**Code Challenge**

Part I:

While designing this application, I started with the database. During this process, I created a new database and added the table to it. Next comes the project, I created a new MVC 5 project added the database context, created the class, added the annotations to the class and added the dbset. From here I went on to create the user interface. I added a controller then added a view. Once all of these items were added I began to add label, text boxes and validators to the form. During this stage I was not concerned about the look of the form, I was mainly concerned about the functionality. From here I got all of the fields on the form working and submitting to the database. After that I made the form look good. I downloaded some assets from Code Authority’s website to use in my presentation.

Here is where I deviated from the coding challenge. I wanted to make a great impression to the developers reviewing the project and also showcase my talents. So, I created a WEB.API project for the form to post to. This process was much like the process for creating the form. Adding the context, the class, annotations and dbset. I added the ability to create a new record in the database, pull all records from the database and pull just one from the database. Then tested the functionality on my localhost. After all of that I logged into my AWS account and copied the WEB.API to a cloud server, tested it to make sure all the functionality was working and opened the port to be accessible to the internet. I then made the contact form to POST to the WEB.API.

I got to thinking how cool would it be to create a Xamarin forms app, so I did. I realize this is not part of the coding challenge but the Corporate Recruiter, Ashlyn mentioned one thing she liked about my resume was the fact that I had Xamarin experience. So I fired up Visual Studio and created the Xamarin app. First, I had to add the navigation base form. Every app needs navigation. Then I added a dummy login screen. I could have added login to the WEB.API but I decided not to, therefore no userID or password is required. Then I set out to display the data on a list view. This was the trickiest part of the programming challenge. I had to figure out how to use async/await on loading the form. So, I used it in the OnAppearing method. It seems to have worked out good. Just a side note I prefer Telerik List view over the one provided by Xamarin Forms. After I got the list view population I wanted to show the detail of the item. So I created a detail page. Clicking the list view item opens the list view detail page.

Sources:

www.tackoverflow.com

https://captcha.com/doc/aspnet/captcha-getting-started-nuget.html?utm\_source=nuget&utm\_medium=nuget&utm\_campaign=4.1.0.0#integration\_mvc

Part III:

To create the Entryway challenge I would start off with HML options. I would group them by category such as Paint, Carpet, Floors. I would put them in divs for each line. I would change the normal option selection to be a square. Trying to style the option might be tricky I would first start by just making the height and width 100px or so then color with css. If that did not work I would create a div around each option and style it with css. For the click I would just create a click method with javascript or jquery on document load that would handle the click of the button.

All source code is on GitHub here:

https://github.com/envisiondata/Code-Authority.git