CAREERFOUNDRY

Achievement 1 Project: JavaScript App (Pokédex)

Objective

To build a small web application with HTML, CSS, and JavaScript that loads data from an external API and enables the viewing of data points in detail.

Context

The best way to learn about programming is to complete a project using that programming language. This project aims to do just that—have you build a complete, fully functioning JavaScript web application. Since you haven't yet learned how to build your own API, you'll be relying on an external data source for this project.

It's best to have complete projects that you can showcase in your portfolio. For this reason, it's important that you don't limit your focus to JavaScript and JavaScript alone, but, instead, consider the app as a whole, including high-quality HTML and CSS. Your app should not only work, but be aesthetically pleasing and easy to use, as well. No one expects you to be a designer (nor to have the perfectly designed app a designer would have), but your final app shouldn't look like a half-finished prototype.

In the following sections, you'll find more information regarding the requirements for your app.

The 5 Ws

1. Who—Your professional network, as well as potential employers.
2. What—A JavaScript app built with HTML, CSS, and JavaScript, which uses an external API.
3. When—Immediately! The code for the app can always be viewed in a repository like GitHub or GitLab.
4. Where—Potential employers, clients, and collaborators might look at the code directly from your GitHub repository or be directed to your GitHub profile from your portfolio site (your project from Intro to Frontend Development).
5. Why—For your potential employers, clients, and collaborators to see how you create the architecture of a JavaScript app, as well as how you test and debug your code.

Features and Requirements

User Goals

Your users should be able to view a list of data and see more details for a given data item on demand. It's up to you to come up with a type of data you want to display (see this list of public APIs for an example); however, if you don't have any special preferences, you can write a small Pokédex app to display a list of Pokémon.

Key Features

- Load data from an external source (API)
- View a list of items
- On user action (e.g., by clicking on a list item), view details for that item

Technical Requirements

Required:

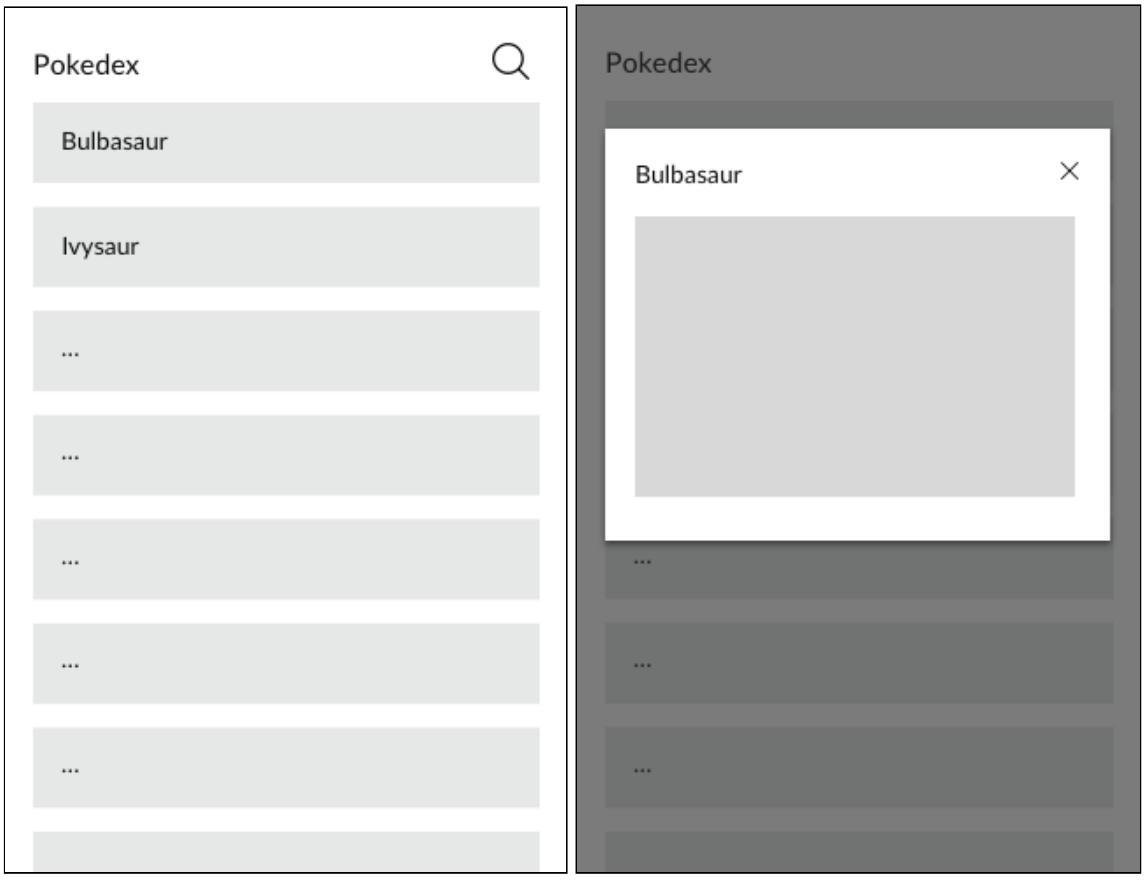
- The app *must* load data from an external API; for instance, the Pokémon API.
- The app *must* display a list of items loaded from that API after the page is loaded.
- The app *must* enable the viewing of more details for a given list item (like a Pokémon) on demand, such as when clicking on a list item.
- The app *must* have CSS styling.
- The JavaScript code *must* be formatted according to ESLint rules.
    - The JavaScript code *may* be formatted via Prettier.
    - The JavaScript code *may* be manually formatted.
- The app *must* use at least one additional complex UI pattern, such as a modal, for details or touch interactions.
    - The app *may* allow searching for items (e.g., searching for Pokémon).
- The app *must* not throw any errors when being used.
- The app should be deployed to a publicly accessible platform like GitHub Pages (you can review how to do this in Intro to Frontend Development's Exercise 10: Code Quality, Testing, & Web Hosting).
- The app *must* work in Chrome, Firefox, Safari, Edge, and Internet Explorer 11.

Nice to Have:

- The app should show loading indicators while loading data.
- The app should handle errors (such as trying to load data while offline) and show user-friendly error messages.

Mock-ups or Other Assets

You'll see mock-ups for a simple Pokédex app below. We'll keep things simple, as the goal is, first and foremost, to have a functional app. Note that the following mock-ups are provided primarily for reference purposes, and depending on the type of data you're displaying, your app may look different. Feel free to get creative and use a different layout for your data should you so please!



Your Project Deliverables

Throughout this Achievement, you'll be working from Exercise to Exercise to complete your Achievement project. For each task, you'll submit a deliverable that directly contributes to the final

product—in this case, your JavaScript app. In Exercise 9, you'll take a break from the Pokédex app to learn about jQuery and you'll complete a short jQuery mini-project: a basic To-Do-List app.

Since the main goal of this project is to learn programming while programming, it's especially important to adhere to good coding practices and maintain a neat, clean codebase. Take all the time you need for each step—it's better to think something through twice if you're unsure about it than to speed ahead and make a mistake.

Below is an Exercise-by-Exercise breakdown of your various project deliverables:

Exercise 1: What is JavaScript?

- Set up the general outline of your app.

Exercises 2 and 3: JavaScript Basics Part 1 & 2

- Learn coding practices focusing on the core principles of JavaScript: objects, arrays, conditionals, and loops.
- Set up a general dataset for your application and display it in a very basic way to the user.

Exercises 4 and 5: JavaScript Functions Part 1 & 2

- Learn about JavaScript functions.
- Learn about the core principles of functional programming (FP) and its comparison with object-oriented programming (OOP).
- Implement a forEach loop to print details on each item in your app's repository.
- Wrap your app's repository in an IIFE to avoid accidentally accessing the global state.
- Update your forEach loops to work within the IIFE.

Exercise 6: DOM Interaction

- Render static data from your data repository on the page.
- Add basic styling to your app.
- Make sure your application is accessible.

Exercise 7: APIs, Ajax & Asynchronous Behavior

- Load data from an external API via fetch.

Exercise 8: Complex UI Elements with JavaScript

- Add one or multiple complex UI patterns to your app.
    - For example, show data details in a modal.

Exercise 9: Introduction to jQuery

- Learn about the popular JavaScript library, jQuery.
- Build a basic To-Do-List app using jQuery.

Exercise 10: Bootstrap & UI Libraries

- Use Bootstrap to create a clean and usable design for your app.

Exercise 11: Performance & Debugging

- Ensure your app works in all necessary browsers.
- Add ESLint rules and ensure your code passes the validation test.
- Make final adjustments to your app such as styling or usability improvements.

Optional: Advanced Deliverables

In addition to all of the above, you can add more advanced features to your app as you wish. Below are some topics you could explore in your app:

- Touch interactions (pointer events), e.g., swiping between items
- Adding a loading indicator
- Making the app more aesthetically pleasing or including more features, such as an item search