**Software Development and Testing IT7320**

**LAB WORK**

Team Members: Submitted to:

Anjali Khandelwal Chalinor Baliuag

Kwino

Kwino

Ralph Saplan

Thanh Huynh

Contents

1. Launch terminal of git shell.

2. Check Git installation with git version check

3. Lunch GitHub for desktop.

Version control system is a software that helps software developers to work together and maintain a complete history of their work without overwriting each other’s changes.

Git is a distributed version control system . DVCS clients not only check out the latest snapshot of the directory but they also fully mirror the repository. If the sever goes down, then the repository from any client can be copied back to the server to restore it. Every checkout is a full backup of the repository. Git does not rely on the central server and that is why we can perform many operations when we are offline. We can commit changes, create branches, view logs, and perform other operations when we are offline. We require network connection only to publish our changes and take the latest changes.

We can install git from www.git-scm .com. It helps developer to work on project on their local computer and they can copy there whole local repo on the server using github to collaborate with other team members.

**1.Launch terminal of git shell:**

Here are all screenshots of installation of git on local computer.

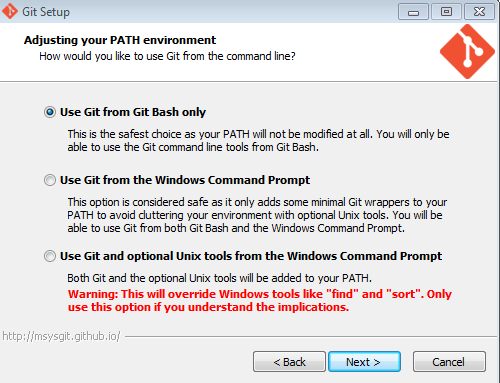
1. Open [www.git-scm.com](http://www.git-scm.com) to get the download link.



1. Download git by clicking on download link. Latest version 2.5.0.

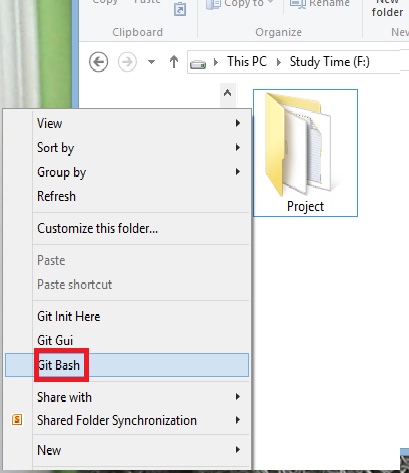


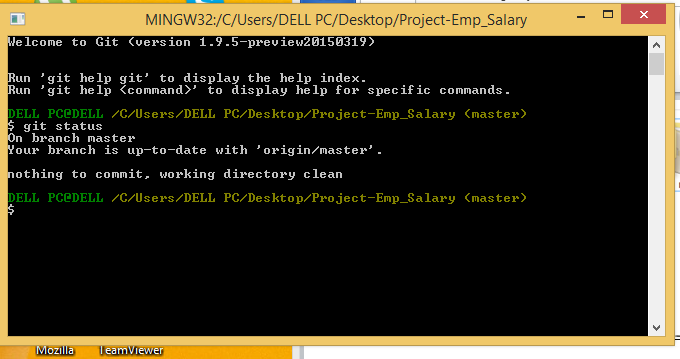
1. Follow the instructions and complete the setup of git. Here are some screenshots of installation of git.





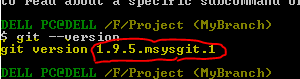
After Completing the Git installation we can open a folder in git bash and after initialize it as a working directory we can start working with git.

****



**2.Check Git installation with git version check:**

To check git version type command “git –version” in git bash.

**** As command line is responding to git commands, it shows git has been installed on computer successfully and it is in working state.

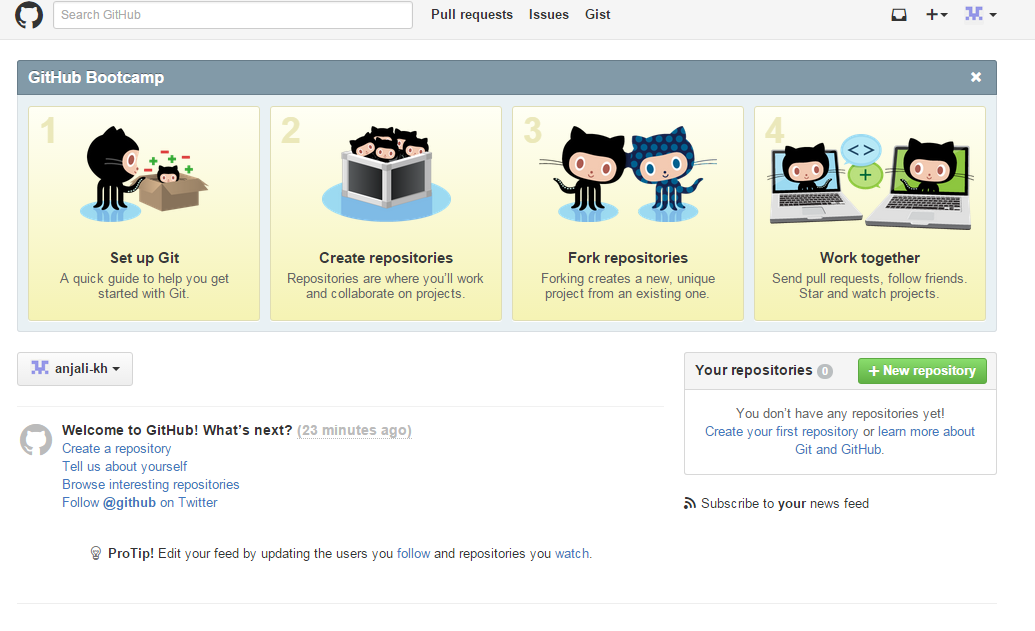
**3.Launch GitHub for desktop:**

GitHub desktop is an application which can be installed on a local computer to manage remote repositories. Moreover we can perform git commands like push, commit directly by clicking on it from the application rather than typing them in command line environment. In order to setup github desktop application we need git hub account.

Sign up on GitHub website [www.github.com](http://www.github.com) to create GitHub account.

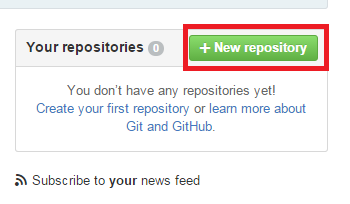


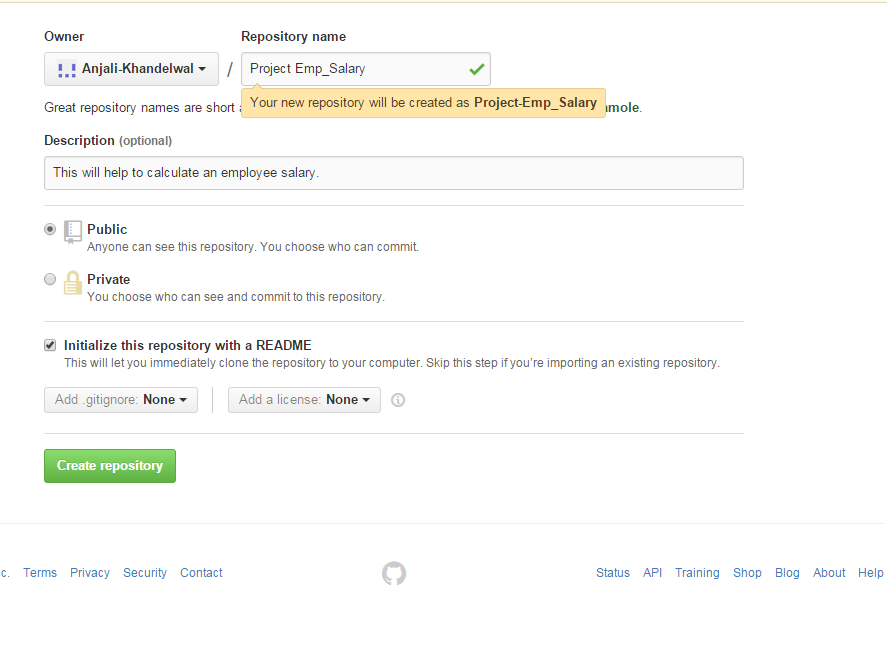
1. After that we are on the new webpage we need to sign up by type in the details like username , email address, password .
2. After finishing sign up we can sign in any time when we need to access GitHub account by click on sign in and type in username and password on website [www.github.com](http://www.github.com) .

The above screenshot is the welcome page of GitHub account.

Now we can copy local repositories to the git hub account and we can get all the changes updates in our local repo from remote repo (which we create in Github).

1. We can create remote repo just by clicking on New repository button in github account.

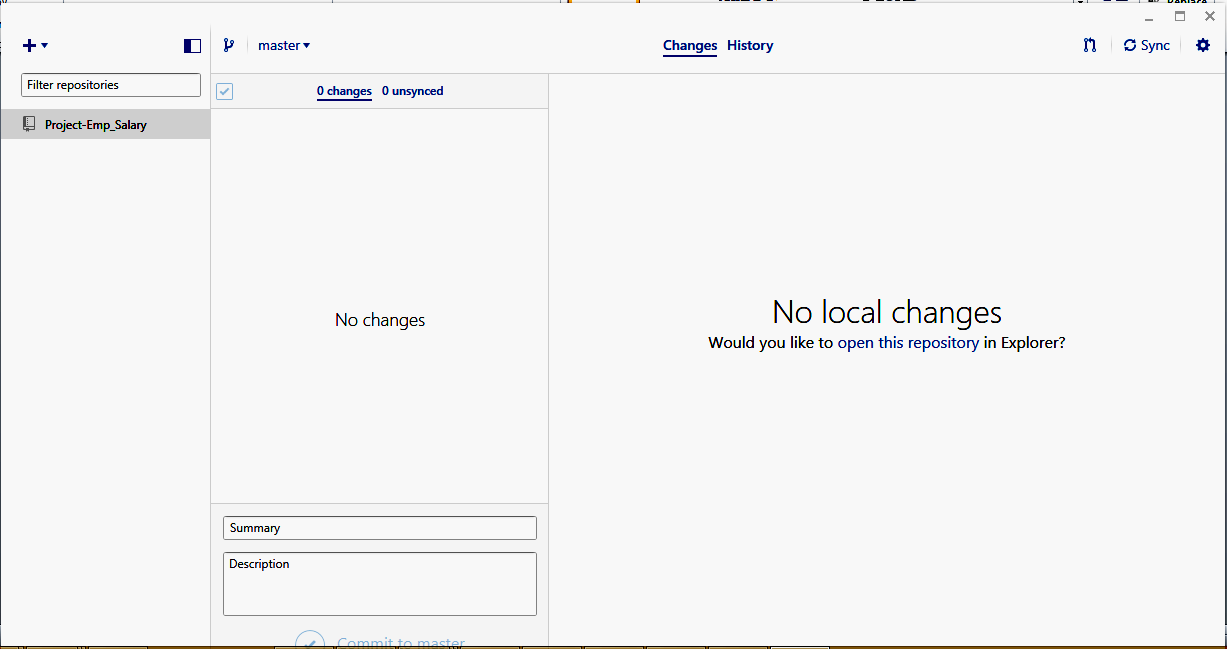




To launch a GitHub Desktop : If we have a github account we can launch github desktop by download it from <https://windows.github.com/> on local computer.



Access it from desktop using shortcut saved on desktop. After every single changes we can synchronize it and push pull would be done automatically.



**Key vocabulary & concepts**

**Local Repository:** Every VCS tool provides a private workplace as a working copy. Developers make changes in their private workplace and after commit, these changes become a part of the repository. Git takes it one step further by providing them a private copy of the whole repository. Users can perform many operations with this repository such as add file, remove file, rename file, move file, commit changes, and many more.

**Commit:** Commit holds the current state of the repository. A commit is also named by SHA1 hash. You can consider a commit object as a node of the linked list. Every commit object has a pointer to the parent commit object. From a given commit, you can traverse back by looking at the parent pointer to view the history of the commit. If a commit has multiple parent commits, then that particular commit has been created by merging two branches.

**Branches:**Branches are used to create another line of development. By default, Git has a master branch, which is same as trunk in Subversion. Usually, a branch is created to work on a new feature. Once the feature is completed, it is merged back with the master branch and we delete the branch. Every branch is referenced by HEAD, which points to the latest commit in the branch. Whenever you make a commit, HEAD is updated with the latest commit.

**Touring Content Versioning on GitHub**

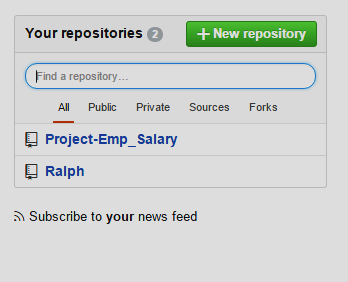
**Browse to one of your repositories:**

We can create repository and browse that in many ways.

We need to browse our repository on github to see the changes done by other members, to push our changes from local git repository. If we are working on a project in a team it is very important to be updated in order to collaborate properly and to make sure every member in team going in a right direction. Every member in team can get the whole copy of remote repository from GitHUb in his local computer. The admin can make the remote repository public or private on github according to the requirements. Public Repositories can be accessed by anyone while private repositories can’t be accessed without owner’s permission.

We can keep as many repositories as we need in GitHub Account and browsing to a particular one is quite simple task.

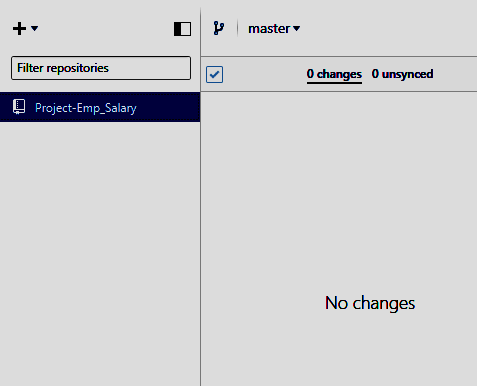
1. Browsing Repository from GitHub Account on web browser

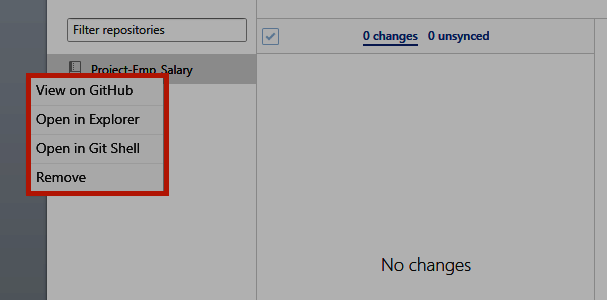


On GitHub account when we login we can select Repository by clicking on name or by typing the name in the blank space and we can filter them by clicking on public, private , sources etc.

1. Browsing Repository from GitHub Desktop App:

GitHub Desktop app provide multiple platforms to browse repository from one place by synchronized it to local git, git hub account. We can open repository in git shell, file explorer, GitHub directly just by making right click on repo. There is no need to push or pull changes manually, using command prompt. This app makes easy to perform all git commands and to browse repository, branches, history etc.

****

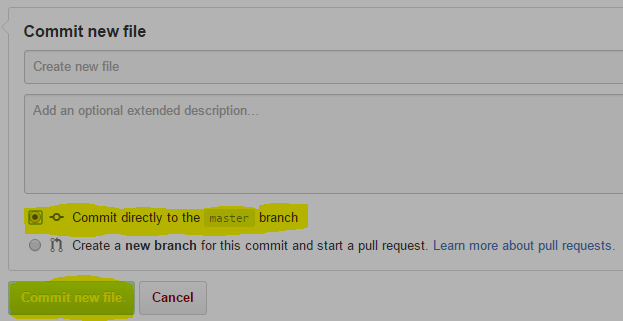
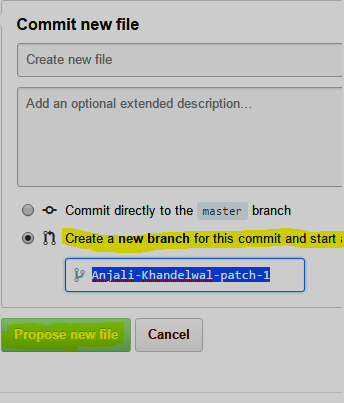


**• Create file(s) and commit the new content :**

We can create files from GitHub web interface directly if we have access to the particular repo as a contributor or an owner.



* On In your repository, browse to the folder where you want to create a file.
* Above the file list, click+.
* In the file name field, type the name and extension for the file. To create subdirectories, type the / directory separator.
* On the **Edit new file** tab, add content to the file.
* To review the new content, click **Preview**.
* At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file.
* Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is master, you should choose to create a new branch for your commit and then create a pull request.
* Click **Propose new file.**

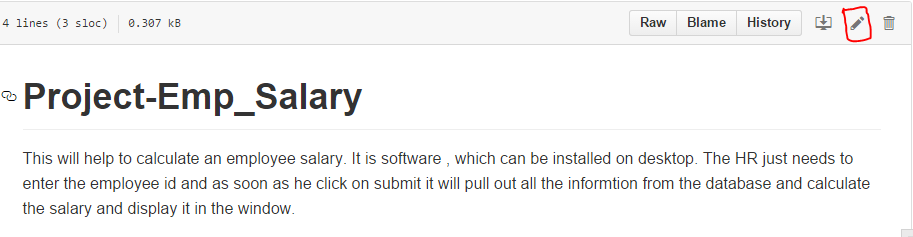


## Editing files in your repository

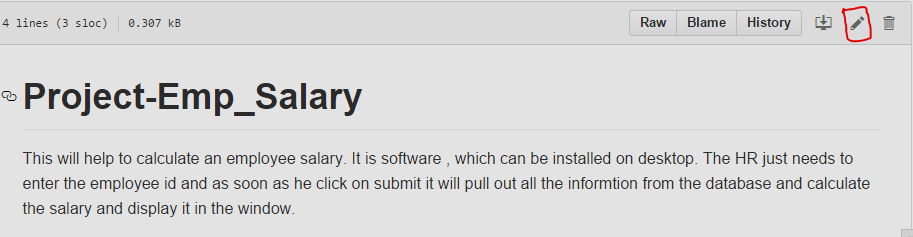
• Edit an existing file's content and commit the change:

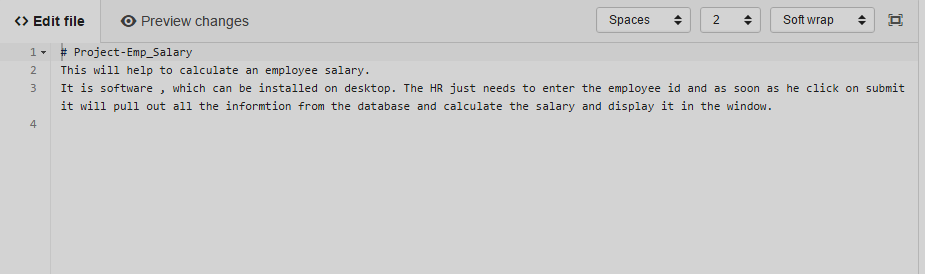
You can edit files directly on GitHub in any of your repositories.

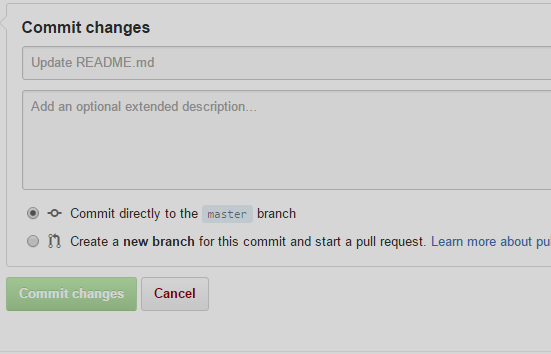
* In your repository, browse to the file you want to edit.



* In the upper right corner of the file view, click on edit icon to open the file editor.
* On the **Edit file** tab, make any changes you need to the file.
* Above the new content, click **Preview changes**





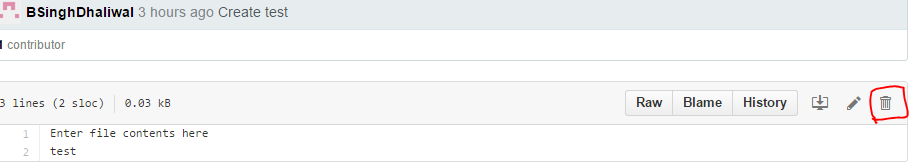


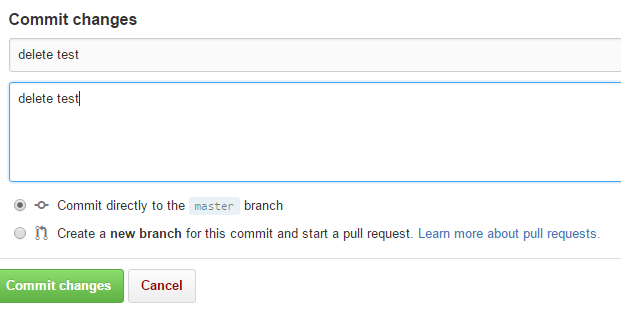
At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file.

Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is “master”, you should choose to create a new branch for your commit and then create a pull request.

• Remove a file and commit the removal

Browse the file you want to remove the click on removal icon on right side corner then type in the commit message in the bottom. Choose the default master branch or if you want to make another branch then pull request as well. And click on commit changes to complete the action.





• Change a filename and commit the path change

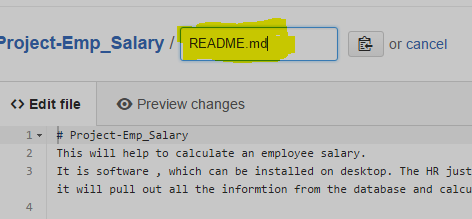
You can rename any file in your repositories directly in GitHub. Renaming a file also gives us the opportunity to move the file to a new location.

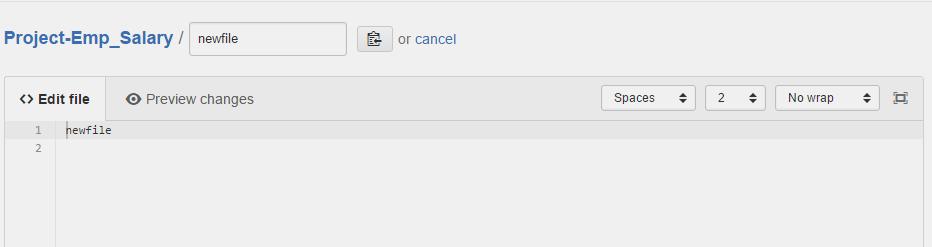
Browse the file which name we want to change. Click on edit icon and then in edit tab we can directly change the name of file. From Here e\we can also change the path of the file as well. The guidelines are listed below to change the path of any file.

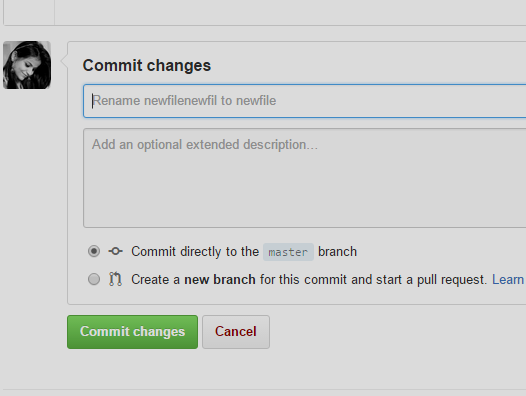
To move the file **into a subfolder**, type the name of the folder you want, followed by /. Your new folder name becomes a new item in the navigation breadcrumbs.

To move the file into a directory **above the file's current location**, place your cursor at the beginning of the filename field, then either type ../ to jump up one full directory level, or type the Backspace key to edit the parent folder's name.

After changing the name at the bottom we need to type in a meaningful commit message.







• Examine the Commits page of change history:

We can check the all history of commits and changes in files, folders and repositories with attached commit messages from the history tab in GitHub web interface.

