# Code versioning – GitHub
## (Introduction to GitHub)
**Arun Avinash Chauhan**


## Introduction:

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. This assignment teaches you essentials of GitHub like repositories, branches, commits, and pull requests. You'll create your own 'Hello World' repository and learn GitHub's pull request workflow, a popular way to create and review code. GIT is a versioning software, and GitHub is something built on top of GIT (You use the GitHub client – it provides a web interface, for you to interact and work with GIT). You can use Git without GitHub but you cannot use GitHub without Git.

To complete this assignment, you need a GitHub account and Internet access.


## Creating your GitHub project (the crux of the second assignment)

The following tutorial (URL) shows step wise instructions which help solve the assignment. It shows step wise instructions to:

(2) create your very own *repository*, (3) make a *branch*, (4) make changes to the *branch* and *push* to GitHub as *commit*, (5) and then *pull* and *merge* the changes back to main *branch*. The link is mentioned below

https://docs.github.com/en/get-started/quickstart/hello-world

Note: Numbers in brackets above indicate the steps in the assignment covered in the tutorial.

Once you are done with following (understanding) the tutorial in the link above, I would like you all to do the following. This is the second lab assignment:

1.  Create an account at GitHub – steps for which are straightforward
    a.  You can check the section titled 'Creating an account on GitHub' below
2.  Create a repository (**ensure it is public**), add a .txt or .cpp file to repository (2M)
3.  Create a new branch (1M)
4.  Make changes to file in branch and push to GitHub as commit (1M)
5.  Open a pull request and merge (2M)
6.  Forking a repository (4M)
    a.  To do this part (#6), check the section titled 'Forking a repository' below
    b.  (#6) of the assignment will require installing Git on your local computer. Details below
    c.  Note: **Ensure that your repo is a public repository**


*Note 1: Select 'public repository' and add check 'Add README file' while creating new repository.*
*Note 2: Browse through the various tabs available in your repository and check various options.*

Whatever is covered in the tutorial will be explained in the lab as well. Meanwhile, you can also follow the tutorial and complete the assignment before you attend lab on Monday / Friday. You can also complete the assignment in lab / after lab. The deadline will be mentioned in the lab.

**Submission:**
1. In lab evaluation, OR
2. A Google form submission link will be provided in the batch email. In the Google form, mention your name, roll number and the GitHub URL of your profile (check screenshot below). Also ensure that your repositories are **public** (in screenshot, check arrows pointing to the privacy status of my repositories, for instance)
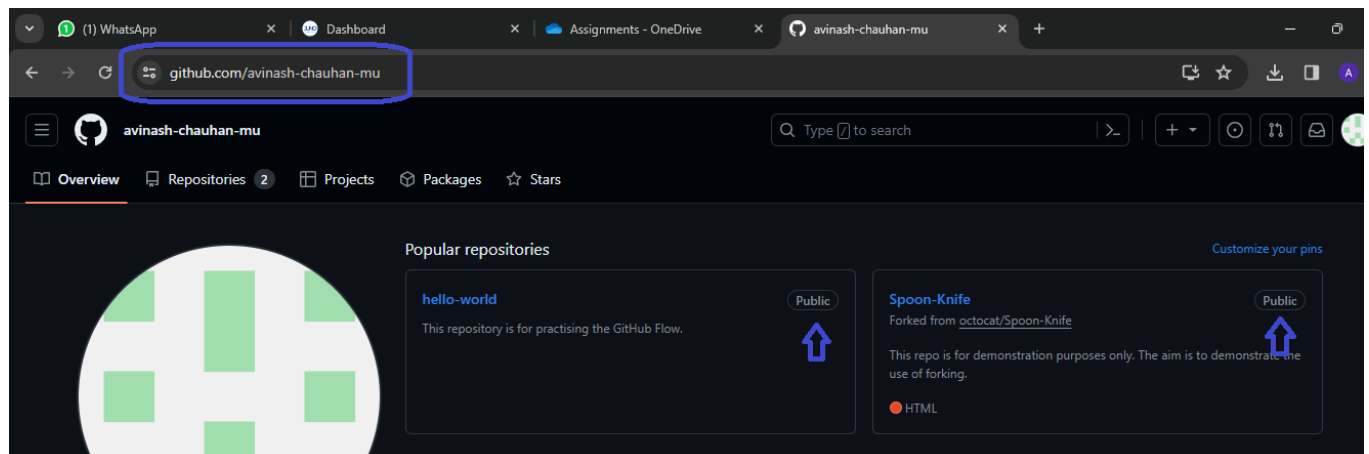


## Creating an account on GitHub (needed for point #1 in Assignment):

Create a personal account on GitHub, which can be created under GitHub for free. To create a free personal account, go to the following link https://github.com/ and select 'Signup'. From that point onwards, the rest of the steps are self-explanatory. Once you make an account, when prompted for how many team members you will work with, select 5-10. Also, select 'Student' when asked who you are (of course). When prompted for what specific features you will be interested in, select 'Collaborative Coding' (that shall do for now). Then select the free plan 'Continue with free' and viola, you are ready!

## Forking a repository (needed for point #6 in Assignment):
*This tutorial requires you to install Git on your local computer. Steps to install Git (on Linux / Windows / Mac) are mentioned here*
*This part of the assignment shows you that you can branch, make changes, and PUSH via GIT*

Forking is at the core of social coding at GitHub. Creating a fork is producing a personal copy of someone else's project. Forks act as a sort of bridge between the original repository and your personal copy. You can *submit Pull Requests* to help make other people's projects better by offering your changes up to the original project.

After forking a certain repository (Spoon-knife), make some changes to the project, and submit a Pull Request as practice. The entire tutorial on forking is available here: https://docs.github.com/en/get-started/quickstart/contributing-to-projects

Some notes on the link above (on forking):

While the following command is mentioned in the link

```
git branch BRANCH-NAME
git checkout BRANCH-NAME
```
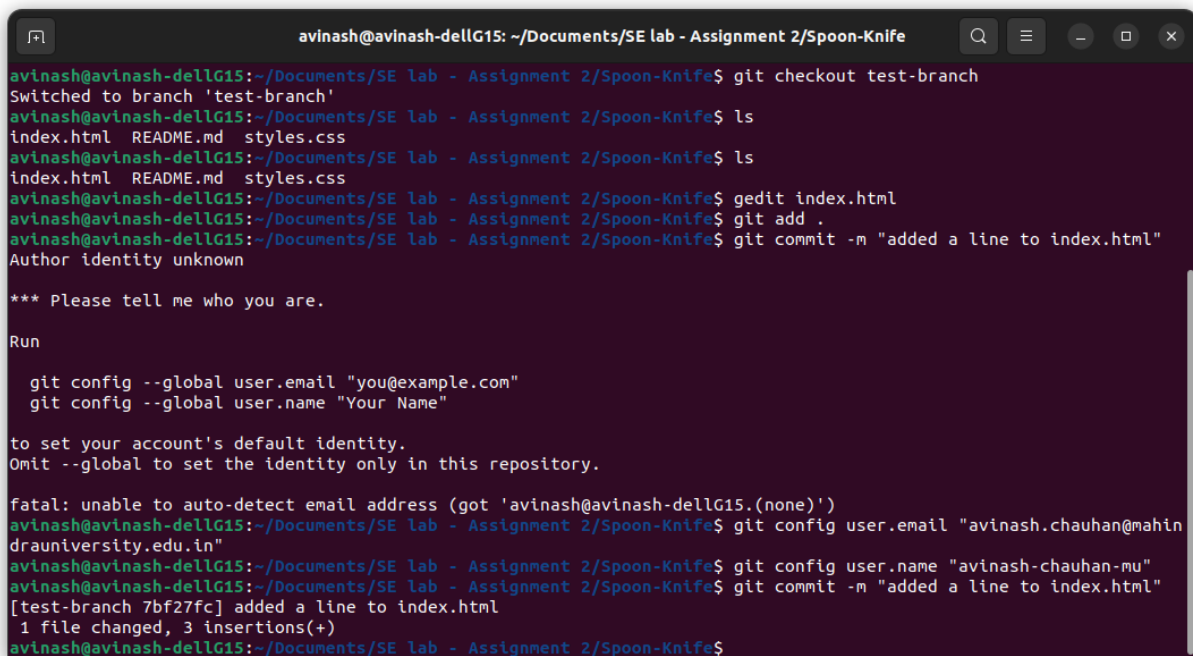
It is unclear where to run the command. Therefore, go to the folder that was created after your local clone command is run, in this case, the name of the folder is 'Spoon-Knife'. Once inside that folder, run the commands mentioned above, where you give BRANCH-NAME a name of your own (I used 'test-branch' as my BRANCH-NAME. This will be a branch of the main branch of Spoon-Knife, where you can work on your changes to code / files.

Note that the following commands

git add .

git commit -m "a short description of the change"

will throw an error stating that the author identity is unknown. To resolve it, follow the screenshot below:



Further, note that right now the 'git PUSH' command is not working (only LINUX) (Git has removed support for password authentication – see screenshot below), and we shall deal with it in the lab (while you are also welcome to work on this issue yourselves). Once the code is 'PUSH'ed, only one step remains – to do a pull request, which is to merge the changes made in the forked repository branch with the main branch.

```
Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'avinash@avinash-dellG15.(none)')
avinash@avinash-dellG15:~/Documents/SE lab - Assignment 2/Spoon-Knife$ git config user.email "avinash.chauhan@mahin
drauniversity.edu.in"
avinash@avinash-dellG15:~/Documents/SE lab - Assignment 2/Spoon-Knife$ git config user.name "avinash-chauhan-mu"
avinash@avinash-dellG15:~/Documents/SE lab - Assignment 2/Spoon-Knife$ git commit -m "added a line to index.html"
[test-branch 7bf27fc] added a line to index.html
 1 file changed, 3 insertions(+)
avinash@avinash-dellG15:~/Documents/SE lab - Assignment 2/Spoon-Knife$ git push
fatal: The current branch test-branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin test-branch

avinash@avinash-dellG15:~/Documents/SE lab - Assignment 2/Spoon-Knife$ git push --set-upstream origin test-branch
Username for 'https://github.com': avinash.chauhan@mahindrauniversity.edu.in
Password for 'https://avinash.chauhan@mahindrauniversity.edu.in@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#clonin
g-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/avinash-chauhan-mu/Spoon-Knife.git/'
avinash@avinash-dellG15:~/Documents/SE lab - Assignment 2/Spoon-Knife$
```

For now (for those on Linux or Mac), you can instead make a branch, make changes to index.html, and push, then pull via GitHub web interface. (Recollect first part of the assignment, we made a branch and made changes via the GitHub web interface)

UPDATE: If you are doing these steps (changes and PUSH via GIT client) on Windows, you will be able to PUSH from GIT client!

Those on Linux/ Mac, create a branch on the repository you forked (the forked repository will be available in your GitHub). Choose a file, for instance I chose index.html, and make changes to the file. Once done making changes to index.html, make a 'Pull' request (check screenshot below. Note: there are more than one ways to create a pull request, you may explore the web interface of GitHub)
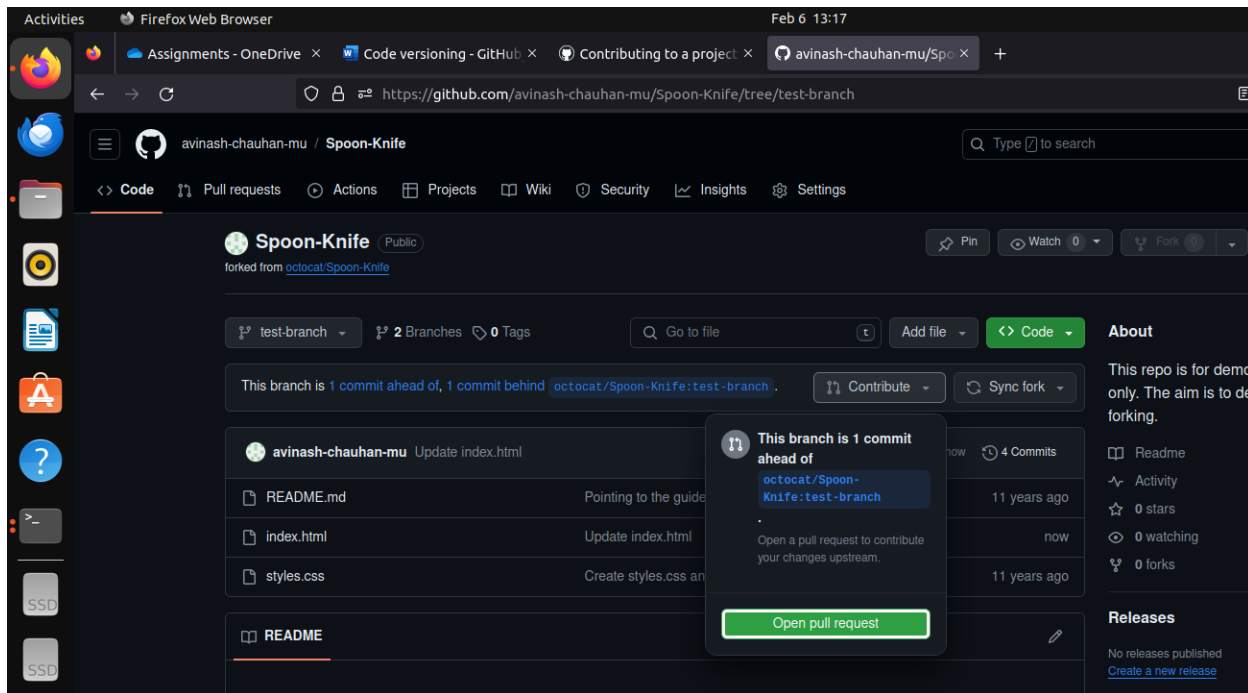
Once you make a pull request, the changes are shown (whether we are able to merge, or if there are conflicts, and so on – check screenshot below) and in case there are no conflicts, you can create a pull request successfully. At this point, the assignment comes to an end (Octocat will not accept your pull request given it is a sample project by GitHub and there will be thousands of pull requests). OPTIONAL: You may instead fork a friend's project, make a branch, commit changes, make a pull request, and your friend can accept that pull request and merge.