**Full Stack Assignment**

This is an opportunity to showcase what you can do! This exercise can be completed in 4-8 hours; however, you can allocate more time if needed. We would like you to show us how you approach this problem and what your solution is! Aspects of this assignment are left intentionally vague so you are able to have fun and create something you want to show off!

**Problem:**

You are interested in looking at the climate in Winnipeg. Using a continuously updating dataset from weatherstats.ca (<https://winnipeg.weatherstats.ca/download.html>) that you found you would like to use it to see some historical data.

Download Settings:

Graphical user interface, text, application, email

Description automatically generated

**Back-end (Recommended: 2-4 Hours):**

Using a framework you are familiar with (Django, NodeJs, Springboot, etc…), create a server that can do the following operations concurrently:

* Upload a bulk data set.
* Query for data by Date & Time (e.g., Querying for May 1st would return multiple weekly forecasts for that day at any given time, however querying for a specific time interval will only return the one weekly forecast)
* Add a new record

**Front-end (Recommended: 2-4 Hours):**

Using React (or other front-end framework you are familiar with), create a web application that queries your back-end server by date and nicely display the data that returns from the server call. i.e., If you query for May 1, 2021, at 05:00 on your app, it will display the weekly forecast at that time, if you query without a specific time, it will display the weekly forecasts throughout the day so that one can compare.

**Requirements:**

* Your solution code should be available via GitHub / GitLab / BitBucket.
* You can put both solutions in one repo or separate them.
* Solutions should be able to run locally. You are encouraged to leverage tech such as Docker to help simplify deployment, but it is not necessary.
* The solution does not need to be perfect. You should focus on what you consider good, maintainable code, and prioritize your project accordingly.
* Write up some documentation explaining your thought process, design decisions, and possible future iterations.