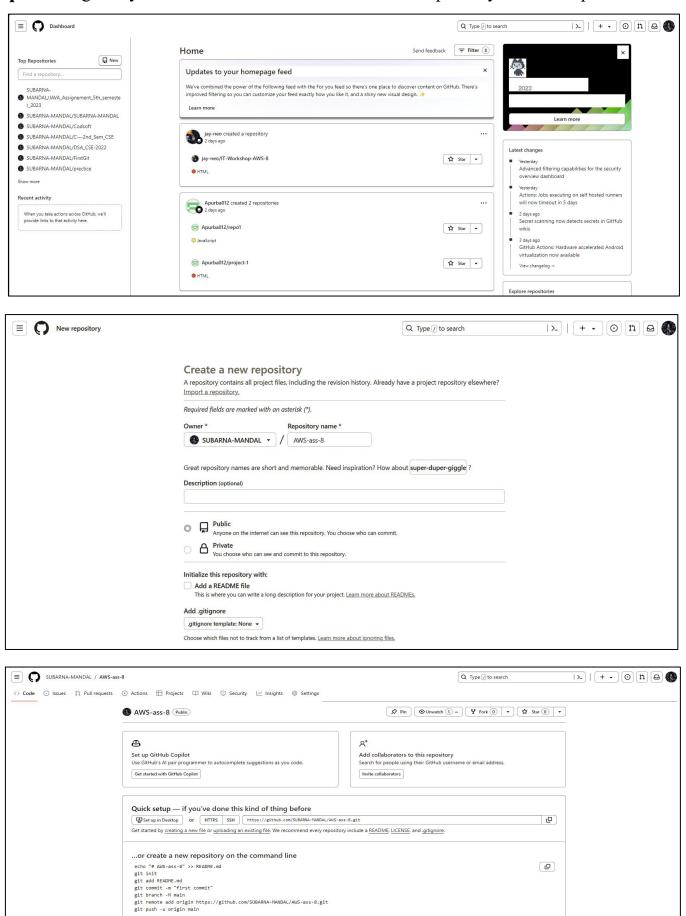
Assignment - 8

Problem statement: Deploy a project from local machine to Github and vice versa.

Procedure:

Step 1: Login to your GitHub account and create a new repository & make it public.



Step 2: After creating new repository, go to <u>Settings</u> → <u>Developer Options</u> → <u>Personal Access Token</u> → <u>Token (Classic)</u> → <u>Generate new token (classic)</u> Give a note name about new token and check all the boxes and set Expiration: 90 days and generate token.

The contractions with contract and place to the contract of an appearance of any place of any pl
Social and indicated and subject 1 Social and residence with a large of the control of the cont
Strategy of productions and service and se
Strategy of Personal and Anticological Strategy of Personal Anticological Strategy of Personal and Anticological Strategy of Personal
Williams of the proof of the pr
Contact Ages Cont
Contraction Total Access to Management of the State of State o
Case, (larray, and assertion Case,
Case, (larray, and assertion Case,
Contractions Co
College Settings College Sett
Colonia productions Colonia production Coloni
Office Service Control Se
Control points Control points
General County County of County Place Plac
Content your of Knowledge Section Recentage Integrative
Settings / Developed Petropy Control security and earlysis Interpretation
Solid states (policy Contract special analysis Contrac
Scoring: December and controls
Contraction of the contract of
Integration ## Applications ## Company Sections Developer Settings
Setting Overlayer Settings
Service of Personal access tokens (classic) Token you have generated their on the service for control Monographic for Control
Company Statistics Section to Special Processing to Section Se
Activities Security (p) Securi
Another Secretary to go Secret
Seturings / Developer settings Seturo Developer settings Develope
Settings / Developer Settings Settings / Developer Settings
Softmar / Developer settings Softmar / Developer Settings
Settings / Developer Settings Q. Type (to search > + 0 1
Secretarian Personal access tokens (classic) Personal access tokens Tokens you have generated that can be used to access the gaster processor of the processor tokens Tokens you have generated that can be used to access the gaster processor Tokens you have generated that can be used to access the gaster processor Tokens you have generated that can be used to access the gaster processor Tokens you have generated that can be used to access the gaster processor Tokens you have generated that can be used to access the gaster processor Tokens tokens T
Personal access tokens (classic) A Outh Apps Personal access tokens (classic) Personal access tokens Personal access Persona
Secretarian Personal access tokens (classic) Personal access tokens Tokens you have generated that can be used to access the gaster processor of the processor tokens Tokens you have generated that can be used to access the gaster processor Tokens you have generated that can be used to access the gaster processor Tokens you have generated that can be used to access the gaster processor Tokens you have generated that can be used to access the gaster processor Tokens you have generated that can be used to access the gaster processor Tokens tokens T
Contract Apps Formal access tolens Five granted tolens Five grante
Cotton Apps Personal access tolens Fore-grained tolens Values (States) October 5 you have generated that can be used to access the good access to the state of t
Ceres (Seaso) Fine gained belans Geta Ceres (Casad) Ceres (Cas
Ceres (Seaso) Fine gained belans Geta Ceres (Casad) Ceres (Cas
Fine private laters (Section Acres to Manager Codes) Ferroral access tokens (Section Acres tokens (Section Acres tokens (Section Acres tokens)) Formula access tokens (Section Acres tokens) Formula access tokens (Section Acres tokens) A Country Acres Acres tokens For granued tokens
Personal access tokens (classic) function lies ordinary Outry back Authorization. C 2004 Girtub Inc. Terms Privacy Security Status Docs Contact Manage codies: Do not there my personal information S Girtub Apps
A OAUth Apps A OAUth Apps Personal access tolkens The spence datasets The spence datas
ROUM Apps Personal access tokens Fire-grained tokens Tokens (classid) Personal access tokens (classid function like ordinary OAuth access tokens. They can be used instead of a password for Git over Basic Authenticate to the API over Basic Authentication. Note new-token What she token for? Expiration * 90 days ** Tokens (processed access for personal tokens. Read more about OAuth access. Fire-grained tokens. Fire-grained tokens.
Personal access tokens Personal access tokens Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTPS, or can be used to authenticate to the API over Basic Authentication. Note Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTPS, or can be used to authenticate to the API over Basic Authentication. Note Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTPS, or can be used to authenticate to the API over Basic Authentication. Note Personal access tokens (classic) function like ordinary OAuth access for like and they are also and
Personal access tokens Personal access tokens Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTPS, or can be used to authenticate to the API over Basic Authentication. Note Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTPS, or can be used to authenticate to the API over Basic Authentication. Note Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTPS, or can be used to authenticate to the API over Basic Authentication. Note Personal access tokens (classic) function like ordinary OAuth access for like and they are also and
Personal access tokens fine-grained tokens Tokens (clessic) Tokens (clessic) Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git owner programment of the programment o
Fine-grained totans Tolares (clessic) Rets Tolares (clessic) Rets Tolares (clessic) Rets Tolares (clessic) Rets Ret
Tolero (Classic)
What sits token for? Expiration * 90 days * The token will expire on Fri, Aid 5 2024 Select scopes Scopes define the access for personal tokens. Read more about OAuth scopes. ### Expositation ### Expositation ### Expositation ### Expositation ### Expositation #### Expositation #### Expositation #### Expositation ###################################
Expiration * Soldays
Select scopes Scopes define the access for personal tokens. Read more about OAuth scopes. Propo
Scopes define the access for personal tokens. Read more about OAuth scopes. Pepo
Scopes define the access for personal tokens. Read more about OAuth scopes. # repo. # repo. # repostatus
repoctatus Access commit status ■ repoc_deployment Access deployment status ■ public_depon Access public reportionies ■ public_depon Access public reportionies ■ repocimite Access repository invitations ■ security_events Read and write security events workflow Update GitHub Action workflows writepackages Upload packages to GitHub Package Registry ■ read-packages Download packages from GitHub Package Registry delete-packages Delete packages from GitHub Package Registry delete-packages Delete packages from GitHub Package Registry admin.org Full control of orgs and teams, read and write org projects ■ writeorg Read and write org and team membership, read and write org projects ■ writeorg Read and write org and team membership, read and write org projects
■ repo_deployment ■ public_repo Access deployment status ■ public_repo Erepoinrite Access repository invitations ■ security_events ■ workflow Update GitHub Action workflows Upload packages Upload packages from GitHub Package Registry ■ read-packages ■ read-packages Upload packages from GitHub Package Registry ■ delete-packages Ueleter-packages Full control of orgs and teams, read and write org projects ■ writtong Read and write org and team membership, read and write org projects ■ writteng Read and write org and team membership, read and write org projects ■ writteng Read and write org and team membership, read and write org projects
■ public_repo Access public repositories ■ repoxim/te Access repository invitations ■ security_events Read and write security events ✓ workflow Update Github Action workflows ✓ writepackages Upload packages to Github Package Registry ✓ reletepackages Download packages from Github Package Registry ✓ deletepackages Delete packages from Github Package Registry ✓ admin.org ■ reletepackages ■ reletepackages and and write org projects ■ writeorg ■ writeorg ■ Read and write org and team membership, read and write org projects
repositivite # security_events # security_events workflow
W workflow Update GitHub Action workflows W writerpackages Upload packages to GitHub Package Registry
W witterpackages Upload packages to Giftlub Package Registry ■ machpackages Download packages from Giftlub Package Registry ✓ eletete packages Delete packages from Giftlub Package Registry ✓ admin.org Full control of orgs and teams, read and write org projects ■ writecorg Read and write org and team membership, read and write org projects
■ readspackages Download packages from GitHub Package Registry ■ delete-packages ■ Delete packages from GitHub Package Registry ■ admin.org ■ Full control of orgs and teams, read and write org projects ■ writecorg ■ Read and write org and team membership, read and write org projects
✓ delete-packages Delete packages from GitHub Package Registry ✓ admin.org Full control of orgs and teams, read and write org projects write.org Read and write org and team membership, read and write org projects
■ admin.org ■ write.org
writecorg Read and write org and team membership, read and write org projects
■ readorg Read org and team memberthly, read org projects ■ manage_unnersorg Manage org runners and runner groups
* * *
Settings / Developer Settings
88 GitHub Apps Personal access tokens (classic) Generate new token Revoke all
Q OAuth Appr
A OAuth Apps Tokage upin have appointed that can be used to accord the Cithfully ADI.
OAuth Apps Tokens you have generated that can be used to access the <u>GitHub API</u> . Tokens you have generated that can be used to access the <u>GitHub API</u> .
Personal access tokens Tokens you have generated that can be used to access the GitHub API. Fine-grained tokens Beta
Personal access tokens Tokens you have generated that can be used to access the <u>GitHub API</u> . Fine-grained tokens Beta Make sure to copy your personal access token now. You won't be able to see it again!
Personal access tokens Tokens you have generated that can be used to access the GitHub API. Fine-grained tokens Beta
Personal access tokens Tokens you have generated that can be used to access the GitHub API. Fine-grained tokens Beta Make sure to copy your personal access token now. You won't be able to see it again!
Personal access tokens Tokens you have generated that can be used to access the <u>GitHub API</u> . Beta Make sure to copy your personal access token now. You won't be able to see it again!
Personal access tokens Tokens you have generated that can be used to access the GitHub API. Fine-grained tokens Beta Tokens (classic) Make sure to copy your personal access token now. You won't be able to see it again!

Step 3: Go to your local machine and make sure git is installed. Next the following commands are the required for the initialize new git repositroy in your local machine and upload local repository into github platform using git cli via https protocol with pre-generated token.

- git init
- git add.
- git commit -m "<your massage here>"
- git remote add <remote> https://<username>:<token>@github.com/<username>/<repo>.git
- git push -u <remote> <branch>

From our local machine: To upload repository in GitHub:

```
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8$ 11
total 0
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8$ cat > index.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Home : AWS</title>
</head>
<body>
<div align="center">
<h1>Home</h1>
</div>

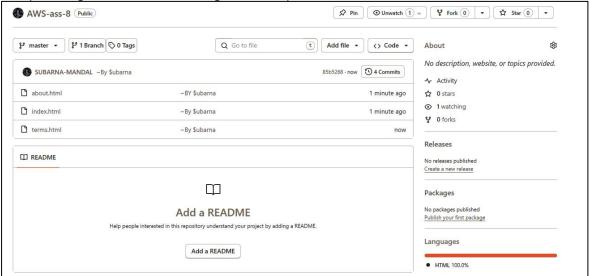
<a href="./terms.html">Terms and Condition</a> |
<a href="./about.html">About</a>

</body>
</html>
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8$ cat > about.html
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<title>About : AWS</title>
</head>
<body>
<div align="center">
<h1>About</h1>
</div>

</body>
</html>
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8$ cat > terms.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Terms and Condition : AWS</title>
</head>
<body>
<div align="center">
<h1>Term and Condition</h1>
</div>
</body>
</html>
Subarna@Subarna-Ideapad MINGw64 /d/IT-lab-assign-8$ git config --get-regexp
'^user\.(name|email)$
user.name SUBARNA-MANDAL user.email subarnamandal17@gmail.com
```

```
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8 (master) $ ls
about.html
              index.html
                             terms.html
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8$ git init
Initialized empty Git repository in D:/IT-lab-assign-8/.git/
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8 (master)$ git add .
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8 (master)$ git commit -m "~By
$ubarna'
[master (root-commit) 4026f8d] ~By $ubarna
 3 files changed, 45 insertions(+)
 create mode 100644 about html
 create mode 100644 index.html
 create mode 100644 terms.html
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8 (master)$ git remote add origin
https://SUBARNA-MANDAL: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@github.com/SUBARNA-
MANDAL/AWS-ass-8.git
Subarna@Subarna-Ideapad MINGW64 /d/IT-lab-assign-8 (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 708 bytes | 708.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/SUBARNA-MANDAL/AWS-ass-8.git
                        master -> master
 * [new branch]
branch 'master' set up to track 'origin/master'.
```

After pushing the files into github ripo "AWS-ass-8" --



From our local machine: To download public repository from GitHub:

```
Subarna@Subarna-Ideapad MINGW64 /d
$ git clone https://github.com/sudip7407/Repo1.git Cloning into 'Repo1'... remote: Enumerating objects: 10, done. remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 10 (delta 1), reused 0 (delta 0), pack-reused 0 Receiving objects: 100% (10/10), 49.46 KiB | 803.00 KiB/s, done.
Resolving deltas: 100\% (1/1), done.
Subarna@Subarna-Ideapad MINGW64 /d
Subarna@Subarna-Ideapad MINGW64 /d$ cd Repo1
Subarna@Subarna-Ideapad MINGW64 /d/Repo1 (main)$ 11
total 183
                                                    6 02:40 'New Text Document.txt'
-rw-r--r-- 1 Subarna 197609
                                        115 Apr
-rw-r--r-- 1 Subarna 197609
                                        194 Apr
                                                    6 02:40
                                                                index.js
-rw-r--r-- 1 Subarna 197609 182693 Apr
                                                    6 02:40
                                                                package-lock.json
                                        297 Apr
-rw-r--r-- 1 Subarna 197609
                                                    6 02:40
                                                                package.json
```