
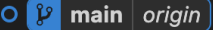


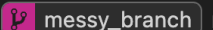
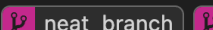

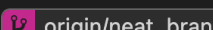


Github link: <https://github.com/rishikagudise/git-collaboration.git>

Graph	Description	Date	Author	Commit
	 <b>main</b> <i>origin</i> added a read me doc	25 Feb 202...	rishikagudise	2d4a8e30
	 <b>develop</b> <i>origin</i> Merge branch 'conflict_branch' into de...	25 Feb 202...	rishikagudise	9d6ae011
	added a line in develop branch is going to create a conflict?	25 Feb 202...	rishikagudise	2836c353
	 <b>conflict_branch</b> <i>origin</i> added a line in conflict branch	25 Feb 202...	rishikagudise	b3528acf
	 <b>messy_branch</b> messy branch 3rd commit	25 Feb 202...	rishikagudise	43491ce4
	messy branch 2nd commit	25 Feb 202...	rishikagudise	84e5aa1d
	messy branch 1st commit	25 Feb 202...	rishikagudise	f848dbfa
	 <b>neat_branch</b>  <b>origin/messy_branch</b> neat_branch 2n...	25 Feb 202...	rishikagudise	aa9bf895
	neat_branch 1st commit	25 Feb 202...	rishikagudise	26db74c8
	 <b>origin/neat_branch</b> develop branch initial commit	25 Feb 202...	rishikagudise	5805a29e
	initial commit	25 Feb 202...	rishikagudise	d7e6446f

Both merges above are fast-forward merges. Explain.

- Both branches, neat\_branch and messy\_branch, are fast forward merges. This is because both branches had a direct path to merge into the develop branch. In this process, commits were added to both neat\_branch and messy\_branch while the develop branch received no commits. Thus, the develop branch was able to point towards the latest commit of each branch, resulting in a fast-forward merge for both the neat\_branch and messy\_branch.

In your own words, write down a scenario that will involve a merge conflict between develop and conflict\_branch.

- If I have a paragraph that is placed in the same spot in both the develop and conflict branch within the index.html and if they hold different content, a merge conflict will occur. This will happen because Git will not know which content to use within that line. When there are these types of overlapping changes, manual resolution is required

In your own words, write down a scenario that will NOT involve a merge conflict between develop and conflict\_branch despite some change on the same file on both branches.

- If I just add a new line in the conflict\_branch's index.html and not in the develop branch's index.html, there will be no merge conflict because this is not a change that will overlap with a change in the develop branch. Git will merge these non-overlapping changes automatically.