\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*GOD first \*

\*Name: Dustin Ledbetter \*

\*Last Updated: 1-9-2020 \*

\*Job Start Date (Randstad Temp): 7-5-2018 \*

\*Job Hired Date (Full Time Employee): 5-16-2019 \*

\*Purpose: Work Stats and Projects worked on at Communicorp \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##---------------------------------------------------------------------------------------------------------------------------------------

General Stats:

Randstad Title: Web Designer 7/10/2018 – 5/16/2019

Communicorp Title: Application Developer Analyst 5/16/2019 - current

Note: My actual job tasks build me as a Full Stack Developer

Number of notable Projects that I've worked on and can share currently: 23

Number of storefronts I am responsible for working on currently: 50+ live sites

Types: A mix of Pageflex, Able Commerce, and custom sites

Time/Project tracking software used currently: Formstack, Harvest, and Fieldglass

Source Control currently: Gitbucket and Github

##---------------------------------------------------------------------------------------------------------------------------------------

##---------------------------------------------------------------------------------------------------------------------------------------

Extra Learning while at Communicorp:

After completing the training courses below I decided to shift my focus to developing my portfolio and personal website. I have used it to test new concepts such as CSS animations and to enhance my knowledge of other core skills in web design. It is still a work in progress, but will be completed in 2020.

FreeCodeCamp.org: (Developer certifications representing approximately 300 hours of coursework)

1. Responsive Web Design Certification: 8/10/2018

2. JavaScript Algorithms and Data Structures Certification: 11/14/2018

LinkedIn Learning: (Certificate of Completions for completed courses)

1. Programming Foundations: Fundamentals Course: 10/29/2018

2. Web Development Foundations: Full-Stack vs Front-End Course: 10/29/2018

3. Guy Kawasaki on How to Rock Social Media Course: 10/29/2018

4. CSS Essential Training Part 1 Course: 11/7/2018

5. CSS Essential Training Part 2 Course: 11/8/2018

6. CSS Essential Training Part 3 Course: 11/13/2018

7. HTML Essential Training Course: 11/13/2018

8. Programming Foundations: Databases: 11/13/2018

Books:

1. T-SQL Fundamentals Third Edition By: Itzik Ben-Gan [Worked through the book completing all course work for each chapter] 9/23/2018

Aflac Training Courses (Certificate of Completions for completed courses):

1. Information Security and Privacy Training Course: 5/21/2019
2. Anti-Fraud Training Course: 5/21/2019
3. Anti-Money Laundering Training Course: 5/21/2019
4. OSHA Fire Prevention and Emergency Action Awareness Training Course: 5/21/2019
5. Financial Exploitation-Elderly and Adults with Disabilities Training Course: 5/22/2019
6. Phishing Training and Assessment Course: 5/29/2019
7. OSHA Workplace Violence Awareness Training Course: 6/7/2019
8. OSHA Slips, Trips, and Falls Training Course: 6/7/2019
9. OSHA Office Ergonomics Training Course: 6/7/2019

##--------------------------------------------------------------------------------------------------------------------------------------

//------------------------//

// Notable Projects //

//-----------------------//

\*\* This list is in chronological order based on when I started to work on these.

\*\* Not all projects I worked on are added to this list

[1]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Learning template Creation, Updating, and Tracking for Aflac Insurance Groups

--Start Date: 7-19-2018

--Purpose: To learn the front side process for Aflac Group

----------------------------

-Technologies Used:-

----------------------------

Pageflex Studio, Adobe Acrobat Pro DC, Visual Studio utilizing C#, Excel, MSSMS (Microsoft SQL Server Management Studio) for creating and running scripts, ASP.NET

------------------

-Description:-

------------------

I spent several weeks working with the front side of Aflac Group who are responsible for setting up templates for processing Insurance group claims. I did this to better understand the site I would be working closely with behind the scenes.

-----------------------

-What I learned:-

-----------------------

-I completed several live cases from start to finish during my time with them learning how to create, edit, and setup templates for processing large groups of insurance claims.

-I learned how the front end process worked, which helped in working on the back end to deliver an easier and more efficient process for the front end team.

-I learned how to setup scripts and queries that would be easy for the team to use when looking for more information on groups.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: I was learning a completely new process and utilizing tools I had never heard of.

--Solution: Reached out and asked as many questions as needed till I felt comfortable on my own. Making sure to double check everything I did as I went.

[2]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Working on the back side for client services

--Start Date: 8-1-2018

--Purpose: To develop and continue to automate front end processes

----------------------------

-Technologies Used:-

----------------------------

Pageflex Studio, Adobe Acrobat Pro DC, Visual Studio utilizing C#, Excel, MSSMS (Microsoft SQL Server Management Studio) for creating and running scripts, ASP.NET

------------------

-Description:-

------------------

I began work on building and updating queries to help the front end team. I was in the process of researching and testing a few ideas for new features when I was shifted to a new project.

-----------------------

-What I learned:-

-----------------------

-I learned how the site was structured and what all processes and steps were working together.

-I learned a lot about the way the developer before me and how he set his code up.

-I learned how to utilize the company's gitbucket account while working on this project.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: Found a glitch in the process for the front end team while working with them.

--Solution: Went through the process of addressing the issue and working to get it resolved while creating the proper documentation of what I did.

--Complication: The original developer who had left the company and whom I had been hired to replace returned and took the project back over.

--Solution: I worked to bring him up to speed on any changes I had made, I then moved on to my next project dealing with tax calculations.

\*Note (2/6/2019): This training has come in handy still as I have served as a backup for the developer whenever he goes on vacations or is out.

[3]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Item Exporter

--Start Date: 8-20-2018

--Purpose: An application to export data from a database to an excel file

----------------------------

-Technologies Used:-

----------------------------

MSSMS (Microsoft SQL Server Management Studio), Visual Studio utilizing C#, Excel

------------------

-Description:-

------------------

I created a published C# deployable application that allows users to export data from several databases.

The user selects which database, which stored procedure, and hits submit. This submit button fires of the process that connects to the db, retrieves the specified data requested, fills it into an excel file, and prompts the user for where they would like to save it. The user can then exit or rerun the process again with the same or different criteria. It is also setup to send emails whenever the process fails to the developer in charge.

-----------------------

-What I learned:-

-----------------------

-I learned to work on a project that requires months of back and forth with the user and team using the application during development.

-I learned how to access the db and stored the retrieved data into an excel file.

-I learned how to setup specific roles and accesses for use of the stored procedures on the db.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: There were complications in several approaches in the development process of learning how to store the retrieved data from the db into an excel file.

--Solution: I placed the data into a better format first and then tried to save it into an excel file.

-- Complication: Because multiple people have access to the db I was working with: The stored procedures were removed and access revoked.

--Solution: Created a script to quickly read stored procedures and give accesses if it happens again.

[4]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Avalara Tax Extension

--Start Date: 9-18-2018

--Purpose: An extension to add Avalara tax calculations to all storefronts

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Avalara Tax Client 2018-2019 (Including the use of their REST API's), Pageflex, MSSMS (Microsoft SQL Server Management Studio), Postman

------------------

-Description:-

------------------

I created an extension for Pageflex storefronts that runs a process to retrieve the tax amount for an order on the storefront when a customer is at the checkout stage for purchasing items. This extension was used to replace an older process that required a lot of manual touches to the storefronts and the use of running of scripts every month. This was going to be an issue with the newer age of tax regulations, so my extension was tasked for creation.

My extension interrupts the storefronts process and instead sends information the user provides to Avalara through their API that returns the tax to be added for the order based upon our nexus and the new laws that are in effect nationwide. I ended up with 10 versions of the extension after I finished developing it. Each version builds off the previous and is more complex in what it retrieves and how the data is processed. The latest version does several important things: It has several checks built in to ensure we interrupt the storefront's process at the right time, It reaches out to the storefront and it's database to pull information for the user and their current order, It sends this information to Avalara and gets it back as needed, It sends out an email upon any failures during the process with detailed information about what went wrong and where in the process it failed, It is sending data to a log file I set up and to the storefront's logs as it processes the different steps, It has an admin page setup on the storefront that allows an admin to change any logins, or database information without having to ever change or touch the source code.

This extension has been added to every storefront at the company I am currently working at: Communicorp. [This is on 40+ LIVE sites]

Side Note:

I built a user start guide of my own to train new developers after or under me on how to use Pageflex and to create their own extensions. This includes a 4 part course using the first seven of my extensions and walking the new developer step by step through how to create their first one and then to build it up into the complex one I finished with. It has well commented code, accompanying walkthroughs and side notes for things I tried and explanations for reasons I did this or that. It should get them up and running in a far shorter time than it took me to.

-----------------------

-What I learned:-

-----------------------

-I learned a lot during this whole project starting with the major pieces such as how to make extensions for Pageflex, how to incorporate C# together with Pageflex and also with using Avalara and their amazing API.

-I now have a very good understanding of Pageflex and how its storefronts and code are setup and how to utilize what I know to setup extensions, web apps, and internal processes to a great extent.

-I learned how to solve issues and to find solutions with little to no help. The documentation for Pageflex is extremely vague and sometimes nonexistent as well as their forums and tips site are no longer hosted online. I relied a lot on sifting through other extensions and code to find pieces I could use and also from using good judgement to test ideas and try different techniques out till I found/made the solution I was needing.

-I learned how important setting things up early that will help later such as the debugging logs and commenting my code as I go. I am thankful for my college professors drilling these core elements into my education.

-I learned how to navigate through a massive database better and to back in to the tables I need when the data is spread out amongst many tables.

-I learned and added more solidarity to my C# knowledge.

-I learned how to setup a detailed and useful walkthrough and user guide that I can use as a template on any future projects

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: When I was given this project the person who was above me and was to train me left for another job before he could teach me anything I needed to know, The Pageflex support and user guides I had to start with were slim to non-existent, and there was almost nothing online to help in learning how to create your own extensions and to learn how to code behind the scenes on Pageflex.

--Solution: I dug through any and all resources we have here at Communicorp and built up a collection of anything I found that might be useful: including source code for three or four other extensions, some user guides that Pageflex gave us, but are extremely vague, and a few other small documents here and there. I then slowly began to tear apart the previous extensions and to build small ones of my own. After the first 4 or 5 extensions I started to have figure out how the system for Pageflex worked more and was able to get my extensions to do more and more.

--Complication: The second biggest complication was in realizing to use Avalara I had to learn and understand how the new tax laws work for 2019. This includes understanding Nexus and when an online store collects tax for an order.

--Solution: I spent time reading up on new tax laws and looking through Avalara's website to understand how the new tax process works. I also learned how to find and see how a state's taxes are put together: state taxes, city taxes, and any special taxes.

--Complication: Realizing that Pageflex uses some internal variables that the developer has no access to and can't use.

--Solution: Learned how the support side of Pageflex works and how to contact them for help. Setting up database calls up to reach out at specific points in the storefront's process and pull these values into my extension to use.

--Complication: newtonsoft issue between older pre-existing extensions and my newly created one.

--Solution: Found a code snippet to place in the web.config file that tells all extensions to look for their correct version.

--Complication: Realizing that debugging would be a nightmare with so many stacked processes

--Solution: Set up the code to log to a log file at the correct times to step by step see where I am in the process if it fails for any reason. Also set the code to send out an email with a detailed explanation of where things stopped working.

--Complication: Realizing that every time a database connection string and avalara account change the source code would need to change.

--Solution: Set up variables throughout the code that are supplied on the storefront from the admin side so that the source code does not need to be touched later.

--Complication: Learning that debug always being on floods the storefront logs with too much unneeded info.

--Solution: Setting up a way on the admin side to turn it off (It still logs a minimal message letting the user know that the process was started and when it is completed); however this being turned off does NOT stop logging to the .txt file I also set up so If anything happens I still can debug using this file.

--Complication: The process of adding the extension requires certain steps to be taken and files to be added the correct places in the storefront's deployment folder.

--Solution: Created a user setup guide, created a user walkthrough of extension, set up a gitbucket repository of all files needed.

[5]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Mack Extension Edit Project

--Start Date: 10-2-2018

--Purpose: To edit an existing web extension to help another developer I work beside

----------------------------

-Technologies Used:-

----------------------------

JavaScript, JQuery, CSS

------------------

-Description:-

------------------

Developer was having an issue with his code not working as he desired. It was supposed to highlight certain cells, but upon changing a value in a dropdown box these highlighted values were no longer highlighting. I worked and developed a solution using JQuery and CSS to accomplish what he needed.

-----------------------

-What I learned:-

-----------------------

-I learned about how the web extensions work at Communicorp

-I learned a few new abilities to use with JQuery

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: no complications

--Solution: no solutions

[6]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Exclude Handling Extension

--Start Date: 11-1-2018

--Purpose: To create an extension that prevents the storefront from adding a handling charge when a certain item is in the cart by itself

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Pageflex

------------------

-Description:-

------------------

I created and built my second extension here at Communicorp that interrupts the storefront when a user goes to purchase an item and checks to see if their cart contains only the certain Item that requires no handling, if so it prevents the handling charge from being added. It has logs setup for future debugging and sends out an email whenever there is an error.

-----------------------

-What I learned:-

-----------------------

-I gained a much deeper understanding of how extensions work and began to find ways to improve upon things I did in my first extension (eventually I went back and improved the Avalara Tax Extension with these changes for better efficiency as well)

-I learned how to work more with Angelo and get the specifications from the client to ensure I build the extension correctly the first time which saved the company money on not having to keep reworking the extension much after the initial build.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: While testing on our dev instance of the storefront I was working on I ran into issue with my extension giving errors of not existing in the deployment

--Solution: I found out that the dev instance and live instance of the storefront deployment were both pointing to the same database which could have caused major issues if not corrected. I got this fixed and got my extension to work as desired

[7]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Randstad-RPG Salary Guide Drop Down Project

--Start Date: 11-15-2018

--Purpose: To create and extension or web service to add functionality to the storefront for the customer's salary guide that would filter results for each dropdown based on a previous dropdown

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Pageflex, MSSMS (Microsoft SQL Server Management Studio), Ajax, ASP.NET, JavaScript, JQuery, HTML 5, CSS

------------------

-Description:-

------------------

I was tasked to work beside another developer who was already implementing a more manual quick fix to a client's request to begin building a less manual extension or web service. I was brought up to speed on the process, helping him to code some of his solution as I learned it. I then began to work on my solution, but kept having to rework things as the customer started to keep changing the scope of the project and what the desired result would be. After a few weeks of this our team decided to have me back off of my solution til the client was fully satisfied and the scope of the project could be fully defined. I helped further develop the other developer's solution with him for a short time until another project came up and I was delegated to move on to it.

I'm noting this project here, because it taught me a lot in regards to being cautious of scope creep and in working with another developer who I was helping to mentor during the development cycle. I have not been tasked with completing my extension/web service solution as of 2/7/2019, but should at some point in the future to add a database to remove some of the manual processes which will save money and time. Still waiting to see how the client reacts to the solution the other developer finished for the client.

-----------------------

-What I learned:-

-----------------------

-I learned to watch out for scope creep and how quickly it can complicate even the most simplest of requests at times

-I gained experience in mentoring and offering guidance to another developer while working together on a project

-I learned the importance of backing off and being patience during the development process to ensure I don't waste time over/under developing a project causing wasted time and money

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: I had tested and tried many ways to get a second dropdown to populate based on the first one with no success.

--Solution: I learned that Pageflex methods in an extension are independent of each other and that global variables are reset as each is called which means I had to find another way to pass variables like cookies.

--Complication: I ran into major scope creep and constant changes while trying to develop alongside another developer creating a similar solution.

--Solution: Our team decided to put my solution on hold and to have me assist the other developer till his was completed. After that I would re-begin working on my solution with a well-defined scope and task.

--Complication: I and the other developer ran into issues when trying to traverse through data that was nested in triple arrays and stored as JSON key pairs.

--Solution: We worked together to get to the data correctly and where we could use it to populate the drop down boxes.

--Complication: I and the other developer ran into an issue with the form on the storefront taking longer to load and causing the page to look frozen.

--Solution: We worked together to add a loading image that would give the user a better experience on the front side.

[8]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Storefront Analytics Rework

--Start Date: 12-7-2018

--Purpose: Originally this was just supposed to include a few minor changes that would allow my Avalara Tax extension and it to work together without file dependency issues, but quickly grew to be much more involved

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Pageflex, MSSMS (Microsoft SQL Server Management Studio)

------------------

-Description:-

------------------

While testing Avalara Tax Extension I ran into issues with a dll version clash between these two extensions. Upon looking into the source code I found a bunch of depreciated code that could be removed. I began by removing and fixing the code, but also began adding comments throughout the code were they were lacking. I also added the logging for future debugging class to this project to fix the original extension which was spamming the storefront logs continuously based on how it was previously setup. I then created an up to date walkthrough and installation guide.

After this time my manager asked me to add in some more functionality to the extension which required a decent amount of digging and reworking of code to solve, but was pulled off and the desired result was successful. I then updated any storefront that contained the older version with the newer version. I discovered there were two to three different versions of this dll installed on the storefront due to a lack of proper updating in the past, but this is not an issue anymore.

I then created a more advanced installation guide to walk a developer through the entire process of setting up google analytics, installation of the extension, and the database setup process. I also figured out how to setup the site search feature of google analytics on a storefront as requested and for when needed in the future again.

-----------------------

-What I learned:-

-----------------------

-I learned more about several of the developers previous to me who worked on this project before me.

-I learned how to setup and work with google analytics to a greater degree than I had before.

-I learned more in working with cookies and how to set and retrieve different values with them.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: finding that my Avalara Tax Extension and Storefront Analytics Extension were crashing each other over a dll version issue.

--Solution: I and a fellow developer found a snippet of code to place in the web.config that forced both to look for the newer version.

--Complication: While adding the new additions for my manager I discovered the rework would be more difficult with premade cookies being used in an odd fashion.

--Solution: Tested and learned how the cookies were being created and figured out how to add to them and alter data in them as needed to get both parts working together.

[9]##------------------------------------------------------------------------------------------------------------------------------------

--Project: Complete Sites and Extensions Project

--Start Date: 12-14-2018

--Purpose: To compile a list of all storefronts we use at Communicorp and determine which are active or unused and to also figure out which storefront have what extensions on them

----------------------------

-Technologies Used:-

----------------------------

Excel, Pageflex, MSSMS (Microsoft SQL Server Management Studio)

------------------

-Description:-

------------------

I started this project to gather a list of active storefronts and what extensions they have on them so that I could add my Avalara Tax Extension and the updated Storefront Analytics Extension as needed. I realized we had a lot of sites that are online, but not being used, whether they were test sites or for sites we no longer have the clients for.

After discussing my idea to compile a list for what sites we have and the extensions on each I began to build a list of requirements that would be good to know as I went. This list changed as I built my complete list and still has new pieces being added even after I had completed this. The core components I focused on were listing the storefronts and whether they were dev, live, active, dead, what members of our IT team have access, if certain group access was required, what servers they were on, and checkbox system for if an extension was present on each site or not. I also built a set of bookmark imports for me and the team that were up to date.

Once I got this completed I began to use it to add my extensions as needed, but I also gave it to the whole team to use as they saw fit. As of 2/7/2019 my completed list has been brought to the attention of the higher ups and a more in depth project is being set in place to use my list along with several other reports to automate and make simpler several manual processes we do here at Communicorp.

-----------------------

-What I learned:-

-----------------------

-I learned the number of storefronts that we have that are active, that are test sites, and that are sites we no longer have clients for.

-I learned a great deal about each storefront as I worked on it gaining a better in depth look into what each one is for and what they do.

-I learned a lot about how our servers and the structure of how the sites have been setup

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: I had no access to more than half of the storefronts I needed to collect information on

--Solution: I worked with another developer who could give me access and slowly got access to each site

[10]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Avalara Tax Rates Finder Web Version

--Start Date: 1-15-2019

--Purpose: To create a web application to display a tax rate for a user supplied zip code

----------------------------

-Technologies Used:-

----------------------------

Visual Studio web service application, ASP.NET, Avalara Tax Client 2018-2019 (Including the use of their REST API's), JSON, ColdFusion

------------------

-Description:-

------------------

I was asked to setup a web application that would display the tax rate of an area when a user supplied a zip code only. The code for this project would then be utilized with ColdFusion to replace another web application that was being used by the previous tax process I replaced. I have since completing this also added in extra functionality for the future including checking to see if the area code is US or Canada and displaying the results correctly for each.

-----------------------

-What I learned:-

-----------------------

-I learned more on Avalara and how it returns its API calls using JSON

-I learned a good bit of information on how the ColdFusion side is setup and how the code works with it.

-I learned how U.S. and Canada taxes differ

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: The only issue I ran into was figuring out how an Avalara API call is returned.

--Solution: Once I found it was JSON, I simply setup a way to get the information I needed from the file

[11]##----------------------------------------------------------------------------------------------------------------------------------

--Project: SFTP Project

--Start Date: 1-24-2019

--Purpose: To SFTP a file over to the server, then check if successful, if yes, then move the file from original location over to a backup folder

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, WinSCP, SFTP file transfers

------------------

-Description:-

------------------

I learned how SFTP works and created a project to achieve this with a file on my desktop. I first succeeded in getting the file to SFTP to the remote location. I then got checking in place to make sure the file was in fact transferred correctly. After that, I set up the code to then on success move the original file over to a backup folder as well. Once I had the core parts working I built in the logging feature and email upon error feature for future debugging and errors.

-----------------------

-What I learned:-

-----------------------

-I learned how SFTP works and how to accomplish this.

-I learned how to setup file checking and more about working with remote files

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: Only complication I ran into was in renaming the files during transfer to remote location. I was trying to add the date to the files and had an issue with not being able to get this done at the right time to prevent the file from overriding another file already on the remote server or backup folder

--Solution: I simply save the file name before I begin the transfer, change the name of the file, then transfer it as needed and set the name back to what it was if transfer fails.

[12]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Preview Report Project

--Start Date: 1-24-2019

--Purpose: To add a preview panel to a report page on a storefront

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Visual Studio utilizing VB, Pageflex, MSSMS (Microsoft SQL Server Management Studio), ASPX, HTML, CSS, JavaScript

------------------

-Description:-

------------------

I originally was tasked to simply fix an issue we had with having the user login twice to get access to the custom report page for the site. I solved this issue by putting in a simple check to ensure security and only allow certain users to reach this page. Then I was asked to add some major reworks to this extension for the client. I thought about keeping the code in VB and rewriting my logging and error checking code to work with it, but decided converting the VB to C# was much faster and helps to meet department standards better. I converted the code, added my logging and error features, and then tested to ensure it was working as desired. I then began work to get the preview panel added to the page. I actually added two extra panels. The first is added on page load and displays a preset report for the last 30 days. The second displays a report based on the user defined time range. I then added some extra security checks and improved logging automation.

-----------------------

-What I learned:-

-----------------------

-I learned a few new tricks and found some differences in converting from VB to C#

-I learned more techniques to use with ASPX

-I learned more advanced features and uses with cookie manipulation

-I learned more advanced skills to use when setting up and creating log files for debugging

-I gained further experience in dealing with a project that’s scope was constantly changing and having to adapt to the changes on the fly.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: Realizing the source code was in VB and all of my add-on features are C#

--Solution: Found it was quicker now and for future debugging to just convert the VB code to C# code

--Complication: Found an issue where the code behind the page was failing when attempting to access the page from an outside link

--Solution: Placed try catches and other safety nets to prevent it.

[13]##----------------------------------------------------------------------------------------------------------------------------------

--Project: INF Form Upload Process Project

--Start Date: 2-27-2019

--Purpose: To add another page to a site that would allow users to batch upload products in an excel file

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Able Commerce, Excel, ASPX, HTML, CSS, JavaScript

------------------

-Description:-

------------------

I was tasked with adding a link to the storefront’s homepage, but only for certain user groups. Then build the page that the link goes to that would allow the specific user’s to upload excel files that contained batches of products. This first part was setup using cookies and authorization techniques to achieve. The second part was actually changed to be a page with a frame that pulls in another site. The inner frame pulls up a several tabbed page that allowed the users to upload in batch, individual, and to see reports of previous uploads.

-----------------------

-What I learned:-

-----------------------

-I learned how our Able Commerce storefronts work and how to work within them.

-I learned how to setup a frame that pulls in another page.

-I learned more about login and authorization passing techniques

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: Was not aware that this was on an Able Commerce Site. I had begun to build the solution to this as if it were a Pageflex site

--Solution: Utilized what I had learned and reworked scope to correctly setup inside the other storefront

[14]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Internal Account User Field Rework Project

--Start Date: 4-23-2019

--Purpose: To add better checks and split a single field into two on a storefront. Also to make sure both fields move through order processing correctly.

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Able Commerce, ASPX, HTML, CSS, JavaScript

------------------

-Description:-

------------------

I was tasked with fixing an issue we had on a storefront where a field on the storefront that user’s entered data in was causing issues further down the line with internal processes. I first broke the field into two fields as the data requested was two variables that were space separated. I then added checks on both sides to ensure the data was only allowed to be submitted when it was in the correct format.

The second part of the fix was then making sure that the two fields were sent correctly into our accounting system and then further down passed correctly to our shipping agency to provide the customer with a correct billing statement. I worked beside another developer who had the lead on this issue. We had to change some code around and check to ensure the process worked as needed.

-----------------------

-What I learned:-

-----------------------

-I learned an immense about how our processes work when an order is placed and how the fields are passed.

-I learned some techniques I wouldn’t have thought of from the other developer as we worked together.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: The process of passing between our systems is very odd and is using some very old convoluted code files that were hard to sift through.

--Solution: We just took the time to isolate the sections we needed and made sure to keep backups as needed

[15]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Credit Card Processing Issue Project

--Start Date: 4-26-2019

--Purpose: To resolve the issue where credit cards were not processing on an Able Commerce storefront

----------------------------

-Technologies Used:-

----------------------------

Able Commerce, CyberSource, ASPX, HTML, CSS, MSSMS (Microsoft SQL Server Management Studio)

------------------

-Description:-

------------------

I worked on a team of two to dig into the site’s deployment folder and on the server it resided on to try and figure out why the process for handling credit card purchases stopped working. This process is setup on several sites, so the first approach was to compare the storefronts and see if we could find any differences, but this issue was not going to be that easy to solve. After making sure that there were no differences we then began to look into what on the server could be causing the issue. We found an error message that was being logged to a normally unmonitored folder that pointed to a TLS issue, but after many failed attempts to solve the issue we reached out to our third party development team and asked if they could find something we missed. At this point the other developer stepped out and I was left in charge of coordinating with our third party help to step them through what we had tried and to work with them to see if they found anything we missed. After several more failed attempts and tests we found a solution together that forced the server to send the TLS call as desired.

-----------------------

-What I learned:-

-----------------------

-I learned much about how the storefront processes credit cards and what procedures are in place behind the scenes.

-I learned more about TLS and the differences in versions and how to handle upgrading to newer versions.

-I formed a better relationship with our third party developers whom I have seen their work, but never spoke to myself.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: We were working with a project from a developer who no longer works here and who left no documentation for us.

--Solution: We had to just sift through the code and slowly piece together where everything was located and how it all worked.

--Complication: We could not initially solve the issue even though we had an idea of what the cause was.

--Solution: We worked with our third party developer group who have more experience with this site till a resolution was found

[16]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Dashboard work for all Pageflex storefronts

--Start Date: 6-3-2019

--Purpose: To update all custom dashboards on Pageflex sites to show data from new accounting system

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Pageflex, MSSMS (Microsoft SQL Server Management Studio), ASPX, HTML, CSS, JavaScript, Microsoft Dynamics AX, Avanti Slingshot, BI360, SlickGrid.JS

------------------

-Description:-

------------------

I was tasked with updating the custom dashboards we have on all of our Pageflex storefronts to display the inventory and order data from our new accounting platform instead of the older one. This has been a slow process. I have had to wait on others who are working more closely with the core upgrading project. This process has major implications and will affect every storefront we have on Pageflex so what is done now matters the most. I have made sure to consider all ideas and possible future development as I can while making this change. Update 8/7/2019: I have continued to work on this project as the scope and work requirements have continued to change and have adapted as needed along the way. Currently 2/3 of the project has been completed now.

-----------------------

-What I learned:-

-----------------------

-I have had to learn how to get the data I need from an ever changing project and adapt as needed.

-I learned much about another program we have that I had not worked with before.

-I learned how to work with SlickGrid.JS.

-I learned how Linked Servers work and how to use them.

-I have advanced my multi-database and multi-query knowledge quite a bit.

-I learned much about how both of our accounting programs work and the differences between the two.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: The project scope was very poor and had much ambiguity in it.

--Solution: I had to think out what was actually needed and approach those above to get a better scope.

--Complication: I was a lower priority for getting data and resources I needed to complete my project.

--Solution: I worked closer to help others who could get me the data I needed and got placed higher in ques as needed for the rest.

--Complication: There was an issue with the data we had from the older dashboard not existing or being in a different format in the new system we are moving to.

--Solution: Setup exceptions and worked to get clarifications as moved forward with the project.

--Complication: The data we needed was being pulled from three separate databases

--Solution: We debated linked servers, but decided to just copy the required data into a single database where we could add some extra fields for future use and easier maintenance.

[17]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Avalara re-work for line item calculations

--Start Date: 6-3-2019

--Purpose: To update the Avalara Tax Extension to now perform tax calculations on orders per line item

----------------------------

-Technologies Used:-

----------------------------

Visual Studio utilizing C#, Avalara Tax Client 2018-2019 (Including the use of their REST API's), Pageflex, MSSMS (Microsoft SQL Server Management Studio), Postman, JSON

------------------

-Description:-

------------------

I had to redesign the part of my Avalara tax extension that handled calculating taxes for storefront orders. It was setup to just send the entire order amount (price totals, shipping, and handling) as a single line item and calculate taxes on that, but as we have moved forward we realized there is a penny discrepancy on orders here and there that can cause issues when accounting for credit cards. To resolve this I had to rebuild this part of the extension to calculate tax on orders differently.

-----------------------

-What I learned:-

-----------------------

-I learned more about different ways in which tax can be calculated and what is acceptable as far as discrepancies are concerned.

-I learned more about dynamically building a JSON file to send and utilizing the return data as needed

-I learned better practices about when and how to calculate taxes from Avalara to ensure the data show on the storefront is the most accurate.

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication: The pre-built structure Avalara gives you to use in your code would not work with how our orders were being generated.

--Solution: I had to build my own process and manipulate the JSON file I wanted to send them dynamically. I also had to build the part that gets the return call and process this data as needed to give the storefront the proper values.

[X]##----------------------------------------------------------------------------------------------------------------------------------

Do to work schedule and more intensive projects that I can’t share as much publicly about. This list moving forward will be brief and just share the basic details about each project.

[18]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Design and develop a free a dynamic free shipping banner for a storefront

--Start Date: 8-12-2020

--Purpose: To dynamically show the user how much they left to order before they get free shipping

[19]##----------------------------------------------------------------------------------------------------------------------------------

--Project: New server setup and remediation efforts

--Start Date: 10-24-2019

--Purpose: Worked to put in security measures on the server and storefronts as needed

[20]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Old server remediation

--Start Date: 10-24-2019

--Purpose: Put the newly discovered fixes and measures into place on older servers as well

[21]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Automated a quarterly project

--Start Date: 11-19-2019

--Purpose: Team lead on a project to automate a process that saved 10+ hours of development time each quarter

[22]##----------------------------------------------------------------------------------------------------------------------------------

--Project: AbleCommerce Site Upgrade

--Start Date: 12-16-2019

--Purpose: Worked within a joint team project to fully redesign and upgrade an ecommerce site

[23]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Pentest remediation and fixes for site

--Start Date: 1-13-2020

--Purpose: To research, solve, and implement fixes for pentest findings

[24]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Setup a subscription service for a storefront

--Start Date: 2-24-2020

--Purpose: design, developed, and implemented a full subscription service onto a storefront

[25]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Lead on team to create storefront diagrams for a new site.

--Start Date: 2-28-2020

--Purpose: Create high-level architectural, container-level architectural, and data flow diagrams

[26]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Setup a new storefront site using new Angular JS

--Start Date: 4-1-2020

--Purpose: To setup a site that had a new structure behind it

[27]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Setup the site that had been pentested and corrected up to LIVE status

--Start Date: 4-15-2020

--Purpose: deployed the site after all fixes were in place

[28]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Created a modified version of Salary Guide for a new client

--Start Date: 4-15-2020

--Purpose: Redesign and developed a more modified version of a previous project for a new client

[29]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Upgraded Server and all storefronts inside it to latest version

--Start Date: 5-26-2020

--Purpose: To ensure the upgrades and all extensions were still functioning after a successful upgrade

[30]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Master Storefront and Server Control Sheet

--Start Date: 6-5-2020

--Purpose: To create an updated sheet containing all important information for the sites and servers under my department’s control

[31]##----------------------------------------------------------------------------------------------------------------------------------

--Project: Primary developer on an ADA Compliance project

--Start Date: 6-8-2020

--Purpose: To make three external facing sites fully ADA complaint for our largest client

[Full Template]##----------------------------------------------------------------------------------------------------------------------------------

--Project:

--Start Date:

--Purpose:

----------------------------

-Technologies Used:-

----------------------------

------------------

-Description:-

------------------

-----------------------

-What I learned:-

-----------------------

-I learned...

----------------------------------

-Complications or Issues:-

----------------------------------

--Complication:

--Solution: