



DMIT2008 Assignment 2

Task: Weather App Templates and Styling Update

Due: Midnight on Friday October 26th, 2018

Weight: 10%

Overview

You must update the existing ES6 Weather App to make use of Handlebars templates and a CSS preprocessing library for styling. All dynamic data display (i.e. current weather and forecast) must be rendered via templates and your application must utilize a CSS preprocessor to update the aesthetic of the user interface.

Expectations

Download and extract the included application zip starter package. Your updated application must demonstrate the following:

- Essentially all requirements of the previous assignment must be adhered to for this assignment, e.g.:
 - Use of Babel.js for compiling to ES5
 - No trace of any third-party libraries, other than those specified in this document
 - Properly documented code using an acceptable commenting convention (e.g. [JSDoc](#))
 - Code must adhere to a widely accepted code convention for code style
 - Babel.js (via node.js and npm) is to be used to compile main.es6.js (ES6) to main.js (ES5)
- Current weather and forecast display must be implemented via Handlebars templates
 - Template(s) must be compiled and only the runtime should be included in the page (i.e. not the full Handlebars library)
 - You must use appropriate built-in helpers where appropriate
 - **[Bonus]** You must use a custom helper to format temperature output as either degrees Celsius (°C) or degrees Fahrenheit (°F)
 - **[Bonus]** Implement the forecast template as a partial
- You must choose a CSS preprocessor for implementing your application style
 - Emphasis is not on what 'looks good' but rather on use of the chosen library

- Implementation of variables, nesting, and mixins at minimum must be included
- [**Bonus**] Implement a temperature switch (either °C or °F) to allow the user to choose the desired measure of temperature (note: how you implement the code for this is up to you, but it must 'work')
- You are solely responsible for all parts of this assignment (all JS code)
- **Any other requirements as laid out by your instructor in class**

The emphasis in this assignment is on functionality, which essentially means that the aesthetics are of lesser (if any) concern. Get your application to work correctly and demonstrate **best practices** and solid understanding of JavaScript in the process.

Delivery

Zip your project folder and submit it to Moodle by the deadline.

- **Do not include the node_modules/ directory in your zip package.**

Seek help if you need it; **Late submissions will not be graded.**

Grading Key

Tasks	Grade	Marks	Total
Handlebars Template used for weather and forecast display (separate file under src directory) Templates are pre-compiled for distribution (dist directory) Appropriate built-in helpers used (e.g. each, if, etc.) A handlebars script is added to package.json for running Handlebars, which will compile templates Helper implemented to render temperature as degrees C or degrees F (pass the helper a parameter to make this work) [bonus] Forecast template is rendered as a partial for the overall weather display [bonus]		5 3 3 1 [3] [1]	12 [16]
Preprocessor Either Less or Sass are used for styling (choose one) Demonstration of variables, nesting, mixins, and anything else you want to use in the less/sass file Both scss/less and Compiled CSS files present in css directory, only compiled CSS linked to index.html A less or sass script is added to package.json for running Less/Sass, which will compile less/scss to css Implement a radio button/switch to select either degrees F or degrees C for temperature units [bonus]		1 5 1 1 [3]	8 [11]
Node.js and Project Formatting Required npm packages are installed Build script is updated to include the handlebars script and less/sass script All development files (raw templates, sass or less files, etc.) are contained in the src directory; all compiled and		1 1 1	5

production ready files are contained in the dist directory		1	
Only production ready files from the dist directory are linked in the index.html file		1	
Code Readability and Efficiency <ul style="list-style-type: none"> • Efficient and maintainable techniques • Code format • Well documented • Etc. • Utilizes your own written code (i.e. not gathered from online sources such as StackOverflow) 		-5	-5

TOTAL MARKS		25
--------------------	--	----

Marking Rubric

Marks	5 Marks Criteria [minus]
5 [0]	Task was completed with the highest of proficiency, adhering to best practices, and followed subject matter guidelines. The task was completed to a professional standard.
4 [-1]	Task was completed well, with some minor mistakes. Well above average work, shows good understanding of the task, and a high degree of competence.
3 [-2]	Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement.
2 [-3]	Task completion is below average, the task was poorly completed. Shows understanding of the task and the requirements to implement, but implementation was poorly executed.
1 [-4]	Some of the task was completed. Shows a lack of understanding in the subject matter and very poor execution.
0 [-5]	Not completed.

Marks	3 Marks Criteria [minus]
3 [0]	Task was completed well, adhering to best practices, and followed subject matter guidelines.
2 [-1]	Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement.
1 [-2]	Some of the task was completed. Shows a lack of understanding in the subject matter and very poor execution.
0 [-3]	Shows a little to no degree of competence in completing the task; not completed.

Marks	1 Marks Criteria
1	Task completed satisfactorily
0	Task was not completed