DOCUMENTATION OF PROJECT WORK FOR COURSE:
"*CT30A2910 Introduction to Web Programming - Verkko-opetus 1.6.2023-31.7.2023*"

STUDENT: *Santeri Naumanen*
SELECTED PROJECT: *"Project 2: Statistic portal"*

General information about the project:

On the site users are able to display the amount of workforce and the amount of new summer houses in years 1980-2019 by municipality. Workforces status is displayed as it was at the end of decade.

Purpose of the data is to present how the workforce is moving to larger cities and smaller municipalities are becoming "summer house villages". However this is not always the case and there are interesting developments across the country.

On the map the development between decades 2000-2009 and 2010-2019 is displayed with colors. Summer house layer uses fill colors. The red color means that less than 50% of summer houses were built than the decade before. Green color means that more summer houses were built than a decade before. Yellow color marks the status between aforementioned colors.

For the workforce, similar colors are used but the outline of the municipality is colored. Red color is used when the workforce has dropped below 80% of the level of the last decade. Green and yellow are used similarly as in the summer house layer.

Users are able to display either workforce or summerhouses individually or both simultaneously by selecting layers from the upper right corner of the map. By default both layers are presented on top of each other.

Users are able to download the chart currently displayed by clicking the button below the data presentation area.

About the implementation:

Following actions are made via the Javascript:

- Two different JSONs are fetched with getData-function from Tilastokeskus.
    - One JSON includes data of new summer houses each decade by municipality
    - Other JSON includes data of the workforce each year by municipality
        - Only years at the end of each decade were fetched to make data the same size as the summer house data.
- Data JSONs were processed into two nested arrays. Nested arrays followed the following layout: array of four (decades) and each member included array 309 members (municipality).
- geoJSON including municipalities geographical information was fetched from Tilastokeskus.
- After fetching all of the external data, a map was created with **Leaflet**. GeoJson information was applied to the map and the amount of summer houses and amount

of workforce were set as layers. Conditional formatting was applied to layers to present the data in an informative way on the map.
- Each geoJSON element on the map was added a listener for clicks. When a user clicks a municipality on the map a chart is presented on the site with data of the clicked municipality.
    - Chart is created with **Frappe-charts**.
    - Users are able to download the chart being presented by clicking the html-button on site. Button is being listened for clicks in javascript.
- Map also supports hovering over to display the name of the municipality. Clicking also shows data of the active layer in a popup window in addition to the chart displayed on the site.

HTML code was kept simple:
- Map and chart elements are created via javascript to divs intended for them.
- Button to download the chart picture is created via HTML code.
- Information about the site is presented with normal <p>-paragraphs in the of the page.

Site was made responsive with CSS styling:
- On desktop environments the map and chart are displayed side by side as the map of Finland is more vertical than horizontal.
- When the width of the screen is below 768 pixels (mobile devices), the chart appears below the map to ensure the map is usable.

Self-assessment of the project work and justification:

Following features were not implemented and therefore following deductions to maximum points should be made:

- Drag’n’drop new data to charts/maps
    - -4 points - not implemented
- The application show relevant data on a map and user has change to change the data
    - -1.5 points - user has not change to change the data
- The application show relevant data on a chart and user has chance to change the data
    - -1.5 points - user has not change to change the data

With a total of -7 points of deductions, the project should receive **23 points out of 30**.