

Pokedex web app

Development
process



TABLE OF CONTENTS

01

Setting up the development environment

02

Creating a small in-memory array list of Pokemons

03

Iterating over the Pokemons array and applying basic initial styling

04

Enhancing the web app basic/initial user interface

05

Replacing the in-memory static list of Pokemons by the complete list fetched from the external API (using AJAX)

06

Implementing a modal for Pokemons in-detailed view

TABLE OF CONTENTS

07

Updating the app UIs using
Bootstrap

08

Completing final adjustments,
deploying the web app and
finalizing the README document

01



Setting up the development environment

Skills used

Code writing

WHAT WAS THE GOAL

Creating the project repository on Github and creating / linking together (when needed) the first initial files:

- *index.html*
- *style.css*
- *scripts.js*
- *README.md*



02

```
let pokemonList = [...]
```

Creating a small in-memory array list of Pokemons

Skills used

Code writing

WHAT WAS THE GOAL

Creating a small array of Pokemons in *script.js* and populating it with a few objects (each object representing one Pokemon and having the same keys) to have some content to work with over the next steps.

```
let pokemonList = [  
  { name: 'balbuzard', height: 7, types: ['grass', 'poison'] },  
  { name: 'charmander', height: 6, types: ['fire', 'air'] },  
  { name: 'squirtle', height: 5, types: ['water', 'grass'] }  
];
```



Iterating over the Pokemons array and **applying** basic initial styling

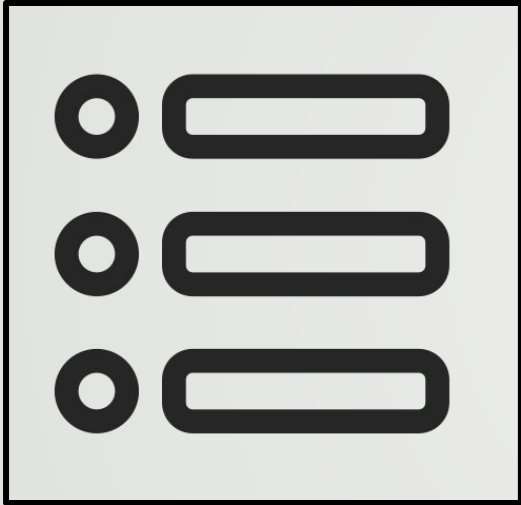
Skills used

Code writing

WHAT WAS THE GOAL

Creating a *forEach* function to iterate over each object within the array of Pokemons previously created and rendering them in the browser.

Applying basic visual elements to the web app, such as special fonts from Google Fonts, spacing adjustment, color and logo.



Enhancing the web app
basic/initial user interface

Skills used

Code writing

WHAT WAS THE GOAL

Updating the main UI to start giving it a real visual. More precisely, creating a button for each Pokemon in the array that appears in the browser, and adding an event listener to each of these newly created buttons that calls a *showDetails* function whenever a button is clicked.

```
function addItem(pokemon) {  
  let pokemonList = document.querySelector('.pokemon-list');  
  let listItem = document.createElement('li');  
  listItem.classList.add('list-group-item');  
  let button = document.createElement('button');  
  button.classList.add('btn', 'btn-danger', 'btn-block');  
  button.innerText = pokemon.name;  
  button.addEventListener('click', function () {  
    showDetails(pokemon)  
  })  
  listItem.appendChild(button);  
  pokemonList.appendChild(listItem);  
};
```



Replacing the in-memory static list of Pokemons by the complete list fetched from the external API (using AJAX)

Skills used

Research
Code writing
Debugging

WHAT WAS THE GOAL

Switching from displaying the list of Pokemons manually entered in an array to displaying Pokemons fetched from the external API (PokeAPI) using AJAX method.

To do this, a new *loadList* function has been created, meant to fetch the name of each pokemon from PokeAPI, as well as their URL redirecting to a page containing all their specific details.

In order to display each Pokemon's details, a second function named *loadDetails* has been created, allowing to display Pokemons images, heights, weights, types and abilities in an on-demand detailed view.

WHAT WAS THE GOAL (SUITE)

Adding polyfills in the code to carry out promise and fetch logics within older browsers that do not support those functionalities (polyfills are pieces of code that mimics newer JavaScript features for older browsers, thus allowing to use as many new features as necessary without worrying about whether or not the older browsers will support the functionalities). The picture on the right is a sample of the polyfill for the *fetch* element.

```
(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? factory(exports) :
  typeof define === 'function' && define.amd ? define(['exports'], factory) :
  (factory((global.WHATWGFetch = {})));
})(this, (function (exports) { 'use strict';

  var support = {
    searchParams: 'URLSearchParams' in self,
    iterable: 'Symbol' in self && 'iterator' in Symbol,
    blob:
      'FileReader' in self &&
      'Blob' in self &&
      (function() {
        try {
          new Blob();
          return true
        } catch (e) {
          return false
        }
      })(),
    formData: 'FormData' in self,
    arrayBuffer: 'ArrayBuffer' in self
  };

  function isDataView(obj) {
    return obj && DataView.prototype.isPrototypeOf(obj)
  }

  if (support.arrayBuffer) {
    var viewClasses = [
```



Implementing a modal for
Pokemons in-detailed view

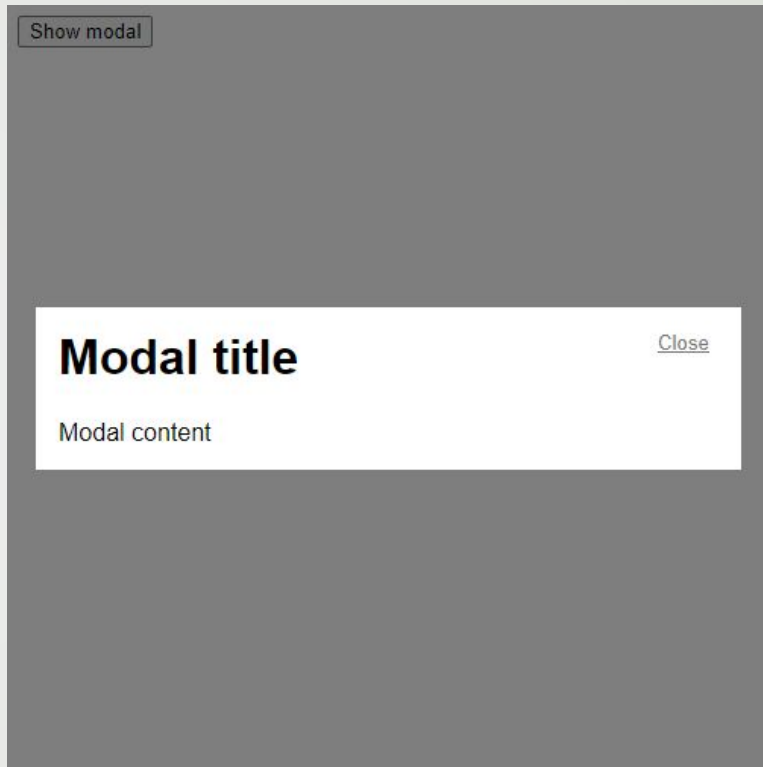
Skills used

Research
Code writing
Debugging

WHAT WAS THE GOAL

Creating and implementing a modal built with HTML, CSS, Javascript and jQuery to show more details about a Pokemon when users click on a Pokemon name / button.

While creating a modal from scratch isn't the most straightforward thing to do to use this UI element (using libraries that provide this feature makes it much faster and easier), it was still important to know how the code for a modal works in the first place. This is why this UI pattern was first implemented without a library - but later replaced by a Bootstrap modal (see next step).





Polishing the web app UIs using Bootstrap

Skills used

Research
Code writing
Debugging

WHAT WAS THE GOAL

Using Bootstrap to build nicer user interfaces.

Up to this point, all the web app was designed using custom CSS. While it was working, it wasn't the most efficient way to design the different visuals, considering Bootstrap offers many pre-written code for professional, quick and easily adjustable responsive layout.

The old codes related to visual design were therefore refactored and replaced with new Bootstrap features (buttons, modals, navigation bar, etc) for better UIs design and structure (while still keeping some useful CSS).

POKEDEX VIEW LARGE SCREEN VISUAL (COMPUTERS AND TABLETS)

Poked-Expert



Pokedex [Contact](#)

Pokemons

bulbasaur

ivysaur

venusaur

charmander

charmeleon

charizard

squirtle

wartortle

blastoise

caterpie

metapod

butterfree

weedle

kakuna

beedrill

POKEDEX VIEW LARGE SCREEN VISUAL (COMPUTERS AND TABLETS)

Poked-Expert

Pokedex Contact

wartortle



height : 10m

weight : 225kg

types : water

abilities : torrent, rain-dish

Close

bulbasaur

charmander

squirtle

caterpie

weedle

kakuna

venusaur

charizard

blastoise

butterfree

beedrill

POKEDEX VIEW SMALL SCREEN VISUAL (MOBILE PHONES)

Poked-Expert



Pokemons

bulbasaur

ivysaur

venusaur

charmander

chameleon

charizard

squirtle

wartortle

blastoise

caterpie

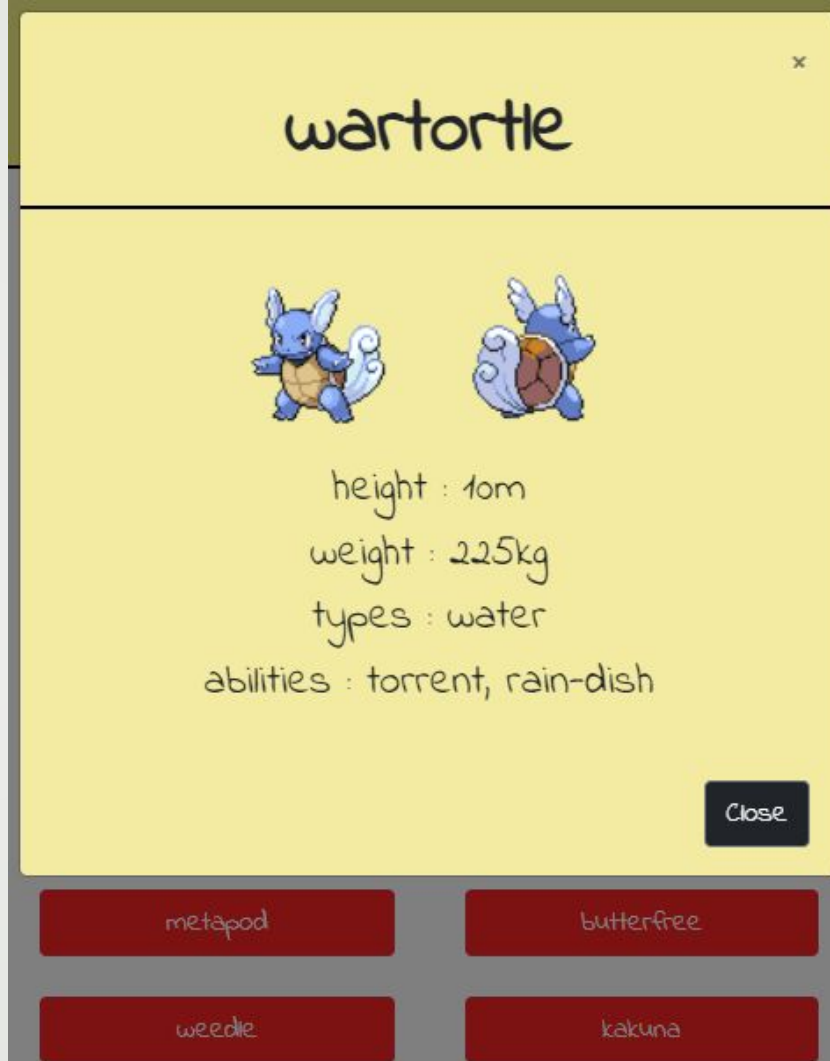
metapod

butterfree

weedle

kakuna

POKEDEX VIEW SMALL SCREEN VISUAL (MOBILE PHONES)



CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

Since I was in my first experiments with Bootstrap, I had some problems at the very beginning because the newly added features from the library did not behave as expected. I later understood that this was due to the interaction between old CSS codes and new Bootstrap features, which created some undesired interference. I therefore removed old CSS codes that were not necessary anymore or that were creating interference with the new Bootstrap features to ensure everything behaves as expected, which resolved the issue.



github-pages

Completing final adjustments,
deploying the web app and
finalizing the README
document

Skills used

Communication
Content writing

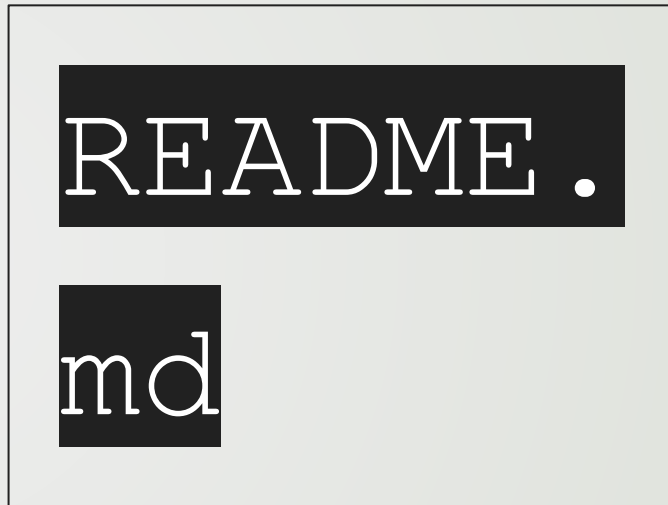
WHAT WAS THE GOAL

Making final styling adjustments to the web app to ensure it looks good and is fully responsive on any devices.

Carrying final testing to ensure everything is up and running before final deployment.

Deploying the web app on Github pages.

Completing the README file shown in the project's Github repository to ensure all information about the web app is available to anyone interested.



CHALLENGES OR SPECIAL POINTS OF CONSIDERATION

Finding the right balance between giving the right level of information, while remaining as synthetic as possible. To help me, I made a first draft, which I then modified at times. I also drew inspiration from other READMEs I've consulted for similar projects and for which I found that the information presented was relevant.

README SAMPLE - FULL VERSION ON GITHUB

