

Go-Learn

The background of the slide is dark with a pattern of concentric, wavy lines in shades of grey and blue. Overlaid on these lines are numerous small, semi-transparent dots in various colors, including red, orange, yellow, and green. The dots are more densely packed in some areas, creating a sense of depth and movement. The overall effect is a complex, organic, and futuristic aesthetic.

What is my Project



- Learning Resource Management System Software
 - Users can be an Admin, a Student or a Teacher
 - Different user roles have different access rights and attributes
 - Teachers and Admin can manage content, while Students can only view content
- Web App built with the [NextJS](#) Framework (Node, React)
 - Use [@shadcn/ui](#) component library for look and feel
 - Use [Tailwind](#) for functional CSS classes
 - Use [Prisma/Supabase](#) for database interaction

Last you heard ...



- Sign in as users with different roles (with session cookies)
- Only admin has been mostly fleshed out:
 - Can manage and create Users
 - Can manage and create Courses
 - Can manage and create Modules within Courses
 - Can manage and create Units within Modules

And now...



- Teachers can...
 - Edit Module content for those Modules that they are assigned to
 - Manage all content within the Modules (Units, Resources and Sections)
 - Create and manage Assignments on Modules that they are assigned to
- Students can...
 - View Modules (and child content) for those that they are assigned to
 - View the Assignments that are on the Modules they are assigned to
 - Make Assignment submissions

Frameworks and Libraries




- NextJS as a full stack framework:
 - Server-Side Rendering (SSR)
 - Server actions
 - Routing for SPAs
- Prisma ORM
 - An ORM (Object Relational Mapper)
 - Provides a simple, Type-safe DB API
- Nodemailer – For sending Emails
- Shadcn/ui – A component Library
- TailwindCSS – CSS framework using utility classes
- Supabase – File Storage in AWS-like buckets

Deployment



- Vercel – A service that provides hosting for web applications at a low price (or for free for various purposes)
 - Can link to a remote Git repository to trigger fresh builds of the latest changes
 - Created a GitHub repository to mirror the CSEEGit one for this purpose
 - Configure the repository to check, set ENV variables, run a build.
- Brevo – An emailer service to send emails to users
 - Provides a front-end to monitor emails. Can see if they have been sent, delivered, opened and interacted with
 - Free for a limited number of emails.

Setbacks



I've had various setbacks with my project, key ones being:

- Docker incompatibility - Could not access database properly from within a docker environment
- Database integrations - Had to change DB from MongoDB (as intended) to Postgres (via Supabase service) as Prisma could not connect to the MongoDB install I could set up
- Supabase file uploads - This was more due to a bit of a learning curve than anything else
- Deploying to Vercel - Setting up and configuring came with some challenges (build, ENV variables, connectivity to a DB, etc)
- Implementing SSR and Middlewares - Late to the development process, so was not possible with the set up as is. A large-scale refactor would be required to implement

What is Next?



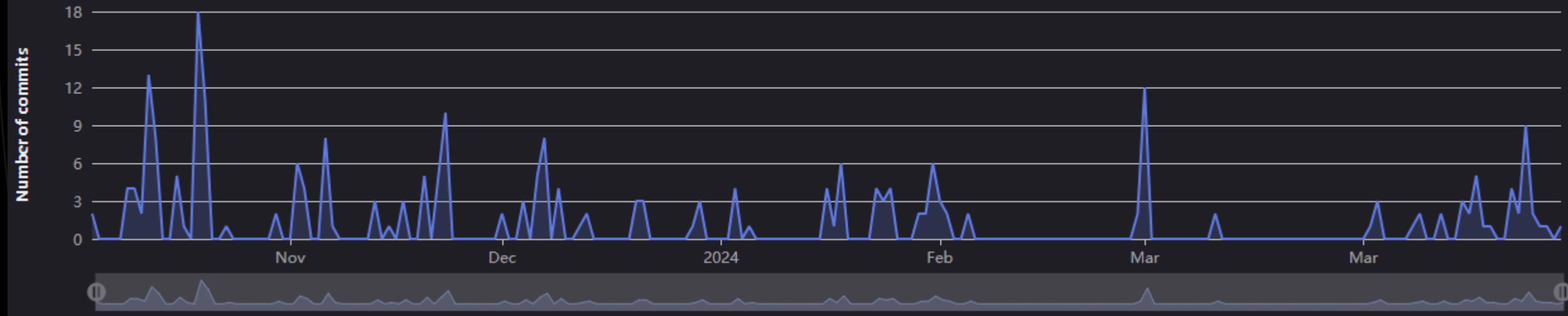
- Assignment Submissions being reviewable (and downloadable) by Teachers for review
- Being able to assign marks and provide feedback on Assignment Submissions
- Student being able to review their marks and feedback in-app
- Student & Teacher communications (completely new and would need research into best practices and potential implementations):
 - Intended in the use case of a forum
 - Q&A potential
 - Possible one-to-one communications between Teachers and their Students

Use of GitLab



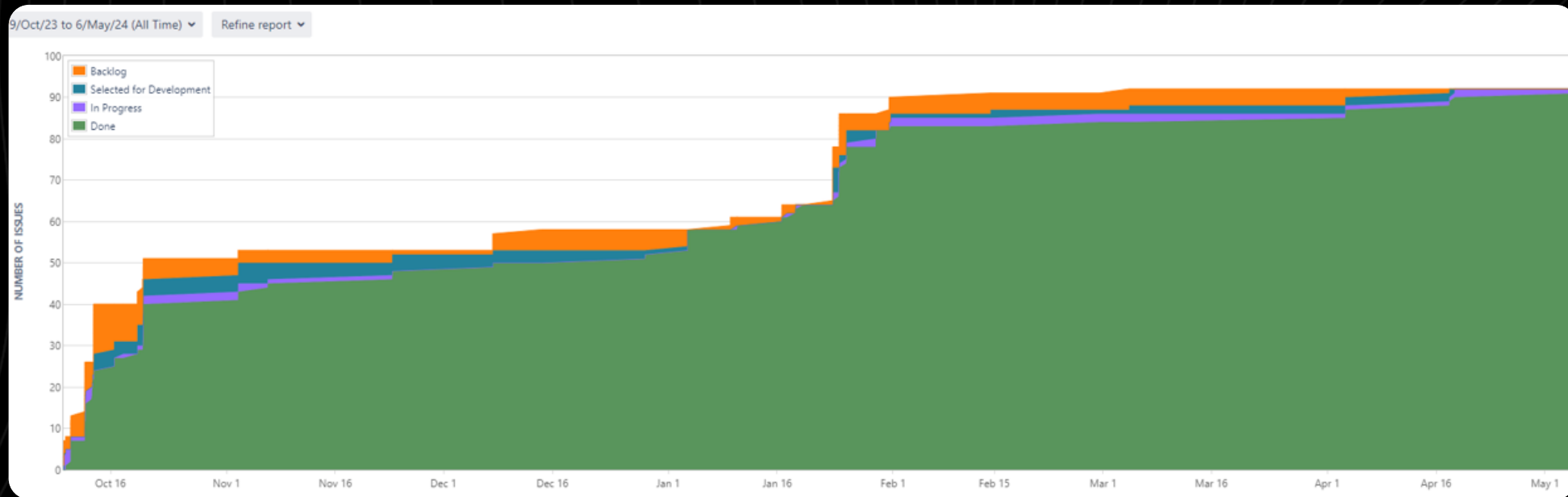
Commits to master

Excluding merge commits. Limited to 6,000 commits.



Use of Jira

- At the beginning of the project when it was just coming off the ground, I kept up well with adding Jira stories and keeping up to date with updating them
- In the spring term, I got quite a lot of work done right at the beginning of term (and caught up the tickets with development), but largely just did optimisation and refactoring work. Very few features were added to, but those that existed were optimised



Demonstration

DEMONSTRATION