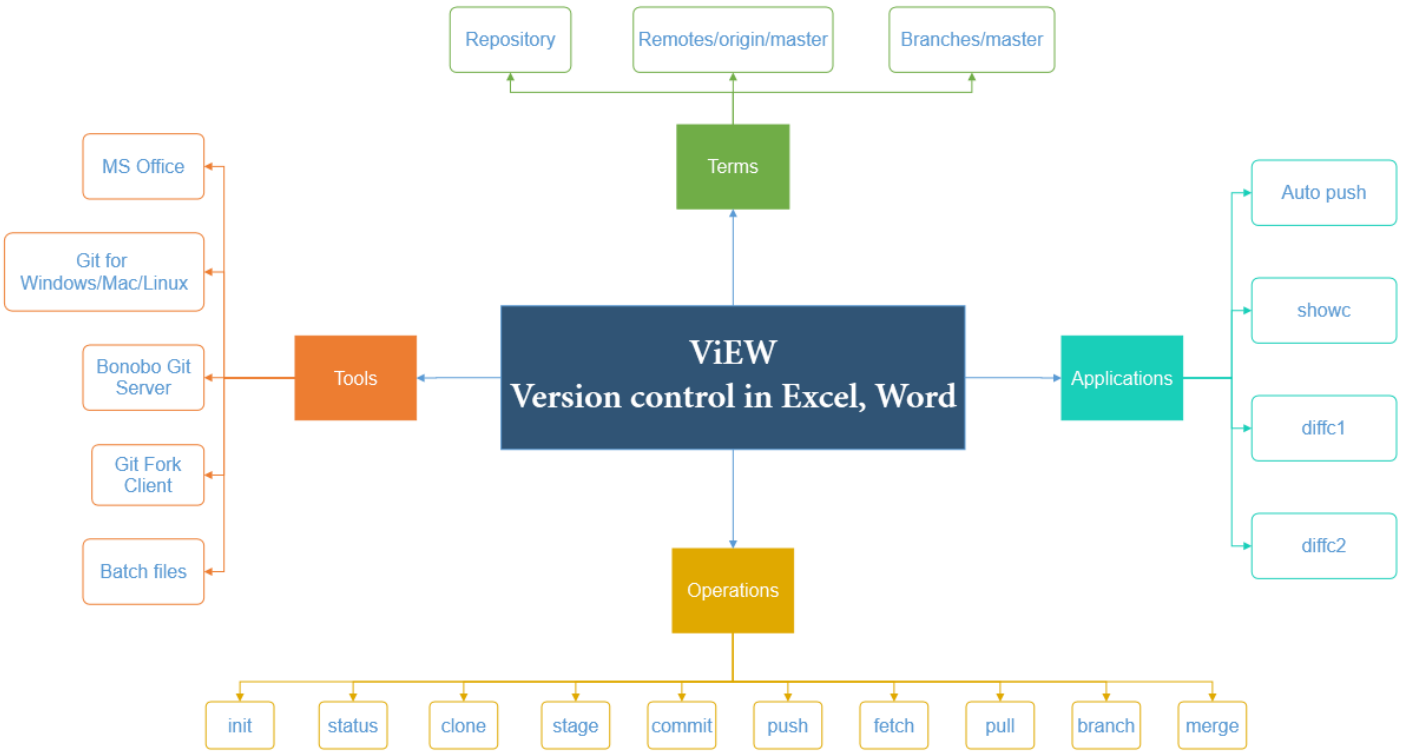
ViEW (Version control system in Excel, Word)

Version: 0.1

A software tool to use Git in Office | Excel, Word



Tools Installation

MS Excel 2013 Professional Plus

Git for Windows

Batch programs

Environment PATH variables

Bonobo Git Server

Fork Git Client

Standard

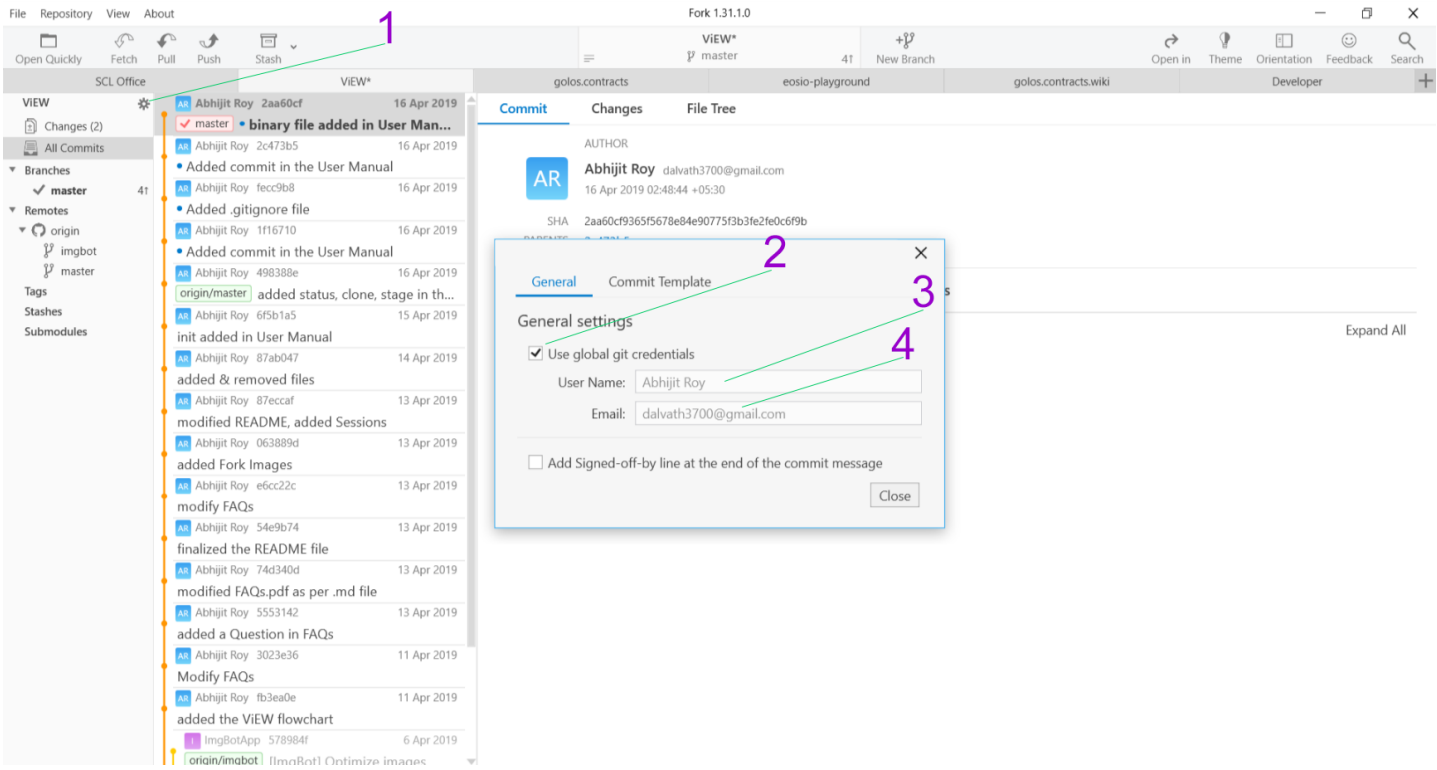
- Click the Setup file and install it as per instruction.

> NOTE: No Administrator permission required.

- After Installation, it would ask for User details- `User Name` , `Email` . Give your full name and Office's email address. This would help in recording the author details when making any file changes.

> NOTE: Although in the beginning, user details are taken, but for each repository there is an option to use different user details as well.

Global User credentials: Can be used for all repository



In the Image above,

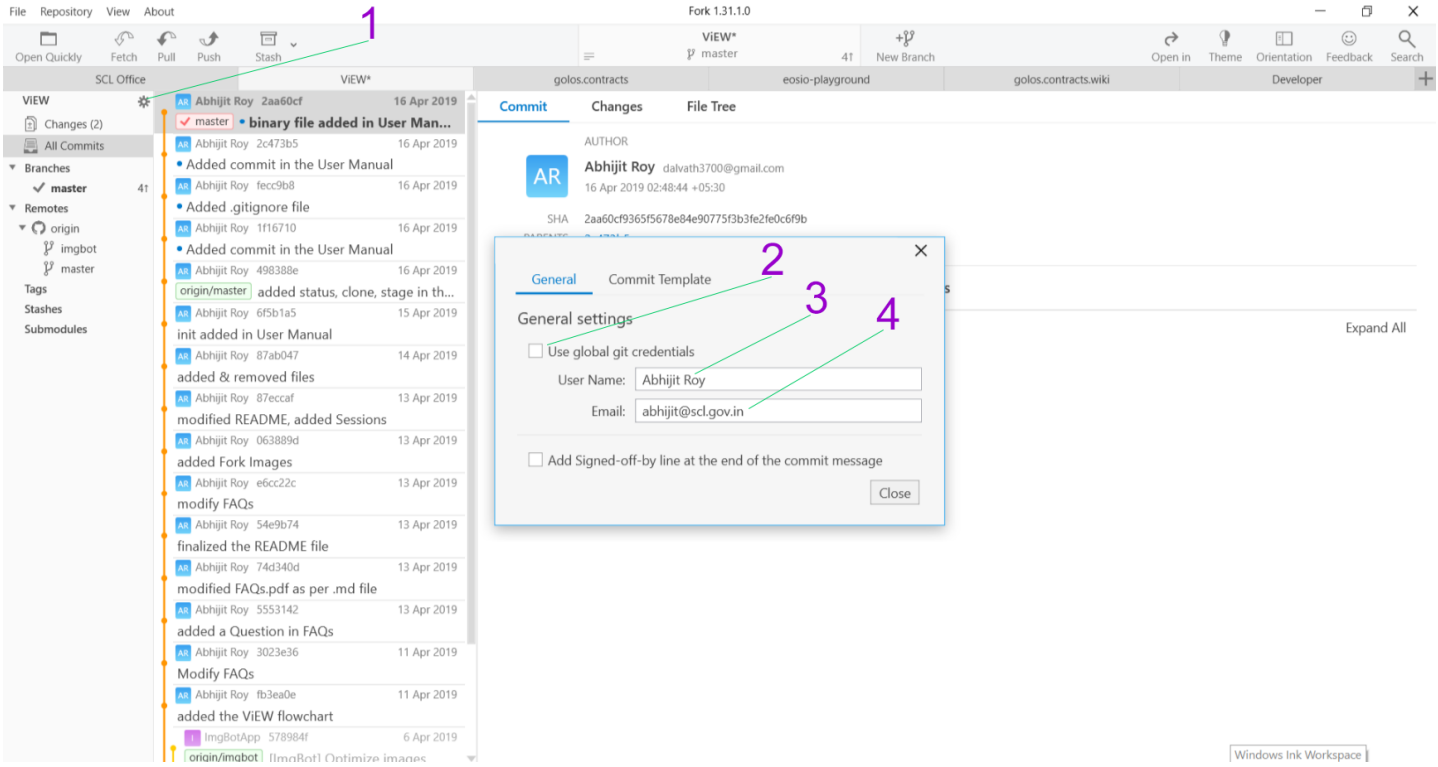
1 - Click the Repository Settings to bring the dialog.

2 - Tick to use the global user credentials for this repository.

3 - Full name (of global) for this repository. Asked during Installation.

4 - Email address (of global) for this repository. Asked during Installation.

Local User credentials: Can be used for a repository



In the Image above,

1 - Click the Repository Settings to bring the dialog.

2 - Untick to use different user credentials (other than global) for this repository.

3 - Full name for this repository.

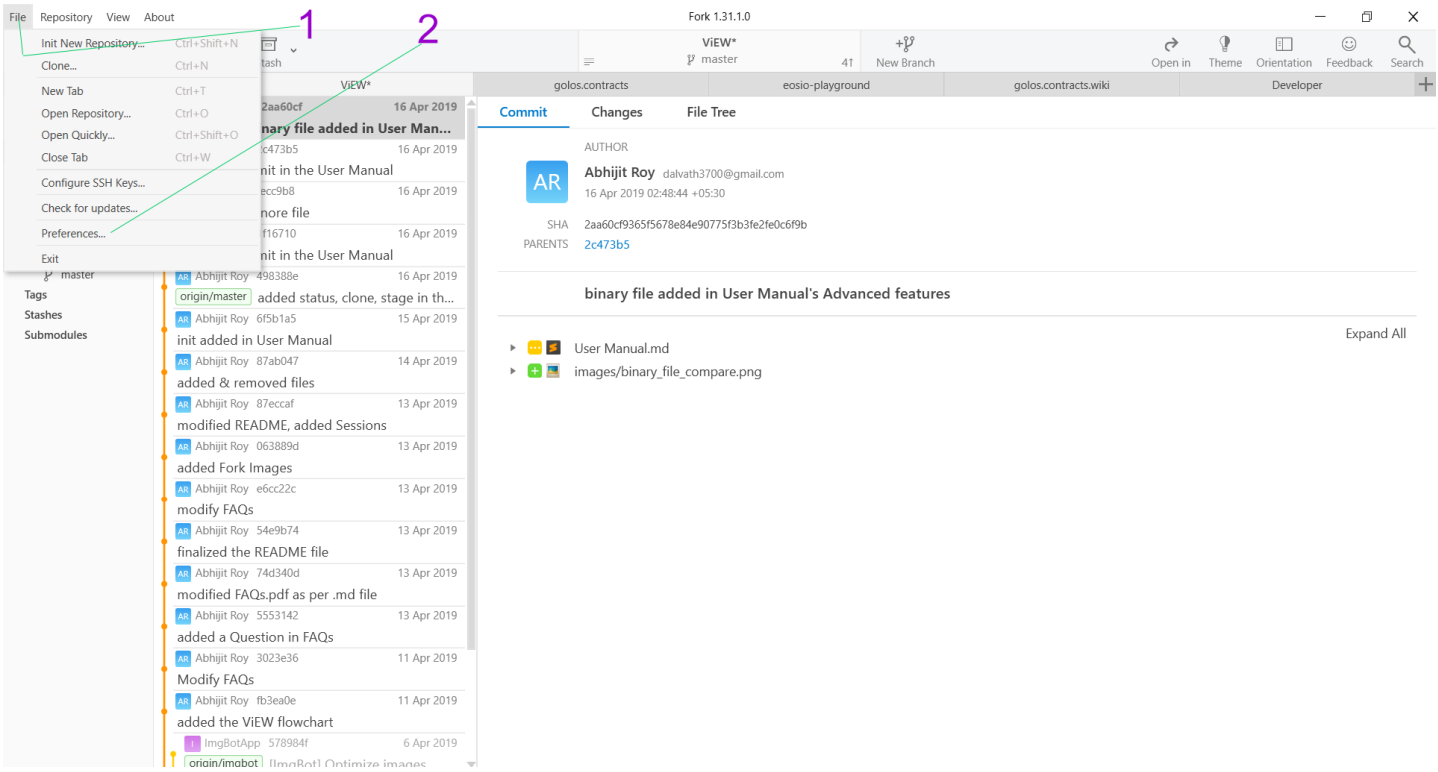
4 - Email address for this repository.

Advanced

- **Open in Sublime Text 3** If am Universal Editor is needed, `Sublime Text 3` can be used in that case.

  Also, it can be merged with the Fork Client.

  Step 1:

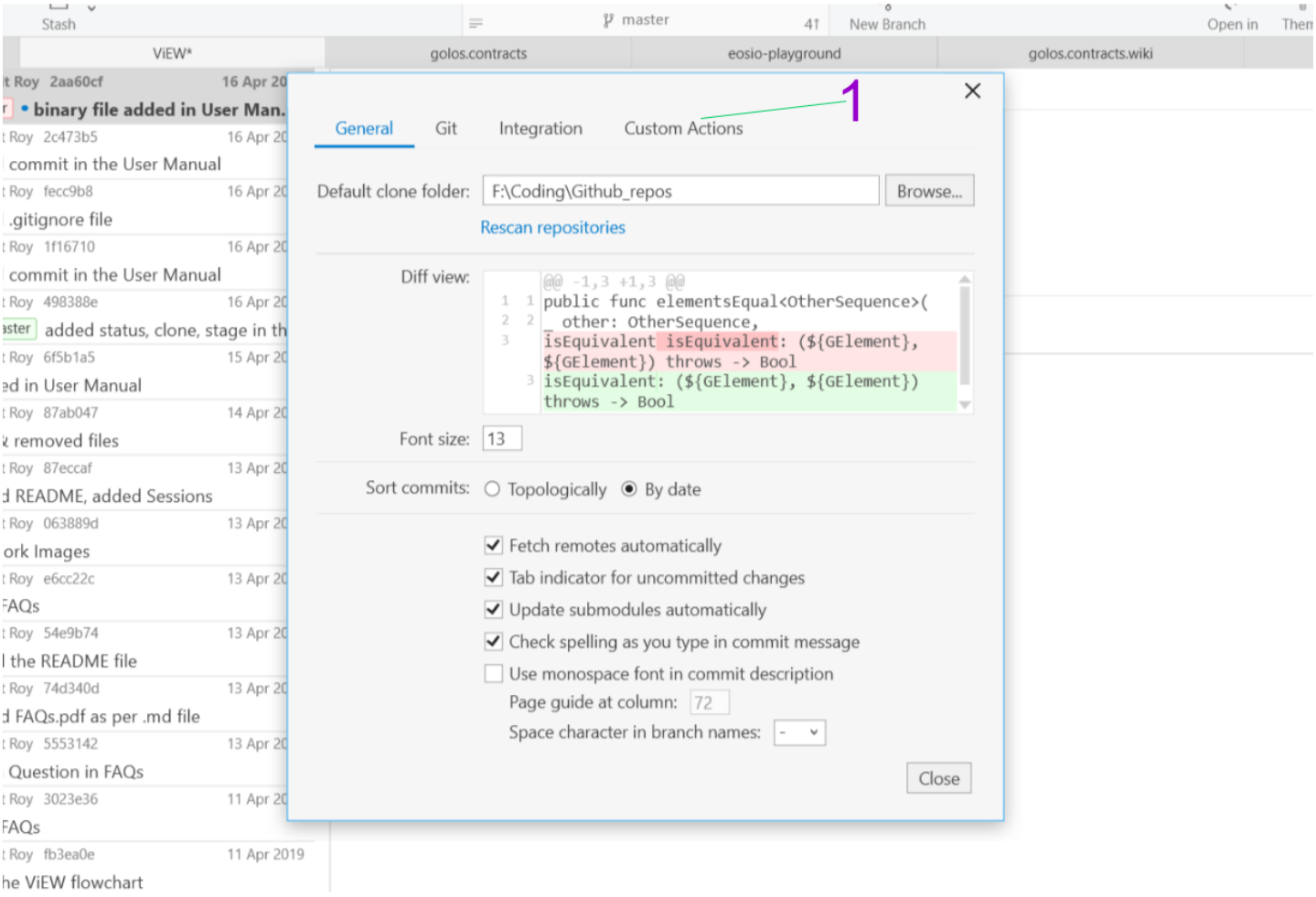


In the Image above,

1 - Click **File** menu option

2 - Click **Preferences..**

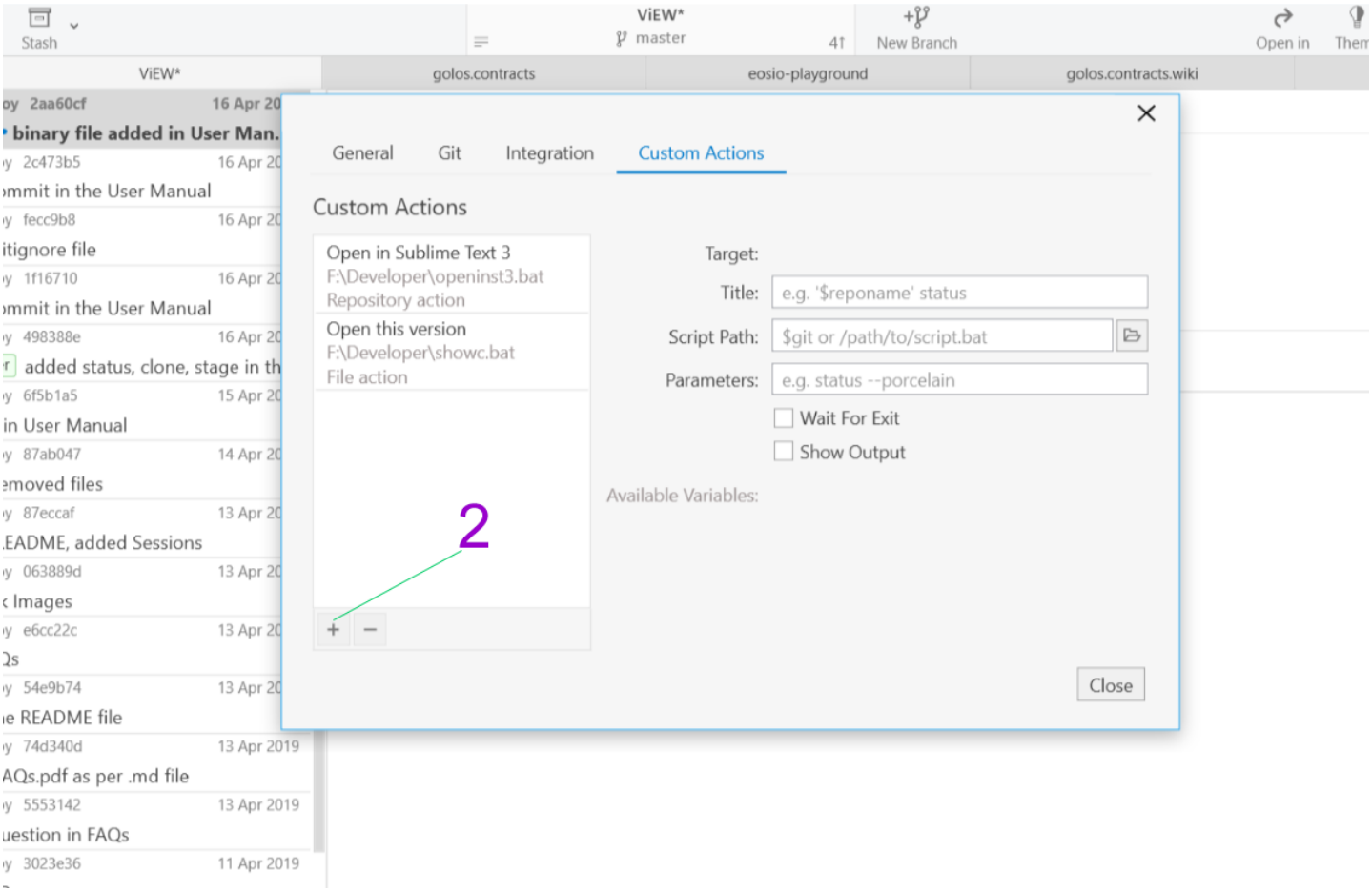
Step 2:



In the Image above,

1 - Now, click the **Custom Actions**

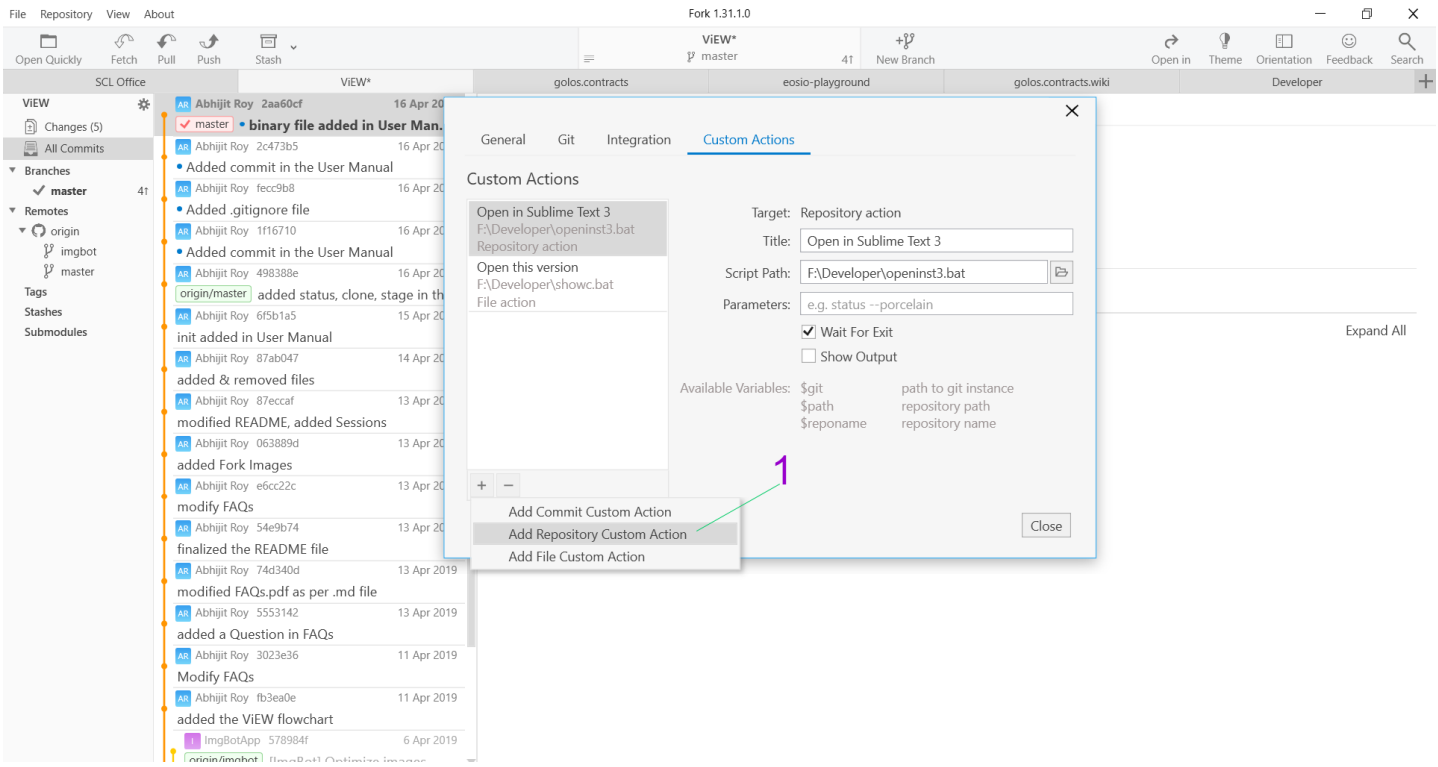
Step 3:



In the Image above,

2 - Click the **+** symbol to create the custom action.

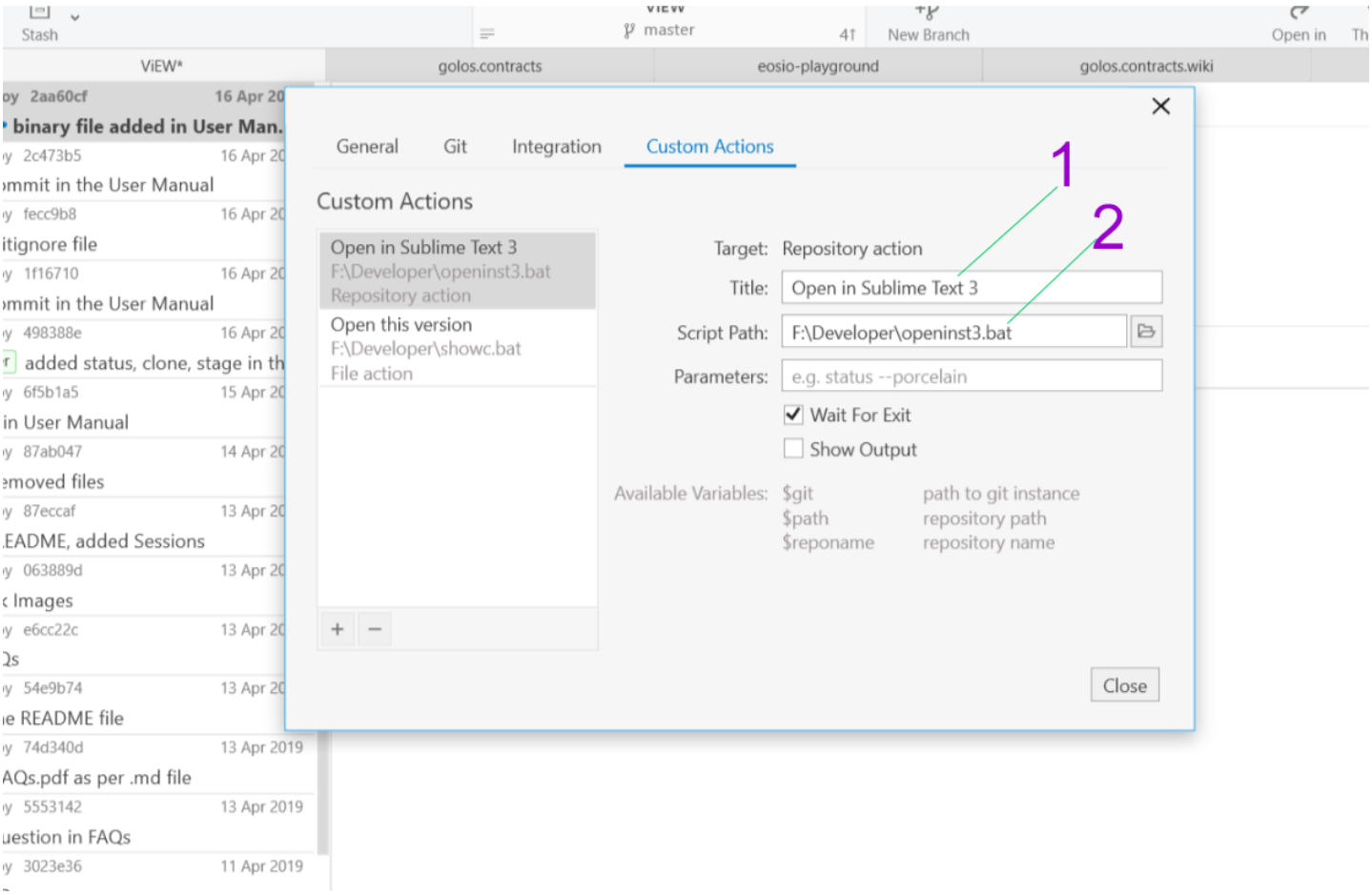
Step 4:



In the Image above,

1 - Click this to show the custom action dialog.

Step 5:

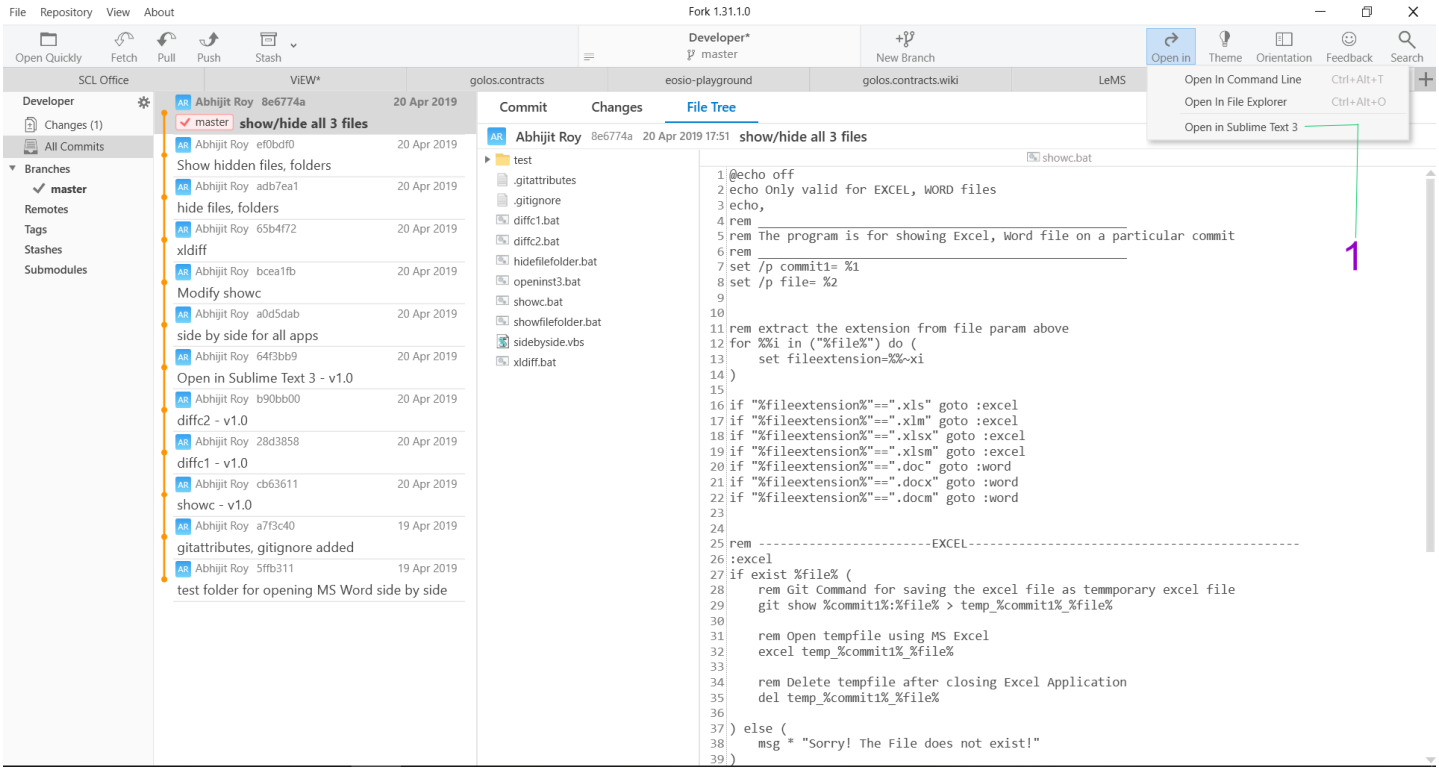


In the Image above,

1 - Write the title for custom action

2 - Link the `.bat` or `.exe` file (from Developer folder, explained above) to open **Sublime Text 3** in current repository path.

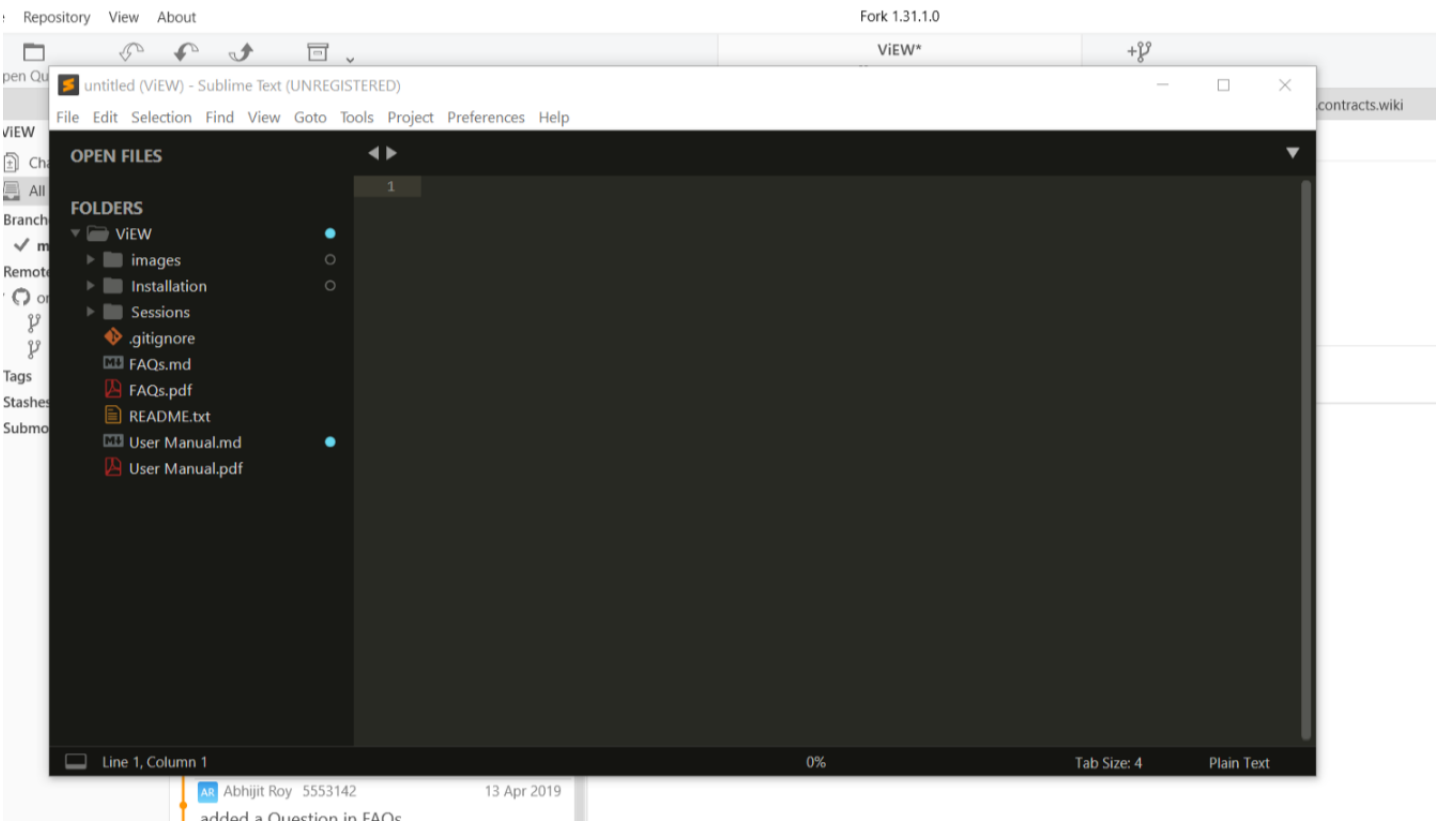
Step 6:



In the Image above,

1 - Finally, this opens **Sublime Text 3** for current repository.

Step 7:



In the Image above,

1 - This is how the repository looks like in **Sublime Text 3**.

> NOTE: Now, this can be used especially, if someone is maintaining project directory with version control, this **Sublime Text 3** is really going to be helpful while coding practice (with too many custom themes with modern colors.)

Terms

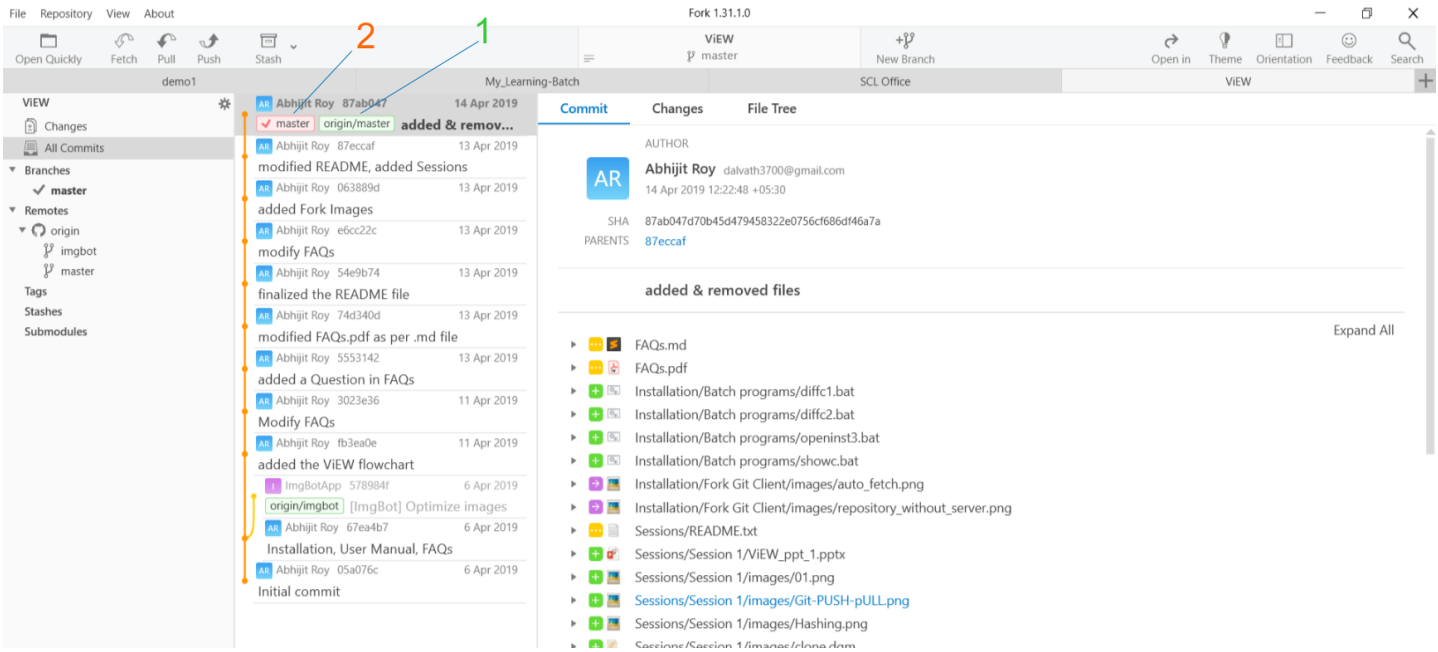
1. `Repository`

   Repository is termed as any folder with files (any format), sub-folders inside it.

2. `remotes/origin/master`

   It refers to the **master** (or main) branch of the repository kept at remote server.

   In the image above,
   `1` - shows the `origin/master`, basically the master (or main) branch of repository (named - ViEW) kept at remote location.

3. `branch/master`

   It refers to the **master** (or main) branch of the repository (named - ViEW) kept at local storage (like PC, desktop).

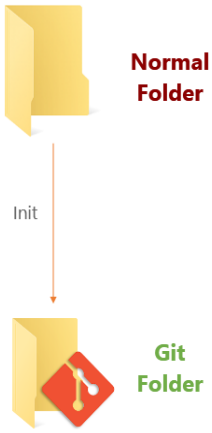
   In the image above,
   `2` - shows the `branches/master`, basically the master (or main) branch of folder kept at local location.

   > NOTE: Both the masters (at remote & local) are in sync.

Operations

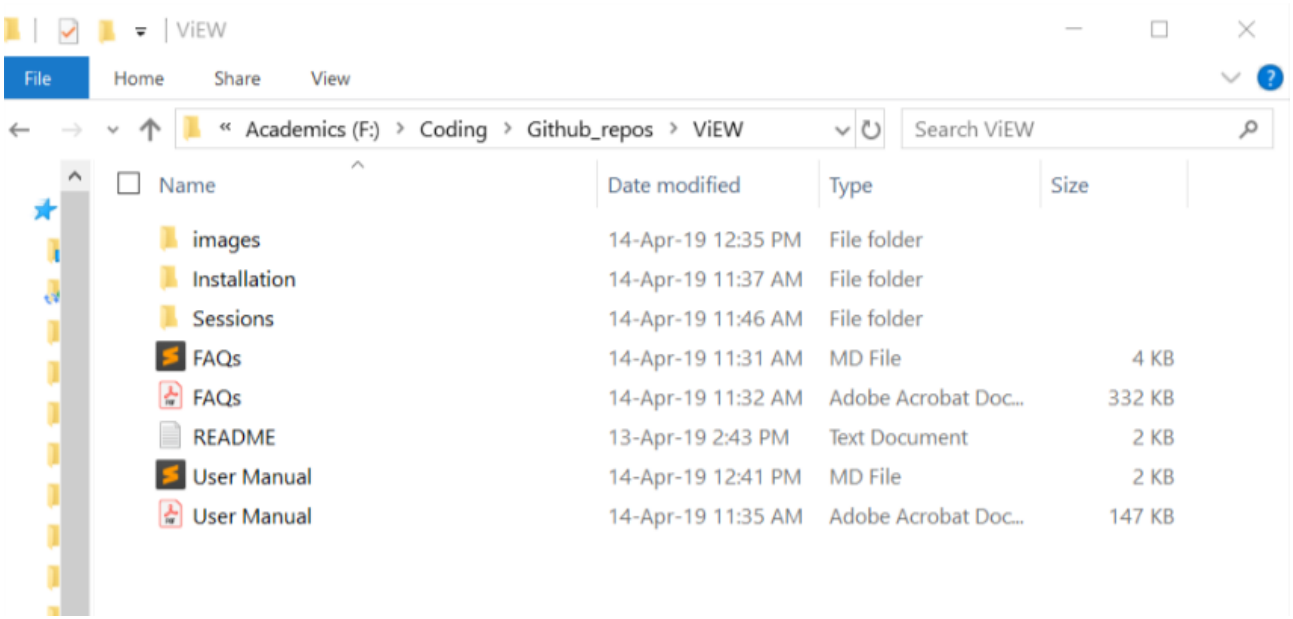
1. `init`

   It refers to initializing a non-git (or normal) folder to git repository.

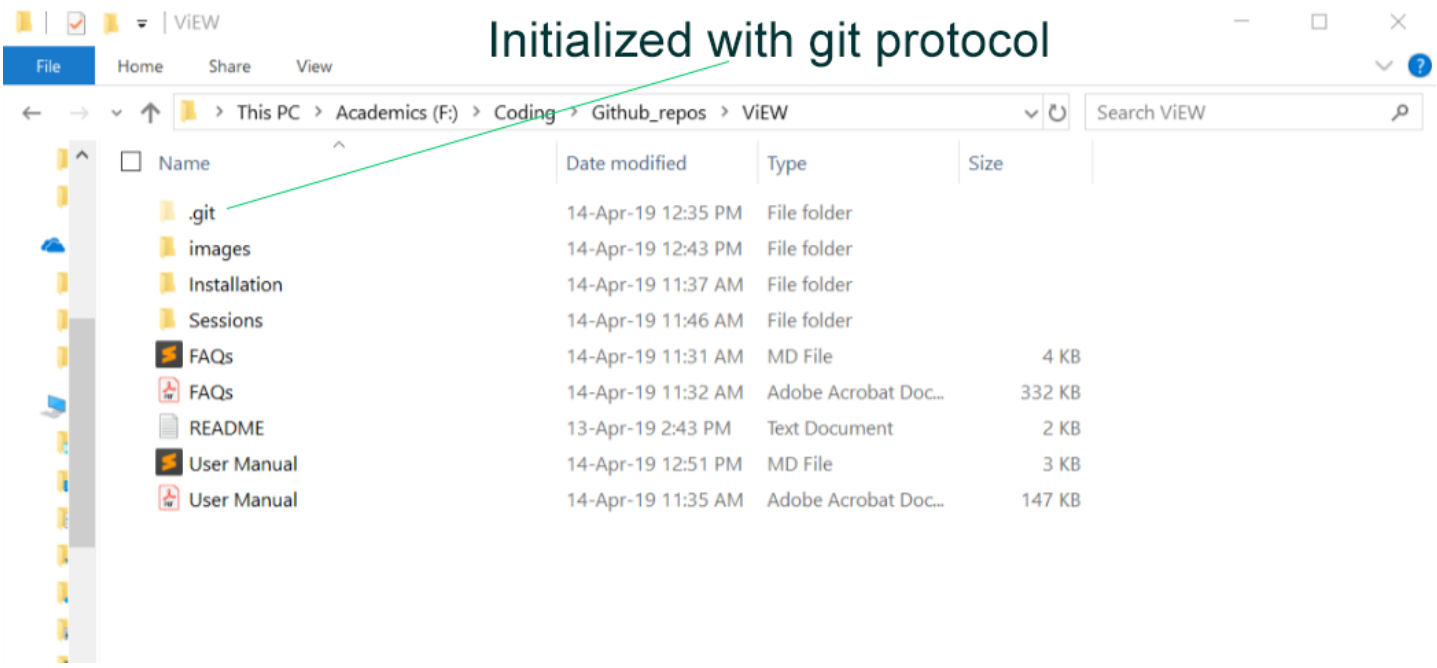


Example:

Normal repository:



Git repository:

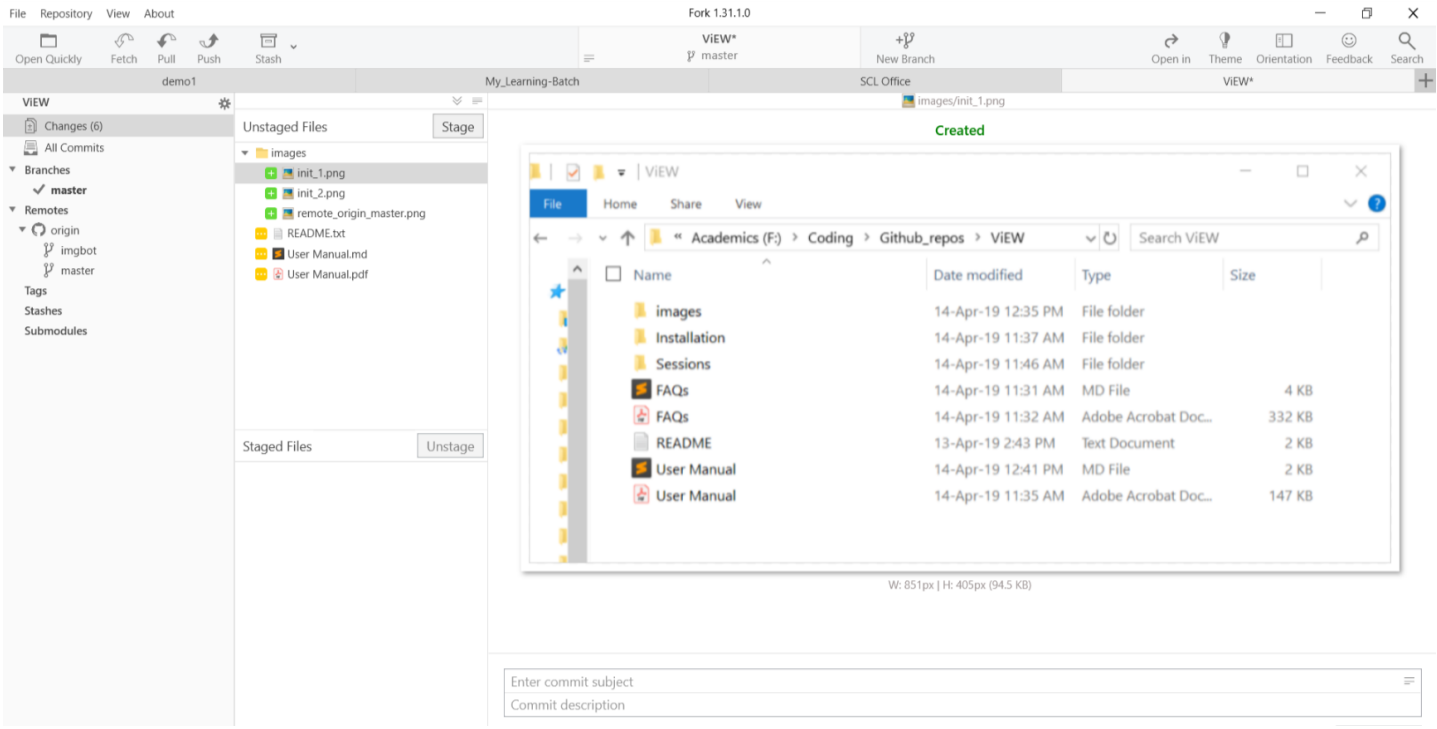


> NOTE: From now onwards, **Git repository** will be called as **repository** (in short).

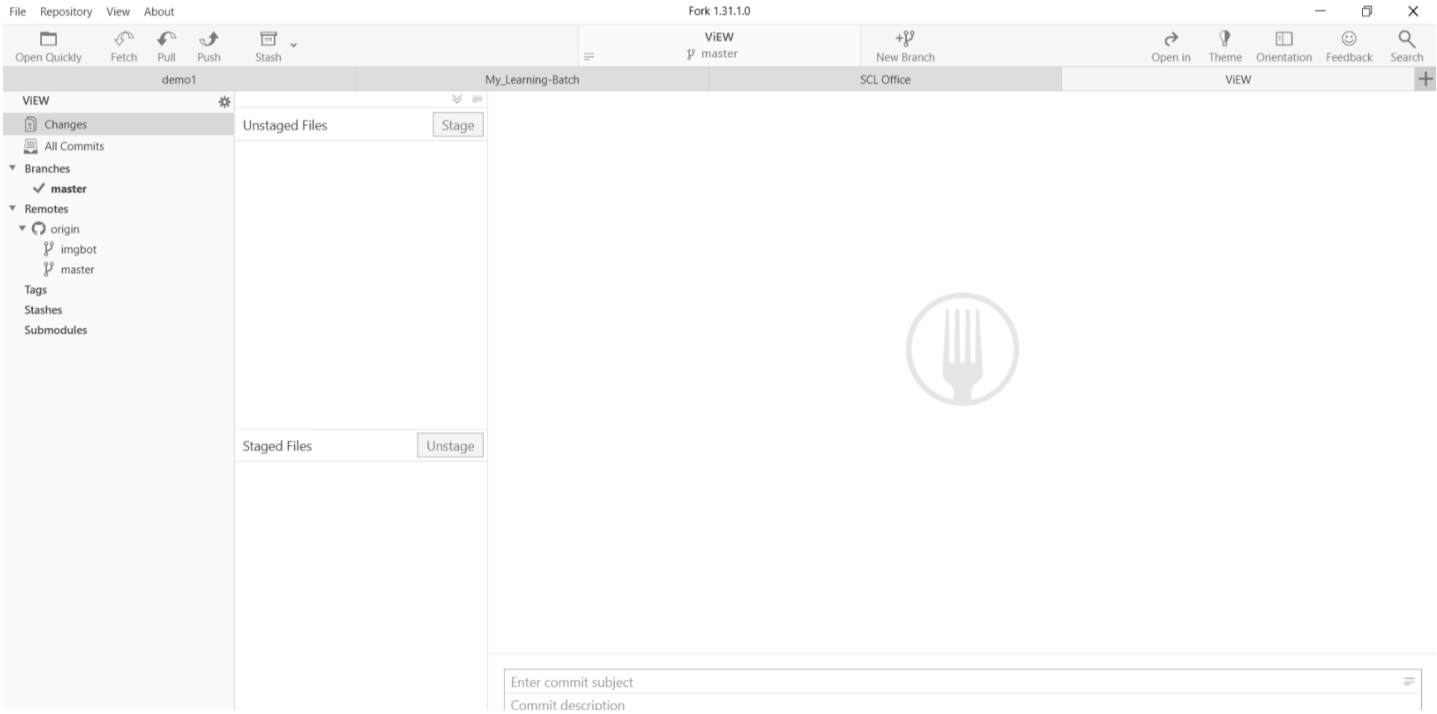
2. `status`

It shows the status of the files (inside repository) changed. To see the changes (if any), click "Changes" on the left pane of Fork Application.

Example 1: Status showing file changes

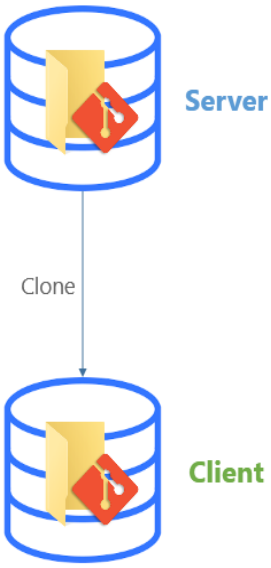


Example 2: Status showing NO file changes

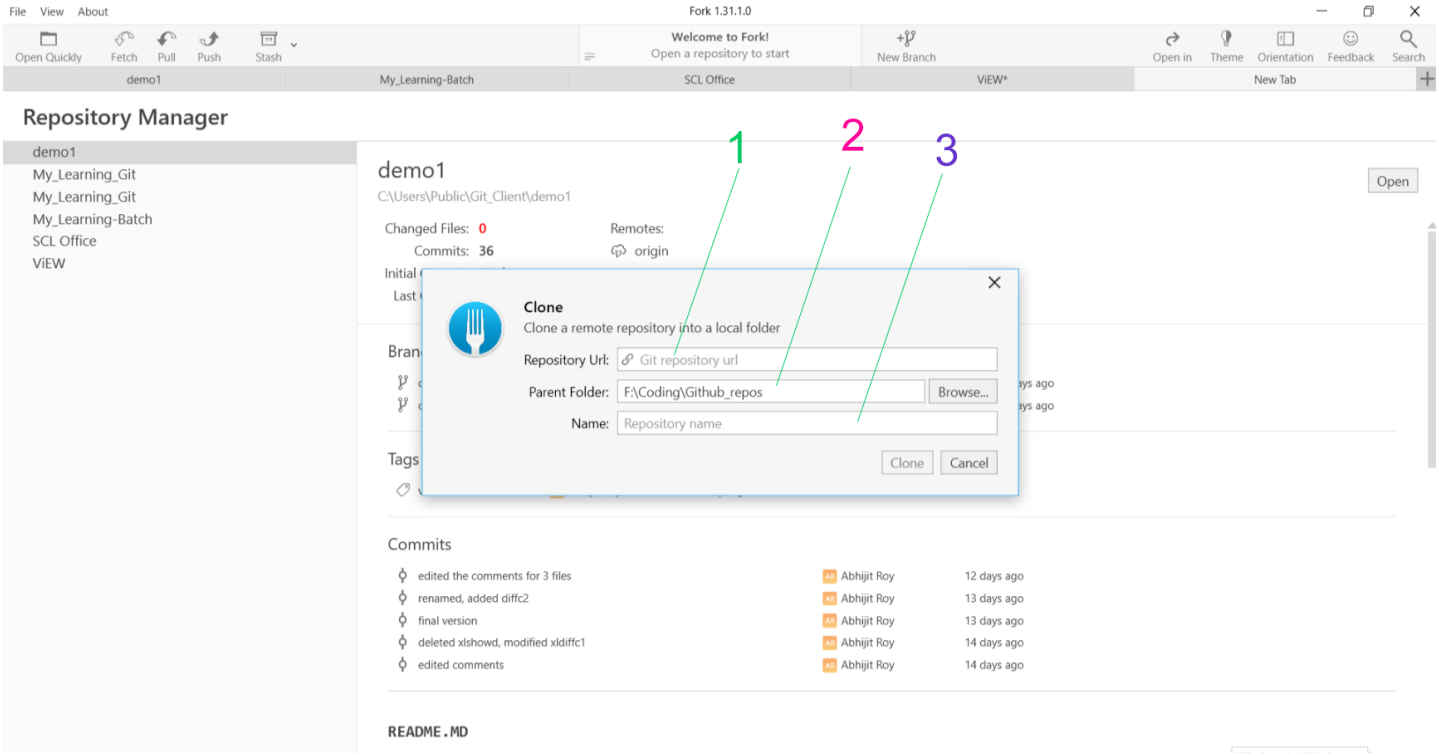


3. `clone`

   This is to clone/download the repository (from remote location) to a desired directory in the local PC/Desktop.



Example: Clone a repository



In the Image above,

1 - Remote URL of the repository. E.g.: "http://localhost/Bonobo.Git.Server/demo1.git"

2 - Local directory where the repository is to be cloned.

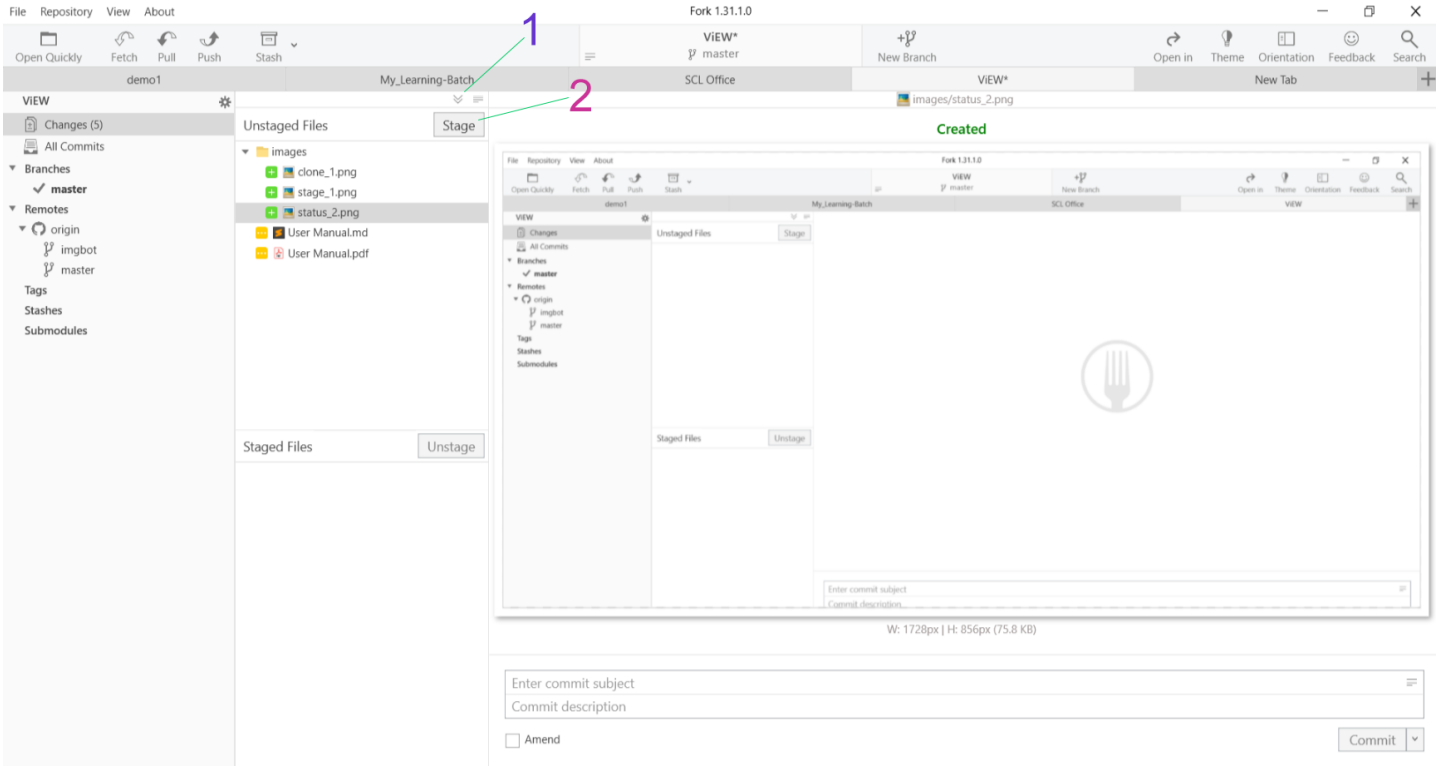
3 - Custom Name for the cloned repository.

4. `stage`

Add file(s) to the Staging area. It's like adding/registering files for recording changes (in the repository).

In the image above, user has to stage the file(s) to **Staging area** before recording changes.

Example: Unstaged Files

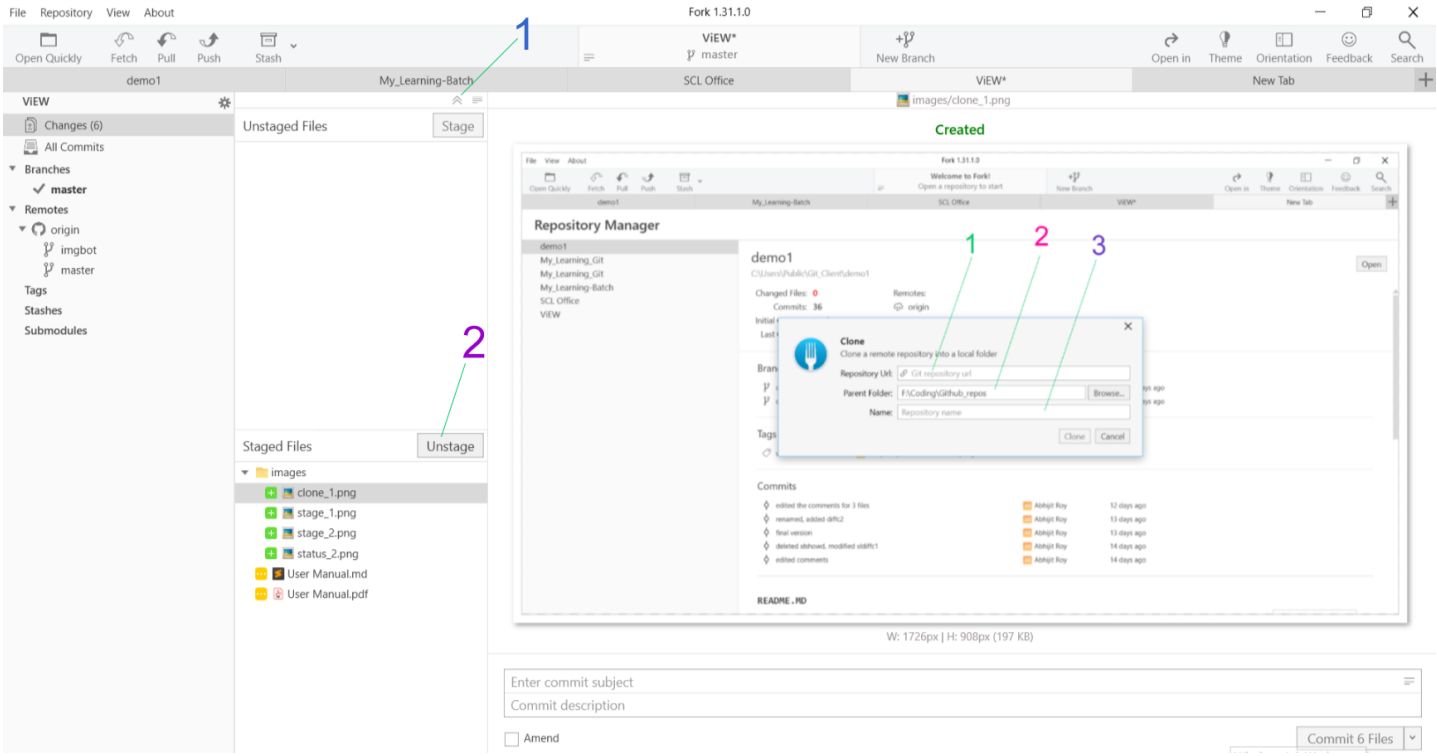


In the Image above,

1 - Stage all file(s)

2 - Stage selected file(s)

Example: Staged Files



In the Image above,

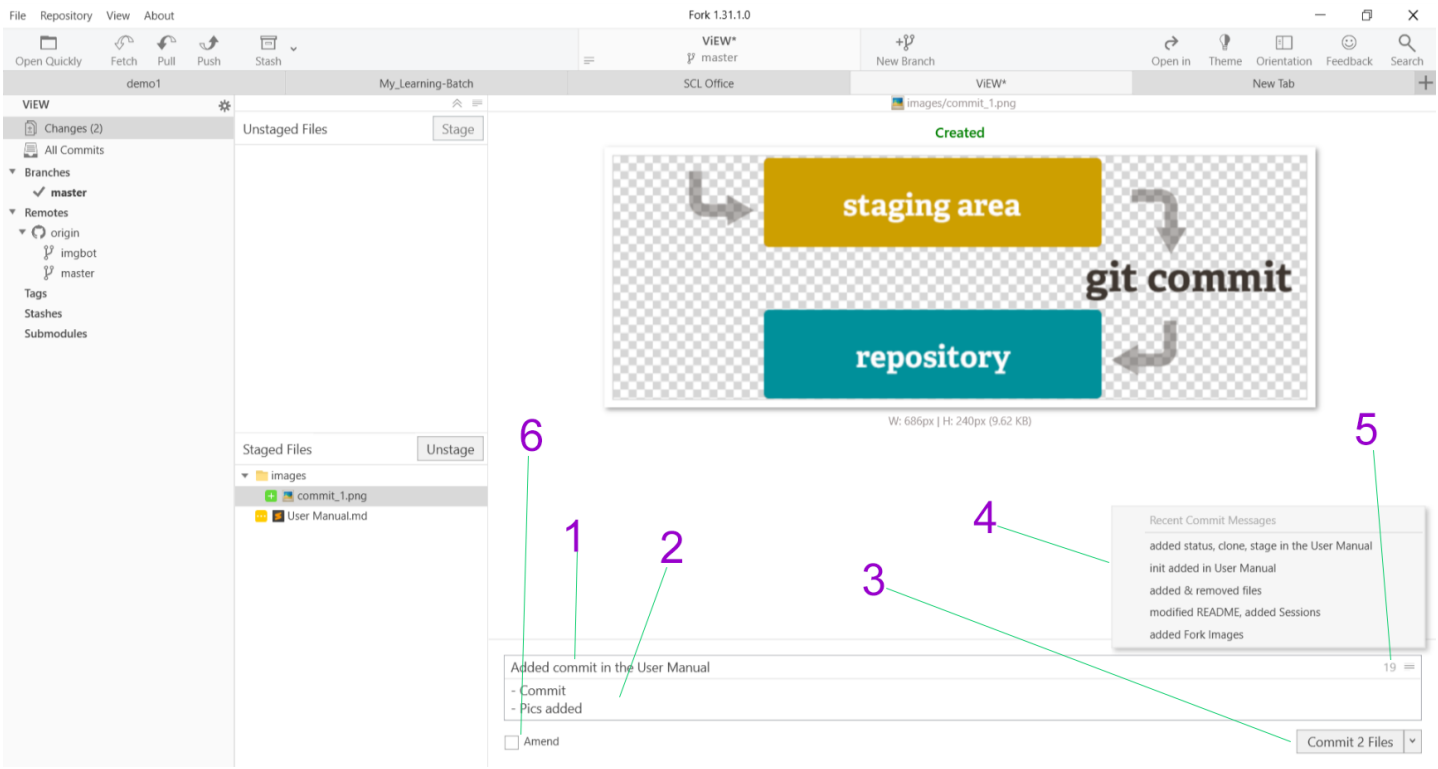
1 - Unstage all file(s)

2 - Unstage selected file(s)

5. `commit`

This is to record file changes and add it to the chain history. Here, **commit** means assigning a random unique no. (called as Cryptographic Hash) to a change.

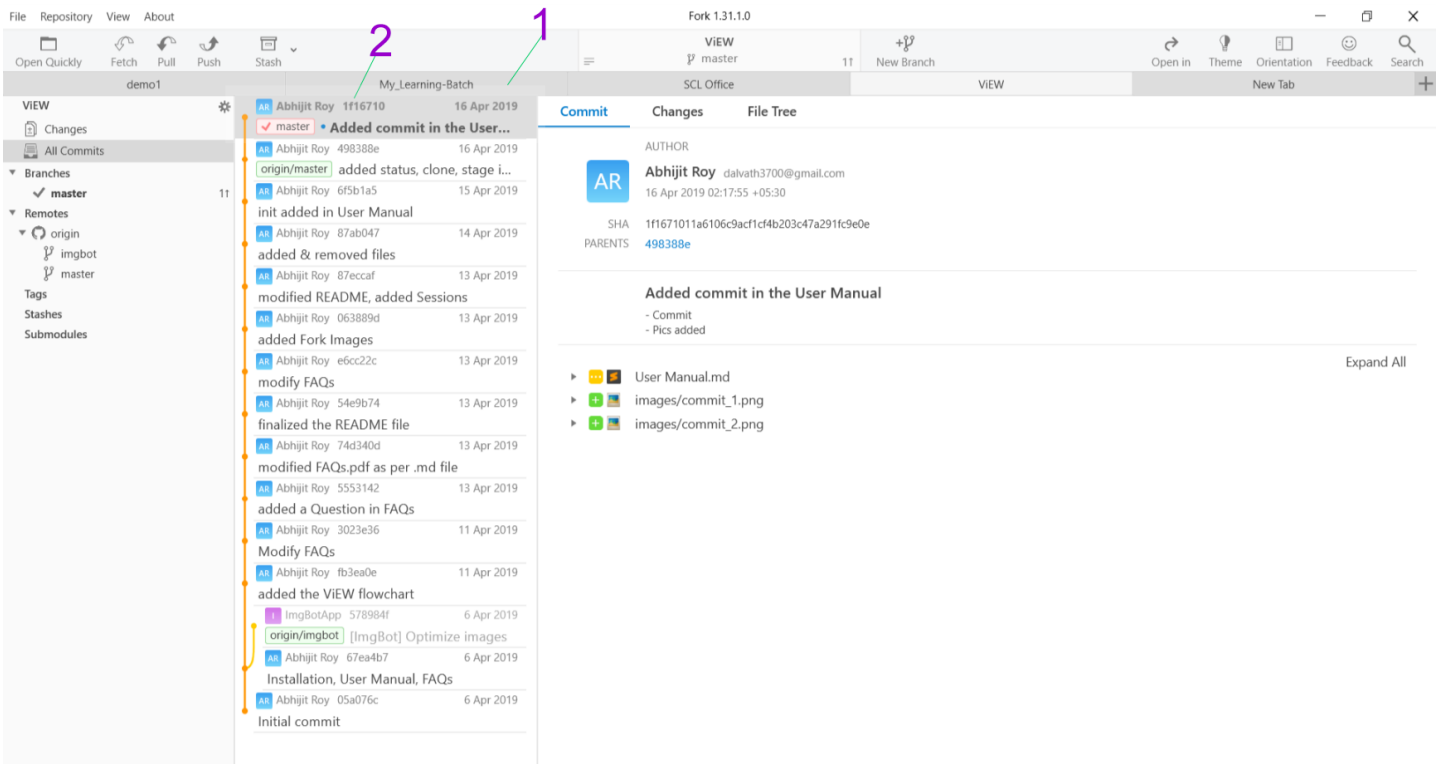
Example: Commit the Staged files



In the Image above,

1 - Commit Message Title
2 - Commit Message Description
3 - Commit button (when clicked => committed/recorded)
4 - Old Commit Message Titles. Can be used when repetitive title required.
5 - Show Old commit message titles
6 - Amend button i.e. when clicked, automatically uses last message title and description

Example: Commit added to the Chain history



In the Image above,

1 - Latest block (with files changes) added

2 - commit Hash (unique Cryptographic Hash using SHA1 Algorithm)

6. `push`
7. `fetch`
8. `pull`

Utility

- `showc`
- `diffc1`
- `diffc2`
- `autopush`
- `allsync`

Advanced Features

- Binary File - TODO
- File change security - TODO
- Old Block security - TODO