

Project Documentation

This project was done as part of "Introduction to Web Programming" at LUT. This documentation will outline relevant parts of development, as well as tools and resources used to implement the final product.

This project falls under Project 1, Statistics Portal. The statistics in question are a number of observations in a given area.

Date of submission:

19.10.2025

Introduction

"Wild Life in Finland" is a website that can display research grade observations of taxons found in Finland on a map. Wildlife is split into the following categories:

- Aves
- Amphibians
- Reptiles
- Mammals
- Fish
- Amphibians
- Insects*
- Arachnids

*The list of Insects is far from complete because the sheer amount of species didn't seem sensible for the current prototype. I might refactor this at a later date, so the taxons are split into their sub categories. For now this seemed like a more user-friendly approach for users that may not come from a scientific background.

The observation data is retrieved from iNaturalist's v1 API. Local data regarding the existing taxons in Finland is from Laji.fi and was converted into json files. (See the Resources section for more information.)

A user can select from a list of species and view where in Finland they have been observed. By selecting several species, they can compare observation points directly.

The option to filter for a species name is also given. The search queries for both scientific and common name.

If no observations can be found, the user receives that information in a toast notification.

On a second tab, all selected species are listed with their common name, scientific name, conservation status and whether or not they are native to Finland. Additionally, a link to their

wikipedia entry is given.

Heatmaps of selected species can be toggled on and off in the top right corner.

A user can also reset the map, essentially clearing their previous submissions and starting anew without manually refreshing their window.

In the future I plan to expand the functionality to split between heat maps and individual observations. The functionality to get observations and display them as tooltips is already in the code, but needs some clean up and is not the way I want it to be at the moment.

How to Run

Static Website

The current website is deployed via Netlify and can be accessed on here:

<https://wildlife-in-finland.netlify.app/>

Please note that in the production version the satellite map does not work. (This will be fixed in a future update)

Locally

Option A:

you can host live server from inside the /dist/ Folder.

Option B:

Inside /Webdev-Project-Work/ run "npm run preview" or "npm run dev". Note that this will probably require installing npm packages.

AI Usage:

LLMs used: ChatGPT, aiStudio (Google Gemini).

Use cases:

Debugging

Display None issue on Select Label

At one point I had struggled with displaying the species list with either the common name or scientific name, because I could not figure out why the toggle was not working.

I used Google Gemini for debugging the issue and it correctly pointed out that the option element is not allowed to have elements inside of it:

```
<!-- Suspected Template Structure -->
<template id="speciesDropDownList">
  <!-- ... grid/category stuff ... -->
  <select class="species-dropdown">
```

```
<option class="species-dropdown-item">
  <!-- THE PROBLEM IS HERE -->
  <span class="species-dropdown-item--scientific"></span>
  <span class="species-dropdown-item--common"></span>
</option>
</select>
</template>
```

Afterwards I refactored the select logic to use a hidden checkbox input instead.

Refactoring

When I began coding, I only had one big JavaScript file. This quickly grew out of hand, as the complexity grew and I knew I had to split and refactor the code eventually. I used Google Gemini to suggest a class based approach.

The given code and structure was used as a jumping off point, although my architecture is different from the suggested one:

Here's a refactored version of your JavaScript code, focusing on encapsulation, best practices, and a modular structure using vanilla JavaScript.

The suggestion breaks down the functionality into distinct modules:

1. **AppConfig**: Centralizes all constants and configuration values.
2. **DataService**: Handles all data fetching (local JSON files and iNaturalist API).
3. **MapManager**: Manages all Leaflet map initialization and interaction.
4. **SpeciesUIManager**: Responsible for rendering species dropdowns and individual species cards.
5. **SelectedSpeciesManager**: Manages the list of selected species, their checkboxes, and "tag" display.
6. **App**: The main orchestrator that initializes all modules and ties together event handling.

It made sense to me to have all API calls in one place and the map in another. However, some of the other classes would have only added more complexity.

StorageHandler

The current solution utilizing iDB is done by chatGPT (StorageHandler.js). I plan to change this eventually, as I get deeper into local storage logic and databank tools. However, for now I wanted to have a rudimentary solution to prevent unnecessary calls to the iNaturalist API.

Points

Outline

Feature	Max points	Suggested Points
Well written PDF report	3	3
Application is responsive and can be used on both desktop and mobile environment	4	4
Application works on Firefox, Safari, Edge and Chrome	3	3
The application has clear directory structure and everything is organized well	2	2
Drag'n'drop new data to charts/maps (there are different data items on screen that user can drag and drop as she wishes)	4	0
The application show relevant data on a map and user has chance to change the data	3	3
User is able to switch between different layers of data on map	2	2
By clicking the map user has an option to get to additional charts covering that area	4	0
There are more than one item of data available (e.g. elections data, employment rate and number of residents) – this means that there are two API calls made	3	0
There are more than two items of data available (e.g. elections data, employment rate and number of residents) – this means that there are three API calls made	2	0
Data is combined and merged to generate new data, which is then visualized	3	3
Users can define what should be done to different data items (e.g. values are added, multiplied together etc. before visualization)	2	2
Able to download the visualization as a PNG (or SVG) image	2	2
TOTAL	37	24

Own Features:

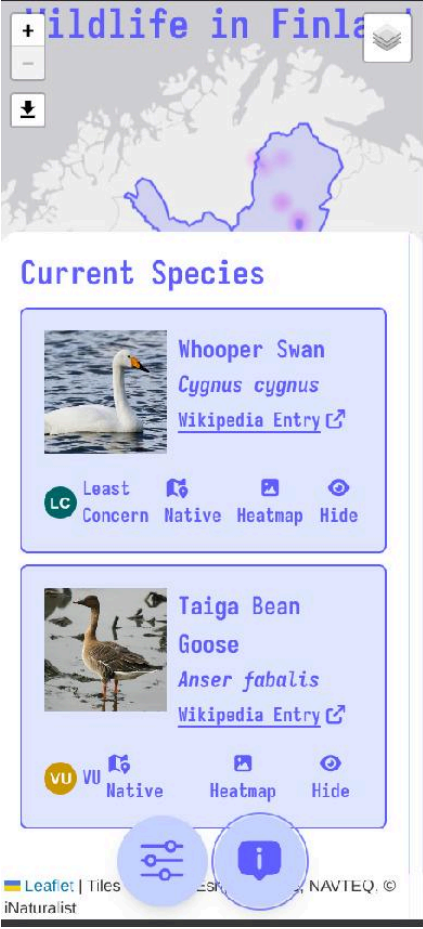

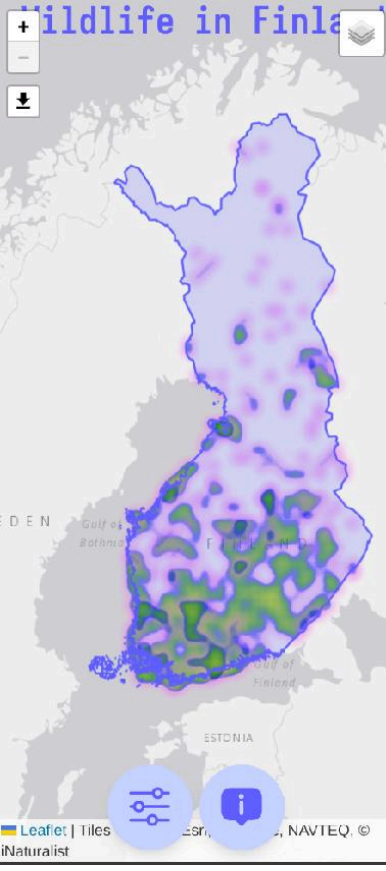
Feature	Max Points	Suggested Points
Select multiple new data and add it to the map.	4	4
Filter available options by using the Search Input	3	3
User can clear the map to get a blank slate without refreshing the page.	1	1
User can toggle between scientific and common name for taxons.	1	1

Feature	Max Points	Suggested Points
User gets state feedback in the form of loading indicator and error and success messages.	3	3
User can dismiss toast notification.	1	1
User can download individual heatmaps as a PNG image.	2	2
User can switch between satellite and outline map view (local feature only).	1	1
User can reset their current selection before submitting it, to clear all selections, but can also deselect individual items.	2	2
TOTAL	18	18

Base Features + own Features, total points = 42.

Features


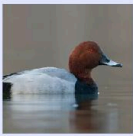
Responsive Design

 <p>This screenshot shows the 'Current Species' section of the application. It features two species cards. The first card is for the 'Whooper Swan' (<i>Cygnus cygnus</i>), showing a photo, the scientific name, a link to the Wikipedia entry, and a 'Least Concern' status icon. The second card is for the 'Taiga Bean Goose' (<i>Anser fabalis</i>), also with a photo, scientific name, Wikipedia link, and 'Vulnerable' status icon. Below the cards are interactive buttons for 'Native', 'Heatmap', and 'Hide' for each species. The bottom of the screen shows a map of Finland with a heatmap overlay.</p>	 <p>This screenshot displays the 'Filter Menu' overlay. At the top, a green toast notification says 'Successfully fetched all info.'. Below this is a 'Category' section with buttons for Aves, Amphibians, Reptiles, Mammals, Fish, Mollusks, Arachnids, and Insects. The 'Aves' category is currently selected. Underneath is an 'Options' section with a toggle switch for 'Display Scientific Names', which is currently turned off. At the bottom, there are buttons for 'Species' (with a settings icon) and 'Info' (with an information icon). The background shows a map of Finland with a heatmap.</p>	 <p>This screenshot shows the 'Map View' of the application. It displays a map of Finland with a heatmap overlay representing species distribution. The map includes labels for neighboring countries like Sweden, Denmark, and Estonia, and bodies of water like the Gulf of Bothnia. At the bottom, there are buttons for 'Species' (with a settings icon) and 'Info' (with an information icon). The map is credited to 'Leaflet Tiles, iNaturalist, NAVTEQ'.</p>
Current Species View	Filter Menu	Map View

File Structure

All assets are inside src/assets (icons for conservation status). JavaScript files are inside the js directory and split into their responsibilities. App.js is responsible for starting the app. CSS uses Tailwind but serves the styling via @apply inside custom class names to prevent bloating the index.html.

Selection

<div><h3>Category</h3><div><div>Aves</div><div>Amphibians</div><div>Reptiles</div><div>Mammals</div><div>Fish</div><div>Mollusks</div><div>Arachnids</div><div>Insects</div></div><h3>Options</h3><div><input type="checkbox"/> Display Scientific Names</div><h3>Species</h3><div><input type="text" value="Filter species..."/></div><div><div>Aves</div><div>Northern Shoveler +</div><div>Red-crested Pochard +</div><div>Common Pochard -</div><div>Ring-necked Duck +</div><div>Ferruginous Duck -</div></div><div><div>Reset</div><div>Clear Map</div><div>Submit</div></div></div>	<div><div>Successfully fetched all info. X</div><div><h3>Category</h3><div><div>Aves</div><div>Amphibians</div><div>Reptiles</div><div>Mammals</div><div>Fish</div><div>Mollusks</div><div>Arachnids</div><div>Insects</div></div><h3>Options</h3><div><input type="checkbox"/> Display Scientific Names</div><h3>Species</h3><div><input type="text" value="Filter species..."/></div><div><div>Aves</div><div>Northern Shoveler +</div><div>Red-crested Pochard +</div><div>Common Pochard +</div><div>Ring-necked Duck +</div><div>Ferruginous Duck +</div></div><div><div>Reset</div><div>Clear Map</div><div>Submit</div></div></div></div>	<div><h3>Current Species</h3><div><div></div><div><div>Ferruginous Duck</div><div><i>Aythya nyroca</i></div><div>Wikipedia Entry</div></div><div><div>NT near threatened</div><div>Native</div></div></div><div><div></div><div><div>Common Pochard</div><div><i>Aythya ferina</i></div><div>Wikipedia Entry</div></div><div><div>VU vulnerable</div><div>Native</div></div></div></div>
<p>Dropdown with possible selections.</p>	<p>Selections are fetched and displayed.</p>	

Toggle

Options



Display Scientific Names

Species

Filter species...

Amphibians

Bombina variegata

Bufo bufo

Bufo viridis

Rana arvalis

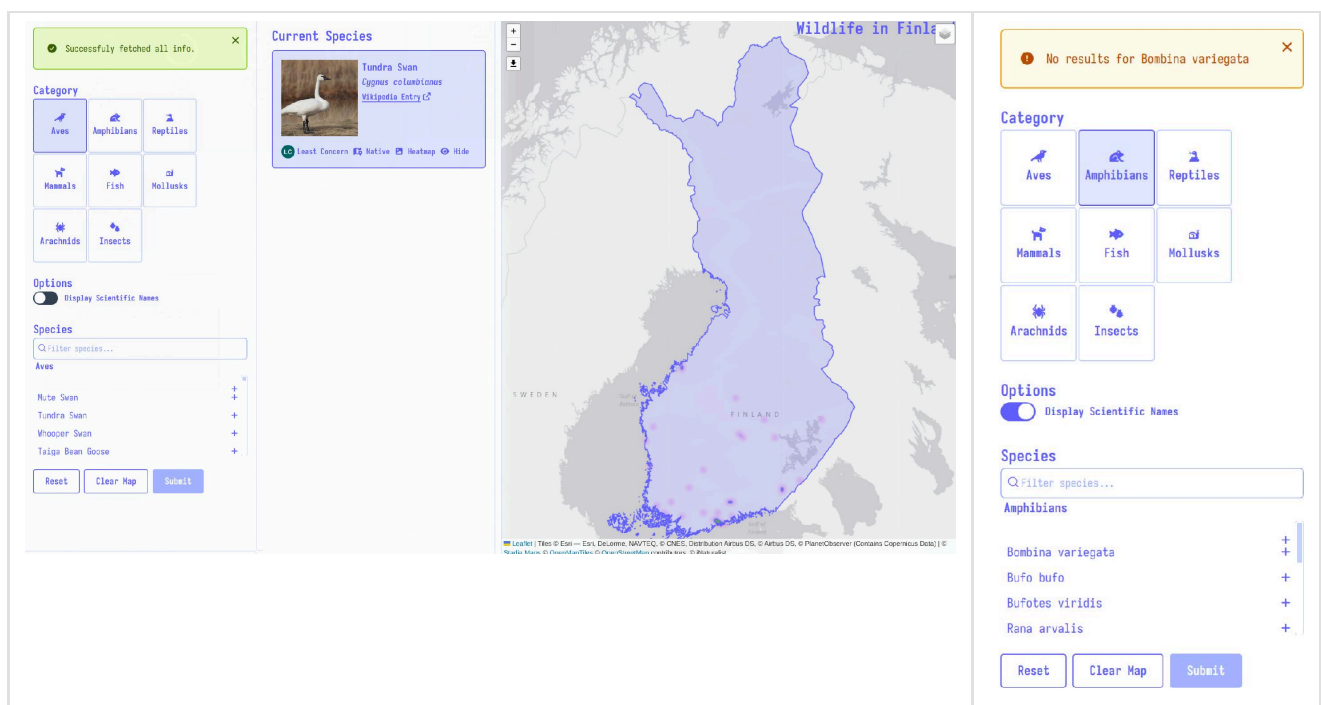
Reset

Clear Map

Submit

Toggle between scientific and common names.

Feedback




get direct feedback from the application.

"no results"
notification


Hide Heatmaps

Current Species




European Toad
Bufo bufo
[Wikipedia Entry](#)

LC Least Concern Native Heatmap Show



European Common Frog
Rana temporaria
[Wikipedia Entry](#)

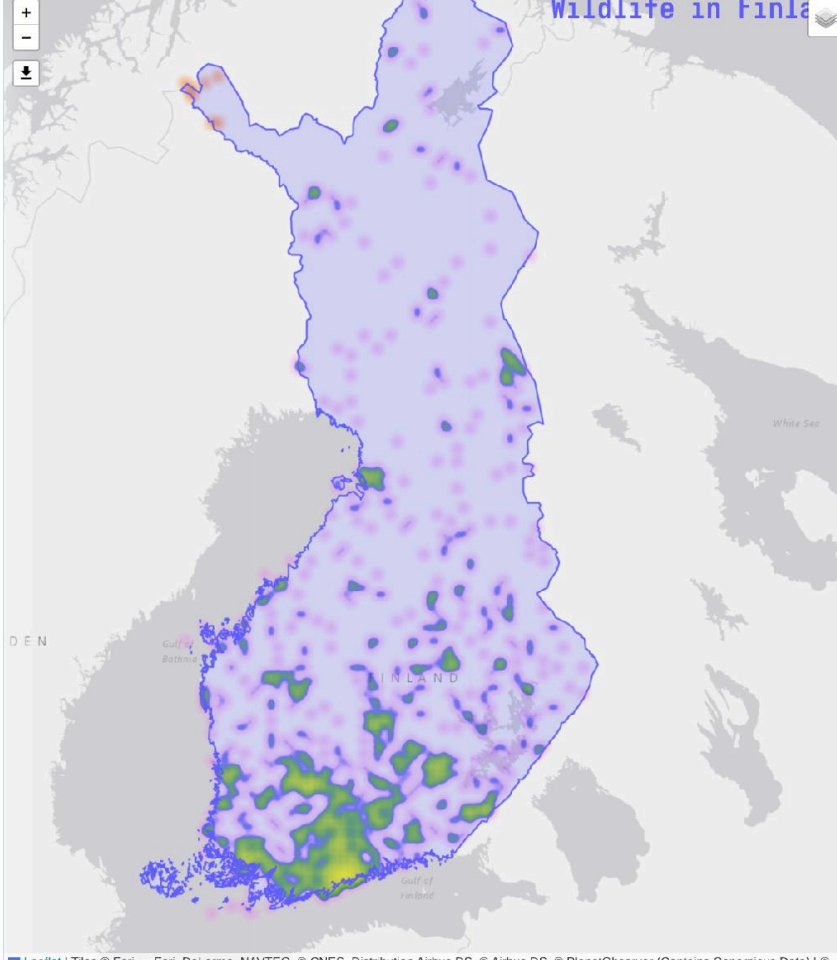
LC Least Concern Native Heatmap Hide



Moor Frog
Rana arvalis
[Wikipedia Entry](#)

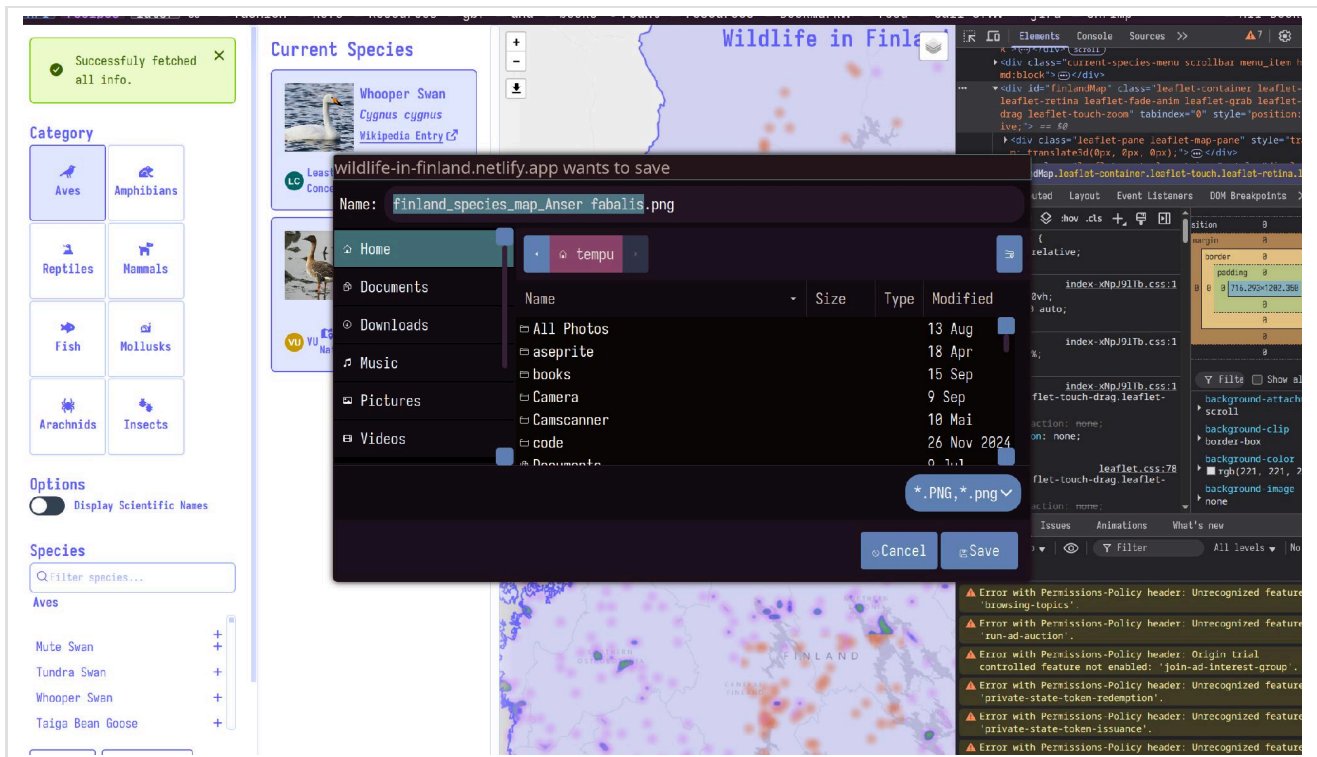
LC Least Concern Native Heatmap Show

Wildlife in Finland



hide individual heatmaps.

Save heatmaps











save individual heatmaps (or save the whole map!)

Clear Map

✔ Map cleared successfully.



Category

 Aves	 Amphibians	 Reptiles
 Mammals	 Fish	 Mollusks
 Arachnids	 Insects	

Options

☐ Display Scientific Names

Species

🔍 Filter species...

Aves

Mute Swan



Tundra Swan



Whooper Swan



Taiga Bean Goose



Reset

Clear Map

Submit



Clear the entire map, to start anew.

Resources

Icons

- [Fontawesome](#)
- [Iconify](#)

Styling

- [Toggle Styling](#)

Tile Layers

- [Leaflet Providers](#)

GeoJson

- [MapScaping](#)

datasets

- [iNaturalist v1 API](#)
- [Laji.fi for taxon lists](#)

Guides

- <https://flowbite.com/docs/forms/radio/>
- <https://gis.stackexchange.com/questions/385444/openstreetmap-baselayer-of-just-one-country>
- <https://gis.stackexchange.com/questions/179630/setting-bounds-and-making-map-bounce-back-if-moved-away>
- https://www.inaturalist.org/observations?place_id=7020&quality_grade=research&subview=map&iconic_taxa=Aves,Amphibia,Reptilia,Mammalia,Actinopterygii,Mollusca,Arachnida,Insecta
- <https://daily-dev-tips.com/posts/vanilla-javascript-replace-all-whitespaces/>
- <https://a11ymatters.com/pattern/checkbox/>
- https://api.inaturalist.org/v1/docs/#/Observation_Tiles
- <https://www.inaturalist.org/pages/api+recommended+practices>
- <https://forum.inaturalist.org/t/heatmaps-for-specific-taxa/43972/2>

- <https://forum.inaturalist.org/t/hotspot-view-for-species-genera-families-etc-taxon-not-just-for-users-and-places-projects/38264>
- <https://github.com/ryanmcdermott/clean-code-javascript?tab=readme-ov-file>
- <https://gomakethings.com/custom-events-with-vanilla-js/>

Tools

- [TSV to Json converter](#)