**Web Coding Assignments**

***Any combination of these environments to use:*** *ASP.NET MVC, Visual Studio 2015, SQL Server 2012/4, MVC, C#, VB, JavaScript, JQuery, Web API…etc.*

***Notes:*** *(This assignment outcome does not determine your selection or rejection for the job. We want to analyse your technical strengths on ASP.NET and weakness or your Web technical capabilities. So relax and enjoy the coding*☺*)*

* *Please try to follow the best practices.*
* *We encourage you to think and implement the below in best manner/approach possible (Try to use any design patterns or any of the framework like MVC).*
* *Consider this as a POC which shows us your capability and expertise level in the Web technology stack.*
* *Timing is not a constraint – take your own time but deliver the best.*
* *If possible try to use the combination of your different web skill sets.*
* *If you feel like using any third party controls then we have Telerik Controls. You can make use of them or any open source library also.*
* *Feel free to use Unit testing if you are willing.*
* *End of the assignment we would like to have a discussion with you on your coding/implementation decisions (like how / why / what …etc.) on which you have to present them with justification.*
* *The basic coding environment is already setup in your box/laptop.*
* Please create your solution structure as: D:\Interview\ASP.NET\<Interview Date>\<Your Name>\<Assignment or Question Name>
* Make a note of Start Time and End Time
* Try not to use internet at first, if stuck ask for help from any one of the interview panel guy. If still not happy then only go for internet.
* The purpose of this round is to check on how hands on you are and what your strong and weak skills are.
* If selected for final round which will happen another day, then you need to run us through your present coding assignment to explain why and what approaches you have considered.

***Question 2:***

1. Design a project management utility with ensuring the following business rules:

Business Rules:

* Allow a user to create Tasks with Start Date, End Date and Description
* Create Resources
* A resource can be allocate minimum a day
* Allocate Resource in tasks, Can allocate more than one person in a task duration
* Resource shouldn’t be allocated to multiple task in same time
* Task shouldn’t have multiple resources in a same time

Tasks:

* Extract resources ideal time
* Display resources involved in a given task
* Display Tasks without resources
* Build an UI to visualize the behaviour (Add Task, Resources and allocate)
* Create unit test cases to validate logic
* Adapt best practices and coding principles
* Create an Architecture/Use Case/Flow diagram

Example:

Valid Scenarios

Task 1: Leana (1st Jan), Dewey (2nd Jan to 7th Jan)

Task 2: Leana (2nd Jan to 4th Jan)

Invalid Scenarios

Task 1: Leana (1st Jan), Dewey (1st Jan)

Task 1, 2: Leana (1st Jan)

|  |  |  |
| --- | --- | --- |
| Tasks 1 | 1-Jan-15 | 7-Jan-15 |
| Tasks 2 | 1-Jan-15 | 4-Jan-15 |
| Tasks 3 | 1-Jan-15 | 4-Jan-15 |
| Tasks 4 | 3-Jan-15 | 6-Jan-15 |
| Tasks 5 | 5-Jan-15 | 8-Jan-15 |
| Tasks 6 | 6-Jan-15 | 9-Jan-15 |
| Tasks 7 | 10-Jan-15 | 13-Jan-15 |
| Tasks 8 | 31-Dec-14 | 1-Jan-15 |
| Tasks 9 | 31-Dec-14 | 1-Jan-15 |
| Tasks 10 | 3-Jan-15 | 4-Jan-15 |
| Tasks 11 | 8-Jan-15 | 9-Jan-15 |
| Tasks 12 | 13-Jan-15 | 22-Jan-15 |
| Tasks 13 | 31-Dec-14 | 9-Jan-15 |
| Tasks 14 | 31-Dec-14 | 9-Jan-15 |
| Tasks 15 | 3-Jan-15 | 12-Jan-15 |
| Tasks 16 | 8-Jan-15 | 11-Jan-15 |
| Tasks 17 | 13-Jan-15 | 16-Jan-15 |
| Tasks 18 | 31-Dec-14 | 3-Jan-15 |
| Tasks 19 | 31-Dec-14 | 9-Jan-15 |
| Tasks 20 | 3-Jan-15 | 12-Jan-15 |

