**Git Commands Working in a group**

1. Project Manager sets up a project.
   1. Create a new project on the GitLab website and save the git commands in a file git-commands.txt. You will need to send the git add remote origin command to your group members,
   2. On your computer initialize git in your project folder:
      1. git init
      2. git remote add origin git@gitlab.uvm.edu:YOURNAME/YOUR-REPO-NAME.git
      3. git add -A
      4. git commit -m "message about what you did"
      5. git push -u origin master (-u is just done first time)
      6. Your project is now all set up (no live set yet but you should be able to do that).
   3. On the GitLab web site add your group members to the project.
2. Group members should set up their local machine:
   1. Make a folder for the project.
   2. cd to the folder.
   3. git init
   4. Add the remote that the project manager emailed  
      git remote add origin git@gitlab.uvm.edu:PROJECTMANAGERNAME/YOUR-REPO-NAME.git
   5. You will need to pull the files to your computer but I will cover that next.
3. Group work flow. **Every Group member** needs to follow these steps **every time**without fail before working on any file. All commands are for your local computer.
   1. Make sure you have the groups master copy:git pull origin master
   2. Check out a branch to work on so you don't mess up the master. Name the branch with your initials (all caps) followed by an underscore and a short name of the task you are working on. For example I am going to work on the navigationgit checkout -b RE\_nav
   3. Make changes to your code as you would normally.
   4. Ftp your files to the server so you can make sure it works (ie in netbeans run the project).
   5. Verify your changes are working in the browser (ie dev folder)
   6. Add your changes to git (you can do git status first to make sure the files changed are the ones you expected to have changed).git add -A
   7. Commit your changes locally on your machine:git commit -m "Added new menu item"
   8. Push your branch to GitLabgit push origin RE\_nav
   9. On the GitLab web site look at your branch and submit a merge request. A member of the team should review the code to make sure it meets the teams standards (ie code review). At this point make sure that the new code will not cause your code to break.
   10. Once the code is approved on the GitLab website (people can make comments) the manager would then **Accept the Merge Request**
   11. After the merge you can delete the branch as it's not needed anymore
   12. The person who created the branch should now delete the branch locally on their machine. Before deleting you need to switch back to the master.git checkout mastergit branch -d RE\_nav
4. To summarize: work flow process (this is what you do daily):
   1. On your computer go to the develop folder you have set up for the project.
   2. git pull origin master
   3. git checkout -b INITIALS\_task
   4. Create/edit your php pages.
   5. Ftp your files to your develop folder (ie run netbeans project).
   6. Verify that the pages are working on the develop web site.
   7. In your computers project folder
      1. git status
      2. git add -A
      3. git commit -m "message about what you did"
      4. git push origin INITIALS\_task
   8. Submit a merge request from the GitLab web site.
   9. Request is reviewed on the GitLab web site.
   10. Project manager Accepts the merge and deletes the branch on the GitLab website.
   11. On your computer:
       1. git checkout master
       2. git branch -d INITIALS\_task
   12. Repeat above till project is done.
   13. Manager would then pull changes live.
   14. ssh to your hosting account:  
       ssh w3.uvm.edu -l yourusername
   15. cd to your live folder.
   16. git pull origin master
   17. Verify that the pages are working on the live web site.
5. You messed up and have merge conflict you cannot fix. What do you do?
   1. Remember all your files are on GitLab, so relax.
   2. Rename your project folder to projectfolder-OLD. The name really does not matter it is just a way to save what you have done (delete it later when you have fixed everything).
   3. Create a new folder with your project name.
   4. Set up netbeans project for this new folder.
   5. Set this new folder up as a git repository.git init
   6. Add the remote to projectgit remote add origin git@gitlab.uvm.edu:PROJECTMANAGERNAME/YOUR-REPO-NAME.git
   7. Follow the normal steps for the group git work-flow, Ie pull origin master, check out branch.
   8. Look at the file that had the merge conflicts and just make your changes. If I know what is messed up I just open the old file, copy the text. Open the new file i just pulled down and paste the old text. Since you pulled the master and checked out a branch you should not have merge conflicts.
   9. Continue on with the normal steps for the group git work-flow.
   10. At some point in time you will need to learn how to deal with merge conflicts but if you are in a crunch for time this often works.