Git Tutorial

What is a version control system?

A version control system allows you to track the history of a collection of files and includes the functionality to revert the collection of files to another version.

Each version captures a snapshot of the files at a certain point in time.

The collection of files is usually *source code* for a programming language but a typical version control system can put any type of file under version control.

The collection of files and their complete history are stored in a *repository*.

What is a distributed version control system?

A distributed version control system does not necessarily have a central server which stores the data.

The user can copy an existing *repository*.

This copying process is typically called *cloning* in a distributed version control system and the resulting repository can be referred to as a *clone*.

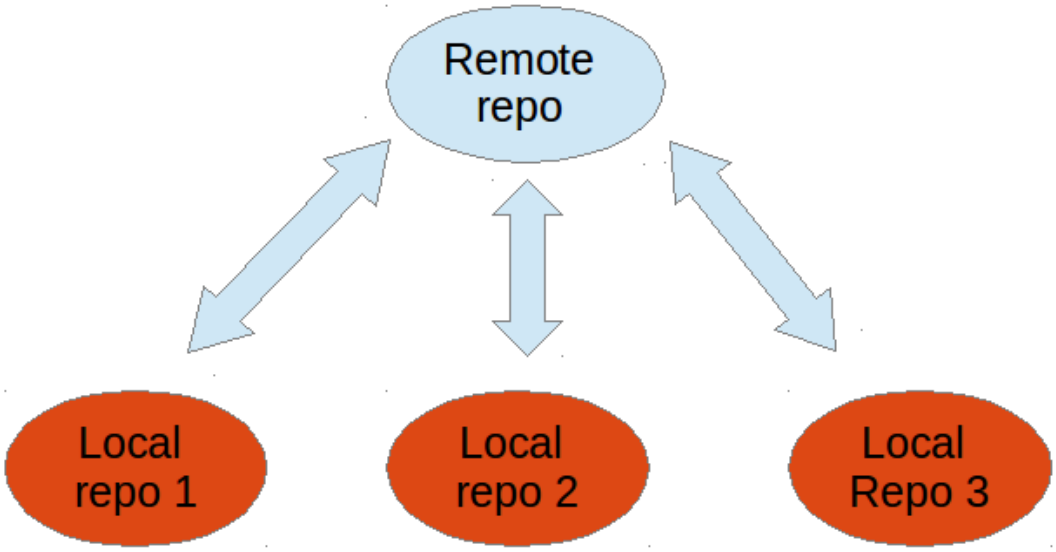
Typically there is a central server for keeping a repository but each cloned repository is a full copy of this repository.

The decision which of the copies is considered to be the central server repository is pure convention and not tied to the capabilities of the distributed version control system itself.

Every *clone* contains the full history of the collection of files and a cloned repository has the same functionality as the original repository.

Every repository can exchange versions of the files with other repositories by transporting these changes.

This is typically done via a repository running on a server which is, unlike the local machine of a developer, always online.



What is Git?

Git is a distributed version control system.

Git originates from the Linux kernel development and is used by many popular open source projects, e.g., the Android or the Eclipse developer teams, as well as many commercial organizations.

The core of Git was originally written in the programming language *C*, but Git has also been re-implemented in other languages, e.g., Java, Ruby and Python.

Important terminology in Git

Local repository and operations

After cloning or creating a repository the user has a complete copy of the repository.

The user performs version control operations against this local repository, e.g., create new versions, revert changes, etc.

If you want to delete a Git repository, you can simply delete the folder which contains the repository.

Remote repositories

Git allows the user to synchronize the local repository with other (remote) repositories.

Users with sufficient authorization can *push* changes from their local repository to remote repositories.

They can also *fetch* or *pull* changes from other repositories to their local Git repository.

What are branches in Git?

Git supports *branching* which means that you can work on different versions of your collection of files.

A branch separates these different versions and allows the user to switch between these versions to work on them.

For example, if you want to develop a new feature, you can create a branch and make the changes in this branch without affecting the state of your files in another branch.

Branches in Git are local to the repository.

A branch created in a local repository, which was cloned from another repository, does not need to have a counterpart in the remote repository.

Local branches can be compared with other local branches and with *remote-tracking branches*.

A remote-tracking branch proxies the state of a branch in another remote repository.

Git supports the combination of changes from different branches.

This allows the developer, for example, to work independently on a branch called *production* for bugfixes and another branch called *feature_123* for implementing a new feature.

Steps in Working with GitHub

Step1:

Create an Account in GitHub

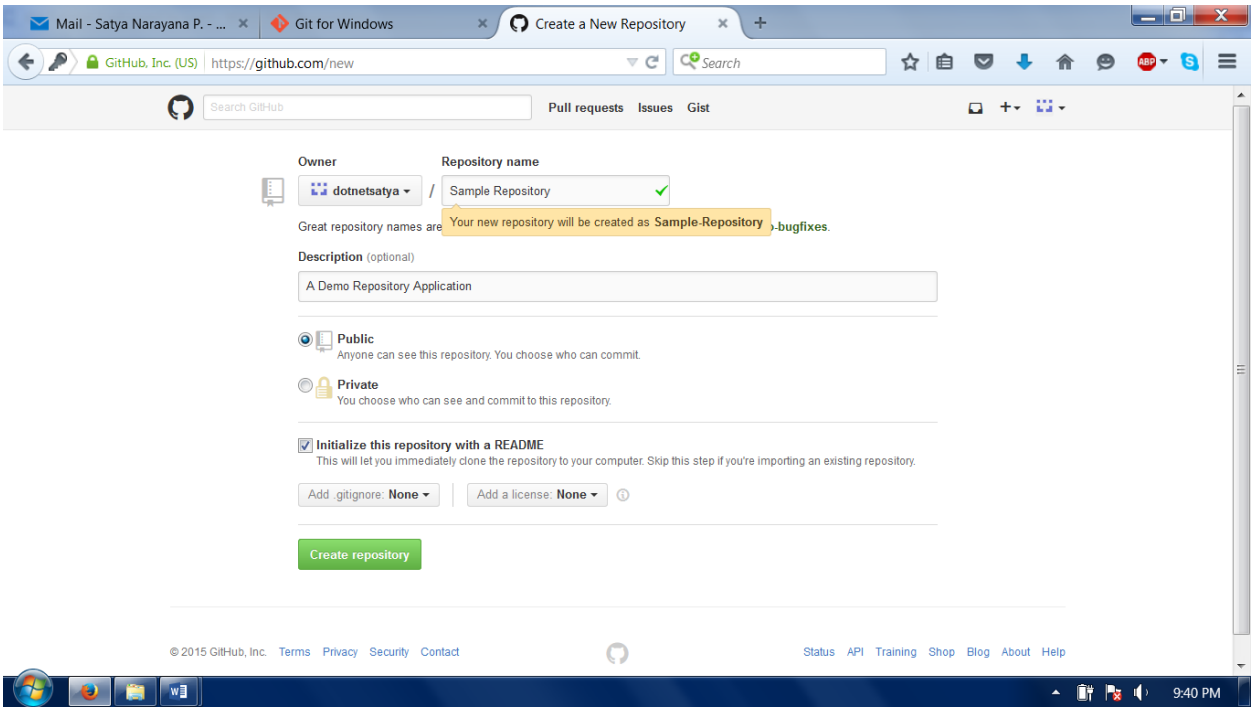
https://github.com

Step2:

Sign In to the GitHub

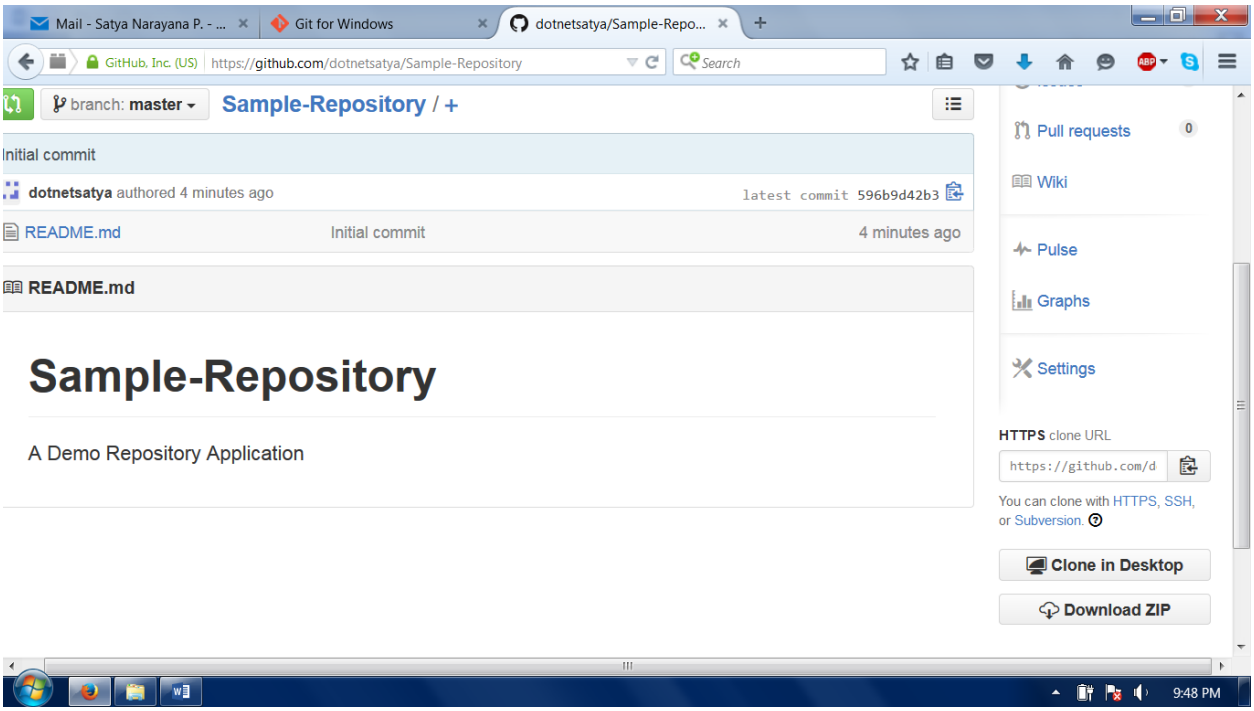
Create a New Repository or fork an Existing Repository

If you are an admin of a repository, then you can add users to the repository and allows permissions to do changes in the repository.

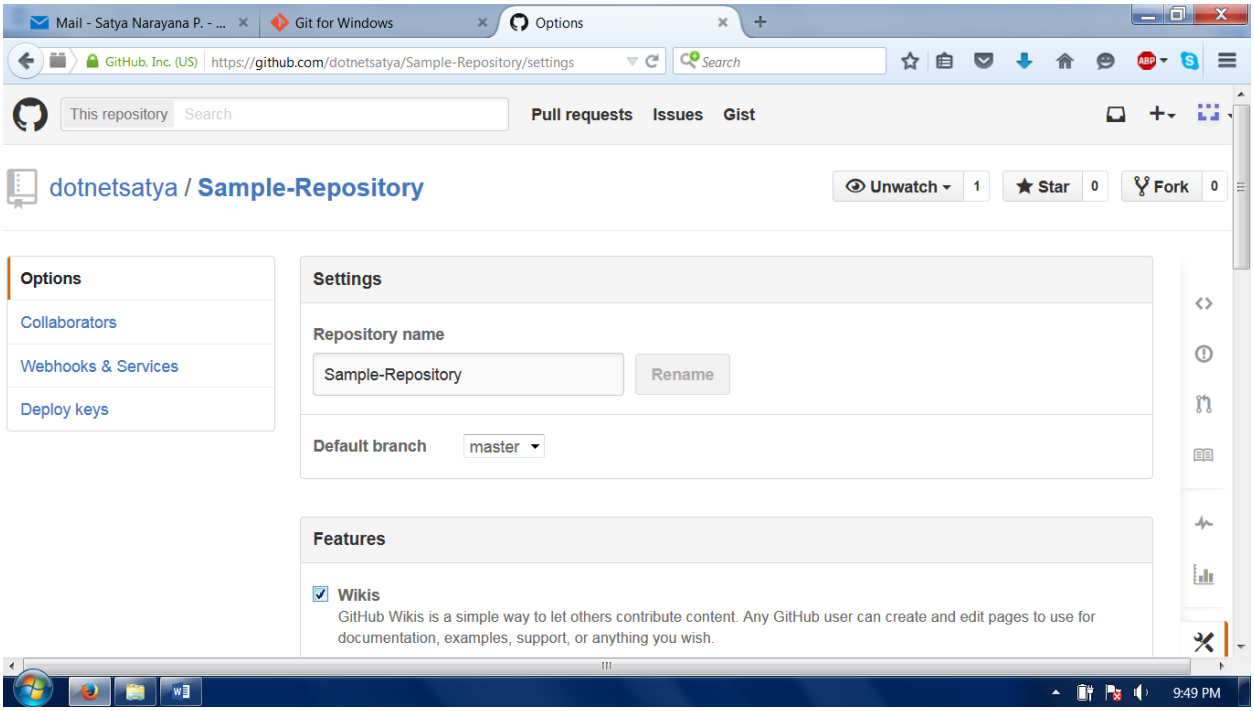


Once the Repository is created:

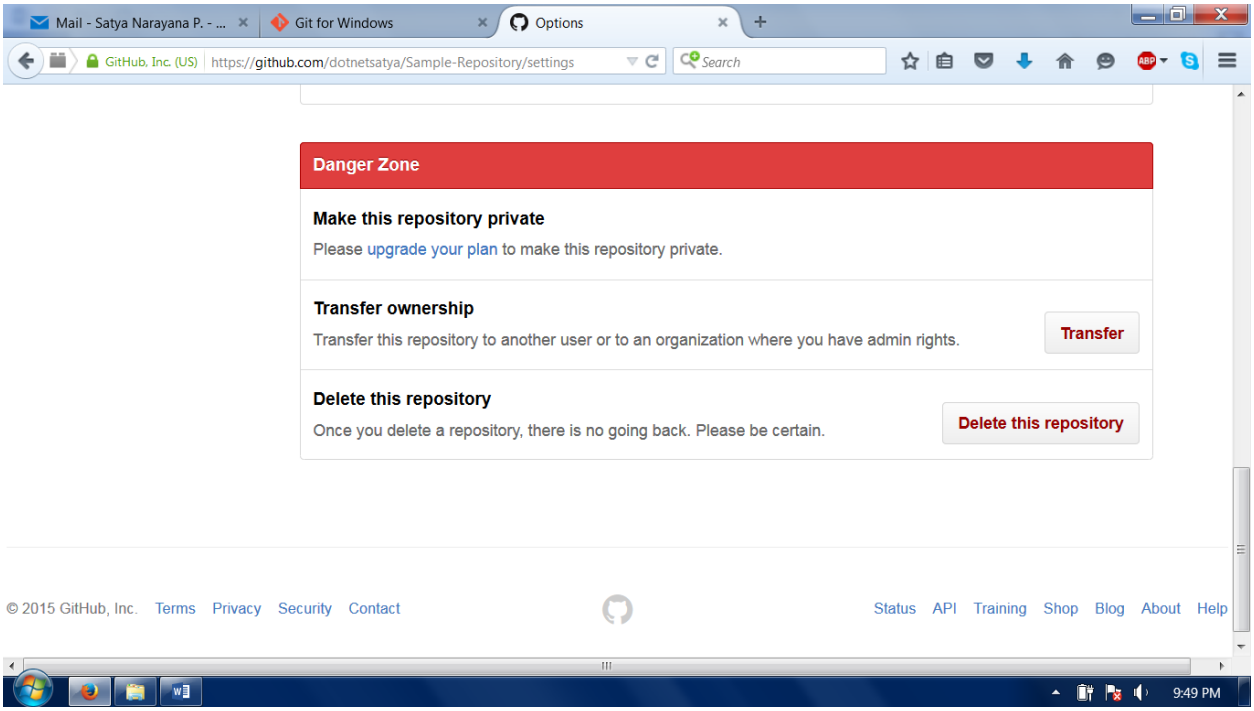
- You can close using URL.
- You can Clone in Desktop
- You can download as Zip**.**



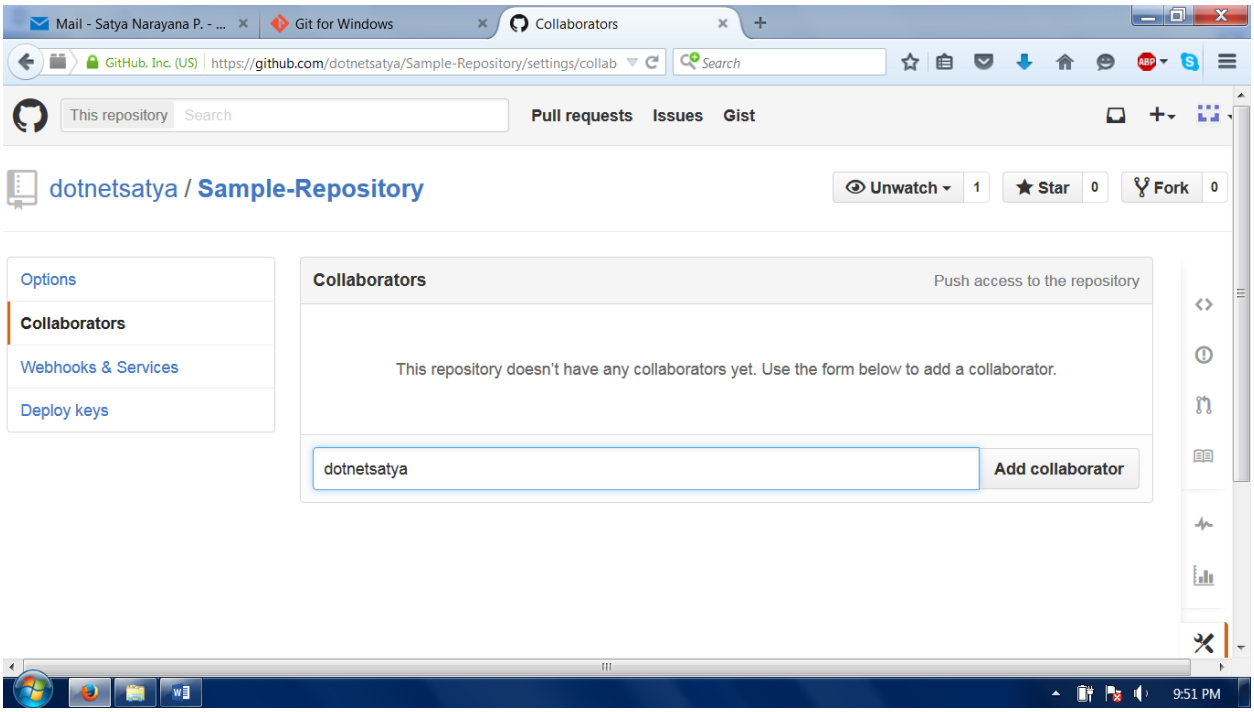
You can also edit the settings



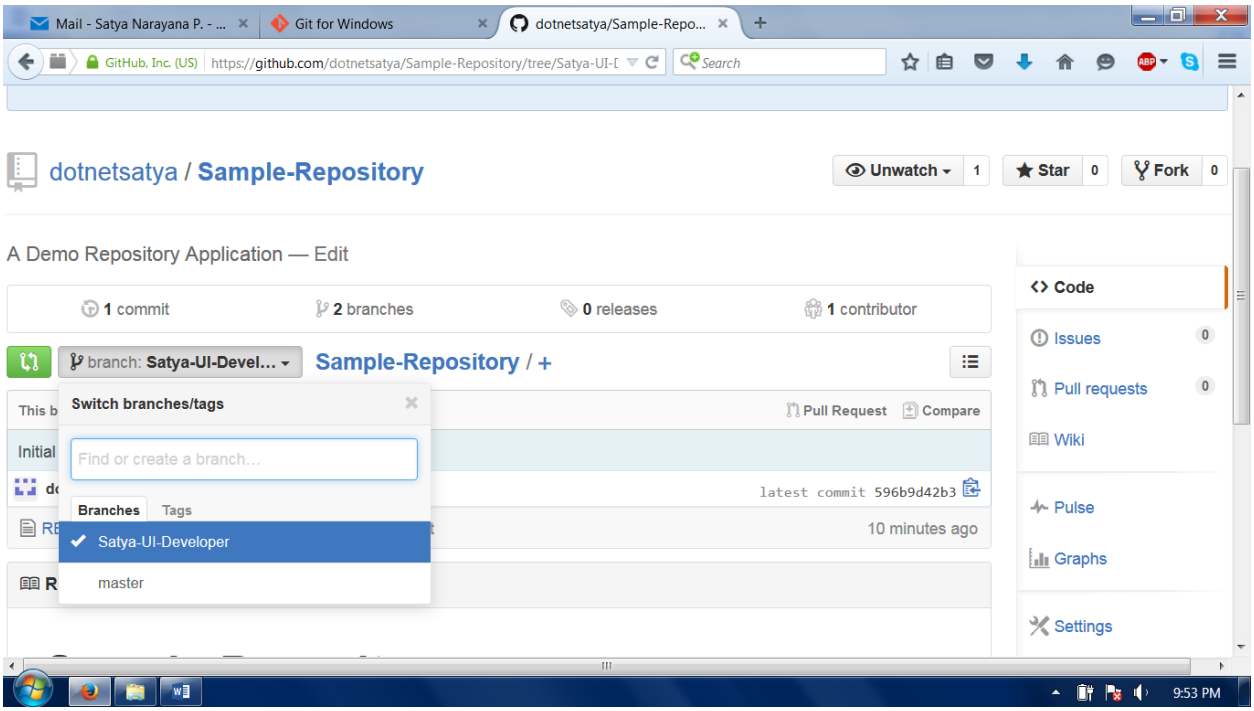
You can make the repository as Private or transfer Ownership or delete the repository.



You can add Collaborators to the repository



You can create Branches in the Repository:



You can choose the Branch, so that we can clone a specific branch.

Working Github in WebStorm

You can download and enable Git in WebStorm

WebStorm

WebStorm is an Integrated Development Environment for the Web developers and coders, built on top of IntelliJ IDEA platform.

WebStorm inherits all the web-related functionality of IntelliJ IDEA for editing HTML, CSS, JavaScript, XML, working with VCS, and the other tools specific to web development.

WebStorm suggests the following advanced features:

- Intelligent Editor: for HTML, CSS, JavaScript, XML, which includes syntax highlighting, documentation lookup, and refactoring's.
- Error-Free Coding: on-the-fly code analysis, error highlighting and quick fixes.
- Project and Code Navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages.
- VCS Integrations: out-of-the-box support for Subversion, Perforce, Git, and CVS with change lists and merge.

JetBrains WebStorm is cross-platform. It works on Windows, OS X and Linux and brings the whole range of precise developers tools, all tied together to create the convenient development environment.

Version Control with WebStorm

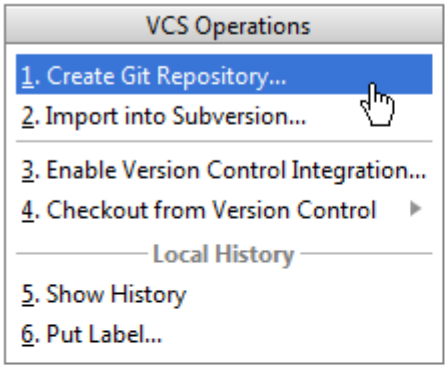
Accessing VCS Operations

To quickly invoke a VCS command using VCS Operations Pop-up

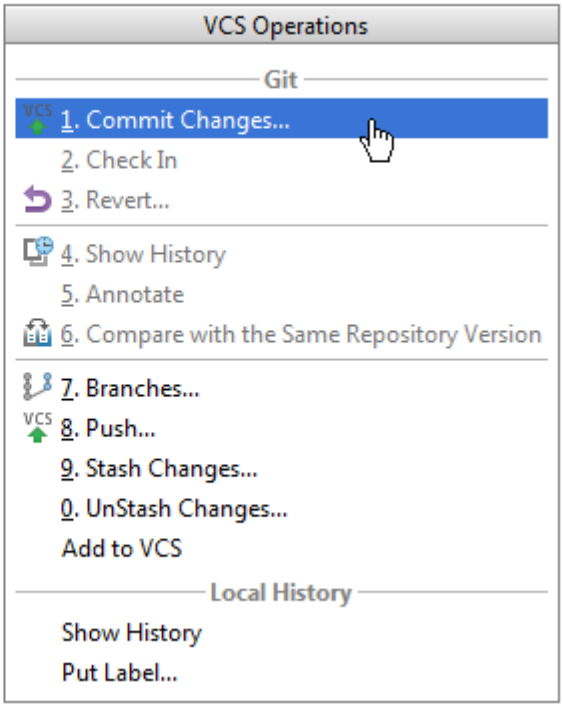
1. Open VCS Operations pop-up, in one of the following ways:
    - On the main menu, choose VCS | VCS Operations Pop-up.

Note that the composition of VCS commands available on the pop-up menu, depends on the specific VCS

No VCS selected



VCS enabled



Choose command from the VCS Operations list. To do that, perform one of the following actions:

- Click the desired command in the list.
- Use up and down arrow keys to select the desired command, and then press `Enter`
- Press number key that corresponds to the desired command in the list.

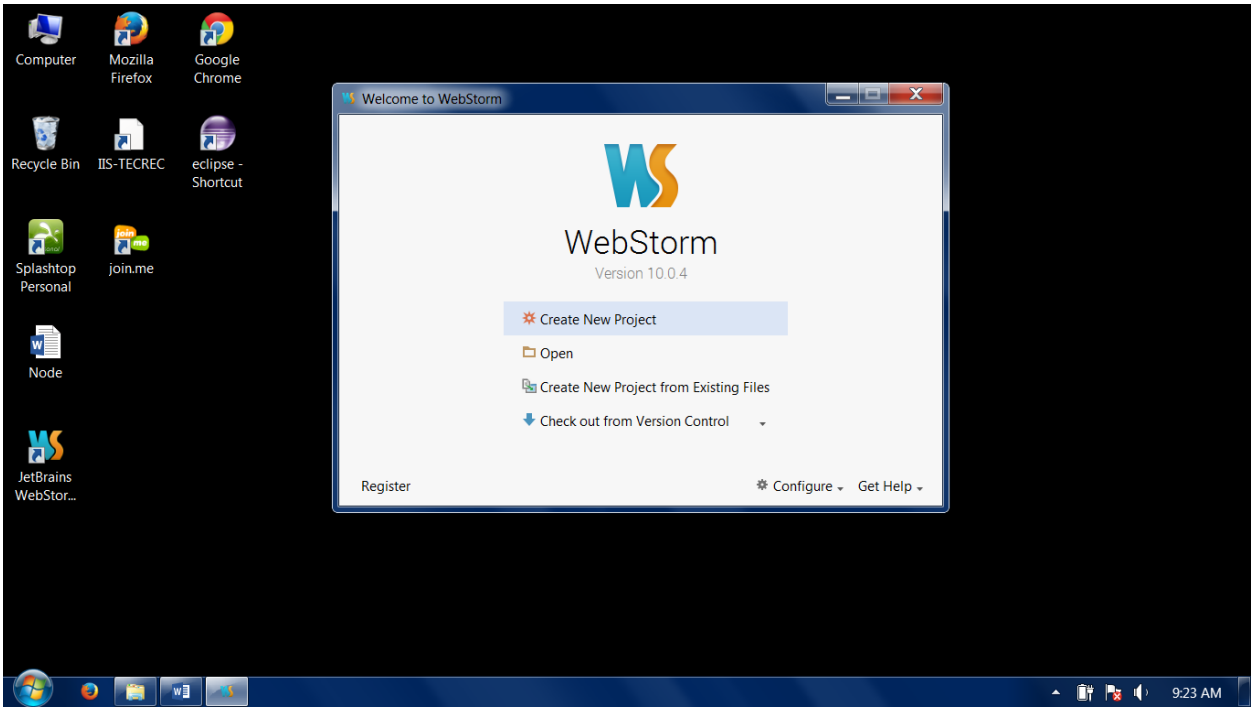
Cloning a Repository from GitHub

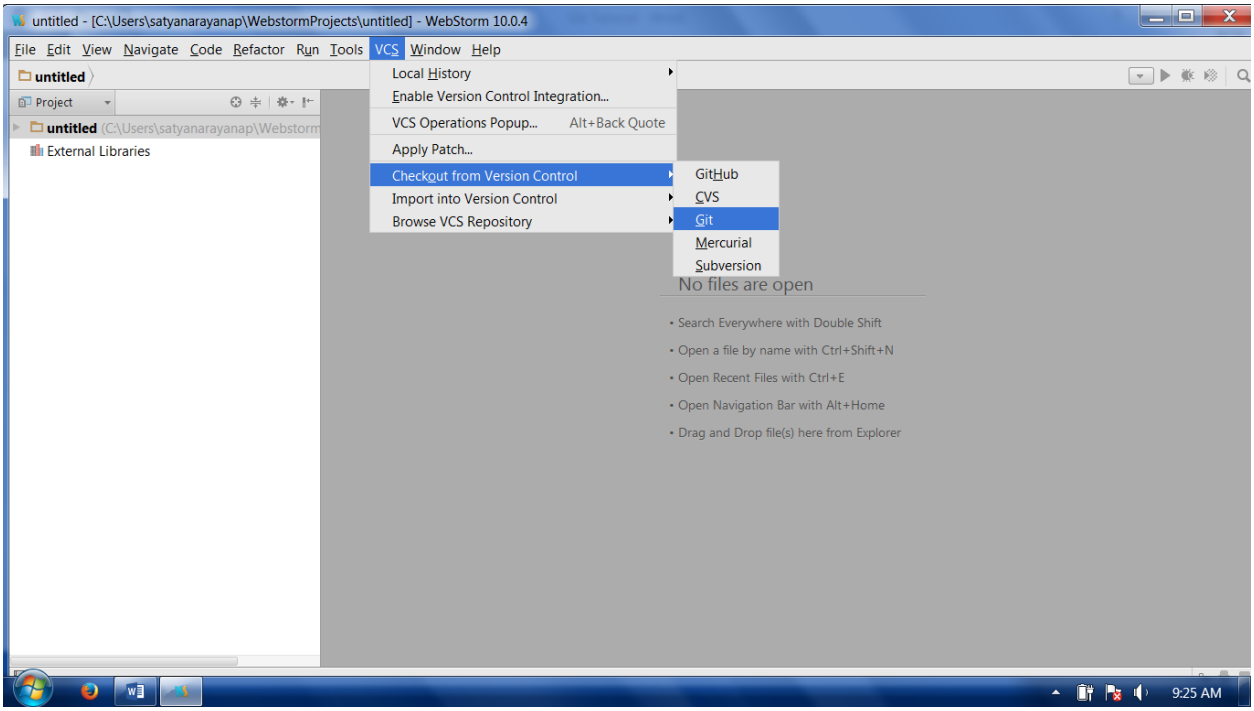
To clone a repository from the github storage

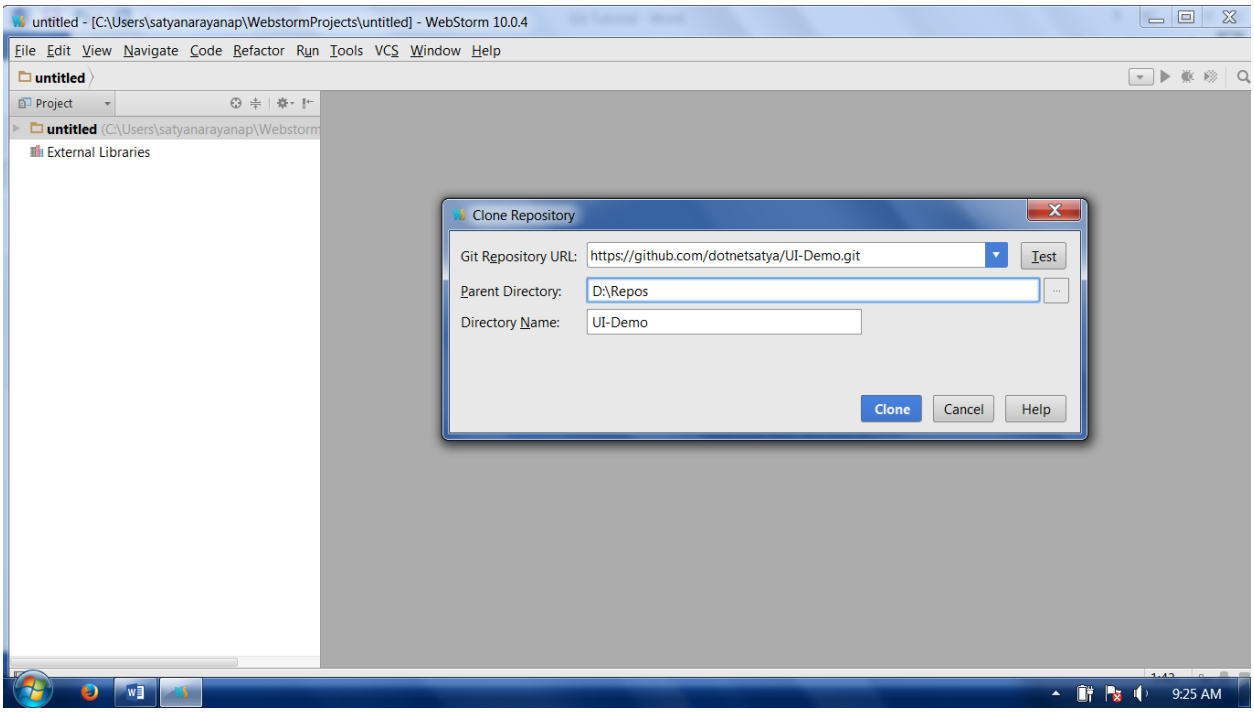
1. Choose VCS | Checkout from Version Control | GitHub on the main menu. WebStorm establishes connection with GitHub using the login and password you registered. Upon establishing connection, the Select Git Hub Repository to Clone dialog box opens.
2. From the Repository drop-down list, select the source repository to clone the data from.
3. To view the details and description of the selected repository in the browser, click the Details on <repository name> here link.
4. In the Folder text box, specify the directory where the local repository for cloned sources will be set up. Type the path to the directory manually or click the Browse button and choose the desired directory in the Select project destination folder dialog box that opens.
5. In the Project name text box, specify the name of the project to be created based on the cloned sources.
6. Click the Clone button to start cloning the sources from the specified remote repository.

Or

You can clone/checkout a repository directly while creating a new Project in WebStorm.


Computer
Mozilla Firefox
Google Chrome
Recycle Bin
IIS-TECREC
eclipse - Shortcut
Splashtop Personal
join.me
Node
JetBrains WebStor...
Welcome to WebStorm
WebStorm
Version 10.0.4
Create New Project
Open
Create New Project from Existing Files
Check out from Version Control
Register
Configure
Get Help
9:23 AM

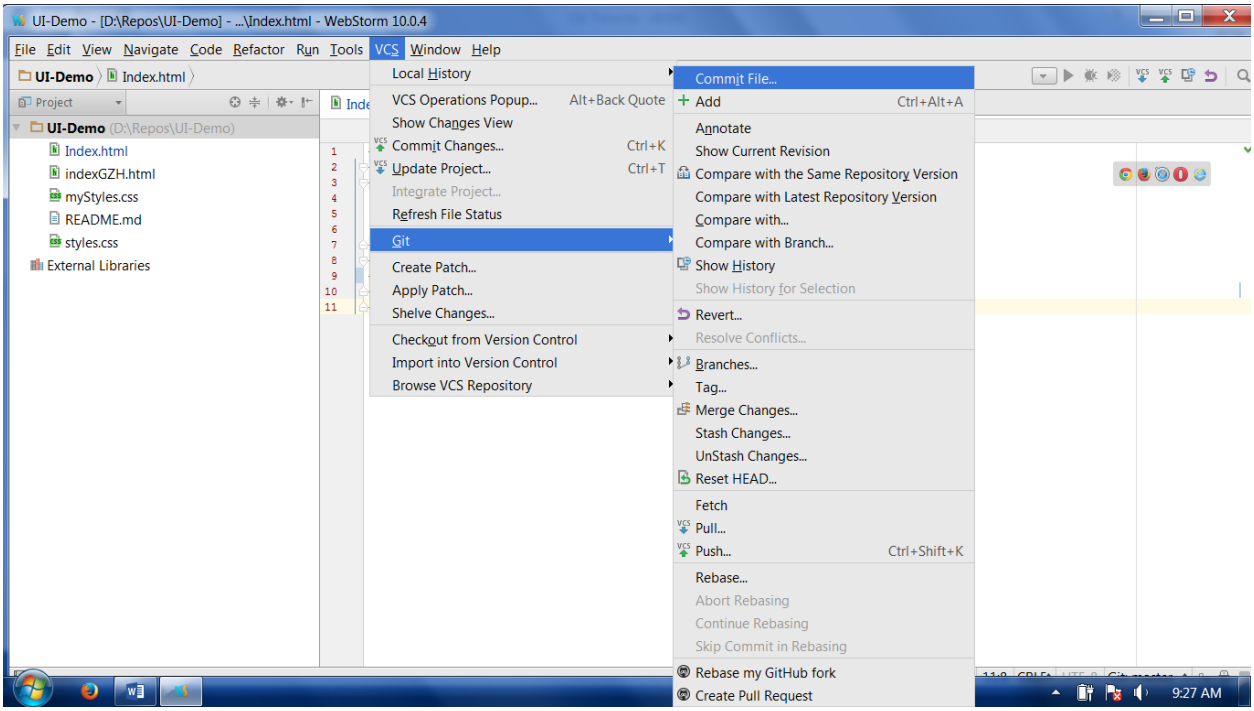

untitled - [C:\Users\satyanarayanap\WebstormProjects\untitled] - WebStorm 10.0.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
untitled
Project
untitled (C:\Users\satyanarayanap\Webstorm
External Libraries
Local History
Enable Version Control Integration...
VCS Operations Popup... Alt+Back Quote
Apply Patch...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
GitHub
CVS
Git
Mercurial
Subversion
No files are open
• Search Everywhere with Double Shift
• Open a file by name with Ctrl+Shift+N
• Open Recent Files with Ctrl+E
• Open Navigation Bar with Alt+Home
• Drag and Drop file(s) here from Explorer
9:25 AM



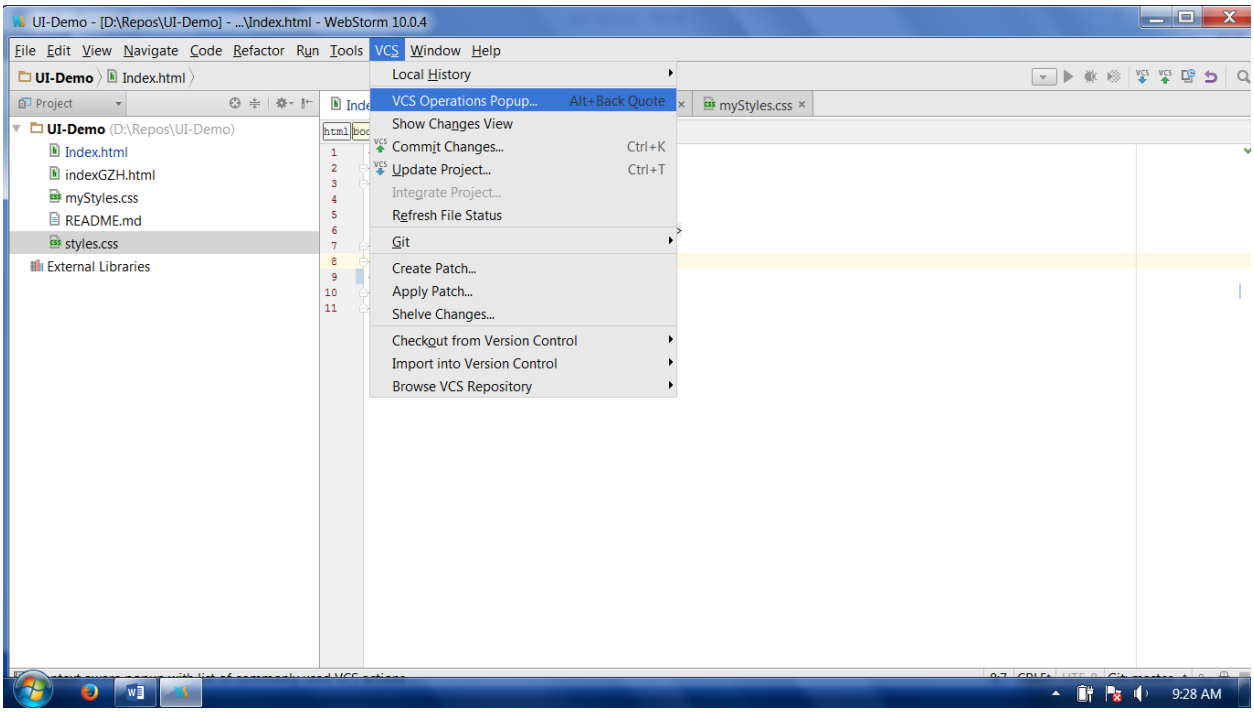
Once the cloning is done, you get a copy of repository in the local system.

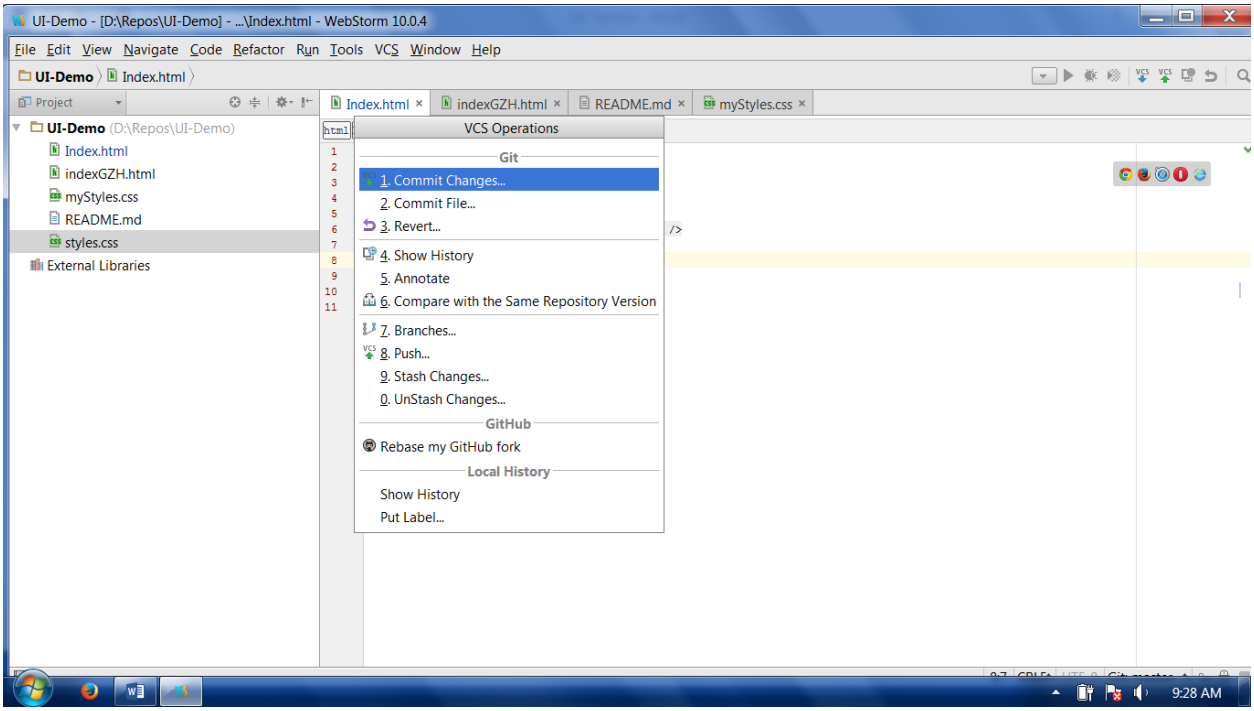
You can add files, edit changes in the existing files.

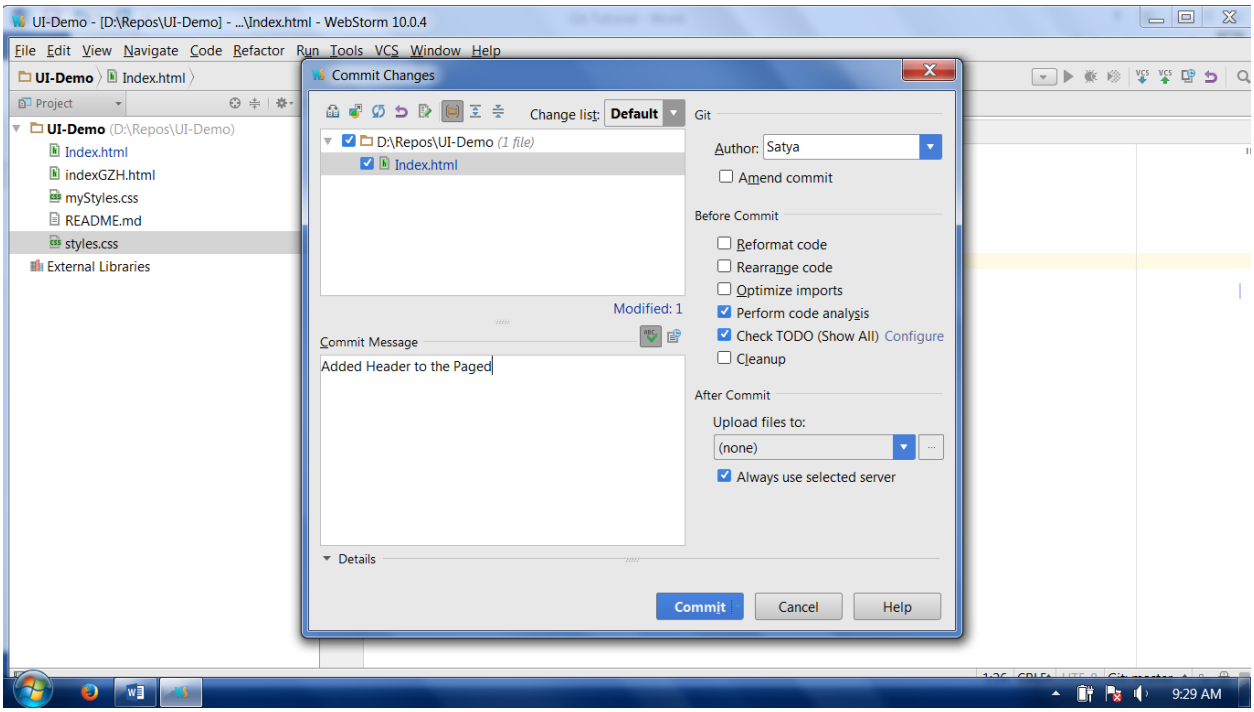
Then, you can commit the changes.



Or You can choose VCS Operations PopUp

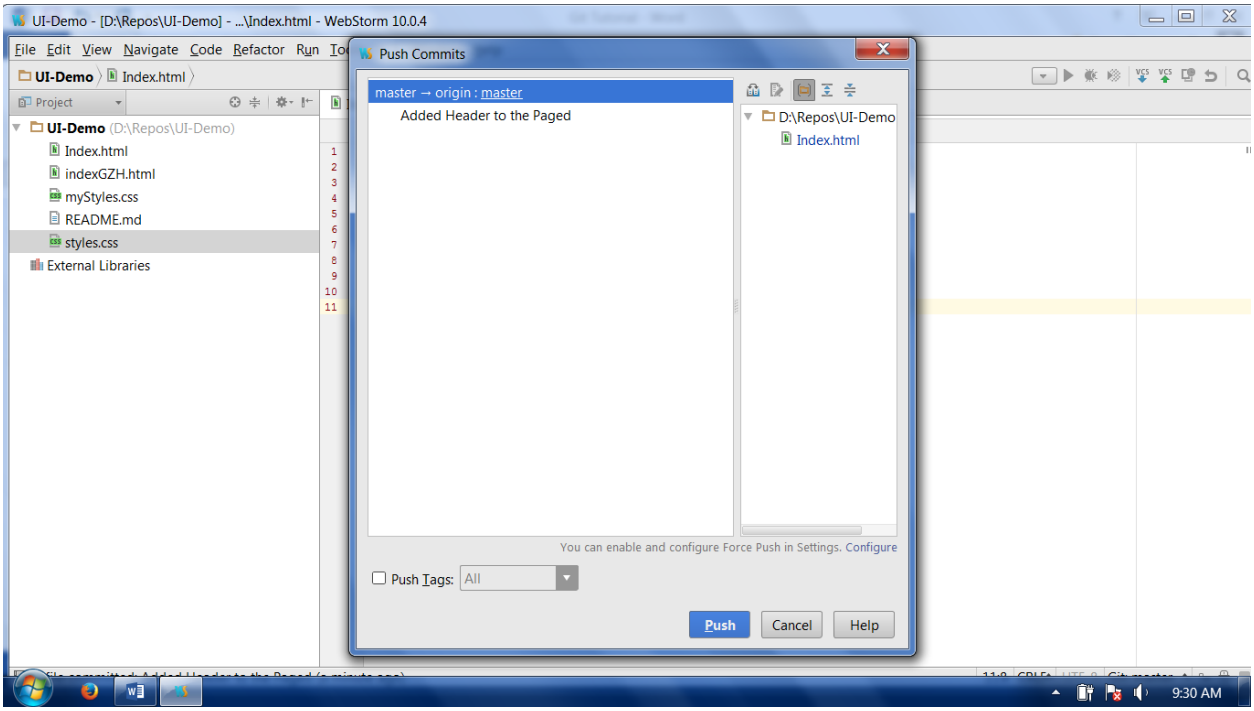






You can give some comment to the changes and click on commit.

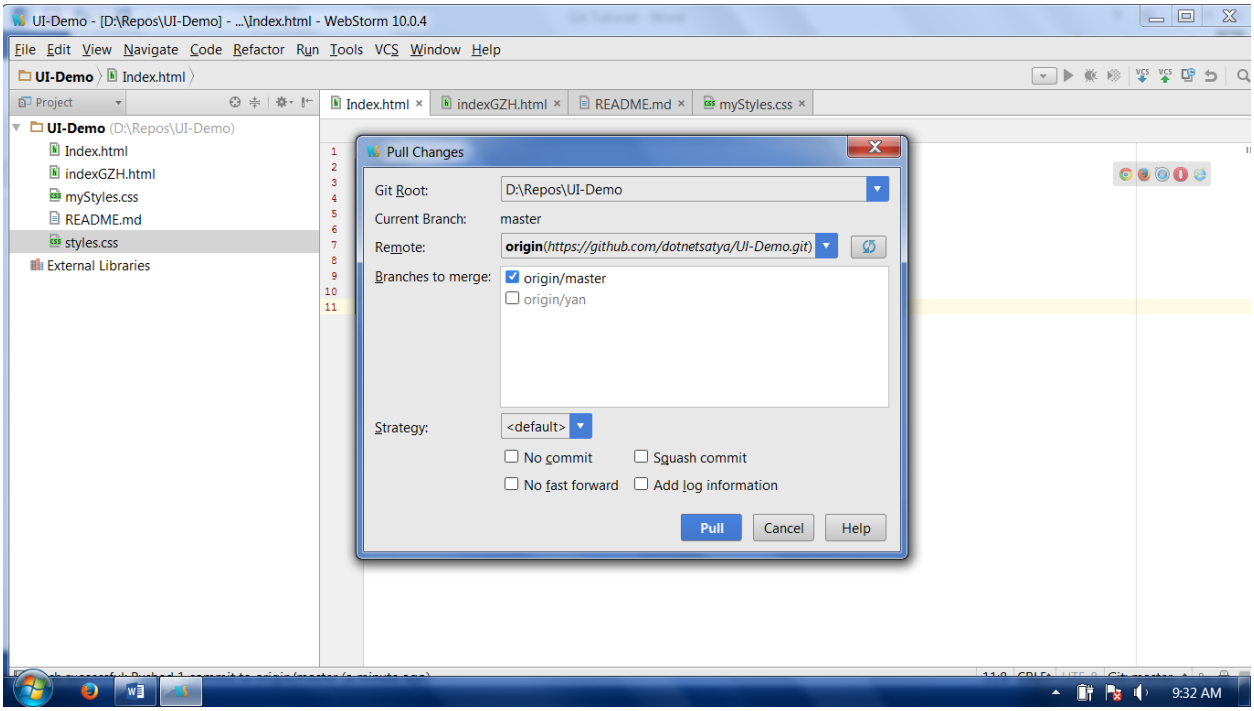
Now, you can Full or Fush the changes to the GitHub main repository.



Now, you can click on Push button.

It will ask you to enter the Master password. You can give the master password and push to the git repository.

And You can also pull the updated copy from the Git repository.



The same way, you can compare branches.

You can merge branches.

You can view history.