1. My proudest professional achievement was the first project that I worked on for the company that I am at now. It was a huge api project that took information from a customer’s api and entered it into our database to serve the 20+ sites that we have of this client. First, I needed to figure out how our sites were using the data so that I could create a schema that would serve those sites the best. I then created an application that grabbed the data and then using dependency injection wrote the data to the new database that was optimized for our sites and the old database and also left it open for new implementations that might be needed in the future. The application was architected and implemented beautifully, however, that was not why I am most proud of this application. The reason why I have the most pride for this application was that I owned every aspect of it. From project management, budgetary concerns, keeping track of changes to scope, to implementing it, I did it all and it turned out to be the most profitable project for the company for that year. That is what makes this project my proudest professional achievement.
2. An article like the one that I read recently that I really liked can be found here: <http://news.mit.edu/2019/toward-artificial-intelligence-that-learns-to-write-code-0614>. Unfortunately, I could not find the exact article, but basically it talked about how machine learning and artificial intelligence is getting to the point where it can write programs. Currently it is at the stage where it is a supplement to current developers in helping them to write better algorithms, however, it is currently only held back by computational power and Moore’s law is still in effect. This will make it so that in the near future, developers may be out of a job. Combine this technology with natural language processing and pretty soon you will be able to tell a computer to create a program to do something for you and it will just create one. This is the future, but what will happen to the entire current workforce of developers that will be there once this becomes the mainstream way of programming?
3. In our country, to be able to go to the doctor and not spend a lot of money doing it we have something called insurance. For a monthly charge, people can get insurance and then the insurance companies pay the doctors. Doctors don’t really like having to deal with the insurance companies because they put rules in place that make it hard for the doctors to get their money because the insurance companies want to keep more of the money that they have coming in from people. However, Availity makes it simple for doctors to charge the insurance companies and have a better chance of following the rules set out by the insurance companies. Availity also makes it so that there is a record of what happens between the doctors and the insurance companies so that even if the insurance company refuses to pay for whatever reason, the doctors can try again and maybe get paid the next time.
4. See the Code Validator folder in the repo.
5. See the Registration folder in the repo.
6. See the CSV Parser folder in the repo.
7. SQL Statements
   1. SELECT \*

FROM Customer

WHERE LastName LIKE ‘S%’

ORDER BY LastName DESC, FirstName DESC

* 1. SELECT c.CustID, c.FirstName, c.LastName, COALESCE(x.Total, 0)

FROM Customer c

LEFT JOIN

(SELECT c.CustID as CustomerID, c.FirstName as FirstName, c.LastName as LastName, SUM(ol.Cost \* ol.Quantity) as Total

FROM Customer c

JOIN [Order] o ON o.CustomerID = c.CustID

JOIN OrderLine ol ON c.CustID = o.CustomerID

WHERE o.OrderDate > DATEADD(MONTH, -6, GETDATE())

GROUP BY c.CustID, c.FirstName, c.LastName) x ON c.CustID = x.CustomerID

* 1. SELECT c.CustID, c.FirstName, c.LastName, COALESCE(x.Total, 0)

FROM Customer c

LEFT JOIN

(SELECT c.CustID as CustomerID, c.FirstName as FirstName, c.LastName as LastName, SUM(ol.Cost \* ol.Quantity) as Total

FROM Customer c

JOIN [Order] o ON o.CustomerID = c.CustID

JOIN OrderLine ol ON c.CustID = o.CustomerID

WHERE o.OrderDate > DATEADD(MONTH, -6, GETDATE())

GROUP BY c.CustID, c.FirstName, c.LastName) x ON c.CustID = x.CustomerID

WHERE x.Total > 100 AND x.Total < 500