On my 23rd day at Surfboard Payments internship, I worked on developing a simple dashboard with two main sections: Mail Fetching and Auto-Assignment. The Mail Fetching section was designed to retrieve incoming emails, while the Auto-Assignment section used a machine learning model to automatically assign emails to developers or support engineers based on their content. The goal was to create an efficient system that could help streamline email management and improve workflow efficiency. For the frontend, I used plain HTML and CSS to ensure a clean and user-friendly interface. The backend required storing emails and their assignments in PostgreSQL, and I also had to implement a machine learning model to categorize and assign emails appropriately.

As I started working on the project, I encountered multiple challenges, especially with the backend server, which continuously displayed errors. Despite trying different approaches and modifying the code several times, the dashboard features were not updating as expected. This made it difficult to achieve the desired functionality, but I remained focused on troubleshooting and learning from my mistakes. One of the most complex parts of the project was integrating a machine learning model to fetch and categorize emails. I explored various techniques to connect real-time emails, including using IMAP and POP protocols. However, I faced difficulties in establishing a proper connection, which prevented me from successfully retrieving real-time emails.

To overcome these issues, I spent a significant amount of time researching and studying different methods to connect to email servers. I learned about the working principles of IMAP and POP, their differences, and how they are used in email communication. IMAP allows real-time synchronization with an email server, while POP downloads emails to a local system. Despite my efforts, I was still unable to establish a stable connection, which made it challenging to test the email fetching functionality.

Although I faced multiple obstacles throughout the development process, I gained valuable insights into frontend and backend development. I learned how to design a structured and visually appealing dashboard using HTML and CSS. On the backend, I explored database management with PostgreSQL and worked on implementing machine learning techniques for email categorization. Even though the system was not fully functional by the end of the day, I understood the core concepts required to build such an application.

This experience taught me the importance of persistence in problem-solving. Despite encountering errors and setbacks, I kept experimenting with different solutions to improve the dashboard’s functionality. I realized that working with real-time data, especially emails, requires careful configuration and handling. Moving forward, I plan to continue refining my approach and learning more about server connections, email fetching techniques, and machine learning implementations. Although my dashboard was not fully operational, the knowledge I gained during the process was invaluable, and I am motivated to keep improving and finding solutions to the challenges I faced.