Autobiography:

Since I was little, I always face difficulties with a positive attitude and believe that getting better than before can lead me toward a successful future. Solve the problems as soon as I can and always get ready are my finest weapons. That is why I can keep my scores being great in graduate school.

My father used to be an IT. Because of that, I knew what is “Programming” for the very first time in my life. Until college, I wrote my first code with “C.” Even though I felt complicated at first, the sense of accomplishment I got was stronger than any other hardware courses I have taken. It is why I chose to major in CS at the graduate school.

After working at Pegatron, I became the project owner of a web application called SVP. Within a year, I improved the system from a booking system to an auto-verification platform with prepared scripts. The `Report` system in SVP accelerates the whole process to another level. Users can edit log parsing rules with a friendly UI and get the required charts immediately.

I am also the project owner of another app called ARUMS. The app assists users in testing UE data like throughput, ping, RTT, etc. It also arranges the data and represents it as charts and tables for problem analysis. From starting to develop the app to deploying it to ITRI and our branch, it took me about three months.

The difficulties that we had were the following:

1. Upgrade versions:
   SVP was in phase 1 when I joined the company. The front-end framework they are using is Vue2 with options API. However, Vue2 reaches the EOL during the development. Being able to utilize the full power of Vue and also keep up with trends, we decide to upgrade to Vue3 with composition API (TypeScript version). Although it’s compatible between Vue2 and Vue3, the structure changes severely. Along with the upgrading process, I learned a bunch of front-end-related knowledge.

2. Customized booking system:
   The booking system in SVP uses an NPM library called TUI Calendar, which is open source. However, the library has a critical con. It can't display schedules with time intervals exceeding 24 hours. At first, we decide to modify the part of

the code in the library and deploy it to the container registry in GitLab. It sure solves the problem but not for ARUMS, which is a commercial web application. I decided to build our schedule by inspecting the functionalities and UI of Google Calendar. I improved a lot during the process.

3. Balance between time pressure and code quality:
   ARUMS front-end was developed by another colleague but transfer to me for looking forwards to build a more professional commercial web application. The time pressure was massive, and so was the requirement of the code quality. I chose to build our app with customized modules, which helps centralize the styles of UI and also maintainable. The difficulty is that I have to consider the general condition for modules to be used in different conditions. It improved my planning ability.

In the future, I'd like to combine the knowledge I got from these years and working experiences in a big team to confirm and enforce all the required knowledge the development team should have. Furthermore, keep up with trends, and share nowadays knowledge with colleagues. Besides my work, I hope I can be a person who has powerful techniques and can bring a fine reputation to the company.