Working for Powerfront

# ASP.NET TEST

## Virtual Data Structure

**Prepared by: Greg Platt**

1300 795 520

Level 10
470 Collins Street
Melbourne
VIC 3000

Level 3
104 Commonwealth Street
Surry Hills
NSW 2010

Powerfront™
online success

# Executive Summary

## Overview

Thank you for taking the time and performing this test towards your potential employment as a programmer with Powerfront.

The test will allows us to determine the following:

- The questions you ask before starting the test
- Your understanding of the virtual data structure
- How quick you can program
- Quality of programming under pressure
- The way you construct your code
- How well you test

We would like to limit this test to 4-8 hours.

This test represents an urgent job given to you in Powerfront. Your honesty, the ability to work under pressure and the way you communicate is very important to us.

Once you have completed the test:

- **Please provide us with a URL to see work your LIVE on the internet.** If you don’t know where to publish your work you can try aspspider.com or brinkster.com or somee.com.
- Please be prepared for some questions from us
- Let us know what you could not achieve and why
- Email the completed source code and database to careers@powerfront.com

This is a simple test and does not represent the complexity of the Powerfront CMS.

Regards,

Greg Platt
CTO

# Task 1: Database

You do not need to program anything in Task 1 – simply database work. We prefer you to use MS SQL Server but if you don't have that capability please use Access. Provide us with the database file (*.mdf | *.ldf) or the database script (*.sql) to create your database.

Please create a database that will allow **the storage of data in virtual tables**. When we say virtual tables we mean that a table and fields (columns) are stored in an object oriented data structure and not in the traditional way such as tables, columns and fields.

In this test we will ask you to create a simple virtual table called "Customer". Each customer has 2 fields "Name" and "Age". We would like to store the following data in the virtual table:

| **Name** | **Age** |
|---|---|
| Hadar | 47 |
| Greg | 38 |
| Michael | 45 |

In Powerfront we call the virtual table row data 'Objects' and the fields or columns 'Properties'.
'Properties' belong to an Object Type (i.e. in this case 'Customers')

To simplify the test, please note:

- □ Fields in your virtual table can be text type only
- □ No validation is required

**Hint**

For your database structure you will need to create the following tables and relate these tables to each other:

- □ Object Type
- □ Object Property
- □ Object Data

Object Type Table is only to store type (i.e. Customers)

Object Property Table is only to store properties of the Object (i.e. Name, Age, City)

Object Data Table is only to store data (i.e. John, 25, Melbourne)

**Important**

Make sure you understand the above instructions correctly as this is the basis for everything else you do on this test.

In your code do not alter the database structure or the table structure.

Please check your published work. Make sure your live site is working properly as this is the first thing that we look at when assessing your test. If your published work is not functioning properly then you will fail the test.

## Task 2: Maintenance Page

Now that you have created the database design you will need to create a maintenance ASP.Net page to manage the data. The page will need to allow the following:

1. Display a list of customers and ages
2. Allow the user to edit and save existing records
3. Allow the user to delete a customer record
4. Allow a user to add a new customer

## Task 3: Add Property Page

Now that the user can maintain the data, we will ask you to write an ASP.Net page that will allow the Administrator to add a new virtual column to the Customer's table. In Powerfront terminology - to add a new property to an object type.

To simplify the test - the ASP.Net page will only allow adding a new property. However once the property is added, we would like to see the maintenance program adopting the new property immediately and being able to add/edit data in that field.

# Task 4: Additions

Should you have additional time please perform validation on user data entry and improve the look and feel and usability of your application. Create a nice-looking and user-friendly interface for your application, and make sure all users of the application can use it easily and intuitively.