Written Retrospective

Throughout this project, our team gained experience with several software development tools and practices. Some of these things are listed below.

Skills Obtained

The main skill we acquired was learning how to connect a Flask frontend to a FastAPI backend. We learned how to consume RESTful APIs securely and effectively using session management. Through the implementation of token authentication with session management, we used /token and /user/me endpoints to manage user identity securely. We also learned the skills to use ReportLab to dynamically generate structured PDF reports for usage history. This was something we used to provide a relay of specified data including user, equipment, and date information. We also acquired the ability to create scannable QR codes for direct checkout links. We also had to work around handling the state and authentication flow for the QR code-based accesses cleanly and securely.

Simple Lessons

A major lesson we learned was in order to manage and secure sessions between a stateless backend and stateful frontend there needs to be a lot of communication amongst the teams working on the project. Another major lesson we learned was that the backend is more important than the initial frontend functionality. If you implement a ton of features but your backend and API do not support those functionalities then they are useless. Having developed a software that was working on a hardcoded data table is much more difficult than anticipated to convert to a software that contacts an API.

Accomplishments

- Full login and session management flow using tokens
- Separate admin vs. user experiences (for example the admin panel)
- Equipment reservation, checkout, and return workflows
- QR code system for direct checkout via scan
- PDF generation with filtering by user, equipment, or all users
- Equipment history views for both admins and regular users

Just short of Accomplished

- Full integration of the reservation page (date and item filtering still works on the hard coded values in branch us15)
- Reliable admin demotion logic (the UI allows for admin demotion but it can break if the last admin demotes themself)

Summary

This project gave us experience with full-stack development, API integration, and role-based user flows. We successfully built a functional system with secure authentication, QR-based equipment checkouts, and dynamic report generation. While some advanced features like full reservation integration and admin demotion need refinement, the project demonstrated our ability to design, build, and debug a collaborative application from the frontend side of development. Overall, as a group, we can all say we enjoyed this project and the time we spent developing and fixing its issues.