DISD

XX/11/2021

I, Raphael Son Hing hereby declare that I did not plagiarize any content in this assignment and that this my own work.  
Signed:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Programming 2b POE

Raphael Son Hing 15004658

Contents

[2 Introduction 2](#_Toc88078451)

[3 Style 3](#_Toc88078452)

[4 Register Page 4](#_Toc88078453)

[5 Login 5](#_Toc88078454)

[6 Schedule 6](#_Toc88078455)

[7 Graph 7](#_Toc88078456)

[8 Extract for Task 1 8](#_Toc88078457)

[9 Conclusion 13](#_Toc88078458)

[10 Various Screenshots 14](#_Toc88078459)

[11 References 18](#_Toc88078460)

# Introduction

Greeting and salutations, it’s time to discuss the our POE project due for mid November.

This time, our trial for the year is to covert my previous project, the Task 2 version of my application, into an ASP Core application.

Gone is the common programming language and C# and into the more complications of HMTL.

Of course, this is a relatively new experience for more for me as Visual Studio is specific on how I can manipulate the coding while still within it’s compatible standards.

Of course this project is accompanied with Github and video links.

Here are the github and youtube links.

<https://youtu.be/9BVc5gHz50c>

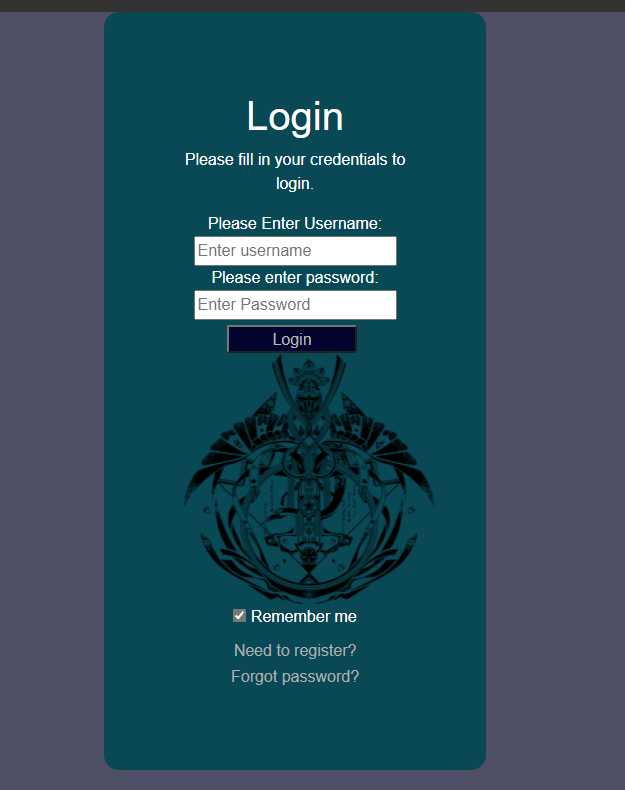
<https://github.com/RaphaelSH/PROGRAMMING2B-POE-15004658-Raphael_Son_Hing>

As a note, my machine is unable to process the project, so I had to resort to utilizing varsity’s resources to program it. Unfortunately this left me with without any decent or practical way to record my video as Varisity’s computers did not let me use any good recording applications, thus my only course for correction was to use my phone camera.

Due to this, video and audio were recorded separately, and I find myself with bad quality video and having to splice them with audio files manually, there may be times an hazing of syncing the files together due to this, so please forgive if they do not properly mesh together.

# Style

For this project I default to using a favoured style of mine, such as exampled here…

I would like to clarify, although I have used this style before, it is not plagiarism as I am the original creator of the style myself. Of course I grant myself permission to put it into use.

The former GUI has been converted into HTML, this means that the smaller efficient GUI is not longer meets the criteria and what must be retired. Salvaging the critical components and then turning the into presentable HTML format

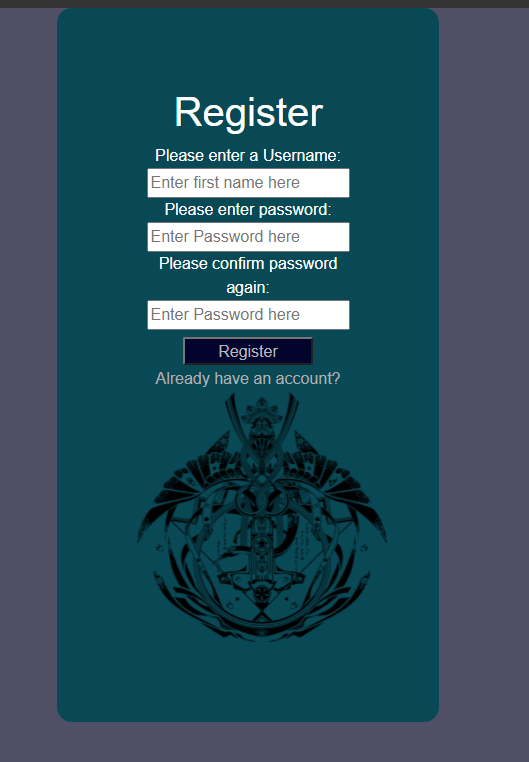
In my perspective, I find darker colours more relaxing to the eyes and the colours compliment each other when limited into a small neat selection.

This is implemented into a style page, where the most common of such uses’ attributes are stored and are called upon to easily put my choice influences my more common functions.

I have included a nav bar as well to provide easier navigation throughout the website, the nav bar is a pseudo there as while normally there would be restrictions to which area a certain user may go due to there permissions, for testing sake, this version allows free reign to go anywhere in the app.

As a firm believer on of pasting one logo everywhere, an app wouldn’t be complete without blasting you into the face with there signature symbol. Also include are logos for Youtube and Twitter, as I feel they make the presentation look more authentic.

# Register Page

With the implementation with a login and register page, her is the classic register page.

As with the several thousands cliché versions of them, the user enters a username and password.

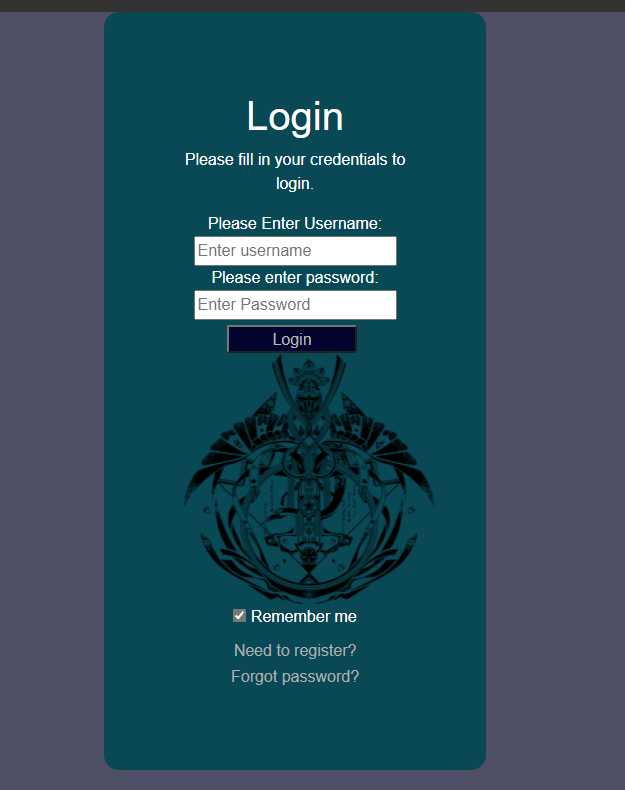
If they meet the criteria (in this case matching passwords), they will create an account upon witch they can utilise to enter the application.

For simplicity sake only a few fields are necessary as with the amount of data taking place in this project, more than necessary attributes would just cause unnecessary complications.

No email field necessary, and a username field to suffice we get ride of the overly long field, as it was a grievance having to type a full email every time I wanted to test it.

No register page would be complete without a link to the Login page as well, the test below the register button is created with this intention.

# Login



The login page is par for the course, when a register page is used.

The login page is simple design and functionality.

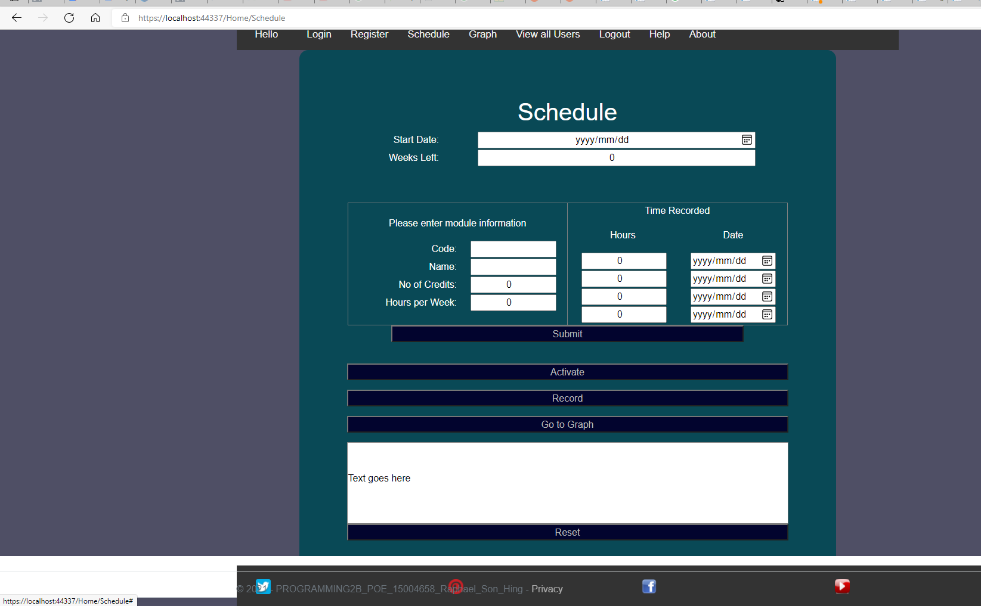
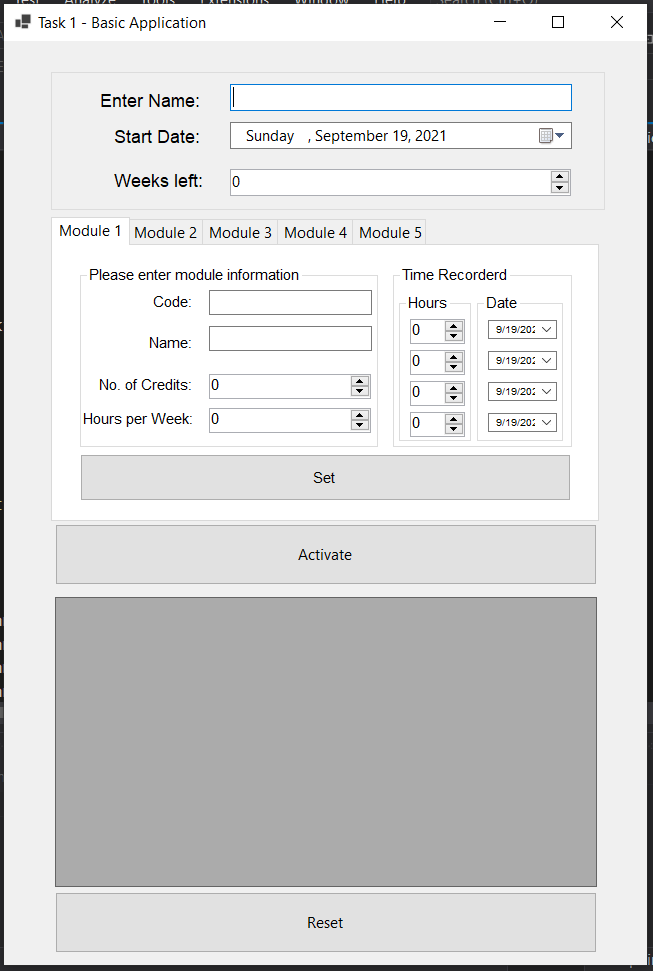
You entered the right data and you are directed to the “Schedule” page.

The option to go back to the register page is in the form of the test of “Need to register”

The Remember me checkbox is there to give it a more authentic feel, even if for now there is non functioning and only here for appearance sake.

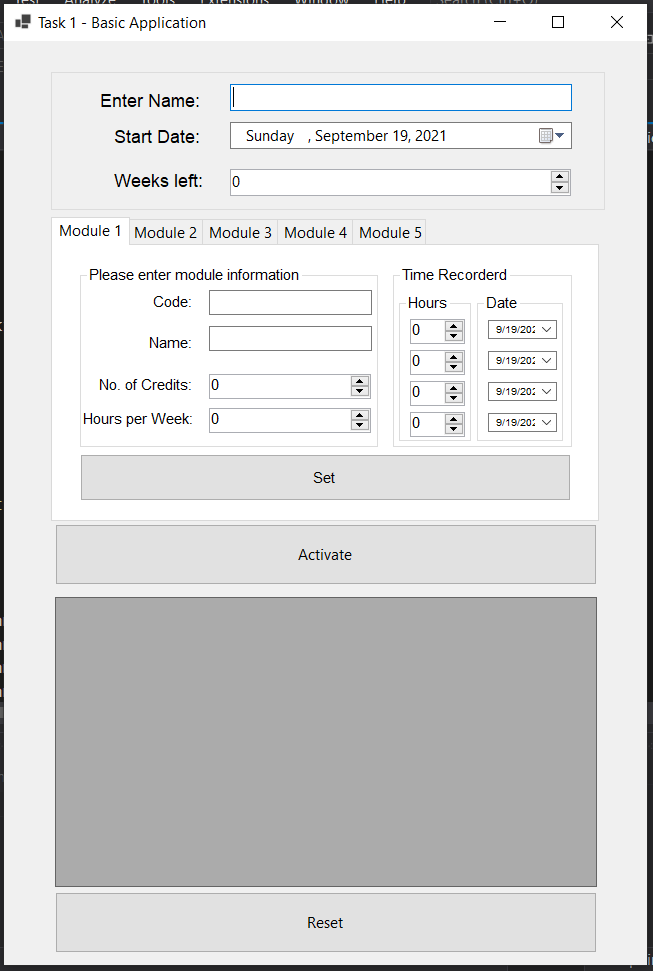
# Schedule

The meat of this application is this schedule option, in comparison with the old version, I have needed to remake it in an HMTL format.

Here is what they look like now and formerly.

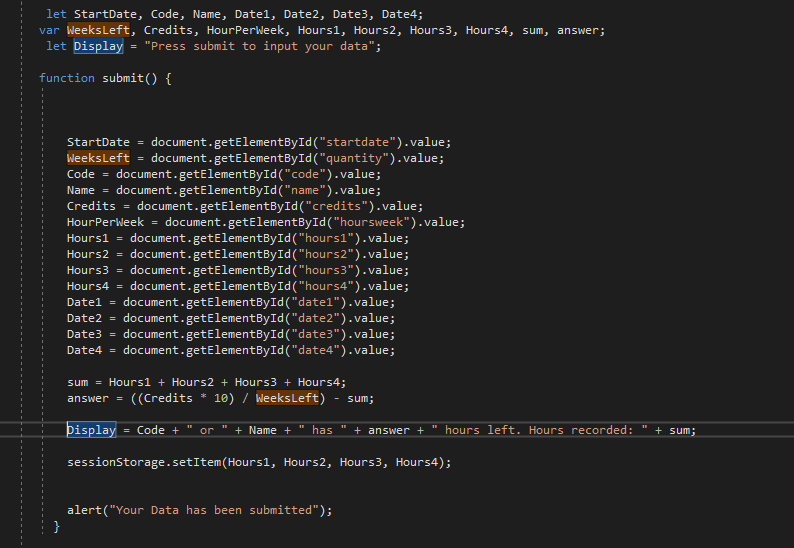
Although it took a fair bit of work on my end to create something that is functionally identical and aesthetically the same but adapted into a new style.

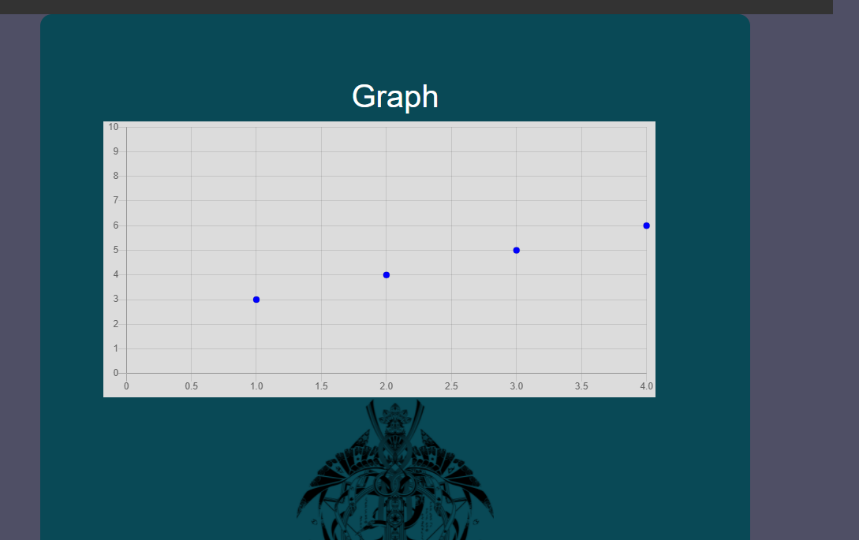
It works like the old one, please see my previous explanations in my Task 1 project if you would like to rehearse how it operates. And while the amount of tables utilized to create the similar appearance needed, nothing more is changed, but a link to a graph function.



# Graph

Using a session storage function, four previous data noting how many hours spent are stored.



These focal points make up the points needed to make a graph based on it.

It’s not extravagant, as is more of less a proof of concept implantation. That said, there isn’t exactly much more one can do with it but the basics. Even with the data , any more extravagant ideas based on the source data elude me.

# Extract for Task 1

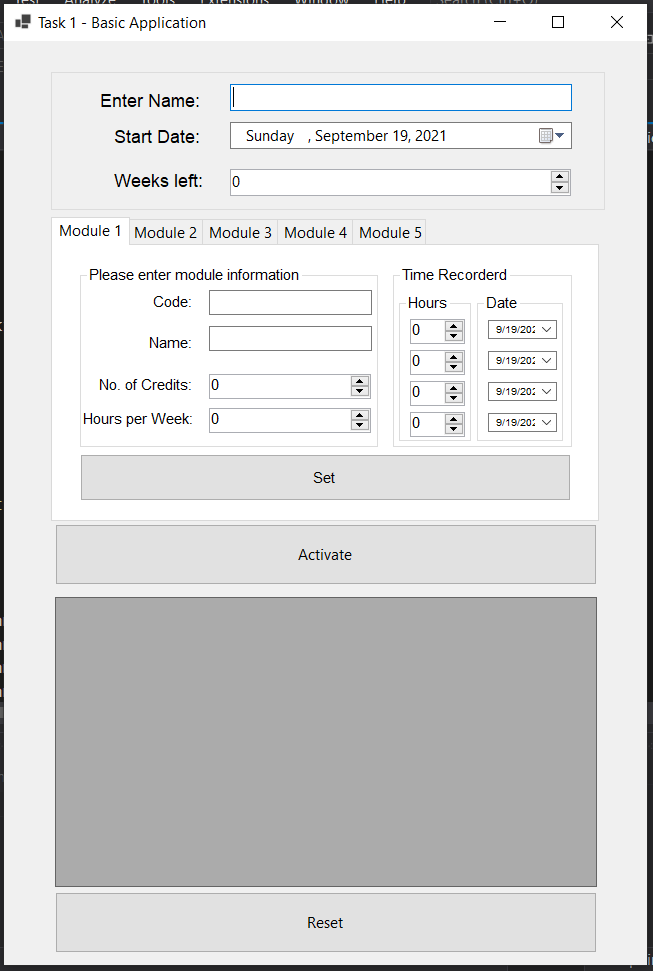
This is an extract from my previous project that explains how the old schedule works.

This is mainly to help familiarize with it.

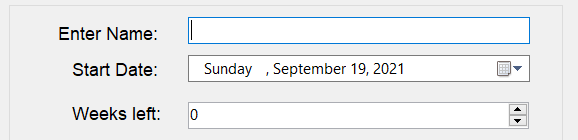
Please ignore it, if you are familiar with it. I do not consider this part of the 1500 word limitation as it is just a review of my previous project and should not count.

Especially noting that the instructions specifically mentioned that the data does not need to be saved outside the activation of the program, I interpreted this as that the need to include various databases or saving information unnecessary.

He basis design is shown.

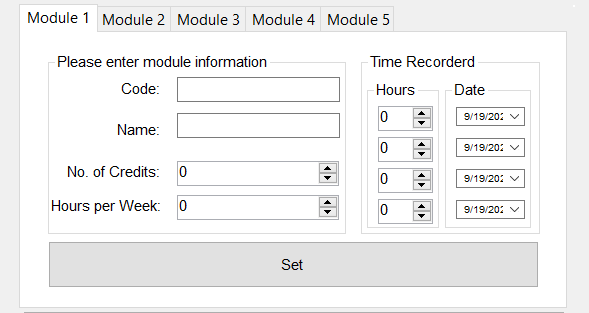


Notably, at various points the instructions deem that I place several areas to handle date and time input, however they don’t seem to have a functional purpose beyond that, and they user can enter the data but it does not have any effect on the program their initial recording.



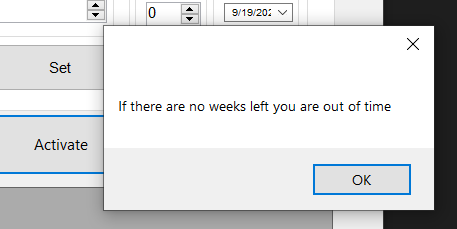
Enter the user’s name, followed by the start date and weeks left.

The theme of reason of why these entries are grouped together is they are universal constants throughout the modules and are not necessary to be changed when inputting new entries.



For convince sake, I have opted to include tabs to enter more entries, one reason is the simplicity to make it easier to use, another reason is as we are recording modules, the amount of entries typically so high an amount that can’t be handled by such a format. Modules do not stray in number by much than 4 or so.

The vital part of this, is that the correct data can only inputted to achieve the required result, so the various input boxes have been set to only allow the correct values to be accepted. If an improper value is put in, a pop-up message shall come in to play and inform the user as so.



One can switch tabs to input different entries, where identical interfaces are ready to be used.

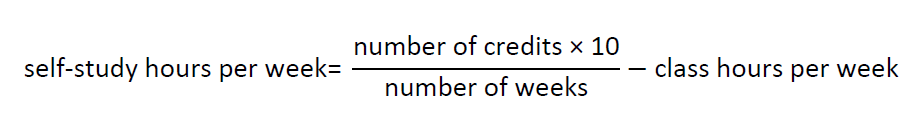
One must always remember to press the set buttons afterwards. This button records the data into the programs and without pressing it, the data will not be saved and will be ignored in the final result.

Another feature is when a module is set, the tab name displayed will be changed to reflect that, ensuring an easier time organizing and will be helpful identifying the tab if one wants to go back.



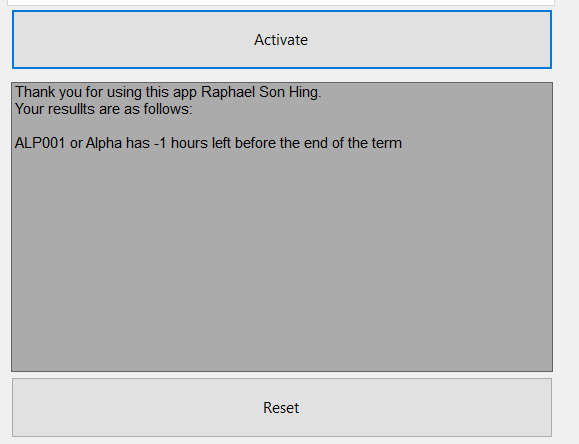
The “Activate” button compiles all the data inputted so far and displays it in the box below. Information for tabs that are left blank are left out to prevent clutter and confusion.

Utilizing the following required formula provided as stipulated.



Information is derived and put into effect.

An example of the display is as follows.



The reset button coverts all the information included in the tabs into a clean state so that the user can restart from the beginning.

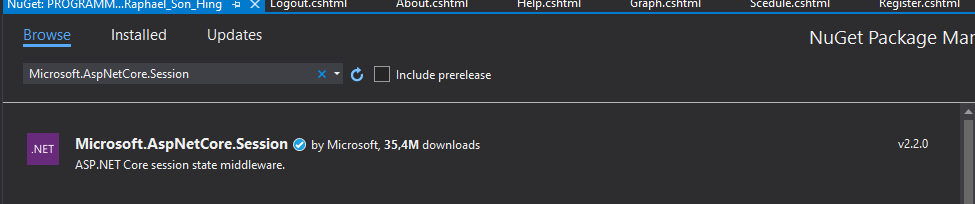
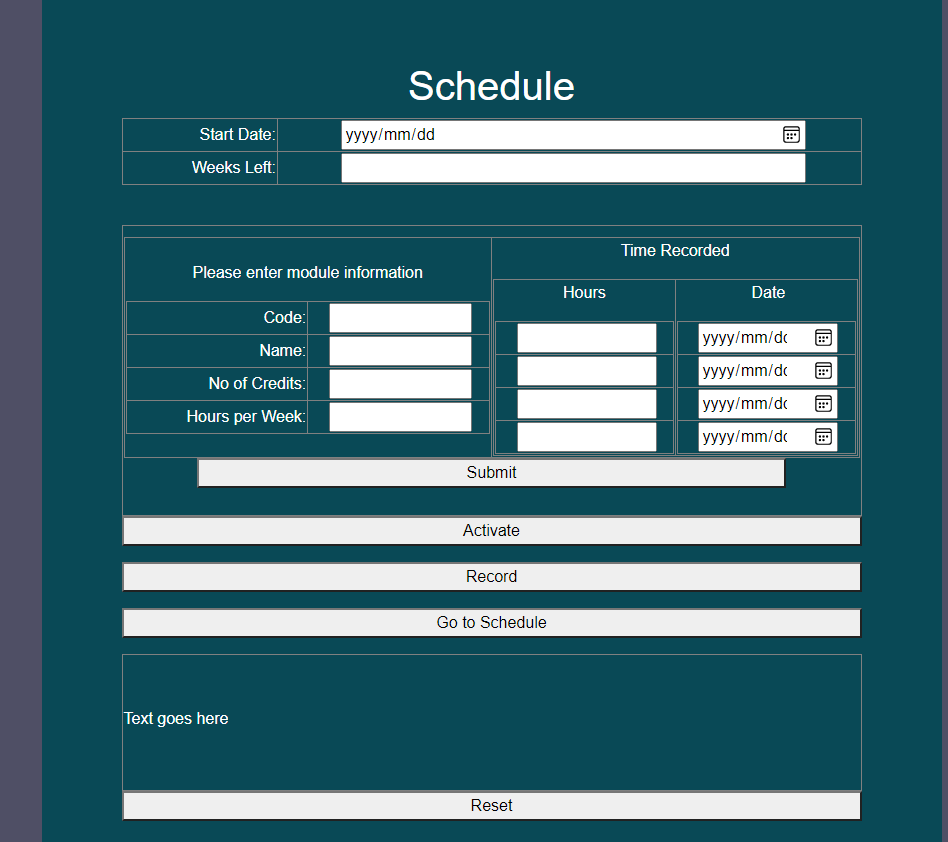
Although one can just go back and update the information for the same result, alternately this option an instead reverts them.

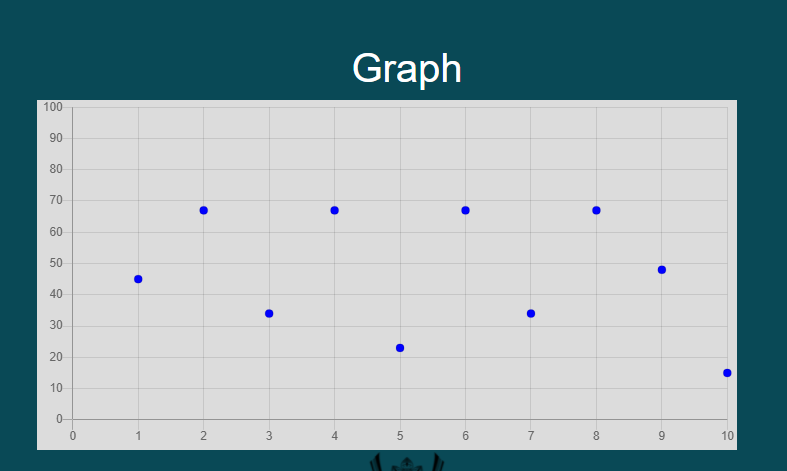
# Conclusion

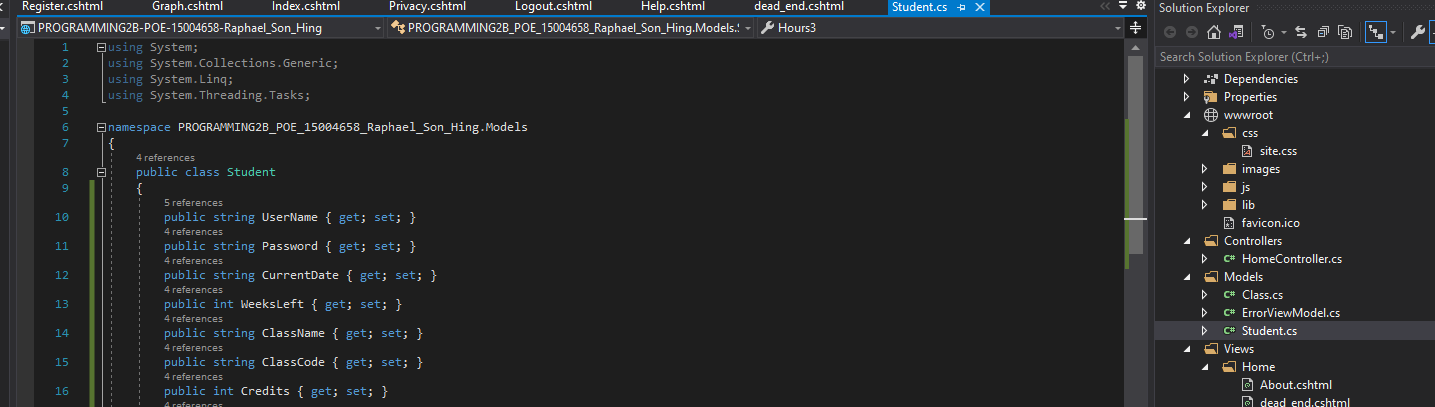
This concludes the report of the main series of the application, there are some bonus areas like the help section and such, but that’s nothing critical to the to program.

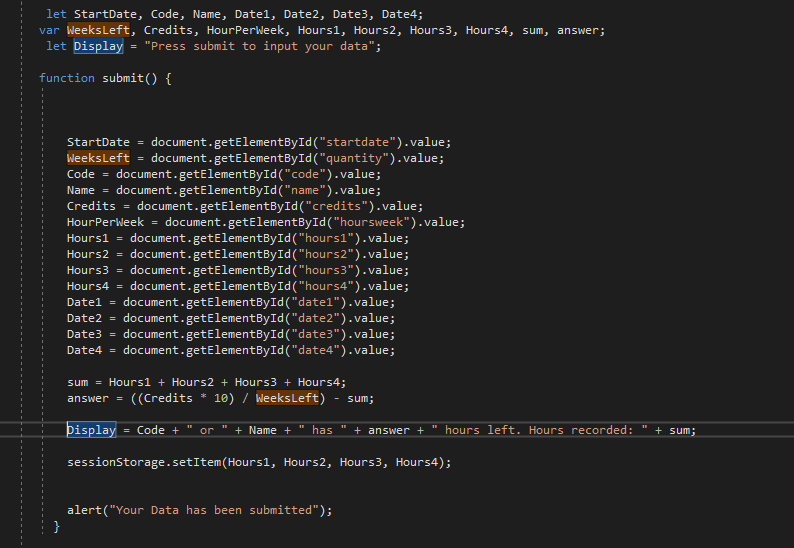
In this endeavour I believe the most important factor is that practice of converting the programming languages from one platform to another and the troubles of finding alternating solutions when programming languages are not compatible with each other at a base level.

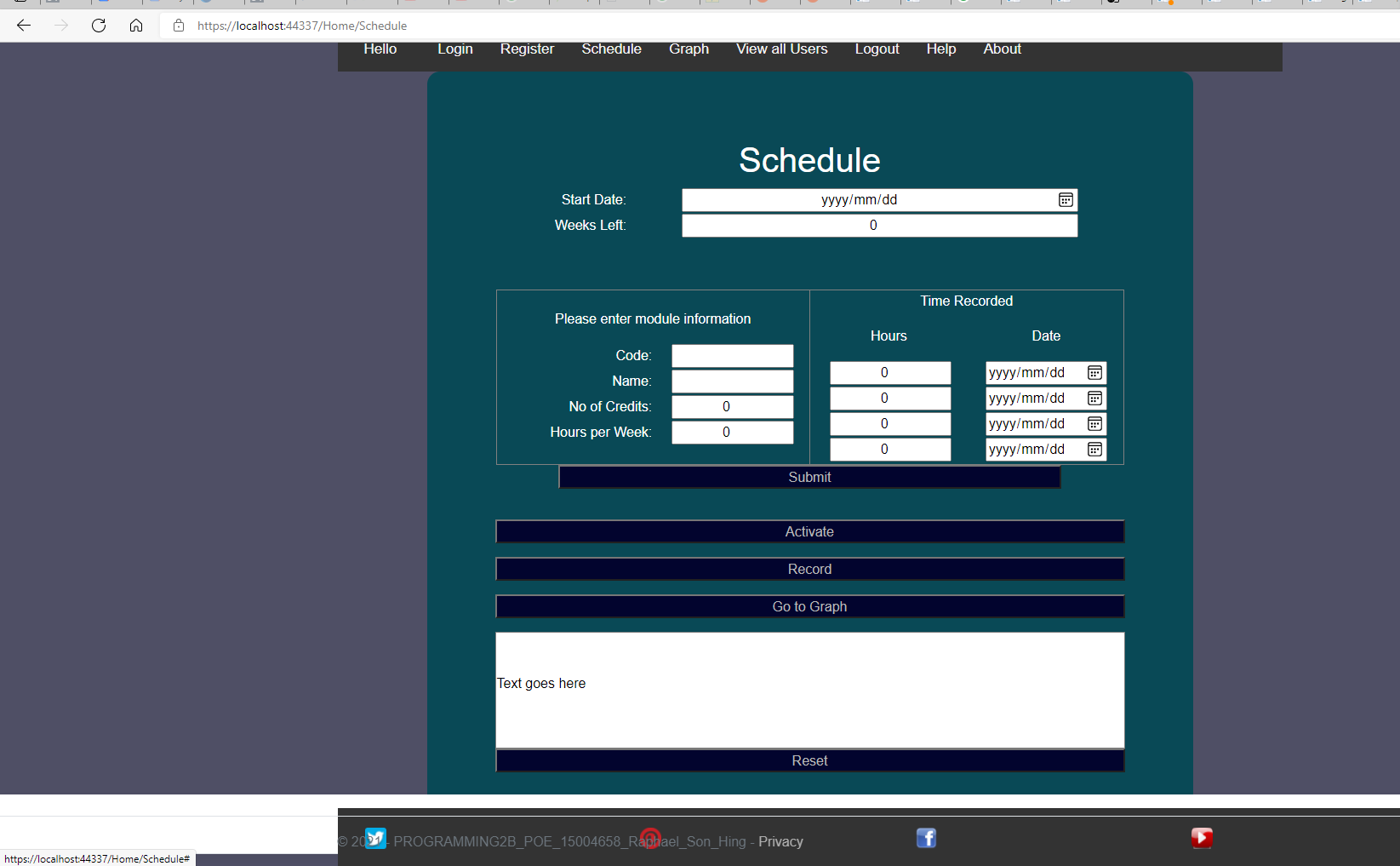
# Various Screenshots









# References

Unknown. (2018, May 18). *DBMS Database Models*. Retrieved from studytonight.com: https://www.studytonight.com/dbms/database-model.php

Unknown. (2018, May 18). *What are Databases? - Examples & Types*. Retrieved from study.com: https://study.com/academy/lesson/what-are-databases-examples-types-quiz.html

Unknown. (2018, May 18). *What is a Database Model* . Retrieved from lucidchart.com: https://www.lucidchart.com/pages/database-diagram/database-models

Unknown. (2021, May 1). *Registration Forms*. Retrieved from Jotform: https://www.jotform.com/form-templates/category/registration

Zameer, A. (2007, December 13). *PHP read text file and insert into MySQL database*. Retrieved from Web Develoment Blog 24: https://www.wdb24.com/php-read-text-file-and-insert-into-mysql-database/#google\_vignette