**What is Git?**

Version control is a set of programs that manage changes to computer files, such as documents, images and source code.

* Distributed revision control system.
* Greatly simplifies concurrent work, merging changes.
* More efficient, better workflow, etc.
* Compare Changes over time.

**What is GitHub?**

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

GitHub essentials like repositories, branches, commits, and Pull.

* Create and use a repository
* Start and manage a new branch
* Make changes to a file and push them to GitHub as Requests commits
* Open and merge a pull request

**Summary of the project**

* Project Title : Product Management
* Front End : PHP
* Back End : MYSQL
* Description
  + In this project user can add Product, edit product and delete Product to management.
  + Admin can add the property details of product.

**Install Git:**

**Step 1. Install Git for Windows**



Figure 1 Install Git for Window

## Step 2 . Install the NetBean IDE to Git.

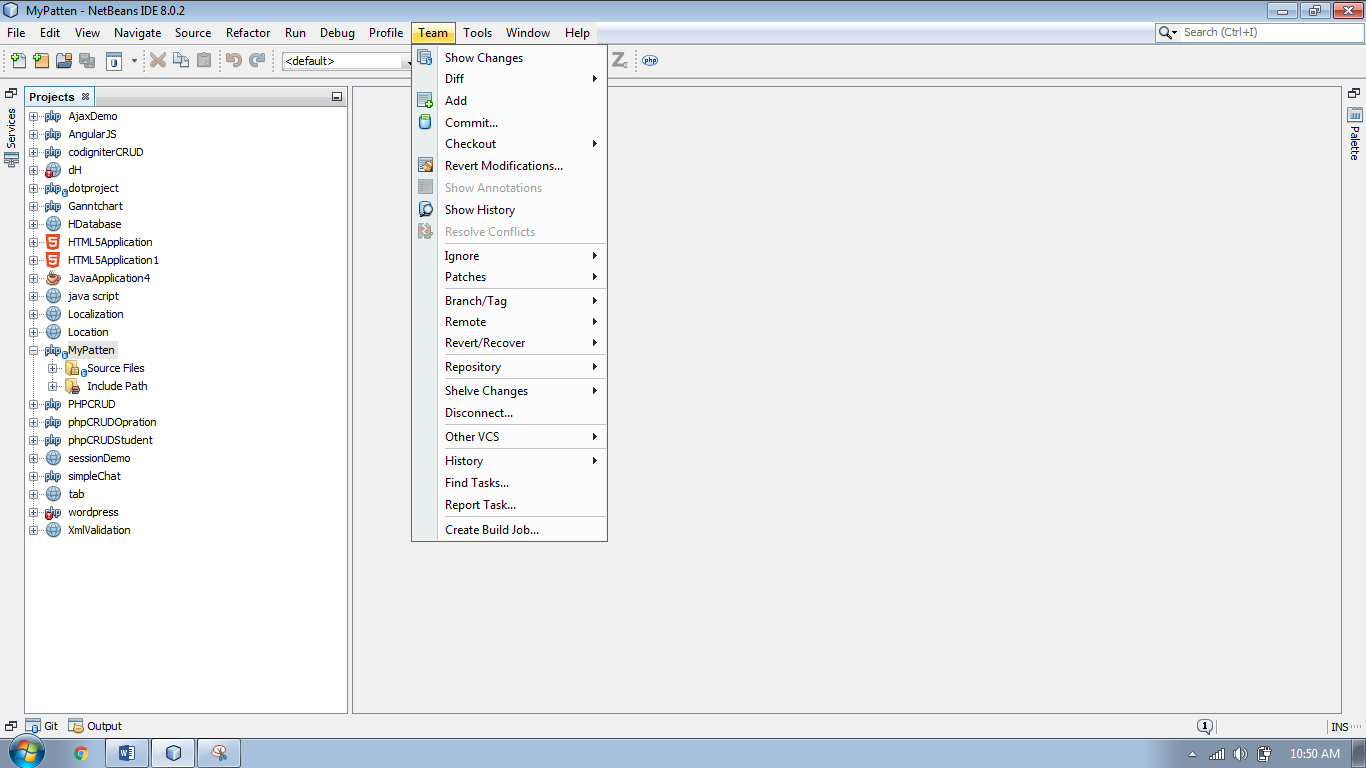


Figure 2 Install the Net Beans of the Git.

Click the “Download” button and let it install. You’ll need to restart Net Beans IDE once it’s installed.

Restart Visual Studio and go to TOOLS > OPTIONS… in Net Beans IDE and navigate to the “Source Control” section. Within this section you’ll see an option to select the current source control plug-in.

**Step 3. Configuration Username and Password with Git.**

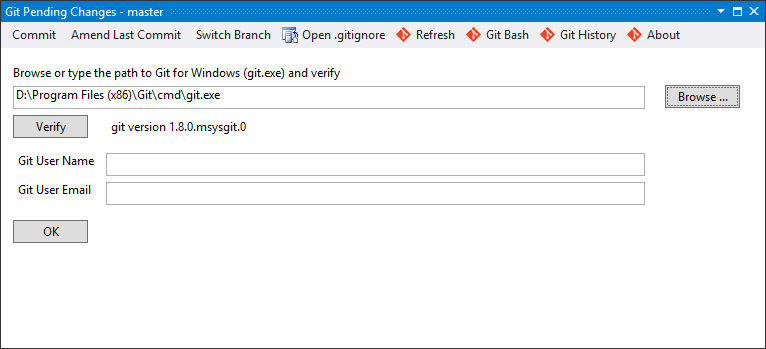
[](https://berniecook.files.wordpress.com/2013/01/gitpendingchanges.jpg)

Figure 4 Configuration Username and Password with Git.

Create a new Net Beans IDE, I’ve chosen a web application. At this point there will be no source control management so you need to add the solution to Git by right-clicking the solution file in the toolbox. Halfway down you’ll see the “Create Git Repository” option – select it.

## Step 5. Create a new Repository in Git

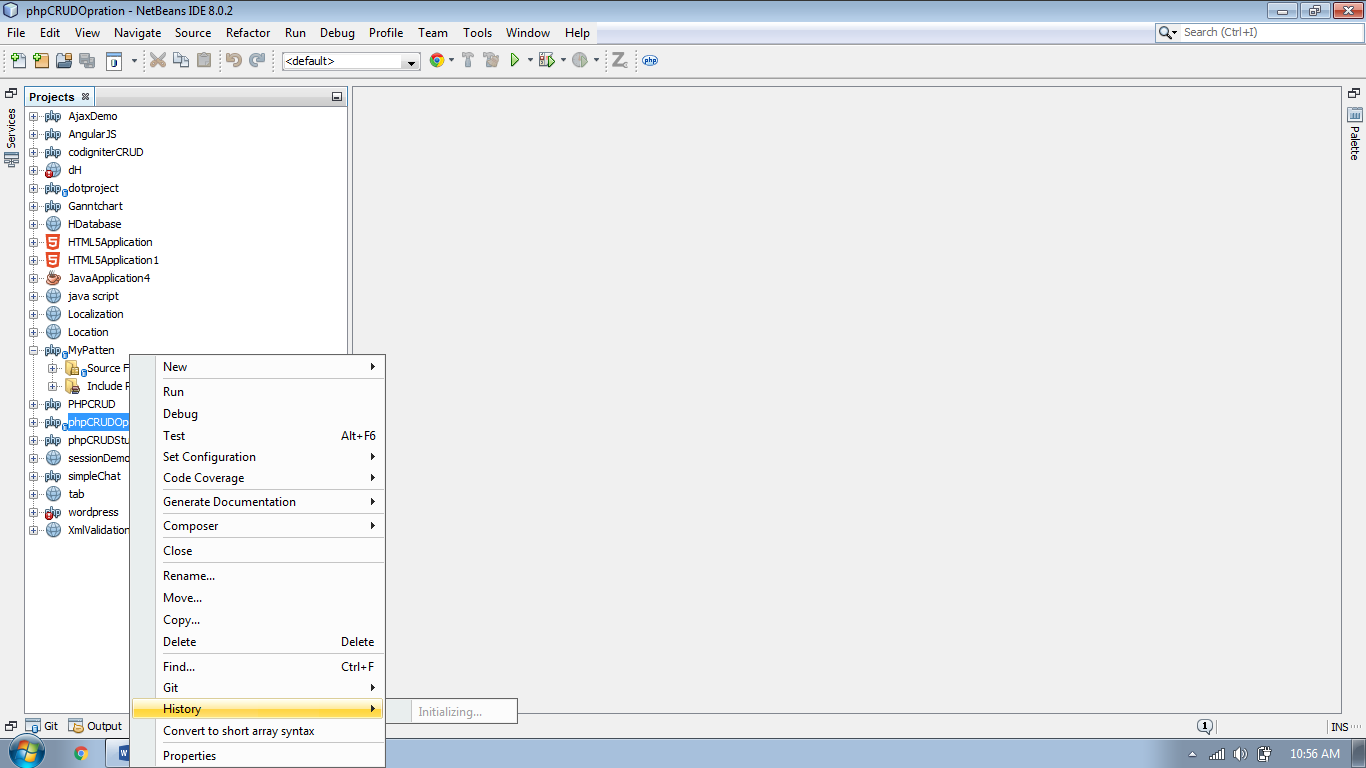


Figure 5 create a new Repository.

**Advantage of git:**

* Branching and merging is very easy to accomplish within the Tool Option.
* Tool option to commit changes to a source code repository.
* First, you select the Changes Button in the Git. Once you provide a commit message, you can commit these changes to your local repository. There are three options:

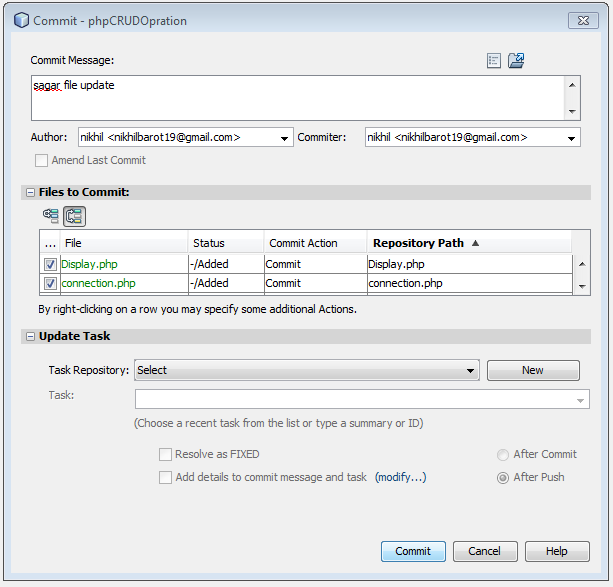


Figure 6 Commit file from Net Beans IDE .

**Initialization of Repository**

1. GitHub

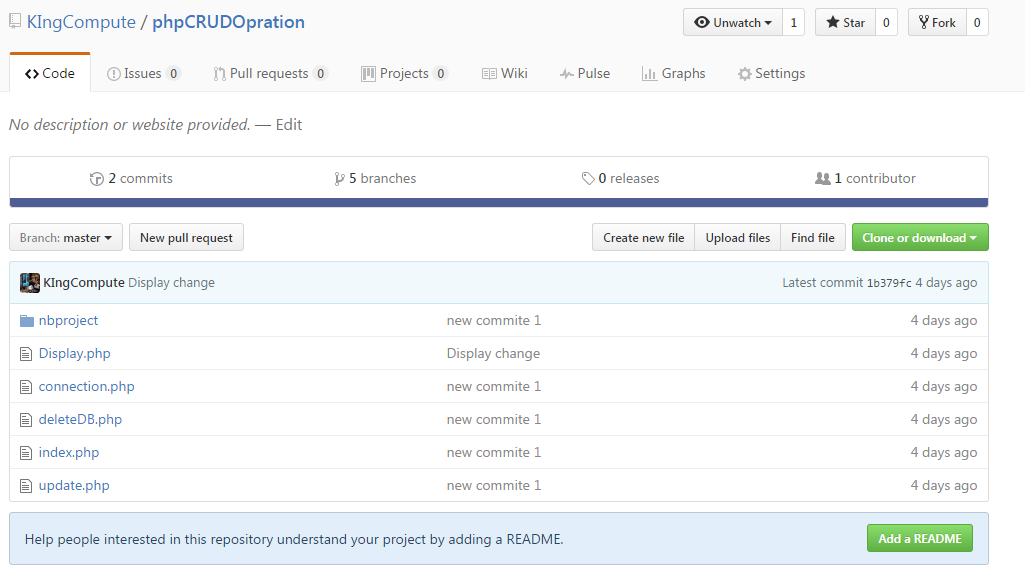


Figure Create new repository in GitHub

1. GitBase

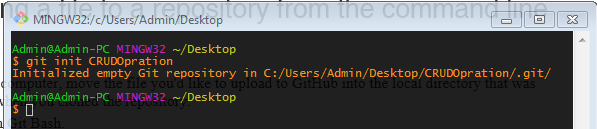
****

Figure Create repository in GitHub

**Cloning Repository**

1. Command Line (GitBase)

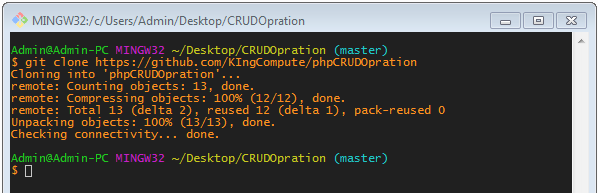
****

Figure Clone repository using command line

1. Git Hub

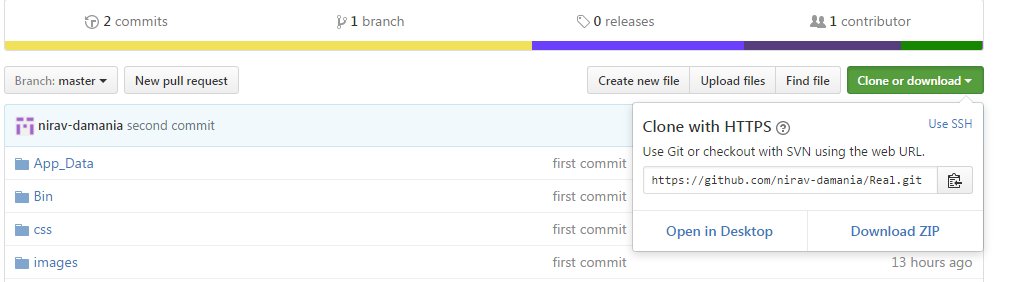
****

Figure Clone repository using GitHub

**Stage file in GitBash (Master Branch)**

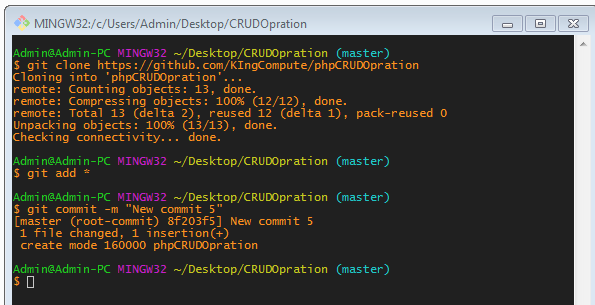
****

Figure Stage all files

**Commit file in GitBash (Master Branch)**

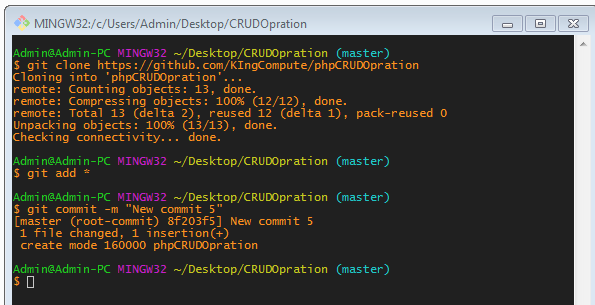
****

Figure Commit all file

**Create a new Branch**

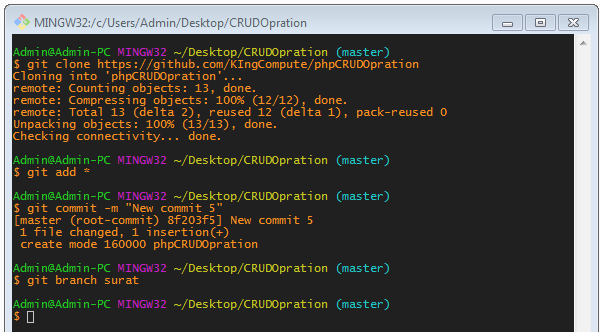


Figure create a new branch (my branch)

**Show all Branch**

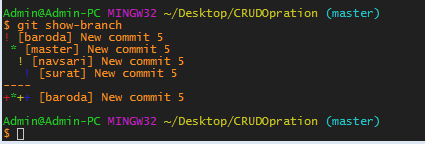
****

Figure Show all branch in repository

**Now add a new module in project. (Add property details)**

**Check status**

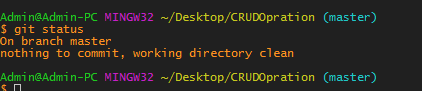
****

Figure check the status

**Checkout to new Branch (Surat)**

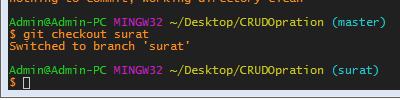
****

Figure Check out to new branch

**Add and commit file in new branch**

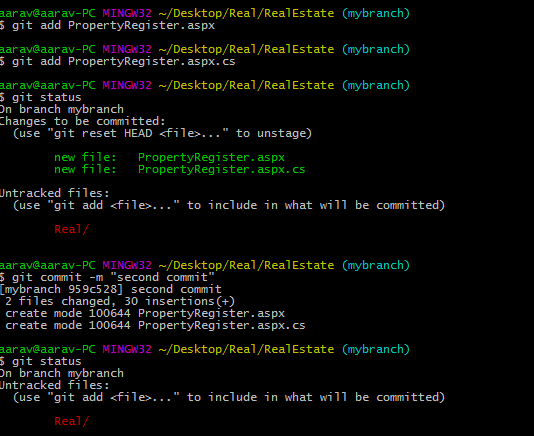
****

Figure Add and commit file

**Checkout to master branch**

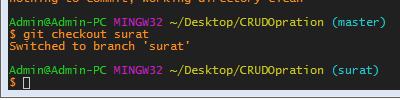
****

Figure Check out to master branch

**Merge new branch with master branch**

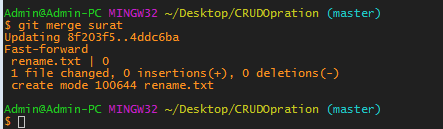
****

Figure Merge branch

**Difference between your working directory and the index**

Command: $ git diff

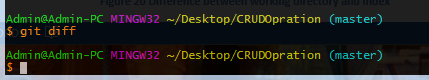
****

Figure Difference between working directory and index

**Difference between two commit file in same branch**

Command : $ git diff commit mybranch

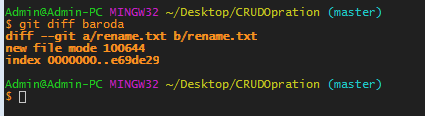
****

Figure Difference between two commit in branch

**Difference between committed file and stage file**

Command: $ git diff –-cached

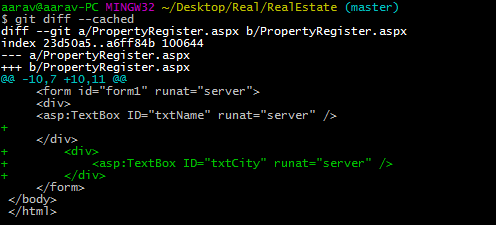
****

Figure Difference between commit file and stage file

**What is tree?**

Git stores content in a manner similar to a UNIX file system, but a bit simplified. All the content is stored as tree and blob objects, with trees corresponding to UNIX directory entries and blobs corresponding more or less to inodes or file contents. A single tree object contains one or more tree entries, each of which contains a SHA-1 pointer to a blob or subtree with its associated mode, type, and filename.

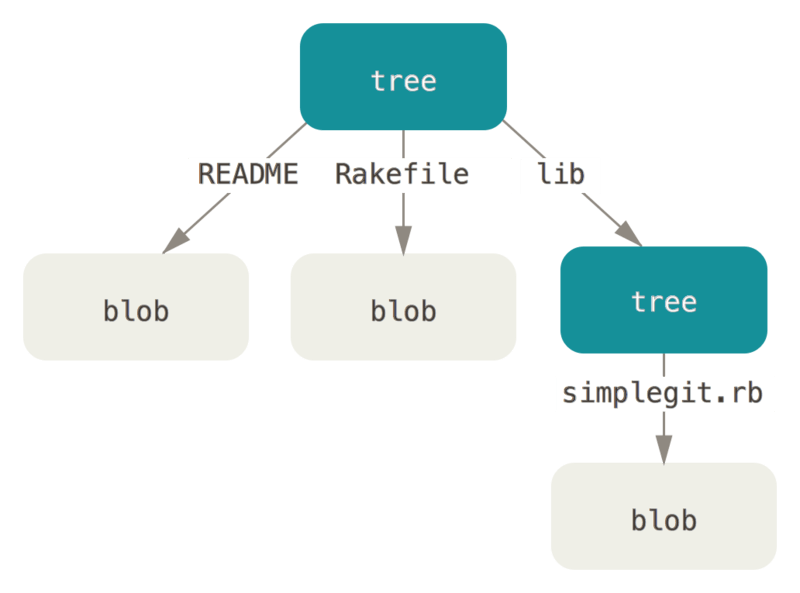


Figure 23 Tree

The tree object contains one line per file or subdirectory, with each line giving file permissions, object type, object hash and filename. Object type is usually one of “blob” for a file or “tree” for a subdirectory

You can use the write-tree command to write the staging area out to a tree object. No –w option is needed – calling write-tree automatically creates a tree object from the state of the index if that tree doesn’t yet exist:

**$** git write-tree

d8329fc1cc938780ffdd9f94e0d364e0ea74f579

**$** git cat-file -p d8329fc1cc938780ffdd9f94e0d364e0ea74f579

100644 blob 83baae61804e65cc73a7201a7252750c76066a30 test.txt

**What is Blob?**

The directory listing gave us the hash of the stored of example\_file.txt. This object is of type “blob” and contains the file snapshot:

$ git cat-file -t 2f781156939ad540b2434d012446154321e41e03

blob

$ git cat-file -p 2f781156939ad540b2434d012446154321e41e03

An example file

Blob is an abbreviation for “binary large object”. When we git add a file such as example\_file.txt, git creates a blob object containing the contents of the file. Blobs are therefore the git object type for storing files.

**What is SHA-1?**

Each object ASCII SHA-1 hash is converted and stored in binary format. If what you need is just a binary version of the ASCII hashes, you can do it with:

**Patch**

Generate patch in email form

Commad: $ git format-patch -1

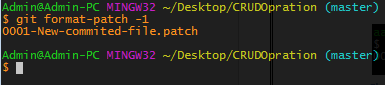


Figure Patch

**Email Send Repository git Command Line**

1. Download and install the latest version of [GitHub Desktop](https://desktop.github.com/). This will automatically install computer, open the **Git Shell** application.
2. Tell Git your name so your commits will be properly labeled. Type everything after the $ here:
3. git config --global user.name "Nikhil Barot"
4. Tell Git the email address that will be associated with your Git commits. The email you specify should be the same one found in your [email settings](https://help.github.com/articles/adding-an-email-address-to-your-github-account/). To keep your email address hidden, see "[Keeping your email address private](https://help.github.com/articles/keeping-your-email-address-private)".

git config --global user.email [nikhilbarot19@gmail.com](mailto:nikhilbarot19@gmail.com)

$ git send-email -to "nikhilbarot19@gmail.com" 0001-New-commited-file.patch

0001-New-commited-file.patch

(mbox) Adding cc: nikhil <nikhilbarot19@gmail.com> from line 'From: nikhil <nikhilbarot19@gmail.com>'

From: nikhil <nikhilbarot19@gmail.com>

To: nikhilbarot19@gmail.com

Subject: [PATCH] New commited file

Date: Tue, 18 Oct 2016 23:47:09 +0530

Message-Id: <20161018181709.4612-1-nikhilbarot19@gmail.com>

X-Mailer: git-send-email 2.9.0.windows.1

The Cc list above has been expanded by additional

addresses found in the patch commit message. By default

send-email prompts before sending whenever this occurs.

This behavior is controlled by the sendemail.confirm

configuration setting.

For additional information, run 'git send-email --help'.

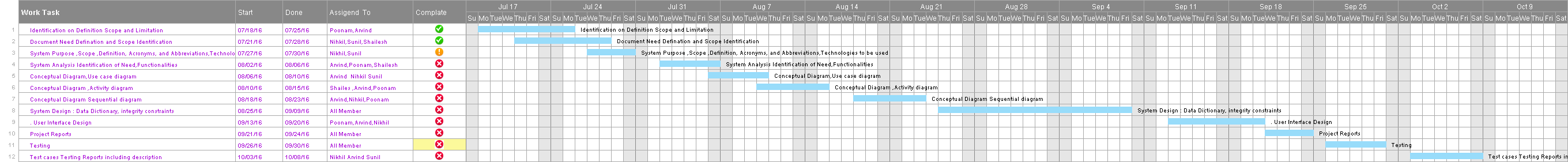
To retain the current behavior, but squelch this message,

run 'git config --global sendemail.confirm auto'.

Send this email? ([y]es|[n]o|[q]uit|[a]ll): **y**

Unable to initialize SMTP properly. Check config and use --smtp-debug. VALUES: server=localhost encryption= hello=localhost.localdomain port=25 at C:\Program Files\Git\mingw32/libexec/git-core\git-send-email line 1384, <FIN> line 1.

**Time Line Chart:**

****

**References**

1. OReilly Version Control With Git 2nd Edition Aug 2012 ISBN 144931
2. Pro Git Second Edition

<https://github.com>

http://stackoverflow.com