Git Version Controlling Practice Report

Team Details

Team Leader: Tanvir Alam
GitHub Username: tanvirdraw-hub
Repository Name: git-version-control-practice-team1
Repository URL: https://github.com/tanvirdraw-hub/git-version-control-practice-team1
Team Members: Islamul Haque Sami, Helal Uddin

1. Adding Collaborators

The team leader created a GitHub repository named *git-version-control-practice-team1* and added all members as collaborators using **Settings** → **Collaborators** → **Add People**. Each member accepted the invitation and gained permission to push changes to the repository. This setup enabled all members to contribute to the same project in real-time.

2. Creating Branches and Committing Files

Each member cloned the repository using GitHub Desktop, then created a new branch named after themselves. They added their Cheat Sheet PDF file, committed the changes, and pushed to the origin branch.

Example (in GitHub Desktop):
→ Current Branch → New Branch → Create
→ Add File → Commit → Push origin

3. Merge Operation

The team leader switched to the *main* branch and merged all member branches using GitHub Desktop:
Branch → **Merge into Current Branch** → **Select Branch**
After merging, all Cheat Sheet PDF files became available in the main branch, combining the team’s contributions.

4. Rebase Operation

Each member rebased their branches to stay updated with the main branch. Rebasing helped maintain a clean commit history and avoided unnecessary merge commits.

GitHub Desktop: Branch → Rebase Current Branch → Select main

5. Cherry-pick Operation

The team leader demonstrated a cherry-pick operation to apply a specific commit from another branch to the main branch. Using GitHub Desktop, the leader selected the commit in the History tab and chose **Cherry-pick Commit.** This allowed importing only a particular change without merging the entire branch.

6. Final Repository and Outcome

After performing Merge, Rebase, and Cherry-pick, all files and commits were pushed to GitHub. The repository history now reflects all teamwork operations clearly. Screenshots of Collaborators, Branches, and Commit History were included in the submission report as visual proof.

7. Conclusion

Through this practice, the team learned how to collaborate efficiently using Git and GitHub Desktop. They understood how to manage version control operations such as branch creation, merging, rebasing, and cherry-picking. This exercise strengthened the team's understanding of real-world collaborative development workflows.