Name: Malik Mannan
Student Number: ST10091422
PROG6212 POE Manual

1. INTRODUCTION

The application is a complete solution developed to suit the different demands of students managing their academic calendars in an efficient and effective manner. This system, which has evolved from a single desktop program to a flexible ASP.NET Core web application, allows users to access their academic data from any device with an internet connection, allowing for flexibility and simplicity in time management.

Initially designed as a desktop program, this Time Management program sought to assist users in properly arranging their academic duties. Users could add, track, and manage modules for each semester, as well as record study hours and calculate self-study requirements, using the desktop version. Recognizing the need for a more adaptable and user-friendly solution, we developed the program into a web-based platform.

The switch to an ASP.NET Core online application has several benefits, including the ability to access academic data not just from desktop computers but also from laptops, tablets, and smartphones. The online application keeps all of the desktop program's functionality, such as user registration, secure login authentication, and the ability to organize and monitor study modules.

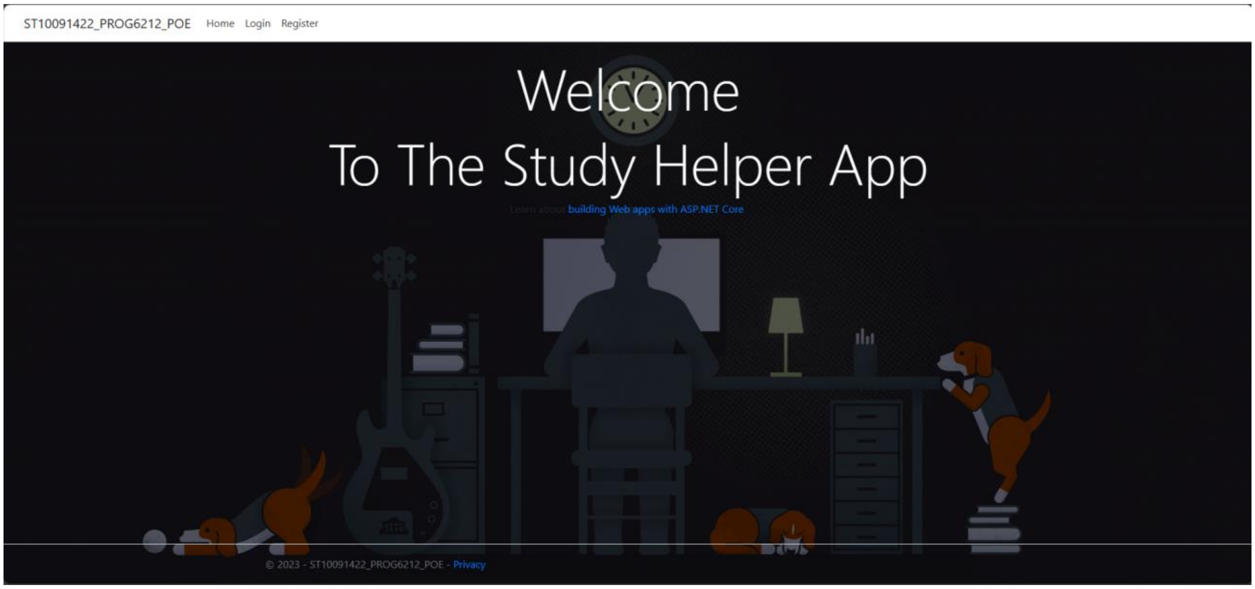
In addition to the main functions, we've included a fascinating new feature to improve the user experience. Users may now see their study trends over time via an interactive graph that displays the amount of hours spent on each module each week. This tool gives useful information regarding study patterns, allowing users to make educated decisions to improve their learning tactics.

2. THE STUDY TRACKER APPLICATION

The database is critical for maintaining user information, program specifics, and study habits. With a focus on data security, the software uses hashing methods to keep only password hashes, protecting user information. The database schema is defined using Structured Query Language (SQL) scripts, which ensures a well-organized and standardized structure that supports the complicated interactions between users and their related study modules.

In addition, the interactive graph feature is driven by JavaScript libraries, notably Chart.js, and provides a simple and visually appealing display of study hours over time. This enhancement improves user insights into their study patterns and enables informed decision-making.

To summarize, the Time Management Application is built on a foundation of technologies such as ASP.NET Core, Entity Framework Core, SQL, and JavaScript libraries to provide a strong, secure, and user-centric solution. This convergence of technologies demonstrates our dedication to remaining at the forefront of innovation while providing our users with a smooth and rewarding experience.



Welcome
To The Study Helper App

Learn about building Web apps with ASP.NET Core

© 2023 - ST10091422_PROG6212_POE - Privacy

Figure 1: study tracker applicaton

To access the application the user must register or login.

2,1, Registration

The Application's registration page serves as a portal for users to access personalized features. The website, built with a user-centric design, prompts users to safely establish an account. The registration procedure requires entering a unique username and password using ASP.NET Core's sophisticated authentication methods. The program uses hashing techniques to save only password hashes to ensure the secrecy of user credentials. Users are guided through the registration process by clear and intuitive form fields, resulting in a smooth onboarding experience. The signup screen exemplifies the application's dedication to user-friendly design and data security, providing the groundwork for a customized user experience.

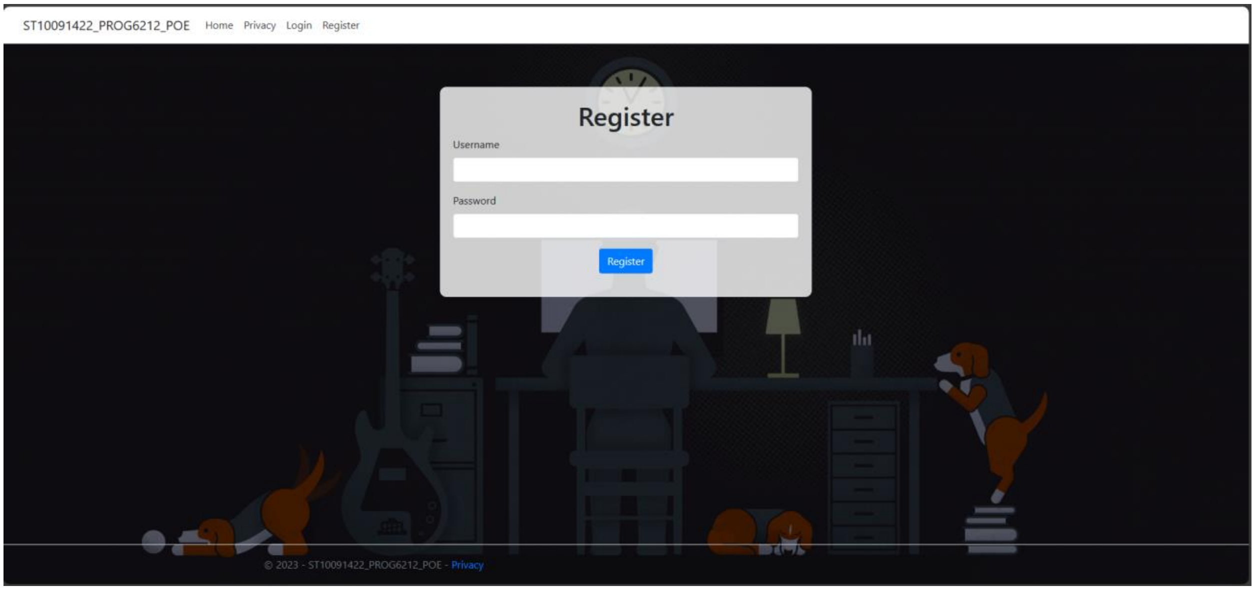


Figure 2: register page to allow the user to be

2.2. Login

The Application's login page offers users a safe and simple access point. Using ASP.NET Core's authentication capabilities, users enter their registered username and password to access their unique data. For increased confidentiality, the program implements strong security mechanisms, saving only password hashes. The login procedure is intended to be easy, with clear prompts and user-friendly components allowing a seamless authentication experience. Users obtain quick and secure access to their academic courses and study data through this login screen, demonstrating the application's dedication to both usability and data security. The login page acts as a portal to customized, efficient, and secure user data.

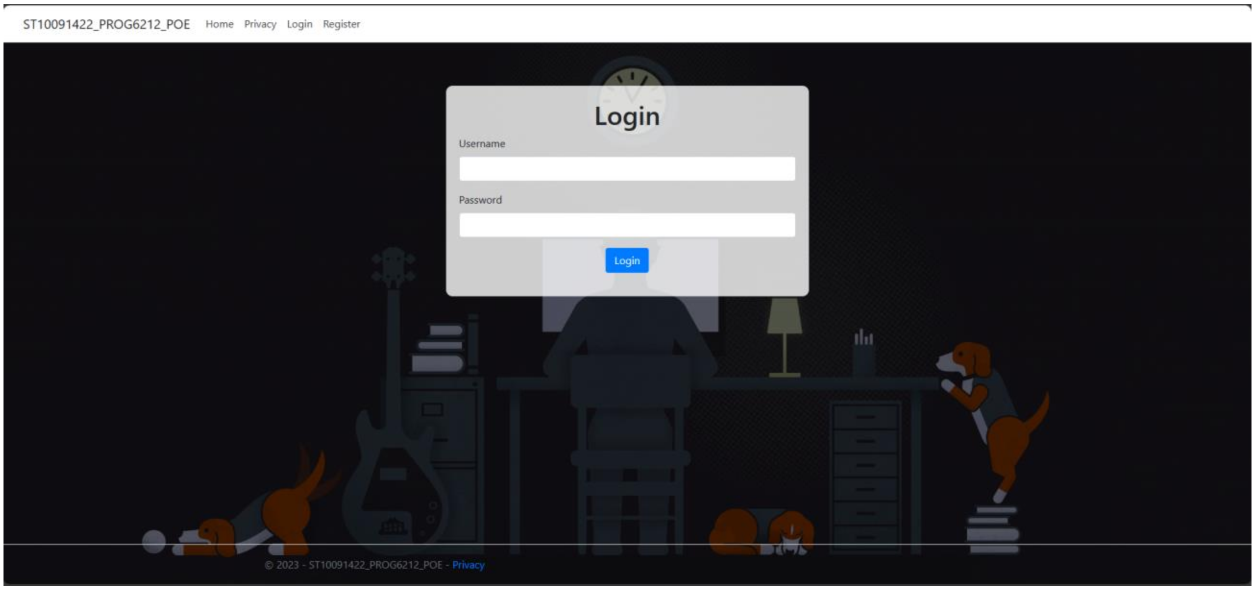


Figure 3: Login page for sign the user in.

2.3. Entering semester data

The new semester creation page is a critical tool for users to smoothly launch and arrange their academic periods. This user-friendly interface allows users to enter important semester information such as the start date and number of weeks. Using the advantages of ASP.NET Core, the page guarantees a responsive and dynamic experience, making it simple to create numerous semesters. Users can customize the program to their own academic schedules by setting the settings for each semester. This dedicated page epitomizes the application's dedication to user-centric design, providing users with a concentrated location to conveniently plan and manage their academic obligations.

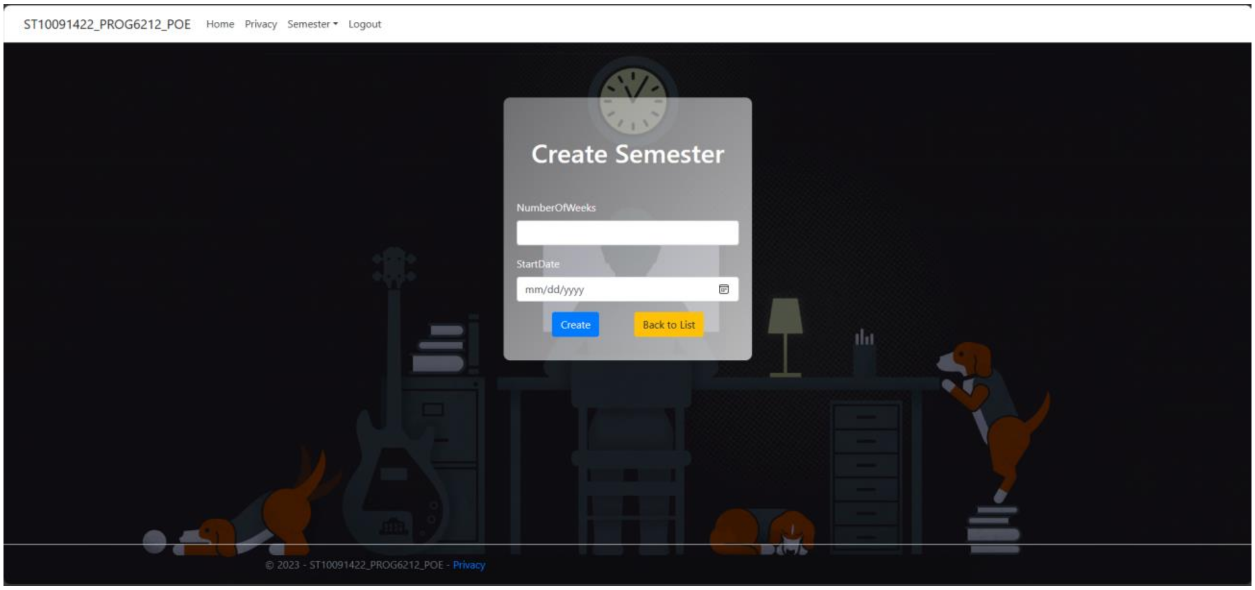


Figure 4: input page of new semester

2.4. Entering module data

The method of entering material for a new module is carefully planned to guarantee a thorough and orderly approach to academic preparation. Users are prompted to establish a semester before adding a module, laying the groundwork for their educational period. This systematic process helps users to deliberately explain their academic obligations within a given timeframe.

Once a semester has been constructed, users may easily enter module information such as the module code, name, credits, and weekly class hours. For this aim, the ASP.NET Core integrated module creation page provides an easy-to-use interface that allows users to build and personalize their academic modules.

The application's novel feature is the automated computation of self-study hours each week. This estimate, based on the credits entered, gives users real-time insight into the weekly study commitment necessary for each subject. The program provides a great tool for users to design and improve their study schedules by dynamically evaluating the number of weeks in the semester and subtracting the class hours each week. The systematic method of needing a semester before adding modules provides users with a well-defined framework for managing their semester data.

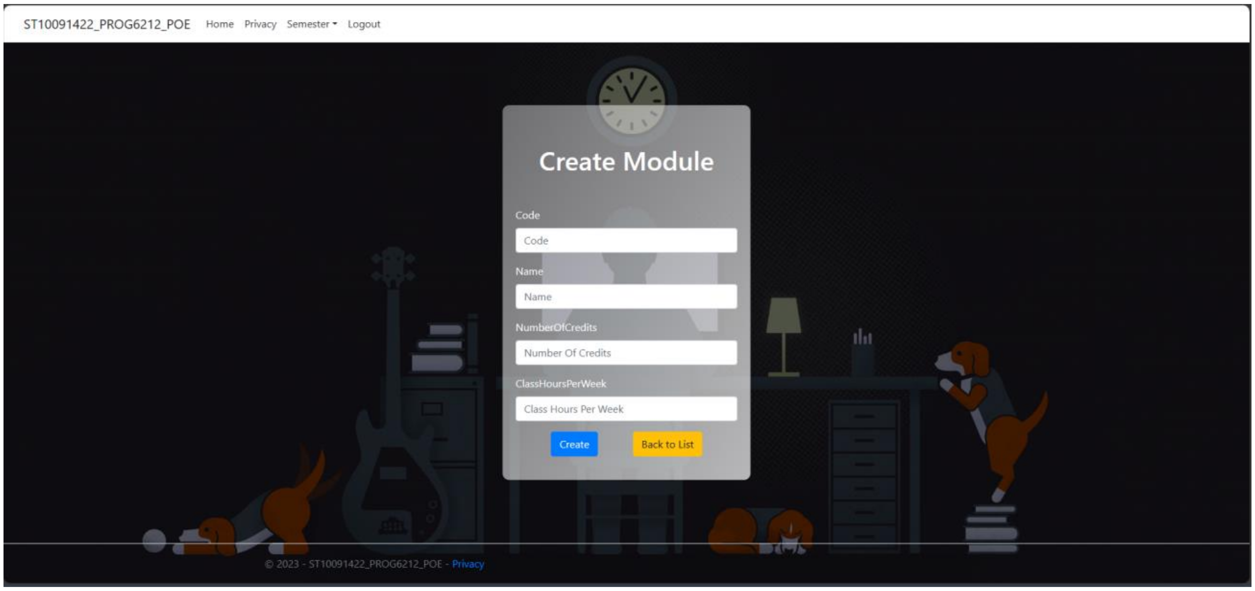


Figure 5: input page of adding a module

2.5. updating number of hours spent on a module

The system dynamically evaluates the recorded study hours on certain days throughout the current week to determine the remaining self-study hours for a module. Users gain real-time insights into their remaining study obligations for each module by deducting this cumulative time from the projected weekly self-study hours. This procedure guarantees that users may organize and allocate their time effectively within the framework of their current academic week.

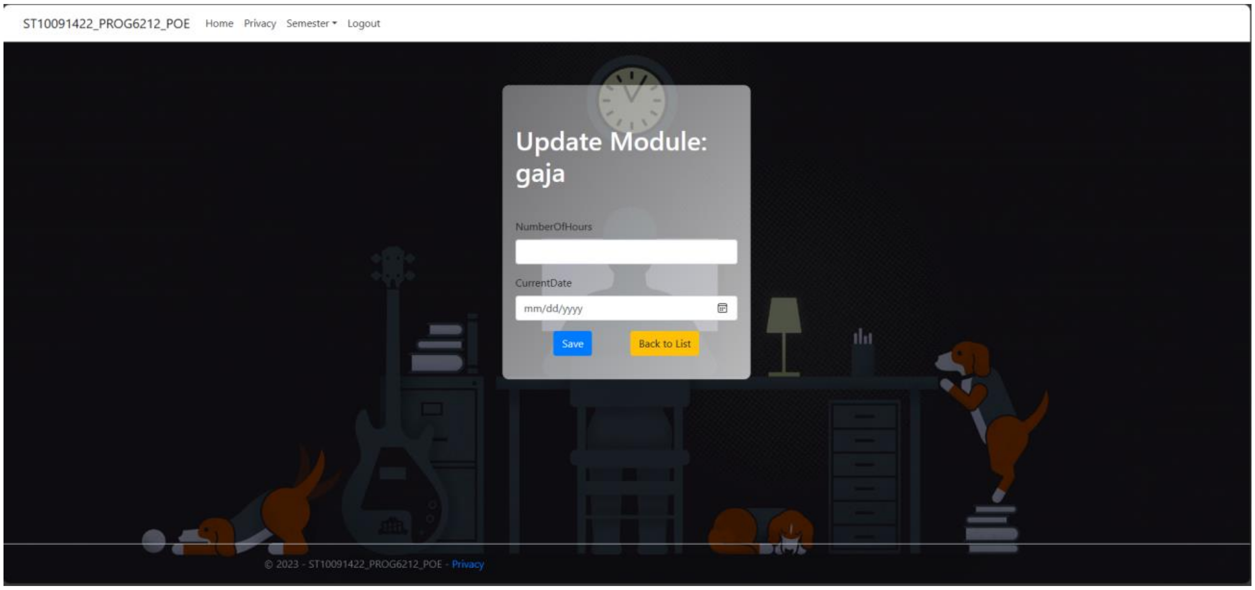
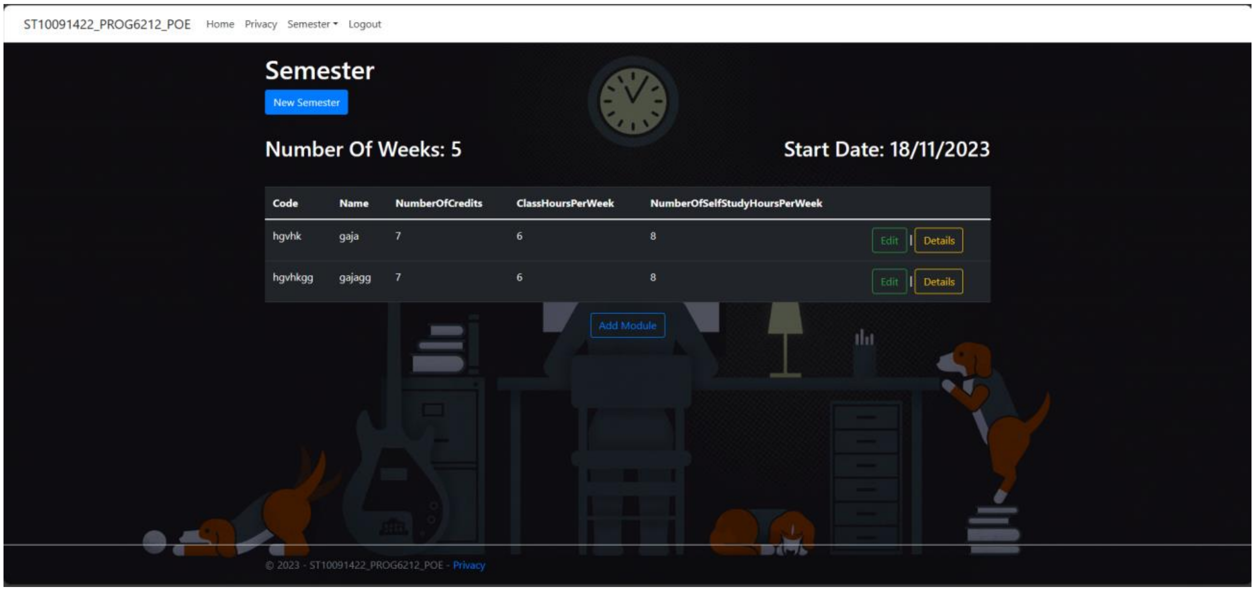


Figure 6: input page to enter number of hours and current date to update the remaining hours

2.6. page view semester and modules data

The Application's customizable dashboard gives a glimpse of the user's academic term, including the semester start date, the total number of weeks, and an ordered list of the user's enrolled modules. Figure 7 below shows how the dashboard looks.



Semester

Number Of Weeks: 5

Start Date: 18/11/2023

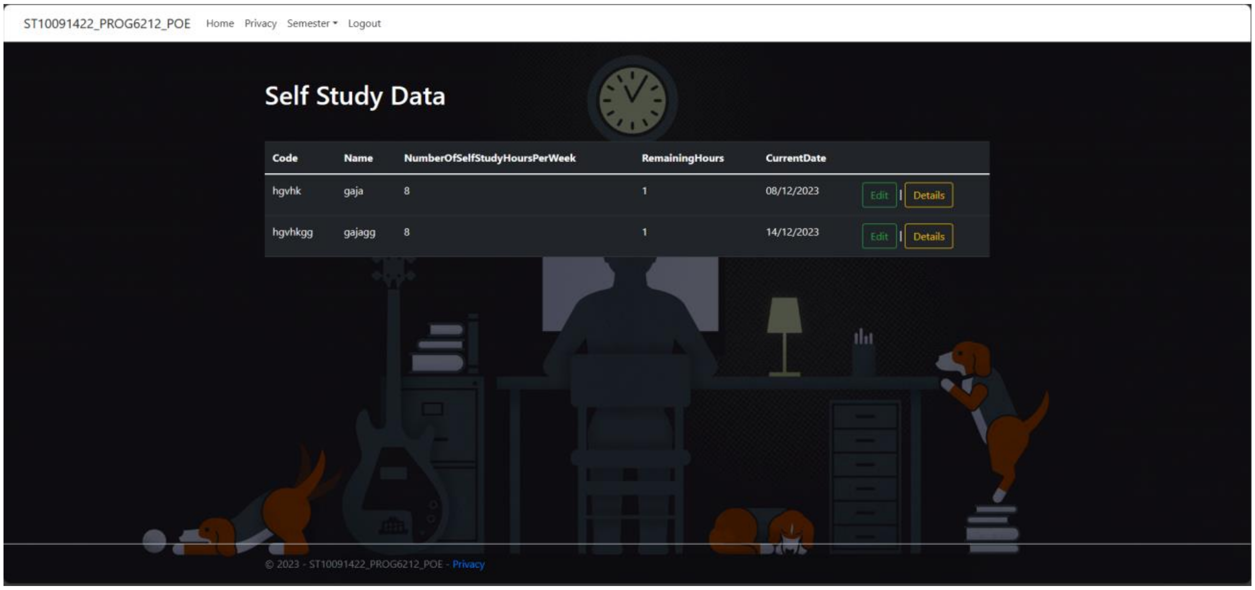
Code	Name	NumberOfCredits	ClassHoursPerWeek	NumberOfSelfStudyHoursPerWeek	
hgvhk	gaja	7	6	8	Edit | Details
hgvhkgg	gajagg	7	6	8	Edit | Details

© 2023 - ST10091422_PROG6212_POE - Privacy

Figure 7: dashboard of semester data

2.7. view of data that contains

The application provides a dashboard that displays data displaying the remaining hours of the module the data the user last worked on the specific module. Figure 8 below shows the specific dashboard.



Self Study Data

Code	Name	NumberOfSelfStudyHoursPerWeek	RemainingHours	CurrentDate	
hgvhk	gaja	8	1	08/12/2023	Edit | Details
hgvhkgg	gajagg	8	1	14/12/2023	Edit | Details

© 2023 - ST10091422_PROG6212_POE - Privacy

Figure 8: dashboard containing self-study data

2.8. Graph

The graph component of the application provides a visual depiction of study trends across time. Users obtain useful insights from a live display of weekly hours spent on each module, boosting their capacity to optimize study habits and enhance overall time management methods.

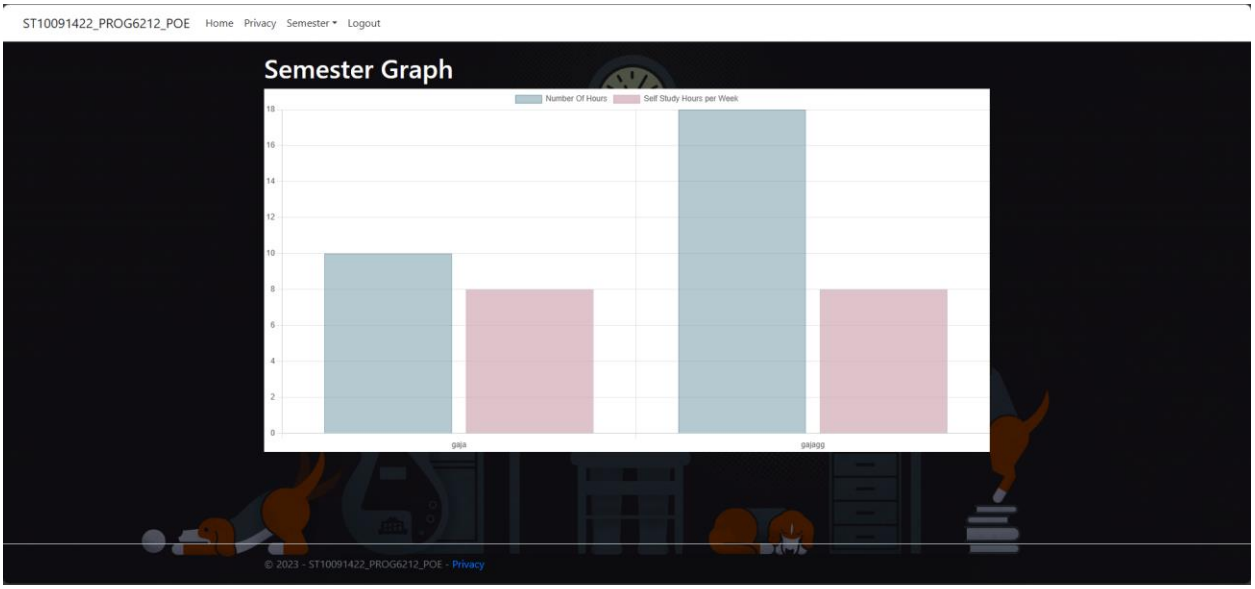


Figure 9: Graph out of the ideal hours and the hours spent

3. CONCLUSION

Finally, the transformation of the Time Management Application from a single desktop solution to a dynamic ASP.NET Core online application marks a significant advancement in improving user accessibility and supporting effective academic planning. The move to a web platform not only assures smooth access across several platforms, but it also offers up new avenues for collaboration and participation.

The basic capabilities of the desktop version remain intact, allowing users to easily register, log in securely, and manage their academic modules. The interactive graph feature enhances the user experience by offering a visual depiction of study trends over time.

As we embrace the digital era, our online application demonstrates our dedication to providing creative solutions that adapt to consumers' changing demands. The Time Management Application continues to empower students and educators alike in their quest of efficient time management and academic achievement by combining the strength of a bespoke class library, ASP.NET Core technology, and user-friendly features. We are excited about the continuous growth of this application, which is motivated by a dedication to excellence and a desire to improve the educational process.

Reference List

Chao, L. 2014. Cloud Database Development and Management, CRC Press.

Ervis Trupja (2021) 29. Adding your first service | ASP.NET MVC, 15/08/2021. [online] Available at: https://youtu.be/zvkSIOs6s-M [Accessed 27 December 2023]

Payload (2021) WPF C# Professional Modern Flat UI Tutorial, 3 April 2021. [online] Available at: https://youtu.be/PzP8mw7JUzI?si=m_tnXOm6f8uKWyQv [Accessed 27 December 2023]

Patrick God (2023) .NET 7 Web API Role-Based Authorization with JSON Web Tokens (JWT) & the dotnet user-jwts CLI. 14 February. Available at: https://youtu.be/6sMPvucWNRE [Accessed 27 December 2023]

Microsoft.learn. 2023. Tutorial: Create a .NET class library using Visual Studio Code, 9 July 2023. [online] Available at: Create a .NET class library using Visual Studio Code - .NET | Microsoft Learn [Accessed 23 August 2023]

tutorialspoint.com. n.d. ASP.NET MVC - Controllers. [online] Available at: https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_controllers.htm [Accessed 27 December 2023]

c-sharpcorner.com. 2021. Cookie Authentication In ASP.NET Core, 17 May 2021. [online] Available at: https://www.c-sharpcorner.com/article/cookie-authentication-in-asp-net-core/ [Accessed 27 December 2023]