EZTask

By

Eric Nortmann

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

_____

Eric Nortmann – Software Dev (Front End) 4/27/20

_____

Abdou Fall - Faculty Advisor 4/27/20

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2020

Table of Contents

List of Illustrations

FIGURES

Abstract

"Great things are done by a series of small things brought together." – Vincent Van Gogh

The main goal of our EZTask program is to allow our users to break down big projects and goals into simple, manageable steps so that they can more easily accomplish big things. EZTask is a .NET web application designed to aid users of all ages and situations in accomplishing their daily home and work tasks. The EZTask program is flexible enough to be used in a range of different environments. It can be implemented into a work environment by an administrator to help organize the workloads of his/her employees and boost their productivity. It also works well for a group of students working together on a project, or even a single user trying to organize their own personal project. EZTask breaks down big, potentially overwhelming projects or ideas into a series of small, easier steps. It makes it easy to work with others by allowing you to share/assign some of these steps, and track who is or isn't accomplishing their goals on time. It also makes the process of getting started on your work each day much smoother by clearly presenting each user with their individual goal for the day right when they login. Clear, concise tutorials combined with an intuitive interface make it simple to add tasks to the list while giving a satisfying text congratulating the completed work. This help app is unique due to its web-based platform, allowing users with a wide variety of phones, computers, and other devices to access their tasks lists so that no one is left out. This will allow group work to be done in a wide variety of long-distance situations as well as easy access from home regardless of the many different types of devices team members or employees might have.

Problem Statement

Problem

According to a study by American Psychological Association in 2007, 80 to 95 percent of college students admit to procrastinating on assignments on a regular basis. Another study was conducted by the University of Chicago to determine that around 20 percent of US adults are "chronic procrastinators" even beyond college.

Applications are out there to help manage tasks in a timely manner, but big projects are hard to plan and coordinate. Applications such as Microsoft Planner and Microsoft Team Foundation Server can help with scheduling tasks but requires the full Microsoft Office suite to use. On top of that, according to the review site Clickup, you "cannot notify people individually" and you "cannot prioritize tasks and subtasks". Much efficiency and time is lost in the planning phase of larger projects. This planning can be worsened by coworkers using multiple devices to keep track of the tasks, all which use different operating systems. Other applications such as Monday.com are "complex to learn, which is frustrating" according to review site Capterra.

Solution

An application that carefully manages each task, who completes it, when it is due, and at what time it's completed can better prevent procrastination, as everyone will see who's responsible each step along the way. Easy-to-use tutorials will help simplify the application. Administrators at the workplace will also be notified if any employee is slacking on their responsibilities. This application will be created using .NET in order to support browser usage

across almost all platforms. Using .NET allows virtually any device to be used to track and complete tasks.

Project Goals

We plan to bring an application that helps manage tasks far easier for solo and group projects at home and work. This application will offer better features for both users and administrators that are lacking on the current market.

Overview

The remainder of this report details how the project will be completed. It includes the sections: project concept, design objectives, methodology/technical approach, user profile, case diagram, Gantt chart, problems encountered, and conclusion.

Discussion

Project Concept

The overall idea of this application is to bring to the market a task managing app that helps far more than what the current market apps have now. While the market has many, there are none with easy task prioritization, notification sending, solo *and* group work management all rolled into one.

The Procrastination Help app was thought of and created by Andy Cress, Eric Nortmann, and Alexander Orr. The idea started as a joke as they were learning about the various deadlines

and requirements they would be responsible for meeting in their senior design class. With a large or lengthy project, often the hardest part is breaking it down into simpler steps, scheduling progress trackers, and figuring out how to effectively and fairly split up the workload amongst team members.

Design Objectives

The goal for this project is to create a program that makes project management and teamwork much simpler and easier to use than any other options out there right now. With big projects, planning and delegating efficiently can be very difficult and time-consuming. Yet, failing to do so can cause problems down the line. For students, this might just result in some bad grades. For large companies or contractors, rushed, late, or poor-quality work can end up costing a lot of time and money.

Our idea is an application/program that would be accessible by all members of a team. It would allow a project manager or "team leader" to create a project, delegate tasks, and set deadlines or due dates for specific tasks. We want this part of the program to be simple and easy to use, helping project managers to get through these steps as quickly and efficiently as possible. With other project management tools out there right now, most notably Microsoft Project, this can be done, but it can be frustrating to work with.

Where we believe our project will really shine is within the individual team members' view. When a team member logs into our program to begin work assigned to them by their project manager or team leader, they should be able to clearly see exactly what is expected of them as their next step. This page should clearly show a title, due date, and a description of the

work they are to complete. No need to worry about what comes next, what other people are working on, what the overall scope of the project is, etc. These things will be viewable on other pages of the program, but the login page for team members will be streamlined to only show the user what they need to see in order to get started on their work, cutting out everything that could be distracting, overwhelming, or just plain unnecessary.

If each individual can see exactly what they are supposed to do and when they need to do it by, this will eliminate much of the confusion and excess information that can overwhelm some people when working on a large project. This should allow each person to get started on each task much more quickly than if they had to read through the entire project instructions and decipher for themselves what parts they should do and when they should get started. This should help to increase productivity, lower stress levels, and eliminate procrastination as long as the project manager/team leader has planned for the project well. Any individual teammates that start lagging behind their assigned due dates will show up easily for the project manager to identify as they start missing their early deadlines, rather than finding out closer to the end of a project that someone is way behind and/or not pulling their weight on the project.

An example of this benefit of our program would be a group of students working on a large research paper. In a traditional student team, they may delegate the work by saying something like, each person is responsible for 5 pages by the due date for the paper. This could be a real problem if the leader of this student group doesn't find out that one of their team members hasn't even started or only has 1 or 2 pages done when the paper is finally due. If this student group used our program, the team leader would have been able to assign each student

5 pages broken down into tasks of 1 page at a time with custom deadlines. With this ability, students wouldn't be able to procrastinate too much without the team leader seeing that they are missing deadlines and being able to do something about it. Worst-case scenario, it would allow the team leader enough time to re-delegate the work if one of the team members just wasn't helping at all. At a larger scale, a company for using our program could identify all kinds of inefficiencies in this way (This supplier always delivers several days late, employee C always turns in his tasks 2 days late, this section of the project is falling way behind schedule already, etc.)

We want to design our program/online application so that it is easy to use and accessible by as many people as possible so that any team can use it despite team members potentially having different types of devices and/or smartphones. We originally wanted to develop an android application but realized that this would be a drawback as many groups of people will likely consist of at least some iPhone owners that we wouldn't want to isolate or prevent from using our app. We also realized that it would be nice to be able to access the program from either a smartphone or a laptop, with some users possibly preferring the ability to use the program on a desktop or laptop. Thus, we decided to develop the program with the .NET framework so that it will be accessible through any modern web-browser. That way it would work for team members working on a desktop/laptop as it would for team members on mobile.

Technical Elements

Application

As mentioned in our design objectives section above, we chose to develop our program with the .NET framework so we can make it accessible to as many people as possible through an online webpage. One problem with this decision is that it will require some extra work compared to an android application. We will need to create more than one user-interface/view depending on if the user is accessing the program from either desktop or mobile. We will also need to setup hosting for our website and database once the project is complete. However, these drawbacks seem more than worth it for the ease-of-access and real-time teamwork functionality it could provide for our users. This is a distinct advantage over Microsoft Project, which will likely be one of our main competitors.

Database

To make all of this work, we will need to develop a database for storing user credentials, permissions and projects that have been created by our users. We will be using Microsoft SQL servers to store our data. The database should be able to store and regularly update the changes made to each of the user-created projects created within our application. Each client, mobile or desktop, will regularly check the database for any changes made to the currently logged in user's projects, and update the user-interface so that the user can see what their teammates are doing.

Networking

For this to work properly, we will need users to have an active internet connection so that any changes made to the projects by other teammates can be updated on each team-members device in real-time. If we decide to, we could easily add an offline mode that would warn users that they won't see any of their teammates changes, or be able to share their changes with teammates until they connect to the internet, but would allow them to view their projects as they were the last time they connected to the internet with the application/program open. This would require a little bit more development to include a way to check if the user is online or not, and to download the user's most current project data each time they login to the program so that it would be saved on their device in case they end up losing internet connection for some time. This could also be an optional setting for users to turn on only if they wanted to, which would prevent users from complaining that the program takes up space on their devices. With this setting turned off, there really shouldn't be hardly any space taken up on our user's devices as everything is saved in our database and accessed over the internet.

User Profiles

Figure 1 and 2 illustrate the user profiles for EZTask. They show the potential users for the application, experience required or recommended to use the application in that position, and the frequency of use expected when using the application.

(Figure 1) USER PROFILE: Group Member/Individual

PROJECT: EZTask
POTENTIAL USERS: - Non-technologically adept individuals
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: This project is targeted to those with little to no technical experience. The application is designed to give tips and tutorials throughout the start and onward to actively teach the user how to properly handle each function. The project should be useable by any person of any varying age and experience with technology, with varying support to fit the demands of the user.
EXPERIENCE WITH SIMILAR APPLICATIONS: Users with experience in using other task-completion projects should have an easier time using the project. This is not required, however, as the project is made for someone with no experience in mind.
FREQUENCY OF USE: There will be an initial setup for the project. Afterwards the application will be used once or more per day depending on the amount of tasks at hand.
KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS: - Easy to use UI - Notifications of any fall-behind behavior - Easy to manage adding and removing tasks and dates

(Figure 2) USER PROFILE: Administrator

PROJECT: EZTask
POTENTIAL USERS: - Non-technologically adept individuals

SOFTWARE, INTERFACE, AND RELATED EXPERIENCE:

This project is targeted to those with little to no technical experience. The application is designed to give tips and tutorials throughout the start and onward to actively teach the user how to properly handle each function.

The project should be useable by any person of any varying age and experience with technology, with varying support to fit the demands of the user.

EXPERIENCE WITH SIMILAR APPLICATIONS:

Users with experience in using other task-completion projects or administrative tasks should have an easier time using the project. This is not required, however, as the project is made for someone with no experience in mind.

FREQUENCY OF USE:

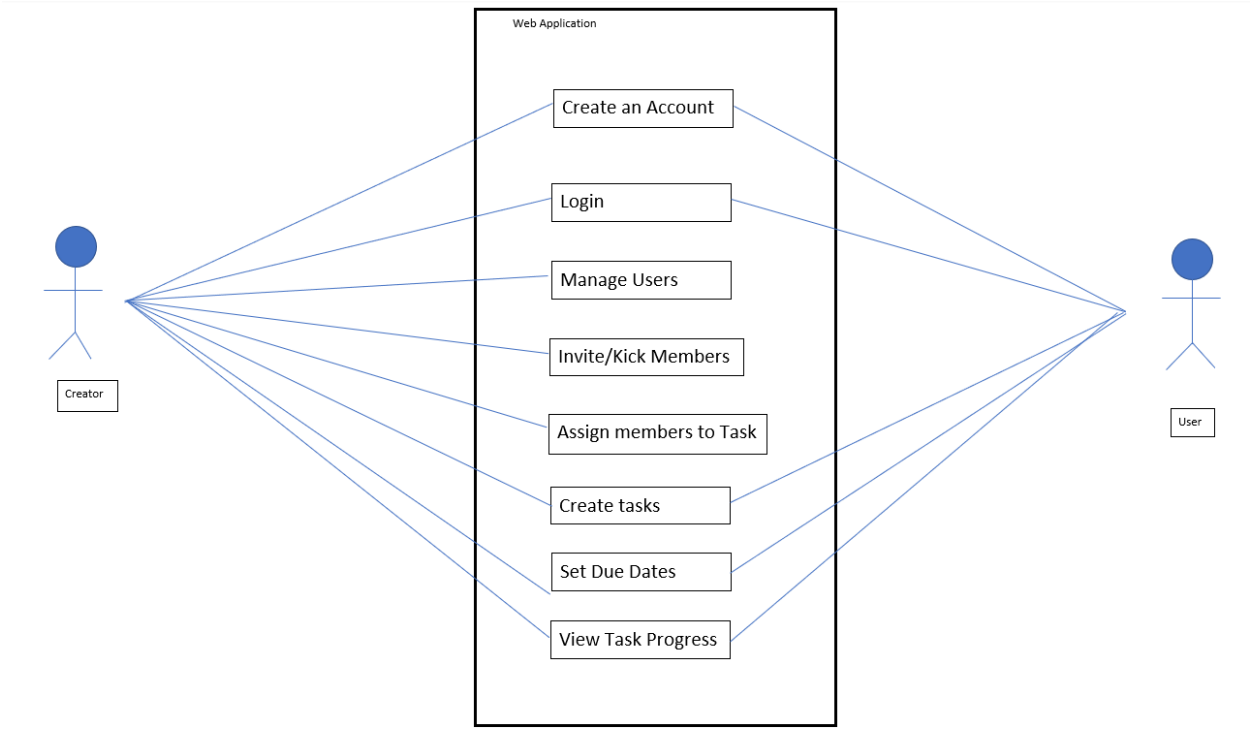
There will be an initial setup for the project as well as connectivity of group members. Afterwards the application will be used once or more per day depending on the amount of tasks at hand.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Easy to use UI
- Notifications of any fall-behind behavior
- Easy to manage adding and removing tasks and dates

(Figure 3) Use Case Diagram

Figure 3, the use case diagram, illustrates the use case for EZTask. It displays all the users and their corresponding tasks they'll be allowed to access.



Technical Discussion

Unfortunately due to extenuating circumstances, the Network/Security and Backend/Persistence portion of our project plan could not be completed. The Front End was completed and just needed the completion of the backend to become fully functional. The UI was designed with a darker theme in mind as it's easier on the eyes and can calm the mind. The interface is intended to be simple and easy to grasp, as to not divert the user.

Testing Approach

Overview and Methodology

For our project, we felt that It would be best to use manual testing with documentation. Additionally, there will be edge case testing to see if we can break our application. The way we set out to achieve this, is whenever we created a feature, we would make a corresponding Test Case to specifically go against the feature in order to test for its full functionality.

Testing Process:

- Both the developers and security specialist will cooperatively create test cases. This ensures that all angles of possibility are covered.
- Features should be tested in steps as each piece begins to work.
- Once code for the given feature is published and each step has been tested, the test should be marked as Passed.
- Any member of the team may run tests as long as they ensure they are testing for all instances listed.
- If the test fails, it must be documented, and the developers must be informed.

As with most things, it is always better to get someone who did not create it to analyze and test. In doing so we can be sure that we are testing for all cases that may have not been thought of when the feature was first created. As we are in different fields of IT, we all do not share the same knowledge that the others possess. This is why when a developer is building a feature, they might not be aware of the security risks at hand.

The first thing we did when testing, is to assume the role of the standard user who will follow along exactly as they are intended to with our web application. This enables us to test the raw, base functionality of the application.

One of our main tests is input validation. As is the nature of our web application, we have a lot of user input. So, we want to be sure that the user cannot enter garbage inputs or invalid dates.

Once base testing of the entire application is finished, we will spend time cooperatively finding ways in which we can break the application and or cause it to fail. We want to be sure that the security is strong, as people will be storing passwords and possibly tasks that they may deem confidential.

Scope of Testing

The main objective of our testing is to validate that all our web applications core functions are performing as intended and delivered upon. Tests are performed during each release of EZTask on new functions added or past functions improved. To achieve this, we tested for each of the End User Roles.

Objectives

The goals we set for our testing is as follows:

- All deliverables and core application features have corresponding test cases that check against each step of the feature.
- The entirety of our Test Cases must align with each End User Roles.
- All Testing must be documented and logged, especially if the test failed and the corresponding party notified.
- All errors and bugs are resolved prior to the IT Expo.

Logging Test and Procedures

The goal of our testing is that no matter what method was used to test a feature, if a bug was found, or it was able to be broken, it will be documented and the team notified as to what caused the failure. We keep all our test instances on file as to view history when certain bugs may reappear it will be an easier fix because we have to documentation listing what was done to resolve said bug.

Each member on the team is responsible for testing features. As previously stated, this ensures that all angles are covered due to each member having a different mindset on how to attack the issue.

Test Results

The best way we determined to handle testing, is to test against the goals of our features rather than specifically the code. This means that we can be sure that at the end, we can confidently deliver on what was promised for our features. Additionally, once a feature is created, it should be tested and passed before a new one is created. This will prevent against a spider web issue where multiple features are built and reliant upon a previous one that is now proven to be faulty.

Tests

Test Name	**Pass/Fail**	**Notes: "This failed because…"**
Users can sign up, create username/password	F	The database was not created
Valid users can log into the program and access their projects	F	Database wasn't created to house permissions
Unauthorized users can not access other users' projects	F	Database wasn't created to house permissions
UI displays correctly on login screen and once logged in	P	
Project steps can be added by project manager/admin users, and not by other users	P/F	Functionality is there, but database permissions not built
Due dates can be added successfully by project manager/admin users, bad dates are rejected by the system	P/F	Functionality is there, but database permissions not built
Due dates can be edited/changed by manager/admin users and not by other users	P/F	Functionality is there, but database permissions not built

All users can mark their assigned steps completed, and other users on the same team can see when this is done	P/F	Functionality is there, but database permissions not built

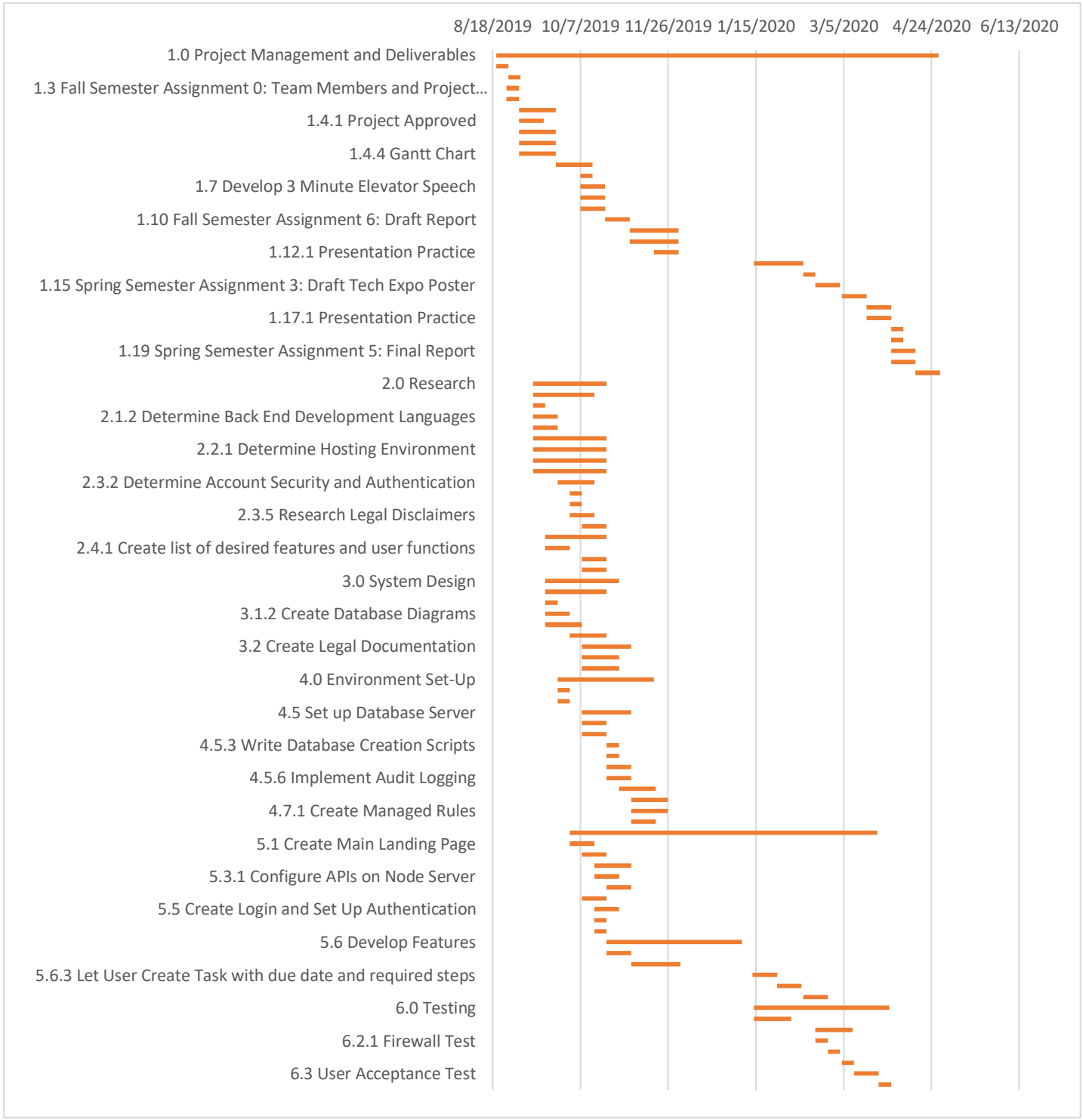
Budget Schedule

One of the benefits of this project is that is free to the user, and at no expense to us. I have a free hosting site and the plan was to also utilize a free database server host.

Project Schedule

Figures 4 and 5, Gannt Chart and WBS respectively, outline the projected work schedule for the team to complete the desired idea for the web application.

(Figure 4) Gantt Chart



EZTask WBS

(Figure 5) WBS

Task Name	Duration (Days)	Start Date	End Date
1.0 Project Management and Deliverables	252	8/20/19	4/15/20
1.1 Team Role Forming	7	8/20/19	8/27/19
1.2 Project Idea and Flesh out	7	8/27/19	9/3/19
1.3 Fall Semester Assignment 0: Team Members and Project Name	7	8/26/19	9/2/19
1.3.1 Project Name	7	8/26/19	9/2/19
1.4 Fall Semester Assignment 1: Team Contract	21	9/2/19	9/23/19
1.4.1 Project Approved	14	9/2/19	9/16/19
1.4.2 Develop Problem and Solution Statement	21	9/2/19	9/23/19
1.4.3 Work Breakdown Structure	21	9/2/19	9/23/19
1.4.4 Gantt Chart	21	9/2/19	9/23/19
1.5 Fall Semester Assignment 2: Project Abstract for Tech Expo	21	9/23/19	10/14/19
1.6 Fall Semester Assignment 3: Team Contract Resubmission	7	10/7/19	10/14/19
1.7 Develop 3 Minute Elevator Speech	14	10/7/19	10/21/19
1.8 Fall Semester Assignment 4: User Profile	14	10/7/19	10/21/19
1.9 Fall Semester Assignment 5: Use Case Diagram	14	10/7/19	10/21/19
1.10 Fall Semester Assignment 6: Draft Report	14	10/21/19	11/4/19
1.11 Fall Semester Assignment 7: Final Fall Semester Report	28	11/4/19	12/2/19
1.12 Fall Semester Oral Presentation	28	11/4/19	12/2/19
1.12.1 Presentation Practice	14	11/18/19	12/2/19
1.13 Spring Semester Assignment 1: Testing Plan/Report	28	1/14/20	2/11/20
1.14 Spring Semester Assignment 2: Abstract	7	2/11/20	2/18/20
1.15 Spring Semester Assignment 3: Draft Tech Expo Poster	14	2/18/20	3/4/20
1.16 Spring Semester Assignment 4: Final Poster	14	3/4/20	3/18/20
1.17 Spring Semester Oral Presentation	14	3/18/20	4/1/20
1.17.1 Presentation Practice	14	3/18/20	4/1/20
1.18 IT Expo	7	4/1/20	4/9/20
1.18.1 IT Expo Exhibit and Preparation	7	4/1/20	4/9/20
1.19 Spring Semester Assignment 5: Final Report	14	4/1/20	4/15/20

1.20 Spring Semester Assignment 6: Safe Assign Final Report	14	4/1/20	4/15/20
1.21 Spring Semester Assignment 7: Final Library Copy	14	4/15/20	4/29/20
2.0 Research	42	9/10/19	10/22/19
2.1 Software Requirements	35	9/10/19	10/15/19
2.1.1 Determine Front End Development Languages	7	9/10/19	10/1/19
2.1.2 Determine Back End Development Languages	14	9/10/19	10/1/19
2.1.3 Find Suitable Database for storing users	14	9/10/19	10/1/19
2.2 Network Requirements	42	9/10/19	10/22/19
2.2.1 Determine Hosting Environment	42	9/10/19	10/22/19
2.3 Security Requirements	42	9/10/19	10/22/19
2.3.1 Research HIPAA Compliance	42	9/10/19	10/22/19
2.3.2 Determine Account Security and Authentication	21	9/24/19	10/15/19
2.3.3 Research Common Malicious Injection Methods	7	10/1/19	10/8/19
2.3.4 Research Custom Port Numbers (TCP/Layer 4)	7	10/1/19	10/8/19
2.3.5 Research Legal Disclaimers	14	10/1/19	10/15/19
2.3.6 Research Login/Credential Storage	14	10/8/19	10/22/19
2.4 Additional Research	35	9/17/19	10/22/19
2.4.1 Create list of desired features and user functions	14	9/17/19	10/1/19
2.4.3 Interview would be Users	14	10/8/19	10/22/19
2.4.4 Budget Analysis	14	10/8/19	10/22/19
3.0 System Design	42	9/17/19	10/29/19
3.1 Create System Diagrams	35	9/17/19	10/22/19
3.1.1 Create Network Diagrams	7	9/17/19	9/24/19
3.1.2 Create Database Diagrams	14	9/17/19	10/1/19
3.1.3 Create Interaction Diagrams	21	9/17/19	10/8/19
3.1.4 Create Wireframe Diagrams	21	10/1/19	10/22/19
3.2 Create Legal Documentation	28	10/8/19	10/29/19
3.2.1 Draft Legal Disclaimers and Privacy Policy	21	10/8/19	10/29/19
3.2.2 Draft HIPAA Information and Consent Form	21	10/8/19	10/29/19

4.0 Environment Set-Up	55	9/24/19	11/19/19
4.1 Import Libraries for Development	7	9/24/19	10/1/19
4.2 Setup GitHub	7	9/24/19	10/1/19
4.5 Set up Database Server	28	10/8/19	11/5/19
4.5.1 Dual role as domain controller	14	10/8/19	10/22/19
4.5.2 Install Microsoft SQL Server	14	10/8/19	10/22/19
4.5.3 Write Database Creation Scripts	7	10/22/19	10/29/19
4.5.4 Link Database Tables	7	10/22/19	10/29/19
4.5.5 Configure User Groups and Access	14	10/22/19	11/5/19
4.5.6 Implement Audit Logging	14	10/22/19	11/5/19
4.6 Set up Web Application Servers	21	10/29/19	11/19/19
4.7 Set up Web Application Firewall	21	11/5/19	11/26/19
4.7.1 Create Managed Rules	21	11/5/19	11/26/19
4.8 Set up File Share Server	14	11/5/19	11/19/19
5.0 Development (Back End and Front End)	175	10/1/19	4/1/20
5.1 Create Main Landing Page	14	10/1/19	10/15/19
5.2 Create Navigation Bar	14	10/8/19	10/22/19
5.3 Set up API Routers	21	10/15/19	11/5/19
5.3.1 Configure APIs on Node Server	14	10/15/19	10/29/19
5.3.2 Configure Backend APIs	14	10/22/19	11/5/19
5.4 Design and Develop Dashboard	14	10/8/19	10/22/19
5.5 Create Login and Set Up Authentication	14	10/15/19	10/29/19
5.5.1 Create User Registration and Confirmation	7	10/15/19	10/22/19
5.5.2 Create Forgot Password	7	10/15/19	10/22/19
5.6 Develop Features	77	10/22/19	2/25/20
5.6.1 Set Alerts for administrators if tasks are overdue	14	10/22/19	11/5/19
5.6.2 Develop task viewing and stats for admins on how quickly members complete tasks	28	11/5/19	12/3/19
5.6.3 Let User Create Task with due date and required steps	14	1/13/20	1/27/20
5.6.4 User sets the level of difficulty to complete each step	14	1/27/20	2/10/20

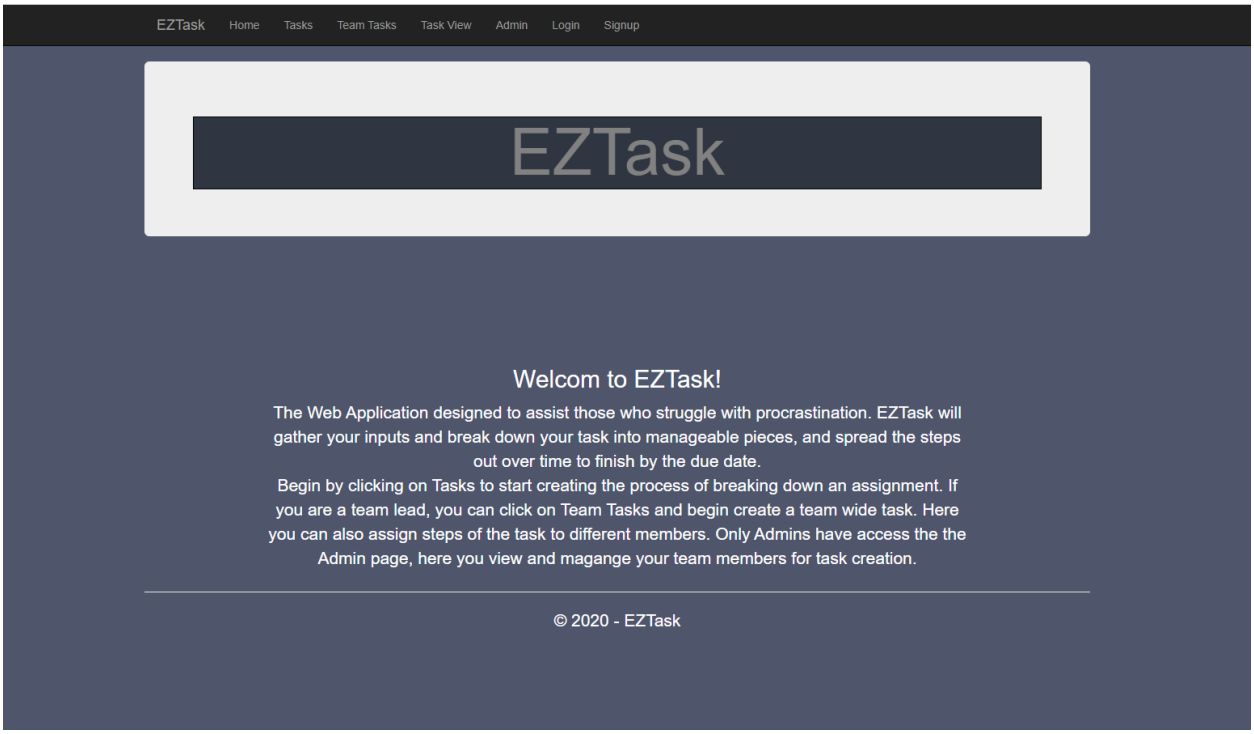
5.6.5 Application then generates schedule to complete assignment on time	14	2/11/20	2/25/20
6.0 Testing	77	1/14/20	4/1/20
6.1 Set Up Mock Clients	21	1/14/20	2/4/20
6.2 Security Test	21	2/18/20	3/11/20
6.2.1 Firewall Test	7	2/18/20	2/25/20
6.2.2 Penetration Test	7	2/25/20	3/4/20
6.2.3 SQL Injection Test	7	3/4/20	3/11/20
6.3 User Acceptance Test	14	3/11/20	3/25/20
6.4 Final Certification Test	7	3/25/20	4/1/20

USER INTERFACE

Home Page

We want our landing page to have a simple clean design and orientation. Complex layouts can be detrimental to the success and purpose of the application. We will also have a concise description of the application so users understand the purpose before they dive in. The color scheme will be calming and not harsh on the eyes, a darker pallet. All levels of users will have access to view and interact with the home page. On this page users will be able to view the other pages within the application available to them, or they can immediately create a task to schedule out. Figure 6 aims to achieve this by showcasing a darker theme of colors and an introduction message to users.

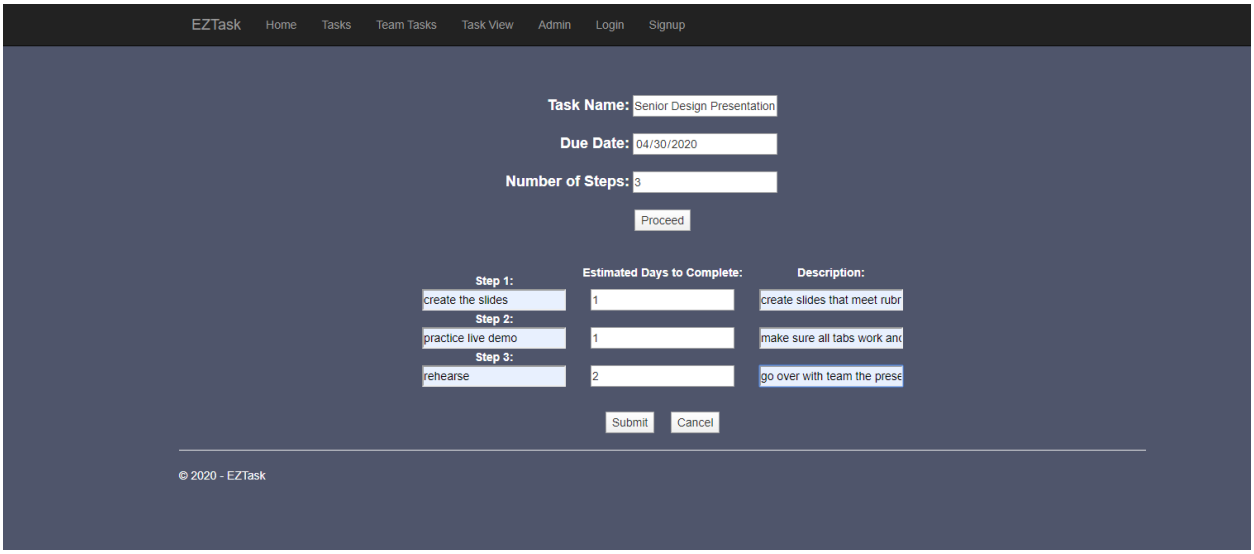
(Figure 6) Home Page



Task Page

The Task Page will be the meat of our application. Here all users will be able to create a task to be scheduled out. On this page the user will enter the final due date for the task and the required steps or dependencies to complete said task. Additionally, the user will be asked to enter the level of difficulty and estimated to of completion for each step/dependency. Once completed, the user will have the option to click the “Proceed” button to get a breakdown of their task. The same layout and theme from the Home Page will carry throughout the application for consistency and simplicity. On the left side of the page will be the same list of pages view from the Home Page. In Figure 7, you will see the same themes and menu sidebar persist from the home page.

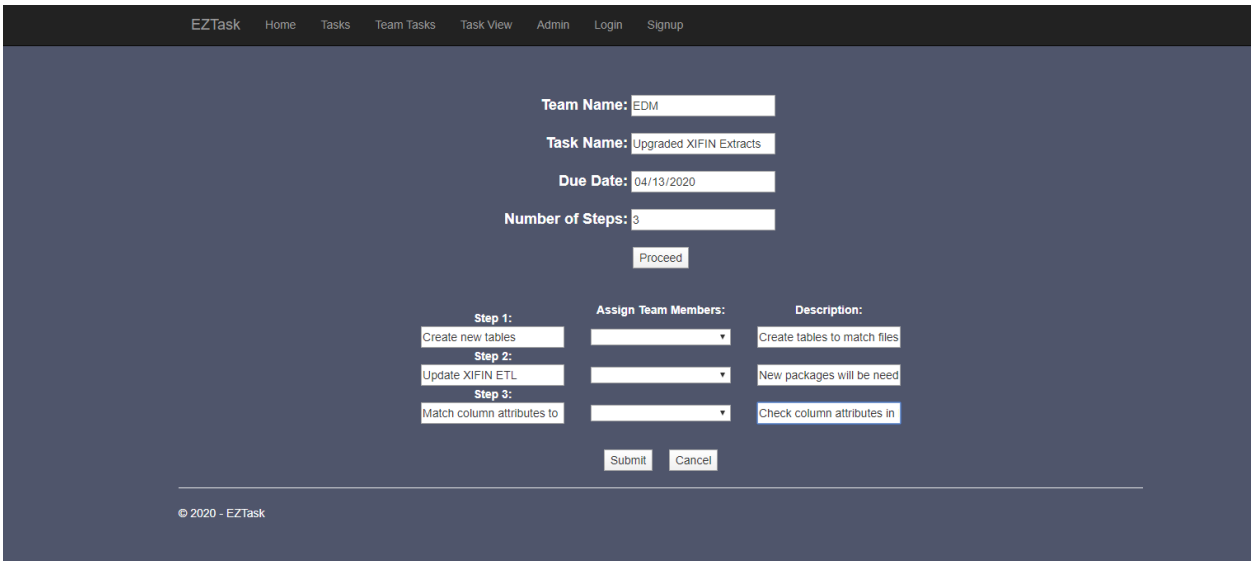
(Figure 7) Task Page



Team Task Page

The Team Task Page is designed for team leads and project managers. This Page will display the tasks created by the Task Page. The team lead will be able to select a task(s) and assign it to a user under their supervision. Again, the page shares the same layout and color themes as its predecessors. Figure 8 is meant to be very similar to the previous task page, however, notice the input of Team Name. Additionally when “proceed” is clicked it will produce a different container than the Task Page one to allow the Team lead to assign task directly to a team member.

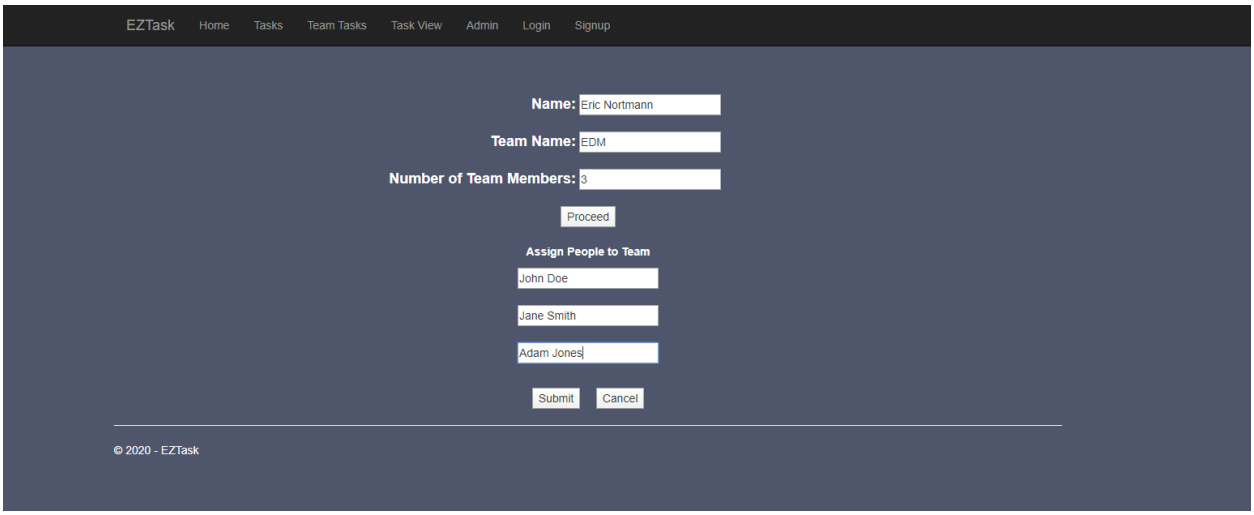
(Figure 8) Team Task Page



Admin Page

The Admin Page, or Team Lead Page, is the page where the Team Leader will be able to manage the users under their supervision. Here, a team is created and users can be assigned to and or removed, by the Team Lead. Additionally, the Team Lead can view each users task progression. This page will only be accessible to users who have registered as a Team Lead. Figure 9 depicts the page where the Admin can register their team and the members on their team.

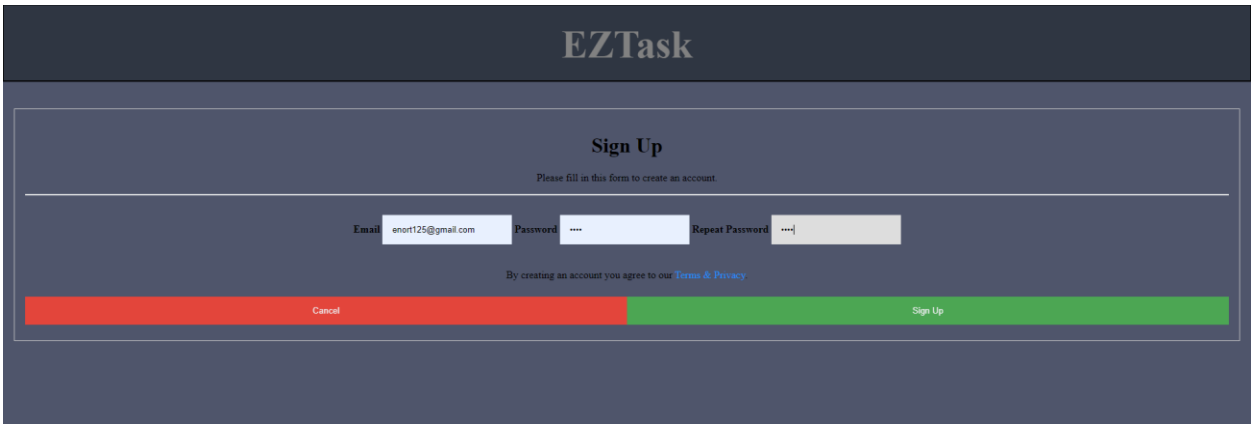
(Figure 9) Admin Page



Signup Page

Figure 10 Signup Page, illustrates what users will see upon registering to become a team member or team admin. Here, users will enter an email, password, and then select whether they are a team admin or not.

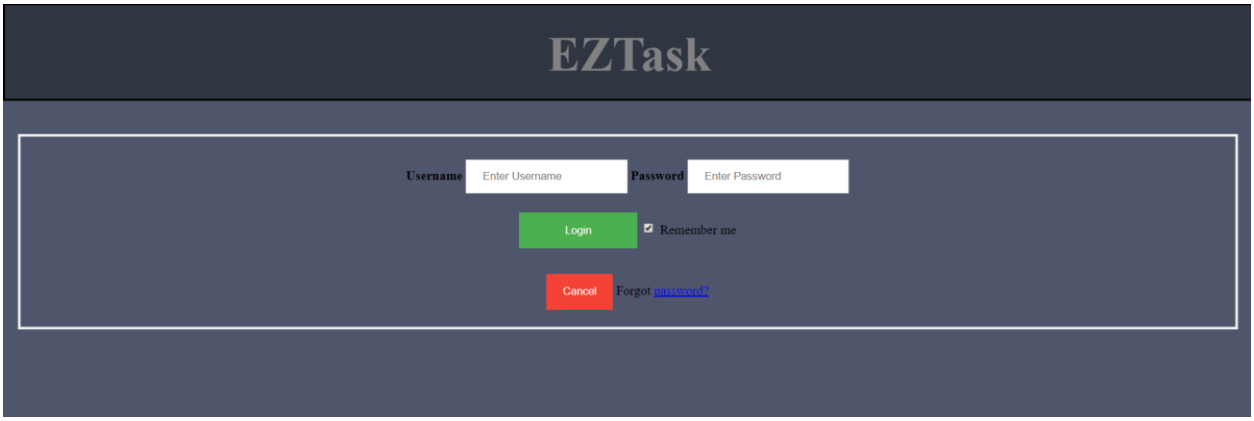
(Figure 10) Signup Page



Login Page

Figure 11 Login Page demonstrates the page users will access to login in in order to receive their proper permissions to view their tasks.

(Figure 11) Login Page



Problems and Problem Analysis

The main problem that I experienced was procrastination and problems within the team dynamic. Essentially, major aspects of the project were not started/completed due to circumstances beyond my control. I fulfilled my end of the project and even tried to do what I could to try and get some sort of backend completed but was unsuccessful. My analysis of the problem is that the group was not prepared for the amount of work required and greatly

underestimated what was needed in order to get this project up and fully running. It is a shame that it had to come to this and leave with an unfinished project, but it what was best for the group. The main takeaway from this being that when trouble is encountered or you might suspect issues with the team, would be to go to the professors immediately. They can provide you with guidance or give the team the kickstart they need in order to get the project to its desired state.

Recommendations for Improvement

If I were to do this all over again, for starters I would make sure the group I was with knew the importance of the class and would be able to dedicate the time necessary to complete the project. Additionally, I feel I've learned a great deal about leading a team and taking charge of a situation. In the past, I used to hang back and just do what was asked of me. Now, I've learned how to ask of others and to oversee project functions, and I quite enjoyed that aspect.

If given more time, I would see to it that the backend and persistence receives the attention it needs to give the web app its full functionality. I would have also played around with the addition of new features that I believe would add more possibilities to the site, but could add some complexity.

After this class concludes, I plan on making it a passion project of mine. I will finally complete the backend so I can actually store tasks. Once this is completed, I can actually use my web app to schedule my own personal tasks to alleviate some of the procrastination I deal with.

Conclusion

Fall Semester 2019

The task of creating a web-based application from nothing, with different degrees and levels of coding experience across all team members, has definitely proved to be challenging. The idea started as an android application to aid those who struggle with procrastination. However, upon learning that only one team member had experience in Android Studio, we wanted to expand on the idea to incorporate businesses and teams rather than individual users. We shifted the idea to be a web-based application. What sets our idea apart from other scheduling assistants is that it will be automated; meaning the application will dictate when the user should begin working on a given task and break that task down into manageable bits as to not overwork and burnout the user. Additionally, our idea will require less user interaction to track progress.

Once we settled on the idea, there were a few speed bumps. Such as, what do we call our idea so that it's simple and easy to learn but also conveying the purpose of our application. The main blockade being the actual logic to run the application is still in development. As of today, the .NET framework for the application has been laid but actual coding towards the layout and logic need to begin soon. We have decided on a database to store users and their credentials for login, as well as their position in a team if they are in one. This database will also have to store tasks and task history for each user.

Spring 2020

The main lesson I learned this semester was how to produce under pressure and with drawbacks. In the past I would've thrown in the towel and complained about not receiving help. However, I was determined to complete my portion of the project and truly give it all that I can to ensure that the project was completed to my best ability. Building off this, the main skill I developed is how to become a team leader and deal with conflict. I really learned this new aspect about myself and have come to find that I really enjoy being a team lead and overseeing a project. I used to be very laid back and avoided confrontation but after this project I learned about how necessary it is to have someone being in a take charge position to ensure the project progresses on schedule.

Since the Fall semester, I have transitioned our application from basic html pages to .NET asp application. Furthermore, I created additional web pages, added dynamic form functionality, aesthetic changes, for each form I also added validation to each input to ensure garbage data wouldn't be accepted. For each page in the application I also created specific css pages to style the page accordingly. In the fall, I created a very basic mockup to illustrate the idea of my web app. In the Spring I completely overhauled what I had for the app and completed the entirety of the front end and UI for the web app. Seeing fellow presentations helped me understand how vital this program is to our success. Being able to see all aspects of what IT can offer is incredibly interesting. Additionally, see what incredible projects my peers are able to come up with is very inspiring and gives me hope for the future. Finally, I also learned a great deal on building and presenting a professional presentation and report for a project. Ideals such as this can be applied to my professional work environment and help me succeed in my field.

References

Hamilton, Audrey. “Psychology of Procrastination: Why People Put Off Important Tasks Until the Last Minute” April 5th, 2010. Accessed October 1st, 2019 https://www.apa.org/news/press/releases/2010/04/procrastination

Hubbard, Brennen. “Study finds up to 95 percent of college students procrastinate” February 26th, 2014. Accessed October 1st 2019 http://thebluebanner.net/study-finds-up-to-95-percent-of-college-students-procrastinate/

Spilker, Josh. “Should You Use Microsoft Planner in Office365? Best 9 Alternatives” March 2nd, 2019. Accessed October 1st 2019 https://clickup.com/blog/microsoft-planner-office-365-alternatives/

H., Johnathan. “Monday.Com Keeps Track of our Progress” August 15th 2019. Accessed January 2nd 2020 https://www.capterra.com/p/147657/monday-com/

Appendix A

Poster