Report

In this project, my responsibility was to implement the core system initialization for MiniGit. This involved creating the foundational repository structure and setting up the initial files and folders required for version control. The goal was to simulate Git’s initcommand to prepare a working repository environment

The init command creates a hidden .minigit/ directory in the current working folder. Inside .minigit/, it creates subdirectories such as objects (to store file blobs) and refs (to store references like branches).  
 Two important files are created:

-.minigit/HEAD — where a pointer to the current branch is kept, initialized to refs/main.

-.minigit/refs/main — an empty file for the main branch reference.

This setup allows the system to manage commits, file snapshots, and branches similarly to Git.

The implementation was done using the std::filesystem library for directory and file operations. Before creating the repository structure, the program checks if .minigit/ already exists to avoid an existing repository.  
 The init function handles creating directories and files as needed. It writes the initial content of HEAD to point to the main branch and creates an empty file for the branch itself.  
 A basic command was implemented to accept user inputs such as init and exit, with plans to add more commands later.

Through this, I gained practical experience with file and directory manipulation in C++. I learned how to organize project data on disk in a way that mimics real version control systems. Designing the initialization step gave me insight into how Git prepares a repository before any commits are made.  
 And I improved skills in building interactive command-line programs.