Take-Home Technical Assessment

Please attempt **only one (1)** of the technical questions below.

There is a front-end (web) and backend (REST API) question. You are to complete the assessment before the interview. We will be going through your submission on that day.

If you have any questions about the technical assignments, do send them to michael_cheng@mom.gov.sg.

Submitting your assignment

- Your code must be hosted on Github, or any other similar service, in a publicly-accessible repository (eg. GitHub / BitBucket / GitLab).
- You may include a section with the assumptions, interpretations you have made about the requirements above or notes on your architecture decisions.
- Do show the progress of your work with atomic git commits.
- Do not host your app on a public server.
- Please provide instructions on how to run your source code locally on our laptop in the README file.
- Containers are acceptable - make sure the Dockerfile is included in the repository.
- Please send us a link to your repository (to the person who sent you this document) when you completed your assessment.

Important!

We will assess your submission holistically (i.e. not just in terms of functionality), including factors such as:

- Readability and code cleanliness
- Good coding practices
- Code structure/design, e.g. modularity, testability

Q1. Frontend Technical Assessment

The Task: Weather Forecast & Traffic Cam Website

In this assessment, you are required to develop a front-end application using any of the framework(s) listed below

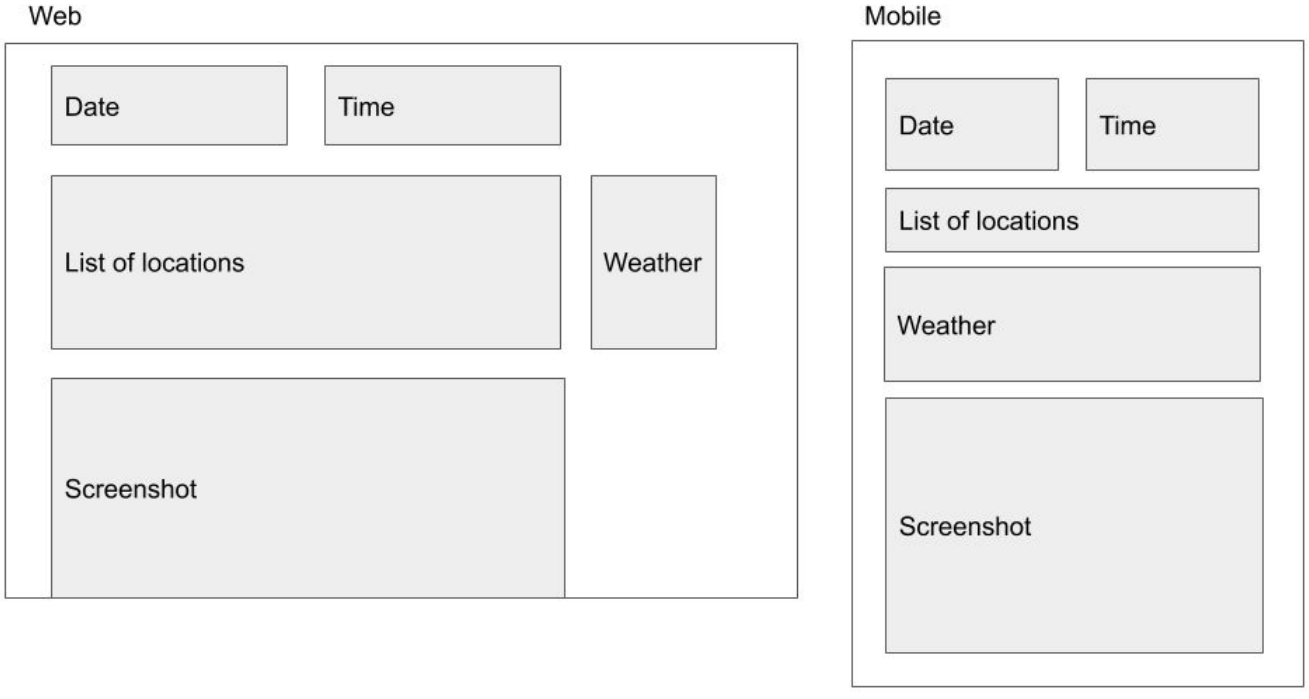
1. ReactJS
2. Angular.js
3. Vue.js
4. jQuery
5. Plain HTML/JS

Requirements

The team wants to create a simple application utilizing the open source data from https://data.gov.sg/developer#realtime. You will be using the following APIs:

1. Traffic Images (https://data.gov.sg/dataset/traffic-images)
2. Weather Forecast (https://data.gov.sg/dataset/weather-forecast)

The brief idea of the user-interface is illustrated in the screenshot below:



Features:

1. Allow the user to choose a date and time, then show a list of locations with traffic cam photos for the specified date and time. (API 1: Traffic Images)
2. Show the list of locations from API 1 (Traffic Images) only has lat/long without name, use a reverse geocoding service (API 2: Weather Forecast) to display more user friendly location names
3. When the user selects a location from the list, show the traffic cam photo, and also the weather info for that location from API 2 (Weather Forecast) (or the nearest available weather info depending on what API 2 can return)

As the location information are returned in latitude and longitude format, you are required to look for an API that does reverse geo-coding to show its human readable names instead.

What are you supposed to do?

- Create a frontend application based on the requirements and mock-up above
- Create proper UI components for the application
- Handle the responsiveness of the UI in the application
- Create the necessary CSS styling ensure the interface is aesthetically pleasing
- Develop an application with good user-experience (UX)
- Feel free to change the design of the application if you have a better idea
- You will be asked to showcase and explain to us some of your development concept on the interview day itself.

Q2. Backend Technical Assessment

The Task: Government Grant Disbursement API

Create a RESTful API that would help your team decide on groups of people who are eligible for various upcoming government grants.

These grants are disbursed based on certain criteria - like total household income, age, occupation, etc. The API should be able to build up a list of recipients and which households qualify for it.

For ease of definition, a household is defined by all the people living inside 1 physical housing unit.

Technical Requirements

- You may attempt to build this API in the following languages:
    - JavaScript / TypeScript
    - Ruby
    - Python (version 3 only)
    - Java / Kotlin
    - Go Lang
    - C#
- You may also use a programming framework.
- All endpoints should use URL path and HTTP verbs that are appropriate for the action taken.
- You do not need to implement any authentication.
- You may use a persistent database store to save the data (eg. MySQL, Postgres, Redis, MongoDB, etc). Include instructions on how to setup the database locally (eg. DB schema, SQL scripts to run, command line instructions to migrate the DB).

End-Points

1. Create Household
    a. This endpoint lets you create the household (housing unit)
    b. Household fields:
        i. HousingType (Possible options: Landed, Condominium, HDB)
2. Add a family member to household
    a. This endpoint lets you add a family member to the household
    b. Family member detail fields:
        i. Name
        ii. Gender
        iii. MaritalStatus
        iv. Spouse (either name of spouse or primary key)
        v. OccupationType (Options: Unemployed, Student, Employed)
        vi. AnnualIncome
        vii. DOB
3. List households
    a. This endpoint lists all the households in the database

    b. Fields:
        i. HouseholdType
        ii. FamilyMembers
            - Name
            - Gender
            - MaritalStatus
            - Spouse (either name of spouse or primary key)
            - OccupationType (Options: Unemployed, Student, Employed)
            - AnnualIncome
            - DOB
4. Show household
    a. This endpoint shows the details of a household in the database
    b. Fields:
        i. HouseholdType
        ii. FamilyMembers
            - Name
            - Gender
            - MaritalStatus
            - OccupationType (Options: Unemployed, Student, Employed)
            - AnnualIncome
            - DOB
5. Search for households and recipients of grant disbursement endpoint.
    a. This endpoint should accept search parameters (eg. Household size, total income) in the URL and return results based on the criteria below.
    b. You may include assumptions about each result in the README.
    c. Grant Schemes:
        i. List households and qualifying family members for **Student Encouragement Bonus**
            - Households with children of less than 16 years old.
            - Household income of less than $150,000.
        ii. List households and qualifying family members for **Family Togetherness Scheme**
            - Households with husband & wife
            - Has child(ren) younger than 18 years old.
        iii. List households and qualifying family members for **Elder Bonus**
            - HDB household with family members above the age of 50.
        iv. List households and qualifying family members for **Baby Sunshine Grant**
            - Household with young children younger than 5.
        v. List households that qualify for the **YOLO GST Grant**
            - HDB households with annual income of less than $100,000.

Disclaimer: All grants mentioned here are fictitious and do not reflect actual grants that are being worked on or implemented by any government ministries.

Optional End-Points (for bonus points)

1. Delete household

    - Remove Household and family members.
2. Delete Family Member
    - Remove Family Member from the Household.