

COGNIZANCE

Task-1

DONE BY:

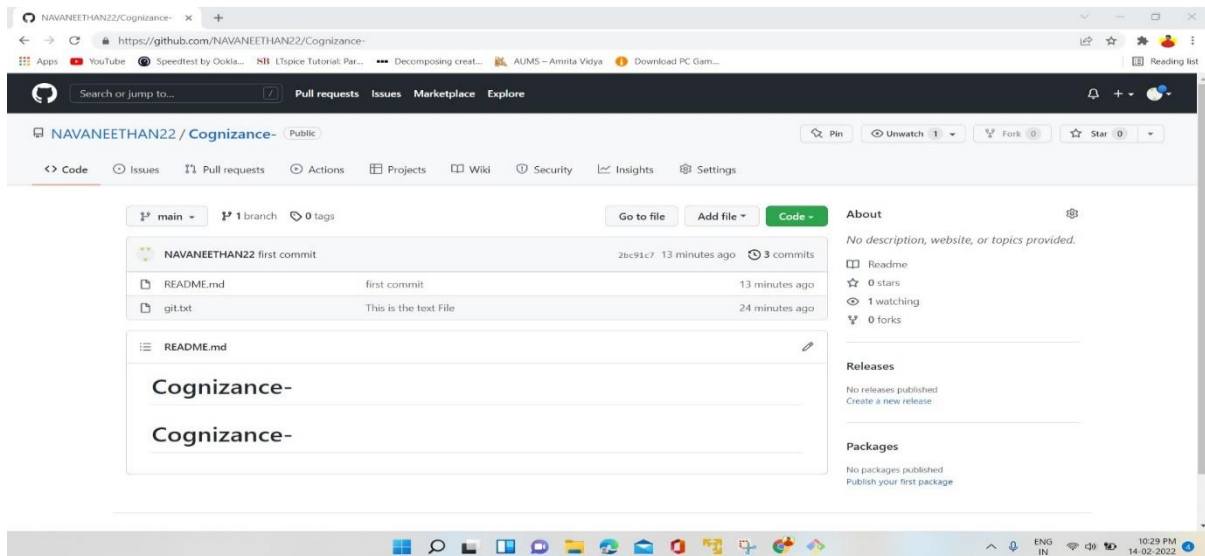
NAVANEETHAN.S

CH.EN.U4CYS21048

Ch.en.u4cys21048@ch.students.amrita.edu

GITHUB AND GIT OPREATIONS

1)web page:



2)To Create Github repo:

```
MINGW64:/c/Users/ADMIN/github
ADMIN@DESKTOP-V6G60VF MINGW64 ~
$ git config --global user.email 12navaneethan@gmail.com

ADMIN@DESKTOP-V6G60VF MINGW64 ~
$ git config --global user.name NAVANEETHAN22

ADMIN@DESKTOP-V6G60VF MINGW64 ~
$ PWD
/c/Users/ADMIN

ADMIN@DESKTOP-V6G60VF MINGW64 ~
$ mkdr
bash: mkdr: command not found

ADMIN@DESKTOP-V6G60VF MINGW64 ~
$ mkdir github

ADMIN@DESKTOP-V6G60VF MINGW64 ~
$ cd github

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github
$ git init
Initialized empty Git repository in C:/Users/ADMIN/github/.git/

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (master)
$ touch git.txt

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (master)
$ git add .

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   git.txt

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (master)
$ git commit -a
hint: Waiting for your editor to close the file... unix2dos: converting file C:/
Users/ADMIN/github/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/ADMIN/github/.git/COMMIT_EDITMSG to Unix form
at...
[master (root-commit) dd06a5e] This is the text File
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 git.txt

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (master)
$ git log
commit dd06a5e1704f852c41a810506ade46b374b980a4 (HEAD -> master)
Author: NAVANEETHAN22 <12navaneethan@gmail.com>
Date: Mon Feb 14 22:04:44 2022 +0530

    This is the text File
```

3)Commands:

```
ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ echo "# Cognizance-" >> README.md

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git init
Reinitialized existing Git repository in C:/Users/ADMIN/github/.git/

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git commit -m "first commit"
[main 2bc91c7] first commit
1 file changed, 1 insertion(+)

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git branch -M main

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git remote add origin https://github.com/NAVANEETHAN22/Cognizance-.git
error: remote origin already exists.

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 762 bytes | 254.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/NAVANEETHAN22/Cognizance-.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$
```

```
ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git checkout
Your branch is up to date with 'origin/main'.

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git checkout -b
error: switch `b' requires a value
```

```

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git clone
fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

    -v, --verbose            be more verbose
    -q, --quiet              be more quiet
    --progress              force progress reporting
    --reject-shallow        don't clone shallow repository
    -n, --no-checkout        don't create a checkout
    --bare                  create a bare repository
    --mirror                create a mirror repository (implies bare)
    -l, --local              to clone from a local repository
    --no-hardlinks           don't use local hardlinks, always copy
    -s, --shared             setup as shared repository
    --recurse-submodules[=<pathspec>]
                             initialize submodules in the clone
    --recursive ...         alias of --recurse-submodules
    -j, --jobs <n>          number of submodules cloned in parallel
    --template <template-directory>
                             directory from which templates will be used
    --reference <repo>      reference repository
    --reference-if-able <repo>
                             reference repository
    --dissociate             use --reference only while cloning
    -o, --origin <name>     use <name> instead of 'origin' to track upstream
    -b, --branch <branch>   checkout <branch> instead of the remote's HEAD
    -u, --upload-pack <path>
                             path to git-upload-pack on the remote
    --depth <depth>         create a shallow clone of that depth
    --shallow-since <time>  create a shallow clone since a specific time
    --shallow-exclude <revision>
                             deepen history of shallow clone, excluding rev
    --single-branch          clone only one branch, HEAD or --branch
    --no-tags                don't clone any tags, and make later fetches not to follow them
    --shallow-submodules     any cloned submodules will be shallow
    --separate-git-dir <gitdir>
                             separate git dir from working tree
    -c, --config <key=value>
                             set config inside the new repository
    --server-option <server-specific>
                             option to transmit
    -4, --ipv4               use IPv4 addresses only
    -6, --ipv6               use IPv6 addresses only
    --filter <args>         object filtering
    --remote-submodules      any cloned submodules will use their remote-tracking branch
    --sparse                 initialize sparse-checkout file to include only files at root

```

```

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git fetch

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git merge
Already up to date.

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git pull
Already up to date.

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git push
Everything up-to-date

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git reset

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git rebase
Current branch main is up to date.

ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git fork
git: 'fork' is not a git command. See 'git --help'.

The most similar command is
    fsck

```

git config --list --show-origin:

```
ADMIN@DESKTOP-V6G60VF MINGW64 ~/github (main)
$ git config --list --show-origin
error: wrong number of arguments, should be 0
usage: git config [<options>]

Config file location
--global          use global config file
--system          use system config file
--local           use repository config file
--worktree        use per-worktree config file
-f, --file <file> use given config file
--blob <blob-id>  read config from given blob object

Action
--get             get value: name [value-pattern]
--get-all        get all values: key [value-pattern]
--get-regexp      get values for regexp: name-regex [value-pattern]
--get-urlmatch    get value specific for the URL: section[.var] URL
--replace-all    replace all matching variables: name value [value-pattern]
--add             add a new variable: name value
--unset           remove a variable: name [value-pattern]
--unset-all      remove all matches: name [value-pattern]
--rename-section  rename section: old-name new-name
--remove-section  remove a section: name
-l, --list        list all
--fixed-value     use string equality when comparing values to 'value-pattern'
-e, --edit        open an editor
--get-color       find the color configured: slot [default]
--get-colorbool   find the color setting: slot [stdout-is-tty]

Type
-t, --type <>    value is given this type
--bool           value is "true" or "false"
--int            value is decimal number
--bool-or-int    value is --bool or --int
--bool-or-str    value is --bool or string
--path           value is a path (file or directory name)
--expiry-date    value is an expiry date

Other
-z, --null       terminate values with NUL byte
--name-only      show variable names only
--includes       respect include directives on lookup
--show-origin    show origin of config (file, standard input, blob, command line)
--show-scope     show scope of config (worktree, local, global, system, command)
--default <value> with --get, use default value when missing entry
```