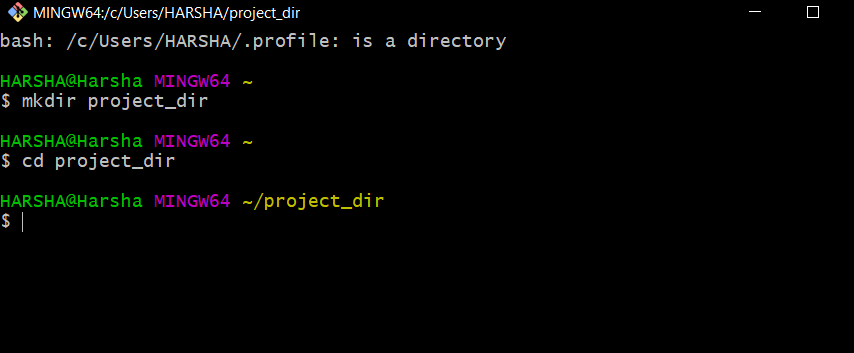
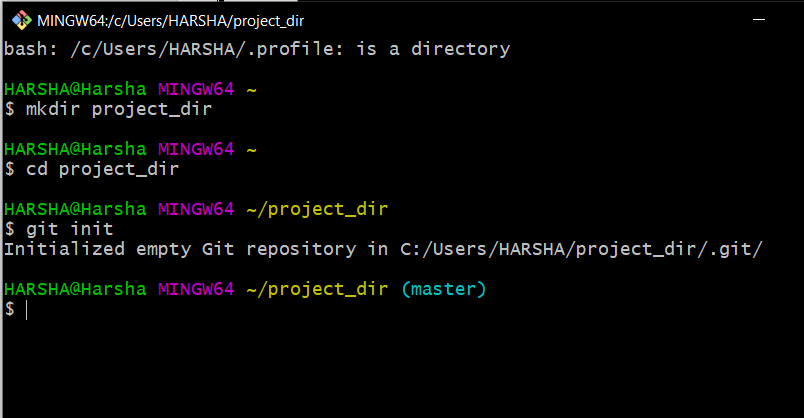
**GIT Assignment -By** K Jyothi Gangarsha **ID:** 259921

**Install GIT & make sure it is added into PATH. Use GIT as local VCS. Steps to follow:**

1. **Create a directory ‘project\_dir’ & cd to ‘project\_dir’.**
2. **Initialize git version database. (git init)**

I’m copy pasting the bash, as the snipping and pasting was time consuming.

1. **Create a new file index.html.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ touch index.html

1. **Check the git status. You should find index.html as untracked file.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.html

nothing added to commit but untracked files present (use "git add" to track)

1. **Stage the index.html file.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git add index.html

1. **Commit index.html**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git commit -m 'first html'

[master (root-commit) e293f4e] first html

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 index.html

1. **Make few changes in index.html & create a new file info.txt file.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ touch info.txt

1. **Check git status. You should find index.html & info.txt as untracked files.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

Untracked files:

(use "git add <file>..." to include in what will be committed)

info.txt

no changes added to commit (use "git add" and/or "git commit -a")

1. **Configure GIT to ignore all txt files.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ touch .gitignore

Now open the gitignore text file and include \*.txt to remove all txt files to get ignored.

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git add .gitignore

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

1. **Again check the git status. You should find only index.html as untracked file.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

1. **State & commit index.html**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git add index.html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitignore

modified: index.html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git commit -m 'index2'

[master 16ea5b4] index2

2 files changed, 4 insertions(+)

create mode 100644 .gitignore

1. **Log all your comments so far.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git log

commit 16ea5b46edf639452e09f860ae142c6055a74e9a (HEAD -> master)

Author: Jyothi Gangarsha Karampudi <kjgangarsha@gmail.com>

Date: Sun Jan 10 13:54:12 2021 +0530

index2

commit e293f4eb3b2db8714d25ec0c74097b4a03272012

Author: Jyothi Gangarsha Karampudi <kjgangarsha@gmail.com>

Date: Sun Jan 10 13:19:24 2021 +0530

first html

1. **Make some changes in index.html.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

1. **Revert the change made in the previous step using git command.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git restore index.html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

nothing to commit, working tree clean

1. **Again change index.html.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

1. **Stage index.html**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git add index.html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.html

1. **Revert back the last stage.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git commit -m 'commit after edit'

[master 657a14a] commit after edit

1 file changed, 1 insertion(+), 1 deletion(-)

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git log --oneline

657a14a (HEAD -> master) commit after edit

16ea5b4 index2

e293f4e first html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git reset HEAD~1

Unstaged changes after reset:

M index.html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

1. **Rename ‘add’ command to ‘my-add’.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git config --global alias.my-add add

1. **Using my\_add command Stage index.html again & commit the changes.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git my-add index.html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git commit -m 'final index'

[master b692cb8] final index

1 file changed, 1 insertion(+), 1 deletion(-)

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

nothing to commit, working tree clean

1. **Revert the last commit.**

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git log --oneline

b692cb8 (HEAD -> master) final index

16ea5b4 index2

e293f4e first html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git reset HEAD~1

Unstaged changes after reset:

M index.html

HARSHA@Harsha MINGW64 ~/project\_dir (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

**GIT Branching**

**Objective: Commit HTML, CSS & JavaScript assignments into GIT. SECTION-1 (HTML assignments) - Steps to follow:**

1. **First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.**
2. **Create an empty directory ‘Assignments’ & cd to ‘Assignments’.**

HARSHA@Harsha MINGW64 ~

$ mkdir assignments

HARSHA@Harsha MINGW64 ~

$ cd assignments

HARSHA@Harsha MINGW64 ~/assignments

$ git init

Initialized empty Git repository in C:/Users/HARSHA/assignments/.git/

1. **Create a file README.txt inside ‘Assignments’ & write few lines about** **the contents of ‘Assignments’ folder.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ touch README.txt

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git add .

1. **Commit README.txt file.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git commit -m 'readme file'

[master (root-commit) ff19bbd] readme file

1 file changed, 5 insertions(+)

create mode 100644 README.txt

1. **Now create a new branch ‘html-assignments’.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git branch html-assignments

1. **Switch to ‘html-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

1. **Copy all HTML assignments inside ‘Assignments’ folder.**

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git status

On branch html-assignments

Untracked files:

(use "git add <file>..." to include in what will be committed)

ex1.html

ex2.html

ex3.html

nothing added to commit but untracked files present (use "git add" to track)

1. **Commit HTML assignments into ‘html-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git status

On branch html-assignments

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: ex1.html

new file: ex2.html

new file: ex3.html

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git commit -m 'html assign1'

[html-assignments 4c0c292] html assign1

3 files changed, 3 insertions(+)

create mode 100644 ex1.html

create mode 100644 ex2.html

create mode 100644 ex3.html

1. **Make minor changes into few files belonging to ‘html-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git status

On branch html-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: ex2.html

modified: ex3.html

no changes added to commit (use "git add" and/or "git commit -a")

1. **Commit those changed files.**

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git status

On branch html-assignments

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: ex2.html

modified: ex3.html

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git commit -m 'minor changes in html'

[html-assignments 16fa4b5] minor changes in html

2 files changed, 4 insertions(+), 1 deletion(-)

1. **Switch to master branch.**

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git checkout master

Switched to branch 'master'

1. **Make minor changes into README.txt file & commit those changes into master.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: README.txt

no changes added to commit (use "git add" and/or "git commit -a")

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git commit -m 'edit readme'

[master 60c57cc] edit readme

1 file changed, 3 insertions(+), 1 deletion(-)

1. **Again switch to ‘html-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git checkout html-assignments

Switched to branch 'html-assignments'

1. **Make minor changes into few files belonging to ‘html-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git status

On branch html-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: ex1.html

no changes added to commit (use "git add" and/or "git commit -a")

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git add .

1. **Commit those changes.**

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git commit -m 'assign2'

[html-assignments 49f03c5] assign2

1 file changed, 1 insertion(+)

1. **Switch to master.**

HARSHA@Harsha MINGW64 ~/assignments (html-assignments)

$ git checkout master

Switched to branch 'master'

1. **Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git merge html-assignments

Auto-merging README.txt

Merge made by the 'recursive' strategy.

README.txt | 2 +-

ex1.html | 2 ++

ex2.html | 3 +++

ex3.html | 2 ++

4 files changed, 8 insertions(+), 1 deletion(-)

create mode 100644 ex1.html

create mode 100644 ex2.html

create mode 100644 ex3.html

1. **Finally delete the ‘html-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git branch -d html-assignments

Deleted branch html-assignments (was 49f03c5).

**SECTION-2 - (CSS assignments) Steps to follow:**

1. **Create a new branch ‘css-assignments’.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git branch css-assignments

1. **Switch to ‘css-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git checkout css-assignments

Switched to branch 'css-assignments'

1. **Copy all CSS assignments inside ‘Assignments’ folder.**

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git status

On branch css-assignments

Untracked files:

(use "git add <file>..." to include in what will be committed)

ex4.css

ex5.css

nothing added to commit but untracked files present (use "git add" to track)

1. **Commit CSS assignments into ‘css-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ add .

bash: add: command not found

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git commit -m 'css1'

[css-assignments a9a6128] css1

2 files changed, 2 insertions(+)

create mode 100644 ex4.css

create mode 100644 ex5.css

1. **Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git status

On branch css-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: README.txt

no changes added to commit (use "git add" and/or "git commit -a")

1. **Commit those changed files.**

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git commit -m 'readme1'

[css-assignments a50aed9] readme1

1 file changed, 1 insertion(+)

1. **Switch to master branch.**

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git checkout master

Switched to branch 'master'

1. **Make minor changes into README.txt file on line 3 & commit those changes into master.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: README.txt

no changes added to commit (use "git add" and/or "git commit -a")

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git commit -m 'readme master'

[master a382181] readme master

1 file changed, 1 insertion(+), 1 deletion(-)

1. **Again switch to ‘css-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git checkout css-assignments

Switched to branch 'css-assignments'

1. **Make minor changes into few files belonging to ‘css-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git status

On branch css-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: ex4.css

no changes added to commit (use "git add" and/or "git commit -a")

1. **Commit those changes.**

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git commit -m 'css branch'

[css-assignments ceafd63] css branch

1 file changed, 1 insertion(+), 1 deletion(-)

1. **Switch to master.**

HARSHA@Harsha MINGW64 ~/assignments (css-assignments)

$ git checkout master

Switched to branch 'master'

1. **Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git merge css-assignments

Auto-merging README.txt

Merge made by the 'recursive' strategy.

README.txt | 1 +

ex4.css | 1 +

ex5.css | 1 +

3 files changed, 3 insertions(+)

create mode 100644 ex4.css

create mode 100644 ex5.css

1. **Finally delete the ‘css-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git branch -d css-assignments

Deleted branch css-assignments (was ceafd63).

**SECTION-3 - (JavaScript assignments) Steps to follow:**

1. **Create a new branch ‘js-assignments’.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git branch js-assignments

1. **Switch to ‘js-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git checkout js-assignments

Switched to branch 'js-assignments'

1. **Copy all JavaScript assignments inside ‘Assignments’ folder.**

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git status

On branch js-assignments

Untracked files:

(use "git add <file>..." to include in what will be committed)

ex6.js

ex7.js

nothing added to commit but untracked files present (use "git add" to track)

1. **Commit JavaScript assignments into ‘js-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git commit -m 'js1'

[js-assignments bdff7ba] js1

2 files changed, 3 insertions(+)

create mode 100644 ex6.js

create mode 100644 ex7.js

1. **Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git status

On branch js-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: README.txt

no changes added to commit (use "git add" and/or "git commit -a")

1. **Commit those changed files.**

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git commit -m 'readme js'

[js-assignments a3d70ee] readme js

1 file changed, 1 insertion(+), 1 deletion(-)

1. **Switch to master branch.**

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git checkout master

Switched to branch 'master'

1. **Make minor changes into README.txt file on line 1 & commit those changes into master.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: README.txt

no changes added to commit (use "git add" and/or "git commit -a")

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git commit -m 'readme js1'

[master 84ea179] readme js1

1 file changed, 1 insertion(+), 1 deletion(-)

1. **Again switch to ‘js-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git checkout js-assignments

Switched to branch 'js-assignments'

1. **Make minor changes into few files belonging to ‘js-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git status

On branch js-assignments

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: ex7.js

no changes added to commit (use "git add" and/or "git commit -a")

1. **Commit those changes.**

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git add .

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git commit -m 'js2'

[js-assignments 51bf1e2] js2

1 file changed, 1 insertion(+), 1 deletion(-)

1. **Switch to master.**

HARSHA@Harsha MINGW64 ~/assignments (js-assignments)

$ git checkout master

Switched to branch 'master'

1. **Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git merge js-assignments

Auto-merging README.txt

Merge made by the 'recursive' strategy.

README.txt | 2 +-

ex6.js | 1 +

ex7.js | 2 ++

3 files changed, 4 insertions(+), 1 deletion(-)

create mode 100644 ex6.js

create mode 100644 ex7.js

1. **Finally delete the ‘js-assignments’ branch.**

HARSHA@Harsha MINGW64 ~/assignments (master)

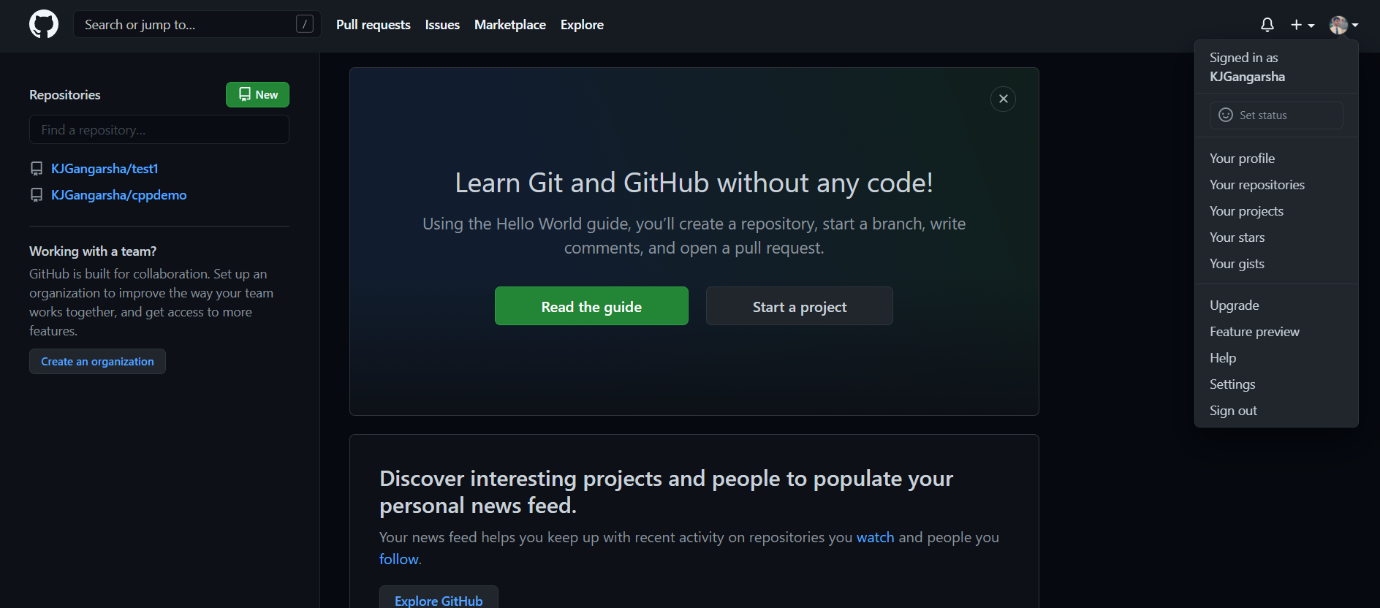
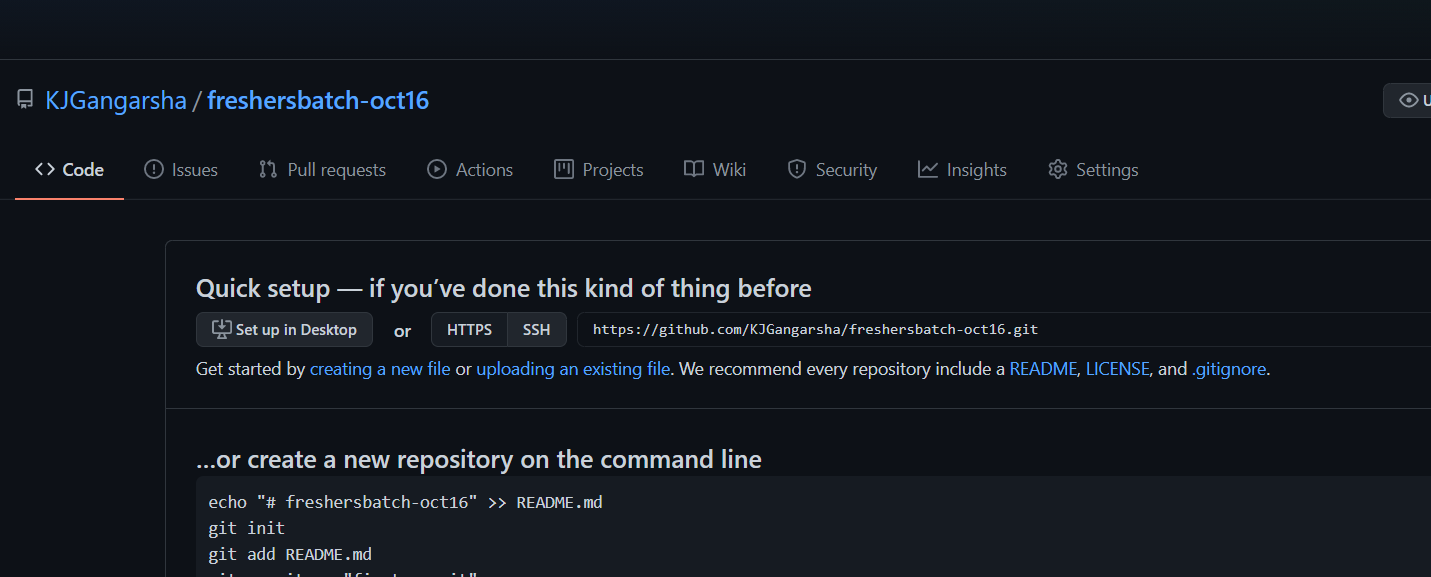
$ git branch -d js-assignments

Deleted branch js-assignments (was 51bf1e2).

**GIT REMOTING**

**Objective: Pushing source code into GITHUB & collaborate team members.**

**SECTION-1 (Pushing assignments to remote repository) - Steps to follow:**

1. **Create a github account if you do not have already.**
2. **Login on into github account.**
3. **Create new public repository ‘freshersbatch-oct16’.**
4. **Commit & push any sample file to this repository under ‘Assignments’ directory.**

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git remote add origin https://github.com/KJGangarsha/freshersbatch-oct16.git

HARSHA@Harsha MINGW64 ~/assignments (master)

$ git push origin master

Enumerating objects: 56, done.

Counting objects: 100% (56/56), done.

Delta compression using up to 4 threads

Compressing objects: 100% (41/41), done.

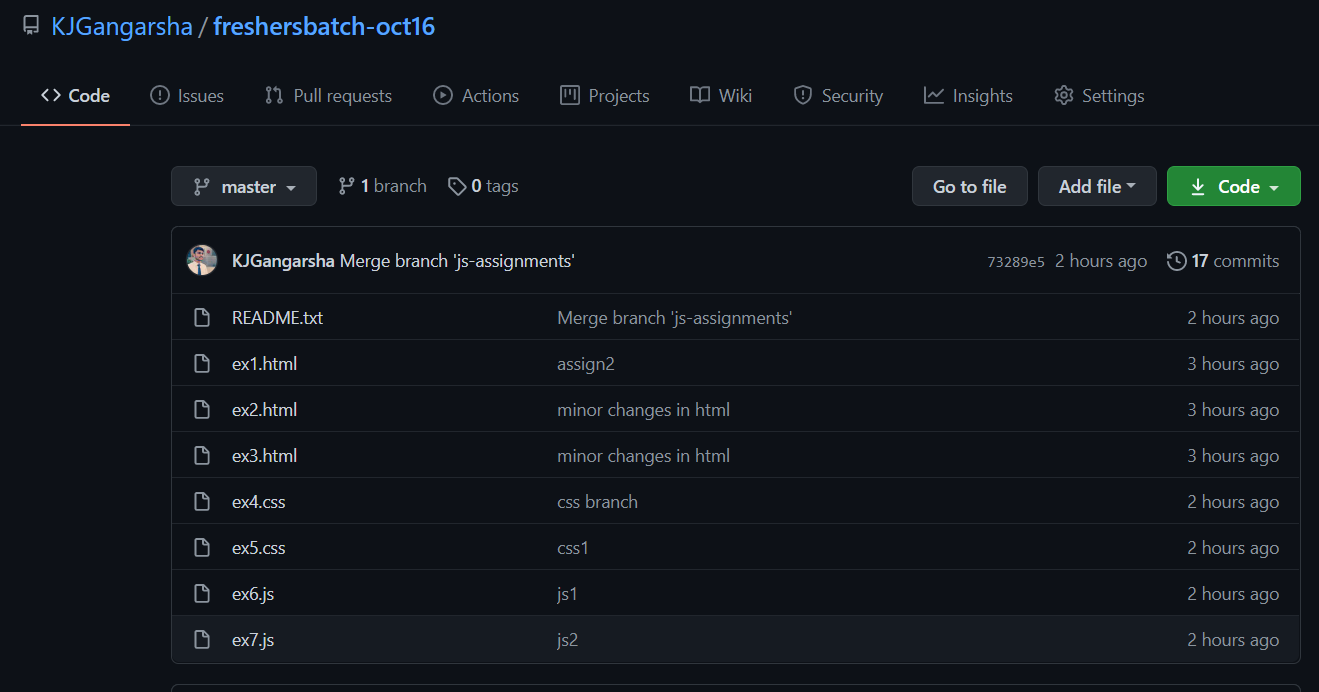
Writing objects: 100% (56/56), 4.55 KiB | 274.00 KiB/s, done.

Total 56 (delta 23), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (23/23), done.

To https://github.com/KJGangarsha/freshersbatch-oct16.git

\* [new branch] master -> master

****

**---THANKYOU---**