# Reflection Log #2

## Section 1 – COMPACT HEADER

**Date:** June 27, 2025  
**Student Name:** Vishnu Narayanan K R  
**Student ID:** 2301623  
**Company Name:** YULCOM Technologies Inc.  
**Faculty Supervisor Name:** Michel Paquette  
**Time Period Covered:** June 9 – June 27, 2025

## Section 2 – TECHNICAL INFORMATION

In the previous reporting period, I completed on boarding and contributed to several features involving UI bug fixes, user role distinction, and data integration (vitals and biometrics) in a Django + React health care project. Since then, I have made considerable progress in terms of ownership of features and independent problem solving. I moved from understanding and improving existing features to implementing and enhancing more complex functionalities, particularly around **appointment management**, **calendar integration**, and **patient data processing**.

The main focus during this period was improving the **appointments system**. I extended the existing functionality by implementing the **Edit Appointment** feature, which required reusing and modifying existing components, troubleshooting issues with form inputs, and integrating it with backend endpoints. Initially, the form failed to load data correctly due to an API error caused by incorrect field names sent from the UI. I identified and fixed this issue myself after isolating the API call and validating the payload against the backend schema.

After completing the edit functionality, I enhanced the **appointment list interface** by adding features like **filtering, sorting by date and service type**, and creating a **dropdown for status changes** directly on the main screen. I also designed and implemented an **Appointment Calendar** that provided a visually intuitive way to display appointments. The calendar included enhancements such as **color-coded appointment types**, and a responsive design that adapted fields dynamically based on API data. This functionality significantly improved user experience.

The second half of the reporting period involved tackling the **recurring appointment** feature. I found that while the front-end had some placeholder logic, the backend support for this functionality was incomplete. This discovery led to a deeper investigation into how we might implement recurring scheduling in a scalable way without disrupting existing logic. This was challenging due to the interconnectedness of the time calculation and scheduling logic.

Alongside feature work, I also resolved bugs related to **vital data redirects**, **patient detail display**, and **appointment creation failures** (due to missing patient UUIDs). I implemented a new **utility function** for active visits, which could be reused across endpoints. Additionally, I made UI improvements to **visit counters** on the homepage and ensured smoother user navigation.

Another important milestone was a **progress validation meeting** with my company supervisor, Mr. Komi, where I was asked to prepare a testing implementation report for review by July 7.

**What worked well:**

* Reusing existing components to build new features saved development time and reduced errors.
* Self-troubleshooting API errors and improving communication between frontend and backend.
* Collaborating with peers (e.g., helping Zafer and Eric debug local environments and merge issues).

**Challenges:**

* Understanding and fixing logic for features like recurring appointments without full backend support.
* Managing complexity when multiple UI components needed to reflect updated API behavior.
* Handling design iterations based on last-minute UI requirement changes from the team.

**Knowledge applied:**

* Advanced React concepts (component reuse, conditional rendering, hook debugging)
* Django REST integration and endpoint validation
* Git/Bitbucket version control and PR workflows
* UX principles for calendar-based navigation and data display

**Knowledge gaps:**

* Advanced backend implementation of time-based recurring logic.
* More exposure to automated frontend testing (e.g., unit/integration tests in React).
* API security patterns for identifying and safely handling duplicate data entries.

**Learning:**  
This phase helped reinforce my confidence in feature ownership and frontend-backend coordination. It taught me how to scope and handle UI-heavy feature requests and improved my debugging and research skills significantly.

## Section 3 – PERSONAL INFORMATION

I continue to feel highly engaged and supported in my internship at YULCOM. The balance between collaborative work and individual responsibility has allowed me to grow both technically and professionally. While the increasing complexity of tasks can be overwhelming at times, especially when features lack documentation, I’ve found that my ability to independently resolve issues has improved significantly.

The team, particularly Quitian and Mr. Komi, have been accessible and open to discussions. I appreciate the trust placed in me to own key features and fix production-level bugs. Helping teammates (Zafer and Eric) resolve their issues also made me feel more embedded as a contributing member of the team.

Going forward, I look forward to continuing work on improving appointment features and implementing testing procedures. I'm also motivated to explore scalable architecture patterns and documentation best practices.