

Student Name : Jonathan O'Leary
 Student Number : [20104311](#)
 Github Url : <https://github.com/joleary95/weather-Top-2-js>
 Deployed URL : <https://weather-top-js-r3.glitch.me/>

| Grade Band | Reading | Station | Member | Features | Code |
|------------------|-------------------------------------|---|--|---|---|
| Baseline | <i>Code Temp Wind Speed</i> | <i>Station Name Latest weather, Temp C, F, Wind Bft, pressure</i> | <i>None</i> | <i>Load and display stations + their readings from json file + display latest weather for station</i> | <i>Zipped archive + readme</i> |
| Release 1 | <i>+ Wind Direction</i> | <i>+ Wind Chill, Wind Compass</i> | <i>None</i> | <i>Dashboard shows station list + button to open station view. Include forms to add new Station + new reading</i> | <i>Glitch Project https://weather-top-2-r1.glitch.me/dashboard</i> |
| Release 2 | | <i>+Lat, Lng Max/Min (Temp, Wind, Pressure)</i> | <i>First Name, Last Name, Email, Password + Stations</i> | <i>+ Members can signup/log in. Members may create any number of weather stations.</i> | <i>Glitch Project & https://weather-top-js-r2.glitch.me/login Github repo https://github.com/joleary95/weather-Top-2-js/tree/release-2</i> |

| | | | | | |
|------------------|----------------------------|----------------------------|---------------------------------------|--|--|
| Release 3 | + Time/ Date | Temp, Wind Pressure Trends | User can edit their personal details. | Members can signup/login and add stations of there choice . (Sample) Readings loaded from a data YAML file | Deployed https://weather-top-js-r3.glitch.me/ + Github repo + history tags) https://github.com/joleary95/weather-Top-2-js/tree/Release-3 |
| Release 4 | <u>Not Complete</u> | <u>Not Complete</u> | <u>Not Complete</u> | <u>Not Complete</u> | <u>Not Complete</u> |

Extra Features :

I don't have any extra features

Technology used:

- JavaScript
- Vs Code
- Bulma
- Glitch

Screen Resolution:

My screen resolution is **3456 × 2234**. I found this by looking at the 'About This Mac' section located when clicking the apple icon in the left hand side of the screen.

Source of text, images & icons:

Text :

I pulled some of my text from other web stations. Then I used chat gpt to generate an extra Bit for me to go with the text to fill more of the page.

Images :

I used one image I sourced from imagur <https://i.imgur.com/EBJ6lrF.jpeg>

Icons :

I used a series of font awesome icons within various locations within my project.

Reflection:

The WeatherTop app was quite challenging and it took quite a while for me to understand the syntax difference in comparison to java at first. It became more understanding once I got my first latest reading published to my cards. As there was a lot of moving between javascript and html. While passing multiple parameters. Doing this over and over helped me grasp a better understanding of how both languages work simultaneously. Unfortunately, due to other commitments, I didn't get to attempt Release 4 before the deadline but was looking forward to it, so will I am to continue this project. This was the result of I got stuck on passing the trends to the view. I got the code working from the javascript side but couldn't implement the arrows and as a result, didn't get the icons either.

I think if I had of started even a little bit sooner I could of had the time to research handlebars helpers more in depth.

All in all, I really enjoy working in the backend more than the front end, either way again I've enjoyed making this App - as ever project teaches you more by getting heavily involved.

Use of AI:

I didn't rely on AI extensively, but since our project involves multiple languages, I found that when attempting to utilize it for complex explanations, it often struggled to fully understand the precise requirements of our application. This sometimes resulted in the generation of incorrect or non-specific code that didn't align with the specific needs of our app.

However,

I found it useful to grasp a range of explanations about something or when searching for something. For instances -

1.

what does this error mean

```
/usr/local/bin/node ./server.js
```

```
Uncaught SyntaxError: Identifier 'tempC' has already been declared
```

```
  at moduleStrategy (internal/modules/esm/translators:119:18)
```

```
  at moduleProvider (internal/modules/esm/loader:468:14)
```

```
--- await ---
```

```
  at processTicksAndRejections (internal/process/task_queues:95:5)
```

```
--- await ---
```

```
  at runMainESM (internal/modules/run_main:55:21)
```

```
  at executeUserEntryPoint (internal/modules/run_main:78:5)
```

```
  at <anonymous> (internal/main/run_main_module:23:47)
```

```
Process exited with code 1
```

ChatGPT

The error message you're encountering, "Identifier 'tempC' has already been declared," indicates that you're trying to declare a variable named `tempC` in a scope where it has already been declared previously. JavaScript does not allow re-declaring variables using the `const` or `let` keywords within the same scope.

2.

what is happening here

```
if (lastReading) {
```

```
  const lastTemperatureCelsius = lastReading.temp;
```

```
  const lastTemperatureFahrenheit = conversions.celsiusToFahrenheit(lastTemperatureCelsius);
```

ChatGPT

In the code snippet you provided:

javascript

Copy code

```
if (lastReading) { const lastTemperatureCelsius = lastReading.temp; const lastTemperatureFahrenheit = conversions.celsiusToFahrenheit(lastTemperatureCelsius); // ... }
```

Here's what's happening:

`if (lastReading)`: This condition checks if `lastReading` is truthy, meaning it's not null, undefined, or any other falsy value. It's a safety check to make sure that you actually have a valid last reading before proceeding.

`const lastTemperatureCelsius = lastReading.temp`;: This line retrieves the temperature value (`temp`) from the `lastReading` object and assigns it to the variable `lastTemperatureCelsius`. This assumes that your `lastReading` object has a property named `temp` that holds the temperature value in Celsius.

`const lastTemperatureFahrenheit = conversions.celsiusToFahrenheit(lastTemperatureCelsius)`;: This line takes the Celsius temperature value (`lastTemperatureCelsius`) and uses the `celsiusToFahrenheit` function from your `conversions` module to convert it to Fahrenheit. It assigns the Fahrenheit value to the variable `lastTemperatureFahrenheit`.

In summary, this code block takes the latest temperature reading in Celsius, converts it to Fahrenheit using the `celsiusToFahrenheit` function, and assigns both Celsius and Fahrenheit values to separate variables for further use in your analytics calculations or other purposes.

3.

would these files cause the application not to run? if they were made before opening the project in glitch from git?

ChatGPT

If files that are crucial for your application's functionality are ignored by `.gitignore` and not present in your Glitch project, it could indeed cause your application not to run correctly or even at all. The `.gitignore` file specifies patterns of files that Git should ignore when tracking changes, and if files necessary for your application are ignored, they won't be included in the repository when you clone or pull it.

For example, if configuration files, dependencies, or essential code files are ignored, your application may not have the required resources to function properly. If you created these files before opening the project in Glitch from Git, and they were ignored by the `.gitignore` file, Glitch might not have fetched these files from the repository, resulting in errors or malfunctioning.

To resolve this issue, you will need to review your `.gitignore` file and make sure that it doesn't exclude any files or directories that are necessary for your application to run. You might need to remove or modify the relevant entries in the `.gitignore` file to ensure that the required files are included in the repository and subsequently available in your Glitch project.

Always be cautious when making changes to your repository's files and `.gitignore` settings, as they can have significant impacts on your project's functionality and organization.

4.

where do i put this

```
// Register the eq helper
const Handlebars = require('handlebars');
Handlebars.registerHelper('eq', function (a, b, options) {
  if (a === b) {
    return options.fn(this);
  }
  return options.inverse(this);
});
```

ChatGPT

You should put the code to register the eq helper in a JavaScript file that is executed before you render your Handlebars templates. Typically, this code should be part of your application's setup or initialization process.

Here's a step-by-step guide on where to put it:

Create a Helpers File (Optional): You can create a separate JavaScript file for your Handlebars helpers. This helps keep your code organized. For example, you can create a file named `handlebars-helpers.js`.

Place the Helper Registration Code: Inside your `handlebars-helpers.js` or any appropriate JavaScript file, place the code to register the eq helper:

javascript

Copy code

```
// Import Handlebars if necessary const Handlebars = require('handlebars'); // Register the eq helper
Handlebars.registerHelper('eq', function (a, b, options) { if (a === b) { return options.fn(this); } return
options.inverse(this); });
```

Include the Helper File: Make sure to include or import the `handlebars-helpers.js` file in your main application file or wherever you set up your Handlebars environment.

Ensure Proper Execution Order: Ensure that this code runs before you start rendering your Handlebars templates. Typically, it should be executed during your application's initialization phase.

Here's an example of how your main application file might look with the helper registration code:

javascript

Copy code

```
// Import necessary modules const Handlebars = require('handlebars'); const express = require('express');
const app = express(); // Register Handlebars as the view engine app.set('view engine', 'hbs'); // Register
Handlebars helpers require('./handlebars-helpers'); // This is where you include your helpers file // Other
setup code... // Start your Express server app.listen(3000, () => { console.log('Server is running on port
3000'); });
```

Make sure to adapt this code to your specific project structure and setup. The key is to ensure that the eq helper is registered before you render your Handlebars templates.

References:

Bulma - <https://bulma.io/documentation/components/card/>

Bulma - <https://bulma.io/documentation/columns/sizes/>

Font Awesome - <https://fontawesome.com/icons/folder-open?f=classic&s=solid&an=fade&pc=%23000000>

FJOLT

- <https://fjolt.com/article/javascript-promise-all#:~:text=The%20Promise.,promise%20to%20finish%2C%20before%20continuing.>

W3schools

- https://www.w3schools.com/jsref/jsref_sort.asp

- https://www.w3schools.com/js/js_object_definition.asp

Handlebarsjs

- <https://handlebarsjs.com/guide/expressions.html#helpers>

- <https://handlebarsjs.com/guide/block-helpers.html#basic-blocks>

Stack Overflow

- <https://stackoverflow.com/questions/59916094/where-to-put-handlebars-registerhelper-function-in-node-js-project>

- <https://stackoverflow.com/questions/8853396/logical-operator-in-a-handlebars-js-if-conditional>

-