# Project Retrospective - P1

Provide answers to the following questions. Be sure to answer the questions completely and explain your logic.

1. **How many extra days did you use for the project?**

   We didn't use any extra days as the deadline was extended by a week.

2. **Given the same goals, how would you complete the project differently if you didn't have any restrictions imposed by the instructor? This could involve using a particular library, programming language, etc. Be sure to provide sufficient detail to justify your response.**

3. **Let's imagine that your next project was to improve and extend P1. What are the features/functionality you would add, use cases you would support, etc? Are there any weaknesses in your current implementation that you would like to improve upon? This should include at least three areas you would improve/extend.**
   - Implement POSIX Client to read the complete file using cp command.
   - Replicate existing files when we have more than two active nodes.
   - Whenever a storage node goes down while writing a file chunk, write the same chunk on the available nodes without showing an error message to write again from start.
   - Support parallel writes by allocating each chunk write to a different thread.
   - Add Unit Tests

4. **Give a rough estimate of how long you spent completing this assignment. Additionally, what part of the assignment took the most time?**

- Figuring out how to identify if all the chunks have been received and when to build the entire file on read was a bit difficult.
- Replica Maintenance: It took some time for us to understand how we can handle both the Primary and Secondary node failure scenarios. Also testing this replica was a bit difficult and confusing.
- Logic to rewrite a file when the file already exists in DFS so that we always read the correct file.

5. **What did you learn from completing this project? Is there anything you would change about the project?**
    - Protobuf was something that was new for all of us that we have learned how to use that after working on this project
    - After reading the HDFS paper we have some knowledge about how DFS works but after the actual implementation of this project, we learned how to create those components from scratch.
    - Got familiar with using Netty and got a better understanding of multi-threading, blocking, and non-blocking I/O

6. **If you worked as a group, how did you divide the workload? What went well with your team, and what did not go well?**
    - Brainstorming project design and assigning components to individuals.
    - We divided the workload equally between us and made sure each of us could work on their part independently without much dependency.
    - Since there weren't many dependencies, none of us had any blockage and could work at our own pace.
    - At the end of the project, we found the classes to be lengthy and there were different methods of performing the same action. We had to spend a few hrs in refactoring the code and merging methods performing similar tasks into a util class.