DEVELOPER - MINI PROJECT

Authors: Xcallibre

Year: 201ф

Version: H

TABLE OF CONTENTS

INTRODUCTION

This document will serve as a short specification for a mini project for applicants wishing to fulfill a role as a developer at Xcallibre (Pty) Ltd. The intention is to provide all the relevant details to be able to complete the project in the suggested time.

All instructions set out in this document should be closely followed to ensure correct deliverables. Any assumptions made based on potential ambiguity within this document, should be noted when submitting the project.

The goal is to verify that **some** of the skills required for the position are tested. In some cases the required skill might not be present, but the project is of a very simple nature and the requirements can still be met by means of basic searching and implementing.

Please also note the last section is this document which will refer to some guidelines and other areas that will be assessed.

PROJECT DETAILS

This project will be split into two small applications. The first being a lightweight WPF (Windows Presentation Foundation) project, and the other being a very small website. Both will be described in detail below.

WPF CLIENT APP

The WPF client application will serve as a starting point for capturing some data. Each row of data would need to be displayed and then exported. The exported data must be marked as such and the exported file will be used in the following application.

Core Requirements

1. WPF executable
2. DataGrid or List displaying captured data
3. Entry form (On the same form or as popup)
4. Field for choosing file saving location
5. Export button

Specific Details

1. Application must provide the user with a form in which the user will type in some data
2. The data must then be stored somehow (DB could be used but could be overkill)
3. Must be able to add several rows of data, hence the DataGrid or list to show this
4. User must then be able to select an export path and file name
5. Data must be exported into an XML format (Your own design). This format MUST NOT be the standard save from a DataGrid in ADO.Net
6. Once data has been exported, the rows present must be marked as "Processed". More rows could be added to create a second / n number of XML exports

WEB APPLICATION

The ASP.Net web application will take the export from the WPF CLIENT APP and import the data into a local database, allowing the users to update some of the information.

Core Requirements

1. ASP.Net web application
2. Ability to upload a text file
3. DataGrids showing the data
4. Inline grid editing and deleting of data
5. Local database file (App_Data folder)

Specific Details

1. The web application must have at least two pages. One for uploading the XML files and another displaying the uploaded data for manipulating
2. Uploading the file can be done in memory or into a folder within the web application
3. File must be read and data inserted in one or more tables, depending on your database design

4. DataGrid displaying the uploaded data with the ability to edit the data in the DataGrid

DATA

No sample data will be provided nor will an indication of what type of data be captured and imported. The following data types must be included as a minimum:

1. String
2. Boolean
3. Integer
4. DateTime

Please note that each record is required to have NO LESS than 7 fields.

SQL

Please use the [AdventureWorksLT2008R2] database for this test. Please save your work using [projectname].sql as the file name

Write a SQL statement to identify which PurchaseOrder records ([SalesLT].[SalesOrderHeader] table) has a SubTotal field that does not equal it's detail tables ([SalesLT].[SalesOrderDetail] table) sum of linetotal field. Please include the following columns in the result set:

1. Salesorderid
2. SubTotal from the header table
3. Sum of linetotal from the detail table
4. The difference between 2 and 3

This needs to be done in a single SQL statement.

Please create tables to record the following: AthleticsClub, Athlete, Events, AthleteEvents.

This can be done using as many tables as you require.Include in the design all Primary and Foreign key constraints to achieve the following:

1. Enforce unique AthleticsClubs
2. Enforce unique Athletes
3. Enforce Unique Events
4. Many athletes can belong to 1 Club, but the Club must exist.
5. An athlete can only belong to 1 club.
6. An athlete can take part in many events but the events must exist.
7. An athlete may not enter the same event more than once.
8. All entries in the EventAthletes table must be existing athletes (must exist in the athlete table).

RULES, GUIDELINES

Below is a list of the rules / guidelines to be followed.

1. Project / Solution names MUST NOT include the candidates name or initials. Nor should any comments be made in the code that will reveal the name of the candidate. The reason for this is that the team assessing your code will be the team that you will be working with and ensuring that the code is anonymous will enable better / fairer assessment thereof.
2. All project source code should be Archived (ZIP, RAR) and email to careers@xcallibre.com. Please ensure that the BIN folder is cleared before archiving it. This will ensure that file sizes are kept small
3. Ensure that the local database is clean before submitting the project.
4. All projects must be submitted via email a week after you have received it by close of business.
5. Assessment will take the following factors into consideration:
    a. Code abstraction
    b. Code re-use
    c. Best use of OO
    d. UI design (But please do not spend all your time on this, less important that the rest)
    e. Code comments (Again comment where feasible and not random comments all over)
6. .Net 4.0 or .Net 4.5 could be used to complete the project.
7. Where advanced features are used, an explanation for the reasoning should be provided. The reasoning should take into consideration future growth and a potential high concurrency rate.
8. Feedback will be provided on all submitted projects, whether successful or not.