UDACITY

PROJECT SPECIFICATION

Mars Dashboard

Environment Setup

CRITERIA	MEETS SPECIFICATIONS
Create a Node and Express Environment	App must use a Node/Express backend. Install the following `npm` packages in order to work with the API and build the rest of the app: • Express • Body parser • Dotenv • ImmutableJS Bolded files/folders are part of the Backend File structure.
Create a Javascript Frontend	Setup the basic frontend dependencies and file structure for a single page application. Some starter code will be provided. Bolded files/folders are part of the Frontend File structure.
Immutable JS installed	Set up Immutable js for this project. For this project we are using a script for the CDN version. You should see the script referenced in `index.html`.

CRITERIA	MEETS SPECIFICATIONS

API and Routes

CRITERIA	MEETS SPECIFICATIONS
Create API Credentials	Set up credentials to connect to the NASA API. • Create an API key through NASA's API. • Add API key to the provided `.env-example` file. • Rename the `.env-example` file to `.env`. **Note**: The `.env-example` file is publicly visible and should not contain any personal information.
Show most recent information from the API	Create a dashboard showing photos and information about the Mars rovers using the NASA API Needs to display a minimum of this information for each rover (there are 3): • Launch Date • Landing Date • Status • Most recently available photos • Date the most recent photos were taken
Create Requests for the Desired Information	Set up the Node/Express app as a middleman to take requests from the frontend, make requests to the API, and pass that information back to the frontend using functional methods.

Functional JS Logic

CRITERIA	MEETS SPECIFICATIONS
Use Pure Functions to organize logic	Build custom pure functions to do logic. No logic should happen outside of these functions.

Use of const/let/var correctly	Use the variable types in the appropriate instance:
	
	• `const` for all values that will not change
	  ◦ `let` for those that will need to change
	  ◦ `var` **only** if an instance comes up where a globally editable value is necessary

Use of array methods correctly	Use the correct method to fit the information they are working with from the API. In the logic you should be able to see: • At least one custom pure function using `map`, just about every API call will require this at some point. • At least one dynamic component on their page (for instance, one that uses an `if statement` to behave differently based on the presence or absence of a value).
Use higher-order	Use at least **2** higher-order functions that are reusable UI elements (i.e. an unordered list function that is

functions	returned by the other dashboard section functions).
CRITERIA	MEETS SPECIFICATIONS

Dynamic UI

CRITERIA	MEETS SPECIFICATIONS
Create an interactive UI	The UI should have a dynamic aspect that allows a user to choose which rover's information they want to see. This will most likely be a set of tabs that can be clicked to see the information.
Create appealing page with mobile first styling	Build upon the given mobile first styling for mobile or desktop. Must be a breakpoint for mobile that comes first in the stylesheet.

Suggestions to Make Your Project Stand Out!

1. Find a good use for ForEach, Reduce, Find, Includes, or Sort methods.
2. Explore other endpoints of the NASA API such as the following:
    - Incorporate the apod image in the styling.
    - Add the Mars weather embed code.
    - Add a wind graph at the location of the rover.
3. Pay extra attention to styling.