SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek

Projektni zadatak iz predmeta

VIZUALIZACIJA PODATAKA

Vodeći uzroci smrti u SAD-u

od 1999. do 2017.godine

Student: Dominik Škrinjar, DRD

Mentor: Denis Ivanović

# U Osijeku, lipanj 2023.

# Sadržaj

[**1. KV1 - Definiranje projektnog zadatka 3**](#_4i6s1smptslt)

[1.1. Projektni zadatak 3](#_wgoiizki7jnh)

[1.2. Podatci 3](#_jjnsdui0w5uv)

[1.3. Obrada podataka 3](#_jfu6ybs7kwrw)

[1.4. Relevantne vrste prikaza za korištene podatke 3](#_1g4rlc1z1va6)

[**2. KV2 - Dizajn vizualizacije podataka. 4**](#_smxrkz6d6wy8)

[2.1. Pitanja na koja vizualizacija daje odgovor 4](#_cqmiylhgneug)

[2.2. Skica vizualizacije podataka 4](#_i61spu8qyxmd)

[2.3. Postojeća rješenja i primjeri 4](#_qmiyijy4jsp8)

[2.4. Prilagodba podataka 4](#_cch1vltulvr)

[2.5. Boje i podatci 4](#_w22k3frzbtb2)

[**3. KV3 - Izrada prototipne vizualizacije podataka 5**](#_k5q372z1w040)

[3.1. Osnovne funkcionalnosti i ponašanja 5](#_yaexlmqqdt0u)

[3.2. Napredne funkcionalnosti i ponašanja: 5](#_y300tbmoybm5)

[3.3. Implementacija osnovnih funkcionalnosti 5](#_g3q3h1rhqzyz)

[3.4. Implementacija osnovnog ponašanja 5](#_itz19pnh67g0)

[**4. KV4 - Izrada konačne vizualizacije podataka 6**](#_sj4kshcbtchw)

[4.1. Implementacija osnovnih funkcionalnosti 6](#_dc19rzmsn9k5)

[4.2. Implementacija osnovnog ponašanja 6](#_exvg8zl4eo3w)

[4.3. Implementacija naprednih funkcionalnosti 6](#_ackz82cvngj2)

[4.4. Implementacija naprednog ponašanja 6](#_le6sbg8zsugz)

[**5. KV5 - Dovršetak projektnog zadatka i pisanje dokumentacije 7**](#_woao7m26bf75)

[5.1. Eventualne preinake i dorade rješenja - u dogovoru s nastavnikom 7](#_116xkgvrmrqh)

[5.2. Izrada dokumenta - projektne dokumentacije 7](#_6zplla712qe7)

[**Literatura 8**](#_ahmk8vkny1f9)

# KV1 - Definiranje projektnog zadatka

## Projektni zadatak

Napraviti vizualizaciju koja će prikazivati broju umrlih od određene bolesti u svakoj od saveznih država. Napraviti i prikazati interaktivnu kartu SAD-a , koja će na prikazivati broj umrlih od određene bolesti u razdoblju od 1999 do 2017. Prikazati kako se mijena broju umrlih od određene vrste bolesti kroz godine

## Podatci

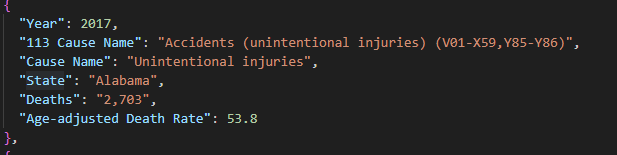
<https://data.cdc.gov/NCHS/NCHS-Leading-Causes-of-Death-United-States/bi63-dtpu>

<https://www.kaggle.com/datasets/mattop/leading-causes-of-death-in-the-united-states>

Podaci prikazuje broj umrlih od 10 vodećih uzroka smrti u razdoblju od 1999 do 2017 godine u SAD-u..

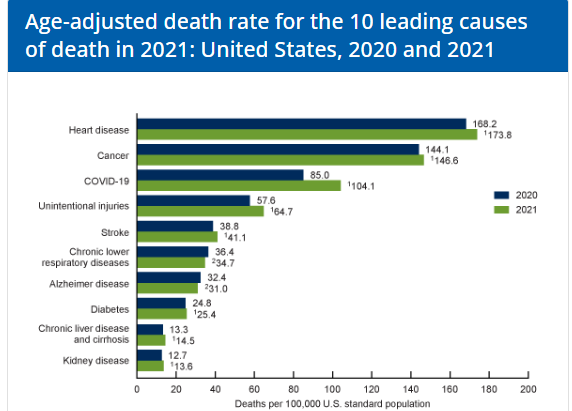
## Obrada podataka

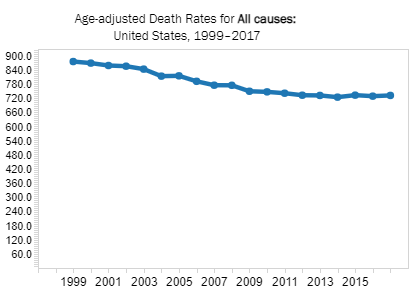
Uzeli smo dva skupa podataka i spojili smo ih, tako što smo filtrirali najvažnije podatke .

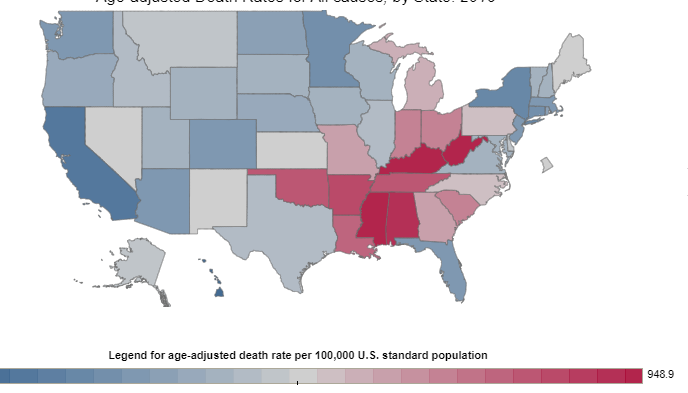


## Relevantne vrste prikaza za korištene podatke

Podatke možemo prikazati na različite načine, pomoću stupčastog grafa, linijskog grafa i interaktivne karte SAD-a. Odabrali smo prikaz interaktivne karte za prikaz podataka.







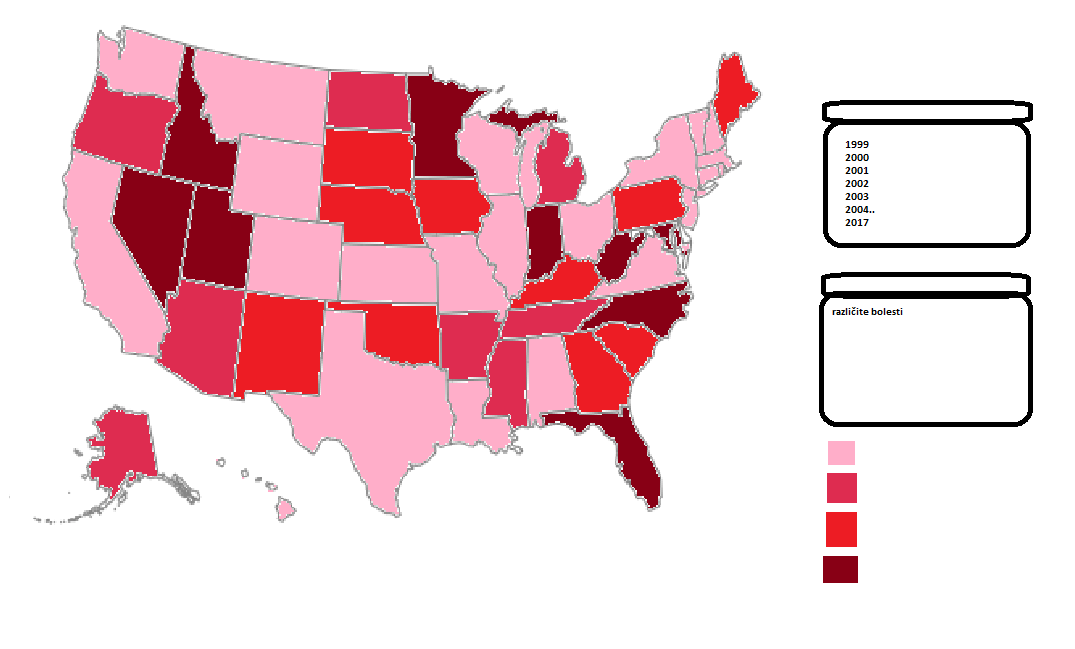
# KV2 - Dizajn vizualizacije podataka.

## Pitanja na koja vizualizacija daje odgovor

Vizualizacija odgovara na pitanja kolika je stopa smrtnosti i broj umrlih od određene bolesti u saveznoj državi, određene godine.

## Skica vizualizacije podataka

Projekt će sadržavati interkativnu mapu koja će iskoćni prozor kada se prolazi mišem preko država. Prikazivati će ime savezne države, legendu koja prikazuje stopu smrtnosti i dva dugmeta ( za godinu I vrstu bolesti).

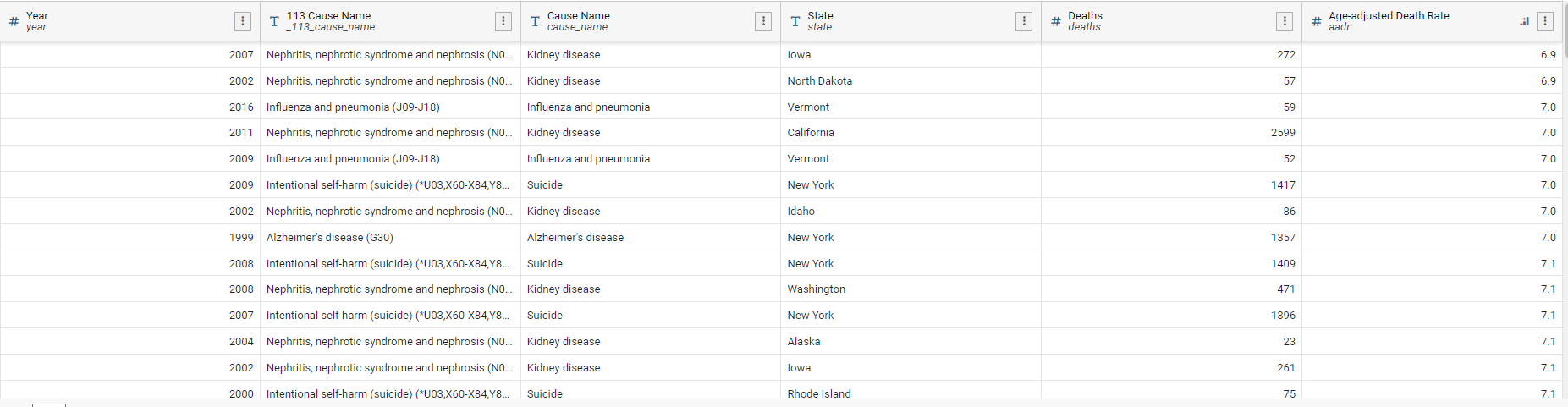


## Postojeća rješenja i primjeri

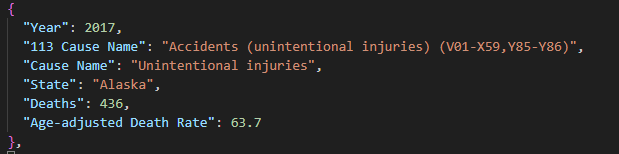
## 

Na slici je prikazano kartu malih obiteljskih farmi te njihovu prodaju u odnosu na ukupnu prodaju.

## 2.4 Prilagodba podataka



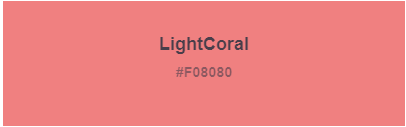
U tablici je prikazano koje godine i od koje bolesti je umrlo ljudi u određenoj saveznoj državi.Zadnji stupac prikazuje stopu smrtnosti na 100 000 ljudi.



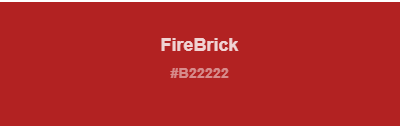
## Boje i podatci

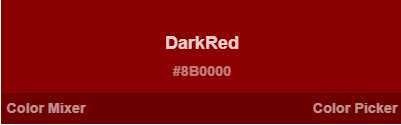
Za lakši prikaz stope smrtnosti koristimo različite nijanse crvene boje :

<10

<20

<30

<50

50+

# KV3 - Izrada prototipne vizualizacije podataka

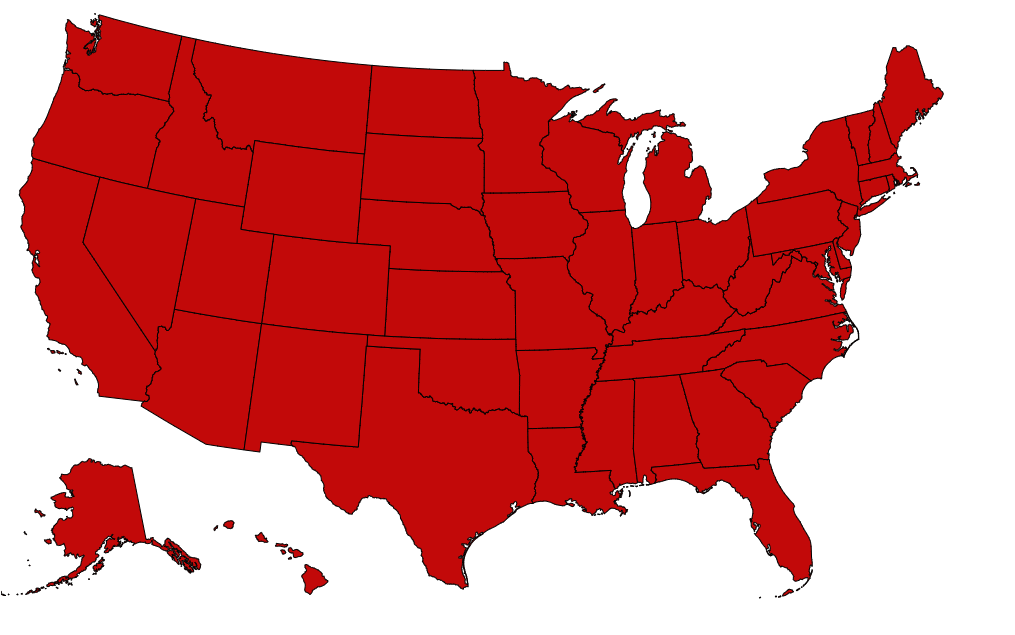
## Osnovne funkcionalnosti i ponašanja

Prikaz karte i podataka prelaskom preko svake savezne države posebno.

## Napredne funkcionalnosti i ponašanja:

Prikaz podataka koristeći slide bar koji nam omogućuje prikaz podataka za određenu godinu , dropdown button za prikaz vrste bolesti i legenda koja prikazuje stopu smrtnosti.

## Implementacija osnovnih funkcionalnosti



<!-- Step 1: Load D3 and TopoJSON libraries -->

  <script src="https://d3js.org/d3.v7.min.js"></script>

  <script src="https://unpkg.com/topojson@3"></script>

  <script>

    (async () => {

      // Step 2: Load the US map data using TopoJSON

      const us = await d3.json('https://d3js.org/us-10m.v1.json');

      const mapData = topojson.feature(us, us.objects.states).features;

     const data = await d3.json("csvjson.json");

      // Step 3: Draw the SVG

      const width = 960;

      const height = 600;

      const svg = d3.select('body')

        .append('svg')

        .attr('width', width)

        .attr('height', height);

      // Step 4: Create an instance of geoPath

      const path = d3.geoPath();

      // Step 5: Use the path to plot the US map

      svg.selectAll('.state')

        .data(mapData)

        .enter()

        .append('path')

        .attr('class', 'state')

        .attr('d', path);

      // Step 6: Add state names

      svg.selectAll('.state-label')

        .data(mapData)

        .enter()

        .append('text')

        .attr('class', 'state-label')

        .attr('transform', function(d) {

          const centroid = path.centroid(d);

          return `translate(${centroid})`;

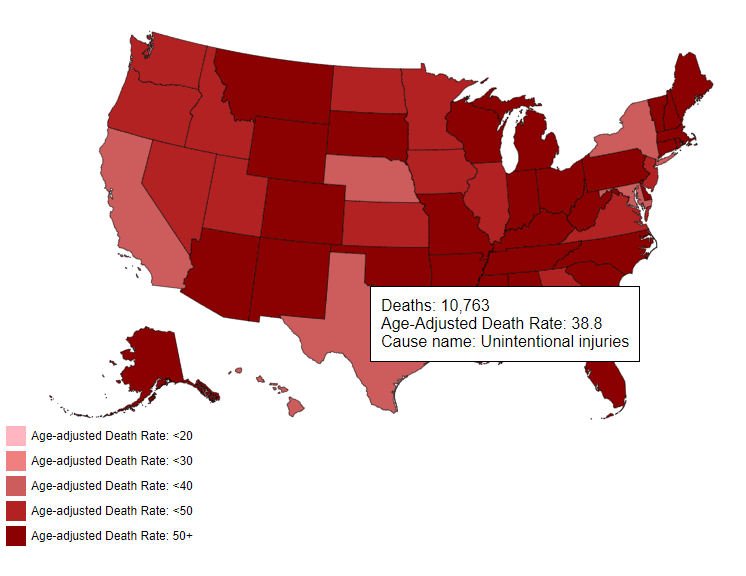
        })

        .text(function(d) {

          return d.states;

        });

## Implementacija osnovnog ponašanja

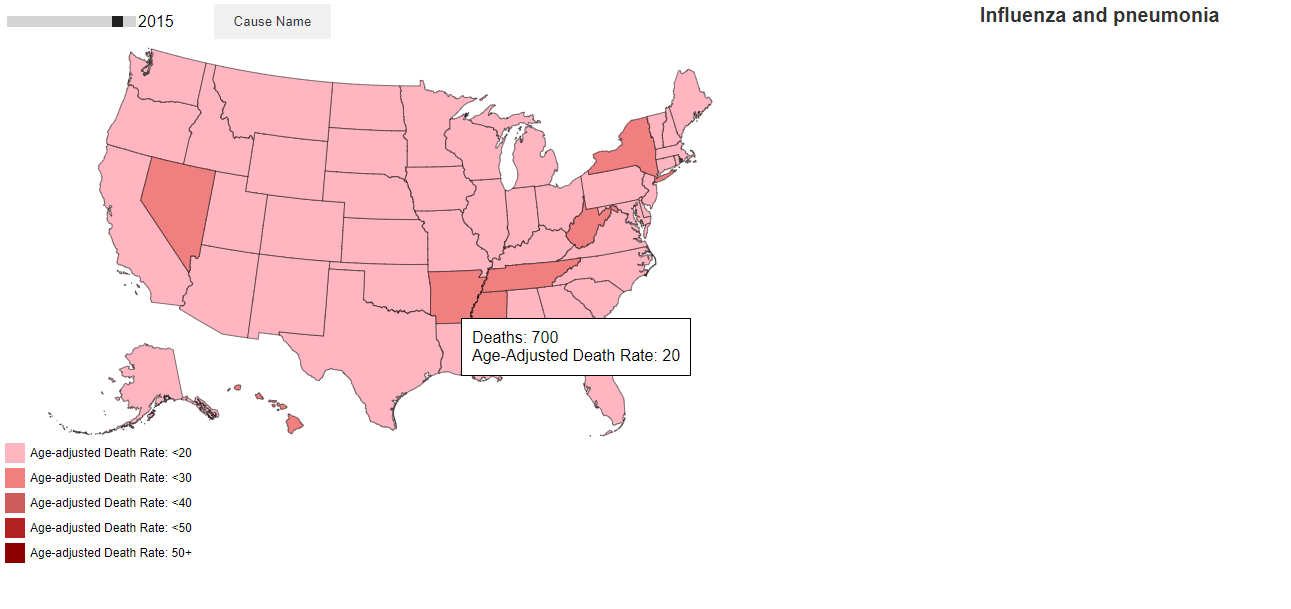


1. ///////////////////LEGENDA/////////////////////
2. var colors = ["#FFB6C1", "#F08080", "#CD5C5C", "#B22222", "#8B0000"];
3. var legend = d3.select("#legend");
4. var deathRate = ["<20","<30","<40","<50","50+",]
5. var legendItems = legend.selectAll(".legend-item")
6. .data(colors)
7. .enter()
8. .append("div")
9. .attr("class", "legend-item");
10. legendItems.append("div")
11. .attr("class", "legend-color")
12. .style("background-color", function(d) { return d; });
13. legendItems.append("div")
14. .attr("class", "legend-label")
15. .text(function(d, i) { return "Age-adjusted Death Rate: " + deathRate[i]; });

# KV4 - Izrada konačne vizualizacije podataka

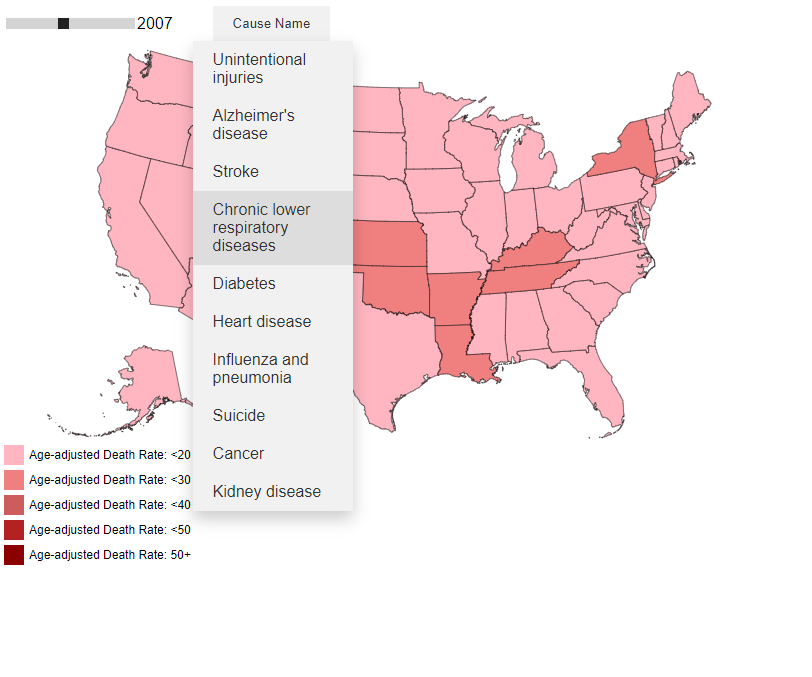
## 4.1 Implementacija osnovnih funkcionalnosti i ponašanja

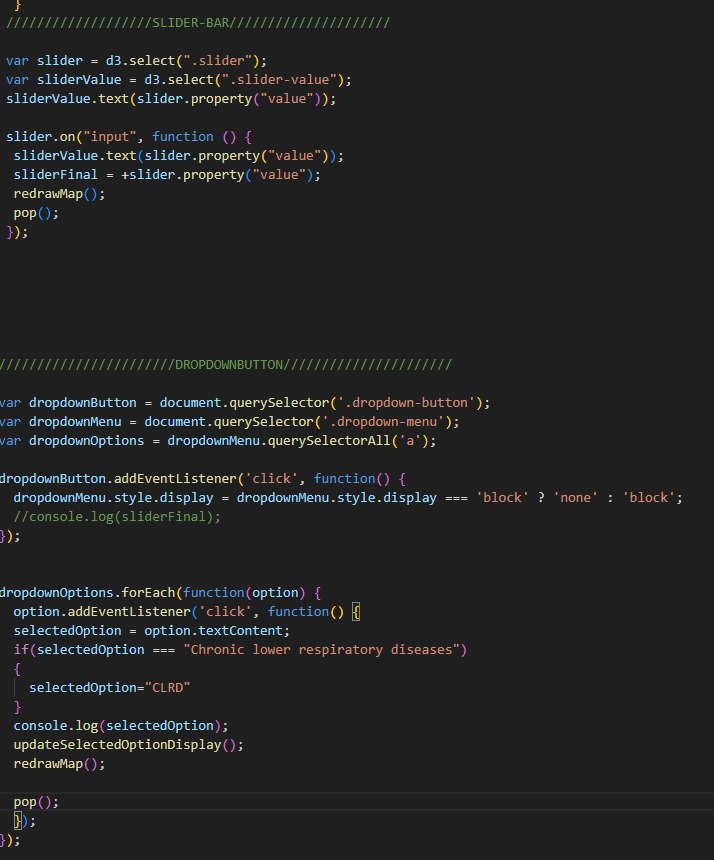
Na slici možemo vidjeti gotove osnovne funkcionalnosti i ponašanja



## 4.2 Implementacija naprednih funkcionalnosti i ponašanja

Na slici možemo vidjeti kako izgledaju napredne funkcionalnosti to jest slide bar i drop down button.

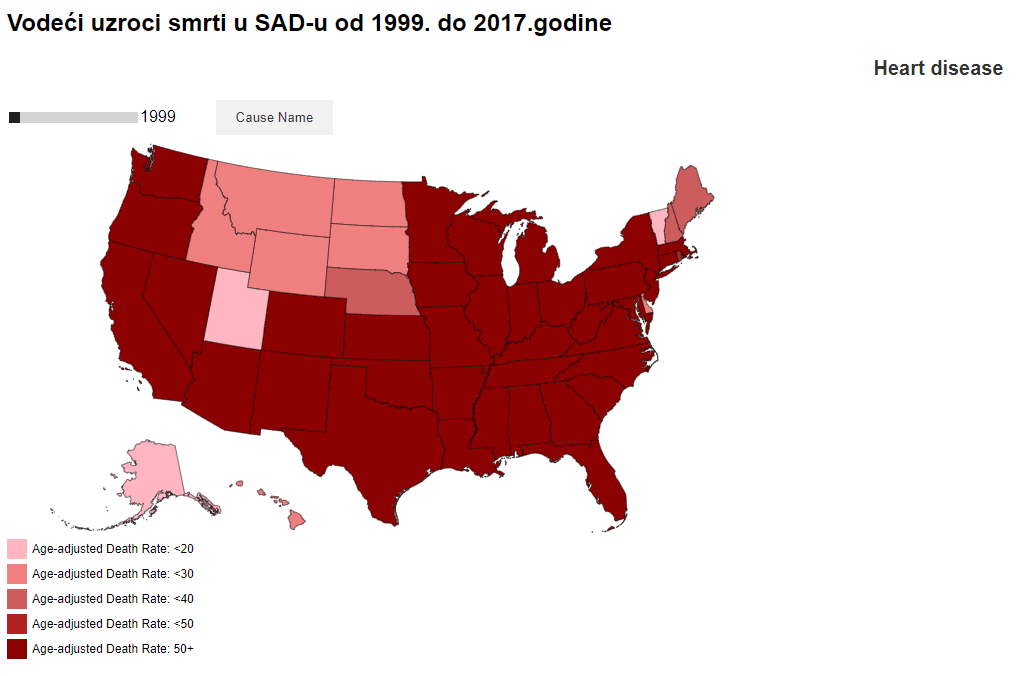




# KV5 - Dovršetak projektnog zadatka i pisanje dokumentacije

## 5.1Eventualne preinake i dorade rješenja - u dogovoru s nastavnikom

Pod ovim zadatkom je poboljšan User Experience tako što smo dodali naslov.Na završetku projekt izgleda ovako:



## 5.2 Izrada dokumenta - projektne dokumentacije

Projektni zadatak prikazuje kickstarter projekte u odnosu na populaciju pojedinih država svijeta.

Korištene tehnologije se HTML, JS, CSS, D3.

Kako bi se projekt pokrenuo, potrebno je imati VS Code s ekstenzijom Live Server, te klonirati projekt iz git-a / extractat zipani file sa merlin-a. Nakon toga je potrebno otvoriti VS Code u tom folderu, te u donjem desnom kutu kliknuti na „Go Live“. Nakon toga se defaultni web browser otvara sam i vizualizacija podataka je spremna za korištenje. Nakon što se otvori browser, user vidi naslov projekta i prikaz mape.

6.Literatura

Wikipedia, HTML, dostupno na: [HTML - Wikipedia](https://en.wikipedia.org/wiki/HTML)  
Wikipedia, JS, dostupno na: [JavaScript - Wikipedia](https://en.wikipedia.org/wiki/JavaScript)  
D3.js, Data-Driven Document, dostupno na: [D3.js - Data-Driven Documents (d3js.org)](https://d3js.org/)

7.Prilozi

Git repozotorij: https://github.com/Snupix031/VP-projekt

Programski kod:

(function() {

  var margin = {top:0, left :0 , right:0, bottom:0},

  height=400 - margin.top-margin.bottom,

  width = 800 - margin.left- margin.right;

  ///////////////////LEGENDA/////////////////////

  var colors = ["#FFB6C1", "#F08080", "#CD5C5C", "#B22222", "#8B0000"];

  var sliderFinal = 1999;

  var selectedOption ="Unintentional injuries";

  var legend = d3.select(".legend");

  var deathRate = ["<20","<30","<40","<50","50+"];

var legendItems = legend.selectAll(".legend-item")

  .data(colors)

  .enter()

  .append("div")

  .attr("class", "legend-item");

legendItems.append("div")

  .attr("class", "legend-color")

  .style("background-color", function(d) { return d; });

legendItems.append("div")

  .attr("class", "legend-label")

  .text("Age-adjusted Death Rate: ")

  .text(function(d, i) {

     return "Age-adjusted Death Rate: " + deathRate[i];

     });

///////////MAPA/////////////////////////////////////////////

  var svg=d3.select("#map")

    .append("svg")

    .attr("height",height+margin.top+margin.bottom)

    .attr("width", width+margin.left+margin.right)

    .append("g")

    .attr("transform","translate("+margin.left + "," + margin.top + ")");

    d3.queue()

      .defer(d3.json,"usa-topojson.json")

      .defer(d3.json,"csvjson.json")

      .await(ready)

 var projection = d3.geoAlbersUsa()

  .translate([width/2,height/2])

  .scale(850)

  var path=d3.geoPath()

    .projection(projection)

    function ready(error,data,mydata)

    {

      var states = topojson.feature(data, data.objects.states).features

      states.forEach(function(state) {

        var stateName = state.properties.name;

        //console.log(stateName);

        // Pronalaženje podataka za određenu državu u "bolestData"

       var stateData = mydata.find(function(d) {

          return d.State === stateName && d.Year === sliderFinal && d["Cause Name"] === selectedOption;

        });

        // Ako se podaci pronađu, dodajte ih kao svojstva države

        if (stateData) {

          state.properties.deaths = stateData.Deaths;

          state.properties.ageAdjustedDeathRate = stateData["Age-adjusted Death Rate"];

          state.properties.causeName=stateData["Cause Name"];

          // Dodajte ostale željene podatke na isti način

        }

      });

      //console.log(data.objects.states.geometries[0].properties.name);

      //console.log(mydata)

      //console.log(states)

      svg.selectAll(".state")

        .data(states)

        .enter().append("path")

        .attr("class", "state")

        .attr("d", path)

      .attr("fill", function(d) {

        var stateName = d.properties.name;

        var stateData = mydata.find(function(data) {

          return data.State === stateName;

        })

      if (stateData) {

          var deathRate = stateData["Age-adjusted Death Rate"];

           return getColor(deathRate);

        }

      });

        function getColor(deathRate) {

          if (deathRate < 20) {

            return colors[0];

          } else if (deathRate < 30) {

            return colors[1];

          } else if (deathRate < 40) {

            return colors[2];

          } else if (deathRate < 50) {

            return colors[3];

          } else {

            return colors[4];

          }

        }

        function getColorCancer(deathRate){

          if (deathRate < 120) {

            return colors[0];

          } else if (deathRate < 130) {

            return colors[1];

          } else if (deathRate < 140) {

            return colors[2];

          } else if (deathRate < 150) {

            return colors[3];

          } else {

            return colors[4];

          }

        }

       ///////////////////SLIDER-BAR/////////////////////

       var slider = d3.select(".slider");

       var sliderValue = d3.select(".slider-value");

       sliderValue.text(slider.property("value"));

       slider.on("input", function () {

        sliderValue.text(slider.property("value"));

        sliderFinal = +slider.property("value");

        redrawMap();

        pop();

       });

      ///////////////////////DROPDOWNBUTTON//////////////////////

      var dropdownButton = document.querySelector('.dropdown-button');

      var dropdownMenu = document.querySelector('.dropdown-menu');

      var dropdownOptions = dropdownMenu.querySelectorAll('a');

      dropdownButton.addEventListener('click', function() {

        dropdownMenu.style.display = dropdownMenu.style.display === 'block' ? 'none' : 'block';

        //console.log(sliderFinal);

      });

      dropdownOptions.forEach(function(option) {

        option.addEventListener('click', function() {

        selectedOption = option.textContent;

        if(selectedOption === "Chronic lower respiratory diseases")

        {

          selectedOption="CLRD"

        }

        console.log(selectedOption);

        updateSelectedOptionDisplay();

        redrawMap();

        pop();

        });

      });

      window.addEventListener('click', function(event) {

        if (!dropdownButton.contains(event.target)) {

          dropdownMenu.style.display = 'none';

        }

      });

      function showPopup(deaths, ageAdjustedDeathRate, state ) {

        popup

          .style("left", d3.event.pageX + "px")

          .style("top", d3.event.pageY + "px")

          .html("Deaths: " + deaths + "<br>Age-Adjusted Death Rate: " + ageAdjustedDeathRate + "<br>State: " + state)

          .style("display", "block");

      }

      // Funkcija za skrivanje popup prozora

      function hidePopup() {

        popup.style("display", "none");

      }

      var popup = d3.select(".popup");

      function redrawMap() {

        svg.selectAll(".state")

          .attr("fill", function (d) {

            var stateName = d.properties.name;

            var stateData = mydata.find(function (data) {

              return data.State === stateName && data.Year === sliderFinal && data["Cause Name"] === selectedOption;

            });

            if (stateData) {

              var deathRate = stateData["Age-adjusted Death Rate"];

              var CaseName = stateData["Cause Name"];

              if (selectedOption === "Cancer") {

                return getColorCancer(deathRate);

              } else {

                return getColor(deathRate);

              }

            }

          })

          .on("mouseover", function(d) {

            var stateName = d.properties.name;

            var stateData = mydata.find(function (data) {

              return data.State === stateName && data.Year === sliderFinal && data["Cause Name"] === selectedOption;

            });

            if (stateData) {

              var deaths = stateData.Deaths || "N/A";

              var state = stateData.State|| "N/A";

              var ageAdjustedDeathRate = stateData["Age-adjusted Death Rate"] || "N/A";

              showPopup(deaths, ageAdjustedDeathRate, state);

            }

          })

          .on("mouseout", function() {

            hidePopup();

          });

      }

      function pop()

      {

        svg.selectAll(".state")

        .data(states)

        .enter().append("path")

        .attr("class", "state")

        .attr("d", path)

        .on("mouseover", function(d) {

          var stateName = d.properties.name;

            var stateData = mydata.find(function (data) {

              return data.State === stateName && data.Year === sliderFinal && data["Cause Name"] === selectedOption;

            });

          // Prikaz popouta s vrijednostima iz podataka države

          if (stateData) {

            var deaths = d.properties.deaths || "N/A";

            var ageAdjustedDeathRate = d.properties.ageAdjustedDeathRate || "N/A";

            var causeName =d.properties.causeName || "N/A";

          }

          // Prikaz ostalih željenih podataka

          d3.select(".popup")

          .style("left", d3.event.pageX + "px")

          .style("top", d3.event.pageY + "px")

          .html("Deaths: " + deaths + "<br>Age-Adjusted Death Rate: " + ageAdjustedDeathRate )

          .style("display", "block");

      })

      .on("mouseout", function() {

        // Sakrij popout kada se miš pomakne iznad države

        d3.select(".popup").style("display", "none");

      })

      }

      function updateSelectedOptionDisplay() {

        var selectedOptionDisplay = document.getElementById("selectedOptionDisplay");

        selectedOptionDisplay.textContent = selectedOption;

      }

      updateSelectedOptionDisplay();

    }

})();

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <style>

            \*{

                font-family:sans-serif;

            }

           p {

                font-size: 0.85 em;

           }

           .state {

            stroke:#000;

            stroke-width: 0.5;

           }

           .state-label {

            fill: black;

            font-size: 12px;

            text-anchor: middle;

            pointer-events: none;

    }

           .slider-container {

            display: flex;

            align-items: center;

            margin-right: 20px;

        }

        .slider {

  flex: 1;

}

#slider {

  width: 100%;

}

#slider-value {

  margin-left: 20px;

  font-size: 18px;

  font-weight: bold;

  color: #333;

}

#selectedOptionDisplay {

  flex: 1;

  text-align: center;

  font-size: 20px;

  font-weight: bold;

  color: #333;

}

.dropdown {

  position: relative;

  display: inline-block;

}

.dropdown-button {

  margin-left: 20px;

  background-color: #f1f1f1;

  color: #333;

  border: none;

  padding: 10px 20px;

  cursor: pointer;

}

.dropdown-menu {

  display: none;

  position: absolute;

  background-color: #f1f1f1;

  min-width: 160px;

  box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2);

  z-index: 1;

  list-style: none;

  padding: 0;

  margin: 0;

}

.dropdown-menu li a {

  display: block;

  padding: 10px 20px;

  text-decoration: none;

  color: #333;

}

.dropdown-menu li a:hover {

  background-color: #ddd;

}

.nav-container {

  display: flex;

}

.popup {

  position: absolute;

  background-color: white;

  border: 1px solid black;

  padding: 10px;

  display: none;

}

.legend-item {

  display: flex;

  align-items: center;

  margin-bottom: 5px;

}

.legend-color {

  width: 20px;

  height: 20px;

  margin-right: 5px;

}

.legend-label {

  font-size: 12px;

}

        </style>

    </head>

    <nav>

      <div><h1><b>Vodeći uzroci smrti u SAD-u od 1999. do 2017.godine</b></h1></div>

      <h3 id="selectedOptionDisplay"></h3>

        <div class="nav-container">

        <div class="slider-container">

          <input type="range" class="slider" min="1999" max="2017" value="1997" step="1">

          <span class="slider-value">1997</span>

        </div>

        <div class="dropdown">

          <button class="dropdown-button">Cause Name</button>

          <ul class="dropdown-menu">

            <li><a href="#">Unintentional injuries</a></li>

            <li><a href="#">Alzheimer's disease</a></li>

            <li><a href="#">Stroke</a></li>

            <li><a href="#">Chronic lower respiratory diseases</a></li>

            <li><a href="#">Diabetes</a></li>

            <li><a href="#">Heart disease</a></li>

            <li><a href="#">Influenza and pneumonia</a></li>

            <li><a href="#">Suicide</a></li>

            <li><a href="#">Cancer</a></li>

            <li><a href="#">Kidney disease</a></li>

          </ul>

        </div>

      </div>

      </nav>

    <body>

      <div class="popup"></div>

        <div id="map"></div>

        <div class="legend"></div>

        <script src="https://cdnjs.cloudflare.com/ajax/libs/d3/4.2.2/d3.min.js"></script>

        <script src="http://d3js.org/topojson.v1.min.js"></script>

        <script src="script.js"></script>

    </body>

</html>