State Population Statistic

Prepared by:

Dung Tran

Table of Contents

1. System requirement 3

2. System Design 3

2.1 Presentation Layer: 3

2.2 Business Layer: 5

3. Deployment Guide: 5

3.1 Web deployment: 5

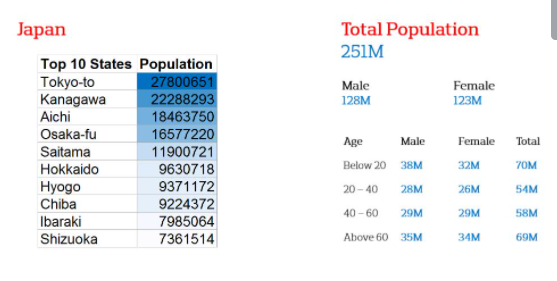
3.2 App deployment: 5

3.3 Verify State Population Web: 7

4. Limitation: 8

# System requirement

Develop a web tool to analyse the total population in Japan by Gender and Age group

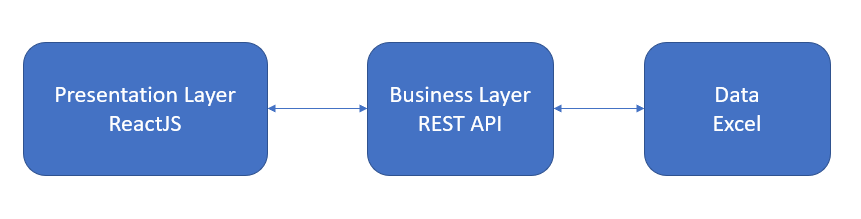


# System Design

## Language and tools

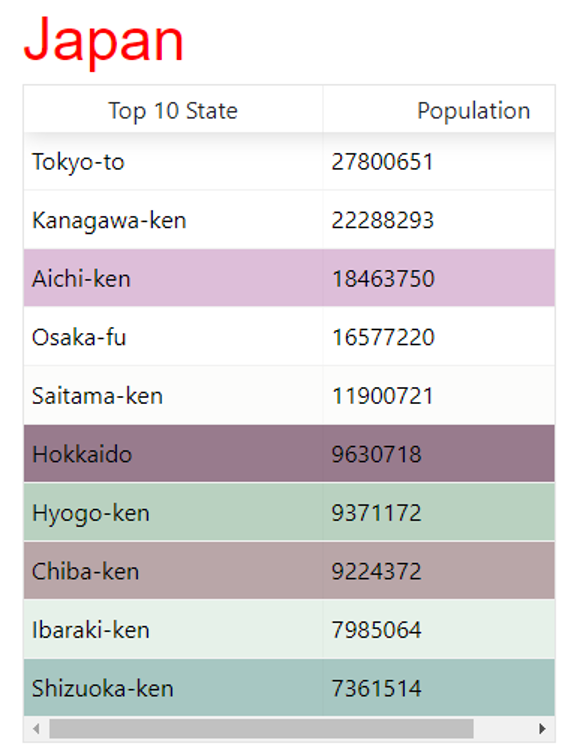
## 

|  |  |
| --- | --- |
| **SN** | **Language** |
| 1 | ReactJS |
| 2 | Visual Studio 2019 |
| 3 | Excel file to store data |

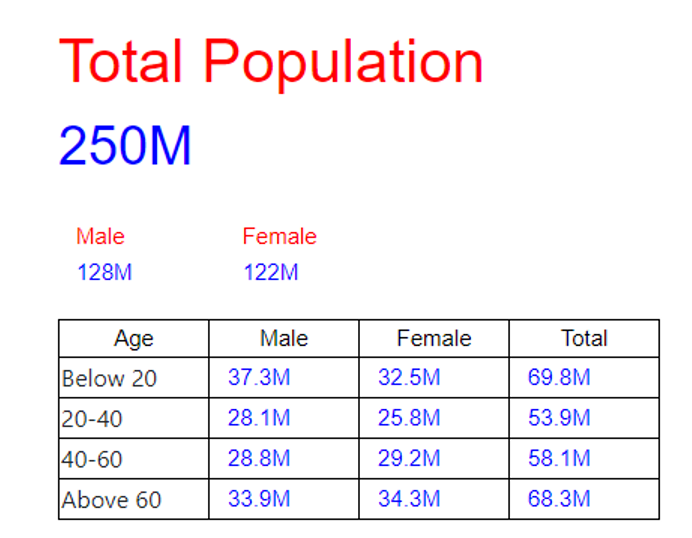


## Presentation Layer:

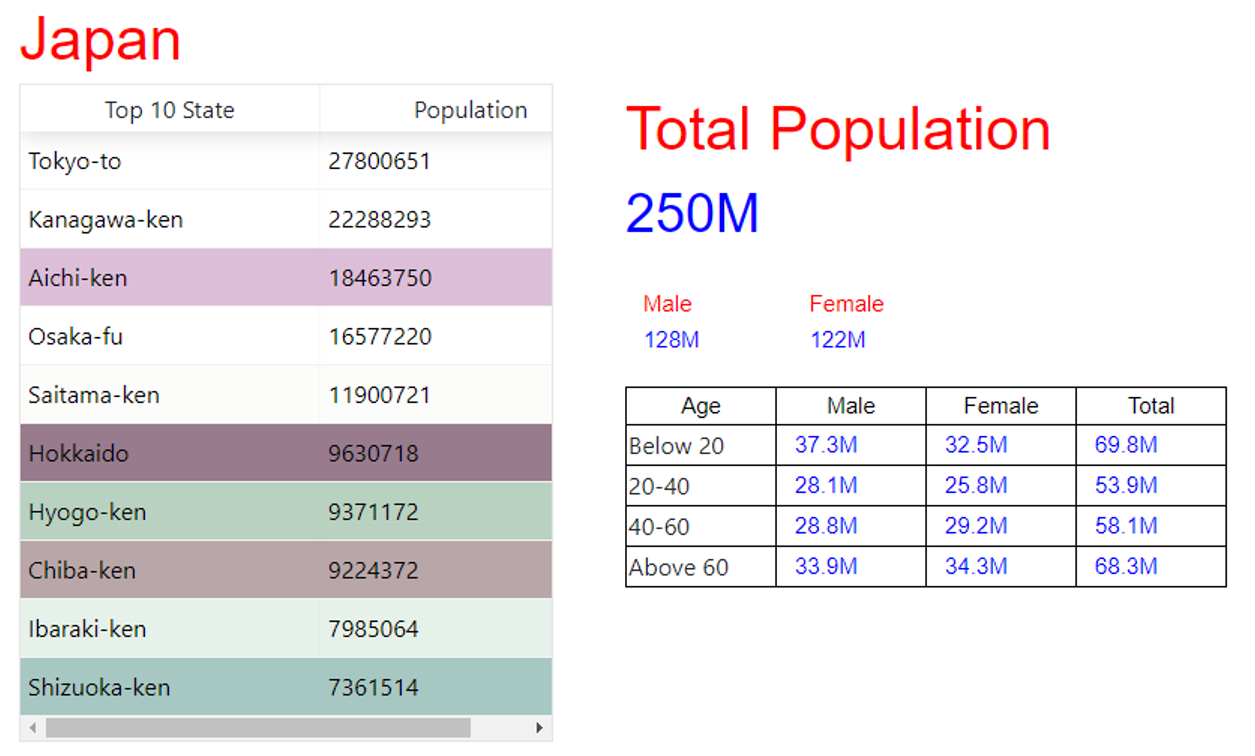
* + Using ReactJS to create the single-page applications.
  + Using Axios to call REST API **one time** to get ten state details and also count total population, total males, total females from excel file to improve the performance
  + Design two components:
    - **Parent component**



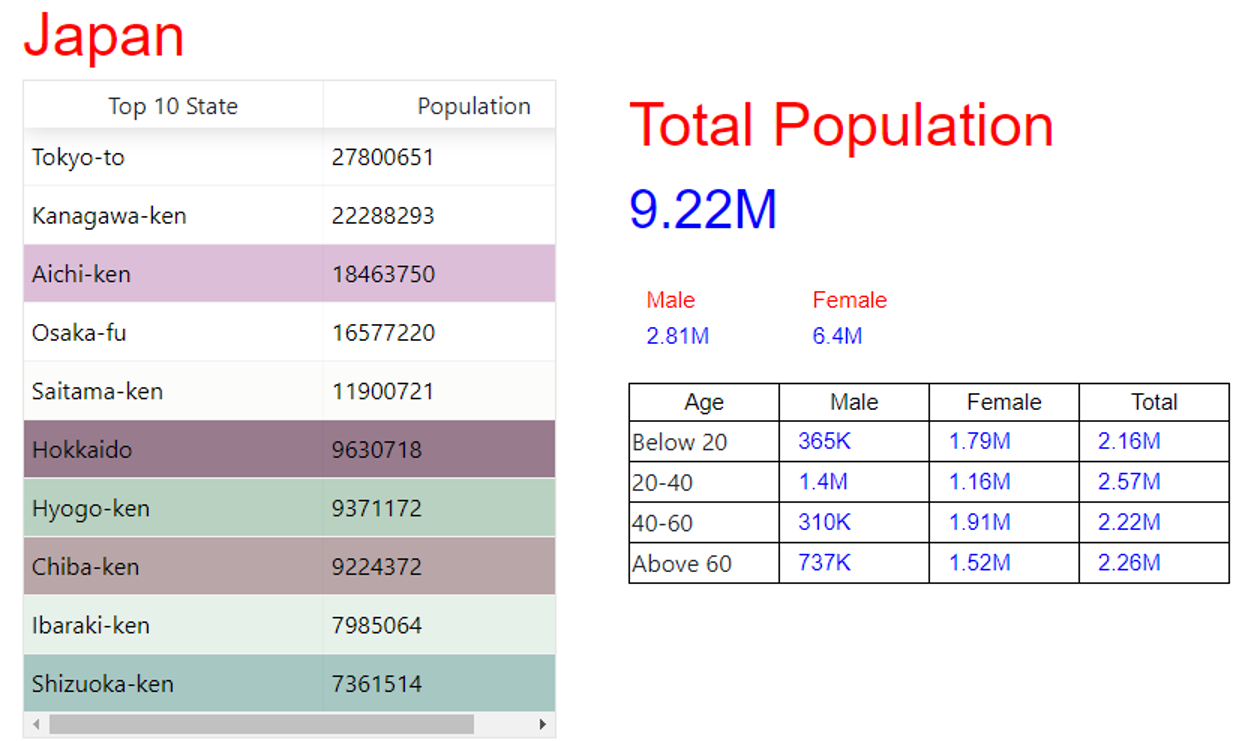
* + - **Child component**

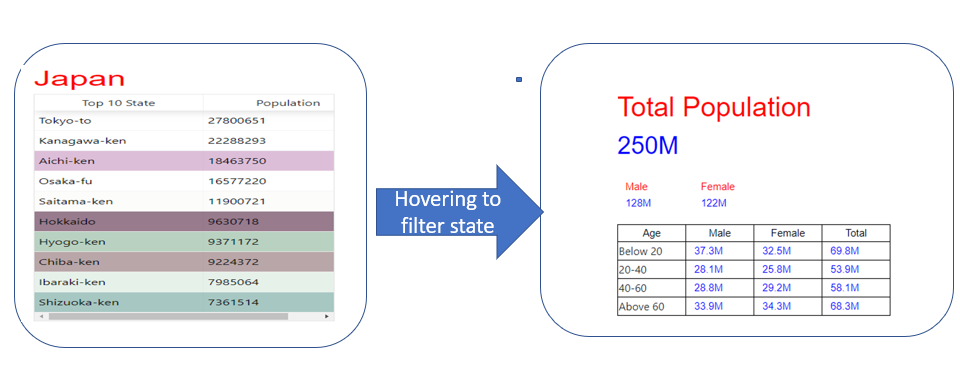


* + By default, and without hovering a state, the stats on the right should display the total population in Japan:
    - If there is no hovering, parent component passes all data of states to Child component



* + Statistics on the right should update based on hovering the cursor over any of the state
    - When hovering the state, the system will filter current data of the selected state and pass to Children component





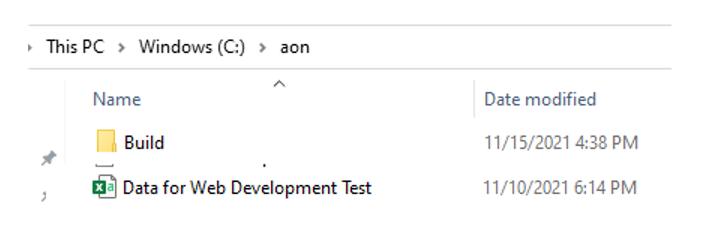
## Business Layer:

* + Using C# (version 2019) to create one REST API
  + Presentation layer will use Axios to call this Rest API
  + Rest API just read excel file **one time** and get all needed data.
  + Rest API must apply **Cross-Origin Resource Sharing** (CORS) when presentation layer calls business layer (REST API)

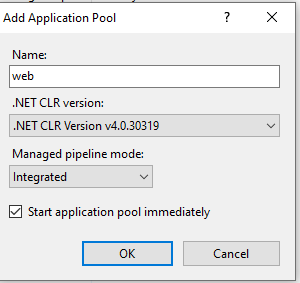
# Deployment Guide:

## 3.1 Web deployment:

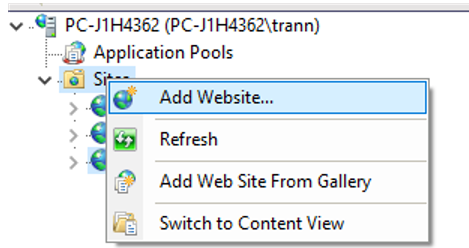
* Unzip file **Build.zip** to C:\AON\ with the structure:

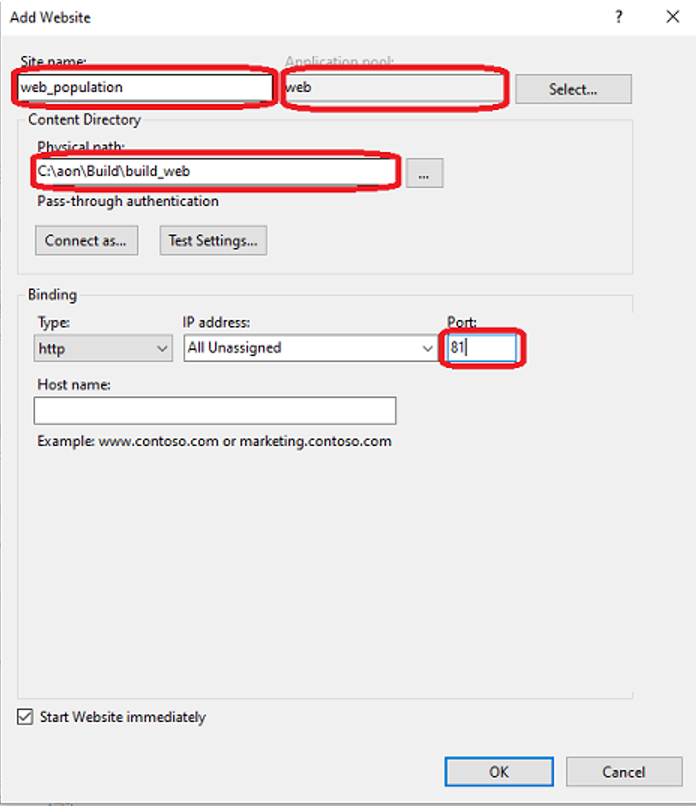


* Open IIS,
* Create new application pool “**web**”

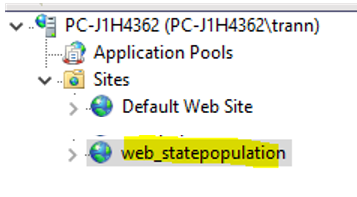


* Create new web application as the picture below:



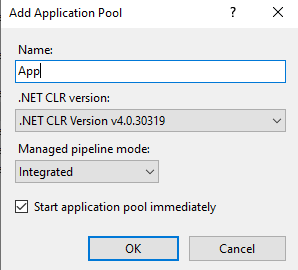


* Click Ok and check the web site is successfully created:

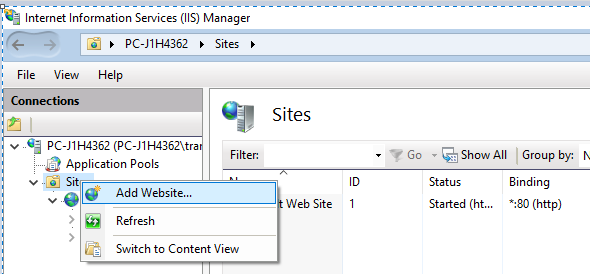


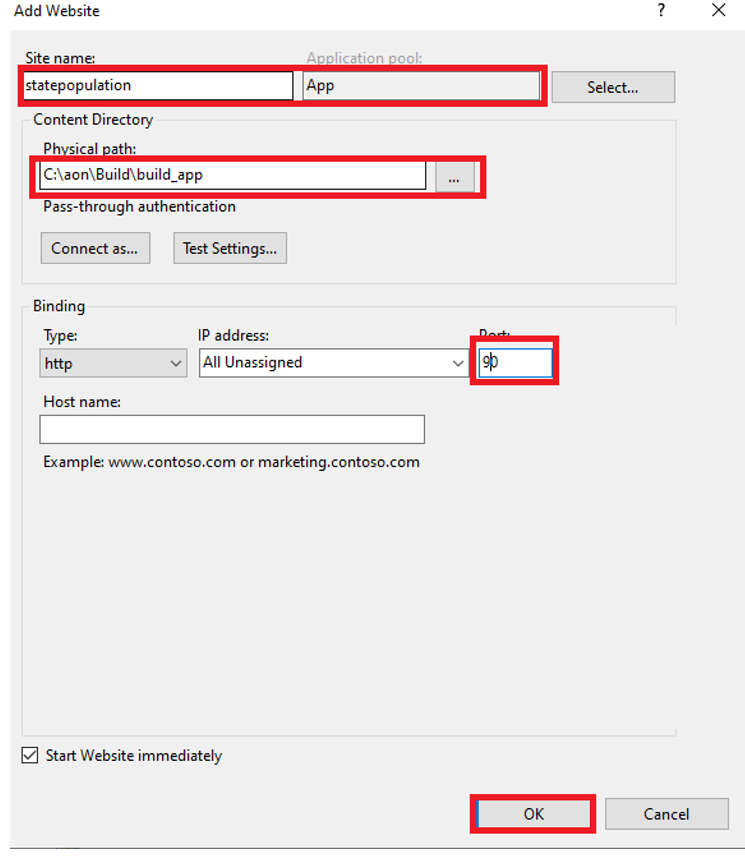
3.2 App deployment:

* Open IIS
* Create new application pool “App”:

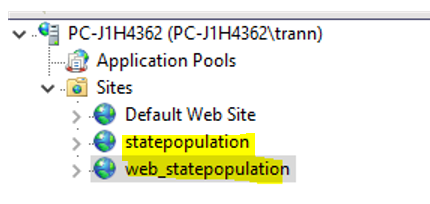


* Create new Rest API as the picture below:

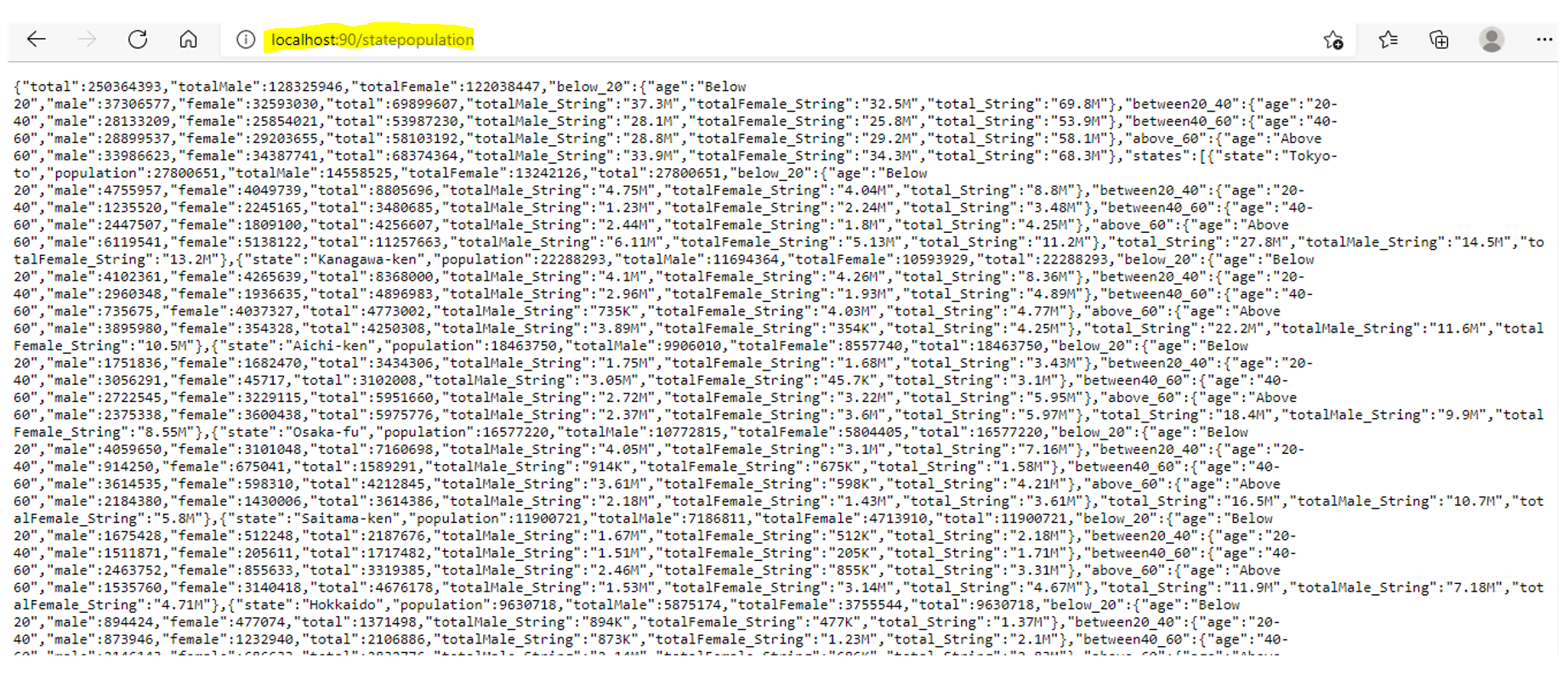




* Click “Ok” to finish. Two web sites are successfully created as the picture below:

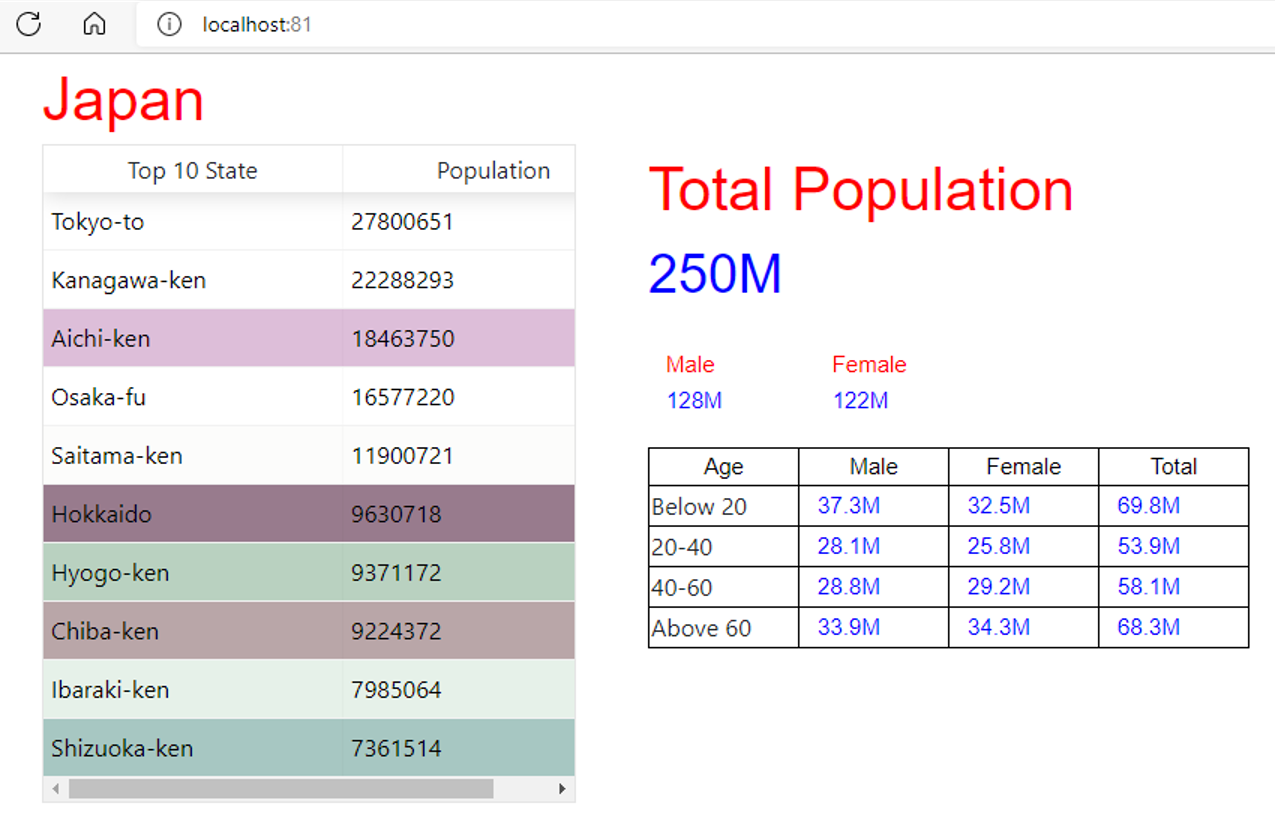


* Open IE or Chrome to test API browser:
  + “**http://localhost:90/statepopulation** to make sure json returned:

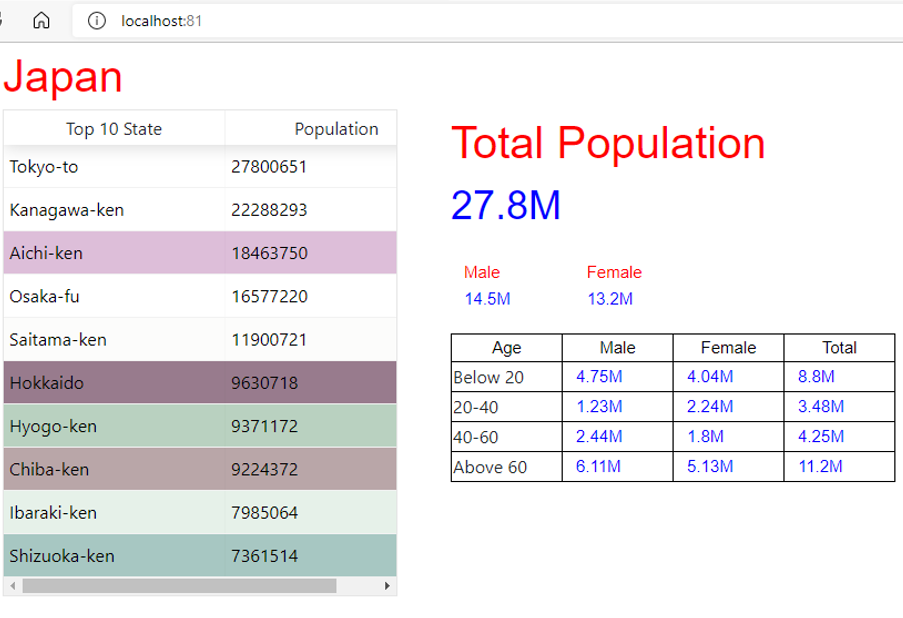


## 3.3 Verify Population Web:

* Using IE or Google Chrome to do sanity: type address “**localhost:81**”



* Move mouse on each State to see data on right changed



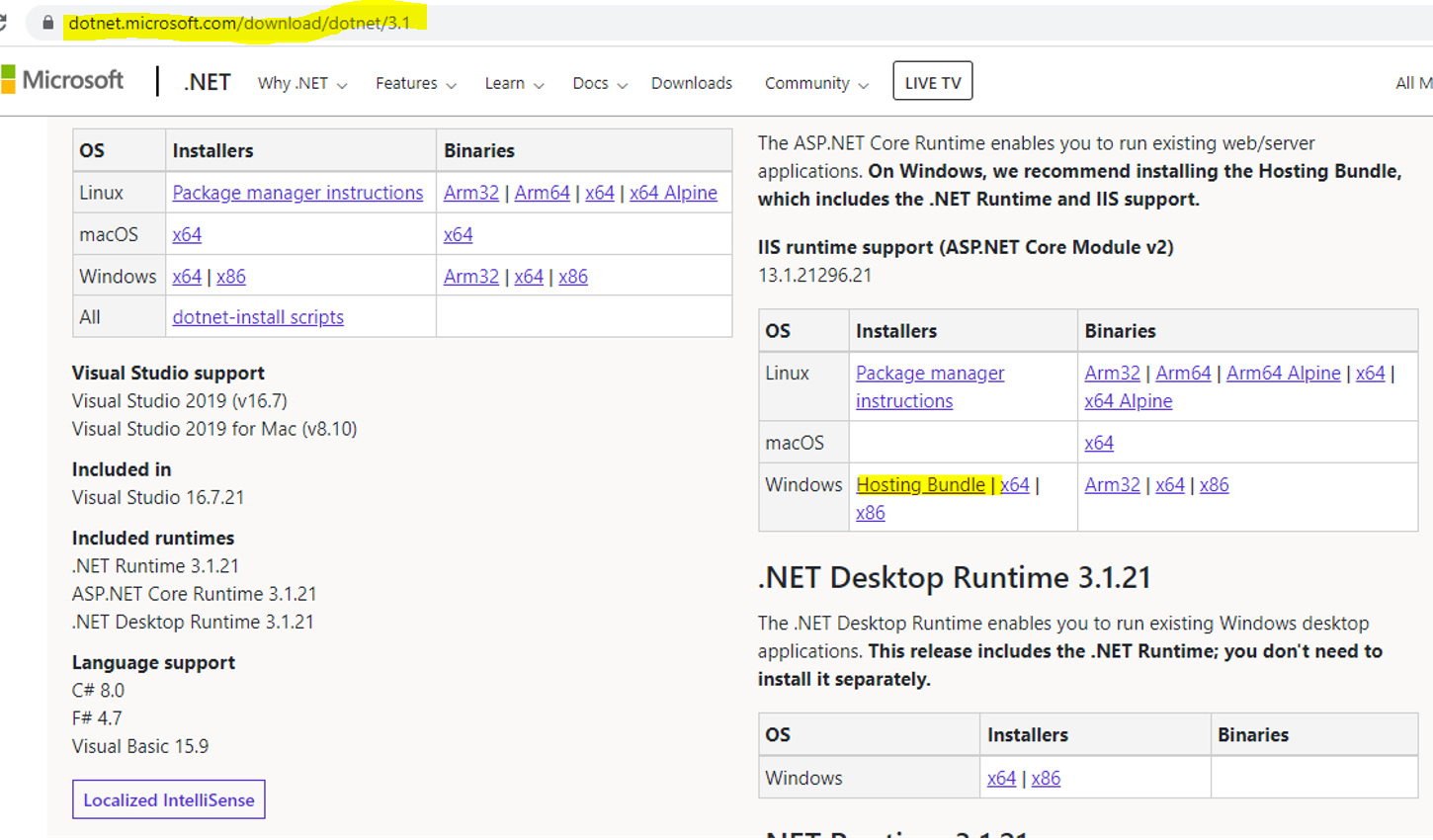
# Limitation:

* States should be shaded based on the total population (Larger the population, darker the shade): This feature needs more time to implement.

# .Net Framework installation

As the Rest API is implemented by .Net code, version 2019. Please install IIS runtime Support (ASP.NET Core Module V2) Hosting Bundle .

Please download from <https://dotnet.microsoft.com/download/dotnet/3.1>



# Contact Details:

If there is any issue when deployment, please contact

* Cell phone number (65) 96304234.
* Send the issue with the pictures to email [trananhdung@yahoo.com](mailto:trananhdung@yahoo.com)