Department of Computer Science
COSC 4P02 - Software Engineering - II

Progress Report II

Instructor: Naser Ezzati-Jivan
Date: April 1, 2024

Contents

1 System Summary

Over the past two months we have developed a compact and reliable learning management system. This system has been implemented on a website intended for both desktop and mobile use. Upon visiting the site the user is prompted to *log in*. Once logged in, the user can access the main features which are a *modules* page for watching lectures, a *quizzes* page for testing the user on the material from the lectures, and a *libraries* page which provides a list of resources for the user to further his or her learning. *Accessibility* options are also provided to make the website easier to use.

1.1 Login

The login feature allows users to log in to the system and access features such as the modules. Visiting the website the user will be greeted with the page shown on figure 1.1

Figure 1.1: The home page.

If we try to access the modules page it will ask us to log in, as displayed in figure 1.2.

Logging in can be done by clicking the “Log in” button which is the very right button on the navigation bar, with the bar itself being located on the top of the screen. Clicking it opens a login modal which can be seen in figure 1.3.

There are three providers the user can choose from. We decided not to store login information ourselves and to offload that burden to one of these providers.

1.2 Modules

Once the user selects a provider and is logged in, he or she can now access main features such as the modules tab, as shown in figure 1.4. A number of aesthetic improvements have been made to this page. Much of the CSS throughout the page has been updated to daisyui css to give the page a more consistent look, and also allow for the theme to be changed more easily. Module items now include icons to signify whether they are videos

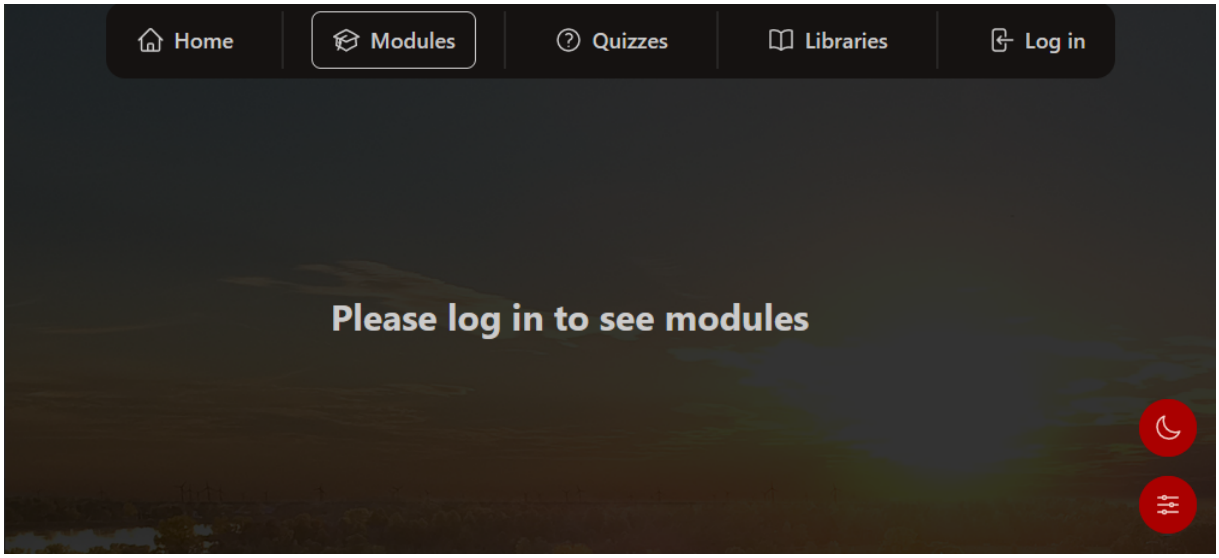

Figure 1.2: The modules page without being logged in.

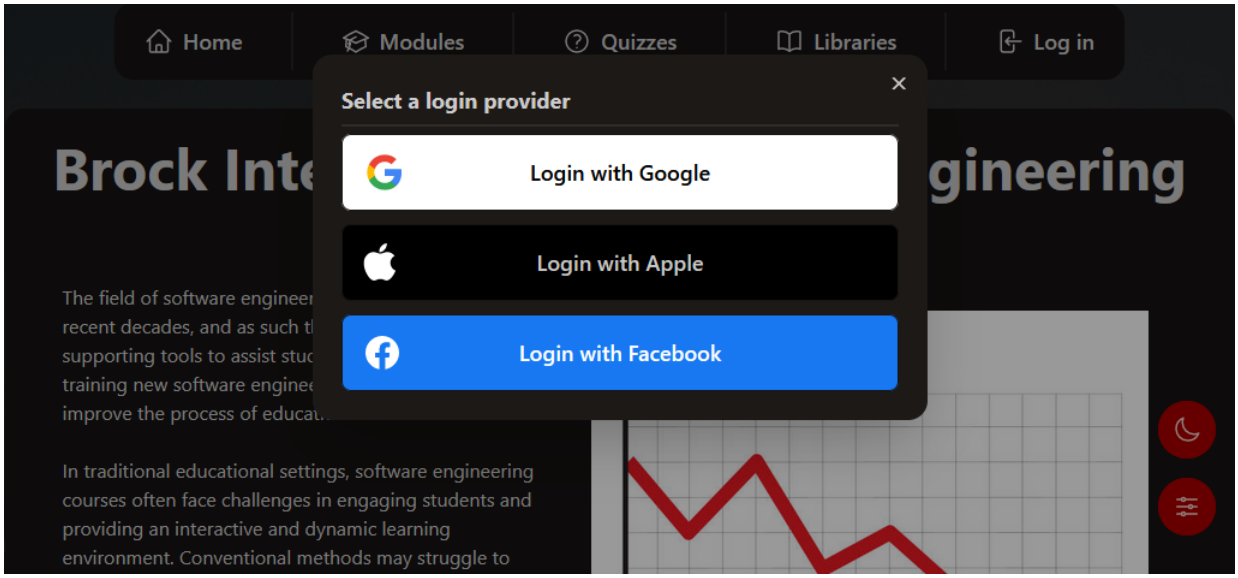

Figure 1.3: The login modal.

or slideshows. The side menu, breadcrumb, and module view have been set to resize relative to one-another so as to keep consistent proportions on different sized screens. Functional improvements have also been made. Upon clicking a module to look through it's items previously the new module would be loaded, now the webpage waits until a user has actually selected an item so they can look through each module without needing to actually open their contents.

1.3 Quizzes

The user can test their understanding of the knowledge learned in the modules page by checking out the quizzes page and doing the quiz questions, which can be seen in figure 1.5. Once a user completes a quiz, that quiz will be immediately graded and then put up on a leaderboard which compares how all the users did on the quiz. The leaderboard can be accessed by clicking the "Toggle Leaderboard" button seen in figure 1.5. The

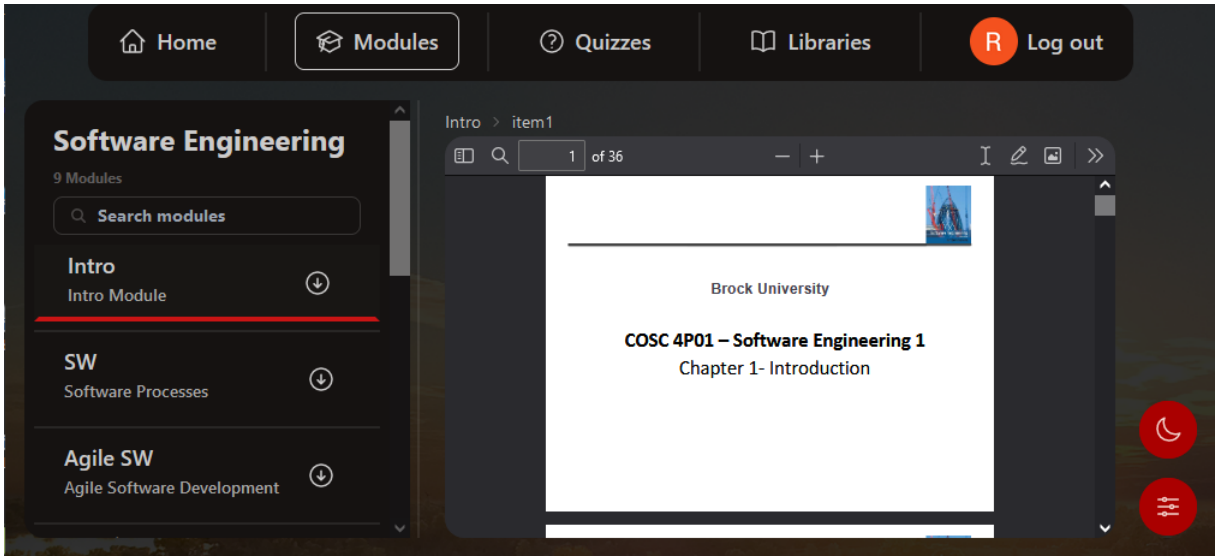

Figure 1.4: The modules page now that a user has logged in.

leaderboard itself can also be seen in 1.7.

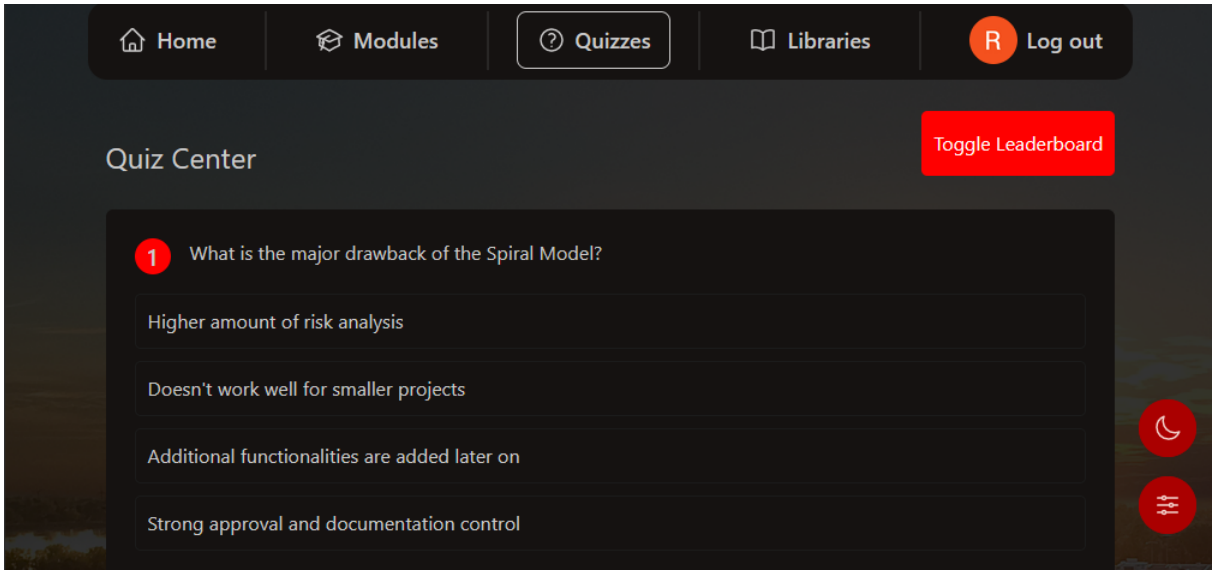

Figure 1.5: The quizes page.

1.4 Libraries

There is also a page called *libraries* which acts as a list of resources the user can use for further learning. The resources here are related to the material from the modules, but there are also things that are not directly related to the material like how to use specific tools such as git and docker. These topics are split off from each other in different tabs, as can be seen in figure 1.6.

1.5 Accessibility

An accessibility menu was implemented as a modal menu that can be opened by clicking a button that floats at the bottom-right of the screen labelled with an icon as depicted

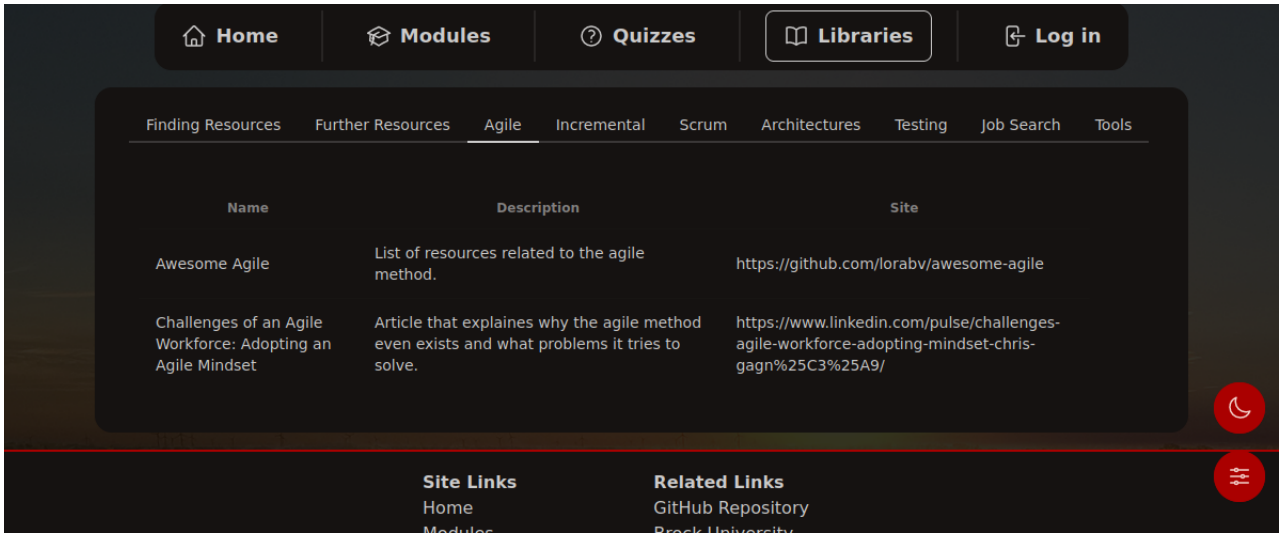

Figure 1.6: The libraries page.

in figure 1.8. Once opened it includes options for selecting a site theme, resizing the website's text, and displaying a list of keyboard shortcuts. Selecting a theme can be done either by toggling between the default dark and light themes of website by clicking "Toggle Theme" or by selecting a specific theme in the drop-down menu as seen in 1.9. More on these themes will be discussed in section 1.6. The websites font size is updated to scale with the slider on the menu.

1.6 Overall Aesthetics

The website received some front-end aesthetic changes that effected all pages. These include a new image of Brock as a background image to make the site feel less flat, and themes. Themes can either be changed in the accessibility menu, or using the toggle theme button above it (see figure 1.8). The toggleable button shifts the page to and from the default dark and light themes, while the accessibility menu's theme selector allows changing to additional non-default themes.

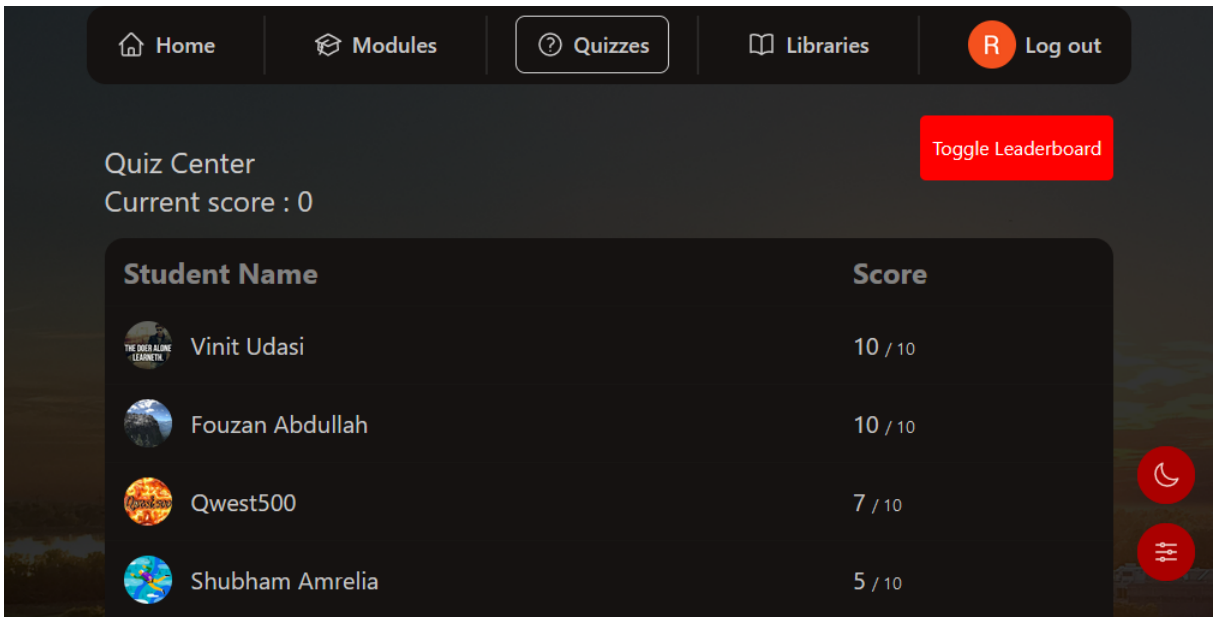

Figure 1.7: The modules page now that a user has logged in.

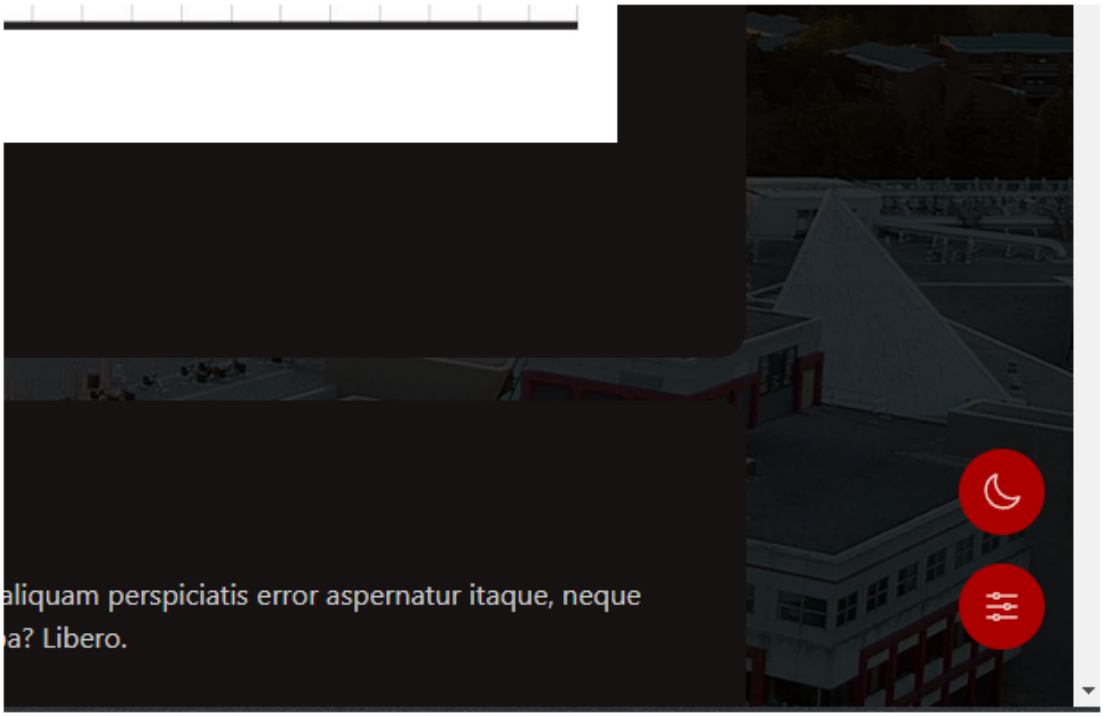

Figure 1.8: The accessibility menu and dark/light mode toggle button.

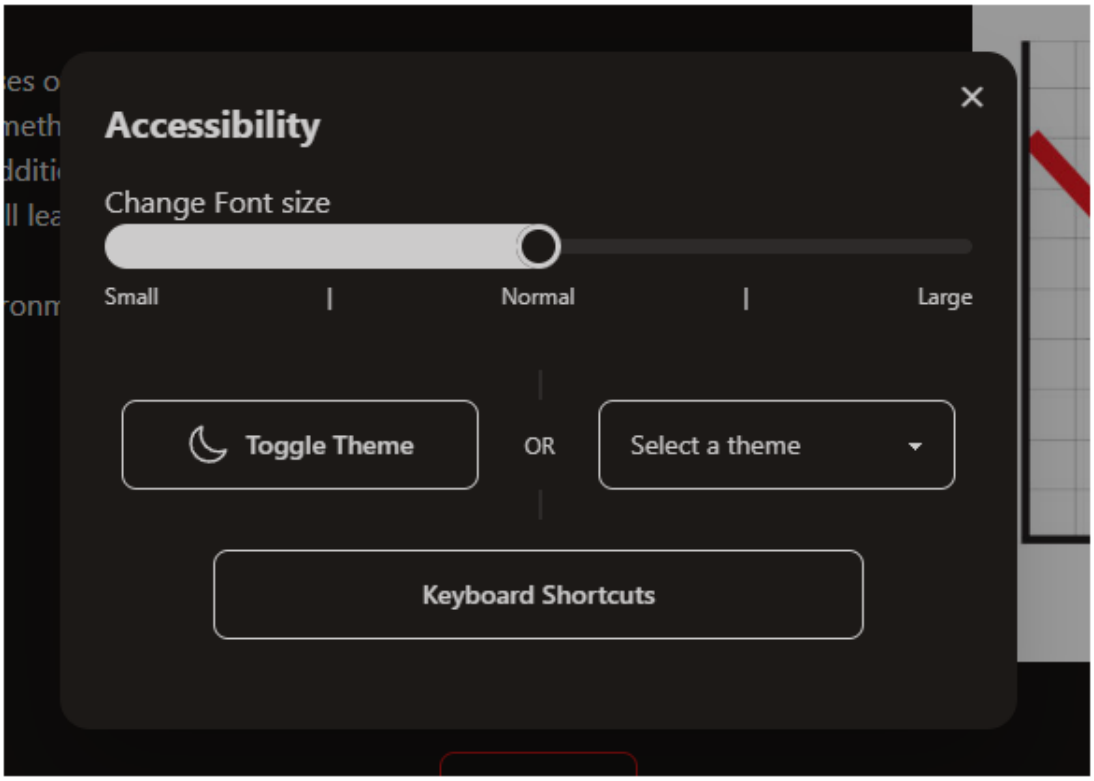

Figure 1.9: The accessibility menu

2 Software Engineering Process

2.1 Overview

We decided to go with scrum. A project like this is bound to have several massive unplanned changes and many more smaller ones, and we believe that in such a situation an agile framework like scrum would be the best choice. Our development consisted of two-week long sprints with meetings every Wednesday at noon. The plan was to create new increments of the project by working on it little-by-little every day. At the end of each sprint we would conduct a sprint retrospective meeting, assign new tasks, and initiate a new sprint. We used Jira to store and assign the backlog of tasks to each person during our sprints.

The following subsection details everything that has occurred throughout the sprints. These notes were mostly taken by Basim. Some were taken by Monty, Vinit, and others.

2.2 Sprint Details

2.2.1 Sprint 1

A problem was that we installed pure Svelte when we should have used Sveltekit for our project. Monty fixed that by installing Sveltekit. Another problem was that our new install didn't have a properly configured *.gitignore* file, which meant that several dozen node modules were all committed to the repository. That also was fixed. Vinit set up routing. Once routing was set up he also created the modules page. Everyone else was given a rundown of Sveltekit to catch them up to speed. Some things to do next were make the navigation bar on the top look nicer and set up a quizzes page.

Raj set up the firebase console and created the libraries page. Shubham worked on the modules page: setting up the PDF and YouTube viewer. Monty gave the styling on the navigation bar a makeover. Julian set up the home page. Chris added button handlers to the home page. Basim and Fouzan worked on the quizzes page. Vinit made general UI changes to make it more appealing.

To do: Vinit will create a database for all our media, Monty will add login-with-Google, Julian will add a footer and a daisyUI theme, Fouzan will implement reactive quizzes, Basim will add accessibility options, Chris will add a search function, Shubh will make an interface for embedding videos, and Raj will compile resources for the libraries page.

2.2.2 Sprint 2

We faced merge conflicts with the search function and some other mistakes that came from inexperience with using tools like Git, as well as general time management issues. Other than that everyone pretty much did their respective jobs.

This week's Jobs were similar as before with some new ones: Vinit making the back end more dynamic and Monty getting the login feature working with the database.

We were behind schedule so some tasks were moved to the third sprint. To do: Monty will

finish up with registering our authentication to the database, Vinit will give the modules page a proper makeover and then finish it off, Chris will add scaling images to the background and implement lazy images, Basim will continue looking at the accessibility menu, Raj will help with login and authentication, and Fouzan will continue working on the quizzes page.

2.2.3 Sprint 3

Some issues faced this time around were: Raj had a weird bug with the login button, Monty had to figure out backend stuff which is always a pain, Julian needs more time, and Basim had difficulty making the accessibility menu function on other pages. The accessibility menu in general was hard to deal with since it still had some residual CSS that hadn't been converted to tailwind or daisyUI yet.

On the bright side Chris added some background images, a logo, content for the home page, and changed some colors.

Monty added a dark-light theme switcher for the accessibility menu, changed CSS to tailwind mainly on the modules page, added inline documentation, refactored the README files, added feedback while authentication is loading, and fixed the modules page to make it look better after the tailwind move.

Basim worked on keyboard shortcuts for the accessibility menu, Shubh gathered question and worked on the logic for the quizzes page, Raj worked on getting other login providers, and Julian also helped with the dark-light theme.

Chris had a merge issue and possibly has to redo some things in his branch before merging it with main.

2.2.4 Sprint 4

Monty finished the theme switcher, switched more CSS to tailwind, added a slider for resizing text, cleaned up the modules page to make it look nicer, and some other general front end stuff on pretty much every page.

Raj worked on adding tabs for the libraries page. Shubham added visual cues on the modules page. Chris fixed his branch and added some background images and a logo to the home page.

Problems faced: the quizzes page was pulling more documents than it was supposed to, quiz questions were stored locally so you could easily cheat, the footer was breaking on certain pages, some merges had to be reverted because they solely contained pure CSS and no tailwind, and inconsistent indentation also lead to some issues.

Tests were done on the current main build to see how slides and videos were loaded by the modules page. Only one slide URL is loaded at any given moment and a new one is loaded when it is selected from the drop-down menu, which is good. YouTube videos are also only loaded when they are selected, even though multiple versions of the video are loaded at the same time—embedded, original, href, etc.

2.3 Sprint Timeline

Dates	Task
Jan 9 - Jan 16	Created **Project Proposal**
Jan 17 - Jan 23	Generated user stories and created **Product and Sprint Backlog**
Jan 23 - Jan 25	Finalize **Product and Sprint Backlog** due **Jan 26**; Meet with TA
Jan 27	Begin **Sprint 1**
Feb 7	Sprint 1 Retrospective Meeting and Create **Progress Report 1**;Begin Sprint 2
Feb 21	Sprint 2 Retrospective Meeting and Submit **Progress Report 1**;Begin Sprint 3; Meet with TA
March 6	Sprint 3 Retrospective Meeting and Create **Progress Report 2**; Begin Sprint 4
March 20	Sprint 4 Retrospective Meeting and Submit **Progress Report 2**; Meet with TA
April 10 - April 20	Web App Deployment and Submission of **Final Report**; Final Presentation

Table 1: Project Timeline

3 Challenges

3.1 Technical

For some of us it was our first time making a web app so there was fair amount of learning on the job. Learning node and the right packages to use was the first step. We also learned that git is a very powerful and useful tool, but with great power comes great responsibility—like making sure you don't accidentally commit a million lines worth of node packages or making sure your branch doesn't conflict with the one you are trying to merge into.

3.2 Personal

Some team members were missing too many meetings. They lacked cohesion with the rest of the group as a result. Some members failed to provide work updates to missed meetings making it difficult to judge whether the work they had been assigned was completed. To deal with these emerging issues, members were sent strongly worded emails to attend more meetings. The last sentence is a joke, but still.

4 Activity

4.1 Contributions

The following are lists of quantifiable contributions for every member.

4.1.1 Monty

- Installed Svelte, Changed Svelte to Sveltekit.
- Added the navbar.
- Set Firebase rules for authentication and database.
- Added login feature and contributed to database on correlated functionalities.
- Added light-dark theme switcher and theme selector.
- Re-skinned the accessibility menu, added text resizing slider and theme functions.
- Installed Tailwind and Daisyui. Changed much of the CSS to tailwind.
- Improved background images to work with multiple themes.
- Made UI improvements in modules.
- Wrote readmes and commenting as documentation for project.
- Added some reactivity to the site to make it look better on different screens.
- Worked on both reports.

4.1.2 Vinit

- Set up routing.
- Set up retrieving media from the database.
- Made general aesthetic UI changes.
- Made backend more dynamic.
- Added major additions to the modules and quizzes page.
- Worked on both reports.

4.1.3 Raj

- Drafted the entirety of the second progress report, later reviewed by other team members.
- Worked on libraries page.
- Initialized Firebase console.
- Helped with login buttons.

4.1.4 Shubh

- Set up PDF and YouTube viewer.
- Gathered questions for the quizzes page.
- Worked on quiz page logic.

4.1.5 Fouzan

- Created quizzes page.
- Made the quizzes page reactive for different sized screens.
- Added several additions to the quizzes page.

4.1.6 Julian

- Set up homepage.
- Design logo.
- Converted CSS to tailwind on the homepage.

4.1.7 Chris

- Added buttons to home page.
- Added background images and styles to the homepage.
- Also converted CSS to tailwind on the homepage and others.

4.1.8 Basim

- Took administrative roles like writing report 1.
- Worked on quizzes.

- Worked on accessability menu.

4.2 Logs

4.2.1 Github Contribution Plots

Figures 4.1 through 4.8 are plots that represent summaries of the total commits each person has contributed.

Figure 4.1: Monty's commits summary.

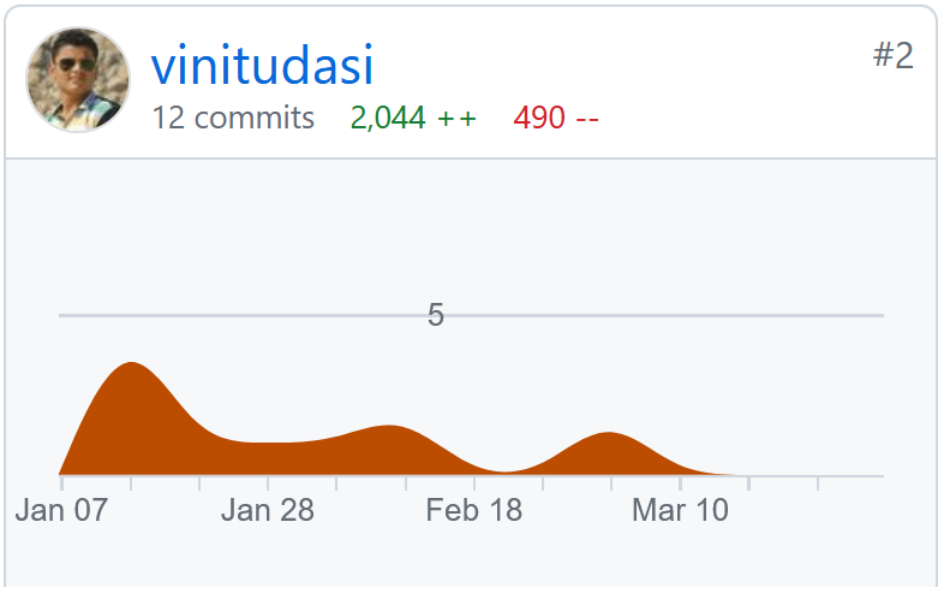

Figure 4.2: Vinit's commits summary.

4.2.2 Performance Reports

The next figures summarize how much of our backlog we were able to complete.

Figure 4.9 is the sprint commitment which shows the average tasks from our backlog completed per sprint.

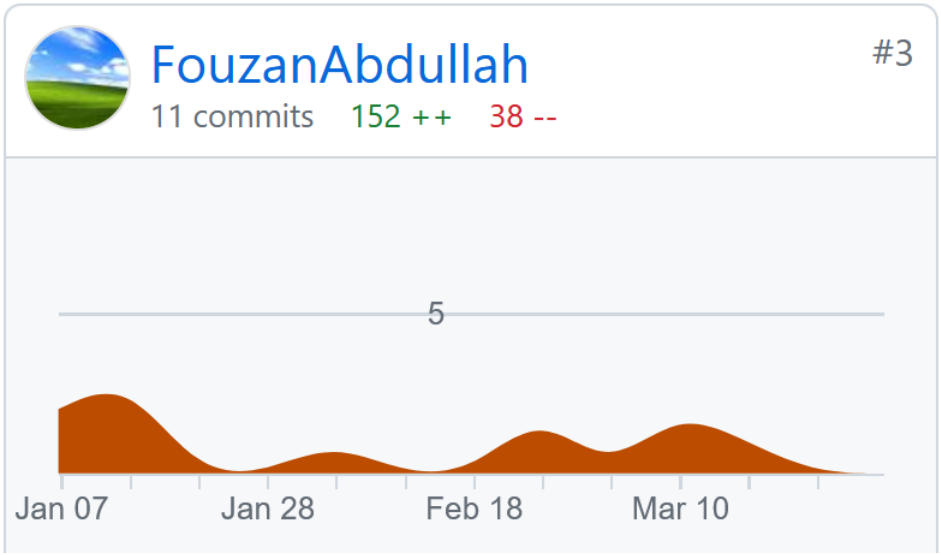

Figure 4.3: Fouzan's commits summary.

The burnout charts, which represent completion of tasks over time compared with expected completion, for sprints 3 and 4 can be seen in figures 4.10 and 4.11, respectively.

The velocity chart for sprint 4, which is basically the same thing but with bars, can be seen in 4.12.

The cumulative flow diagram, which shows progress done compared with to-do over time, can be seen by referring to figure 4.13.

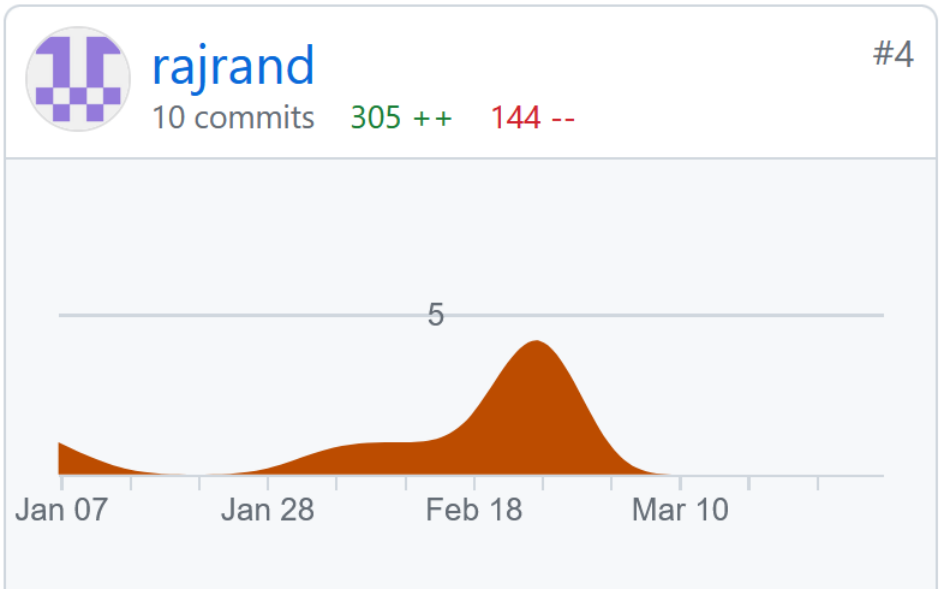

Figure 4.4: Raj's commits summary.

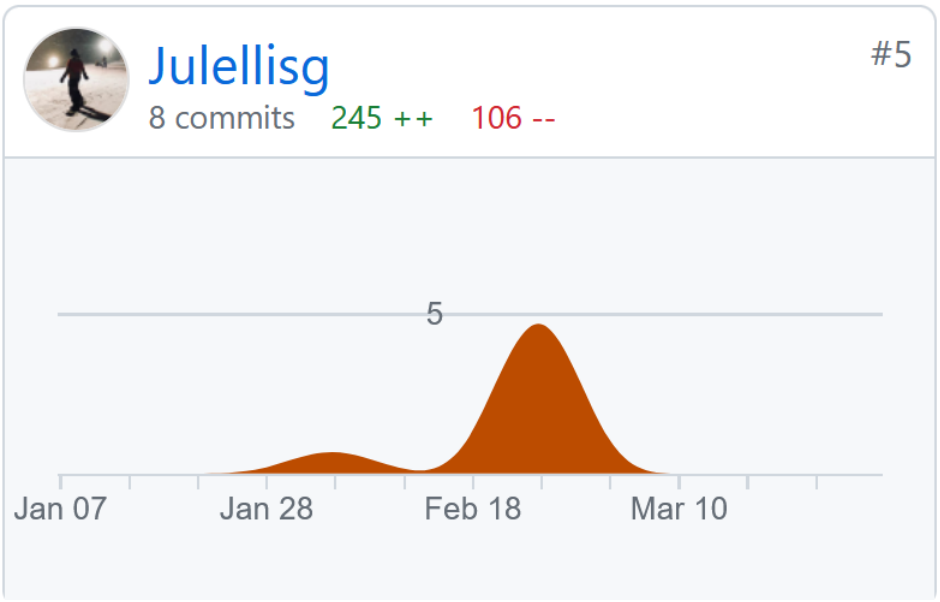

Figure 4.5: Julian's commits summary.

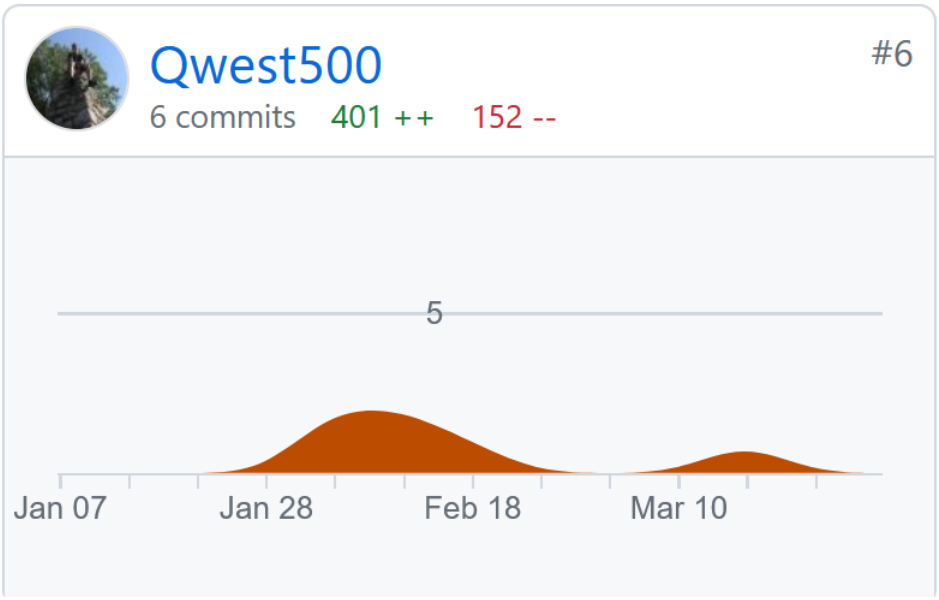

Figure 4.6: Chris's commits summary.

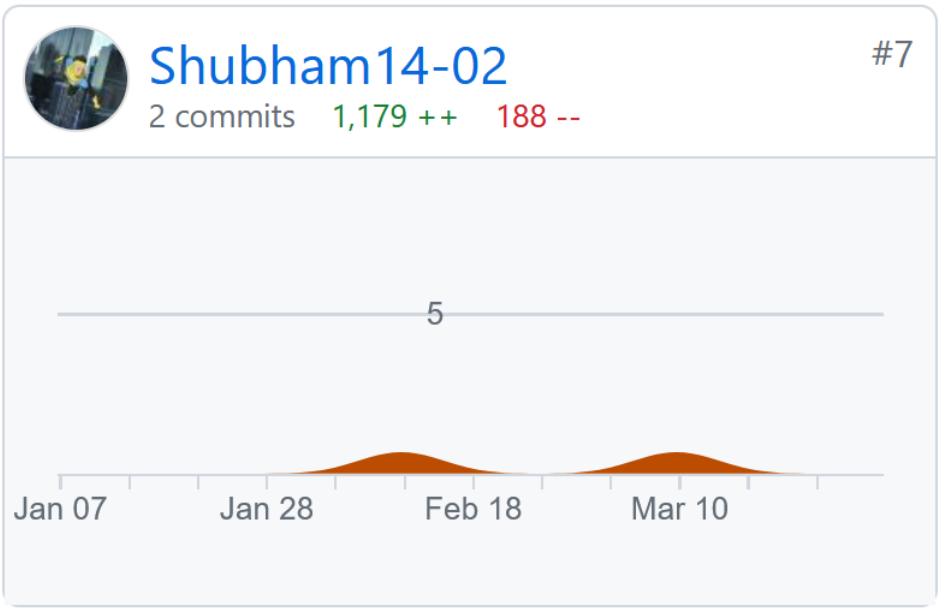

Figure 4.7: Shubh's commits summary.

Figure 4.8: Basim's commits summary.

Figure 4.9: Sprint commitment chart.

Figure 4.10: The Sprint 3 burnup chart.

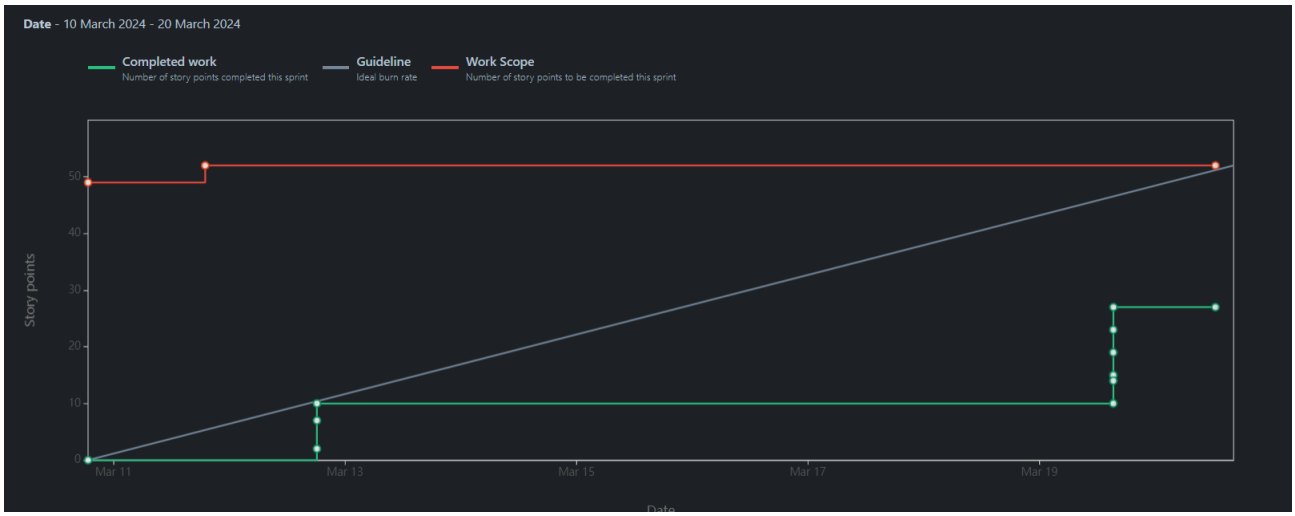

Figure 4.11: The Sprint 4 burnup chart.

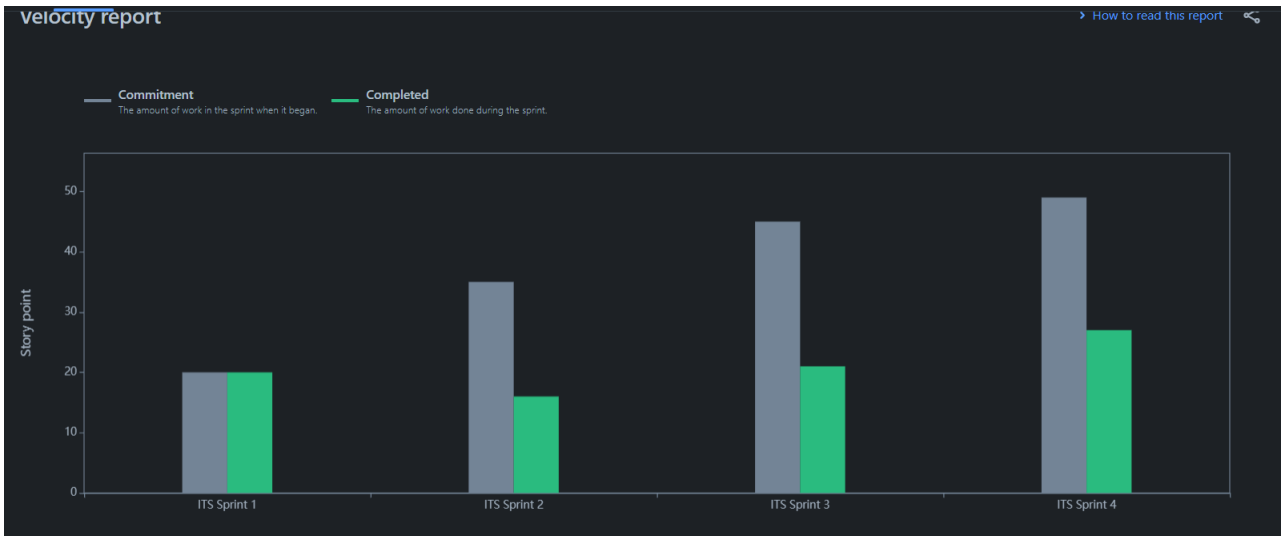

Figure 4.12: The Sprint 4 velocity chart.

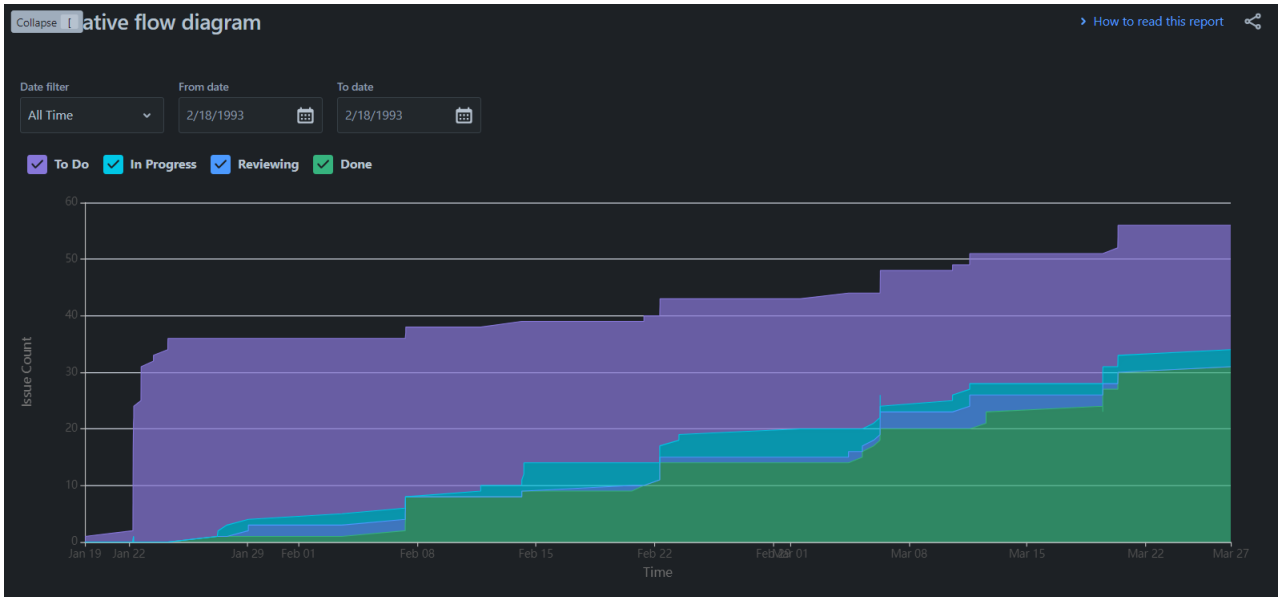

Figure 4.13: The cumulative flow diagram.

5 Github Repository for the Project

https://github.com/SWE-2024/COSC-4P02

6 Team

1. **Fouzan Abdullah**
   Student Number: 6840797
   Role: Product Owner

2. **Basim Ahmed**
   Student Number: 7022494
   Role: Developer and Scrum Master

3. **Vinit Udasi**
   Student Number: 6847800
   Role: Developer

4. **Shubham Amrelia**
   Student Number: 6846877
   Role: Developer

5. **Julian Ellis Geronimo**
   Student Number: 6756597
   Role: Developer

6. **Monty Oshinov**
   Student Number: 6759286
   Role: Developer

7. **Rajan Randhawa**
   Student Number: 6996441
   Role: Developer

8. **Chris Orr**
   Student Number: 6755383
   Role: Developer