", function (d) {  if (d.parent !== focus)  this.style.display = "none";  });  }  function zoomTo(v) {  var k = diameter / v[2];  view = v;  node.attr("transform", function (d) {  return "translate(" + (d.x - v[0]) \* k + "," + (d.y - v[1]) \* k + ")";  });  circle.attr("r", function (d) {  return d.r \* k;  });  }  });  </script>  </body>  </html> |



Include the above given JSON file as external data and run the project to visualize result in browser which looks as follows



1. Create another web project and use below data in index.html to visualize Goal 2

|  |
| --- |
| <html>  <head>  <title>Goal 2</title>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">    </head>  <body>  <svg width="960" height="960" font-family="sans-serif" font-size="20" text-anchor="middle"></svg>  <script src="https://d3js.org/d3.v4.min.js"></script>  <script>  var svg = d3.select("svg"),  width = +svg.attr("width"),  height = +svg.attr("height");  var format = d3.format(",d");  var color = d3.scaleOrdinal(d3.schemeCategory10);  var pack = d3.pack()  .size([width, height])  .padding(1.5);  d3.csv("presentation2.csv", function (d) {  d.value = +d.value;  if (d.value)  return d;  }, function (error, classes) {  if (error)  throw error;  var root = d3.hierarchy({children: classes})  .sum(function (d) {  return d.value;  })  .each(function (d) {  if (id = d.data.id) {  var id, i = id.lastIndexOf(".");  d.id = id;  d.package = id.slice(0, i);  d.class = id.slice(i + 1);  }  });  var node = svg.selectAll(".node")  .data(pack(root).leaves())  .enter().append("g")  .attr("class", "node")  .attr("transform", function (d) {  return "translate(" + d.x + "," + d.y + ")";  });  node.append("circle")  .attr("id", function (d) {  return d.id;  })  .attr("r", function (d) {  return d.r;  })  .style("fill", function (d) {  return color(d.package);  });  node.append("clipPath")  .attr("id", function (d) {  return "clip-" + d.id;  })  .append("use")  .attr("xlink:href", function (d) {  return "#" + d.id;  });  node.append("text")  .attr("clip-path", function (d) {  return "url(#clip-" + d.id + ")";  })  .selectAll("tspan")  .data(function (d) {  return d.class.split(/(?=[A-Z][^A-Z])/g);  })  .enter().append("tspan")  .attr("x", 0)  .attr("y", function (d, i, nodes) {  return 13 + (i - nodes.length / 2 - 0.5) \* 10;  })  .text(function (d) {  return d;  });  node.append("title")  .text(function (d) {  return d.id + "\n" + format(d.value);  });  });  </script>  </body>  </html> |



Include the above CSV file as external data and run the project to visualize result in browser which looks as follows

