**Assignment: Casual Employee role-based salary calculation system.**

**Scenario:**

1. Management often needs to hire casual employees to lessen the workload over the festive periods.
2. You are asked to assist the management team with a casual employee role-based salary calculation system.
3. Management assigns employees to tasks and every task has its own duration.
4. An employee can be assigned multiple tasks but cannot work more than 12 hours a day.
5. Casual employees’ salary is calculated based on the role they were assigned to.
6. Casual employee roles are categorised into the following levels and each role has a Rate per hour:
   1. Casual Employee Level 1
   2. Casual Employee Level 2
7. Management needs to be able to capture total hours worked for each employee for a specific day.
8. Changing the hourly rate or changing the casual employee role should not affect previously captured hours.
9. Management needs to be able to do the following:
   1. Create and Edit Casual Employees and assign a Role.
   2. Capture/Store Employee Profile Image.
   3. Create and Edit Employee Roles.
   4. Change Employee Role rate per hour.
   5. Create Tasks.
   6. Assign Casual Employees to one or more Tasks.
   7. A Task can have multiple Employees assigned.
   8. View total due to Casual Employee over a specific timeframe.
   9. Capture hours worked for an Employee for a specific Task.
10. Employee needs to be able to login and view their salary calculated.

**Instruction**

1. Create a .Net Core 3.x Web Application using C# and MySQL or SQL Server database for data storage.
2. The frontend should use MVC.
3. Data should be accessed via a WebAPI that uses an ORM (EntityFramework).
4. Visual Studio 2015/2017/2019 community edition can be used.

**Expectation:**

1. Authentication and Authorization
2. Server-side table processing
3. Unit of Work
4. Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | **Not Attempted** | **Not Achieved** | **Expectations Achieved** | **Exceed Expectations.** | **SCORE** | **COMMENTS** |
| 1: Introduction. Introduce this database application. Provide the name of your application and describe the different features it entails | 0 | 5 | 10 | 12 |  |  |
| 2: Database design specifications. Provide the details of at least three tables as well as a code snapshot of the fields included in each table. | 0 | 5 | 10 2 tables with details and code snapshot. | 12 3 tables with details and code snapshot. |  |  |
| 3: User interface design. Provide the details and snapshots of the website layout, design and the views/pages used. | 0 | 5 | 10 1 page – details and screenshot | 12 2 pages with details and screenshot |  |  |
| 4: Database manipulation. Provide the details of the data manipulation and code required. Provide explanations of the application code that was used to manipulate the data. The code should include comments. | 0 | 5 | 10 Add, edit and delete manipulation of one table, including code. | 20 Add, edit and delete manipulation of more than one table, including code. |  |  |
| 5: Application – How did you test the application? What were the results and findings and suggestions? | 0 | 5 | 10 Explanation but not tested. | 14 Explanation and tested |  |  |
|  |  |  |  |  | **TOTAL: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/70** | |
|  |  |  |  |  | **PERCENTAGE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_%** | |