Task 1 Code Explanation

/\*The function is used to dynamically load a json file containing weather data

across cities in the Uk

window.onload() function is used here to load contents of the data to the webpage

\*/

window.onload = function() {

    const cityWeatherFile = './weather.json'; // declare a constant variable  to hold the json file

    /\*The fecth method is used here to access the resources in the json file.

    The fetch method handles request and response to the web browser.

    First it makes a request to the server, then it returns a promise that resolves into a response object.

    The then function handles the data while the catch functions handles any error

    \*/

    //the fetch method takes in a compulsory argument which is the weather.json file in this case

    fetch(cityWeatherFile)

        .then((response) => {

            //the response is accessed here and it is important to ensure that the response status is 200 using an if statement

            if (response.status !== 200) {

                console.log("Error occured: " + response.status); // log error to console if response status is not 200

            } else {

                return response.json(); // return the json object of the output if response status is satisfied

            }

        })

        .then((data) => {

            /\*The text in the reponse is being examined here

            The getWeatherInfo() method is called here and takes in data as an argument

            The data displayed consists of the defined attributes in the getWeatherInfo() method and it is matched with

            attributes in the weather.json file.

            \*/

            getWeatherInfo(data); // function to display weather details of cities

            /\*\*

             \* The setTimeout function is used here to set interval at which the data in the cityWeatherFile will be uploaded

             \* The data will update every 5000 milliseconds which is equivalent to 5seconds

             \*/

            setTimeout(function() {

                getWeatherInfo(data); // the getWeatherInfo() method is called here again to keep displaying every 5 seconds

                window.onload(); // window.onload() method is being called here again so that whenever the page refreshes, the content are loaded immediately

                console.log(data); //display the data on the console

            }, 5000);

        })

        .catch((error) => {

            var div = document.getElementById('update-status-message'); // declare variable to get the html element where status message will be displayed

            div.append('Error Loading Data!!!'); //use append method to join the error message to the variable holding the status message

            console.log('Error loading data: ' + error); //display error message to the console

        });

    /\*This function is used to generate icons depending on the current condition of the city weather

    the function takes in an argument called description that is used to check different  conditions of the weather.

    The icons have been placed in a folder called weather\_icons

    \*/

    const generateIcons = (description) => {

        if (description == 'Snow') {

            //If the current condition is Snow, snow icon is displayed from weather-icons folder

            return "../weather\_icons/snow.png";

        } else if (description == 'Cloud') {

            //If the current condition is Cloud, cloud icon is displayed  from weather-icons folder

            return "../weather\_icons/cloud.png";

        } else if (description == 'Sleet') {

            //If the current condition is Sleet, sleet icon is displayed  from weather-icons folder

            return "../weather\_icons/sleet.png";

        } else if (description == 'Sun') {

            //If the current condition is Sun, sun icon is displayed  from weather-icons folder

            return "../weather\_icons/sun.png";

        } else if (description == 'Hail') {

            //If the current condition is Hail, hail icon is displayed from weather-icons folder

            return "../weather\_icons/hail.png";

        } else if (description == 'Heavy cloud') {

            //If the current condition is Heavy cloud, heavy cloud icon is displayed from weather-icons folder

            return "../weather\_icons/heavycloud.png";

        } else if (description == 'Heavy rain') {

            //If the current condition is Heavy rain, heavy rain icon is displayed from weather-icons folder

            return "../weather\_icons/heavyrain.png";

        } else if (description == 'Mist') {

            //If the current condition is Mist, mist icon is displayed from weather-icons folder

            return "../weather\_icons/mist.png";

        } else if (description == 'Rain') {

            //If the current condition is rain, rain icon is displayed from weather-icons folder

            return "../weather\_icons/rain.png";

        } else if (description == 'Sun and cloud') {

            //If the current condition is Sun and cloud, sun and cloud icon is displayed from weather-icons folder

            return "../weather\_icons/sun\_and\_cloud.png";

        } else if (description == 'Thunderstorm') {

            //If the current condition is Thunderstorm, thunderstorm icon is displayed from weather-icons folder

            return "../weather\_icons/thunderstorm.png";

        }

    };

    /\*This function handles how the weather data is displayed on the webpage

    The getWeatherInfo() method takes in an argument called data

    The data is accessed depending on the attribute of the data needed to be displayed

    Table formart is used to display the data

    \*/

    const getWeatherInfo = (data) => {

        let table = document.getElementById('table-display'); // declare a variable that stores the html content to be displayed

        table.innerHTML = ''; // set the innerHTML content of the table to an empty string to ensure no item is currently displayed on the table

        let row = document.createElement("tr"); // create a table row element and store in a variable called row

        //In the innerHTML of the table row created, create table headers and define their contents using row.innerHTML

        row.innerHTML = "<th>City Id</th>" +

            "<th>City Name</th>" +

            "<th>Current Condition</th>" +

            "<th>Cloud Icon</th>" +

            "<th>Temperature</th>" +

            "<th>Wind Speed</th>" +

            "<th>Wind Direction</th>" +

            "<th>Wind Chill Factor</th>";

        table.append(row); //use the append method to add the defined table head content in the table row to the table

        /\*\*

         \* This for-loop loops through the array of the data and adds the content to a table column also called table data

         \* i is a counter in the for loop that loops through array of element in the data and increments until the last element in the data is reached

         \*/

        for (let i = 0; i < data.cities.length; i++) {

            let row = document.createElement('tr'); //create a table row element and store it in a variable called row

            row.setAttribute('class', "active-row"); // this is used to set a class attribute called active-row to the row element to check for the rows that are active

            /\*The elements in the data which includes the id, name, current condition, weather icon, temperature, windpseed, wind direction, windchill factor,

             of each cities in the data. They are added to the innerHTML of the row element in the table

             The generateIcons() method is called in the weather icon column and adds the appropriate icon to each row.

             It takes in an argument which is the the current condition and adds the icon that matches the current condition

            \*/

            row.innerHTML =

                '<td>' +

                data.cities[i].id +

                '</td>' +

                '<td>' +

                data.cities[i].name +

                '</td>' +

                '<td>' +

                data.cities[i].currentCondition +

                '</td>' +

                `<td><img src = ${generateIcons(data.cities[i].currentConditio n)}  alt = ""  width= 35px height = 35px></td>` +

                '<td>' +

                data.cities[i].temperature + "°C" +

                '</td>' +

                '<td>' +

                data.cities[i].windSpeed + "mph" +

                '</td>' +

                '<td>' +

                data.cities[i].windDirection + "°" +

                '</td>' +

                '<td>' +

                data.cities[i].windChillFactor + "°" +

                '</td>';

            table.appendChild(row); // The data in the row element containig the cities weather details are added to the html table using the append method

        }

    };

};