COMP3322 Modern Technologies on World Wide Web

Assignment Three

Total 16 points

Deadline: 17:00 March 24, 2023

Overview

You are going to design and develop a Web app that retrieves and displays 'real-time' passenger flight information from Hong Kong Airport Open Data. This REST web service provided by HK Airport returns historical data (including current calendar day) in JSON format. For this assignment, we only display current calendar date departure and arrival passenger flight information. The Web app should be nicely rendered on any desktop computer and smart phone.

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) create dynamic contents, (2) carry out AJAX communication for retrieving Open Data, and (3) selectively display flight information.
4. To practice using CSS styling to design a flexible and responsive layout.

Hong Kong Airport Open Data

We are going to make use of the data provided by the REST API of the Airport Authority for building our Web App.

Website: https://data.gov.hk/en-data/dataset/aahk-team1-flight-info

You can download the data dictionary about the request parameters and the response data set from here. https://www.hongkongairport.com/iwov-resources/misc/opendata/Flight_Information_DataSpec_en.pdf

To get the departure data for calendar date 2023-02-20, use this URL
https://www.hongkongairport.com/flightinfo-rest/rest/flights/past?date=2023-02-20&lang=en&cargo=false&arrival=false

To get the arrival data for calendar date 2023-02-20, use this URL
https://www.hongkongairport.com/flightinfo-rest/rest/flights/past?date=2023-02-20&lang=en&cargo=false&arrival=true

Please note that the date string must be in the format YYYY-MM-DD.

The datasets for departure and arrival flights have common fields and a few unique fields. Here are the example JSON data returned by the system on **2022-12-25**.

Each flight schedule consists of a time field, a list of flight no. and airline, and the status of the flight. For departure flight, it has the destination list, terminal, aisle, and gate no. for the passengers to on board the plane. **Please note that each departure flight may have more than one flight no. and airline and may go to more than one destination.** For the arrival flight, it has the origin list, baggage belt no., arrival hall, and parking stand. **It is also possible that an arrival flight may have more than one origin airport.** For the departure and arrival schedules, it may contain one or more flights which was/were scheduled in a date before the current calendar date but being departed at / arrived on a time in the current calendar date.

Please read the data dictionary file to learn the meaning of individual field and perform a few GET requests **using Firefox** to examine the returned JSON datasets.

Departure


```
0:
    date:                "2022-12-24"
    arrival:             false
    cargo:               false
  list:
    0:
        time:            "23:45"
      flight:
        0:
            no:          "LH 797"
            airline:     "DLH"
        1:
            no:          "CX 6799"
            airline:     "CPA"
        2:
            no:          "NZ 4213"
            airline:     "ANZ"
        status:          "Dep 00:04 (25/12/2022)"
        statusCode:      null
      destination:
          0:             "FRA"
        terminal:        "T1"
        aisle:           "F"
        gate:            "29"
1:
    date:                "2022-12-25"
    arrival:             false
    cargo:               false
  list:
    0:
        time:            "00:25"
      flight:
        0:
            no:          "CX 105"
            airline:     "CPA"
        1:
            no:          "AY 5095"
            airline:     "FIN"
        2:
            no:          "LH 7013"
            airline:     "DLH"
        3:
            no:          "LX 9513"
            airline:     "SWR"
        4:
            no:          "QR 5830"
            airline:     "QTR"
        5:
            no:          "BA 4567"
            airline:     "BAW"
        status:          "Dep 01:33"
        statusCode:      null
      destination:
          0:             "MEL"
        terminal:        "T1"
        aisle:           "BC"
        gate:            "49"
    1:
        time:            "00:40"
      flight:
```


Arrival


```
0:
    date:                "2022-12-25"
    arrival:             true
    cargo:               false
  list:
    0:
        time:            "00:20"
      flight:
        0:
            no:          "UO 781"
            airline:     "HKE"
        1:
            no:          "CX 5781"
            airline:     "CPA"
        status:          "At gate 01:00"
        statusCode:      null
      origin:
          0:             "SIN"
        baggage:         "13"
        hall:            "B"
        terminal:        ""
        stand:           "S41"
    1:
        time:            "01:40"
      flight:
        0:
            no:          "HX 766"
            airline:     "CRK"
        status:          "At gate 01:25"
        statusCode:      null
      origin:
          0:             "BKK"
        baggage:         "12"
        hall:            "B"
        terminal:        ""
        stand:           "W65"
    2:
        time:            "03:20"
      flight:
        0:
            no:          "HX 637"
            airline:     "CRK"
        1:
            no:          "AI 7175"
            airline:     "AIC"
        status:          "At gate 03:00"
        statusCode:      null
      origin:
          0:             "KIX"
        baggage:         "13"
        hall:            "B"
        terminal:        ""
        stand:           "W40"
    3:
        time:            "04:15"
      flight:
        0:
            no:          "UO 623"
            airline:     "HKE"
```

The destination and the origin airports are encoded by the standard IATA code. To provide useful information to the users, we have prepared another JSON file – iata.json, which contains more descriptive information about an airport. For examples,

IATA code: HKG
{
"iata_code": "HKG",
"name": "Hong Kong International Airport",
"continent": "AS",
"iso_country": "HK",
"iso_region": "HK-U-A",
"municipality": "Hong Kong"
}
	IATA code: MEL
	{
	"iata_code": "MEL",
	"name": "Melbourne International Airport",
	"continent": "OC",
	"iso_country": "AU",
	"iso_region": "AU-VIC",
	"municipality": "Melbourne"
	}

IATA code: KIX
{
"iata_code": "KIX",
"name": "Kansai International Airport",
"continent": "AS",
"iso_country": "JP",
"iso_region": "JP-27",
"municipality": "Osaka"
}
	IATA code: WUH
	{
	"iata_code": "WUH",
	"name": "Wuhan Tianhe International Airport",
	"continent": "AS",
	"iso_country": "CN",
	"iso_region": "CN-42",
	"municipality": "Wuhan"
	}

Unfortunately, we **cannot directly fetch** HK Airport flight data through our web app. It is because the hongkongairport.com server does not set up the CORS setting. Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. To overcome this limitation, we have prepared a **flight.php** program, which allows our web server to act as a client to request flight data from Hong Kong Airport and relay the data to our web app. The flight.php program accepts the same set of query strings as the Hong Kong airport server.

flight.php?date=2023-02-25&lang=en&cargo=false&arrival=true

Please place the flight.php program and the iata.json file under the public_html folder of your web server container.

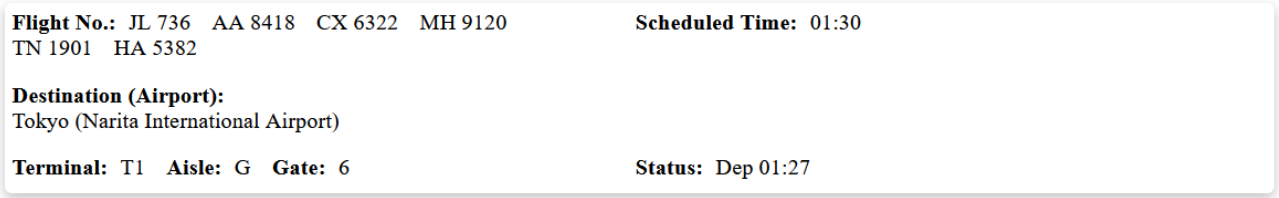
```
public_html
|
├── flight.php
├── iata.json
├── styles.css
└── index.html
```

For this assignment, you are going to develop the index.html and styles.css files.

Requirements

- You must implement the program using **vanilla JavaScript**. No other JS libraries and CSS libraries are allowed.
- You should develop the application using the course's LAMP docker containers. If you experienced technical problems using docker, use the department's i7.cs.hku.hk server for the development and testing.
- Retrieve HKA Open Data and iata.json **using AJAX XHR object or fetch().**
- Extract the data for each flight in the departure and arrival datasets.

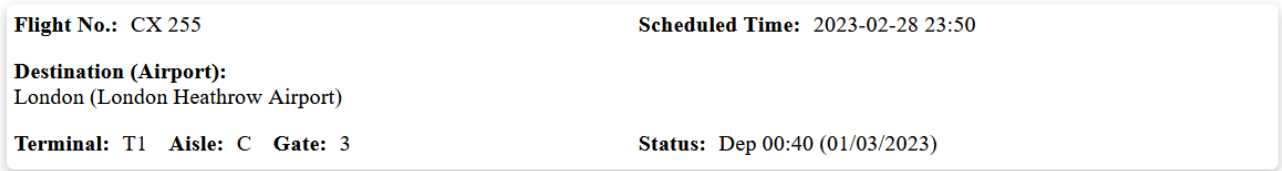
For a departure flight, list all the flight no., the scheduled time, destination location and airport, terminal, aisle, gate no., and the status of the flight. Here is an example departure flight on March 1, 2023, and the corresponding JSON data.




```
{
  "time": "01:30",
  "flight": [
    {
      "no": "JL 736",
      "airline": "JAL"
    },
    {
      "no": "AA 8418",
      "airline": "AAL"
    },
    {
      "no": "CX 6322",
      "airline": "CPA"
    },
    {
      "no": "MH 9120",
      "airline": "MAS"
    },
    {
      "no": "TN 1901",
      "airline": "THT"
    },
    {
      "no": "HA 5382",
      "airline": "HAL"
    }
  ],
  "status": "Dep 01:27",
  "statusCode": null,
  "destination": [
    "NRT"
  ],
  "terminal": "T1",
  "aisle": "G",
  "gate": "6"
}
```

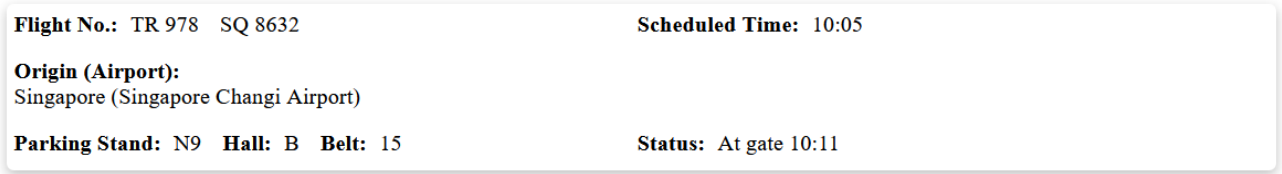

You can obtain the location and airport information from the **municipality and name fields** of the iata.json dataset.

Here is another example departure flight departed on March 1, 2023, but was scheduled on Feb 28, 2023 and the corresponding JSON data.




```
{
 "time": "23:50",
 "flight": [
  {
   "no": "CX 255",
   "airline": "CPA"
  }
 ],
 "status": "Dep 00:40 (01/03/2023)",
 "statusCode": null,
 "destination": [
  "LHR"
 ],
 "terminal": "T1",
 "aisle": "C",
 "gate": "3"
}
```


For the arrival flight, list all the flight no., the scheduled time, origin location and airport, parking stand, hall, belt no., and the status of the flight. Here is an example arrival flight on March 1 and the corresponding JSON data.




```
{
 "time": "10:05",
 "flight": [
  {
   "no": "TR 978",
   "airline": "TGW"
  },
  {
   "no": "SQ 8632",
   "airline": "SIA"
  }
 ],
 "status": "At gate 10:11",
 "statusCode": null,
 "origin": [
  "SIN"
 ],
```

```
  "baggage": "15",
  "hall": "B",
  "terminal": "",
  "stand": "N9"
 }
```


Here is another example arrival flight arrived on March 1, 2023, but was expected to arrive on Feb 28, 2023 and the corresponding JSON data.

Flight No.: CX 764 AC 9768 VN 3560 OM 5629 JL 7922 QR 5831 BA 4562 **Scheduled Time:** 2023-02-28 23:05

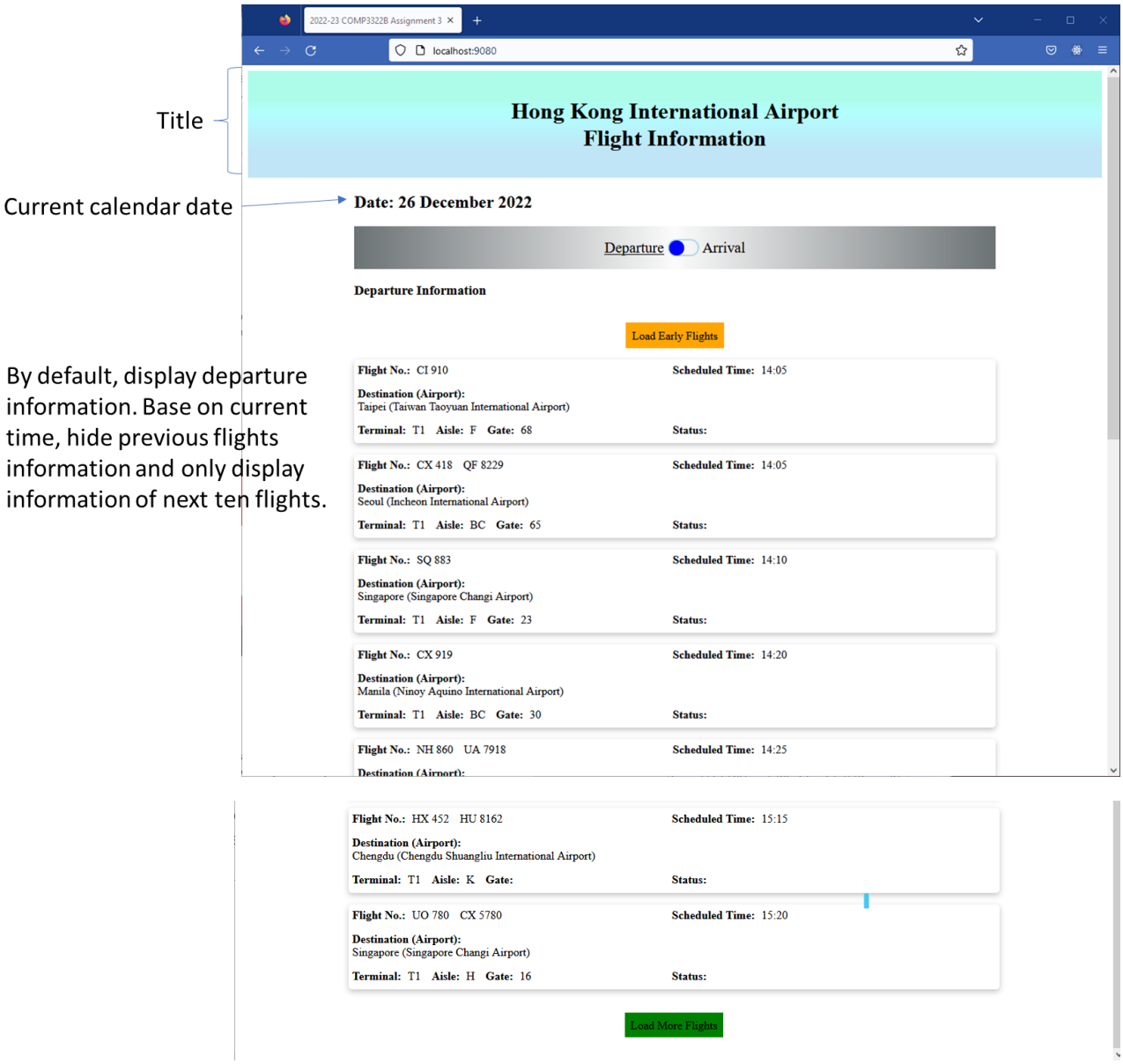
Origin (Airport):
Ho Chi Minh City (Tan Son Nhat International Airport)

Parking Stand: W65 **Hall:** B **Belt:** 12 **Status:** At gate 00:18 (01/03/2023)

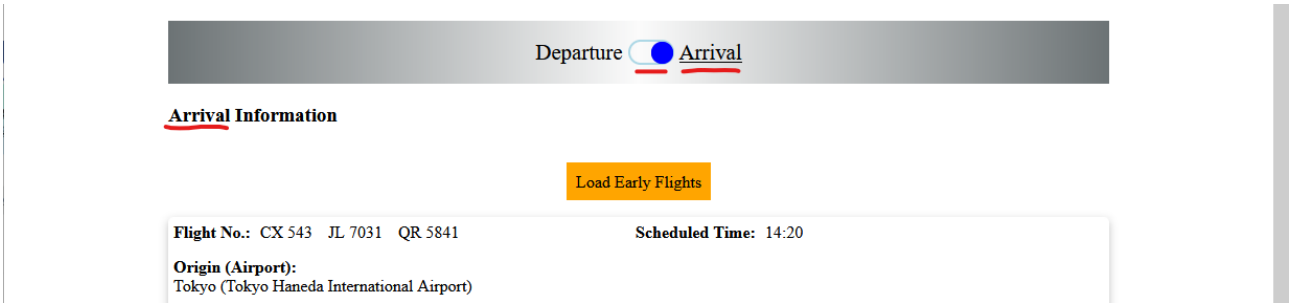

```
 {
  "time": "23:05",
  "flight": [
   {
    "no": "CX 764",
    "airline": "CPA"
   },
   {
    "no": "AC 9768",
    "airline": "ACA"
   },
   {
    "no": "VN 3560",
    "airline": "HVN"
   },
   {
    "no": "OM 5629",
    "airline": "MGL"
   },
   {
    "no": "JL 7922",
    "airline": "JAL"
   },
   {
    "no": "QR 5831",
    "airline": "QTR"
   },
   {
    "no": "BA 4562",
    "airline": "BAW"
   }
  ],
  "status": "At gate 00:18 (01/03/2023)",
  "statusCode": null,
  "origin": [
   "SGN"
```

```
  ],
  "baggage": "12",
  "hall": "B",
  "terminal": "",
  "stand": "W65"
 }
```


- Our Web App consists of a title, the current date, and a mechanism to switch between displaying the departure flights or arrival flights. The page displays the departure flights by default.
  Here is the screenshot of a **sample** implementation.



- Implement a mechanism that allows user to select which set of flight information to display. There is a toggle switch for the user to switch between displaying departure flights and arrival flights. Implement an animation to slowly flip the switch from left to right and from right to left. Also, adjust the headings and the flight information accordingly.



- Based on current time, hide previous flight information, and only display information of next ten flights. Implement mechanisms that allow user to view previous flight information and load more flight information on that date.
  When the user clicks on 'Load Early Flights' button, the system shall display all previous flights starting from the 1st flight entry on that calendar date. When the user clicks on 'Load More Flights' button, the system shall display all coming flights on that date.

  When the user selects another dataset to display, e.g., switching from displaying departure flights to arrival flights, the system should resume the original setting of the display, i.e., ***it bases on current time, recalculates the starting flight, displays next ten flights, and displays the 'Load Early Flights' and 'Load More Flights' buttons***.

- The base document of our Web app is the **index.html** file. You can add any HTML tags to the <body> part of the file. It is up to you to decide whether placing all JavaScript code in the index.html file or in an external .js file. To display the flight information, 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.
- You **should implement** appropriate CSS settings for rendering the Web app on a desktop browser (with at least 1000px screen width) and a smartphone.

Testing platform

We shall place all your submitted files in the LAMP container set and use Chrome and/or Firefox to test the programs.

Submission

Please finish this assignment before March 24 Friday 17:00. Submit the following files:

1. index.html
2. styles.css
3. any relevant JS or CSS files

Grading Policy

Points	Criteria
6.0	• Able to download current calendar date flight information. • Correctly extract and display the required departure and arrival flight information.
2.5	Implement the mechanism (with animation) to switch between displaying the departure flights and arrival flights.
5.0	• Implement the 'Load Early Fights' and 'Load More Flights' buttons. • Based on current time, the system displays the next ten flights information.

	• After switching, returns to display the next ten flights information according to current time.
2.5	Styling and layout ▪ Support both desktop and mobile settings ▪ Basic styling
-1.0	Not using index.html as the Web app main page
-5.0	Not using JavaScript to retrieve the Open Data and iata.json to build the body part of the whole web page
-8.0	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.***