ensia

Software Engineering Project

Group 1 – Team 2

Team Members:

Mohamed Ramy Guettal (Team Leader)

Mokhlis Yacine Bouyahia Imad Eddine Smail

Nour Tliba Nasrellah Kheroubi

Khaled Zaabat

Team Retrospective Analysis

During the development of our collaborative team portfolio website, the most significant technical challenge arose from dependency management and merge conflicts in the shared repository. Each member contributed an individual portfolio component, utilizing distinct front-end libraries and frameworks. Consequently, merging the collective `package.json` file became highly complex, as redundant and incompatible dependencies frequently caused build and runtime issues. This required extensive coordination and testing to ensure consistency across the project environment, demanding clear communication and precise version control.

A notable merge conflict occurred when two members mistakenly submitted pull requests to the `main` branch instead of the `develop` branch. The merging lead, unaware of this oversight, merged these requests directly into `main`, resulting in an unstable production branch. To resolve the issue, the lead executed a rollback operation on the `main` branch using Git, restoring it to the last stable commit. The erroneous pull requests were then correctly merged into the `develop` branch, followed by thorough testing before integration back into `main`. This incident reinforced the importance of consistent branch management and rigorous review before merging.

The pull request and peer review process played a critical role in maintaining software quality. Through systematic code inspection and review discussions, the team identified integration errors early and improved adherence to coding standards. This process not only reduced technical debt but also enhanced collaboration by fostering accountability among members. Overall, despite initial coordination challenges, the structured Git workflow and disciplined peer review significantly contributed to the robustness and maintainability of the final software artifact.