


PROBLEM STATEMENT :

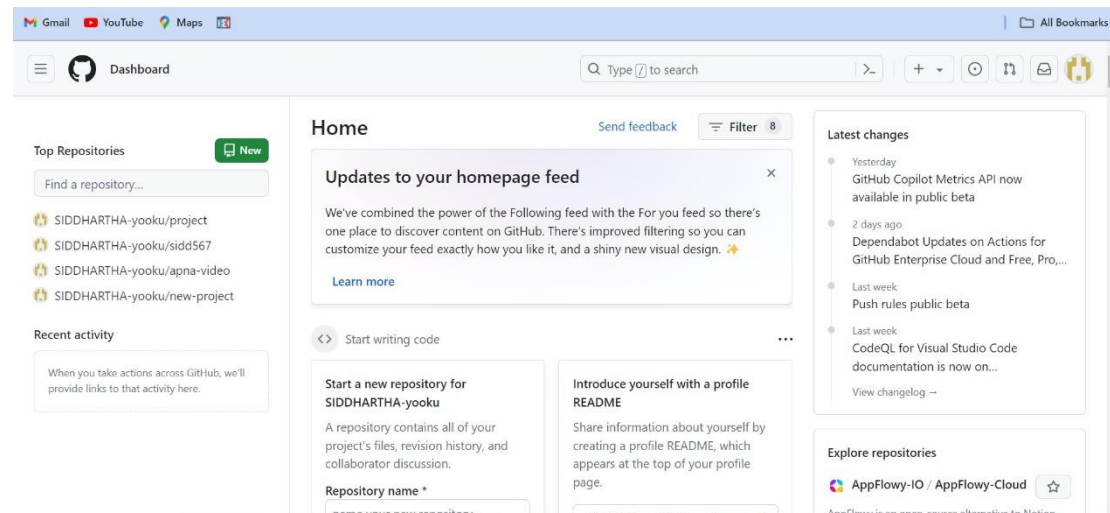
8) Deploy a project from a local machine to GitHub and vice versa.

1. Search for GitHub and sign in or log in if already you have an account with your Username or email address and password.



The image shows the GitHub sign-in page. At the top is the GitHub logo and the text "Sign in to GitHub". Below this is a form with two input fields: "Username or email address" containing "shaw33224@gmail.com" and "Password" with masked characters. There is a "Forgot password?" link next to the password field. A green "Sign in" button is below the fields. At the bottom, there is a link for "Sign in with a passkey" and a link for "New to GitHub? Create an account".

2. Now click on “New”.



3. Under “Create a new repository”, write the “Repository name” and click on “Public” then click on “Create repository”.

Required fields are marked with an asterisk (*).

Owner * SIDDHARTHA-yooku / Repository name * newrepo3
newrepo3 is available.

Great repository names are short and memorable. Need inspiration? How about [literate-chainsaw](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: None
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

4. “newrepo3” is successfully created.

5. Go back to “Home” and click on “Settings”

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: None
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: None
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

① You are creating a public repository in your personal account.

[Create repository](#)

© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

SIDDHARTHA-yooku / newrepo3

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

newrepo3 Public Pin Unwatch

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

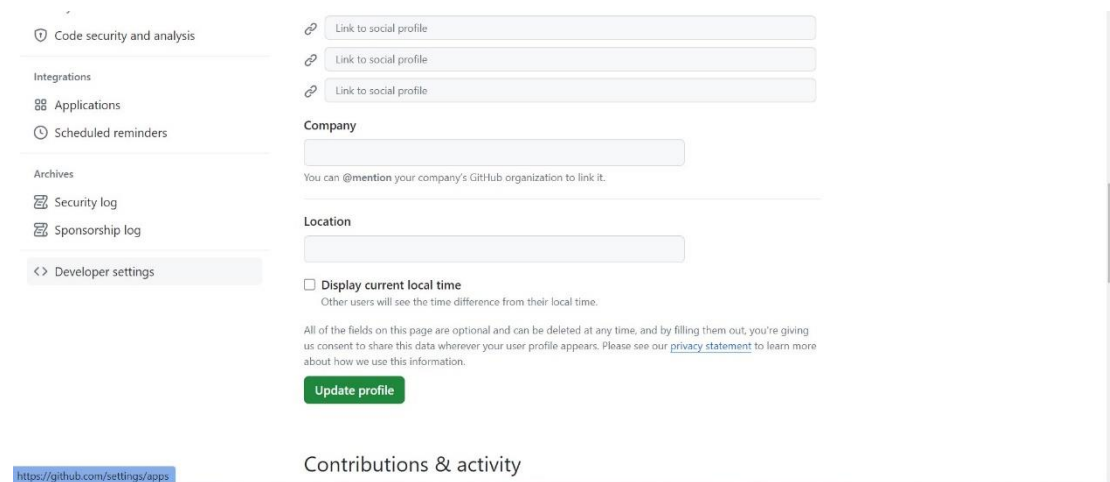
Add collaborators to this repository
Search for people using their GitHub username
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before
[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/SIDDHARTHA-yooku/newrepo3.git>
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

[https://github.com/settings/profile](#)

- SIDDHARTHA-yooku
- Your Copilot
- Your organizations
- Your enterprises
- Your stars
- Your sponsors
- Your gists
- Upgrade
- Try Enterprise Free
- Feature preview
- Settings
- GitHub Support
- GitHub Community
- Sign out

6. Under “Settings”, select “Developer settings”.



The screenshot shows the GitHub Developer Settings page. On the left, a sidebar contains a list of settings categories: Code security and analysis, Integrations, Applications, Scheduled reminders, Archives, Security log, Sponsorship log, and Developer settings (which is highlighted). The main content area is titled 'Developer settings' and includes sections for 'Link to social profile' (three links), 'Company' (a text input field), 'Location' (a text input field), and a checkbox for 'Display current local time'. Below these is a green 'Update profile' button. At the bottom, there is a link to 'Contributions & activity' and a URL bar showing 'https://github.com/settings/apps'.

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

< > Developer settings

Link to social profile

Link to social profile

Link to social profile

Company

You can @mention your company's GitHub organization to link it.

Location

☐ Display current local time
Other users will see the time difference from their local time.

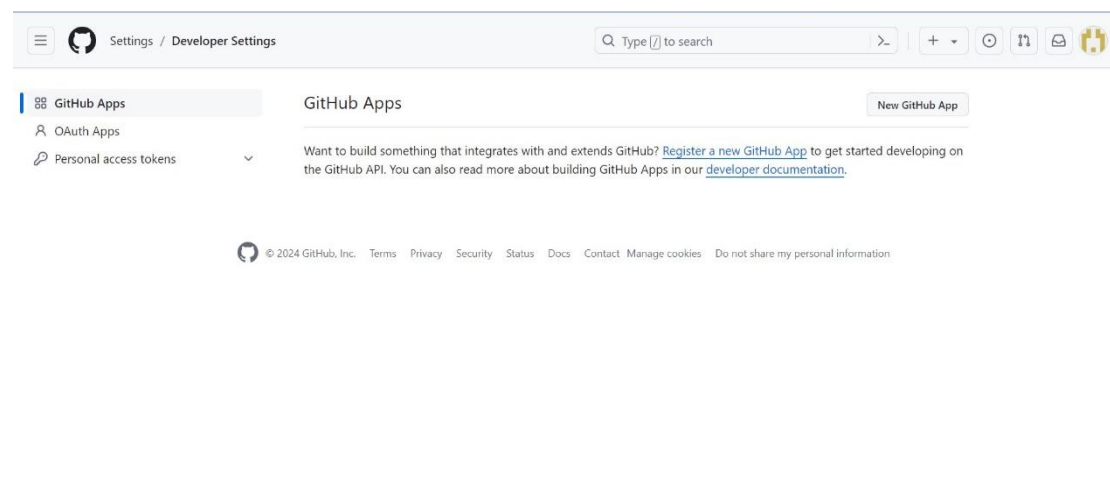
All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Contributions & activity

<https://github.com/settings/apps>

7. After clicking on “Developer settings”, under “Personal access tokens” click on “Tokens(classic)”.



The screenshot shows the GitHub Developer Settings page for 'Personal access tokens'. The sidebar on the left has 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' (which is highlighted). The main content area is titled 'GitHub Apps' and includes a 'New GitHub App' button. Below this is a section for 'Personal access tokens' with a 'Generate new token(classic)' button. The footer contains copyright information for GitHub, Inc. and links to Terms, Privacy, Security, Status, Docs, Contact, Manage cookies, and Do not share my personal information.

Settings / Developer Settings

Type [j] to search

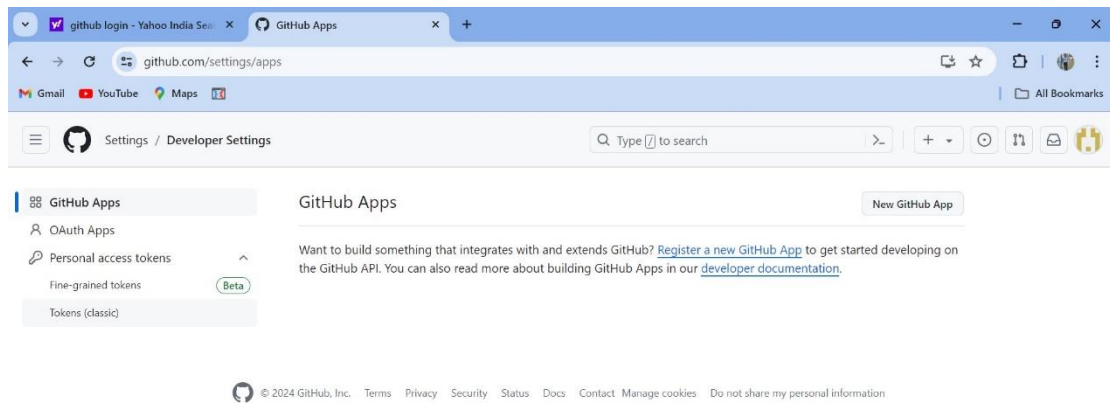
GitHub Apps

New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

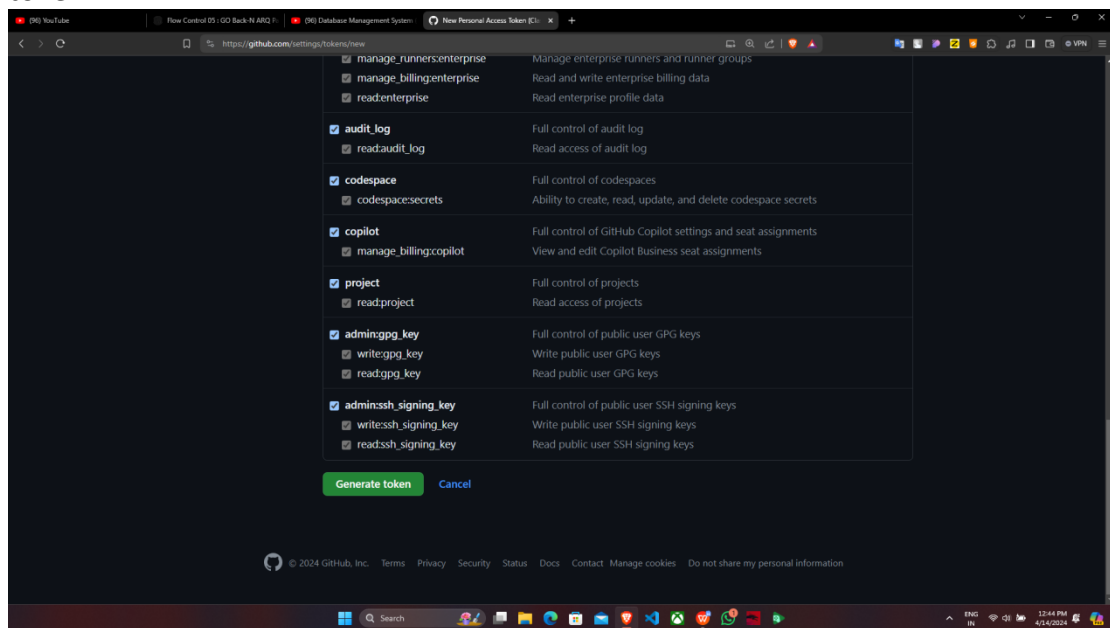
© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

8.Go to “Generate new token”, click on “Generate new token(classic)”.

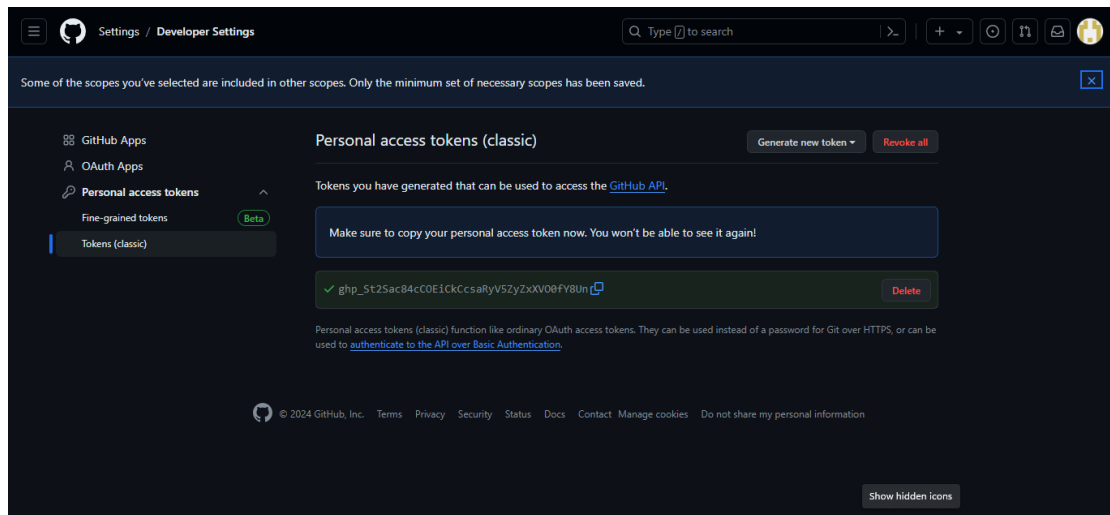


https://github.com/settings/tokens

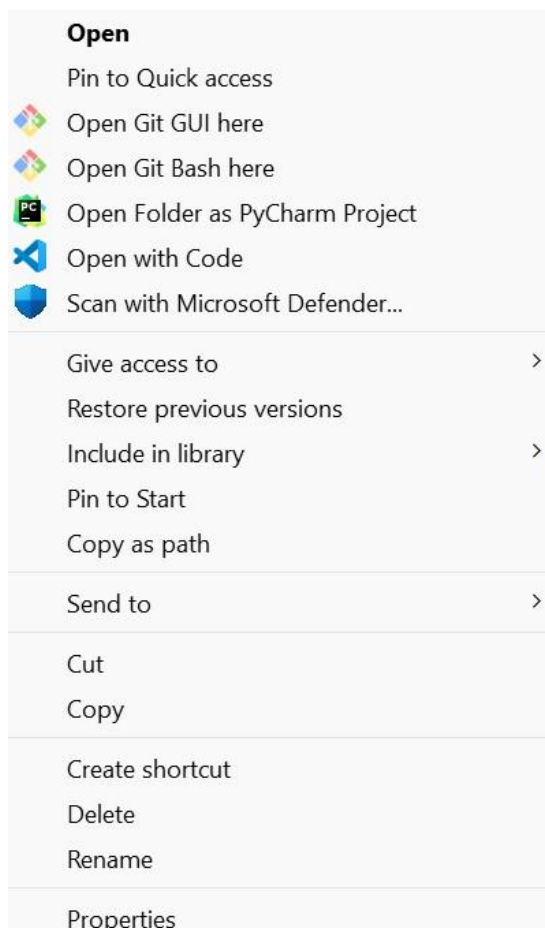
9. Under “New personal access token(classic)”,give the token name ,set the expiration to 90 days and click on all the checkboxes. Then click “Generate token”.



10. Token is generated successfully. Copy the token in notepad.



11. Now create a new folder in your desktop then right click on it and go to “Show more options” then select “Git Bash Here”.



12. Type the following codes: After this command one window appears, go to “Token” then enter the token and click on “Sign in”.

13. Type the following codes:

```
--[no]-reference-if-able <repo>      reference repository
--[no]-dissociate                    use --reference only while cloning
-o, --[no]-jorigin <name>           use <name> instead of 'origin' to track upstream
-b, --[no]-branch <branch>          checkout <branch> instead of the remote's HEAD
-u, --[no]-upload-pack <path>       path to git-upload-pack on the remote
--[no]-depth <depth>                create a shallow clone of that depth
--[no]-shallow-since <time>          create a shallow clone since a specific time
--[no]-shallow-exclude <revision>    deepen history of shallow clone, excluding rev
--[no]-single-branch                 clone only one branch, HEAD or --branch
--no-tags                           don't clone any tags, and make later fetches not to follow them
--tags                             opposite of --no-tags
--[no]-shallow-submodules            any cloned submodules will be shallow
--[no]-separate-git-dir <gitdir>    separate git dir from working tree
-c, --[no]-config <key=value>       set config inside the new repository
--[no]-server-option <server-specific> option to transmit
-4, --ipv4                          use IPv4 addresses only
-6, --ipv6                          use IPv6 addresses only
--[no]-filter <args>                object filtering
--[no]-also-filter-submodules        apply partial clone filters to submodules
--[no]-remote-submodules             any cloned submodules will use their remote-tracking branch
--[no]-sparse                       initialize sparse-checkout file to include only files at root
--[no]-bundle-uri <uri>             a URI for downloading bundles before fetching from origin remote
```

```
Admin@BL2-066 MINGW64 ~/Desktop/aws project (master)
$ git clone https://github.com/mounta02314/myawsproject1.git
Cloning into 'myawsproject1'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 0), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.
Admin@BL2-066 MINGW64 ~/Desktop/aws project (master)
$
```

```
$ git add .
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git status
On branch master

No commits yet

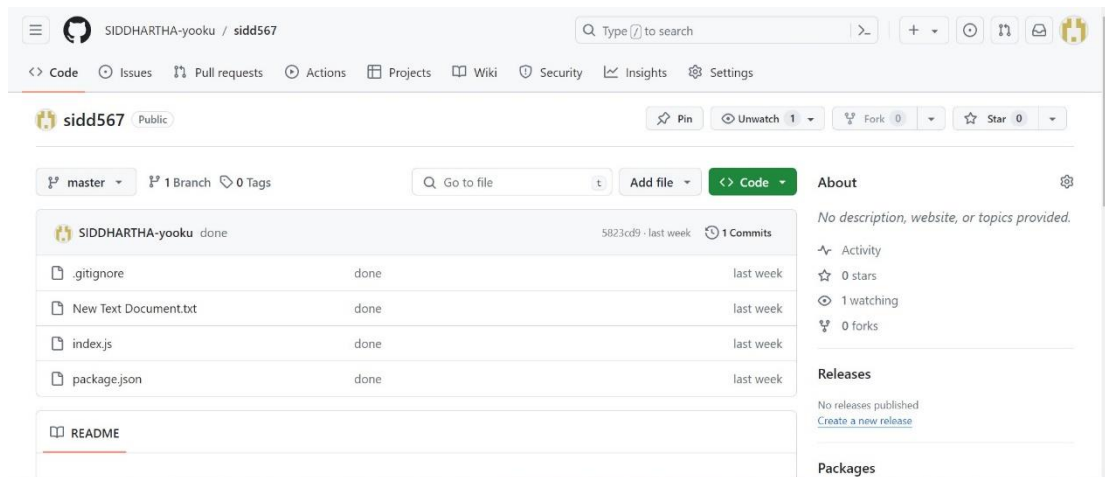
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   New Text Document.txt
        new file:   index.js
        new file:   package.json

Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.email
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.email soumyadipdhara01@gmail.com
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global --add user.email "shaw33224@gmail.com"
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global --add user.email "shaw33224@gmail.com"
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global --add user.email "shaw33224@gmail.com"
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.email shaw33224@gmail.com
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.name SOUMYADIP76
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global --add user.name "SIDDHARTHA-yooku"
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.name SIDDHARTHA-yooku
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ |
```

```
(use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   New Text Document.txt
        new file:   index.js
        new file:   package.json

Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.email
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.email soumyadipdhara01@gmail.com
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global --add user.email "shaw33224@gmail.com"
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global --add user.email "shaw33224@gmail.com"
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global --add user.email "shaw33224@gmail.com"
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.email shaw33224@gmail.com
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.name SOUMYADIP76
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global --add user.name "SIDDHARTHA-yooku"
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git config --global user.name SIDDHARTHA-yooku
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git commit -m "done"
[master (root-commit) 5823cd9] done
 4 files changed, 33 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 New Text Document.txt
 create mode 100644 index.js
 create mode 100644 package.json
Admin@BL2-066 MINGW64 ~/Desktop/aws project/myawsproject1 (master)
$ git remote add origin https://github.com/SIDDHARTHA-yooku/sidd67.git
```

14. Now go back to GitHub and click on “SIDDHARTHA-YOOKU”/newrewrepo3”. Here we can see that all the files are uploaded here.



For deploying the project from GitHub to local machine, the steps are as follows:-

1. We have to create a new folder on our desktop and then right-click on the folder then go to "Select more options" and select "Open with Git Bash".

```
haw3@DESKTOP-2C64RVK MINGW64 ~/OneDrive/Desktop
git clone https://github.com/moumita02314/myawsproject1.git
Cloning into 'myawsproject1'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.

haw3@DESKTOP-2C64RVK MINGW64 ~/OneDrive/Desktop
```

2. In the terminal we have to enter the following commands:

git clone https://github.com/arijit-giorno/newrepo1.git

3. The files have been successfully cloned. Hence, the deployment of the project from GitHub to the local machine has been completed.

