IP Phase 1 Code Challenge: Flatacuties [Compulsory]

- Due No Due Date
- Points 15
- Submitting a website url or a file upload

Phase-1-code-challenge-Flatacuties.zip (https://moringa.instructure.com/courses/648/files/412996?wrap=1)

Download the file above and unzip it to get started. Refer to the previous code challenge instructions for the process of starting and submitting this challenge.

Learning Goals

- Implement a mini web app to practice on array iteration, DOM, Events and Server communications.

Introduction

For this challenge, you will be working on *Flatacuties*, an app where you can vote for the cutest animal! You will be using a local API and building out the frontend for our app, *Flatacuties*.

The instructions below will walk you through the process of ideation and planning your app: deciding on your user interface, planning how the information will be laid out on the page, etc. You should work through all the planning steps before you start doing any coding.

Requirements

For this project, you must:

- Have a well-written README file.
- Fetch data from a local server running JSON DB server.

Project Setup & Pre-requisite Data

- In your project directory, create a *db.json* file and use this **data (https://docs.google.com/document/d/1EUcHU9gkydR3lfJDTebW5iNHP2BCMRcv508R7BAXSvo/edit?usp=sharing)** for your server DB.
- Run the following command to get the backend started:

  json-server --watch *db.json*

- Test your server by visiting this route in the browser:

  http://localhost:3000/characters

Project Setup

Once you have the plan in place for the application you want to build take the following steps:

- Create a new project folder.
- Create a new GitHub repository (**NB: ENSURE IT IS PRIVATE**).
- Add your TM as a contributor to the project. (**This is only for grading purposes. We promise we won't steal your code**)
- Please make sure you regularly commit to the repository.

Project Guidelines

Your project should conform to the following set of guidelines:

Core Deliverables:

As a user, I can:

1. See a list of all animal names. You will need to make a GET request to the following endpoint to retrieve the character data

GET /characters

Example Response:

```
    {
      "id": 1,
      "name": "Mr. Cute",
      "image": "https://thumbs.gfycat.com/EquatorialIckyCat-max-1mb.gif",
      "votes": 0
    },
    {
      "id": 2,
      "name": "Mx. Monkey",
      "image": "https://thumbs.gfycat.com/FatalInnocentAmericanshorthair-max-1mb.gif",
      "votes": 0   }, ….]
```


2. Click on an animal’s name to see its details i.e image and number of votes. Note, you should only be displaying the details of one animal at a time. You can either use the character information from your first request, or make a new request to this endpoint to get the character's details.

GET /characters/:id

Example Response:


```
 {
      "id": 1,
      "name": "Mr. Cute",
      "image": "https://thumbs.gfycat.com/EquatorialIckyCat-max-1mb.gif",
      "votes": 0
```

```
    },
```

3. When viewing an animal's details, I should be able to add the number of votes for each animal. This number of votes should then be displayed together with the animal's details. **No persistence is needed for the votes.**

Bonus Deliverables

These bonus deliverables are here if you want an extra challenge and won't affect your score. **Make sure to commit your work to save your progress before attempting the bonus deliverables!**

1. Add a reset button that resets the votes back to 0
2. Have a form for adding animals.

Flatacuties Rubris

Criteria	Ratings					Pts
DOM Manipulation	**5 pts** **Full Marks** All of the above, plus added additional functionality not described in the deliverables.	**4 pts** **4 pts** Structured HTML creation code cleanly and in a reusable way, using a semantically correct HTML structure without any unnecessary elements.	**3 pts** **3 pts** Successfully rendered and updated the DOM as described by the Deliverables.	**2 pts** **2 pts** Rendered elements to the DOM, but with some errors.	**1 pts** **1 pt** Did not properly render elements to the DOM.	5 pts
Event Handling	**5 pts** **Full Marks** All of the above, plus added additional functionality not described in the deliverables.	**4 pts** **4 pts** Structured code in a clean and reusable way, splitting functions, using descriptive names and using target properties effectively.	**3 pts** **3 pts** Successfully attached event listeners to handle DOM events and met all of the Deliverables.	**2 pts** **2 pts** Attached event listeners, but incompletely or with some errors.	**1 pts** **1 pt** Did not attach event listeners to respond to events.	5 pts
Communication with the Server	**5 pts** **Full Marks** All of the above, plus added additional functionality not described in the deliverables.	**4 pts** **4 pts** Code structured in a clean and reusable way, splitting into functions and reusing them where needed, with clear function and variable naming.	**3 pts** **3 pts** Able to perform a GET request successfully as described in the Deliverables.	**2 pts** **2 pts** Partially able to communicate with the server, but incompletely or with some errors.	**1 pts** **1 pt** Unable to communicate with the server.	5 pts
						Total Points: 15