

Creating a repo named “Cognizance” with git Bash: -

```
MINGW32/c/Users/user/Cognizance

user@hackers-device MINGW32 ~ (master)
$ mkdir Cognizance

user@hackers-device MINGW32 ~ (master)
$ cd Co
Cognizance/ Contacts/ Cookies/

user@hackers-device MINGW32 ~ (master)
$ cd Cognizance/

user@hackers-device MINGW32 ~/Cognizance (master)
$ git init
Initialized empty Git repository in C:/Users/user/Cognizance/.git/

user@hackers-device MINGW32 ~/Cognizance (master)
$
user@hackers-device MINGW32 ~/Cognizance (master)
$ touch readme.md

user@hackers-device MINGW32 ~/Cognizance (master)
$ git commit -m "enter commit"
On branch master

Initial commit

Untracked files:
  (use "git add <files>..." to include in what will be committed)
  readme.md

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

user@hackers-device MINGW32 ~/Cognizance (master)
$ git remote add origin git@github.com:Gaurav17524/Cognizance.git

user@hackers-device MINGW32 ~/Cognizance (master)
$ curl -u Gaurav17524:ghu_V... -XPUT https://api.github.com/user/repos -d '{"name":"Cognizance"}'
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 21 0 0 100 21 0 15 0:00:01 0:00:01 --:--:-- 15{
{"id":459283165,
 "node_id":"R_kgDOG2Aa3Q",
 "name":"Cognizance",
 "full_name":"Gaurav17524/Cognizance",
 "private":false,
 "owner":{"
 "login":"Gaurav17524",
 "id":66777671,
 "node_id":"MDQ6VXN1cjY2Nzc3Njcxc",
 "avatar_url":"https://avatars.githubusercontent.com/u/66777671?v=4",
 "gravatar_id":"",
 "url":"https://api.github.com/users/Gaurav17524",
 "html_url":"https://github.com/Gaurav17524",
 "followers_url":"https://api.github.com/users/Gaurav17524/followers",
 "following_url":"https://api.github.com/users/Gaurav17524/following{/other_user}",
 "gists_url":"https://api.github.com/users/Gaurav17524/gists{/gist_id}",
 "starred_url":"https://api.github.com/users/Gaurav17524/starred{/owner}{/repo}",
 "subscriptions_url":"https://api.github.com/users/Gaurav17524/subscriptions",
 "organizations_url":"https://api.github.com/users/Gaurav17524/orgs",
 "repos_url":"https://api.github.com/users/Gaurav17524/repos",
 "events_url":"https://api.github.com/users/Gaurav17524/events{/privacy}",
 "received_events_url":"https://api.github.com/users/Gaurav17524/received_events",
 "type":"User",
 "site_admin":false
 },
 "html_url":"https://github.com/Gaurav17524/Cognizance",
 "description":null,
 "fork":false,
 "url":"https://api.github.com/repos/Gaurav17524/Cognizance",
 "forks_url":"https://api.github.com/repos/Gaurav17524/Cognizance/forks",
 "keys_url":"https://api.github.com/repos/Gaurav17524/Cognizance/keys{/key_id}",
 "collaborators_url":"https://api.github.com/repos/Gaurav17524/Cognizance/collaborators{/collaborator}",
 "teams_url":"https://api.github.com/repos/Gaurav17524/Cognizance/teams",
 "hooks_url":"https://api.github.com/repos/Gaurav17524/Cognizance/hooks",
 "issue_events_url":"https://api.github.com/repos/Gaurav17524/Cognizance/issues/events{/number}",
 "events_url":"https://api.github.com/repos/Gaurav17524/Cognizance/events",
 "assignees_url":"https://api.github.com/repos/Gaurav17524/Cognizance/assignees{/user}",
 "branches_url":"https://api.github.com/repos/Gaurav17524/Cognizance/branches{/branch}",
 "tags_url":"https://api.github.com/repos/Gaurav17524/Cognizance/tags",
 "blobs_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/blobs{/sha}",
 "git_tags_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/tags{/sha}",
 "git_refs_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/refs{/sha}",
 "trees_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/trees{/sha}",
 "statuses_url":"https://api.github.com/repos/Gaurav17524/Cognizance/statuses{/sha}",
 "languages_url":"https://api.github.com/repos/Gaurav17524/Cognizance/languages",
 "stargazers_url":"https://api.github.com/repos/Gaurav17524/Cognizance/stargazers",
 "contributors_url":"https://api.github.com/repos/Gaurav17524/Cognizance/contributors",
 "subscribers_url":"https://api.github.com/repos/Gaurav17524/Cognizance/subscribers",
 "subscription_url":"https://api.github.com/repos/Gaurav17524/Cognizance/subscription",
 "commits_url":"https://api.github.com/repos/Gaurav17524/Cognizance/commits{/sha}",
 "git_commits_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/commits{/sha}",
 "comments_url":"https://api.github.com/repos/Gaurav17524/Cognizance/comments{/number}",
```

```
MINGW32/c/Users/user/Cognizance

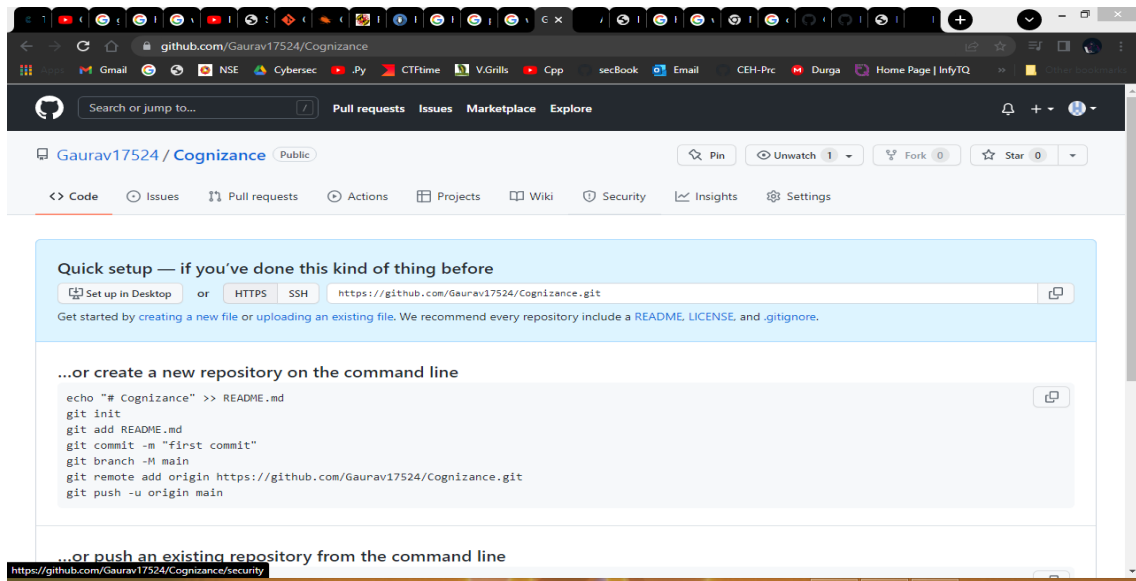
{"id":459283165,
 "node_id":"R_kgDOG2Aa3Q",
 "name":"Cognizance",
 "full_name":"Gaurav17524/Cognizance",
 "private":false,
 "owner":{"
 "login":"Gaurav17524",
 "id":66777671,
 "node_id":"MDQ6VXN1cjY2Nzc3Njcxc",
 "avatar_url":"https://avatars.githubusercontent.com/u/66777671?v=4",
 "gravatar_id":"",
 "url":"https://api.github.com/users/Gaurav17524",
 "html_url":"https://github.com/Gaurav17524",
 "followers_url":"https://api.github.com/users/Gaurav17524/followers",
 "following_url":"https://api.github.com/users/Gaurav17524/following{/other_user}",
 "gists_url":"https://api.github.com/users/Gaurav17524/gists{/gist_id}",
 "starred_url":"https://api.github.com/users/Gaurav17524/starred{/owner}{/repo}",
 "subscriptions_url":"https://api.github.com/users/Gaurav17524/subscriptions",
 "organizations_url":"https://api.github.com/users/Gaurav17524/orgs",
 "repos_url":"https://api.github.com/users/Gaurav17524/repos",
 "events_url":"https://api.github.com/users/Gaurav17524/events{/privacy}",
 "received_events_url":"https://api.github.com/users/Gaurav17524/received_events",
 "type":"User",
 "site_admin":false
 },
 "html_url":"https://github.com/Gaurav17524/Cognizance",
 "description":null,
 "fork":false,
 "url":"https://api.github.com/repos/Gaurav17524/Cognizance",
 "forks_url":"https://api.github.com/repos/Gaurav17524/Cognizance/forks",
 "keys_url":"https://api.github.com/repos/Gaurav17524/Cognizance/keys{/key_id}",
 "collaborators_url":"https://api.github.com/repos/Gaurav17524/Cognizance/collaborators{/collaborator}",
 "teams_url":"https://api.github.com/repos/Gaurav17524/Cognizance/teams",
 "hooks_url":"https://api.github.com/repos/Gaurav17524/Cognizance/hooks",
 "issue_events_url":"https://api.github.com/repos/Gaurav17524/Cognizance/issues/events{/number}",
 "events_url":"https://api.github.com/repos/Gaurav17524/Cognizance/events",
 "assignees_url":"https://api.github.com/repos/Gaurav17524/Cognizance/assignees{/user}",
 "branches_url":"https://api.github.com/repos/Gaurav17524/Cognizance/branches{/branch}",
 "tags_url":"https://api.github.com/repos/Gaurav17524/Cognizance/tags",
 "blobs_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/blobs{/sha}",
 "git_tags_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/tags{/sha}",
 "git_refs_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/refs{/sha}",
 "trees_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/trees{/sha}",
 "statuses_url":"https://api.github.com/repos/Gaurav17524/Cognizance/statuses{/sha}",
 "languages_url":"https://api.github.com/repos/Gaurav17524/Cognizance/languages",
 "stargazers_url":"https://api.github.com/repos/Gaurav17524/Cognizance/stargazers",
 "contributors_url":"https://api.github.com/repos/Gaurav17524/Cognizance/contributors",
 "subscribers_url":"https://api.github.com/repos/Gaurav17524/Cognizance/subscribers",
 "subscription_url":"https://api.github.com/repos/Gaurav17524/Cognizance/subscription",
 "commits_url":"https://api.github.com/repos/Gaurav17524/Cognizance/commits{/sha}",
 "git_commits_url":"https://api.github.com/repos/Gaurav17524/Cognizance/git/commits{/sha}",
 "comments_url":"https://api.github.com/repos/Gaurav17524/Cognizance/comments{/number}",
```

```
MINGW32/c/Users/user/Cognizance
{
  "subscription_url": "https://api.github.com/repos/Gaurav17524/Cognizance/subscription",
  "commits_url": "https://api.github.com/repos/Gaurav17524/Cognizance/commits{/sha}",
  "git_commits_url": "https://api.github.com/repos/Gaurav17524/Cognizance/git/commits{/sha}",
  "comments_url": "https://api.github.com/repos/Gaurav17524/Cognizance/comments{/number}",
  "issue_comments_url": "https://api.github.com/repos/Gaurav17524/Cognizance/issues/comments{/number}",
  "contents_url": "https://api.github.com/repos/Gaurav17524/Cognizance/contents/{+path}",
  "compare_url": "https://api.github.com/repos/Gaurav17524/Cognizance/compare/{base}...{head}",
  "merges_url": "https://api.github.com/repos/Gaurav17524/Cognizance/merges",
  "archive_url": "https://api.github.com/repos/Gaurav17524/Cognizance/{archive_format}/{ref}",
  "downloads_url": "https://api.github.com/repos/Gaurav17524/Cognizance/downloads",
  "issues_url": "https://api.github.com/repos/Gaurav17524/Cognizance/issues{/number}",
  "pulls_url": "https://api.github.com/repos/Gaurav17524/Cognizance/pulls{/number}",
  "notifications_url": "https://api.github.com/repos/Gaurav17524/Cognizance/notifications?<since>,{all,participating}",
  "labels_url": "https://api.github.com/repos/Gaurav17524/Cognizance/labels/{name}",
  "releases_url": "https://api.github.com/repos/Gaurav17524/Cognizance/releases/{id}",
  "deployments_url": "https://api.github.com/repos/Gaurav17524/Cognizance/deployments",
  "created_at": "2022-02-14T18:41:19Z",
  "updated_at": "2022-02-14T18:41:19Z",
  "pushed_at": "2022-02-14T18:41:19Z",
  "git_url": "git://github.com/Gaurav17524/Cognizance.git",
  "ssh_url": "git@github.com:Gaurav17524/Cognizance.git",
  "clone_url": "https://github.com/Gaurav17524/Cognizance.git",
  "svn_url": "https://github.com/Gaurav17524/Cognizance",
  "homepage": null,
  "size": 0,
  "stargazers_count": 0,
  "watchers_count": 0,
  "language": null,
  "has_issues": true,
  "has_projects": true,
  "has_downloads": true,
  "has_wiki": true,
  "has_pages": false,
  "forks_count": 0,
  "mirror_url": null,
  "archived": false,
  "disabled": false,
  "open_issues_count": 0,
  "license": null,
  "allow_forking": true,
  "is_template": false,
  "topics": [
  ],
  "visibility": "public",
  "forks": 0,
  "open_issues": 0,
  "watchers": 0,
  "default_branch": "main",
  "permissions": {
    "admin": true,
  },
}
```

```
MINGW32/c/Users/user/Cognizance
{
  "updated_at": "2022-02-14T18:41:19Z",
  "pushed_at": "2022-02-14T18:41:19Z",
  "git_url": "git://github.com/Gaurav17524/Cognizance.git",
  "ssh_url": "git@github.com:Gaurav17524/Cognizance.git",
  "clone_url": "https://github.com/Gaurav17524/Cognizance.git",
  "svn_url": "https://github.com/Gaurav17524/Cognizance",
  "homepage": null,
  "size": 0,
  "stargazers_count": 0,
  "watchers_count": 0,
  "language": null,
  "has_issues": true,
  "has_projects": true,
  "has_downloads": true,
  "has_wiki": true,
  "has_pages": false,
  "forks_count": 0,
  "mirror_url": null,
  "archived": false,
  "disabled": false,
  "open_issues_count": 0,
  "license": null,
  "allow_forking": true,
  "is_template": false,
  "topics": [
  ],
  "visibility": "public",
  "forks": 0,
  "open_issues": 0,
  "watchers": 0,
  "default_branch": "main",
  "permissions": {
    "admin": true,
    "maintain": true,
    "push": true,
    "triage": true,
    "pull": true
  },
  "allow_squash_merge": true,
  "allow_merge_commit": true,
  "allow_rebase_merge": true,
  "allow_auto_merge": false,
  "delete_branch_on_merge": false,
  "allow_update_branch": false,
  "network_count": 0,
  "subscribers_count": 1
}

user@hackers-device MINGW32 ~/Cognizance (master)
$
```

Repository named “Cognizance” was created with the command line: -



Task 2: -

Create a folder with name "Task-1" in the repository and perform the Git operations using Git Bash

Creating a file in the main repository: -

```
MINGW32/c/Users/user/Cognizance
user@hackers-device MINGW32 ~ (master)
$ ls
AppData/ Downloads/ NTUSER.DAT{fcc73ed7-0b03-11e3-9710-88353a1cb0c0}.TM.b1f README.md Videos/
'Application Data'@ Favorites/ NTUSER.DAT{fcc73ed7-0b03-11e3-9710-88353a1cb0c0}.TMContainer000000000000000001.regtrans-ms Recent@ 'VirtualBox VMs'/
Cognizance/ Links/ NTUSER.DAT{fcc73ed7-0b03-11e3-9710-88353a1cb0c0}.TMContainer000000000000000002.regtrans-ms 'Saved Games' new.html
Contacts/ 'Local Settings'@ NetHood@ Pictures/ Searches/ ntuser.dat.LOG1
Cookies@ Music/ PrintHood@ SendTo@ Start Menu@ ntuser.dat.LOG2
Desktop/ 'My Documents'@ PycharmProjects/ Templates@ testing_repo/
Documents/ NTUSER.DAT

user@hackers-device MINGW32 ~ (master)
$ cd Cognizance

user@hackers-device MINGW32 ~/Cognizance (master)
$ https://github.com/Gaurav17524/Cognizance.git
bash: https://github.com/Gaurav17524/Cognizance.git: No such file or directory

user@hackers-device MINGW32 ~/Cognizance (master)
$ ls -lart
total 20
drwxr-xr-x 1 user 197121 0 Feb 15 00:20 ./
-rw-r--r-- 1 user 197121 0 Feb 15 00:22 README.md
drwxr-xr-x 1 user 197121 0 Feb 15 00:33 Task-1/
drwxr-xr-x 1 user 197121 0 Feb 15 00:33 ./
drwxr-xr-x 1 user 197121 0 Feb 15 00:35 .git/

user@hackers-device MINGW32 ~/Cognizance (master)
$ code .

user@hackers-device MINGW32 ~/Cognizance (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  index.html
  index.py
  readme.md

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

user@hackers-device MINGW32 ~/Cognizance (master)
$ git add -A

user@hackers-device MINGW32 ~/Cognizance (master)
$ git status
On branch master

No commits yet
```

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

user@hackers-device MINGW32 ~/Cognizance (master)
$ git add -A

user@hackers-device MINGW32 ~/Cognizance (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
        new file:   index.py
        new file:   readme.md

user@hackers-device MINGW32 ~/Cognizance (master)
$ git commit
[master (root-commit) f648241] Initial commit
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
create mode 100644 index.py
create mode 100644 readme.md

user@hackers-device MINGW32 ~/Cognizance (master)
$ git remote add origin https://github.com/Gaurav17524/Cognizance.git
git branch -M main
git push -u origin main
error: remote origin already exists.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 231 bytes | 10.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Gaurav17524/Cognizance.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

user@hackers-device MINGW32 ~/Cognizance (main)
$ track Task-1
bash: track: command not found

user@hackers-device MINGW32 ~/Cognizance (main)
$ git add Task-1

user@hackers-device MINGW32 ~/Cognizance (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

Making some dummy files inside the Task-1 folder through git bash

```
user@hackers-device MINGW32 ~/Cognizance (main)
$ touch Task-1

user@hackers-device MINGW32 ~/Cognizance (main)
$ git init
Reinitialized existing Git repository in C:/Users/user/Cognizance/.git/

user@hackers-device MINGW32 ~/Cognizance (main)
$ git add Task-1
warning: LF will be replaced by CRLF in Task-1/index.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in Task-1/readme.txt.
The file will have its original line endings in your working directory

user@hackers-device MINGW32 ~/Cognizance (main)
$ git branch --move master main
fatal: a branch named 'main' already exists

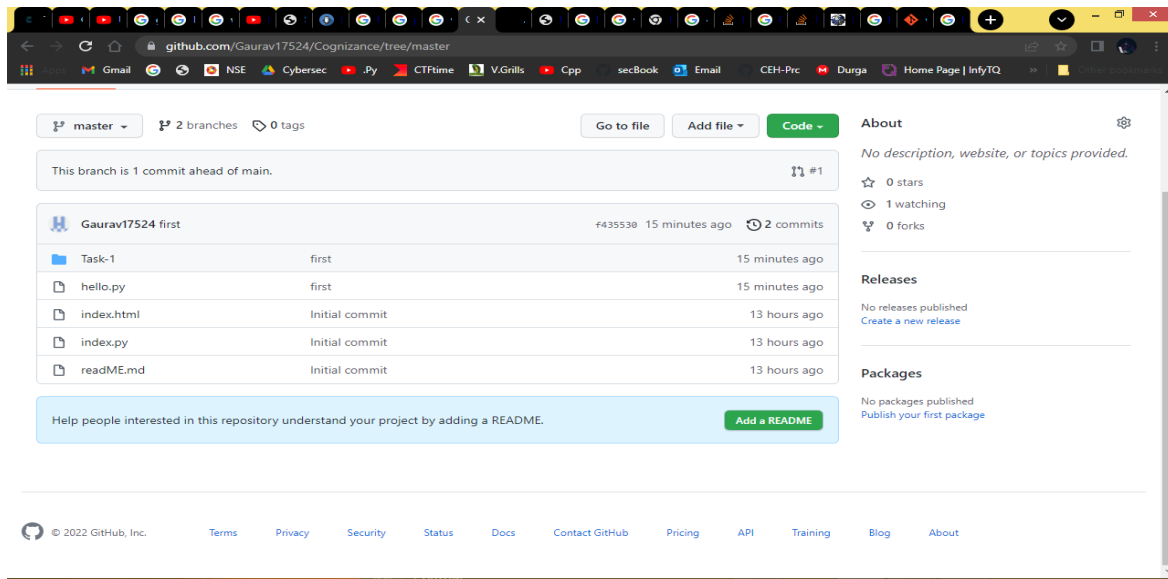
user@hackers-device MINGW32 ~/Cognizance (main)
$ git branch --move master

user@hackers-device MINGW32 ~/Cognizance (master)
$ git commit -m "First"
[master f435530] First
4 files changed, 4 insertions(+)
create mode 100644 Task-1/hello.css
create mode 100644 Task-1/index.js
create mode 100644 Task-1/readme.txt
create mode 100644 hello.py

user@hackers-device MINGW32 ~/Cognizance (master)
$ git remote add origin https://github.com/Gaurav17524/Cognizance.git
error: remote origin already exists.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 492 bytes | 246.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for "master" on GitHub by visiting:
remote:   https://github.com/Gaurav17524/Cognizance/pull/new/master
remote:
To github.com:Gaurav17524/Cognizance.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

"Task-1" folder was made in the GitHub repo named "Cognizance"



Some files were made inside task-1 folder: -

```
MINGW32/c/Users/user/Cognizance
user@hackers-device MINGW32 ~/Cognizance (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
user@hackers-device MINGW32 ~/Cognizance (main)
$ git remote add origin https://github.com/Gaurav17524/Cognizance.git
git branch -M main
git push -u origin main
error: remote origin already exists.
Everything up-to-date
branch 'main' set up to track 'origin/main'.
user@hackers-device MINGW32 ~/Cognizance (main)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .' ?
hint: Turn this message off by running
hint: "git config advice.addEmptyPathspec False"
user@hackers-device MINGW32 ~/Cognizance (main)
$ git add Task-1
user@hackers-device MINGW32 ~/Cognizance (main)
$ ls
Task-1/ index.html index.py readME.md
user@hackers-device MINGW32 ~/Cognizance (main)
$ cd Task-1
user@hackers-device MINGW32 ~/Cognizance/Task-1 (main)
$ touch readme.txt
user@hackers-device MINGW32 ~/Cognizance/Task-1 (main)
$ echo >> readme.txt
user@hackers-device MINGW32 ~/Cognizance/Task-1 (main)
$ echo >> readme.txt
user@hackers-device MINGW32 ~/Cognizance/Task-1 (main)
$ cat > readme.txt
hello,this is readme....please readme
wq
:wq
user@hackers-device MINGW32 ~/Cognizance/Task-1 (main)
$ git status
On branch main
```

Task 3: -

Upload the screenshot of the git bash using This command in your computer.

```
MINGW32/c/Users/user/Cognizance
user@hackers-device MINGW32 ~/Cognizance (master)
$ git config --list --show
error: ambiguous option: show- (could be --show-origin or --show-scope)
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-atty]

Type
-t, --type <v>  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

user@hackers-device MINGW32 ~/Cognizance (master)
$ |
```

Task 4:-

Some files were pushed for staging: -

```
MINGW32/c/Users/user/Cognizance
$ git remote add origin https://github.com/Gaurav17524/Cognizance.git
error: remote origin already exists.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git status
On branch master
Your branch is up to date with 'origin/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:   Task-1/hello.css

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

user@hackers-device MINGW32 ~/Cognizance (master)
$ git add -A
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Task-1/hello.css

user@hackers-device MINGW32 ~/Cognizance (master)
$ git commit -a -m "second edit"
[master 03200a3] second edit
1 file changed, 2 insertions(+), 1 deletion(-)

user@hackers-device MINGW32 ~/Cognizance (master)
$ push -u origin master
bash: push: command not found

user@hackers-device MINGW32 ~/Cognizance (master)
$ git push -u origin master
Enumerating objects: 2, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 431 bytes | 431.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Gaurav17524/Cognizance.git
   f435530..03200a3  master -> master
branch 'master' set up to track 'origin/master'.

user@hackers-device MINGW32 ~/Cognizance (master)
$ |
```

```
MINGW32:/c/Users/user/Cognizance/Task-1
bash: push: command not found

user@hackers-device MINGW32 ~/Cognizance (master)
$ git push -u origin master
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 431 bytes | 431.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:Gaurav17524/Cognizance.git
   f435530..032d0a1  master -> master
branch 'master' set up to track 'origin/master'.

user@hackers-device MINGW32 ~/Cognizance (master)
$ ls
Task-1/ hello.py index.html index.py readme.md

user@hackers-device MINGW32 ~/Cognizance (master)
$ code .

user@hackers-device MINGW32 ~/Cognizance (master)
$ ls
Task-1/ hello.py index.html index.py readme.md

user@hackers-device MINGW32 ~/Cognizance (master)
$ cd Task-1

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ ls
hello.css index.js readme.txt

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ code .

user@hackers-device MINGW32 ~/Cognizance/Task-1 (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:   hello.css

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

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ git checkout hello.css
Updated 1 path from the index

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ |
```

Git checkout: - Replace the current working file with the file of the branch

```
MINGW32:/c/Users/user/Cognizance/Task-1

-r, --remotes          act on remote-tracking branches
--contains <commit>   print only branches that contain the commit
--no-contains <commit>
                        print only branches that don't contain the commit
--abbrev[=<n>]         use <n> digits to display object names

Specific git-branch actions:
-a, --all              list both remote-tracking and local branches
-d, --delete           delete fully merged branch
-D                    delete branch (even if not merged)
-m, --move            move/rename a branch and its reflog
-M                    move/rename a branch, even if target exists
-c, --copy            copy a branch and its reflog
-C                    copy a branch, even if target exists
-l, --list             list branch names
--show-current         show current branch name
--create-reflog        create the branch's reflog
--edit-description    edit the description for the branch
-f, --force           force creation, move/rename, deletion
--merged <commit>     print only branches that are merged
--no-merged <commit>  print only branches that are not merged
--column[=<style>]    list branches in columns
--sort <key>          field name to sort on
--points-at <object>  print only branches of the object
-l, --ignore-case     sorting and filtering are case insensitive
--format <format>     format to use for the output

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ git branch testbranch

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ git branch
* master
  testbranch

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ git checkout testbranch
Switched to branch 'testbranch'
M   hello.css
M   index.js
M   readme.txt

user@hackers-device MINGW32 ~/Cognizance/Task-1 (testbranch)
$ git checkout master
Switched to branch 'master'
M   hello.css
M   index.js
M   readme.txt

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ |
```

```
MINGW32:/c/Users/user/Cognizance

user@hackers-device MINGW32 ~/Cognizance (master)
$ ls
Task-1/ hello.py index.html index.py readme.md

user@hackers-device MINGW32 ~/Cognizance (master)
$ git pull
Updating 032d0a1..7317331
Fast-forward
   readme.md | 2 ++
   test.html | 12 ++++++++
   2 files changed, 14 insertions(+)
   create mode 100644 test.html

user@hackers-device MINGW32 ~/Cognizance (master)
$ ls
Task-1/ hello.py index.html index.py readme.md test.html

user@hackers-device MINGW32 ~/Cognizance (master)
$ git checkout contact
error: pathspec 'contact' did not match any file(s) known to git

user@hackers-device MINGW32 ~/Cognizance (master)
$ git fetch contact
fatal: 'contact' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git checkout
M   Task-1/hello.css
M   Task-1/index.js
M   Task-1/readme.txt
Your branch is up to date with 'origin/master'.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 615 bytes | 3.00 KiB/s, done.
From github.com:Gaurav17524/Cognizance
+ [new branch]      contact -> origin/contact

user@hackers-device MINGW32 ~/Cognizance (master)
$ git checkout
M   Task-1/hello.css
M   Task-1/index.js
M   Task-1/readme.txt
```

```
fatal: Could not read from remote repository.
Please make sure you have the correct access rights
and the repository exists.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git checkout
M Task-1/hello.css
M Task-1/index.js
M Task-1/readme.txt
Your branch is up to date with 'origin/master'.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git fetch origin
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 615 bytes | 3.00 KiB/s, done.
From github.com:Gaurav7224/Cognizance
+ [new branch] contact -> origin/contact

user@hackers-device MINGW32 ~/Cognizance (master)
$ git checkout
M Task-1/hello.css
M Task-1/index.js
M Task-1/readme.txt
Your branch is up to date with 'origin/master'.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git branch
* master

user@hackers-device MINGW32 ~/Cognizance (master)
$ git checkout contact
Switched to a new branch 'contact'
M Task-1/hello.css
M Task-1/index.js
M Task-1/readme.txt
branch 'contact' set up to track 'origin/contact'.

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git branch
* contact
  master

user@hackers-device MINGW32 ~/Cognizance (contact)
$ ls
Task-1/ contact.html hello.py index.html index.py readme.md test.html

user@hackers-device MINGW32 ~/Cognizance (contact)
$ |
```

Git merge: -

Git merge will combine multiple sequences of commits into one unified history

```
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 615 bytes | 3.00 KiB/s, done.
From github.com:Gaurav7224/Cognizance
+ [new branch] contact -> origin/contact

user@hackers-device MINGW32 ~/Cognizance (master)
$ git checkout
M Task-1/hello.css
M Task-1/index.js
M Task-1/readme.txt
Your branch is up to date with 'origin/master'.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git branch
* master

user@hackers-device MINGW32 ~/Cognizance (master)
$ git checkout contact
Switched to a new branch 'contact'
M Task-1/hello.css
M Task-1/index.js
M Task-1/readme.txt
branch 'contact' set up to track 'origin/contact'.

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git branch
* contact
  master

user@hackers-device MINGW32 ~/Cognizance (contact)
$ ls
Task-1/ contact.html hello.py index.html index.py readme.md test.html

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git checkout master
Switched to branch 'master'
M Task-1/hello.css
M Task-1/index.js
M Task-1/readme.txt
Your branch is up to date with 'origin/master'.

user@hackers-device MINGW32 ~/Cognizance (master)
$ git merge contact
Updating 7317331..4cf4100
Fast-forward
 contact.html | 12 ++++++
 1 files changed, 12 insertions(+)
 create mode 100644 contact.html

user@hackers-device MINGW32 ~/Cognizance (master)
$ |
```

Git rebase: -

Rebase is one of two Git utilities that specializes in integrating changes from one branch onto another. ... The other change integration utility is git merge. Merge is always a forward moving change record. Alternatively, rebase has powerful history rewriting features.


```
MINGW32/c/Users/user/Cognizance/Task-1
[contact 290616b] rebase test
4 files changed, 5 insertions(+), 1 deletion(-)
create mode 100644 bl.txt

user@hackers-device MINGW32 ~/Cognizance (contact)
$ ls
Task-1/ bl.txt contact.html hello.py index.html index.py readME.md test.html

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git rebase master
Current branch contact is up to date.

user@hackers-device MINGW32 ~/Cognizance (contact)
$ ls
Task-1/ bl.txt contact.html hello.py index.html index.py readME.md test.html

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git branch test

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git checkout test
Switched to branch 'test'

user@hackers-device MINGW32 ~/Cognizance (test)
$ touch bl.txt

user@hackers-device MINGW32 ~/Cognizance (test)
$ git add -A

user@hackers-device MINGW32 ~/Cognizance (test)
$ git commit -m "rebasng"
[test 752c9dc] rebasing
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Task-1/bl.txt

user@hackers-device MINGW32 ~/Cognizance (test)
$ ls
Task-1/ bl.txt contact.html hello.py index.html index.py readME.md test.html

user@hackers-device MINGW32 ~/Cognizance (test)
$ git rebase master
Current branch test is up to date.

user@hackers-device MINGW32 ~/Cognizance (test)
$ cd Task-1

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ ls
bl.txt hello.css iindex.js readme.txt

user@hackers-device MINGW32 ~/Cognizance/Task-1 (master)
$ |
```

Git Commit: -

The git commit command captures a snapshot of the project's currently staged changes.

```
MINGW32/c/Users/user/Cognizance

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git log --oneline
ebee1c1f (HEAD -> contact, origin/contact) resetting
290616b rebase test
4cf4100 (origin/master, master) testing fetching
7317331 test fetching
0e2d25f test fetch
558b293 second commit
13ad74f Update readME.md
032d0a1 second edit
f435530 first
f648241 Initial commit

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git reset --soft HEAD-1

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git status
On branch contact
Your branch is behind 'origin/contact' by 1 commit, and can be fast-forwarded.
(use 'git pull' to update your local branch)

Changes to be committed:
  (use 'git restore --staged <file>...' to unstage)
    modified:   contact.html

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git reset ebee1c1f

user@hackers-device MINGW32 ~/Cognizance (contact)
$ git status
On branch contact
Your branch is up to date with 'origin/contact'.

nothing to commit, working tree clean

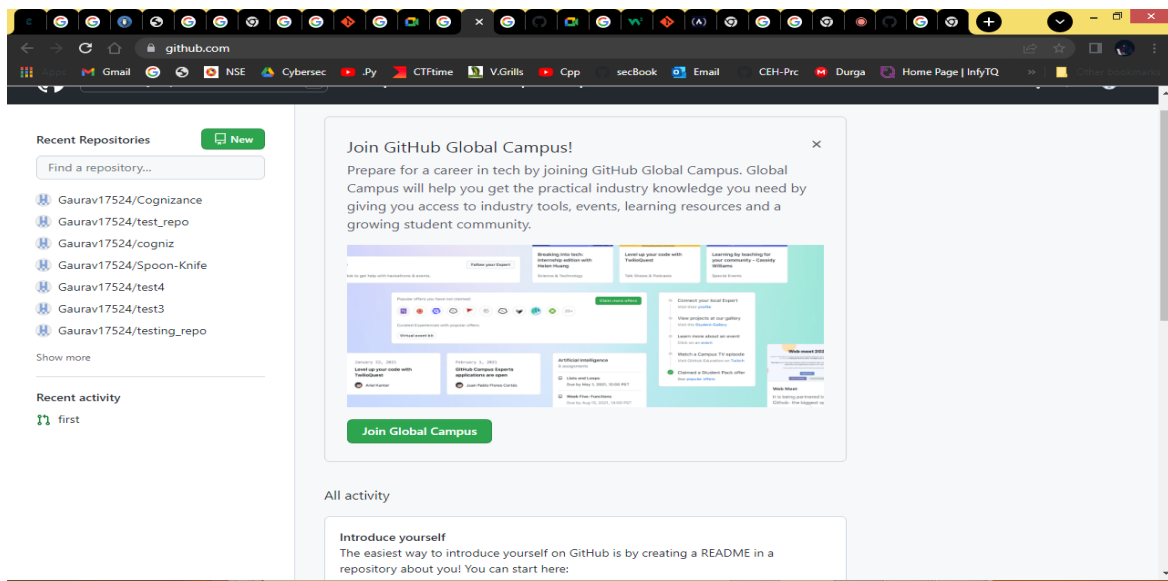
user@hackers-device MINGW32 ~/Cognizance (contact)
$ git commit
$ git commit
On branch contact
Your branch is up to date with 'origin/contact'.

nothing to commit, working tree clean

user@hackers-device MINGW32 ~/Cognizance (contact)
$ |
```

Forking the git repository

A GitHub fork is a copy of a repository (repo) that sits in your account rather than the account from which you forked the data from.



Git clone: -

git clone is a Git command line utility which is used to target an existing repository and create a clone, or copy of the target repository. ... Cloning a local or remote repository.

```
user@hackers-device MINGW32 ~ (master)
$ git clone https://github.com/illuminate/html.git
Cloning into 'html'...
remote: Enumerating objects: 609, done.
remote: Total 609 (delta 0), reused 0 (delta 0), pack-reused 609
Receiving objects: 100% (609/609), 98.18 KiB | 437.00 KiB/s, done.
Resolving deltas: 100% (260/260), done.
user@hackers-device MINGW32 ~ (master)
$
```