DMIT2008 Assignment 1: Current Weather Display App

Introduction

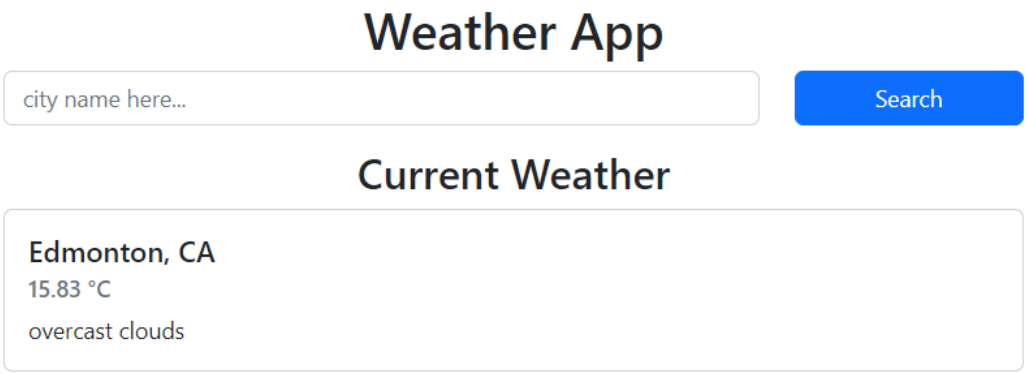
This assignment will test most of your knowledge of our JavaScript Review in DMIT2008 by displaying the current weather of a given city. This will focus on your knowledge of using NPM, Node.js, parcel, Importing and Exporting functions from other files.

Note you'll be using OpenWeatherMap api (and more specifically the Get Weather Endpoint) which you'll have to sign up for it and use an API key in your requests (more on that later in the assignment). You might find it useful to use a RESTful API Client that you have most likely been shown in class so that you peruse the javascript objects

Overview of functionality

Here is the sample functionality in the image below.

- When you click "search" after entering a city, the application will use the OpenWeatherMap api to get the current weather of the City that has been searched (also it will clear the input of the



Required Tasks

- Install the packages bootstrap and parcel, and setup the required scripts using npm.
    - Your project should be using parcel and not live server when ran (it won't work anyways if you try it that way).
    - Your scripts need to be correct and should be the same as other projects you have done in class with your instructor.
    - Bootstrap should be included in your index.js file so your project looks like the images.
    - NOTE: In your solution there should be no node_modules folder or "parcel-cache" folder. Marks will be taken off if they are included.
- Create a function named "getWeather" (that takes a single cityName as a parameter) in the "api/base.js" file that will use (and return) fetch to call the Rest Endpoint of Get Weather Endpoint, return all of the data from that REST API endpoint.

    - Your function should be a promise and should resolve and return the data so that the function itself is a promise.
    - This needs to be in the "api/base.js" file or you'll be penalized on this.
    - Export that function from that file.
- Create a function named "renderWeather" in the "dom/weather.js" file that will use the DOM API to overwrite the innerHTML of the weather-container element.
    - This needs to be in the file "dom/weather.js" file or you'll be heavily penalized on this.
    - It needs two arguments:
        - Weather data (that you'll pass in from the api endpoint)
        - The element that you'll want to change the innerHTML.
    - It will use the HTML in the "dom/weather.js" and use a template string to replace the strings below with the content from the weather data that you'll be passing in. Note: you'll have to use your knowledge of javascript objects and arrays to get all of the data.
        - CITY_NAME_HERE
        - COUNTRY_CODE_HERE
        - CURRENT_WEATHER_DEGREES_HERE
        - WEATHER_DESCRIPTION_HERE
- In the index.js file import the two functions created above from the "dom/weather.js" and "api/base.js" files.
- In the index.js, select the form (with the id "weather-search") and add an event listener that listen to "submit" events and prevent the form from being submitted.
    - In the event handler (of the event listener) the code will call the "getWeather" function with the value of the form input.
    - In the resolved promise from the "getWeather" function the "renderWeather" function will be called with the data from the RESTful API call and the selected element the class "weather-container".

Marking key

Tasks	Grade	Marks	Total
NPM and Packages. • Packages are installed correctly and are in the correct dependency section (dev dependency vs dependency) • Scripts are setup correctly and the source attribute in the package.json is setup correctly. • Node_modules folder and parcel cache are included (if you included them it's negative marks on your assignment)		**3** **3** **-3**	
Fetch • The fetch request is takes in a city, and returns the data from the fetch request as a promise. • The function "getWeather" is in the correct file and exported correctly • The function "getWeather" is imported correctly in the index.js file		**5** **1** **1**	

Dom API and Manipulation. • The “renderWeather” function is in the correct file and exported correctly. • The function “renderWeather” is imported correctly in the index.js file • The “renderWeather” item function uses template strings and the correct attributes from the JavaScript object to display the weather correctly. • The form is intercepted and prevented from submitting in the index.js. • The event handler of the form calls the “getWeather” api function and displays the correct information on the page.		**1** **1** **5** **1** **5**	
• **Code Formatting and Style**		**-3**	

Marking Rubric

Marks	5 Marks Criteria
5	**Task was completed with the highest of proficiency adhering to best practices and followed subject matter guidelines all tasks were completed to a professional standard.**
4	**Task was completed well some minor mistakes. Well above average work shows good understanding of the task and high degree of competence**
3	**Satisfactory work some features missing or incorrectly implemented. Show a moderate level of understanding in the task with room for improvement.**
2	**Below average work. Task was poorly complete. Show understanding of the task and the requirements to implement but implementation was poorly executed.**
1	**Some of the task was completed. Showed a lack of understanding in the subject matter and very poorly executed**
0	**Not completed.**

Marks	**3 Marks Criteria**
3	**Proficient shows a high degree of competence in completing task.**
2	**Capable above average degree of competence in completing task**
1	**Satisfactory shows a satisfactory degree of competence in completing task.**
0	**Shows a limited degree of competence in completing task.**

Marks	**1 Marks Criteria**
1	**Task Completed satisfactorily**
0	**Task was not executed.**