

Reflective Essay

It was a difficult but rewarding experience to work on the Feeding Dashboard project. The project involved building a cross-platform dashboard (CCU) so that staff could identify patients to refer to a dietitian. Through the application and application of machine learning, the system analyzed physiological data of the patients and emphasized those who required a diet consultation. As a member of the team, I did three primary things: coding the interface, managing the database, and writing the project file

Role and Contributions

My early major contribution involved interface design and layout. Though I was not the lead developer, I made some contributions in crafting an understandable and responsive user interface. Through JavaScript, I assisted in implementing modules that presented registered patients as well as the dietary referral status of the same. It was necessary for the interface to remain clean and navigable, especially in the hectic working environment in which staff labor.

Second, I was the driver who transitioned our data processing system from static CSV files to a horizontally scalable, lightweight database. Flat files were used to work with data initially, but once we began to scale in size, performance was an issue. I pushed for SQLite because it was easy to use and cross-platform compatible. I assisted in the development of the database schema, scripted data ingestions, and query optimizations to the point that patient data was readily accessible.

I also spent a great deal of time on documentation and the final write-up. This included documenting the system architecture, user flow, and the technical process we employed. I wanted to make our processes easy to understand for both technical and non-technical readers. This document was significant in terms of bringing out the project and how it could be impactful.

Challenges and Obstacles

Like in any project out there in the real world, we hit a few road bumps along the way. The most critical issue we faced was the problem of inconsistent data. We had hoped to have all of our CSV files nicely formatted, but absent or malformed records meant test failures. To rectify this situation, I helped the team create validation rules and data-cleaning operations that served to significantly increase stability.

Another issue was integrating the machine learning model with the front-end. The model was predicated on specific data points, and there were inconsistencies between how these fields were labeled in the CSVs and how they were actually reflected in the UI. We handled this by normalizing field names and creating a mapping system to wed the back end with the front-end presentation.

Our initial use of JSON files for temporary data storage also started to become a problem. As the dataset grew larger, performance grew increasingly sluggish. The transition to SQLite was a revelation, and I was involved in the rewriting of backend logic to allow this transition to be smooth and effective.

Mistakes We Learned From

One early mistake was underestimating how messy real-world data can be. We had thought that there would be consistency in the data set and it would be normalized and ready for use this did lead to people being incorrectly identified as needing to see a dietician

Another issue was the original layout. It was overloaded with information, making it hard to read. We redesigned it to include sections, allowing staff to navigate between brief summaries and full patient details. This change made a big difference in terms of usability.

Lessons Learned

This project taught me a lot about handling real-world data. Clean, perfect datasets do not exist, and designing systems that can handle well with inconsistencies is the key to their success. By checking early, we avoided wasting time and a multitude of issues later in the process.

I also learned about the value of communication when working together as a team with varied roles—UI, backend, and machine learning development all required coordination. Tools like Git and WhatsApp were used to coordinate with everyone. This helped me improve as a good collaborative team player.

Through this project, I learned to combine two basic aspects the area under focus with backend. Designing images conveying information effectively this is where I learned to appreciate the place of design especially in a healthcare setup where quick understanding is a game-changer.

The report was a learning process. Breaking down technical processes into simple, understandable language was not simple, but it made me a better communicator. The report was not only an introduction to the project but also a good guide for future versions.

Final Thoughts

On the whole, the project gave me a chance to work on actual issues, deploy technical skill, and grow as a team. Picking things up through interface design, database building, and reporting, I gained things which will stay with me—namely in flexibility, teamwork, and system design.