Lab 4: Extreme Weather Records by State

How it works

Note: Due to the client-side script in this application, live demo is not available.

For this lab, you need to create a Web page that displays extreme weather records by state. The extreme weather records of all states are stored in a JSON file named *extremes.json*. The Web page filters data by state. When a user selects a state from the selection list, the Web page displays all records of the selected state.

Extreme Weather Records by State

Select a State: Alabama

State	Element	Notes	Value	Date	Location	Station
Alabama	All-Time Maximum Temperature	N/A	112°F	September 6, 1925	CENTREVILLE	011520
Alabama	All-Time Minimum Temperature	N/A	-27°F	January 30, 1966	NEW MARKET 2	015867
Alabama	All-Time Greatest 24-Hour Precipitation	Although higher amounts have been observed in other states, the 32.52 inches of rainfall that was observed in 24-hrs over July 19-20, 1997 at Dauphin Island Sea Lab in conjunction with Hurricane Danny is the greatest 24-hr rainfall that has been directly observed at an officially established observation station within the coterminous United States.	32.52"	July 19 - 20, 1997	DAUPHIN ISLAND #2	012172
Alabama	All-Time Maximum 24-Hour Snowfall	N/A	20"	March 13, 1993	WALNUT GROVE	018648
Alabama	All-Time Maximum Snow Depth	N/A	22"	January 24, 1940	REFORM	016847
Alabama	Hail: Weight	(View report)	9.8ounces	March 19, 2018	Walter	n/a
Alabama	Hail: Circumference	(View report)	13.75"	March 19, 2018	Walter	n/a
Alabama	Hail: Diameter	(View report)	5.38"	March 19, 2018	Walter	n/a
Alabama	Hail: Volume	(View report)	19.8cubic inches	March 19, 2018	Walter	n/a

This Web application is very similar to the cd catalog application you finished in class. The new feature in this lab is filtering by state, which can be achieved with conditional statements. Refer to the cd catalog application when you are working on this lab.

Step-by-step instructions

1. Download data files from Canvas and extract the **Lab04** folder into **htdocs/I211** folder. Inside the folder, there should be four files:
    - *styles.css*: the external css style sheet.
    - *extremes.json*: this file stores the extreme weather records of all states.
    - *loadJSONDoc.js*: This JavaScript file contains a function named **loadJSON**, which loads a local JSON file and returns the string representation of JSON objects defined in the document.
    - *index.html*: The file currently contains a selection list for all 50 states and a table for displaying the extreme weather records.
2. Open all the files in NetBeans. Carefully examine the code. The only file that needs to be modified is the *index.html* file. Preview the file in a browser. You should see a header and a selection list.
3. For the *index.html*, complete the anonymous function registered with window's **onload** event. The function should do two things:
    - Load the local JSON file named *extremes.json*.
    - Register the **onchange** event handler with the selection list so that when a new option is selected, the **display** method gets called.
4. Complete the **display** function in *index.html*: This function accepts two parameters: a state and the string representation of the JSON document that contains extreme weather records of all 50 states. It filters the data stored in the JSON document and retrieves only the records of the state that matches the first argument. For each extreme weather record, the function appends one row to the table to display all extreme weather records of the state. Please note, you need to convert "degrees F" to "°F" and "inches" to "″".

Code style guidelines:

1. Code style is as important as the code itself. Code style includes enough comments, adequate space between code blocks, and proper indentation, etc. Read details and view sample code from http://pear.php.net/manual/en/standards.php.
2. Please provide sufficient comments in your source code. Source code is a language for people, not just for computers. Comment as you go and don't wait for later. Ask yourself: "How will the next person know that?" Commenting code shows your professionalism, but also helps your grader understand your code.
3. Every class file should contain a header in this format:

```
/*
* Author: your name
* Date: today's date
* Name: file name
* Description: short paragraph that explains what the class is for
*/
```

4. Indent your code. Leave enough space between code blocks. You can always use Alt+Shit+F in NetBeans to automatically format your code.

Turning in your lab

Your work will be evaluated on completeness and correctness. Thoroughly test your code before you turn it in. It is your responsibility to ensure you turn in the correct files. You will NOT receive any credit if you turn in the wrong files whether or not you've completed the lab.

1. Zip the entire **Lab04** folder and save it as *Lab04.zip*.
2. Upload the *Lab04.zip* file in Canvas before the lab's deadline.

Grading rubric

Your TAs will assess your lab according to the following grading rubric. You should very closely follow the instructions in this handout when working on the lab. Small deviations may be fine, but you should avoid large deviations. You will not receive credits if your deviation does not satisfy an item of the grading rubric. Whether a deviation is small or large and whether it satisfies the requirement are at your TAs' discretion. Here is the breakdown of the scoring:

Modifying index.html (15 points)

Activities	**Points**
Load JSON objects from *extremes.json*	1
Complete the **display** function	12
Handle the **change** event of the selection list	2

Programming style (5 points)

Activities	**Points**
Comment your code: comments must be specific to your code	3
Use white spaces to separate code sections	1
Indent and line up code	1