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

Progress Report One - Final Project

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

Contents

Brock Interactive Training Engineering System (BITES)

1 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

2 Overview

In this progress report, we have provided a comprehensive overview of our team's advancements in the development of the **BITES** Project. This report also covers details about our iterative procedures in sprints and how sprints are used to efficiently plan and complete tasks. We will also explore the difficulties we had while developing our project, providing insight into how we dealt with and overcame these difficulties. We will present our GitHub log, which provides a thorough record of project activity and individual contributions from each team member, in order to improve openness and collaboration.

3 System Updates

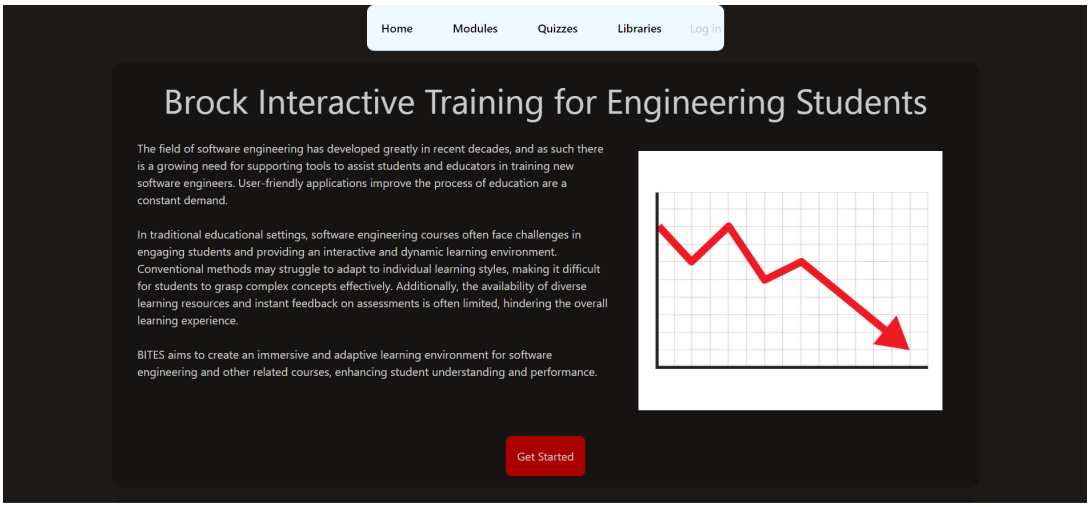
Initially, We decided to have a skeleton website ready to work around and develop more complex features as we go. So, We made all the main pages with plain-text and minimal UI. Similar to a Learning Management System, we made a landing page, a page to contain all modules, a page to contain all the libraries and a page for quizzes.

Once these pages were ready, we moved to more complex tasks such as backend for Modules and User authentication. We had a working app by the end of the second sprint. A screenshot of the application is attached below

3.1 Landing Page

The Landing page was decided to be a simple page with description of what the application does and contain links to other pages. Users can get an idea of what the website is providing and what features it contains. We decided to go with minimal layouts and

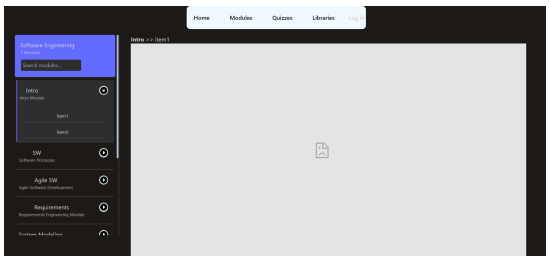
make UI changes as we move further in sprints. A screenshot of landing page is attached below.



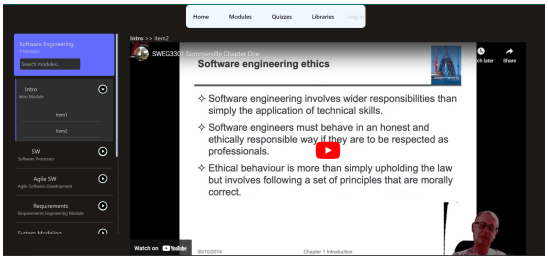
(a) Landing Page

3.2 Modules

Modules page contains all the academic material needed to understand software engineering methodologies and related concepts to it. There is a slide and a video attached to every module.



(a) Slide



(b) Video

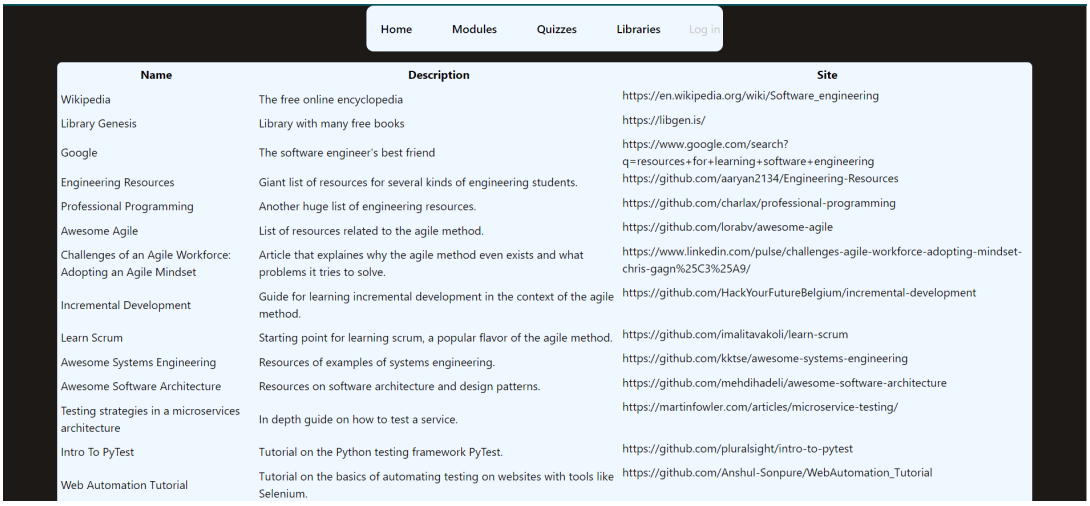
The slides are not visible because we have migrated the slides to a new database and will be working with that to implement an instructor's view which will allow instructors to add new content. These were initially vercel blobs just for viewing the slides.

3.3 User Authentication

We have implemented third-party authentication, specifically utilizing Google's authentication services to enhance the user authentication process. This integration allows users to log in to our system without entering any credentials and use google's log in services.

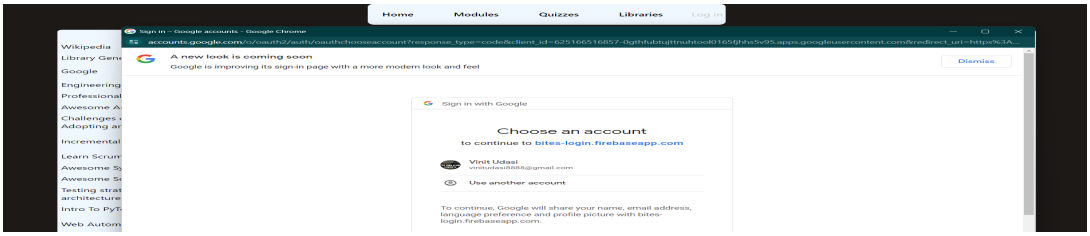
3.4 Libraries

The libraries page contains all the helpful materials, links to some learning tools and links to tutorials as well.



Name	Description	Site
Wikipedia	The free online encyclopedia	https://en.wikipedia.org/wiki/Software_engineering
Library Genesis	Library with many free books	https://libgen.is/
Google	The software engineer's best friend	https://www.google.com/search?q=resources+for+learning+software+engineering
Engineering Resources	Giant list of resources for several kinds of engineering students.	https://github.com/aaryan2134/Engineering-Resources
Professional Programming	Another huge list of engineering resources.	https://github.com/charlax/professional-programming
Awesome Agile	List of resources related to the agile method.	https://github.com/lorabv/awesome-agile
Challenges of an Agile Workforce: Adopting an Agile Mindset	Article that explaines why the agile method even exists and what problems it tries to solve.	https://www.linkedin.com/pulse/challenges-agile-workforce-adopting-mindset-chris-gagn%25C3%25A9/
Incremental Development	Guide for learning incremental development in the context of the agile method.	https://github.com/HackYourFutureBelgium/incremental-development
Learn Scrum	Starting point for learning scrum, a popular flavor of the agile method.	https://github.com/imalitavakoli/learn-scrum
Awesome Systems Engineering	Resources of examples of systems engineering.	https://github.com/kktse/awesome-systems-engineering
Awesome Software Architecture	Resources on software architecture and design patterns.	https://github.com/mehdihadeli/awesome-software-architecture
Testing strategies in a microservices architecture	In depth guide on how to test a service.	https://martinfowler.com/articles/microservice-testing/
Intro To PyTest	Tutorial on the Python testing framework PyTest.	https://github.com/pluralsight/intro-to-pytest
Web Automation Tutorial	Tutorial on the basics of automating testing on websites with tools like Selenium.	https://github.com/Anshul-Sonpure/WebAutomation_Tutorial

(a) Libraries



(a) Log-in

4 Contributions

Before starting the work, we decided on some technical and non-technical requirements that were essential for the system. Every member of the team is a developer and additionally **Fouzan Abdullah** and **Basim Ahmed** are Product Owner and Scrum Master respectively. In the initial meetings, we decided on some rules about working in collaboration so that none of the progress was lost. We are using **Github** for our version control and **JIRA** for progress tracking (Backlog, User-stories, Bugs). **Basim Ahmed** was tasked with taking meeting notes, managing the meetings and resolve any conflicts while **Fouzan Abdullah** was tasked with managing the product backlog and providing feedback during / at the end of every sprint.

Each developer is assigned user stories at the beginning of every sprint. Developers work on their user stories and log any bugs they encounter. Each developer was given an additional task of researching for better alternatives while working on their user stories to make the application more efficient and sustainable. **Monty Oshinov** was able to set up an initial structure for the system (codebase) using Sveltekit (problems discussed in depth later). A table providing details of contribution of all members is attached.

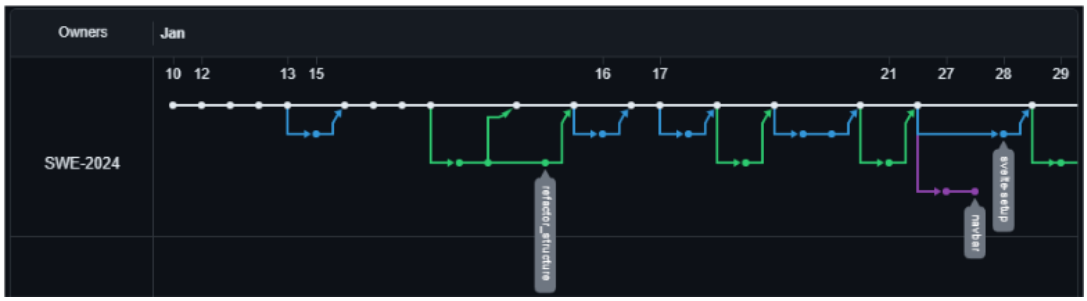
Team Members	Contributions
Fouzan Abdullah	Set up the skeleton page for quizzes. Implement a reactive quiz page. Continued updating of the quiz page
Basim Ahmed	Set up the skeleton page for quizzes. I created and continuing to work on an accessibility menu
Vinit Udasi	Made UI changes to make it more appealing. Created a database for mass media. Fixed the UI of modules and made the backend more dynamic. Worked on creating database and started work on developing a dynamic view for interacting with modules
Monty Oshinov	Setup base Sveltekit project. Installed and setup ESLint and code formatting. Added Tailwind and DaisyUI to the project. Worked on Nav bar with others, later touched up the navbar. Added client-side firebase initialization. Implemented firebase authentication with Google provider and added login UI for multiple providers. Wrote readmes for documenting requirements, directory structure, and usage instructions for the project. Integrating login with database.
Rajan Randhawa	Compiled a list of websites for the libraries tab.. Styled the table for the library. Moved the variables to a file on the backend. Helping with login and authentication.
Julian Ellis Geronimo	Set up the skeleton landing page. Added the first DaisyUI theme to use across all other pages. Added the footer to be persistent across all pages with the navigation bar. Continuing to change all standard CSS to implement Tailwind CSS utility classes and DaisyUI themes.
Shubham Amrelia	Worked on the modules page and working on setting up a pdf viewer and YouTube video selector
Chris Orr	Became familiar with Github and added button handlers to the landing page. Added reactive search feature to modules page to allow easy access by content, title or description. Debugged and re-wrote search feature. Added scaling images to the background and lazy images.

Table 1: Contributions of Team Members

Each member was instructed to commit their work, so their contribution was noted and any progress they made wasn't lost. Commit graphs of all the members are attached below.

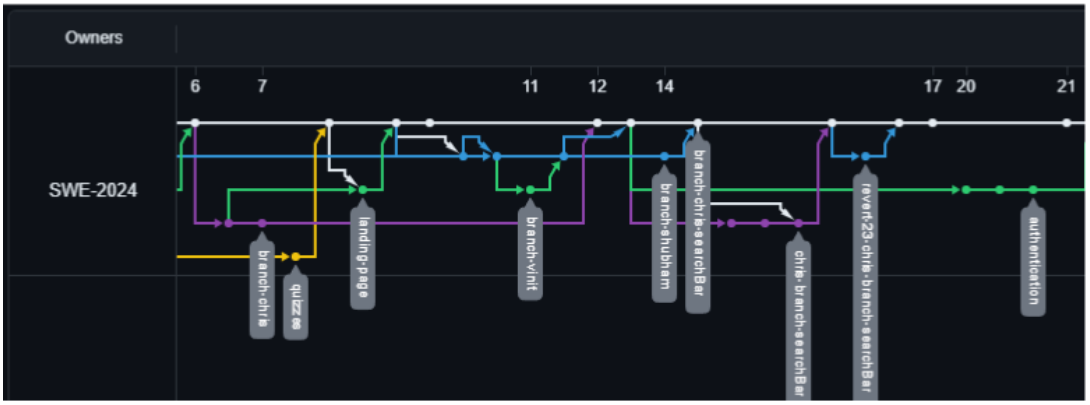
(a) Commit Graphs

Sprint-wise Github logs are also attached below.



(a) Sprint 1 Log up till 29 January

(a) Sprint 1 Log till 7 February



(a) Sprint 2 Log

(a) Sprint 3 Log (ongoing)

5 Implementation

5.1 Frontend

For the frontend, We narrowed it down to to options namely ReactJS and **Svelte**. These are both very popular frameworks for frontend. We put it to poll and decided that **Svelte** was a better option for what we had in mind for the project. **Svelte** is one of the most popular frameworks for develpoing web applications in recent times, It is quicker than most and also easy to learn.

5.2 Backend

Since we decided to have a skeleton website ready, our first sprint did not take backend into account. In the second sprint, we started work on User Authentication and a suitable backend for modules. For authentication, we used firebase and firestore. We were able to successfully set it up and get it working. For the modules, we first used vercel blobs to store slides and hard-code the modules but as we had to implement an instructor's view, we decided that we need to migrate all the modules to a good database and we have narrowed it down to firestore and Amazon Web Services. More work will be done on the backend of modules page in the upcoming sprints.

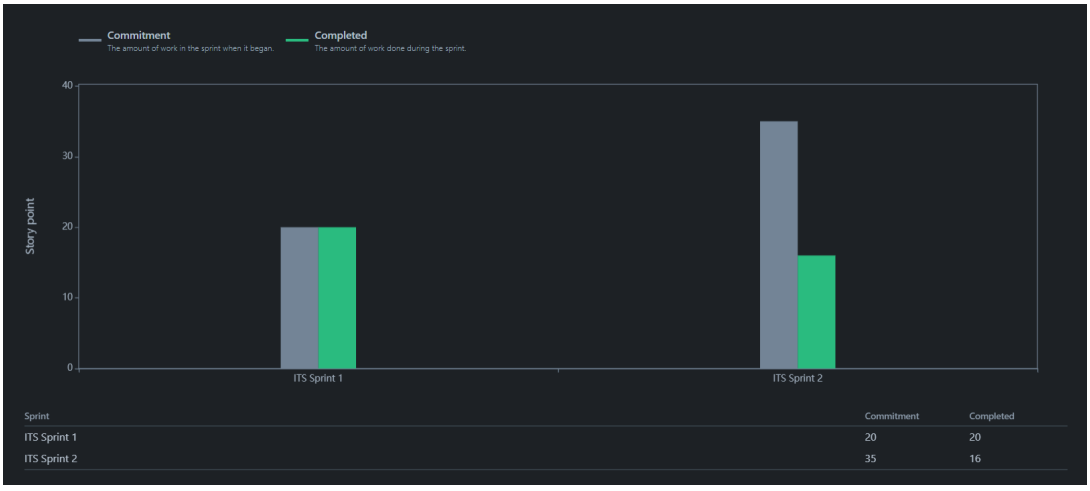
6 Sprints

So far, we have completed two sprints and are in the middle of the third sprint. First Sprint was aimed towards familiarizing ourselves with all the required tools and second sprint was aimed at creating a Minimum Viable Product.

Below are the sprint velocity charts.

Sprint	Tasks
Sprint One	Created the Codebase; Created Landing Page;Created the navbar;Created the modules page; Created the Quiz page; Set up routing; created the libraries page;
Sprint Two	Set up firebase console; set up login with google; created database for storing mass media; interactive UI for modules; Searchbar for modules page; improved libraries page
Sprint Three (ongoing)	Add footer; Implement Admin/Instructor accounts; add visual queue for completed modules; set up lazy loading of images; Develop a theme for the website; Develop a dynamic quiz page; Add accessibility menu.

Table 2: Sprint Work



(a) Sprint Velocities

7 Challenges Faced

7.1 Codebase Setup - I

Initially, one of our team members set up the codebase for the project through a different way. We realized that sveltekit could also be used for setting up the codebase and had simpler routing, so the codebase had to be revamped.

7.2 Codebase Setup - II

After installing the new codebase, a '.gitignore' file was missing in the global level to prevent accidentally pushing libraries and dependency files onto the repository. One of our team members accidentally pushed the NodeJS libraries. The branch had to be reverted

7.3 Collaboration

Minor mistakes such as bugs after merging a pull request, pushing into main branch directly. Team members acknowledged that they need to be careful.

7.4 Communication Gap

A meeting was missed by 7 of the 8 team members (sprint retrospective). We apologized to the teammate and acknowledged that better time management was needed by the team members.

8 Links to Tools Used (Clickable)

- **Github**
- **Svelte**
- **Firebase**

9 Github Repository For the Project: Click Here!

- or copy : https://github.com/SWE-2024/COSC-4P02
- All the updated changes, releases and documents will be uploaded to this github repository