Read the Story Behind the Code See the Page in Action

The Ntara Team Search Project

See the Page in Action
Welcome to the Story Behind the Code

Initial Project Definition

After the staff at Ntara provided the opportunity to interview in person twice to help determine if I might be a good fit for the role, I was assigned the following sample project.

1. Create a single web page in asp.net C#
with a textbox and a button that, when clicked,
searches a given dataset and returns the entire record of results on screen

2. Include a drop-down that defaults to all columns but can be selected to search only one column

3. Provide error checking where appropriate

4. Please use a technology framework like Angular, React, or Vue to help showcase how you'll interact with the frontend.

Initial Skillset Matchup

My experience includes coding in multiple languages, but as of the project start date my C# skills were a bit rusty. I have also never singlehandedly connected a database to a Razor page. Technologies such as Angular, React and Vue are new.

Given that fact, part of my planning had to include time researching, learning and brushing up on multiple items.

Planning Timeline

Tuesday, February 18
Received the Project
Received email instructions for the project. Opened them, sent a response confirming that I had received the email, and looked over the project as a brief preview. I let Ntara know that I was planning to go more in depth on the project over the weekend.

Tuesday, Febrary 18 through Friday, February 21
Focus on Current Job
During the week, I continued to focus on getting an inRiver PIM ready for implementation at my current full time job.

Saturday, February 22
6:00 am - 6:00 pm
Renew Required Skillsets
I set aside one day to catch up on the technology required. Researched Razor and database implementation. Reviewed C# syntax. Built a couple of sample database projects using Visual Studio.

Sunday, February 23
6:00 am - 6:00 pm
Focus on the Ntara Project
Wrote and implemented code specifically for the project.

Skillset Acquisition

On Saturday, I focused on aspects of Visual Studio and C# that would directly relate to the project.

After browsing a variety of articles, I found this tutorial to be helpful with the task at hand. During the explanation of each step later, I have added links to other pages that I found helpful.

Razor Pages with Entity Framework Core in ASP.NET Core - Tutorial

This tutorial explains how to create a Razor page and connect it with a database. I created three practice pages using these instructions. It even explains how to introduce a search function, which is called out in the project definition.

Ntara Project Logistics

Once I began working specifically on the Ntara project on Sunday, the first thing I did was finalize and write down the mapping for each general task that I imagined would be required to complete the project. Here is a general list followed by specific details about implementation.

1. Set up the Razor Project

2. Migrate the data into the database

3. Build a page to display the data as a table

4. Introducing search capability into the page

5. Modify the search to change behavior based on a dropdown

Setting up the Razor Project

For this task, I was able to begin by adding the Scaffolded Item entitled *Razor Components using Empty Framework (CRUD)* to the page. The automatically generated pages were then available for modification for the project.

The methodology I used is Documented Here.

Moving the CSV Data Into the Database

Once I had the setup, I modified the code in this example to seed the database. Since the original csv had a lot of rows, I wrote the code that I introduced to the new Teams class by using Excel. Specifically, I concatenated data fields from the original csv with the appropriate repeated code. I was able to then copy these concatenated values directly into the code. While it worked, this section of code remains formatted as pasted with limited line breaks. I left it that way because I found it easier in this case to scan differences in the rows without multiple lines to scroll through.

The project definition did not include specific rules about data types for the columns in the csv. Given the fact that while some of the data would work as a double or an integer, other data existed that would not fit that type. Without further direction from the end user, I am unsure whether the text is meaningful in the context of the specific gameday presentation. For example, NULL could mean something to the broadcasters that helps them with their announcement. Probably not, but verification from them might change my approach to the data types. From the standpoint of the user, the fact that all values are stored as strings does not impact the usability.

Building a Page to Display the Data as a Table

At this point, I already had pages available that were generated by the *Razor Components using Empty Framework (CRUD)* Scaffolding. I modified the index.cshtml file in the Teams directory with a layout that would work with the project data.

Introduce Search Capability to the Page

The methodology I used for accomplishing this is documented here.

Modifying Search to Respond to Dropdown

Once I was at this step, I thought with a few more modifications I was home free. Not so fast. This functionality was actually one of the most challenging of the entire project. It may have been because I was getting tired at this point, but it took some thought (and some exception errors) to get it set up. I ended up creating an *if, else if, else Statment* that worked.

Some Things I Learned (so far)

Setting up this project was very interesting to me. It was a challenge and it opened up a lot of questions about how things work. Of course, not having contact with the user that is requesting the page makes developing a solution difficult. I could then ask if the NULL values should be removed from the visible data and whether the columns should be sortable. User feedback on the page would be extremely helpful.

On Saturday, the learning curve was more challenging, but I was able to explore more topics and even try some things that weren't directly in the project scope. I took a look at the broader perspective of what can be done, and a few fun ideas even came up. For example, I have a short term plan to create and publish a database of musical bands as a bit of a practice and exploration. The functionality sparked a lot of ideas, but as of the end of day Saturday those were put temporarily on hold. The task at hand was fulfilling the specific requirements of the Ntara Project on Sunday morning.

On Sunday, I was more used to the environment. Instead of exploring I focused fully within the parameters of the project. Given the fact that my focus would have to return to work on Monday, my deadline was 6:00 pm Sunday. I expected to face more issues than I did early in the day, but surprisingly things were coming together as expected. As I mentioned earlier, I did begin experiencing issues with the dropdown task in the search requirements, but now can look back and see the solution easier. This leads me to believe that fatigue may have been in play since I had been coding for a while at that point. The good news is that I was able to complete a working version around 4:00 pm on Sunday. I looked things over and made a few updates until 6:00 pm. I then took my notes and translated them into the document that you are reading.

Time Spent Coding

Sunday - 12 hours (6:00 am - 6:00 pm)
Saturday - 12 hours of research time that was not necessarily directly related to the solution (6:00 am - 6:00 pm)

See the Page in Action