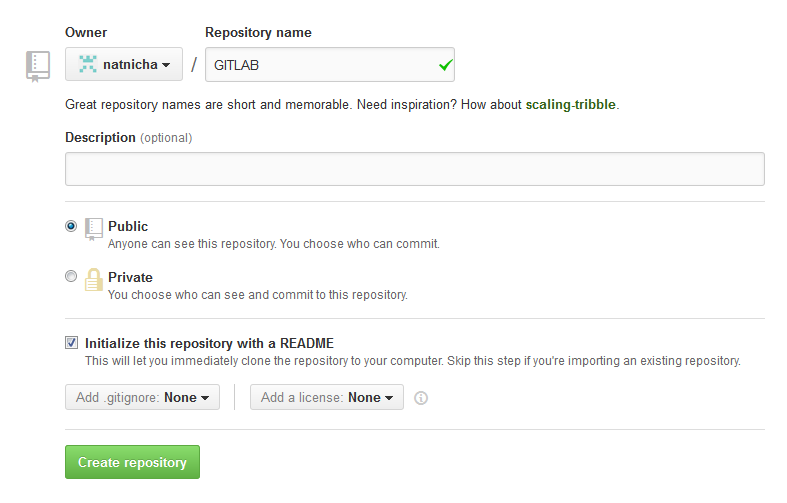
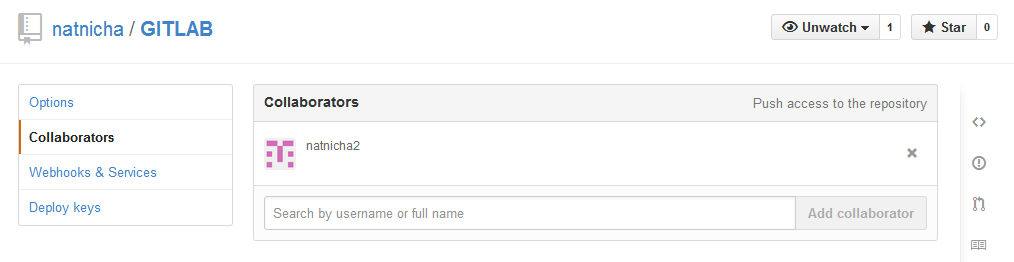
1. Create a blank repository in GitHub under User A’s account called “GITLAB” with 1 Readme file. Print hub status. Both users A and B will use this repository. Add User B as collaborator to this repository.





2. In some path create a folder called “CPE333-A” for User A. Make this a git repository and pull data from GitHub. Show status.



3. In some path create a folder called “CPE333-B” for User B. Make this a git repository and pull data from GitHub. Show status.



4. User A. Use file manager to put 2 C programs (Program #1 and Program #2), and 1 MS Word document into CPE333-A.

5. User A. Add all files to git. Then commit. Show Status.

6. User A. Push your local repository to the GitHub repository. Show screen.

7. User B pulls data from GitHub. Show status.

8. User B change 1 function in Program #2. Print local status User B.

9. User B create a new C program Program #3. Do not put into Git yet. Print local status User B.

10. User B adds Program #3 to Git control (git add). Print local status User B.

11. User B commits changes. Print local status User B.

12. User B pushes changes to the GitHub repository. Print hub status. Print User B status.

13. User A change something in C Program #1. Print local status User A.

14. User A stages its changes and commits at once. Print local status User A.

15. User A pulls repository from GitHub.

16. User A pushes changes to GitHub repository. Print status User A.

17. User A pulls repository from GitHub.

- All 3 repositories have same copy now.

18. Then User A changes Program #1 at 2 places and User B changes same Program #1 at 2 places. Both add and commit changes. Print status of User A and User B.

19. User A pulls from GitHub and then pushes to GitHub. User B then pulls and pushes to GitHub, but there’s conflict. Resolve this merge conflict for User B and commit change. Show each step.

20. Then User A pulls the changes. Show the Program#1 for both users that they are now the same.