Finland statistics portal project

by Artturi Siven - 0608366

Components of my project explained

In my project there are three main components (+miscellaneous things):

1. **Map showing Finnish parliamentary election results in 2023.**

I have used leaflet to showcase a map of Finland, drawing all the municipalities there with a geojson file used in exercise for week 5. I have used statsfin to gather the election results for each municipality and used this data to find out the party that got the most votes as well as how many votes percentage wise that party had gotten. Then I colored each municipality according to the party color that won in that municipality. It was quite a lot of work to match the election data to the municipality codes from the geojson data, as the election dataset was massive, and quite confusing to work with. Lastly, I also allowed the user to pick from Google maps or Open street maps.

Another feature of the map is the drag and drop feature, where the user can drag and drop one of two items on any municipality to showcase more data about that municipality in regards to the item that was dragged and dropped. This whole feature took more work than I wish to admit, as it was quite complicated to create a system for figuring out what municipality was the item being dragged over, as leaflet didn't really have a built in feature for this. It did have this thing called mouseover, but it was very difficult to implement that to properly work with my draggable items and then actually directing the user to the new webpage. To make this feature even more user friendly though, I created a system where the municipality that is being hovered over gets highlighted. To make this feature work I utilized another js script called tinycolor, which allowed me to brighten the color of the municipality being hovered over, to make the effect more pronounced.

Formatting for the pc was quite simple. I centered everything, and aligned the drag and drop box with the map. I made the draggable items change style when you hover over them to make them seem more interactive. And to make the page work with smaller screens, I simply decided to remove the drag and drop feature because it seems like a very awkward feature on a touch screen, so instead the user can just click on a municipality and there they can find links to the two possible charts. These were also there for the large screen users. Also, on mobile it shows text about this, instead of about instructions for dragging and dropping.

2. **Chart showcasing the election results in a specific municipality from 1999 to 2023**

I used frappe charts in this part. And to make this work properly, I had to gather all the election data from 7 different elections which means there were a lot of data points. I wanted to show the trend of how many votes each party has gotten in those elections, and this allows the user to really explore potential trends for each municipality.

It wasn't too difficult to figure out the displaying of data on the chart, but I also wanted to allow the user to disable some line charts to only for example compare the trends of two or three parties, as the chart with all the parties enabled looks a bit messy. I initially thought this would be a simple task, but frappe did not have their own feature for hiding or showing line charts, so I had to create my own system for this. To make it work i created checkmark boxes for each party, and if one of the checkmark boxes is unchecked, the page reloads, info about the checkboxes is sent through the url, and using this info a new chart is built with only the checkboxes that were checked. This system works surprisingly well, but it took me way too long to figure out.

And of course there is also an export svg system, to save the svg of the chart, which was very easy to implement.

Formatting this page was very straightforward for the pc, as the frappe chart has built in responsive design and the checkmarks as well as the go back button were also very simple to implement. However, on mobile a strange thing happens where the chart and the legend get really small, so that you cannot really see anything. Also the legend almost disappears and there didn't seem to be a way to not let this happen. The popup window when you click on any of the points on the map did not change in size, and remained way too large, so I disabled it. How I solved the issue with the chart was to force it to have maximum size, and allow the user to scroll on it, so that they can see the whole chart. This was tricky to implement, however I got it working in the end. This feature is very useful for the mobile user as it makes the exported svg a lot more visible, because before this, the svg would keep the same crumbled up look as it was on the screen. Also, I made the button for exporting as well as the checkmark boxes larger, to make them easier to click with touch.

3. **Chart showcasing population age as well as total population in a specific municipality from 1998 to 2022**

This component is very similar to the previous one, as they are both charts created using frappe. However, for this I decided to showcase the population of the municipality by different age groups, as well as showing the overall populations. This allows the user to see the age of the population as well as the trend of the population. Here the trickiest part was showing the line chart for the total population with the bar charts for each age group. But in the end it wasn't too difficult as I could simply use the same fetched data for both datasets on the chart.

I also added a feature of adding data to the chart. This allows the user to explore the data and potentially see in what direction the population is going as well as the population for each age group.

I used the same formatting features for this as for the previous component.

4. **Miscellaneous things.**

Here I wanted to add some features I didn't find a good place for in the above text. One main thing is the loading screen I implemented to show the user that something is loading instead of showing a white screen. I used this in all the components (map, population chart, election chart), and in addition to this, I made it so that it hides all the other components on the population chart and election chart pages, and on the map page, it only hides the map. This should make the user experience more straightforward and less confusing.Point evaluation
I have taken some points away from:

Project points evaluation

Features I have deducted points from:

- ➔ **The application show relevant data on a map and user has change to change the data:**
  This is because it shows relevant info, but the user doesn't really have a chance to change it. They can only change the map from Google map to Open street map.

- ➔ **The application show relevant data on a chart and user has a chance to change the data:**
  The application shows relevant info on the charts, and the user can change the data by going to different municipalities. Also there are two options for data for each municipality

- ➔ **Users can define what should be done to different data items (e.g. values are added, multiplied together etc. before visualization):**
  Here the user cannot do much. They can only add new data points.

Other features and justification for full points:

- ➔ **Well written report:**
  I would argue this is a well written report with good formatting.

- ➔ **Application is responsive and can be used on both desktop and mobile environment:**
  I have put a lot of consideration for both mobile and pc users; changing button sizes, fonts, hiding some elements on mobile, creating a scrollable chart for mobile.

- ➔ **Application works on Firefox, Safari, Edge and Chrome:**
  I have tested the webpages on all the mentioned browsers, and it works.

- ➔ **The application has clear directory structure and everything is organized well:**
  The code is separated so that css, js, other html pages have their own folder and index.html is found in the main project folder.

- ➔ **Drag'n'drop new data to charts/maps:**
  I implemented this feature on the map. User can drag and drop an item on the map.

- ➔ **The application show relevant data on a chart and user has a chance to change the data:**
  The application shows relevant info on the charts, and the user can change the data by going to different municipalities. Also there are two options of data for each municipality.

- ➔ **User is able to switch between different layers of data on map:**
  User can switch between google maps layer and open street maps layer

- ➔ **By clicking the map user has an option to get to additional charts covering that area:**
  User can click on a municipality on the map and get more info about that municipality.

- ➔ **There are more than two items of data available (e.g. elections data, employment rate and number of residents) – this means that there are two API calls made:**
  Data is fetched with two api calls. One for election results and one for age of population

- ➔ **Able to download the visualization as a PNG (or SVG) image:**
  User can download SVG of the chart.

- ➔ **Municipality gets highlighted when an item is being dragged over it:**
  This feature was a very complicated and difficult feature to implement, and while my implementation is not perfect, comparing the amount of work this took compared to for example the drag and drop feature I would say they are comparable.

- ➔ **User being able to hide some parties data from the chart:**
  This was a confusing feature to get started with, and in the end it was quite difficult to get integrated. It works well and efficiently.

Feature	**Points**
Well written PDF report	3
Application is responsive and can be used on both desktop and mobile environment	4
Application works on Firefox, Safari, Edge and Chrome	3
The application has clear directory structure and everything is organized well	2

Drag’n’drop new data to charts/maps	4
The application show relevant data on a map and user has change to change the data	2
The application show relevant data on a chart and user has a chance to change the data	3
User is able to switch between different layers of data on map	2
By clicking the map user has an option to get to additional charts covering that area	4
There are more than two items of data available (e.g. elections data, employment rate and number of residents) – this means that there are two API calls made	4
Users can define what should be done to different data items (e.g. values are added, multiplied together etc. before visualization)	1
Able to download the visualization as a PNG (or SVG) image	2
Features by me:	
Municipality gets highlighted when an item is being dragged over it	4
User being able to hide some parties data from the chart	2
Total points	**40**

Total 40 points.I think this is a fair score considering the amount of time and work I’ve spent on this project as well as the amount I have learned over the course of creating this project.

Ps. I like to comment that frappe and leaflet were fine tools, however maybe a bit too simple if you wanted to do more with them. As most of my limitations came from the fact that these tools had severe restrictions on what you could actually accomplish with them.