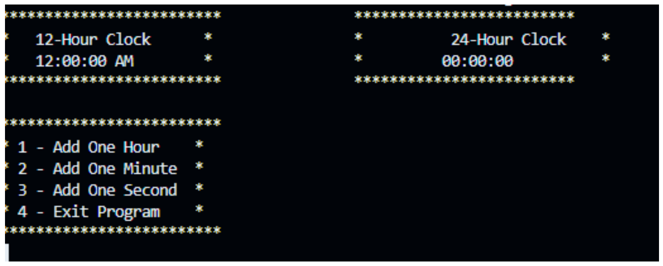
3-2 Milestone Two: Enhancement One (Software Engineering and Design)

1. Briefly describe the artefact. What is it? When was it created?

This clock application was created as part of my CS 210 - Programming Languages class during my undergraduate program. It is a static 12 and 24-hour clock created in C++, which allows users to manipulate the time that is shown on it by an hour, minute or second using numbers as inputs (either 1, 2, 3 or 4 to exit the program). To make the program accessible and easy to run, I have containerised my program on Docker. In doing so, I've effectively eliminated the need for anyone who is interested in collaborating or adding to my program (either the original one or the enhanced one) to install any extensions, plugins or special software that is additional to their usual code editor, thereby meeting **course outcome 1, which is to employ strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science**.



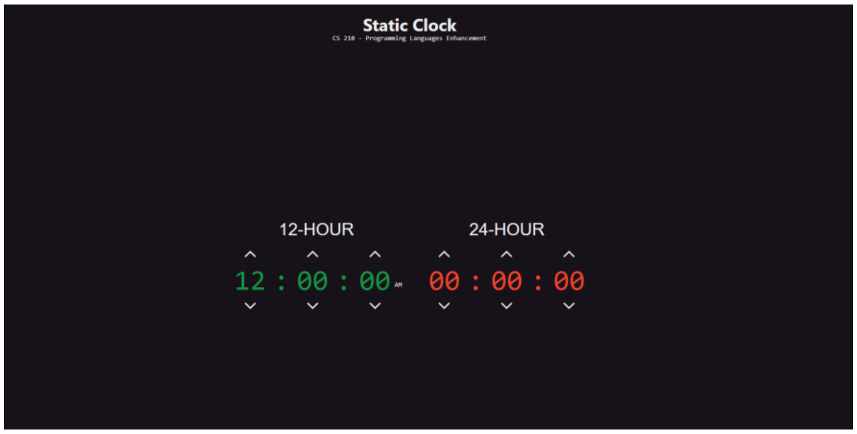
```
**************************          **************************
*   12-Hour Clock      *            *         24-Hour Clock    *
*   12:00:00 AM        *            *        00:00:00          *
**************************          **************************

**************************
* 1 - Add One Hour    *
* 2 - Add One Minute  *
* 3 - Add One Second  *
* 4 - Exit Program    *
**************************
```

Photo: C++ interface (Static Clock) - Photo by Faizah

2. Justify the inclusion of the artefact in your ePortfolio. Why did you select this item? What specific components of the artefact

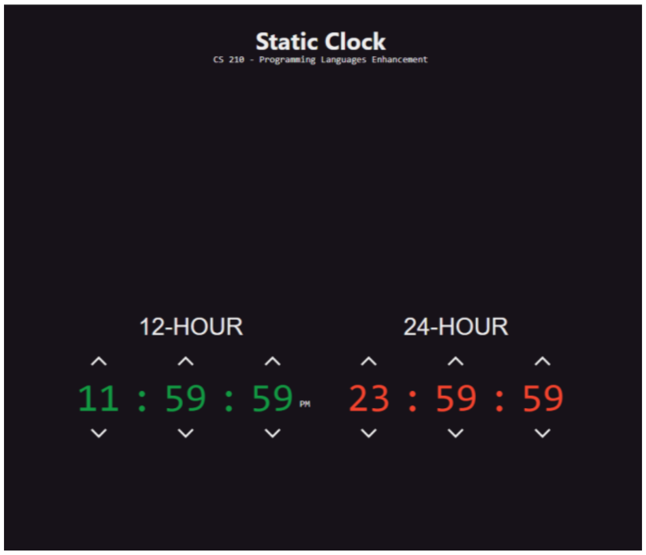
showcase your skills and abilities in software development? How was the artefact improved?

I chose to include this artefact in my ePortfolio as it would have been something quite unique, and it gives me a chance to revisit my old work to see just how far I’ve come. Recently, I’ve been putting a lot of work into learning and improving on my understanding of the React and Vue frameworks, specifically Next JS and Nuxt JS. The decision to rebuild my static clock using one of these frameworks was one that I was excited about as I could give it a visually-appealing front-end and thereby allow more people to use the application for their specific needs.



Screenshot: Static Clock enhancement (Screenshot by Faizah)

Building this static clock on Next JS 14, Tailwind and TypeScript was a good opportunity to highlight my adaptability and proficiency in modern web development frameworks. It also shows that I understand both low-level and high-level programming, and that I’m capable of shifting from a performance-oriented, system-level language like C++ to a more design-friendly and component-based front-end stack. Having translated not only functionality but also ensuring the user interface (UI) is user-friendly and reactive shows my solid grasp of software design and front-end architecture.



Screenshot: Static clock set at 23:59:59. (Screenshot by Faizah).

A specific component of note is that I took into account separation of concerns, which meant that I placed the time logic into the TimeControl component, which handles user interactions for changing the hour, minute and second of the clock. By doing this, I made my application scalable and maintainable, demonstrating reusable code practices. This was done instead of repeating the same code, which would have been cumbersome. I kept my code clean and prevented the UI component from being cluttered with logic, allowing for easier testing and debugging.

Another key element is the use of TypeScript and Tailwind CSS. TypeScript ensures type safety, which improves code quality and reduces bugs. This is a widely adopted practice in the industry due to its added layer of security and clarity. Instead of writing custom CSS for every element, I used Tailwind, which is a utility-first CSS framework. It is efficient, and helps me to maintain a clean and responsive UI . This is consistent with course outcome 4, which is **to demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value**

and accomplish industry-specific goals.

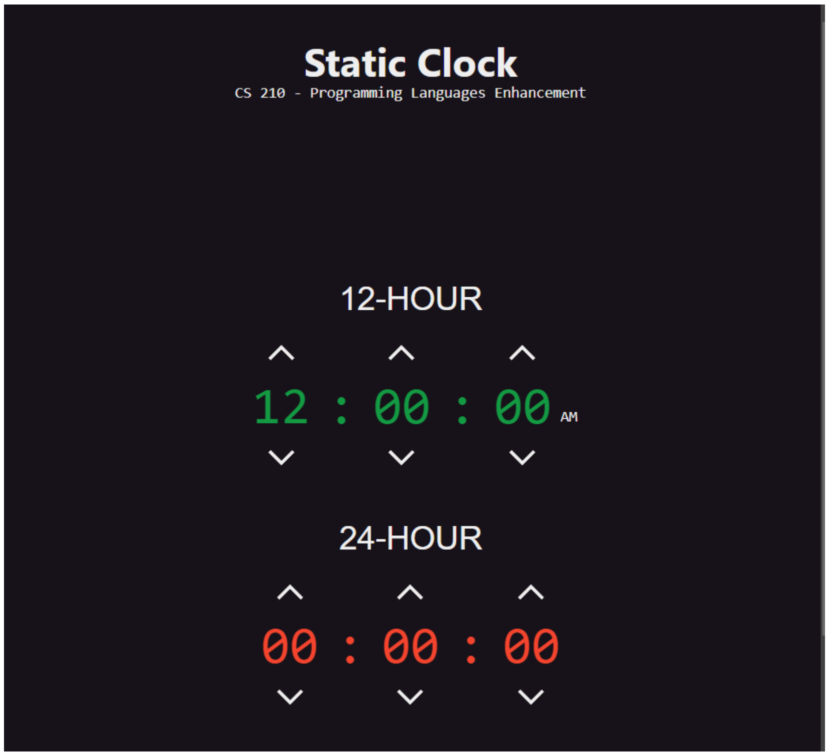
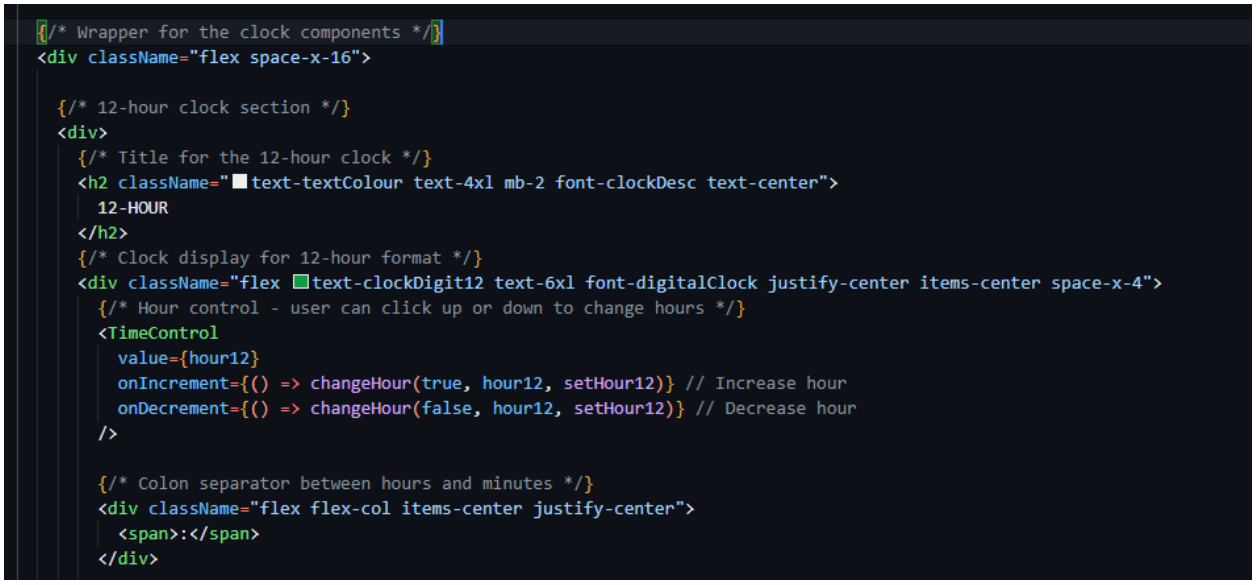


Photo: Screen capture of responsive design. Screenshot by Faizah.



```
{/* Wrapper for the clock components */}
<div className="flex space-x-16">

  {/* 12-hour clock section */}
  <div>
    {/* Title for the 12-hour clock */}
    <h2 className="text-textColour text-4xl mb-2 font-clockDesc text-center">
      12-HOUR
    </h2>
    {/* Clock display for 12-hour format */}
    <div className="flex text-clockDigit12 text-6xl font-digitalClock justify-center items-center space-x-4">
      {/* Hour control - user can click up or down to change hours */}
      <TimeControl
        value={hour12}
        onIncrement={() => changeHour(true, hour12, setHour12)} // Increase hour
        onDecrement={() => changeHour(false, hour12, setHour12)} // Decrease hour
      />

      {/* Colon separator between hours and minutes */}
      <div className="flex flex-col items-center justify-center">
        <span>:</span>
      </div>
```

Photo: Use of TypeScript and Tailwind CSS. Also shows the separation of concerns between the clock logic and UI components. Screenshot by Faizah.

3. **Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

In addition to achieving Course Outcome 2, which is to design, develop and deliver professional-quality oral, written and visual communications (through my extensive README that details information on the project, how to run it and how to use it), my work on this enhancement has also enabled me to achieve Course Outcome 1. This outcome is to use strategies for building collaborative environments to enable diverse audiences to support and contribute to it - through including containerisation and maintaining clear code with comments that direct and explain my code. Separating the clock logic and the front-end code was one of the ways in which I did this - it is easy to access my code and to quickly make changes where they need to be made. I also deployed the app on another platform to generate a link so that people can interact with my project without having to dive into technical steps (to cater to those who don't want to).

Additionally, I have also met Course Outcome 4, in which I utilised the modern Next JS 14, Tailwind CSS and TypeScript to make my enhancement while keeping the original application's functionality the same, or improving it slightly through being able to toggle from AM and PM clocks.

I have therefore met my planned course outcome (2), and additionally met Course Outcomes 1 and 4 through this enhancement. I further intend to add a feature for a live clock - particularly one that can adapt to time zones per a user's request. However, this is external to this enhancement.

4. Reflect on the process of enhancing and modifying the artefact. What did you learn as you were creating it and improving it? What challenges did you face?

When I first started working on this enhancement, I realised that I was making the same mistake from my earlier projects where I combined both the logic and front-end UI into a single, tangled mess. At the same time, it felt easier because I could see everything in one place. Maintaining and updating the code, however, was frustrating. Every time I needed to update something, I had to sift through the endless lines of code.

The turning point for me came through trial and error, and some feedback from my friends. I then understood the concept of *separation of concerns*. Breaking my logic and front-end UI into separate components, I now have a cleaner and more maintainable code. This restructuring wasn't without challenges - understanding how to organise my components effectively took time and a lot of patience. I also had to refactor parts of my codebase, which at times had it break. However, the hard work was worth it as I now have a program that I am proud to share, and hopefully have people collaborate with me on. I can also confidently add new features without worrying about breaking anything else.

My experience with this enhancement has taught me the technical skill of separating concerns and also the importance of keeping my code maintainable for possible collaborations.