Thanks again for your interest to work in Tranzact!

Below you will find a programming challenge, please use C#, .NET Core 5.0 (or greater) for backend and vanilla JavaScript (**the only allowed external library is jQuery**) for frontend.

We will evaluate your skills in object-oriented, web (HTML) and JavaScript programming.

We expect the code to be production-quality, and can easily be maintained and evolved, not just a barebones algorithm, your submission should be no later than February 11, create public repo on GitHub.com and provide us with a link.

**Premium Calculator**

1. Build a web service, that receives following parameters:

* Date of Birth
* State
* Age
* Plan

And provide a premium as a result, based on the following table.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Carrier** | **Plan** | | **State** | **MonthOfBirth** | **Age** | | **Premium** | |
| Qwerty | A | | NY | September | 21-45 | | 150.00 | |
| Qwerty | B | | NY | January | 46-65 | | 200.50 | |
| Qwerty | A, C | | NY | \* | 18-65 | | 120.99 | |
| Qwerty | A | | AL | November | 18-65 | | 85.5 | |
| Qwerty | C | | AL | \* | 18-65 | | 100.00 | |
| Qwerty | A | | AK | December | 65+ | | 175.20 | |
| Qwerty | A | | AK | December | 18-64 | | 125.16 | |
| Qwerty | B | | AK | \* | 18-65 | | 100.80 | |
| Qwerty | A, C | | \* | \* | 18-65 | | 90.00 | |
| Asdf | A | NY | | October | 21-45 | 150.00 | |
| Asdf | B | | NY | January | 46-65 | | 184.50 | |
| Asdf | A | | NY | \* | 18-65 | | 129.95 | |
| Asdf | A | | AL | November | 18-65 | | 84.5 | |
| Asdf | B | | WY | \* | 18-65 | | 100.00 | |
| Asdf | B, C | | AK | \* | 18-65 | | 100.80 | |
| Asdf | A, C | | \* | \* | 18-65 | | 89.99 | |

\* Wildcard represents criteria that matches anything within the context of use. I.e., for State ‘\*’ means any valid US state, for MonthOfBirth ‘\*’ means any month.

Age and Date of Birth should be validated that they do match each other. I.e., if DOB is 01.01.1958, value 5 for Age is invalid.

If value in the Plan column listed as A, C it means that Premium amount is the same for Plan A and Plan C.

I. e. for 50 years old customer who lives in New York and interested in plan C (or plan A) the applicable rows are:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Carrier** | **Plan** | **State** | **MonthOfBirth** | **Age** | **Premium** |
| Qwerty | A, C | NY | \* | 18-65 | 120.99 |
| Asdf | A, C | NY | \* | 18-65 | 119.95 |

Result of the web service call should be provided in a JSON format like so:

{

    "carrier": "Qwerty",

"premium":"123455.12"

},

{

    "carrier": "Asdf",

"premium":"123413.10"

}

1. Build a web site that consumes the web service created in the previous task; result needs to be printed in a textbox in the page, next to it there should be a drop down with frequencies (Monthly, Quarterly, Semi-Annually, Annually) and next to it two textboxes for the calculated values of Annual and Monthly; these values need to be calculated automatically each time the drop-down value selected changes. Age control should be auto-populated based on Date of Birth.

Bonus points:

* Validate fields before calculation.
* Disable controls if value has not been retrieved.
* Validate only numeric entries on fields.
* Handle all possible exception when calculating values.

Graphical user interface

Description automatically generated

Frequencies meaning:

* Monthly: each month
* Quarterly: each 3 months
* Semi-Annually: each 6 months
* Annually: (each 12 months)

**Example**:

If URL/webservice returns 300.0 and selected frequency is quarterly, that means that 300.0 will be paid every 3 months.

Expected results:

Monthly = 100.0 (300.0 / 3)

Annually = 1200.0 (300.0 \* 4)