**An Introduction to GitHub**

**Introduction**

Being able to work on a single version of a project with a group of people not only relieves a lot of the hassle with keeping track of progress and versions, but it also helps improve the overall quality of work because everyone is on the same page. While Google Docs and Microsoft SkyDrive are great collaboration tools in their own right, they are not exactly meant for coding. Enter GitHub: “a web-based hosting service for software development projects” that allows developers to “Build software better, together”. GitHub enables a user to create an unlimited number of public repositories in which each repository can have an unlimited number of collaborators. Not only does this allow you to work on cool projects with your colleagues, but it also allows anyone with a GitHub account to copy your repository to their computer, look at your awesomeness, and potentially suggest improvements to your code. Having one place to store all of your projects not only eliminates potential confusion of project locations, but it is also fantastic to have when going for job interviews because the most current version of any project is right at the employer’s fingertips.

**Purpose**

The purpose of this exercise is to get you familiarized with the basics of GitHub including setting up an account, creating your first repository, copying it to your computer, making some changes in the repository, and sending said changes back out to GitHub proper. We will also go over how to add collaborators to a project. By the end of this exercise, we hope you all will begin to feel comfortable with the basic principle and concepts of GitHub.

**Before We Begin…**

Please, if you all run into any trouble during the course of this exercise, do not hesitate to seek help me. I want to make sure you all get off on the right foot. I’m here for you!!! ☺

**Procedure**

**Part 1: Setting Up an Account**

In order to become apart of the GitHub community, you first need to set up an account. So let’s get to it!

1. Navigate to the website [https://GitHub.com/](https://github.com/).
2. On the GitHub home page, you should see the GitHub slogan “Build software better, together.” On the righthand side of the slogan is the registration field. Fill out the three fields accordingly. Make sure to use information that you will remember, especially your username (I used my JMU e-ID when I registered). Click Sign up for GitHub.
3. After you have signed up, GitHub will ask you to choose a plan. PLEASE DO NOT SELECT ANYTHING WITH A DOLLAR FIGURE (it is highly unnecessary for our purposes)! Select the plan that is Free and press continue.
4. After selecting your plan GitHub should reroute you to your dashboard. You might see a message to confirm your email. Go ahead and do just that so you don’t have to stare at that message forever. Congratulations! You are now a member of the GitHub community!!!

**Part 2: Creating Your First Repository**

Now that we have your account set up, it’s time to create your first repository. We’re going to denote this repository as the ‘Hello Repo’.

1. On your GitHub dashboard, navigate over to the right side of the page and locate the section that says Your repositories. Click the button next that says + New repository.
2. On the next page that pops up, go ahead and enter a repository name (in this case, feel free to type Hello Repo or something similar).
3. You may add a description to the repository if you wish although in this case it is unnecessary.
4. Make sure the Public satellite button is filled in and to check the box that says Initialize this repository with a README (more on the README in a second). Click Create repository. Github will direct you into your brand new repository.

**Part 3: Installing GitHub and Cloning Your Repository to Your Computer**

Having created your first GitHub repository, let’s go ahead and bring it onto your computer. To do so, we’ll use Command Prompt/Terminal

1. Inside your brand new repository, navigate over to the right side of the page and look for a field titled HTTPS clone URL. Copy the URL listed below. We will use this URL to bring your repository onto your computer.
2. Depending on your operation system, navigate to <https://windows.github.com/> or <https://mac.github.com/>. Download GitHub and follow the onscreen instructions. The reason we are downloading GitHub is so that we can use some of its commands in Command Prompt/Terminal.
3. Once you have GitHub installed, depending on your operating system, open Command Prompt (Windows) or Terminal (Mac).
4. In Command Prompt/Terminal, type git config --global user.name “YOUR NAME” and git config --global user.email “YOUR EMAIL ADDRESS”. This sets up GitHub locally on your computer so that your commits/pushes will be labeled correctly.
5. Now, navigate to a directory in which you would like to clone your repository to (I suggest Desktop or Documents).
6. From here, type git clone <https://github.com/your_username/your_repo.git>
7. Wait a second for GitHub to clone your repository and then check the directory you chose to make sure it is there. Note: make sure to leave your Command Prompt/Terminal open!

**Part 4: Adding/Editing Files in Your Repository and Committing to GitHub**

Now that you have your repository on your computer, let’s add a couple of test files to it, commit the changes, and push them out to GitHub for everyone to see.

1. On your computer, navigate into your repository. It will appear as a folder in the directory you downloaded it to.
2. Let’s title the test files Hello and World respectively. In Command Prompt/Termainl, type touch Hello.txt and touch World.txt. Note that the file format we are using is plain text.
3. Open up the two test files by typing start Hello.txt and start World.txt (Windows) or open Hello.txt and open World.txt.
4. Add some text to these files. It can be anything (just make sure it’s appropriate). Make sure to save each file when you are finished.
5. On your open Command Prompt/Terminal type git add \*. This tells GitHub that changes have been made to your repository and to put said changes in a staging area for the next commit.
6. Next, type git commit –m “Added two test files”. This step provides a title for the changes you just staged in the step above and prepares the files to be sent to the GitHub server.
7. Finally, type git push origin master. This command pushes the changes to the repository out to the GitHub server online. If all goes well and no errors show up, you should be able to see the changes you made to your repository online (you may need to refresh the page).

**Part 5: Adding Collaborators to a Repository**

The last portion of this exercise will take you through adding collaborators to your project/repository. This is a fairly straightforward process but it is especially crucial for allowing others to add onto your work.

1. In your repository on GitHub’s website, navigate over to the right-hand side of the page and click on Settings.
2. On the Settings Page, look on the left-hand side of the page and click on Collaborators (you may need to confirm your password to continue).
3. Under Collaborators, type in the username of the person you wish to add as a collaborator and click on that username from the dropdown menu when it appears. Click Add collaborator.
4. And that’s it! You now know how to add collaborators to your project/repository.

**Wrap-Up**

Congratulations! You have just downloaded and hopefully become familiar with one of the major key tools to this class!!! Over the course of time, we feel that you will see how GitHub allows you to share your work with others and manage/work on files in a simpler, cleaner, and more group-oriented way.

**References**

1. <https://github.com/>
2. <https://help.github.com/>
3. <http://en.wikipedia.org/wiki/GitHub>
4. <http://www.youtube.com/watch?v=0fKg7e37bQE>