

# COMP3322 Modern Technologies on World Wide Web

## Assignment Two

Total 18 points

Deadline: November 2, 2023 23:59

### Overview

You are going to design and develop a Web app that displays (1) the current weather condition of Hong Kong, (2) temperature, rainfall, and air quality data of your current location, (3) temperature, rainfall, air quality data of various districts/locations in Hong Kong, and (4) 9-day weather forecast of Hong Kong. Your program retrieves data from various Open Data sources via the corresponding APIs. The Web app should be nicely rendered on mobile platforms and desktop computers.

### Objectives

1. A learning activity to support ILO 1 and ILO 2.
2. To learn how to make use of Open Data.
3. To practice using JavaScript to (1) build a Web page **from scratch**, (2) carry out AJAX communication for retrieving Open Data, and (3) create dynamic contents.
4. To practice using CSS styling to design responsive Web application.

### Open Data

We are going to make use of the following Open Data datasets for building our Web App.

- ✕ HKO Current Weather Report - provides current weather report of Hong Kong and is **updated hourly**.  
Website: <https://data.gov.hk/en-data/dataset/hk-hko-rss-current-weather-report>  
API: <https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=rhrread&lang=en>  
Documentation: [https://data.weather.gov.hk/weatherAPI/doc/HKO\\_Open\\_Data\\_API\\_Documentation.pdf](https://data.weather.gov.hk/weatherAPI/doc/HKO_Open_Data_API_Documentation.pdf)

This dataset consists of:

- rainfall data of various districts
- current weather icon
- UV index
- update time
- warning message
- temperature data of various locations
- humidity
- others: lighting data, rainstorm reminder, special weather tips, message of tropical cyclone position, etc.

```

▶ rainfall:          {...}
▶ icon:             [...]
  iconUpdateTime:    "2023-10-06T06:45:00+08:00"
▶ uvindex:          {...}
  updateTime:        "2023-10-06T13:02:00+08:00"
▶ temperature:      {...}
▶ warningMessage:    [...]
  mintempFrom00To09: ""
  rainfallFrom00To12: ""
  rainfallLastMonth: ""
  rainfallJanuaryToLastMonth: ""
▶ tcmessage:         [...]
▶ humidity:          {...}

```

- ✖ HKO 9-day Weather Forecast - provides a 9-day weather forecast of Hong Kong and is **updated twice daily**.

Website: <https://data.gov.hk/en-data/dataset/hk-hko-rss-9-day-weather-forecast>

API: <https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=fnd&lang=en>

Documentation:

[https://data.weather.gov.hk/weatherAPI/doc/HKO\\_Open\\_Data\\_API\\_Documentation.pdf](https://data.weather.gov.hk/weatherAPI/doc/HKO_Open_Data_API_Documentation.pdf)

This dataset consists of:

- weather forecast data
- others: general situation, sea temperature, and soil temperature.

```

▶ generalSituation:  "Tropical Cyclone Koinu w...of the South China Sea."
▼ weatherForecast:
  ▶ 0:               {...}
  ▶ 1:               {...}
  ▶ 2:               {...}
  ▶ 3:               {...}
  ▶ 4:               {...}
  ▶ 5:               {...}
  ▶ 6:               {...}
  ▶ 7:               {...}
  ▶ 8:               {...}
  updateTime:        "2023-10-06T11:30:00+08:00"
▶ seaTemp:           {...}
▶ soilTemp:          [...]

```

- ✖ Weather Station Information - provides basic information of HKO weather stations and is **static**.  
API: <https://ogciopsi.blob.core.windows.net/dataset/weather-station/weather-station-info.json>  
Documentation: <https://ogciopsi.blob.core.windows.net/dataset/weather-station/weather-station-citydashboard-dataspec.pdf>

This dataset consists of an array of HKO weather stations; each station consists of :

- station name in English

- station code
- latitude and longitude values of that station
- others

▼ 0:	
station_code:	"CLK"
station_name_en:	"Chek Lap Kok"
station_name_tc:	"赤鱓角"
station_name_sc:	"赤鱓角"
date_of_first_operation:	"1997-06-01"
latitude:	22.30944444
longitude:	113.9219444
above_mean_sea_level:	6
▶ 1:	{-}
▶ 2:	{-}
▶ 3:	{-}
▶ 4:	{-}

- ✧ OGCIO Air Quality Health Index of Individual Air Quality Monitoring Stations - provides current Air Quality Health Index (AQHI) of individual Air Quality Monitoring stations provided by Environmental Protection Department. The dataset is **updated hourly**.

Website:

API: <https://dashboard.data.gov.hk/api/aqhi-individual?format=json>

This dataset consists of an array of air quality data of different monitoring stations; each station consists of:

- station's name
- AQHI
- health risk level
- publish date

▼ 0:	
station:	"Central/Western"
aqhi:	5
health_risk:	"Moderate"
publish_date:	"2023-10-06T13:30:00"
▶ 1:	{...}
▶ 2:	{...}
▶ 3:	{...}
▶ 4:	{...}
▶ 5:	{...}
▶ 6:	{...}
▶ 7:	{...}
▶ 8:	{...}
▶ 9:	{...}

- ✧ Air Quality Monitoring Station Information - provides the latitude and longitude coordinates of all general stations and roadside stations.

The dataset is prepared by Dr. Anthony Tam using the information provided by the Environmental Protection Department.

## API: aqhi-station-info.json

This dataset consists of an array of air quality monitoring stations; each station consists of:

- station's location
- latitude and longitude values of that station

```
▼ 0:
  station: "Central/Western"
  lat: 22.28489089
  lng: 114.14442071
▶ 1: {}
▶ 2: {}
▶ 3: {}
▶ 4: {}
▶ 5: {}
▶ 6: {}
▶ 7: {}
▶ 8: {}
```

- ✱ The Reverse Geocoding API is provided by the Nominatim search engine for OpenStreetMap. Website: <https://nominatim.org/release-docs/develop/api/Reverse/>  
API: <https://nominatim.openstreetmap.org/reverse?format=json&lat=<value>&lon=<value>&zoom=18&addressdetails=1>

For example, to get the address information of HKU Main Building (lat: 22.28408, lon: 114.13790), we use this API query:

<https://nominatim.openstreetmap.org/reverse?format=json&lat=22.28408&lon=114.13790&zoom=18&addressdetails=1>

We obtain the following data:

```
place_id: 236400419
▼ licence: "Data © OpenStreetMap contributors, ODbL 1.0. http://osm.org/copyright"
osm_type: "relation"
osm_id: 4246029
lat: "22.28269745"
lon: "114.13845499995571"
class: "amenity"
type: "university"
place_rank: 30
importance: 0.5296783459293937
addresstype: "amenity"
name: "The University of Hong Kong"
▶ display_name: "The University of Hong K...sland, Hong Kong, China"
▼ address:
  amenity: "The University of Hong Kong"
  road: "Pok Fu Lam Road"
  borough: "Sai Wan"
  city_district: "Central and Western District"
  city: "Hong Kong Island"
  state: "Hong Kong"
  ISO3166-2-lvl3: "CN-HK"
  country: "China"
  country_code: "cn"
```

## Requirements

- Retrieve all Open Data datasets using AJAX (XHR or fetch()).
- Extract required data from the Current Weather Report and 9-day Weather Forecast report

Current weather	Current Weather Report API (WR)
Current weather icon	WR.icon[0] //show the first one
Current temperature	WR.temperature.data[1].value //Hong Kong Observatory
Current humidity	WR.humidity.data[0].value //show the first one
Current rainfall	WR.rainfall.data[13].max //Yau Tsim Mong
Current UV index	WR.uvindex.data[0].value //could be missing //show the first one
Current warning	WR.warningMessage[ ] //could be missing //show all warning messages
Last update	WR.updateTime
District Temperatures	WR.temperature.data[0..N] //N could be changing from time to time
Temp. of each district	WR.temperature.data[i].place, WR.temperature.data[i].value
Weather forecast	9-day Weather Forecast API (WF)
9-Day Forecast	WF.weatherForecast[0..8]
Forecast icon	WF.weatherForecast[i].ForecastIcon
Forecast date	WF.weatherForecast[i].forecastDate
Forecast week	WF.weatherForecast[i].week
Forecast temperatures	WF.weatherForecast[i].forecastMintemp, WF.weatherForecast[i].forecastMaxtemp
Forecast humidity	WF.weatherForecast[i].forecastminrh, WF.weatherForecast[i].forecastmaxrh
PSR (Probability of Significant Rain)	WF.weatherForecast[i].PSR

Weather Icons list - [https://www.hko.gov.hk/textonly/v2/explain/wxicon\\_e.htm](https://www.hko.gov.hk/textonly/v2/explain/wxicon_e.htm)

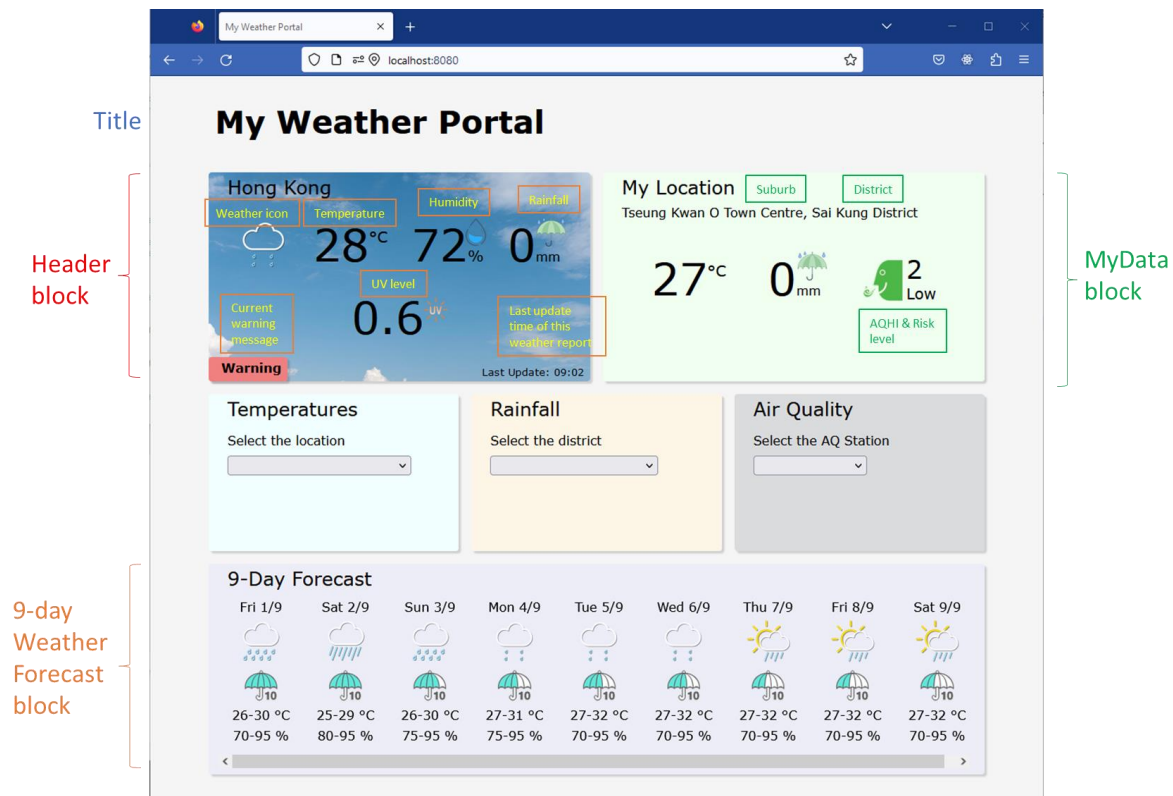
You can get any weather icon via the following path, e.g. icon no. 63

<https://www.hko.gov.hk/images/HKOWxlconOutline/pic63.png>

PSR icons – We can borrow the icons from a HKO website:

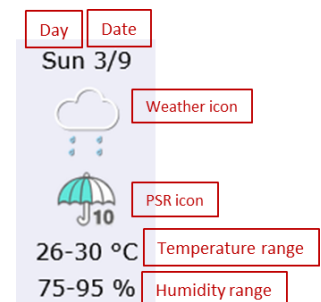
<https://www.hko.gov.hk/en/wxinfo/currwx/fnd.htm?tablenote=true>

- Create a 'data' folder under your site and place the two static JSON files – aghi-station-info.json and weather-station-info.json in this folder. In addition, place all image files in the 'images' folder.
- Our Weather Web App consists of a title and six blocks of information. Here is the screenshot of a **sample** implementation for the desktop computer (with window's width > 1000px).



The app **should have**

1. A Header block that shows current HK weather condition - the information includes a weather icon, the temperature value at HKO, the humidity value at HKO, the rainfall value of Yau Tsim Mong district, the UVindex level at King's Park (if present), warning message(s) (if present), and last update time of the **HKO Current Weather report**. If present of the warning message(s), display a "button" on the bottom left corner, which allows the user to toggle between showing or hiding the warning message(s).
2. A MyData block that shows the temperature, rainfall, and air quality information at your current location - the data includes your current district and suburb, the temperature of a nearby weather station, the rainfall value of your district, and the AQHI & health risk level of a nearby air quality monitoring station.
3. Three blocks that allow the user to select a target location for getting the temperature value, rainfall value, and the AQHI value of a location/district/station defined in the **HKO Current Weather report** and the **AQHI of individual air quality monitoring stations**. To make it easier for the user to select the target location, **order** the dropdown list (or whatever mechanism) in alphabetical order of the locations.
4. A 9-day weather forecast block that shows all 9 days' forecast data retrieved from the **HKO 9-day Weather Forecast** report. For showing the 9-day forecast, the app should show the predicted information of each forecast date: weather & PSR icons, day and date, temperature range, and humidity range. The data should be arranged in chronological order.



Here are some example screenshots of the Web app running on the mobile device and desktop device.



Figure 1 This view shows the Web app in a mobile setting with UVindex and Warning button.

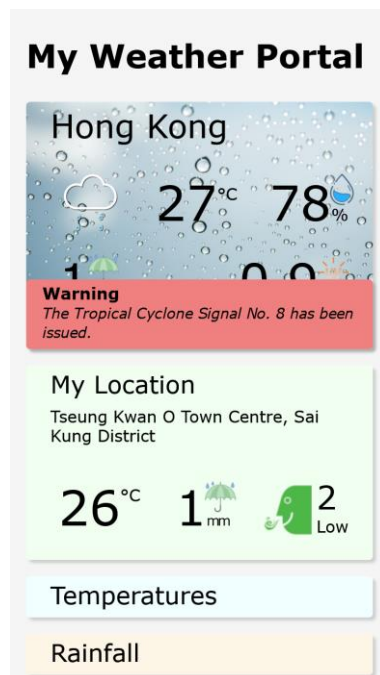


Figure 2 This view shows the Web app in a mobile setting with the warning message.

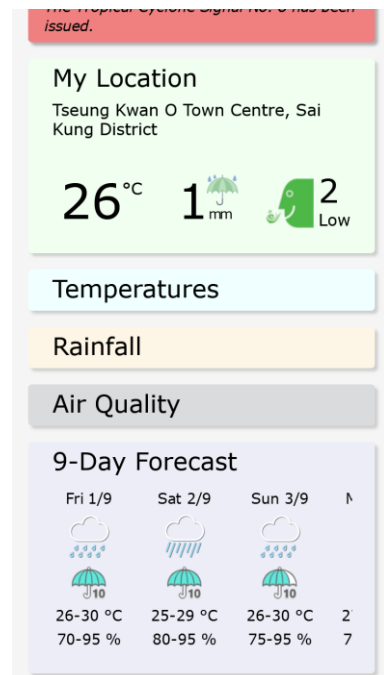


Figure 3 This view shows the bottom half of the Web app in a mobile setting.

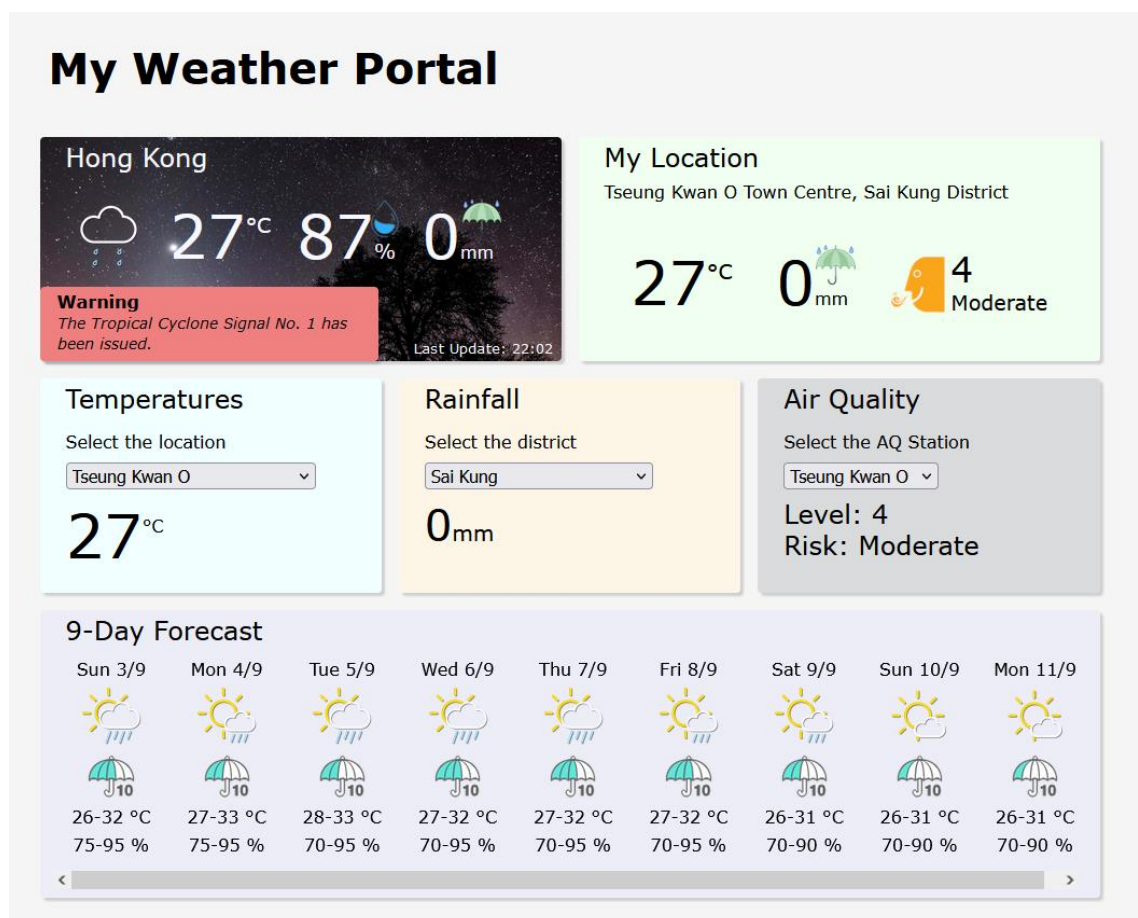


Figure 4 This view shows the Web app in a desktop setting with the warning message.

- Responsive design of the app

Your design should allow the app to adjust the layout, position, spacing, dimension of the contents when displaying the app on a mobile phone or a wide desktop browser. You can use suitable layout for the responsive design.

There is one requirement for the mobile layout. To reduce space occupied by the temperature, rainfall, and air quality blocks, shrink all three blocks and **just show the headings**. Add an on-click event to the blocks to **toggle** them **on or off**. In addition, the app **only shows one block** in full size at a time, i.e., when one block is in full display, the other blocks will be shrunk to display the headings only.

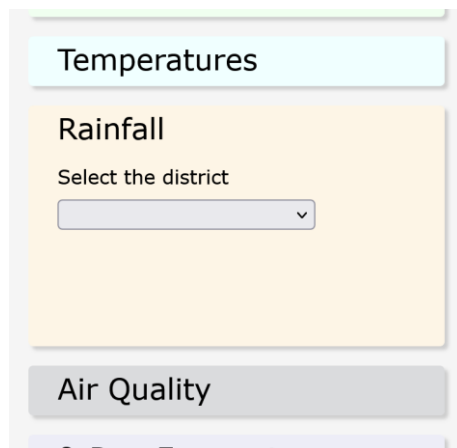


Figure 5 The user can toggle a block to be displayed in full by clicking on the block.

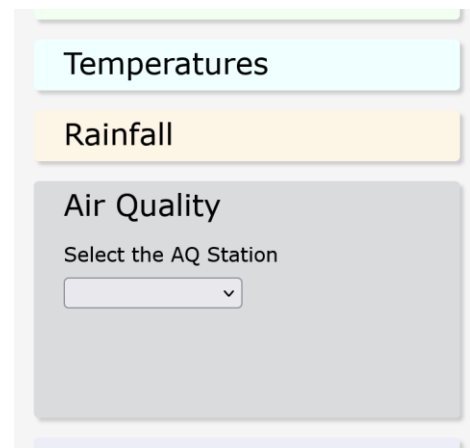


Figure 6 When toggle another block to be displayed in full, shrink other blocks to display the headings only.

- For the MyData block, we need to make use of the following additional resources:

- **HTML5 Geolocation API** to get the geographical position (latitude and longitude coordinates) of the device.
- **Reverse Geocoding API** provided by the Nominatim search engine for OpenStreetMap to obtain the address information (**District and Suburb**) of a geographical position (returned by the device).

Not all returned address information has the suburb field or city\_district field. Use the following logic to obtain the information:

```
if has suburb field
  return suburb
else if has borough field
  return borough as suburb
else if has town field
  return town as suburb
else
  return 'Unknown'
```

```
if has city_district field
  return city_district as district
else
  search for any field that has the
  value contains 'District'
  if found
    return the value of that field
    as district
  else
    return 'Unknown'
```

Here are two examples.



suburb: Happy Valley  
city\_district: Wan Chai District  
city: Hong Kong Island  
state: Hong Kong  
ISO3166-2-lvl3: CN-HK  
country: China  
country\_code: cn

suburb: Tseung Kwan O Town Centre  
town: Tseung Kwan O  
city: Sai Kung District  
municipality: New Territories  
state: Hong Kong  
ISO3166-2-lvl3: CN-HK  
country: China  
country\_code: cn

- Use the returned district information, we retrieve the **rainfall** data of your district from the HKO Current Weather report.
- To get the temperature data and AQHI data of your location, we make use of your current position data to **find the nearby** weather station and air quality monitoring station. We can base on the 'Equirectangular approximation' formula to calculate the approximate distance between your current position with a weather/air quality station. You can find more information at <https://www.movable-type.co.uk/scripts/latlong.html#equirectangular>. And here is the JavaScript excerpted from this Website:

```
const x = (λ2-λ1) * Math.cos((φ1+φ2)/2);  
const y = (φ2-φ1);  
const d = Math.sqrt(x*x + y*y) * R;
```

where  $\phi$  is Latitude,  $\lambda$  is Longitude,  $R$  is earth's radius (mean radius = 6,371km); we must convert the latitude and longitude coordinates to radians before passing them to the cosine function.

For example,

```
φ = latitude * Math.PI/180;  
λ = longitude * Math.PI/180;
```





Based on the **list of places in the temperature dataset**, we look up their latitude\_longitude coordinates via the OGCIOW Weather Station Information API. Use the above Equirectangular approximation formula, we find out which place in the temperature dataset is near to your current location and we set this as the temperature of your current location. We use the same mechanism to locate a nearby air quality monitoring station and use this station location to retrieve the AQHI and Health Risk level from the OGCIOW Air Quality Health Index API.

For example, we set the device location to Heng On Estate, Ma On Shan (Latitude: 22.4176519698°, Longitude: 114.2279348°). Using the Nominatim search engine for OpenStreetMap, it identifies the location as

- address:
  - ISO3166-2-lvl3: "CN-HK"
  - city: "Sha Tin District"
  - country: "China"
  - country\_code: "cn"
  - municipality: "New Territories"
  - residential: "Heng On Estate"
  - road: "Hang Kam Street"
  - state: "Hong Kong"
  - suburb: "Ma On Shan Town Centre"
  - town: "Ma On Shan"

Using the formula, the App reported the temperature data at 'Shatin' and the AQHI data at 'Shatin' monitoring station.

- For the Header block, we would like to use different background images at different timings (e.g., daytime & night-time) and weather conditions (raining and no rain).

<p>During daytime without raining</p> 	<p>During daytime with rain</p> 
<p>During night-time without raining</p> 	<p>During night-time with rain</p> 

- The base document of our Web app is the **index.html** file. You **cannot add** any HTML tags to the `<body>` part of the file. Instead, you use JavaScript to dynamically create all HTML elements and their contents during runtime and use CSS and JavaScript to set the styling and layout. In principle, your task is to **implement the two files** - style.css and main.js. **Note:** you can have different names for the two files, and you can structure your code in multiple .js and .css files. **You are not allowed to use external libraries (e.g., Bootstrap, jQuery) for the task.** If you use different filenames for the two files or adding other CSS style files or JavaScript files, please update the index.html file accordingly.

```

<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <script src="main.js"></script>
  <title>My Weather Portal</title>
</head>
<body>
  <!--
  Use JavaScript to add all HTML elements
  -->
</body>
</html>

```

- You **should implement** appropriate CSS settings for rendering the Web app on a mobile device (with viewport width less than 500px) and a desktop browser.

## Resources

Here are some (optional) resources you can use for building the Web app.

- images.zip - this file contains all images used in the sample implementation.
- data.zip - this file contains sample AQHI data from OGCIO and 9-day forecast and weather data from HKO; this would be useful for you to examine the weather data and test the Web app

without getting the data from HKO. The weather data set contains records with the warning message, raining day, & UV index.

- index.html - this is the base document of your Web app.

### Testing platform

We shall place all your submitted files in the LAMP container set and use Chrome and/or Firefox to test the programs. Please make sure that your Web app is **using real-time data** from HKO and AQHI APIs (rather than running on the provided sample data).

### Submission

Please finish this assignment before **November 2, Thursday 23:59**. Submit the following files:

1. index.html
2. styles.css and other .css files
3. main.js and other .js files

### Grading Policy

Points	Criteria
4.5	Correctly display the header block <ul style="list-style-type: none"><li>▪ Showing weather icon (0.4)</li><li>▪ Showing temperature, humidity, &amp; rainfall data (0.4 each)</li><li>▪ Hiding/showing UV index (0.4)</li><li>▪ Hiding/showing warning message(s) (1)</li><li>▪ Showing last updated time (0.3)</li><li>▪ Changing background images (1.2)</li></ul>
3.0	Correctly display the MyData block <ul style="list-style-type: none"><li>▪ Showing addressing info (District &amp; Suburb) (1.0)</li><li>▪ Showing rainfall data (0.4)</li><li>▪ Showing temperature (0.8)</li><li>▪ Showing AQ info (0.8)</li></ul>
1.0	Correctly display temperature block <ul style="list-style-type: none"><li>▪ Locations are listed in alphabetical order (0.3)</li><li>▪ Provide a suitable selection mechanism (0.3)</li><li>▪ Display the temperature correctly (0.4)</li></ul>
1.0	Correctly display rainfall block
1.0	Correctly display air quality block
2.5	Correctly display 9-day forecast block <ul style="list-style-type: none"><li>▪ Showing the weather icon (0.3)</li><li>▪ Showing the PSR icon (0.5)</li><li>▪ Showing the Date and Day (0.3)</li><li>▪ Showing the range of temp (0.3)</li><li>▪ Showing the humidity range (0.3)</li><li>▪ Able to list all 9-day forecast (0.8)</li></ul>
3.0	Styling and layout <ul style="list-style-type: none"><li>▪ The overall look of the Web app (2.0) – color scheme, position &amp; spacing &amp; dimension of the contents, aesthetic design, etc.</li><li>▪ Responsive design (1.0) – adjustment of the layout and position &amp; spacing &amp; dimension of the contents when displaying the app on a mobile phone or a wide desktop browser</li></ul>

2.0	Correctly implement the flexible and responsive design of the temperature, rainfall, and air quality blocks when rendering the app on a mobile device
-1.0	Not using index.html as the Web app main page
-2.0	Not using JavaScript to build the body part of the whole web page
-2.0	Not using real-time Open Data from HKO Weather report API and 9-day Forecast API and OGCIQ Air Quality Health Index API
-5.0	Use external JavaScript/CSS libraries

### Plagiarism

Plagiarism is a very serious academic offence. Students should understand what constitutes plagiarism, the consequences of committing an offence of plagiarism, and how to avoid it. ***Please note that we may request you to explain to us how your program is functioning as well as we may also make use of software tools to detect software plagiarism.***