**Development Project**

To gain a better understanding of your technical, design and logic skills, we would like you to create a project using the technologies of your choice (preferably .NET, C#, AngularJS, Angular) with the following requirements.

**Task**

As a Member I would like to have an ability to choose various options on the screen So that I can display them on the screen

Screen 1: Develop an UI which accepts the below data.

1. Name
2. Age
3. Date of Birth
4. Next Button To navigate to screen 2
5. Validations preventing submit enabled till all 3 fields filled

Screen 2: Develop an UI which accepts the below data.

1. Occupation (drop down)
2. Sum Insured. (Minimum 1000, max 1000000)
3. Monthly Expenses Total
4. State  (drop down  State 1 and state 2 as values)
5. Post code (Min and Max -  4 numbers eg 1234)
6. Button – Previous to navigate to last screen (Screen 1)  and retain all values
7. Button Calculate to use the Calculation

* Total Value  = (Sum Insured \* Occupation Rating Factor) /(100 \* 12 \* Age)

1. Show the above calculation as a label on screen 2 next to the button  (Label called – Total Value)

The UI provides a below list of occupations

Occupation Drop down values

|  |  |
| --- | --- |
| **Occupation** | **Rating** |
| Cleaner | Light Manual |
| Doctor | Professional |
| Author | White Collar |
| Farmer | Heavy Manual |
| Mechanic | Heavy Manual |
| Florist | Light Manual |

There is a factor associated with each rating as below,

Occupation Rating

|  |  |
| --- | --- |
| **Rating** | **Factor** |
| Professional | 1.1 |
| White Collar | 1.45 |
| Light Manual | 1.70 |
| Heavy Manual | 2.1 |

**Submission**

1. If you have any questions, please do not hesitate to contact Soma Theganahally.
2. The solution does not have to be feature complete, but will serve as the basis for the selection.
3. Add all assumptions and clarifications about solution to the README file
4. Store you code in git repository (Ex: Github or BitBucket). We want to see the evolution of your solution through your commits, so commit early and often.

Once you have completed, you may send the link to the repository and the instructions to run your project to**Mark Thomas**[mark.thomas@dws.com.au](mailto:mark.thomas@dws.com.au)

**Tips from DWS team members who have sat the test previously:**

1. TAL want to see a polished project, not a rushed job. Please spend the time!
2. Unit tests
3. Use of dependency inject where possible
4. Good coding practices -- variable, method, class names, solution structure etc
5. Good understanding of SQL -- mainly SQL server
6. Documentation -- .README file that shows how to get the project running, commit history with messages stating what's in the commit