Aandishah Tehzeeb Samara

IDCE 30262: Web Mapping and Open-Source GIS

Professor Korde

31st October 2021

**Lab 8: Collaborative programming with GitHub**

* ***Question 1: What are the temporary interaction limits allowed on GitHub? (4 points)***

Enabling an interaction limit for a repository restricts certain users from commenting, opening issues, creating pull requests, reacting with emojis, editing existing comments, and editing titles of issues and pull requests.

When you enable an interaction limit, you can choose a duration for the limit: 24 hours, 3 days, 1 week, 1 month, or 6 months. After the duration of your limit passes, users can resume normal activity in your repository.

There are three types of interaction limits.

* + Limit to existing users: Limits activity for users with accounts that are less than 24 hours old who do not have prior contributions and are not collaborators.
  + Limit to prior contributors: Limits activity for users who have not previously contributed to the default branch of the repository and are not collaborators.
  + Limit to repository collaborators: Limits activity for users who do not have write access to the repository.
* ***Question 2: What are the differences between "Clone" and "Fork"? Which one do you think is more useful (or you will be more likely to use) in your own project? (4 points)***

Any public Git repository can be forked or cloned. A fork creates a completely independent copy of Git repository. In contrast to a fork, a Git clone creates a linked copy that will continue to synchronize with the target repository.

For our project, it will make sense to clone the repo so we (all the team members) can keep collaborating by adding to the repo.

* ***Question 3: What is a pull request on GitHub? Why is it necessary to have pull requests? (4 points)***

Pull requests let me tell others about changes I pushed to a branch in a repository on GitHub. Once a pull request is opened, I can discuss and review the potential changes with collaborators and add follow-up commits before my changes are merged into the base branch.

* ***Questions 4: Explore the functionalities of GitHub Desktop. List two things that you see are different in the GitHub desktop from the GitHub web interface. (4 points)***

The fundamental difference between GitHub it contains no working copy and is able to receive pushes. You cannot use the desktop to see the files the way you can on the web interface. You are able to make any changes to the local copy and then track those changes in the desktop, but unlike the web interface, which you can use to make direct changes to.

* ***Questions 5: Modify any file in your cloned repository. Take a screenshot of the tracked changes in GitHub desktop (The screenshot will look something like the previous figure) (4 points)***

Text

Description automatically generated