WEB SERVICES AND SERVICE-ORIENTED ARCHITECTURE

PROGRAMMING ASSIGNMENT 1

Design of the application:

For the web application, the dashboard has been built for the users wherein they can check their latest updates such as current weather, number of coronavirus cases, latest USD currency value. The user would be required to enter the zip code based on which all of the details would be fetched. Also, the user can choose to convert currencies of any two countries.

Features

The web application implements the following features:

1. Fetch the city and state from the input zip code.
2. Fetch the current weather from the zip code.
3. Display the count of current coronavirus cases from the zip code entered.
4. Display today's USD amount
5. Convert between currencies of any two countries.

API's used

For the purpose of this web application, the following web services were used:

1. Find state and city on basis of ZipCode (REST): https://www.zipcodeapi.com/API

The service is a REST service and given a zip code, the API returns information such as city, state, longitude, latitude, timezone among other information. The web application used city, state, and timezone information to display on the user's dashboard.

Sample Request:
https://www.zipcodeapi.com/rest/kqD3NtfpL6AmOsAP1rQVmHacZy0aGyioSqwPwQ1gfhqjqMWx3adhwyTBt5L9Ne7S/info.json/14623/degrees

Sample Response:


```
{"zip_code":"14623","lat":43.087343,"lng":-
77.642087,"city":"Rochester","state":"NY","timezone":{"timezone_identifier":"
America\/New_York","timezone_abbr":"EST","utc_offset_sec":-
18000,"is_dst":"F"},"acceptable_city_names":[],"area_codes":[585,716]}
```

2. Current weather updates (REST): https://openweathermap.org/api

This REST API from openweathermap.org returns weather data for many locations. I used the current weather data API from the collection of all the APIs offered. The current weather data API returns longitude, latitude, weather, timezone, wind, country among all other information. I used '*weather.main*', '*weather.description*', '*weather.icon*', '*name'*, '*sys.country*', '*main.temp*', and '*main.feels_like*' fields. Full details about the response fields can be found here: https://openweathermap.org/current#current_JSON

Sample Request:
http://api.openweathermap.org/data/2.5/weather?zip=14623&appid=0827e2a24e5b696e8570714f34824e50&units=metric

Sample Response:


```
{"coord":{"lon":-
77.6344,"lat":43.0834},"weather":[{"id":803,"main":"Clouds","description":"br
oken
clouds","icon":"04n"}],"base":"stations","main":{"temp":2.84,"feels_like":-
2.91,"temp_min":2.22,"temp_max":3.33,"pressure":1012,"humidity":75},"visibili
ty":10000,"wind":{"speed":5.14,"deg":260},"clouds":{"all":75},"dt":1614143435
,"sys":{"type":1,"id":5698,"country":"US","sunrise":1614167669,"sunset":16142
07197},"timezone":-18000,"id":0,"name":"Rochester","cod":200}
```


3. CoronaVirus updates (REST): https://anypoint.mulesoft.com/exchange/68ef9520-24e9-4cf2-b2f5-620025690913/covid19-data-tracking-api/

This REST API provided by MuleSoft is an excellent source to get the latest COVID-19 information by zip code. Amongst the collection of APIs provided by MuleSoft, I chose the one with New York Times as the data source since it is the most updated source. Given a zip code, the API returns information such as box coordinates of longitude and latitude, and death count, and positive count per county. I used the information about the death count and positive county for the given zip code to display on the user's dashboard.

In addition, this REST API also provides a way to access the past 7 days' information for the zip code which is displayed on the dashboard for a more comprehensive view for the user.

Sample Request:
https://localcoviddata.com/covid19/v1/cases/newYorkTimes?zipCode=14623&daysInPast=7

Sample Response:


```
{
  "zipCd": "14623",
  "counties": [
    {
      "countyName": "Monroe County",
      "geo": {
        "rightTopLatLong": -77.6352000000,
```

```
        "leftBottomLatLong": 43.0876130000,
        "leftTopLatLong": 43.0876130000,
        "rightBottomLatLong": -77.6352000000
      },
      "historicData": [
        {
          "date": "2021-02-22",
          "deathCt": 976,
          "positiveCt": 51675,
          "recoveredCt": null
        },
        {
          "date": "2021-02-21",
          "deathCt": 972,
          "positiveCt": 51565,
          "recoveredCt": null
        },
        {
          "date": "2021-02-20",
          "deathCt": 969,
          "positiveCt": 51424,
          "recoveredCt": null
        },
        {
          "date": "2021-02-19",
          "deathCt": 968,
          "positiveCt": 51256,
          "recoveredCt": null
        },
        {
          "date": "2021-02-18",
          "deathCt": 964,
          "positiveCt": 51135,
          "recoveredCt": null
        },
        {
          "date": "2021-02-17",
          "deathCt": 960,
          "positiveCt": 50936,
          "recoveredCt": null
        }
      ]
    }
  ]
}
```

4. Currency conversion (SOAP): http://currencyconverter.kowabunga.net/converter.asmx

This is a SOAP API that I used for displaying currency-related information on the user's dashboard. This SOAP API supports various operations for different tasks. Some of those operations include GetConversionAmount, GetConversionRate, GetCultureInfo, GetCurrencies, GetCurrencyRate, GetCurrencyRates, and GetLastUpdateDate. Amongst all the operations provided I used the following operations:

- GetCurrencies: To fetch the list of all the available currencies.

  URL: http://currencyconverter.kowabunga.net/converter.asmx?op=GetCurrencies

  Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCurrencies xmlns="http://tempuri.org/" />
  </soap:Body>
</soap:Envelope>
```

  Sample Response:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCurrenciesResponse xmlns="http://tempuri.org/">
      <GetCurrenciesResult>
        <string>USD</string>
        <string>INR</string>
      </GetCurrenciesResult>
    </GetCurrenciesResponse>
  </soap:Body>
</soap:Envelope>
```

- GetCurrencyRate: This operation returns the current rate for a given currency. For example for USD, the API returns 1.21

URL: http://currencyconverter.kowabunga.net/converter.asmx?op=GetCurrencyRate

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCurrencyRate xmlns="http://tempuri.org/">
      <Currency>USD</Currency>
      <RateDate>2021-02-16</RateDate>
    </GetCurrencyRate>
  </soap:Body>
</soap:Envelope>
```

Sample Response:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetCurrencyRateResponse xmlns="http://tempuri.org/">
      <GetCurrencyRateResult>1.21479859</GetCurrencyRateResult>
    </GetCurrencyRateResponse>
  </soap:Body>
</soap:Envelope>
```

- GetConversionRate: This operation converts currencies of two countries.

URL: http://currencyconverter.kowabunga.net/converter.asmx?op=GetConversionRate

Sample Request:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetConversionRate xmlns="http://tempuri.org/">
      <CurrencyFrom>USD</CurrencyFrom>
      <CurrencyTo>INR</CurrencyTo>
      <RateDate>2021-02-16</RateDate>
    </GetConversionRate>
  </soap:Body>
</soap:Envelope>
```

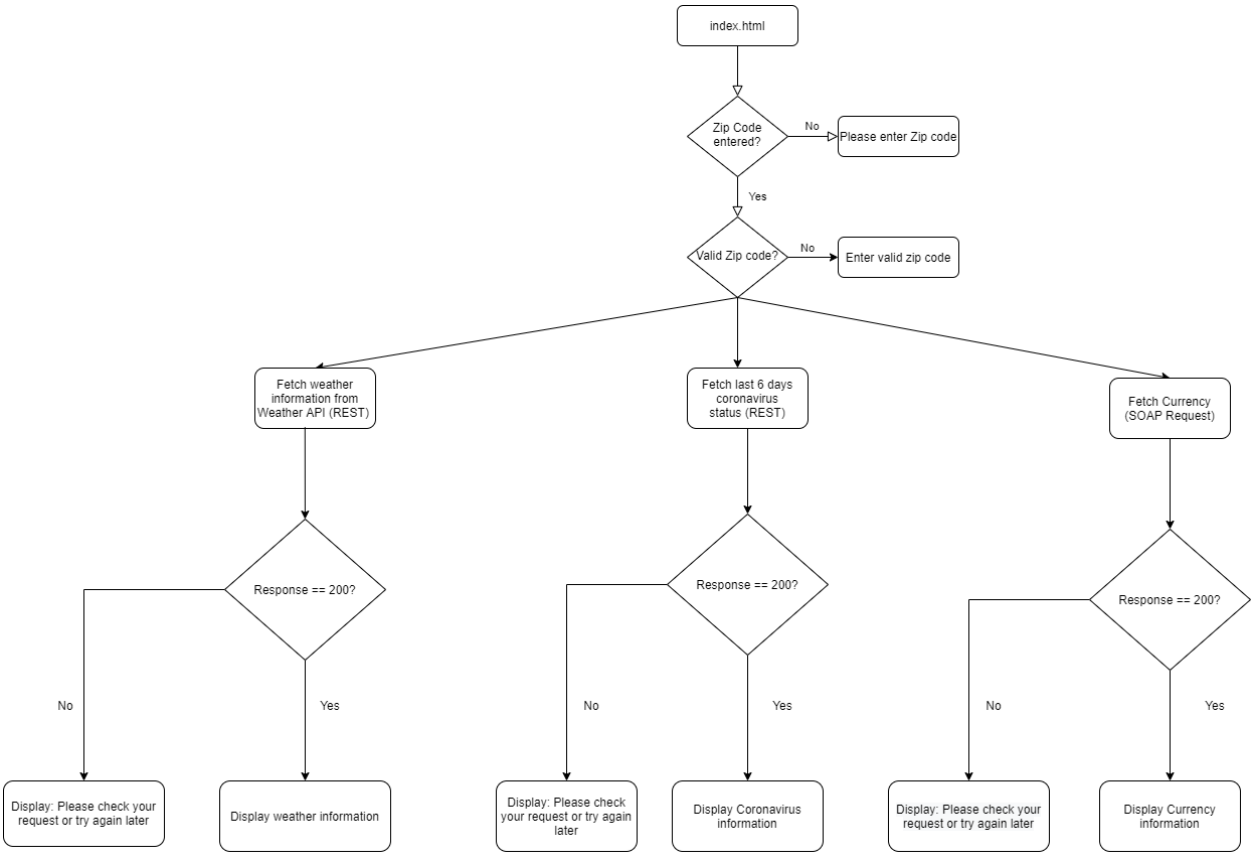
Sample Response:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

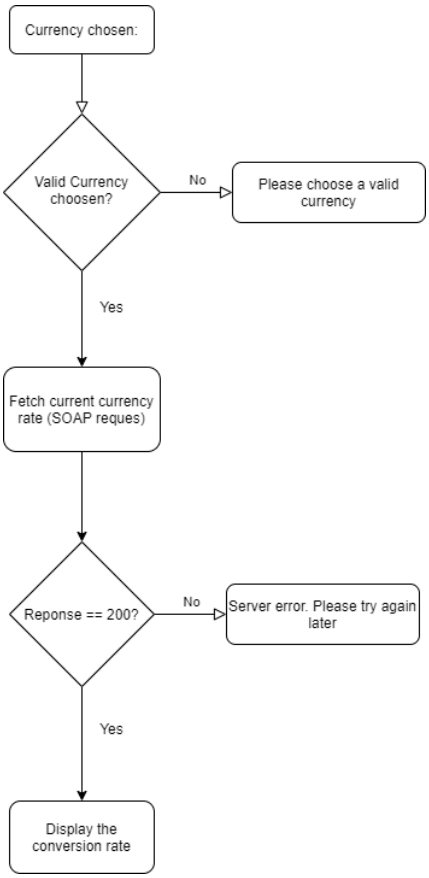
```
    <GetConversionRateResponse xmlns="http://tempuri.org/">
      <GetConversionRateResult>72.68</GetConversionRateResult>
    </GetConversionRateResponse>
  </soap:Body>
</soap:Envelope>
```

Flowchart:

The following figure displays the flow of the application:

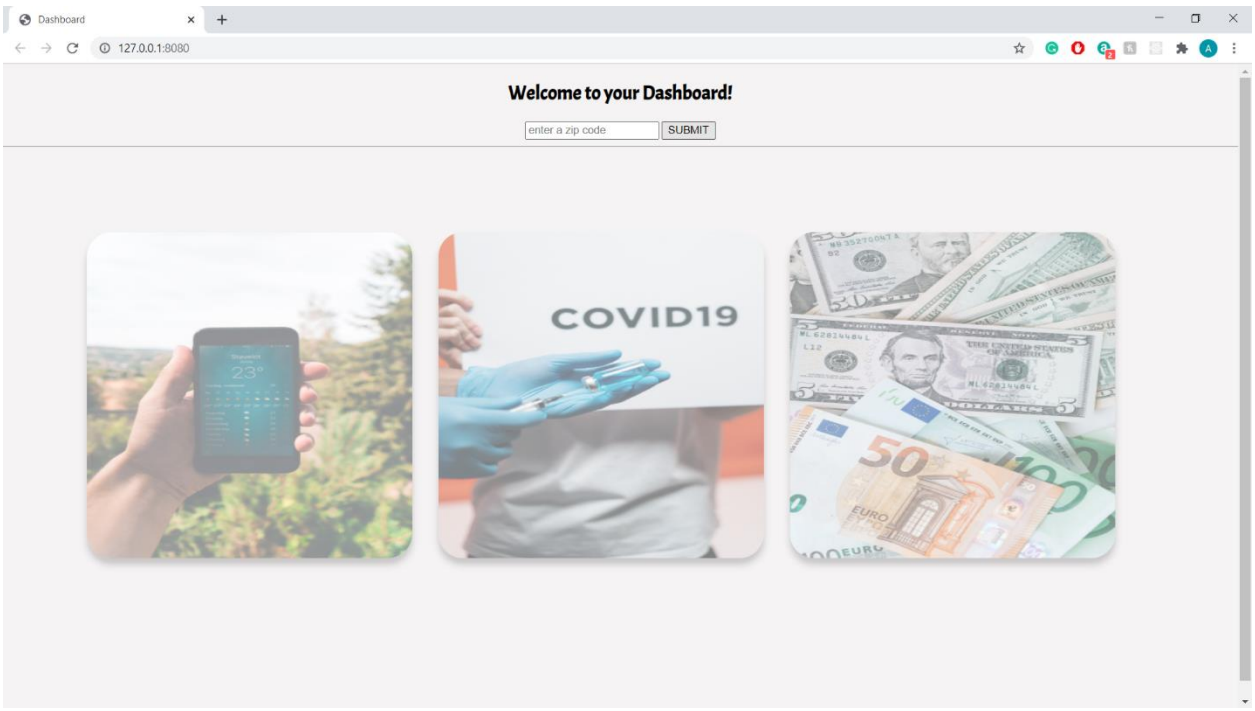


Flowchart for Currency:

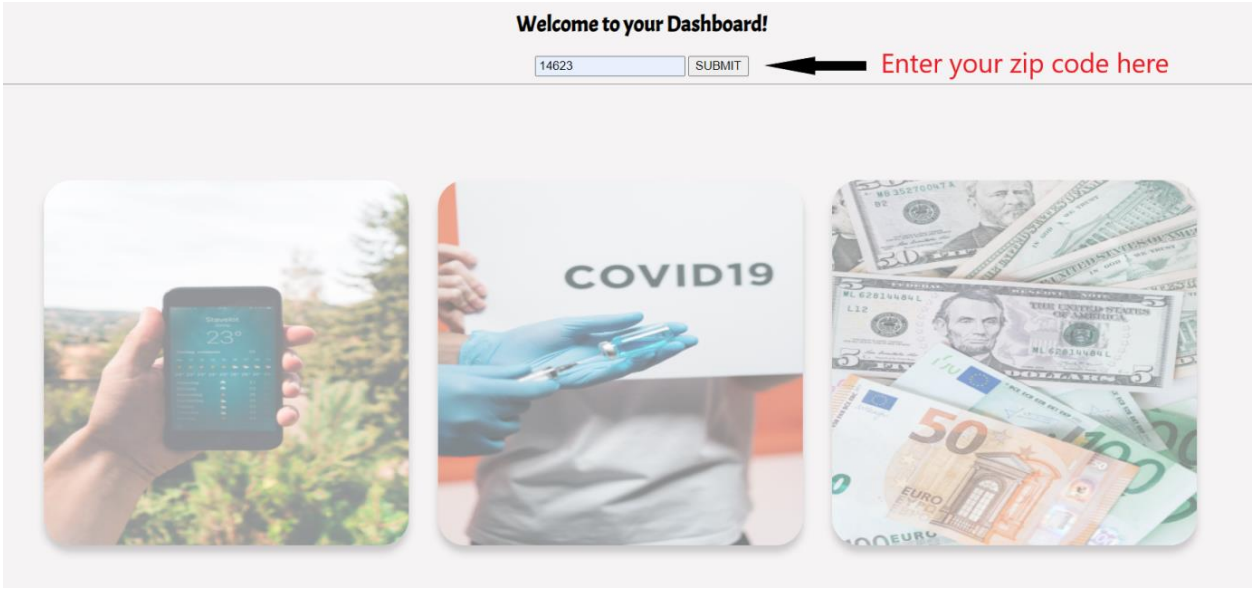


Screenshots of the application:

- Run `npx live-server` from the root folder to run the application.
- The application will open at http://127.0.0.1:8080/ and display the HTML page as follows:

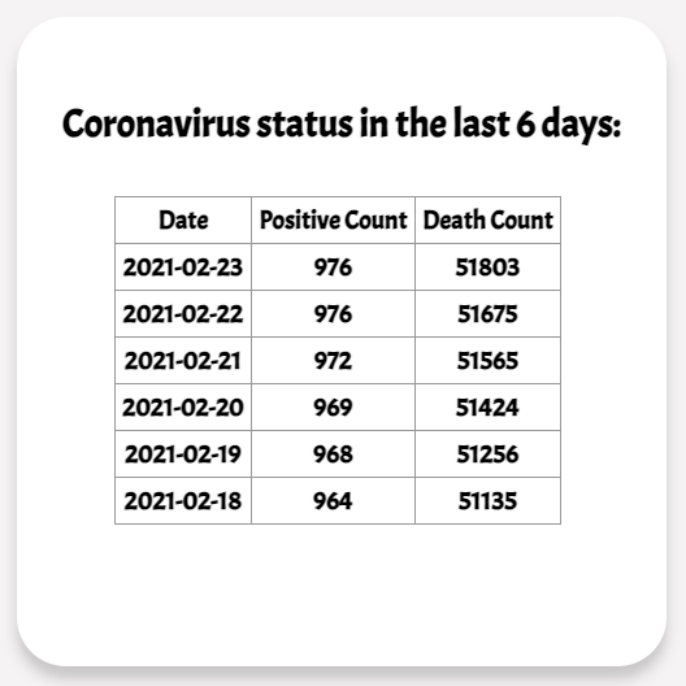


- Enter a zip code in the input box and press Submit



- The application displays three information on each individual card:
    - Weather: Fetched from https://openweathermap.org/api

- Coronavirus: Fetched from https://anypoint.mulesoft.com/exchange/68ef9520-24e9-4cf2-b2f5-620025690913/covid19-data-tracking-api/



Coronavirus status in the last 6 days:

Date	Positive Count	Death Count
2021-02-23	976	51803
2021-02-22	976	51675
2021-02-21	972	51565
2021-02-20	969	51424
2021-02-19	968	51256
2021-02-18	964	51135

- Currency: Fetched from http://currencyconverter.kowabunga.net/converter.asmx

Additionally, the user can also use the dropdowns to convert between currencies of different countries. A screenshot of the conversion between USD and INR is shown below:

Languages and Frameworks used:

1. HTML – for display the page content
2. CSS – for styling the web page
3. JavaScript: for invoking web services and displaying the response on UI.

No frameworks were used for this web application.

REFERENCES:

1. OpenWeather Map API - https://openweathermap.org/api
2. CurrencyConverter API - http://currencyconverter.kowabunga.net/converter.asmx
3. MuleSoft Anypoint public Covid API - https://anypoint.mulesoft.com/exchange/68ef9520-24e9-4cf2-b2f5-620025690913/covid19-data-tracking-api/