# CS-1520 Spring 2023
# MidTerm Project

## 1 Introduction

In this midterm project, you will create a program that:

- Allows the user to select a text file using a File browser.
- Reads the data from the selected text file.
- Allows the user to select what dataset to be displayed (temperature, wind, precipitation, stock market).
- Display the dataset using Plotly 3rd party tool.
- Display statistics about that dataset: Average, Min, and Max values.
- Save and load this data to and from the localstorage.

By creating this program, you will be implementing basically all the features that we have been discussing in this course, such as:

- HTML
- JavaScript
- CSS
- The DOM
- Getting DOM references using getElementByID
- Event listeners
- JavaScript Functions
- JavaScript Variables
- Variable Scope
- Web Storage

## 2 Project Description

As you run the project, the first screen to be presented is shown in Figure 1.

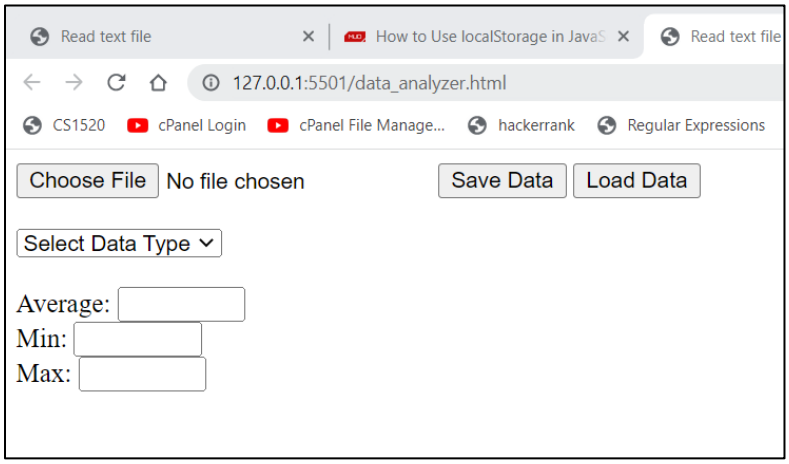


**Figure 1: Initial Screen**

The data to be loaded can be found in Appendix A. Create a text file and place the contents of Appendix A inside your file.

## 2.1 The choose file functionality

By clicking on the choose file button, a file browser should be displayed as shown in the Figure 2 below. This functionality is obtained by simply creating an <input> HTML tag with its type setup to “file” (instead of our customarily “text”).

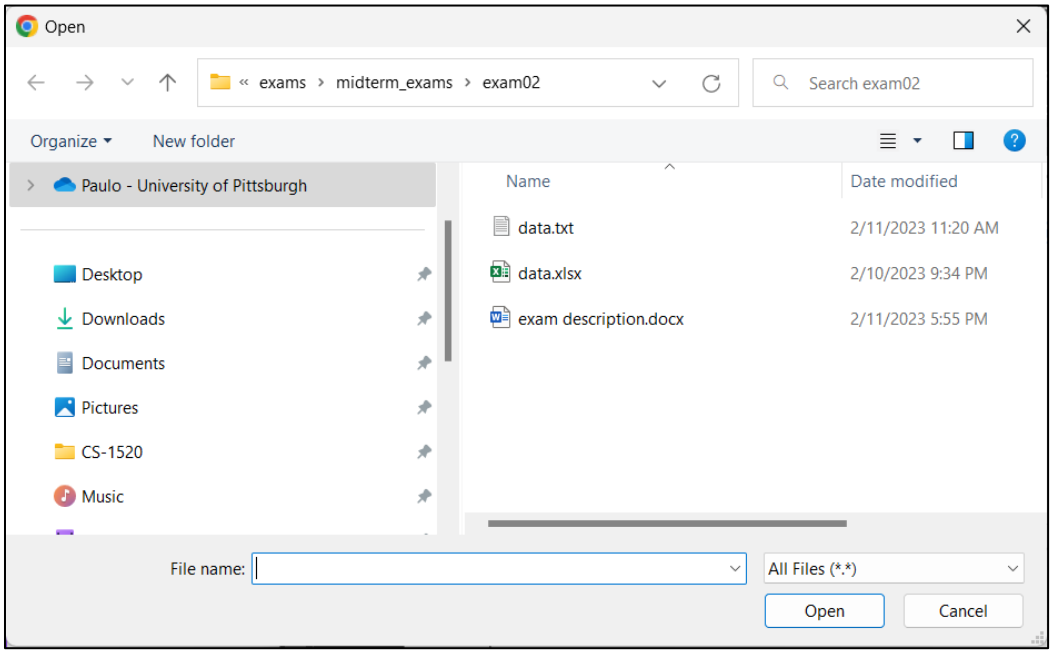


**Figure 2: The File Browser**

After selecting the file, your program should load the data from the text file.

A good source of information on how to read a text file in JavaScript is giving by this youtube video:

https://www.youtube.com/watch?v=V_hgGUBVEDA

You will need to adapt it to your needs, of course.

## 2.2 The select data type

The drop-down select data type is used to allow the user to select what data type they want to display the analysis.

The drop-down options are show below:

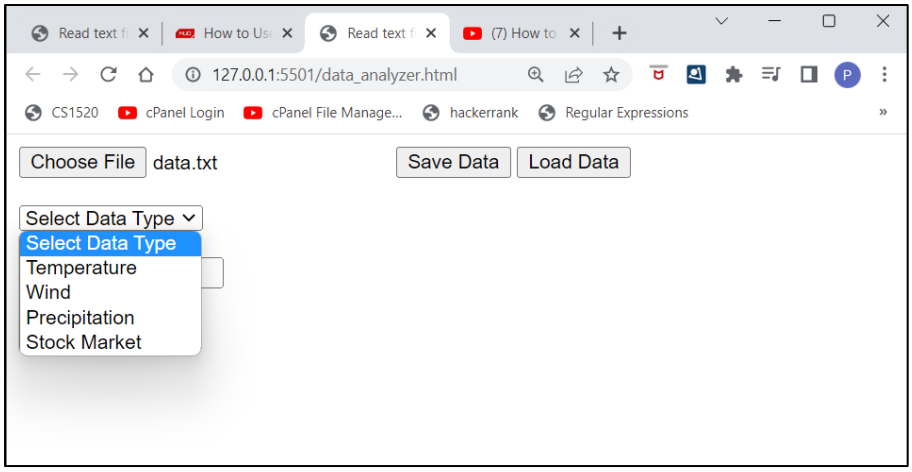


**Figure 3: Selection Options**

To listen to changes in the selection from the drop-down menu, add a "change" event listener to the select HTML tag. The associated function to be run when this event occurs should read the file lines and run the statistics on the data (see video referenced in Section 2.1. The data analysis consists of computing the average, min, and max values, and plotting the data.

An example of this behavior is shown in Figure 4 below.

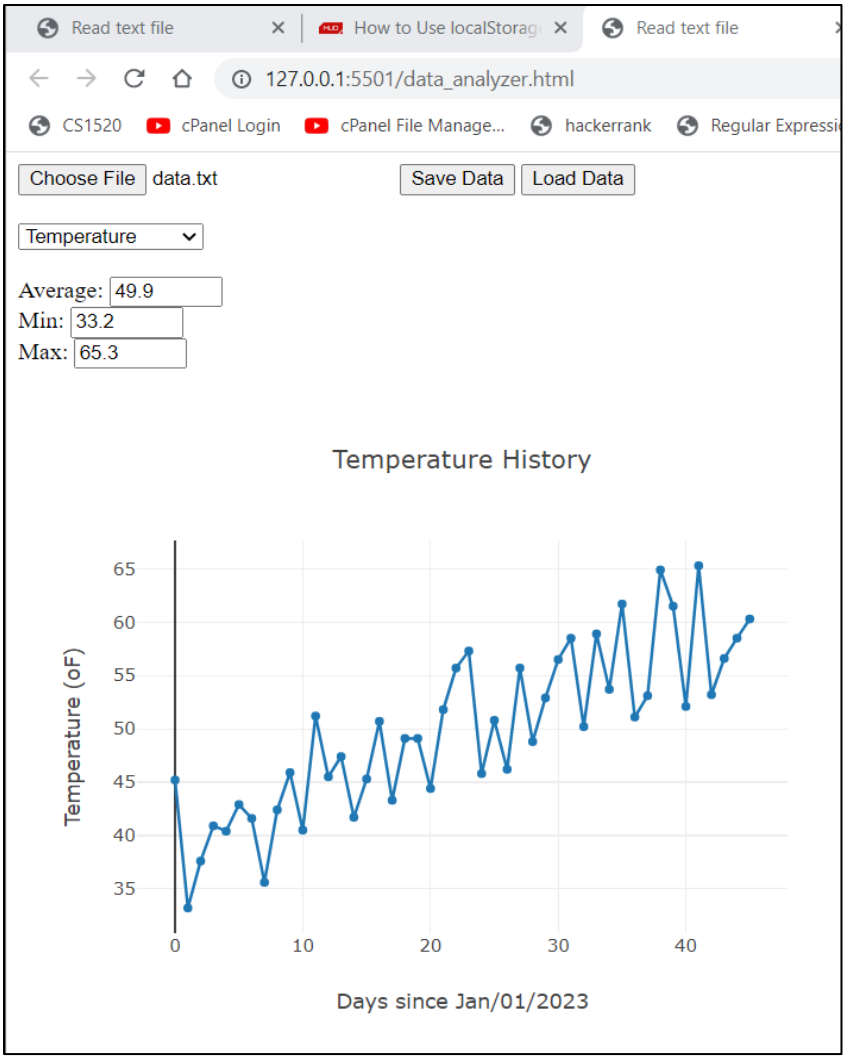


**Figure 4: Data analysis for the temperature data**

Pay attention to the title of the x-axis, y-axis, and the plot itself. They change based on the data type being displayed. A good source of information about the plot tool is in the W3Schools Plotly. Adapt the given example to your needs.

## 2.3 The Save Data Functionality

After loading the data from the text file for the first time, your application can save the data in the local web storage. Use the command:

```
localStorage.setItem(key, value);
```

where the key parameter can be any name, such as “dataset” and the value is a String. Sine the “result” attribute of the file File reader contains a long single string with all the lines, you can use this attribute as the “value” of the above command.

```
let value = fr.result;
```

## 2.4 The Load Data Functionality

If you have already saved the data from the text file into the web local storage, the load data button can be used instead of the “choose file” button.

Clicking on the load saved data button and the associated function shall load the data using the following command:

```
let data = localStorage.getItem('data');
```

Remember that this data variable has now all the entries in a long string. You will need first to split it by the “\n” and later each line shall be split by the “\t”. This is the same process as the one that you should use when the user selects the “choose file” button. Therefore, as a suggestion to avoid code duplication, you could create a function that does this task and you call it from both places. (But it is up to you).

# Appendix A
# The Data

The first column is temperature (oF), the second one is wind (mph), the third one is precipitation (inches), and the last column is stock market (points). Note: Each line is separated by “\n” and each item in a given line is separated by tabs (“\t”) instead of spaces

| | | | |
|---|---|---|---|
| 45.2 | 0.3 | 2.4 | 66.1 |
| 33.2 | 9.7 | 3.2 | 67.8 |
| 37.6 | 0.4 | 2.8 | 65.5 |
| 40.9 | 0.5 | 0.7 | 73.5 |
| 40.4 | 4.8 | 1.9 | 69.1 |
| 42.9 | 1.6 | 1.9 | 68.1 |
| 41.6 | 1.3 | 1.2 | 74.4 |
| 35.6 | 1.2 | 2.5 | 78.5 |
| 42.4 | 2.7 | 4.8 | 82.9 |
| 45.9 | 5.2 | 3.6 | 75.7 |
| 40.5 | 4.2 | 2.5 | 71.6 |
| 51.2 | 6.4 | 4.2 | 76.0 |
| 45.5 | 4.0 | 3.4 | 78.9 |
| 47.4 | 5.3 | 1.7 | 72.9 |
| 41.7 | 5.6 | 3.1 | 72.7 |
| 45.3 | 3.2 | 1.4 | 84.2 |
| 50.7 | 7.6 | 1.4 | 68.3 |
| 43.3 | 0.8 | 1.1 | 86.2 |
| 49.1 | 10.0 | 2.5 | 85.1 |
| 49.1 | 7.9 | 1.2 | 82.3 |
| 44.4 | 2.4 | 3.8 | 84.6 |
| 51.8 | 6.7 | 0.7 | 75.8 |
| 55.7 | 3.7 | 1.1 | 76.0 |
| 57.3 | 7.5 | 0.1 | 74.5 |
| 45.8 | 6.7 | 3.4 | 80.6 |
| 50.8 | 6.5 | 4.0 | 70.5 |
| 46.2 | 9.0 | 0.8 | 84.8 |
| 55.7 | 1.0 | 4.9 | 81.4 |
| 48.8 | 5.5 | 4.8 | 83.4 |
| 52.9 | 3.7 | 1.8 | 75.9 |
| 56.5 | 0.3 | 1.2 | 90.7 |
| 58.5 | 5.0 | 0.5 | 86.3 |
| 50.2 | 6.7 | 3.8 | 87.3 |
| 58.9 | 6.1 | 0.3 | 78.6 |
| 53.7 | 3.7 | 2.4 | 82.7 |
| 61.7 | 6.0 | 4.9 | 77.8 |
| 51.1 | 9.1 | 0.3 | 80.2 |
| 53.1 | 6.8 | 2.8 | 92.6 |
| 64.9 | 1.5 | 2.5 | 94.7 |
| 61.5 | 1.0 | 3.9 | 82.1 |
| 52.1 | 9.0 | 3.3 | 84.7 |
| 65.3 | 1.3 | 0.3 | 91.4 |
| 53.2 | 2.5 | 2.2 | 92.9 |
| 56.6 | 1.0 | 0.3 | 93.7 |
| 58.5 | 1.6 | 3.7 | 88.5 |
| 60.3 | 5.6 | 1.5 | 95.6 |