Class Summary (Post-Class Update)

Hi!

Thank you for attending today's class. Here's a quick summary of what we covered:

Key Highlights from Today's Class

- Topic Covered: Git Branching, Merging & Conflict Resolution
- Main Concepts Learned:
    - Branches act as independent lines of development, allowing safe experimentation and parallel work without risking the main codebase.
    - Merging integrates changes from one branch into another, with options like fast-forward merges or rebasing for cleaner histories.
    - Merge conflicts occur when changes overlap but are normal and resolvable through manual editing, followed by staging and committing.

See you in the next class!

- Practical Examples/Case Studies Discussed:
    - Live terminal demo: Created branches, made changes, merged without conflicts, and intentionally induced a conflict by editing the same line in README.md on separate branches, then resolved it.
    - Hands-on exercise: Students cloned a repo, created a feature branch, added changes, merged back to main, and practiced resolving a guided conflict scenario.
- Tools/Resources Introduced:
    - Git commands: Including git checkout -b, git merge, git rebase, git stash, and git reflog for troubleshooting.
    - Interactive tools: GitHub for pull requests and code reviews; recommended resources like git-scm.com docs, learngitbranching.js.org tutorial, and the free "Pro Git" book.

Assignments / Next Steps

- Practice branching in a personal project: Create at least two feature branches, make commits, merge one, and resolve a simulated conflict in the other before the next session.
- Review the session resources: Visit learngitbranching.js.org for interactive practice and git-scm.com for command references. Next class: Session 9 on Git in CI/CD Pipelines—come prepared with questions on team workflows!