# KV3 - Izrada prototipne vizualizacije podataka

## Osnovne funkcionalnosti i ponašanja

* + 1. Ključne funkcionalnosti koje će biti potrebne za prikaz podataka:
* Izrada stupčastog/linijskog grafikona za prikaz knjiga pročitanih po mjesecima za jednu ili više godina
* Izrada elementa za prikaz ukupnog broja pročitanih stranica za odabranu godinu ili za više odabranih godina
* Izrada pie charta za prikaz žanrova pročitanih u jednoj godini ili više godina
* Izrada donut charta za prikaz ukupnog broja pročitanih knjiga
  + 1. Osnovne vrste ponašanja:
* Prikaz različitih grafikona nakon odabira godine/a
* Prikaz liste pročitanih knjiga za odabrani mjesec
* Prikaz ukupnog broja pročitanih stranica za odabranu godinu ili za više odabranih godina
  + 1. Elementi I interakcije:
       1. Elementi s kojima će korisnici moći vršiti interakciju:
* Godine od 2014. Do 2024.
* Stupci u grafikonu
* Lista pročitanih knjiga
  + - 1. Definiranje interakcije između korisnika i vizualizacije s pripadajućim opisom:
* Odabir godine/a prikazuje različite grafikone vezane za čitanje u odabranoj godini odnosno godinama
* Klikom na stupac u grafikonu prikazuje se lista pročitanih knjiga u tom mjesecu
* Klikom na naziv knjige u listi pročitanih knjiga prikazuju se detalji o knjizi

## Napredne funkcionalnosti i ponašanja:

* + 1. Identificirati napredne funkcionalnosti koje će biti potrebne za analizu podataka:
* Omogućavanje izbora više godina za usporednu analizu svih elemenata odosno usporedba žanrova, broja pročitanih kniga po mjesecima I općenito te ukupan broj pročitanih knjiga u određenom razdoblju
  + 1. Definirati napredne vrste ponašanja:
* Klikom na pojedinu knjigu prikazuju se dodatne informacije o knjizi
* Različite animacije za prikaz grafova
  + 1. Definirati interakcije koje će omogućiti korisnicima dodatnu analizu podataka:
* Hover na linijskom grafu prikazuje broj pročitanih knjiga
* Klikom na pojedinu knjigu prikazuju se dodatne informacije o knjizi

## Implementacija osnovnih funkcionalnosti

* + 1. Izraditi kod koji omogućava prethodno definirane funkcionalnosti:
* Izrada stupčastog/linijskog grafikona za prikaz knjiga pročitanih po mjesecima za jednu ili više godina:

function drawChart(year) {

      d3.csv("books.csv").then(function(data) {

        d3.select("#chart").selectAll("\*").remove();

        if (!year) return;

        var filteredData = data.filter(function(d) {

          return d.Year == year;

        });

        var months = d3.range(1, 13);

        var booksReadPerMonth = months.map(function(month) {

          return {

            month: month,

            booksRead: filteredData.filter(function(d) {

              return +d.DateRead.split("/")[1] === month;

            }).length

          };

        });

        var margin = { top: 50, right: 50, bottom: 60, left: 50 },

            width = 1000 - margin.left - margin.right,

            height = 450 - margin.top - margin.bottom;

        var x = d3.scaleBand()

            .domain(months.map(String))

            .range([0, width])

            .padding(0.1);

        var y = d3.scaleLinear()

            .domain([0, d3.max(booksReadPerMonth, function(d) { return d.booksRead; })])

            .nice()

            .range([height, 0]);

        var xAxis = d3.axisBottom(x)

          .tickFormat(function(d) {

            var monthNames = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"];

            return monthNames[parseInt(d) - 1];

          });

        var yAxis = d3.axisLeft(y)

          .ticks(5);

        d3.select("#chart").selectAll("\*").remove();

        var svg = d3.select("#chart").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("g")

            .attr("class", "x axis")

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

            .call(xAxis)

            .selectAll("text")

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

            .attr("dx", "-.8em")

            .attr("dy", ".15em")

            .attr("transform", "rotate(-65)");

        svg.append("g")

            .attr("class", "y axis")

            .call(yAxis);

        var barchart = svg.selectAll("rect")

          .data(booksReadPerMonth)

          .enter()

          .append("rect")

          .attr("x", function(d) { return x(String(d.month)); })

          .attr("y", function(d) { return y(d.booksRead); })

          .attr("height", function(d) { return height - y(d.booksRead); })

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

          .attr("class", function(d) { return "bar bar-" + String(d.month); })

          .on("click", function(d) {

            var clickedMonth = d3.select(this).attr("class").split(" ")[1].split("-")[1];

            showBooksByMonth(clickedMonth, year);

          });

        var totalBooks = d3.sum(booksReadPerMonth, function(d) { return d.booksRead; });

        svg.append("text")

        .attr("class", "total-books")

        .attr("x", width / 2)

        .attr("y", "-10px")

        .attr("text-anchor", "middle")

        .text("Total books read: " + totalBooks)

        .attr("font-weight", "600")

        .attr("font-size", "1.4em");

      });

    }

* Izrada elementa za prikaz ukupnog broja pročitanih stranica za odabranu godinu ili za više odabranih godina:
* Izrada pie charta za prikaz žanrova pročitanih u jednoj godini ili više godina:

function drawPieChart(year) {

    d3.csv("books.csv").then(function(data) {

      d3.select("#pie-chart").selectAll("\*").remove();

      if (!year) return;

      var filteredData = data.filter(function(d) {

          return d.Year == year;

      });

      var genreCounts = {};

      filteredData.forEach(function(d) {

          var genres = d.Genre.split(",");

          genres.forEach(function(genre) {

              genre = genre.trim();

              genreCounts[genre] = (genreCounts[genre] || 0) + 1;

          });

      });

      var genreData = Object.keys(genreCounts).map(function(genre) {

          return { genre: genre, count: genreCounts[genre] };

      });

      var totalBooks = genreData.reduce(function(acc, d) { return acc + d.count; }, 0);

      var width = 350,

          height = 250,

          padding = 100,

          radius = Math.min(width, height) / 2;

      var svg = d3.select("#pie-chart")

          .append("svg")

          .attr("width", width + 2\*padding)

          .attr("height", height + 2\*padding)

          .append("g")

          .attr("transform", "translate(" + (width / 2 + padding) + "," + (height / 2 + padding) + ")");

        genreData.sort(function(a, b) {

            return b.count - a.count;

        });

        var customColors = ['#FAC55A', '#DF832F', '#D2611A', '#C44004', '#9D4410', '#76491C', '#4E4D27', '#275233', '#00563F'];

        var colorMap = {};

        genreData.forEach(function(d, i) {

            colorMap[d.genre] = customColors[i];

        });

        var colorScale = d3.scaleOrdinal()

            .domain(genreData.map(function(d) { return d.genre; }))

            .range(genreData.map(function(d) { return colorMap[d.genre]; }));

        var pie = d3.pie()

            .value(function(d) { return d.count; });

        var arc = d3.arc()

            .innerRadius(0)

            .outerRadius(Math.min(width, height) / 2 - 1);

        var outerArc = d3.arc()

            .innerRadius(radius + 50)

            .outerRadius(radius);

        var arcs = svg.selectAll(".arc")

            .data(pie(genreData))

            .enter().append("g")

            .attr("class", "arc");

        arcs.append("path")

            .attr("d", arc)

            .attr("fill", function(d) { return colorScale(d.data.genre); })

        svg.append("text")

            .attr("class", "pie-chart")

            .attr("x", "0px")

            .attr("y", "-185px")

            .attr("text-anchor", "middle")

            .text("Genres read:")

            .attr("font-weight", "600")

            .attr("font-size", "1.4em");

      });

    }

* Izrada donut charta za prikaz ukupnog broja pročitanih knjiga
  + 1. Testirati funkcionalnosti i osigurati da su ispravne, tj. da funkcioniraju na očekivani način. Potrebno potvrditi slikom.

## Implementacija osnovnog ponašanja

* + 1. Izraditi kod koji omogućava ranije definirano ponašanje:
* Prikaz različitih grafikona nakon odabira godine/a
* Prikaz liste pročitanih knjiga za odabrani mjesec
* Prikaz ukupnog broja pročitanih stranica za odabranu godinu ili za više odabranih godina
  + 1. Testirati ponašanje i osigurati da je ispravno, tj. da funkcionira na očekivani način: