In this project my responsibility was to implement checkout and merge features in our minigit version control system. These commands are useful for handling branches and switching between different versions of a project.

My checkout command is designed to allow users to move to a different commit or branch. It works by switching the working directory which is used to match the selected version. The first thing the system does is checking the input if it is a valid commit hash or branch name. After that, it tries to find the correct commit. Then the current working directory will be cleared except for the .minigit folder. Next, files listed in the selected commit are restored from the stored blobs in the objects folder. Then finally, our system will update the head file to point to the new commit and clear the staging index to prepare itself for new changes.

When we come to the merge command, it is created to allow our users to combine changes from one branch into the current branch. The advantage of this is while using multiple versions of a project. My program first reads the commit history of both branches and this helps it to find the most recent common base or ancestor. Then it tries to compare the files in this base, in the current commit and also in the branch which is to be merged. For each file the system makes a decision whether to keep the current version, or to use from the other branch or in the worst case it tries to handle a conflict by updating the file with clear conflict makers and by notifying the user. Finally a new file will be created that combines the two branches and then the program will update the head to point to the new merged commit.

Working on these two commands (checkout and merge), I now understand how version control systems like git can manage some file changes and also branch histories. I also learned how to detect and handle merge conflicts and how to read and write structured data to manage commits. The whole project taught me how control systems like git work and it made me familiar with them and gave me confidence in building more advanced command-line programs.