**CSE 5335-002 Web Data Management**

**Project 1**

**Design Document**

Seeram Likitha

1001363714

[likitha.seeram@mavs.uta.edu](mailto:likitha.seeram@mavs.uta.edu)

Table of Contents

[1. Project Overview 3](#_Toc464667289)

[2. Requirement Analysis 3](#_Toc464667290)

[2.1 Software Requirements 3](#_Toc464667291)

[2.2 Hardware Requirements 3](#_Toc464667292)

[3. Assumptions 4](#_Toc464667293)

[3.1 Input Value 4](#_Toc464667294)

[3.2 Parameters 4](#_Toc464667295)

[4. Implementation 5](#_Toc464667296)

[4.1 HTML File 5](#_Toc464667297)

[4.2 CSS File 5](#_Toc464667298)

[4.3 Javascript 5](#_Toc464667299)

[4.3.1 Time Parameters 5](#_Toc464667300)

[4.3.2 Temperature Parameters 6](#_Toc464667301)

[4.3.3 Visibility Parameter 6](#_Toc464667302)

[4.3.4 Rain and Snow Parameters 6](#_Toc464667303)

[4.3.5 Suggestions 7](#_Toc464667304)

[4.4 Running the application 7](#_Toc464667305)

[5. Conclusion 8](#_Toc464667306)

# Project Overview

The project is about extracting weather report of a city by accessing Rest API of a weather application “Open Weather Map”. “Open Weather Map” is a weather application that provides weather reports of all cities across the globe at the current point of time. This application allows users to sign up to get an access key to the API. This key is used in the Project to gather the weather information and display it accordingly.

We also need to suggest the users to carry an umbrella or a coat basing on different weather conditions.

# Requirement Analysis

We need to have an API access key to call the API which in turn returns a response object in the form of JSON. The JSON object has all the weather details of the city, using which we can access all the required parameters.

## 2.1 Software Requirements

Following software are required for the project:

* XAMP Control Panel (for windows users)
* Sublime Text Editor
* A Web browser

## 2.2 Hardware Requirements

A minimum of following hardware are necessary for this project:

* An Operating System to support the above software
* 2GB RAM

# Assumptions

A few assumptions are made while working on this project. They are listed below.

## 3.1 Input Value

It is assumed that the users provide the correct input by typing the name of the city correctly.

As OpenWeatherMap has a **flexible search engine**, it will return the weather report of the city by making an approximation match of the typed in value with the actual name of the city. That is, even though the city name is not typed properly, it searches for a city that could match with the entered value and gives the result of that city.

For example, if the user types the city name incorrectly or partially as ‘lon, UK’ it gives the results of the city ‘london’.

## 3.2 Parameters

From the JSON response, we will be displaying only the required parameters of weather. They are:

* City Name
* Longitude
* Latitude
* Sunrise
* Sunset
* Temperature
* Minimum Temperature
* Maximum Temperature
* Pressure
* Humidity
* Clouds
* Visibility

**Additional parameters** like ‘Weather description’ and ‘Weather Icon’ are also implemented in this project.

# Implementation

This section includes the implementation details of HTML, javascript and CSS files.

## 4.1 HTML File

The HTML page is based on **HTML5** technology which means that the structure of the webpage developed is semantically maintained to include the structure of the page in HTML, all the styling in CSS file and the ajax code in javascript.

It includes 3 sections.

* Section 1: It includes heading and a form to type in the city name.
* Section 2: This is the output section. Javascript code handles this section by displaying the weather conditions of the city in a tabular format.
* Section 3: This section the description of the city and a suggestion to users based on the weather condition in that city.

## 4.2 CSS File

A CSS file is linked to the HTML. This file includes different types of styles assigned to the selectors.

For the heading, a background image has been added which takes its source from a web link. So we need internet connection to display the image.

Header {

background-image: url("http://syque.com/ds/pix/summer\_hols\_07/dolphin\_sky\_panorama\_1\_800.jpg");

}

## 4.3 Javascript

The javascript file contains the ajax code to request the weather API and get the response from it. When the **readyState** is 4 i.e., when the request is successfully processed and the response is ready, we parse the JSON response and extract the parameters that we like to display.

### 4.3.1 Time Parameters

The time parameters like ‘Sunrise’ and ‘Sunset’ are in the UNIX time format in the JSON response. We need to convert this time and display the time in a 24-hour format as shown below.

Conversion –

var mTime = new Date(json.sys.sunrise\*1000);

Display –

<td>"+mTime.getHours()+":"+mTime.getMinutes()+":"+mTime.getSeconds()+" CST</td>

### 4.3.2 Temperature Parameters

We are displaying the 3 parameters of temperature. They are:

* Temperature: Temperature of the city at this moment
* Minimum Temperature: Minimum temperature that can be possible at this moment
* Maximum Temperature: Maximum temperature that can be possible at this moment

The JSON response has the temperatures in ‘Kelvin’ units by default. We need to change this to Fahrenheit by adding “units=imperial” while calling the API as shown below.

xhr.open("GET", "proxy.php?q="+city+"&appid="+api\_key+"&format=json"+"**&units=imperial"**, true);

### 4.3.3 Visibility Parameter

The Visibility parameter is derived from the weather group id starting with ‘7’. (Group 7xx: Atmosphere)

This group includes the weather conditions like mist, smoke, fog, sand, dust, tornado, volcanic ash, etc. during which the visibility is not clear.

### 4.3.4 Rain and Snow Parameters

The Rain and Snow parameters of a place are defined only if that city’s weather has been recorded with rain or snow conditions in the last 3 hours. So, JSON response for a city may or may not have these parameters. Hence the code including these parameters is maintained in **try catch blocks** to catch the error in cases of **‘undefined’** values of these parameters.

try { var rains = JSON.stringify(json["rain"]["3h"]);

var snows = JSON.stringify(json["snow"]["3h"]);}

catch(err) {

//Error Caught

}

### 4.3.5 Suggestions

Users are given suggestions based on the weather conditions in the city. Suggestions are made by considering the rain and temperature parameters.

If the rain parameter of the city has a value greater than 0, then we display - **Hey, It's raining. Take an umbrella when you go outside.**

If and above condition is not satisfied, we then go for the next condition to check if the Temperature of the city is less than 68F. Then we display - **Hey, It's chilling outside. Make sure to carry a jacket when you step out.**

If the above conditions are not satisfied, then we display - **Normal weather outside with no rain.**

These suggestions are included to the HTML file by using the innerHTML property as shown below:

document.getElementById("suggestion1").innerHTML = "<p> Suggestion here </p>";

## 4.4 Running the application

Make sure to include your project folder in the hotdocs folder in xampp. Apache and MySQL modules should be running in XAMPP control panel.

You need to hit the link **localhost/project1/weather.html** to run the application in a browser.

# Conclusion

Project1 has helped me learn concepts of web designing like:

* HTML5
* Advanced CSS
* AJAX calls in javascript

Clubbing all these concepts in a single application gave me a better picture to understand how these components are connected and work in a web application.