**Invoice Cloud – Software Engineer Challenge**

Thank you for participating in the Invoice Cloud interview process. For the final part of this interview process, we would like for you to review the following 4 tasks. These tasks touch on technologies that are used daily to create and/or maintain Invoice Cloud systems. Each task stands on its own but are similar in function. The difference lies in the technology used.

We would like to get an idea of what you are capable of in relation to what you will work on with us, and you will do that by completing these tasks. Many have completed the challenge, and many have not. We encourage you to take the challenge!

As a final note, use your best judgement and common sense if you have any questions. The expectation is if something isn’t clear, do what you think needs to be done.

**IMPORTANT TIPS:**

1. Our platform consists of mostly VB.NET, with use of ADO.NET for database interaction. While we love Entity Framework, we don’t use it here. Please make use of ADO.NET. **C# IS OK.**
2. Please put each Task into its own folder so it’s easier for us to review.
3. Please make sure your solution for each task works “out of the box”. One common thing we run into is the MDB file is stored in a hard-coded location on your computer. When shared with us, that directory doesn’t exist, so we must modify your solution, so it knows where the MDB file is. Make use of relative paths to avoid this issue.
4. The task descriptions are basic by design. You can differentiate yourself by filling in the gaps. For example, I don’t describe what your form should look like when creating a new Customer. This is a great opportunity for you to make your form look good. This is also a great opportunity for you to create your own customer experience. How would you design it if you were building this for your customers? Please take advantage of this opportunity.

**Your database**

We have attached a CustomersDb.mdb file that will serve as your test database. This file contains the Customers table that is referenced in the tasks below.

**Note. I’m OK with you using C# if it’s easier. Keep in mind most of our platform is in VB.NET. This may be a good opportunity to pick it up and apply it.**

**Task 1 - VB.NET, ASP.NET WebForms**

Please create an ASP.NET WebForms project using VB.NET that will support basic CRUD operations to manage the Customers table.  Add support for viewing, adding, editing, and deleting Customers.

**Task 2 – JQuery, VB.NET, ASP.NET WebAPI**

Please create an HTML web site that will support basic CRUD operations to manage the Customers table. Add support for viewing, adding, editing, and deleting Customers. The web site should interface, using JQuery / AJAX calls, with a VB.NET ASP.NET WebAPI service to support pulling and updating information from the Customers table.

**Task 3 – Angular, VB.NET, ASP.NET WebAPI**

Please create an Angular application that will support basic CRUD operations to manage the Customers table. Add support for viewing, adding, editing, and deleting Customers. The Angular app should interface with a VB.NET ASP.NET WebAPI service to support pulling and updating information from the Customers table.

**Task 4 – VB.NET, ASP.NET MVC**

Please create an ASP.NET MVC project using VB.NET that will support basic CRUD operations to manage the Customers table. Add support for viewing, adding, editing, and deleting Customers.