

PROGRAM 1: CREATE AND MANAGE LOCAL REPOSITORY USING GIT

```
MINGW64:/c/Users/ramsr/myproject
$ 
$ ramsr@RAMLAHK MINGW64 ~
$ mkdir myproject
$ cd myproject
$ git init
Initialized empty Git repository in C:/Users/ramsr/myproject/.git/
$ touch index.html test.txt script.py
$ git status
On branch master

No commits yet

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

    index.html
    script.py
    test.txt

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

$ git add index.html
$ git commit -m "first commit -added html, text, and python files"
Author identity unknown

*** Please tell me who you are.

Run
  git config --global user.name "Your Name"
  git config --global user.email "you@example.com"

To set your account's default identity,
  'git config --global user.name "Your Name"'
  'git config --global user.email "you@example.com"'.

fatal: unable to auto-detect email address (got 'ramsr@RAMLAHK.(none)')

$ git config --global user.name "lakshmi"
$ git config --global user.email "lakshmisree2905@gmail.com"
git: 'config' is not a git command. See 'git --help'.
The most similar command is
  config

$ git status
On branch master

No commits yet

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

    index.html
    script.py
    test.txt

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

$ git add index.html
$ git commit -m "first commit -added html, text, and python files"
[master (root-commit) 55f862f] First commit -added html, text, and python files
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html

$ git config --list
diff.guitool=gitgconv-astextplain
filter.lfs.cleanexit-lfs.clean = %F
filter.lfs.smudge-git-lfs.smudge -- %F
filter.lfs.process-git-lfs.filter-process
http.sslbackend=openssl
http.sslhackbar=chunnel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=lakshmisree2905@gmail.com
core.repositoryformatversion=0
core.fifetime=false
core.ignoredeletions=true
core.symlinks=false
core.ignorecase=true

$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    index.html
    script.py
    test.txt

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

$ git add index.html
$ git commit -m "first commit -added html, text, and python files"
[master (root-commit) 55f862f] First commit -added html, text, and python files
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html

$ git config --list
diff.guitool=gitgconv-astextplain
filter.lfs.cleanexit-lfs.clean = %F
filter.lfs.smudge-git-lfs.smudge -- %F
filter.lfs.process-git-lfs.filter-process
http.sslbackend=openssl
http.sslhackbar=chunnel
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.ignorecase=true

$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    index.html
    script.py
    test.txt

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

$ git add index.html
$ git commit -m "first commit -added html, text, and python files"
[master (root-commit) 55f862f] First commit -added html, text, and python files
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html

$ git config --list
diff.guitool=gitgconv-astextplain
filter.lfs.cleanexit-lfs.clean = %F
filter.lfs.smudge-git-lfs.smudge -- %F
filter.lfs.process-git-lfs.filter-process
http.sslbackend=openssl
http.sslhackbar=chunnel
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.ignorecase=true
```

```
MINGW64:/c/Users/ramsr/myproject
$ 
$ ramsr@RAMLAHK MINGW64 ~
$ git config --global user.name "lakshmisree"
$ git config --global user.email "lakshmisree2905@gmail.com"
$ git config --global http.sslbackend=openssl
$ git config --global http.sslhackbar=chunnel
$ git config --global core.autocrlf=true
$ git config --global core.fscache=true
$ git config --global core.symlinks=false
$ git config --global pull.rebase=false
$ git config --global credential.helper=manager
$ git config --global credential.https://dev.azure.com.usehttppath=true
$ git config --global init.defaultbranch=master
$ git config --global user.name=lakshmisree2905@gmail.com
$ git config --global core.repositoryformatversion=0
$ git config --global core.fifetime=false
$ git config --global core.ignoredeletions=true
$ git config --global core.symlinks=false
$ git config --global core.ignorecase=true

$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    index.html
    script.py
    test.txt

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

$ git add index.html
$ git commit -m "first commit -added html, text, and python files"
[master (root-commit) 55f862f] First commit -added html, text, and python files
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html

$ git config --list
diff.guitool=gitgconv-astextplain
filter.lfs.cleanexit-lfs.clean = %F
filter.lfs.smudge-git-lfs.smudge -- %F
filter.lfs.process-git-lfs.filter-process
http.sslbackend=openssl
http.sslhackbar=chunnel
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.ignorecase=true

$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    index.html
    script.py
    test.txt

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

$ git add index.html
$ git commit -m "first commit -added html, text, and python files"
[master (root-commit) 55f862f] First commit -added html, text, and python files
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html

$ git config --list
diff.guitool=gitgconv-astextplain
filter.lfs.cleanexit-lfs.clean = %F
filter.lfs.smudge-git-lfs.smudge -- %F
filter.lfs.process-git-lfs.filter-process
http.sslbackend=openssl
http.sslhackbar=chunnel
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.ignorecase=true

$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    index.html
    script.py
    test.txt

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

$ git add index.html
$ git commit -m "first commit -added html, text, and python files"
[master (root-commit) 55f862f] First commit -added html, text, and python files
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html

$ git config --list
diff.guitool=gitgconv-astextplain
filter.lfs.cleanexit-lfs.clean = %F
filter.lfs.smudge-git-lfs.smudge -- %F
filter.lfs.process-git-lfs.filter-process
http.sslbackend=openssl
http.sslhackbar=chunnel
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.ignorecase=true

$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    index.html
    script.py
    test.txt

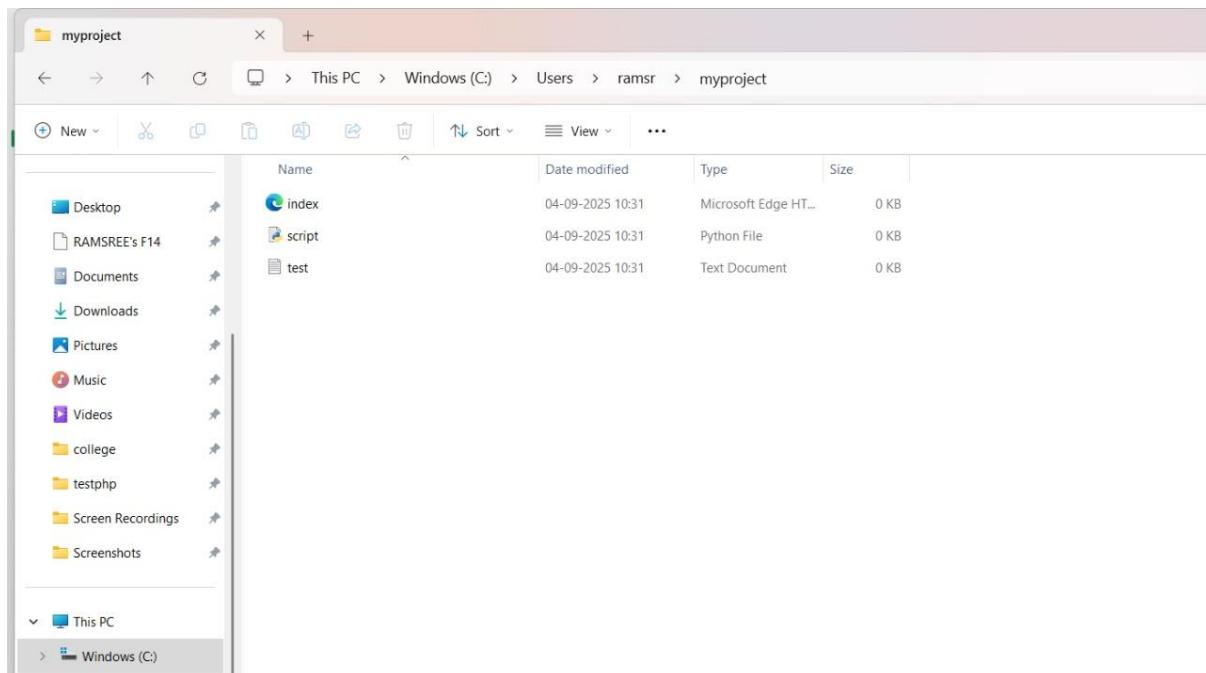
nothing added to commit but untracked files present (use "git add" to track)

$ git add index.html
$ git commit -m "first commit -added html, text, and python files"
[master (root-commit) 55f862f] First commit -added html, text, and python files
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html

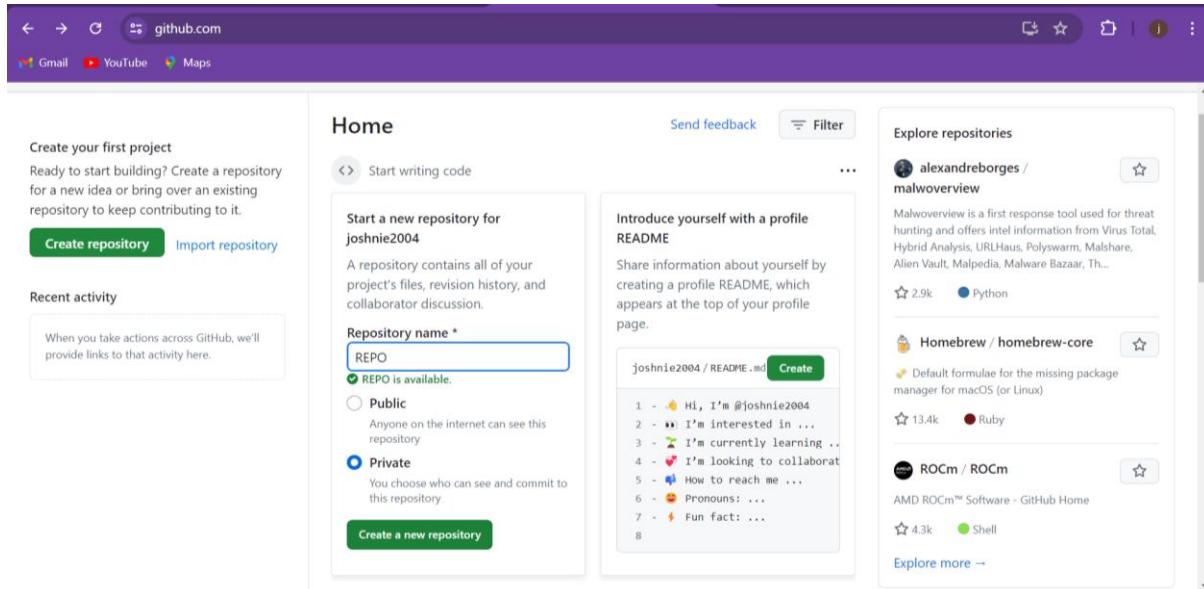
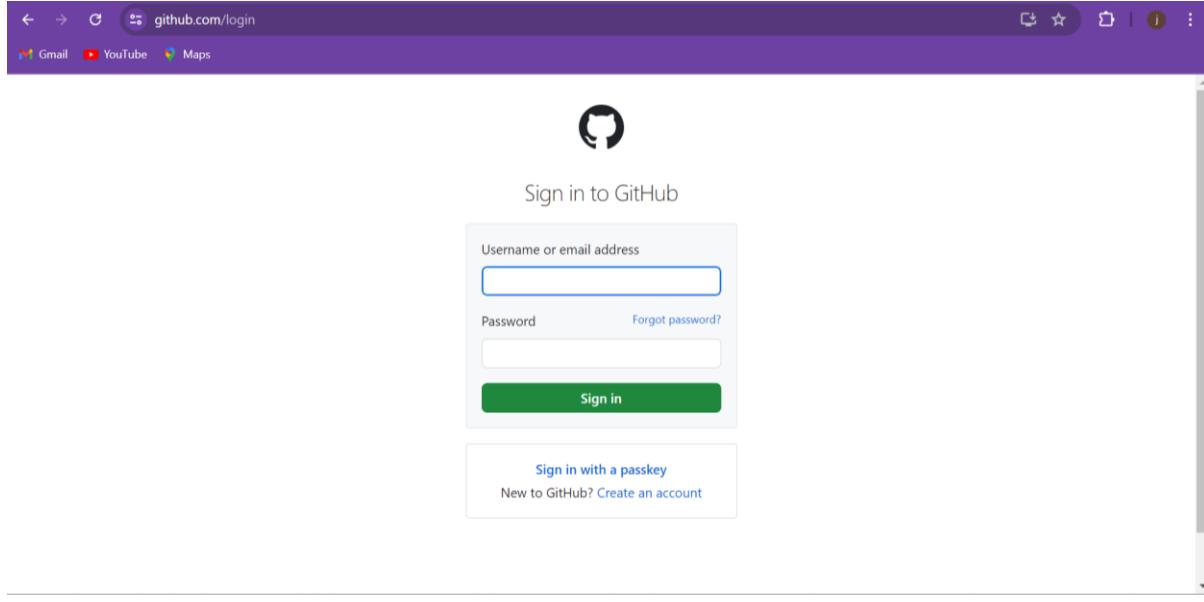
$ git config --list
diff.guitool=gitgconv-astextplain
filter.lfs.cleanexit-lfs.clean = %F
filter.lfs.smudge-git-lfs.smudge -- %F
filter.lfs.process-git-lfs.filter-process
http.sslbackend=openssl
http.sslhackbar=chunnel
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.ignorecase=true
```

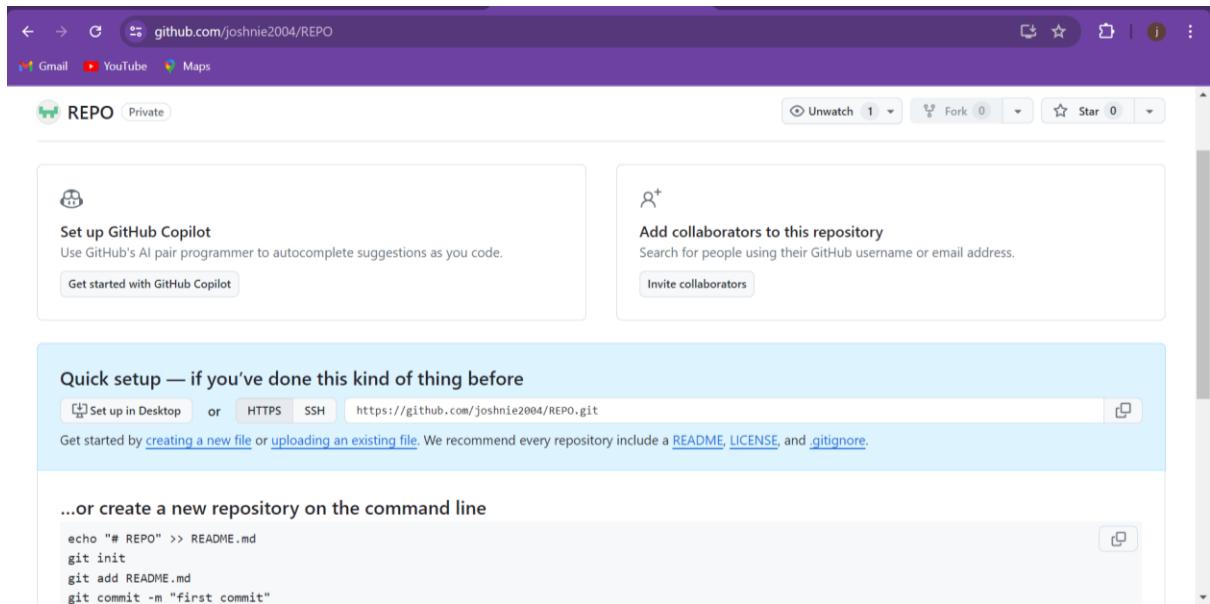
```
MINGW64/cUsers/ramsr/myproject
$ git add .
rangsraM@AK MINGW64 ~/myproject (master)
$ git commit -m "First Commit -added html,txt, and python files"
[master b28e60] First Commit -added html,txt, and python files
 2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 script.py
create mode 100644 test.txt
rangsraM@AK MINGW64 ~/myproject (master)
$ git status
On branch master
nothing to commit, working tree clean
rangsraM@AK MINGW64 ~/myproject (master)
$ git log --oneline
b28e60 (HEAD => master) First Commit -added html,txt, and python files
$ git branch
* master
rangsraM@AK MINGW64 ~/myproject (master)
$ git branch new-feature
rangsraM@AK MINGW64 ~/myproject (new-feature)
$ git checkout new-feature
Switched to branch 'new-feature'
rangsraM@AK MINGW64 ~/myproject (new-feature)
$ git branch -a
  master
* new-feature
rangsraM@AK MINGW64 ~/myproject (new-feature)
$ git branch -d new-feature
error: Cannot delete branch 'new-feature' used by worktree at 'c:/users/ramsr/myproject'
rangsraM@AK MINGW64 ~/myproject (new-feature)
$ git
rangsraM@AK MINGW64 ~/myproject (new-feature)
$ git checkout main
error: pathspec 'main' did not match any file(s) known to git
rangsraM@AK MINGW64 ~/myproject (new-feature)
$ git checkout Lakram
error: pathspec 'Lakram' did not match any file(s) known to git
rangsraM@AK MINGW64 ~/myproject (new-feature)
$ git
rangsraM@AK MINGW64 ~/myproject (new-feature)
$ git checkout master
Switched to branch 'master'
rangsraM@AK MINGW64 ~/myproject (master)
$ git branch -d new-feature
Deleted branch new-feature (was b28e60).
28°C
Windy
Search web & PC
ENG IN
1053
04-09-2025
```

OUTPUT:



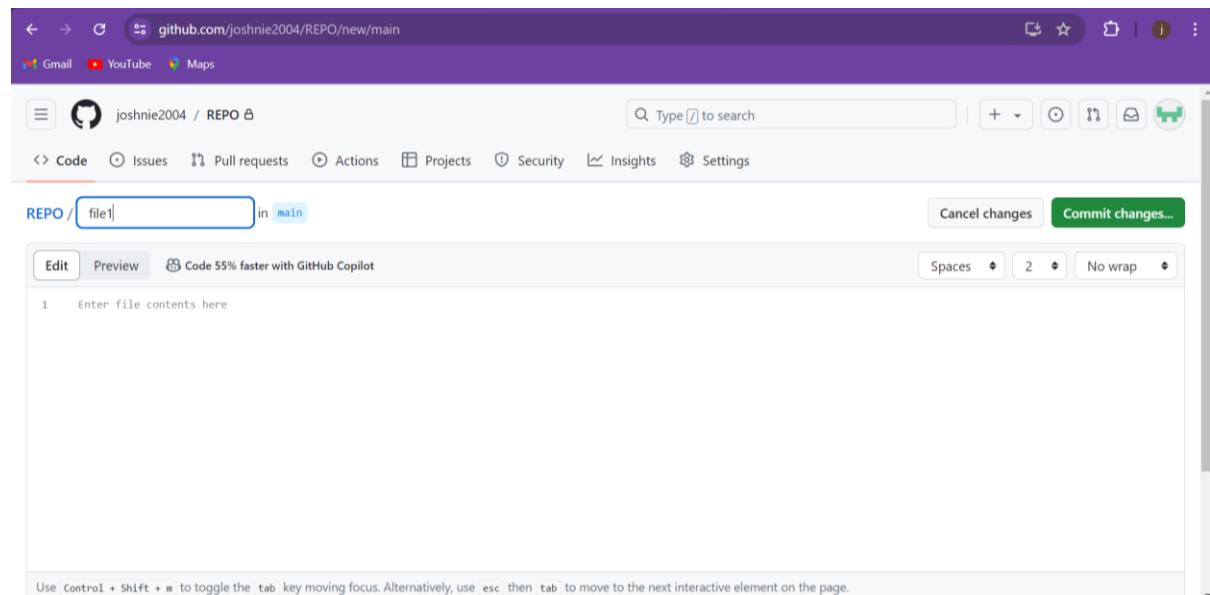
PROGRAM 2: REMOTE REPOSITORY



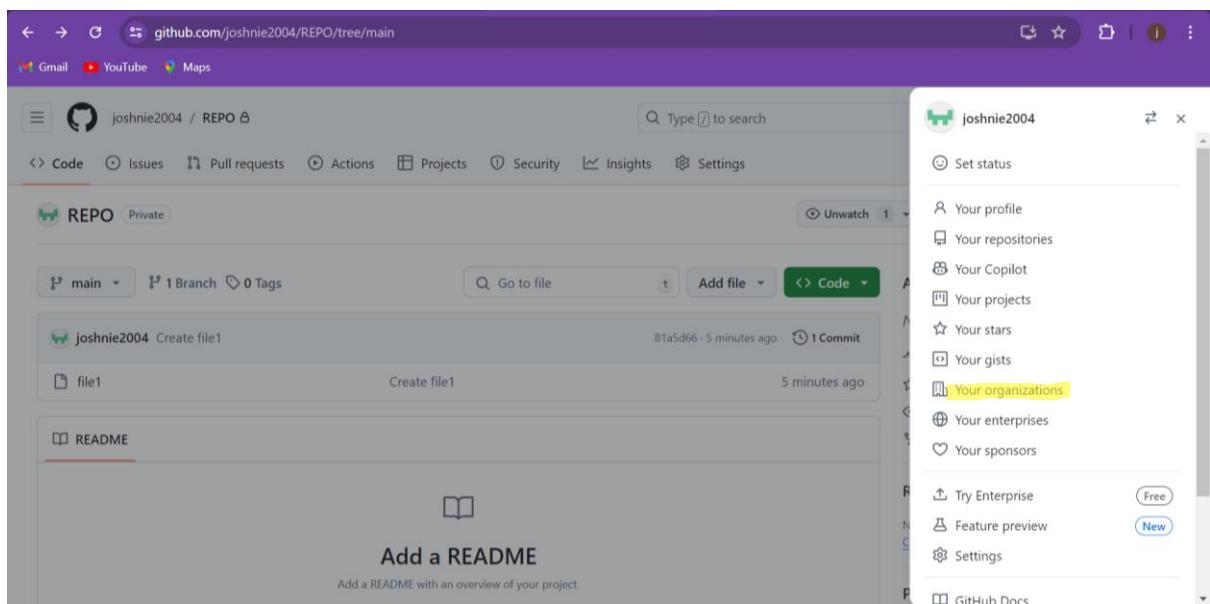
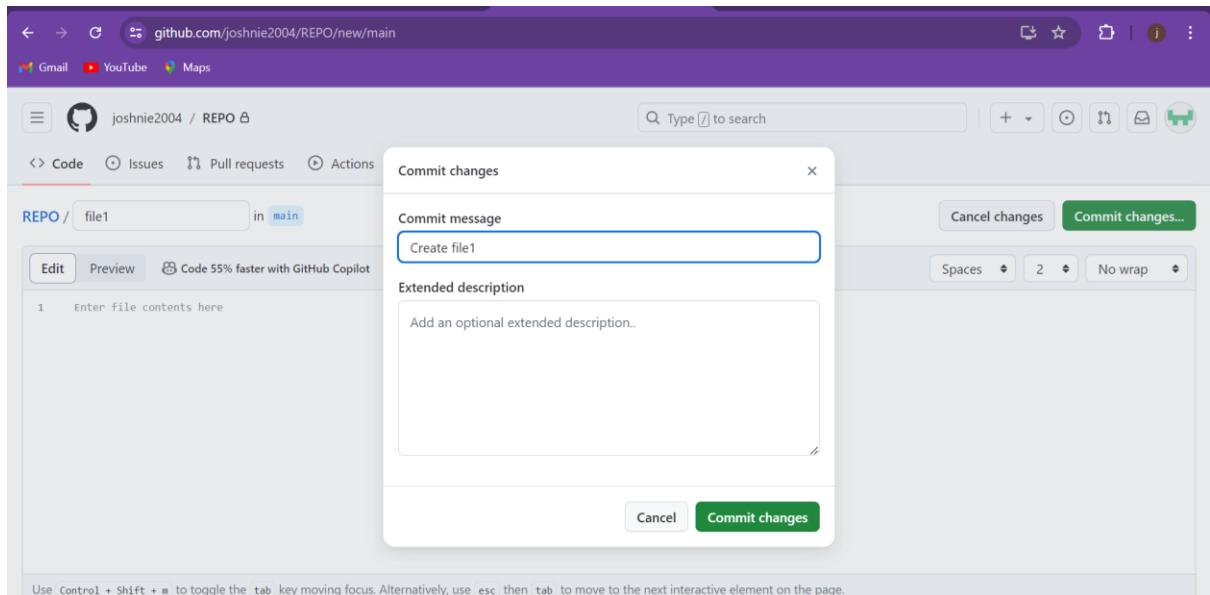


The screenshot shows the GitHub repository creation interface. At the top, there's a purple header bar with the URL "github.com/joshnie2004/REPO". Below it is a toolbar with links to Gmail, YouTube, and Maps. The main content area has two sections: "Set up GitHub Copilot" (with a "Get started with GitHub Copilot" button) and "Add collaborators to this repository" (with a "Invite collaborators" button). A blue banner below these sections says "Quick setup — if you've done this kind of thing before" and provides instructions for setting up the repository via desktop, HTTPS, or SSH, along with links to README, LICENSE, and .gitignore files. It also suggests creating a new repository on the command line with the following commands:

```
echo "# REPO" >> README.md
git init
git add README.md
git commit -m "first commit"
```



The screenshot shows the GitHub file editor interface. The URL in the header is "github.com/joshnie2004/REPO/new/main". The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The main area shows a file named "file1" in the "main" branch. The file content is a single line: "1 Enter file contents here". There are buttons for "Edit", "Preview", and "Commit changes...". A note at the bottom of the editor says: "Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page."



github.com/settings/organizations

Gmail YouTube Maps

Settings

joshnie2004 (joshnie2004)
Your personal account

Public profile Account Appearance Accessibility Notifications

Access Billing and plans Emails Password and authentication Sessions SSH and GPG keys Organizations

Organizations

You are not a member of any organizations.

New organization

Transform account

Turn joshnie2004 into an organization

This screenshot shows the GitHub Settings page for the user 'joshnie2004'. The left sidebar has sections like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys, and Organizations. The Organizations section is currently selected. The main area displays a message: 'You are not a member of any organizations.' with a 'New organization' button. Below this is a 'Transform account' section with a button to turn the user into an organization. A sidebar on the right lists various account settings.

github.com/organizations/plan

Gmail YouTube Maps

Type [] to search

Choose a plan

Pick a plan for your organization

Free
The basics for individuals and organizations
\$ 0 USD per user/month
[Create a free organization](#)

MOST POPULAR Team
Advanced collaboration for individuals and organizations
\$ 4 USD per user/month
[Continue with Team](#)

Enterprise
Security, compliance, and flexible deployment
\$ 21 USD per user/month
[Start a free trial](#) [Contact Sales](#)

> [Unlimited public/private repositories](#)

This screenshot shows the GitHub 'organizations/plan' page. It features three main plan options: 'Free', 'Team', and 'Enterprise'. The 'Team' plan is highlighted as 'MOST POPULAR'. Each plan includes a price per user/month and a call-to-action button. A note at the bottom indicates unlimited repositories.

The screenshot shows the GitHub account organization creation interface. At the top, there's a purple header bar with browser navigation icons and the URL github.com/account/organizations/new?plan=free&ref_cta=Create%2520a%2520free%2520organization&ref_loc=cards&ref_page=%2Forganiz.... Below the header, there are links to Gmail, YouTube, and Maps. The main content area has a purple background with white text. It starts with a placeholder "Tell us about your organization" and a large bold heading "Set up your organization".

Organization name *

joshniet

This will be the name of your account on GitHub.
Your URL will be: <https://github.com/joshniet>.

Contact email *

joethangaraj2004@gmail.com

This organization belongs to:

My personal account

i.e., joshniet2004

A business or institution

For example: GitHub, Inc., Example Institute, American Red Cross

Verify your account

The screenshot shows the continuation of the GitHub organization creation interface. The header and initial form fields are visible at the top. The main content area now features a large empty rectangular box for "Add-ons".

Add-ons

Get GitHub Copilot Business in this organization

Boost developer productivity for \$19/user/month. Pay only for assigned seats after setup.
[See Copilot Business docs.](#)

I hereby accept the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

Next



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

The screenshot shows the GitHub organization setup page for the organization "joshniet". The title bar includes links to Gmail, YouTube, and Maps. The main heading is "Welcome to joshniet" with the subtext "Start collaborating". Below this, there's a section titled "Add organization members" with a note about permissions and a link to "Learn more about permissions for organizations". A search bar is provided for finding members. A large green button at the bottom right says "Complete setup". A yellow "Skip this step" button is also visible. The footer contains standard GitHub links like Terms, Privacy, Security, Status, Docs, Contact, and Manage cookies.

The screenshot shows the GitHub organization overview page for "joshniet". The top navigation bar has links for Overview, Repositories, Projects, Packages, Teams, People, and Settings. The main content area features a large profile picture placeholder for "joshniet" with a "Follow" button. A central message says "We think you're gonna like it here." Below this, there are two suggested tasks: "Invite your people" and "Customize members' permissions". On the right side, there are sections for "View as: Public", "Discussions", and "README file".

The screenshot shows a GitHub repository page for 'joshnie2004/REPO'. The repository is private. It contains one branch ('main') and no tags. There is one commit from 'joshnie2004' titled 'Create file1' made 12 minutes ago. A file named 'file1' was created at the same time. The README section is present but empty. The sidebar includes sections for About, Activity, Releases, and Packages.

About

No description, website, or topics provided.

Activity

1 Commit

Releases

No releases published
[Create a new release](#)

Packages

The screenshot shows the 'Create a new fork' form on GitHub. The owner is set to 'joshniet' and the repository name is 'REPO'. The 'REPO is available' checkbox is checked. A note states that forks are named the same as their upstream repository. The 'Description (optional)' field is empty. The 'Copy the main branch only' checkbox is checked, with a note that it allows contributing back to the original repository. A note also says you are creating a fork in the 'joshniet' organization. The 'Create fork' button is visible at the bottom.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner * Repository name *

joshniet / REPO

REPO is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Copy the main branch only

Contribute back to joshnie2004/REPO by adding your own branch. [Learn more](#).

ⓘ You are creating a fork in the joshniet organization.

Create fork

A screenshot of a GitHub repository page for 'joshniet / REPO'. The page shows a green 'Code' button highlighted. The repository is private and was forked from 'joshnie2004/REPO'. It contains one branch ('main'), one file ('file1'), and a README. The 'About' section indicates no description, website, or topics provided. There are 0 stars, 0 forks, and 0 releases.

github.com/joshniet/REPO

Code Pull requests Actions Projects Security Insights Settings

REPO Private
forked from joshnie2004/REPO

main 1 Branch 0 Tags

This branch is up to date with joshnie2004/REPO:main .

joshnie2004 Create file1 81a5d66 - 17 minutes ago 1 Commit

file1 Create file1 17 minutes ago

README

About

No description, website, or topics provided.

Activity

Custom properties

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

git clone https://github.com/YOUR_USERNAME/forked-repo.git

OUTPUT:

OS (C):				
This PC > OS (C) >				
Search OS (C):				
	New	Sort	View	...
> app	Name	Date modified	Type	Size
> Apps	app	17-09-2023 11:03	File folder	
> Autodesk	Apps	01-01-2022 05:08	File folder	
> backup	Autodesk	10-01-2023 11:01	File folder	
> dell	backup	01-01-2022 05:08	File folder	
> Documents	dell	01-01-2022 05:08	File folder	
> Drivers	Documents	17-07-2024 20:39	File folder	
> Intel	Drivers	01-01-2022 05:08	File folder	
> jdk1.3.1_28	Intel	19-07-2024 13:02	File folder	
> PerfLogs	jdk1.3.1_28	09-05-2023 17:52	File folder	
> Pictures	PerfLogs	07-05-2022 10:54	File folder	
> Program File:	Pictures	20-04-2023 20:39	File folder	
> Program File:	Program Files	19-07-2024 19:53	File folder	
> Program File:	Program Files (x86)	27-01-2024 13:27	File folder	
> rep	rep	19-07-2024 20:53	File folder	
> rep1	rep1	19-07-2024 20:41	File folder	
> Saved Picture	Saved Pictures	07-02-2024 22:36	File folder	
> TurboC++	TurboC++	19-11-2022 09:31	File folder	
19 items	1 item selected			

rep1				
This PC > OS (C) > rep1 >				
Search rep1				
	New	Sort	View	...
> app	Name	Date modified	Type	Size
> Apps	REPO	19-07-2024 20:44	File folder	
> Autodesk				
> backup				
> dell				
> Documents				
> Drivers				
> Intel				
> jdk1.3.1_28				
> PerfLogs				
> Pictures				
> Program File:				
> Program File:				
> rep				
> rep1				
> Saved Picture				
1 item	1 item selected			

PROGRAM 3: CREATE AND PUSH A FILE FROM LOCAL TO REMOTE REPOSITORY

```
MINGW64:/z/devopsdanu
$ mkdir devopsdanu

MINGW64:/z/devopsdanu
$ mkdir devopsdanu
mkdir: cannot create directory 'devopsdanu': File exists

MINGW64:/z/devopsdanu
$ "mkdir devopsdanu"
bash: mkdir devopsdanu: command not found

MINGW64:/z/devopsdanu
$ mkdir devops1

MINGW64:/z/devopsdanu
$ cd ^
MINGW64:/z/devopsdanu
$ cd devopsdanu

MINGW64:/z/devopsdanu
$ git init
Initialized empty Git repository in //KCW-LAB-dc/DATA/BSCCS/3BSCCSA/3BSCCSA11/devopsdanu/.git/

MINGW64:/z/devopsdanu (master)
$ touch index.html

MINGW64:/z/devopsdanu (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)

MINGW64:/z/devopsdanu (master)
$ nano index.html

MINGW64:/z/devopsdanu (master)
$ cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
<title>MY FIRST PAGE </title>
</head>
<body>
<p> hii </p>
</body>
</html>

MINGW64:/z/devopsdanu (master)
$ git add index.html
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

MINGW64:/z/devopsdanu (master)
$ git commit -m "Add index.html with basic HTML content"
[master (root-commit) 913b061] Add index.html with basic HTML content
Committer: unknown <3BSCCSA11@kcw.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
```

```
MINGW64:/z/devopsdanu
```

```
3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ git add index.html
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ git commit -m "Add index.html with basic HTML content"
[master (root-commit) 913b061] Add index.html with basic HTML content
Committer: unknown <3BSCCSA11@kcw.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 9 insertions(+)
create mode 100644 index.html

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ git status
On branch master
nothing to commit, working tree clean

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ git remote add origin https://github.com/dhanuja1709/devopsdanu.git

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ git commit -m "Add index.html"
On branch master
nothing to commit, working tree clean

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/dhanuja1709/devopsdanu.git'

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 320 bytes | 16.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/dhanuja1709/devopsdanu/pull/new/master
remote:
To https://github.com/dhanuja1709/devopsdanu.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

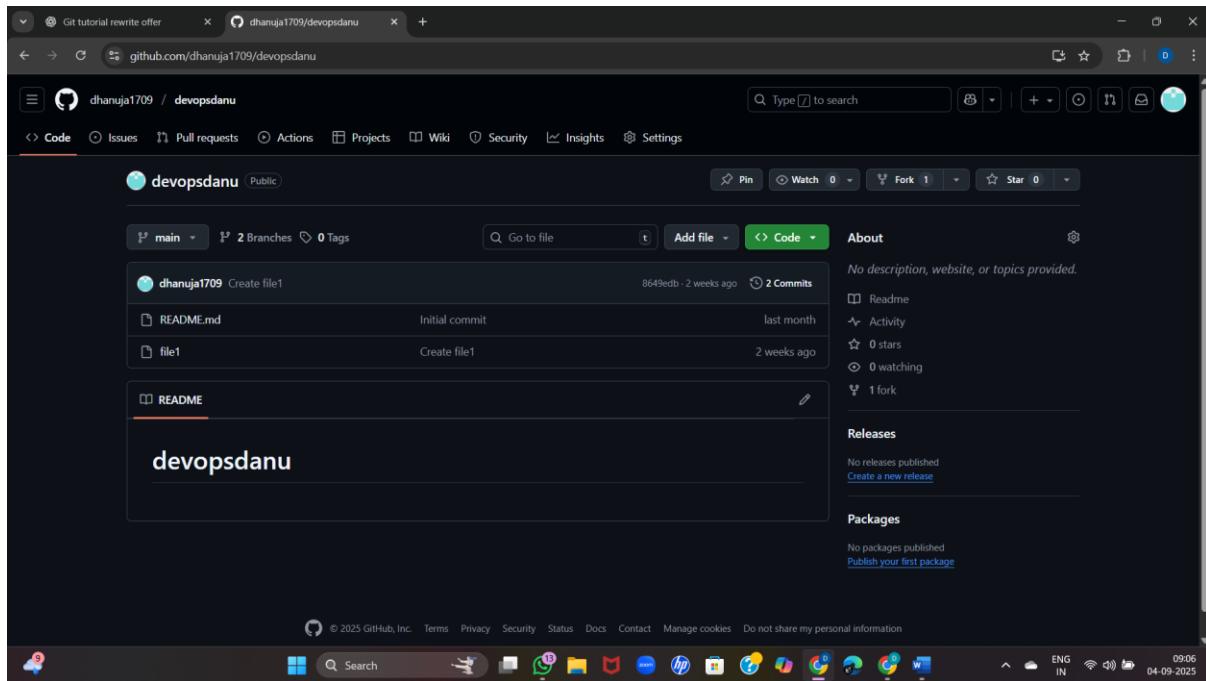
3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ git branch
* master

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ ^C

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$ commit
bash: commit: command not found

3BSCCSA11@KC-LAB10-TH-PC11 MINGW64 /z/devopsdanu (master)
$
```

OUTPUT :

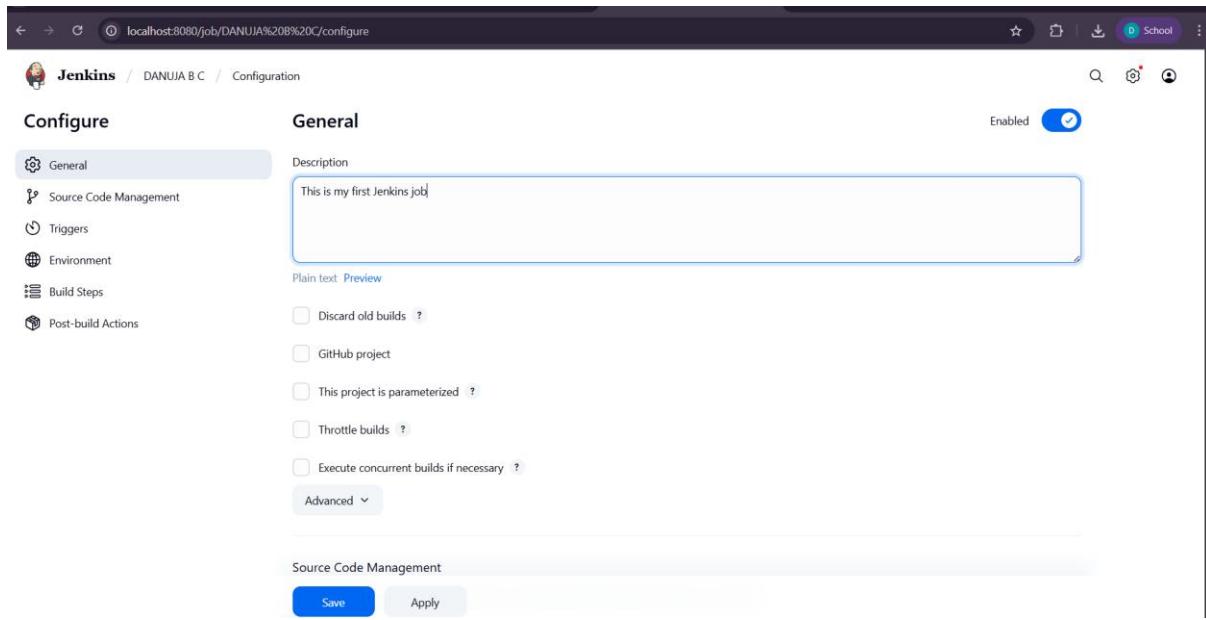


PROGRAM 4 : CREATE FIRST JOB IN JENKINS

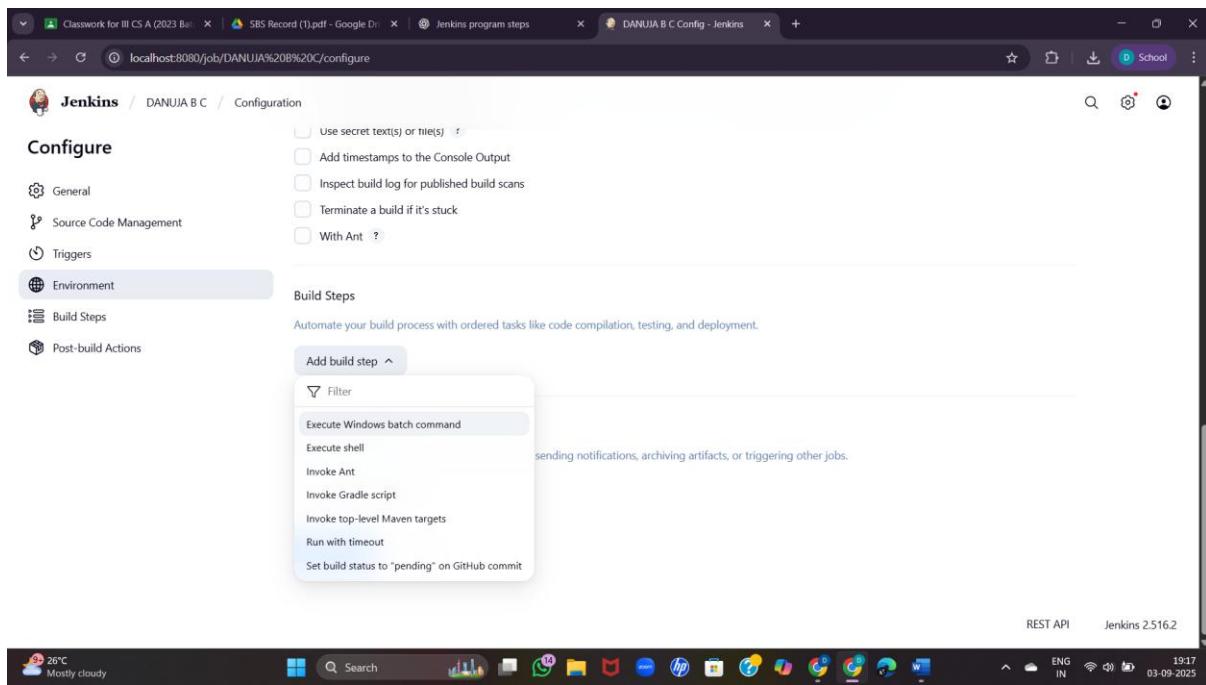
The screenshot shows the Jenkins Dashboard at localhost:8080. The main header says "Welcome to Jenkins!". Below it, a message states: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." A "Create a job" button is available. On the left, there are sections for "Build Queue" (empty) and "Build Executor Status" (0/2). On the right, there's a "Set up a distributed build" section with links for "Set up an agent" (computer icon), "Configure a cloud" (cloud icon), and "Learn more about distributed builds" (info icon).

localhost:8080/newjob REST API Jenkins 2.516.2

The screenshot shows the "New Item" dialog. It starts with a search bar containing "DANUJA B C". Below it, a "Select an item type" section lists four options: "Freestyle project" (selected, described as a classic general-purpose job type), "Pipeline" (described as orchestrating long-running activities across multiple agents), "Multi-configuration project" (described as suitable for testing across multiple environments), and "Folder" (described as creating a namespace for nested items). At the bottom is an "OK" button.



The screenshot shows the Jenkins General configuration page for a job named "DANUJA B C". The "Enabled" checkbox is checked. In the "Description" field, the text "This is my first Jenkins job" is entered. Below the description, there are several optional checkboxes: "Discard old builds", "GitHub project", "This project is parameterized", "Throttle builds", and "Execute concurrent builds if necessary". A "Save" button is visible at the bottom.

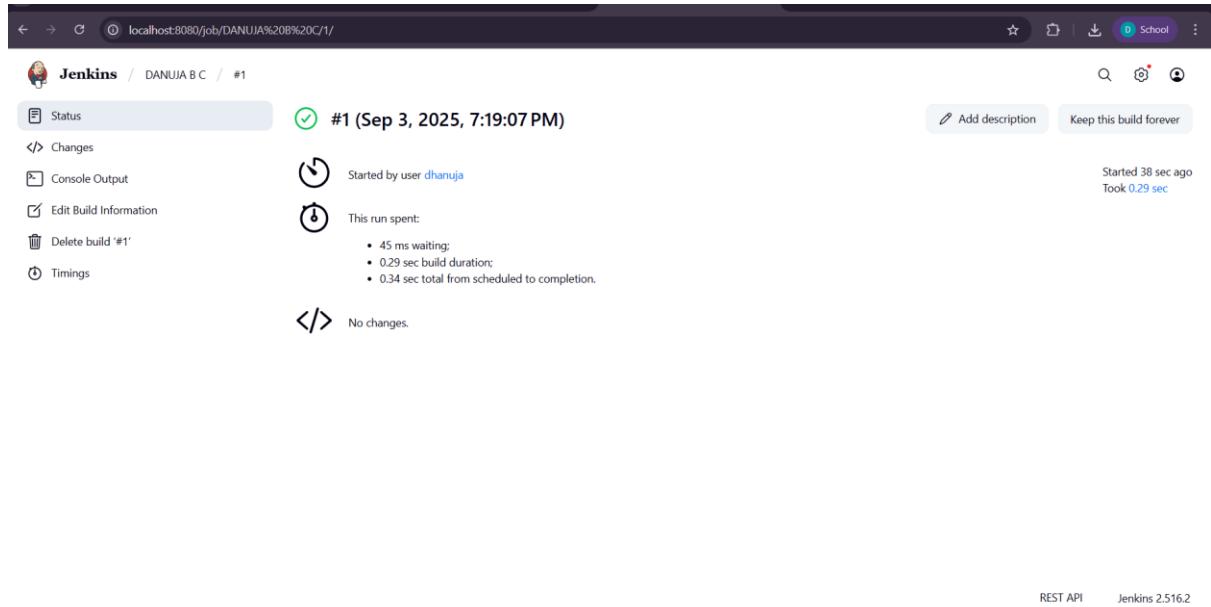


The screenshot shows the Jenkins Build Steps configuration page for the same job. Under the "Build Steps" section, a dropdown menu titled "Add build step" is open, showing options like "Execute Windows batch command", "Execute shell", "Invoke Ant", "Invoke Gradle script", "Invoke top-level Maven targets", and "Run with timeout". A tooltip for "Execute Windows batch command" states: "Automate your build process with ordered tasks like code compilation, testing, and deployment." The Jenkins interface includes a header with tabs for "General", "Source Code Management", "Triggers", "Environment", "Build Steps", and "Post-build Actions". The "Build Steps" tab is currently selected. The status bar at the bottom shows weather information (26°C, Mostly cloudy), system icons, and the date/time (03-09-2025).

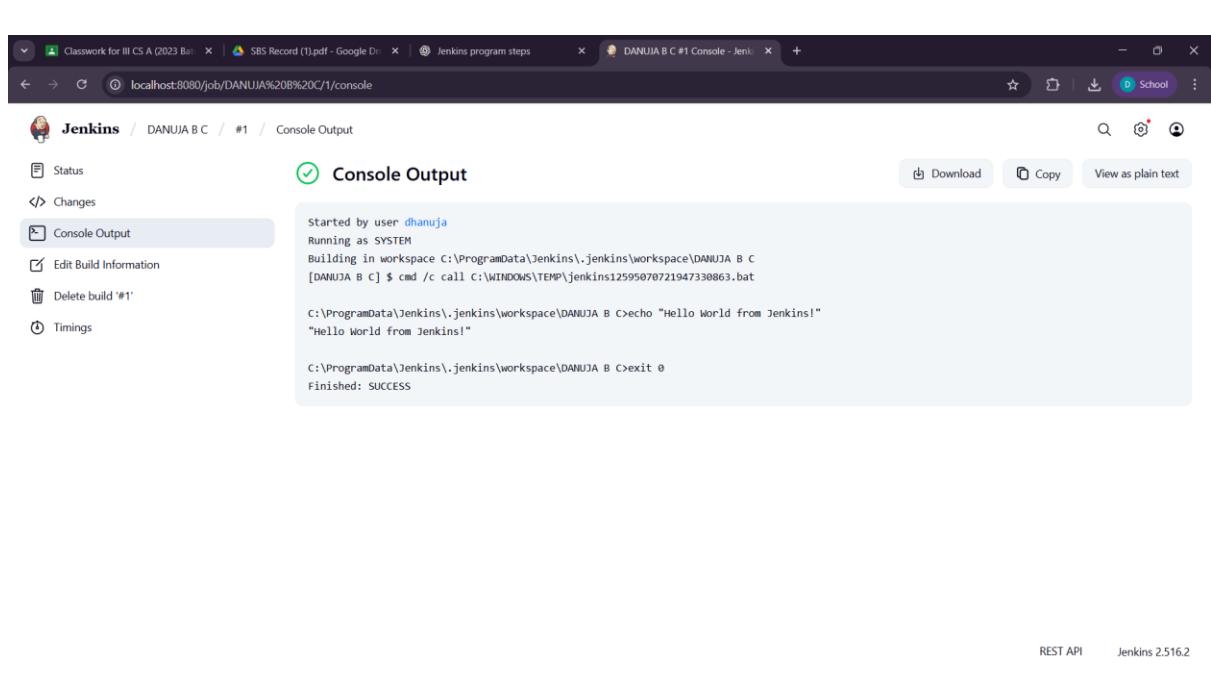
The screenshot shows the Jenkins configuration interface for a job named "DANUJA B C". On the left, a sidebar lists "General", "Source Code Management", "Triggers", "Environment", "Build Steps" (which is selected), and "Post-build Actions". The main area is titled "Build Steps" with the sub-section "Execute Windows batch command". It contains a command box with the text "echo \"Hello World from Jenkins!\"". Below the command box are "Advanced" and "Add build step" buttons.

The screenshot shows the Jenkins job status page for "DANUJA B C". The left sidebar includes "Status", "Changes", "Workspace", "Build Now", "Configure", "Delete Project", and "Rename". The main content area displays the job name "DANUJA B C" and the message "This is my first Jenkins job". Below this is a "Permalinks" section. A "Builds" card shows one build entry: "#1 7:19 PM". At the bottom right, there are links for "Edit description", "REST API", and "Jenkins 2."

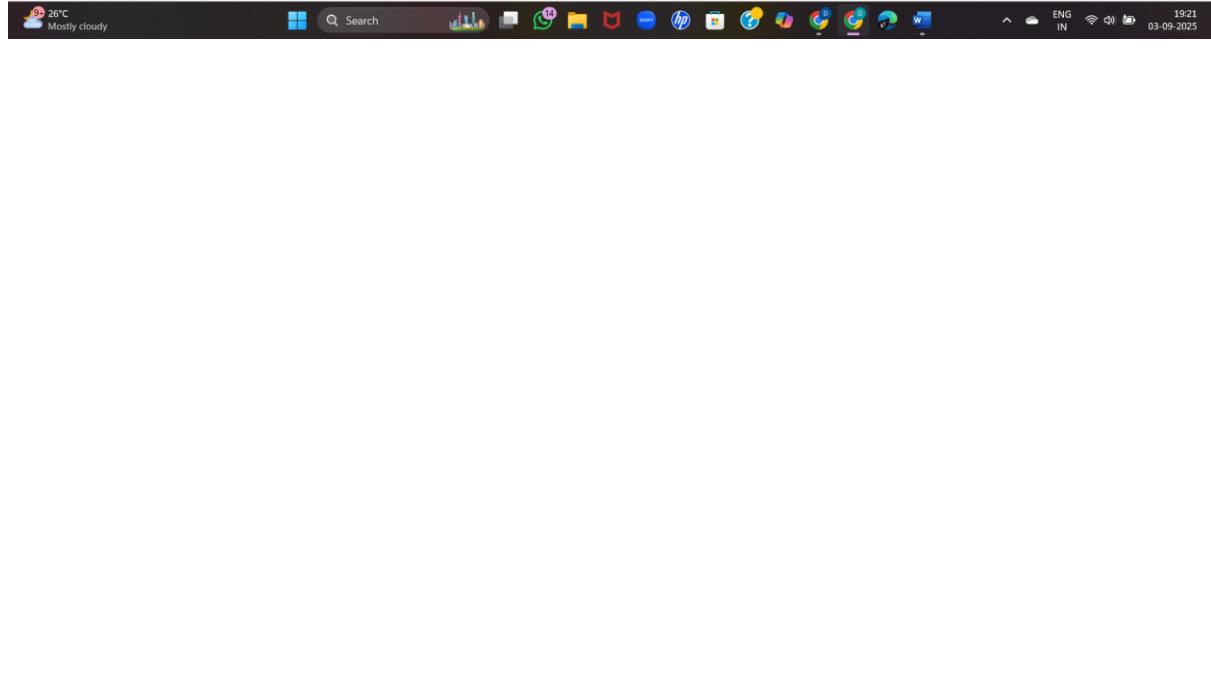
OUTPUT:



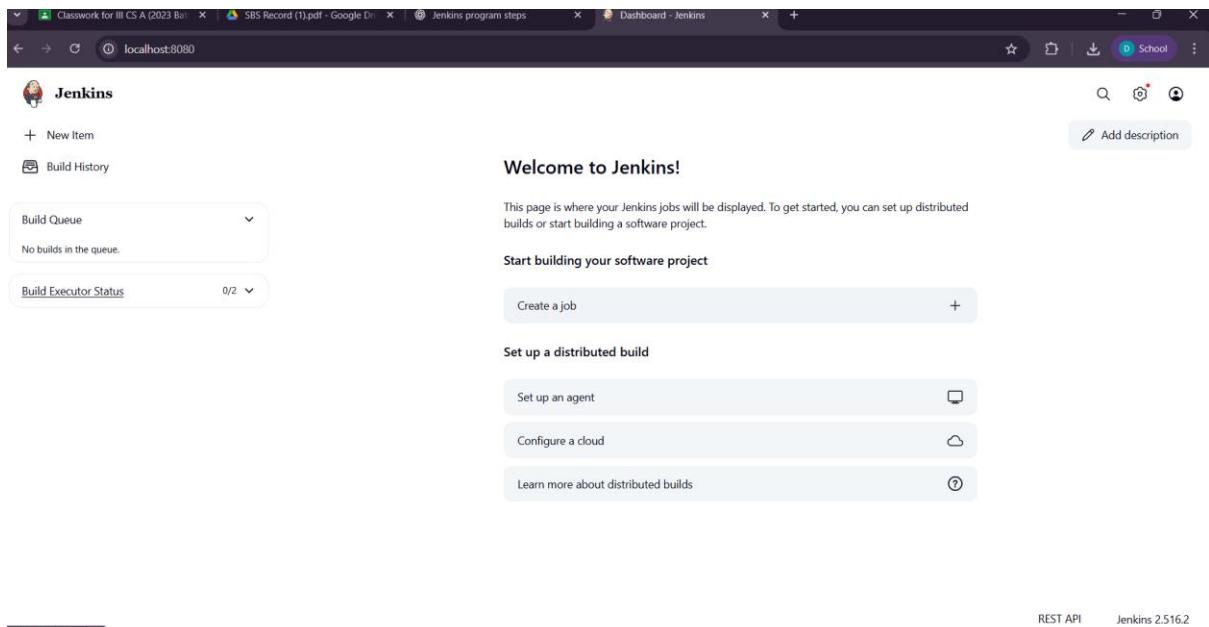
A screenshot of a web browser showing a Jenkins job build summary. The URL is `localhost:8080/job/DANUJA%20B%20C/1/`. The build is labeled '#1 (Sep 3, 2025, 7:19:07 PM)'. It was started by user `dhanuja`. The build spent 45 ms waiting, 0.29 sec on build duration, and 0.34 sec total from scheduled to completion. There were no changes made.



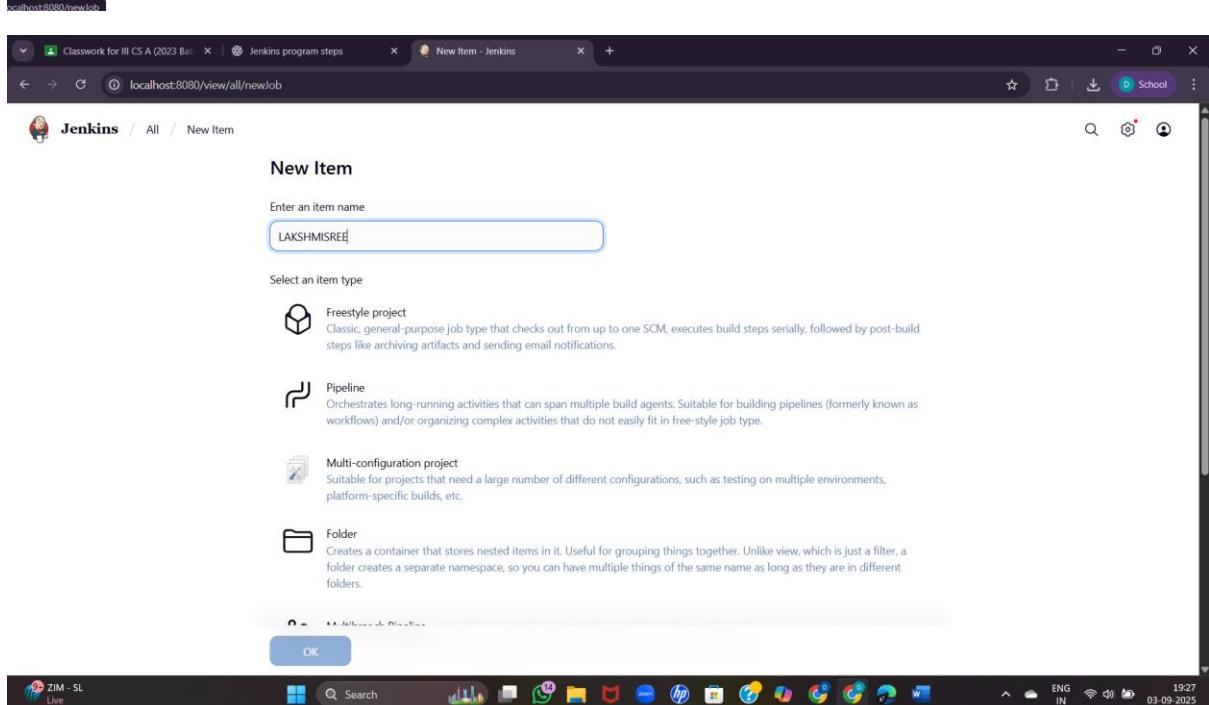
A screenshot of a web browser showing the Jenkins console output for build #1. The URL is `localhost:8080/job/DANUJA%20B%20C/1/console`. The output shows the build was started by user `dhanuja` and ran as SYSTEM. It built in workspace `C:\ProgramData\Jenkins\.jenkins\workspace\Danuja B C` and executed the command `cmd /c call c:\Windows\TEMP\jenkins12595070721947330863.bat`. The output includes the message "Hello World from Jenkins!" and ends with "Finished: SUCCESS".



PROGRAM 5: CREATING SCRIPTED PIPELINE PROJECT



The screenshot shows the Jenkins dashboard at localhost:8080. The main header says "Welcome to Jenkins!". Below it, a message states: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." A "Create a job" button is prominently displayed. On the left, there are sections for "Build Queue" (empty) and "Build Executor Status" (0/2). On the right, there's a "Set up a distributed build" section with links for "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". The top right corner shows "REST API" and "Jenkins 2.516.2".



The screenshot shows the "New Item" dialog in Jenkins. The title bar says "localhost:8080/view/all/newJob". The main area is titled "New Item" and asks "Enter an item name". A text input field contains "LAKSHMISREE". Below it, a "Select an item type" section lists four options: "Freestyle project" (selected), "Pipeline", "Multi-configuration project", and "Folder". Each option has a description and a small icon. At the bottom is an "OK" button. The taskbar at the bottom includes icons for ZIM-SL, Search, File Explorer, Task View, Control Panel, HP, Google Chrome, and Microsoft Word. The system tray shows "ENG IN", "19:27", and the date "03-09-2025".

Jenkins / LAKSHMISREE / Configuration

Configure

General

Description: Scripted Pipeline Project

Plain text [Preview](#)

Discard old builds ?
 Do not allow concurrent builds
 Do not allow the pipeline to resume if the controller restarts
 GitHub project
 Pipeline speed/durability override ?
 Preserve stashes from completed builds ?
 This project is parameterized ?

Save **Apply**

Enabled

Jenkins program steps | Classwork for III CS A (2023 Ba... | LAKSHMISREE Config - Jenkins | localhost:8080/job/LAKSHMISREE/config | + | School | ...

Jenkins / LAKSHMISREE / Configuration

Configure

Pipeline

Definition: Pipeline script

Script ?

```
1v node {  
2v   stage('build') {  
3v     echo 'building...'  
4v   }  
5v   stage('deploy') {  
6v     echo 'deploying...'  
7v   }  
8v }
```

try sample Pipeline...

Use Groovy Sandbox ?

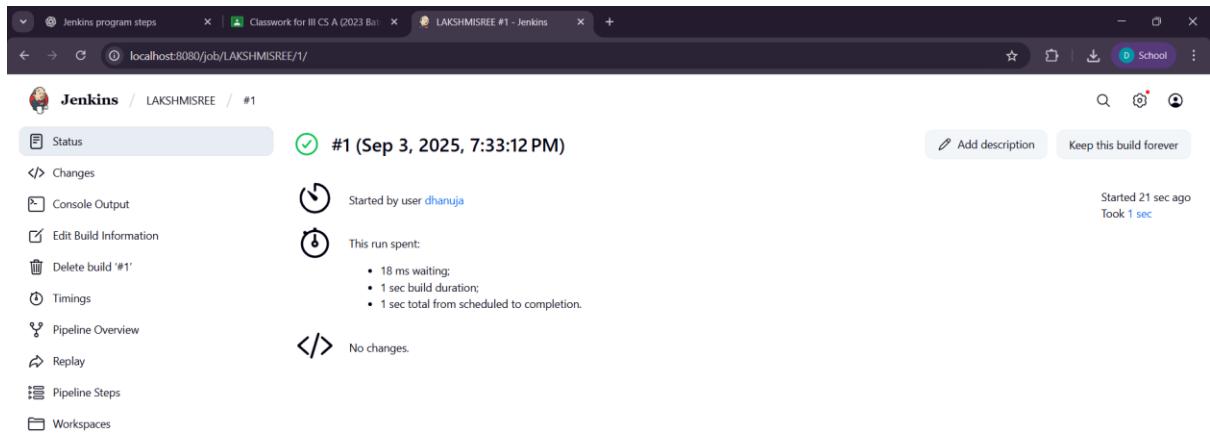
Pipeline Syntax

Save **Apply**

26°C Partly cloudy | Search |  ENG IN 19:32 03-09-2025

The screenshot shows a web browser window with the Jenkins LAKSHMISREE project page. The page title is "LAKSHMISREE" and it is described as a "Scripted Pipeline Project". On the left, there is a sidebar with links for Status, Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, and Pipeline Syntax. Below this is a "Builds" section showing one build from today at 7:33 PM. The system tray at the bottom of the screen displays the Windows taskbar with icons for weather (26°C Partly cloudy), search, file explorer, and other system utilities. It also shows the date (03-09-2025) and time (19:33).

OUTPUT :



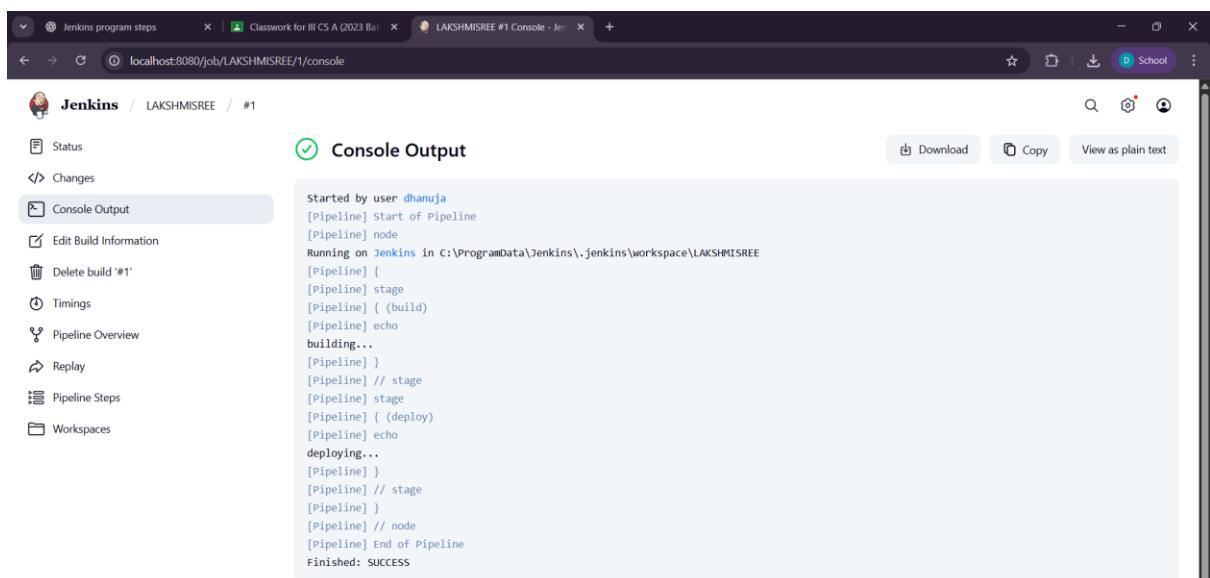
The screenshot shows the Jenkins job #1 status page. The build was started by user dhanuja on Sep 3, 2025, at 7:33:12 PM. It took 1 sec and completed 21 seconds ago. There are no changes or console output.

Started by user **dhanuja**
Started 21 sec ago
Took 1 sec

No changes.

Build History: #1 (Sep 3, 2025, 7:33:12 PM)

Actions: Status, Changes, Console Output, Edit Build Information, Delete build '#1', Timings, Pipeline Overview, Replay, Pipeline Steps, Workspaces.

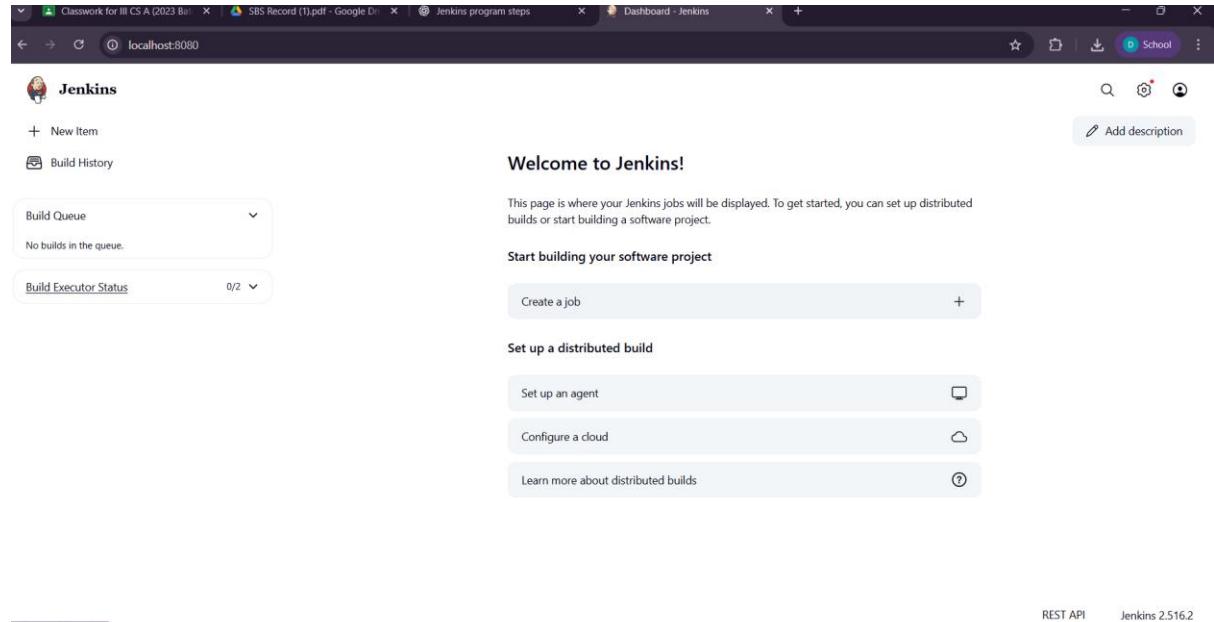


The screenshot shows the Jenkins job #1 console output. The pipeline starts, runs on Jenkins, and ends successfully.

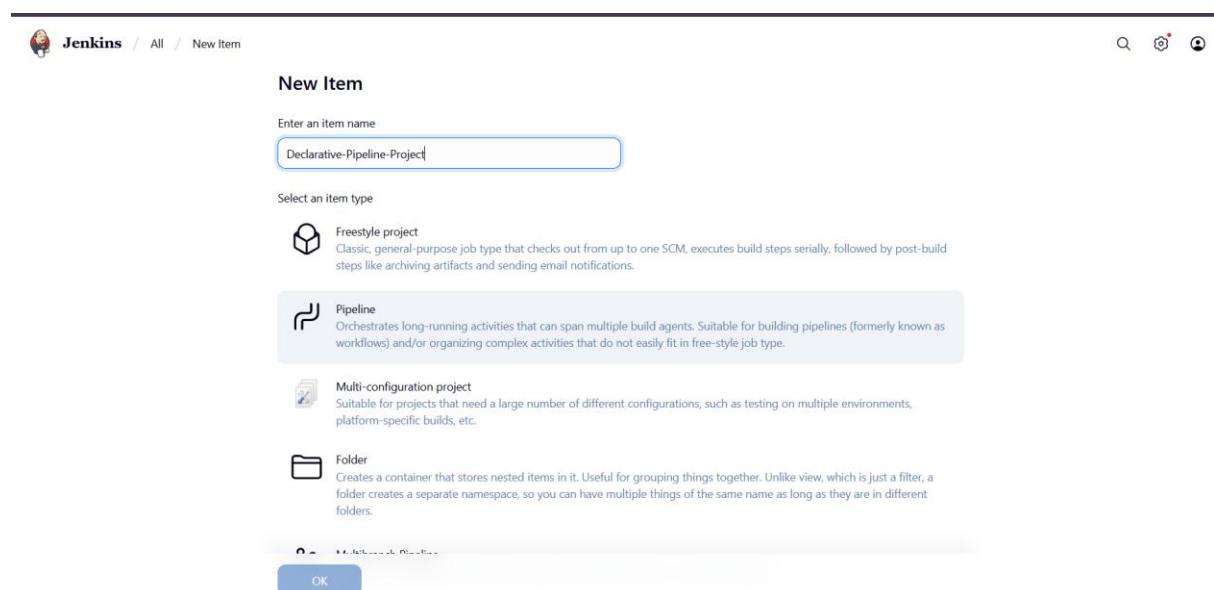
```
Started by user dhanuja
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\LAKSHMISREE
[Pipeline] {
[Pipeline] stage
[Pipeline] { (build)
[Pipeline] echo
building...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deploy)
[Pipeline] echo
deploying...
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



PROGRAM 6: CREATING DECLARATIVE PIPELINE PROJECT



The screenshot shows the Jenkins dashboard at localhost:8080. The main header says "Welcome to Jenkins!". Below it, there's a section titled "Start building your software project" with a "Create a job" button. To the left, there are two dropdown menus: "Build Queue" (No builds in the queue) and "Build Executor Status" (0/2). On the right, there are three main sections: "Set up a distributed build" (with links to "Set up an agent", "Configure a cloud", and "Learn more about distributed builds"), "Manage agents" (with a "Manage" link), and "Manage clouds" (with a "Manage" link). At the bottom, it says "localhost:8080/new-job" in the address bar, "REST API" and "Jenkins 2.516.2" in the footer, and has a search and refresh icon.



The screenshot shows the "New Item" dialog box. At the top, it says "Enter an item name" with the value "Declarative-Pipeline-Project". Below that, it says "Select an item type" and lists four options: "Freestyle project" (selected), "Pipeline" (highlighted with a blue border), "Multi-configuration project", and "Folder". Each option has a brief description. At the bottom of the dialog is an "OK" button.

Jenkins / Declarative-Pipeline-Project / Configuration

Configure General

Description

This is a Jenkins Declarative Pipeline project!

Plain text [Preview](#)

Discard old builds ?

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

GitHub project

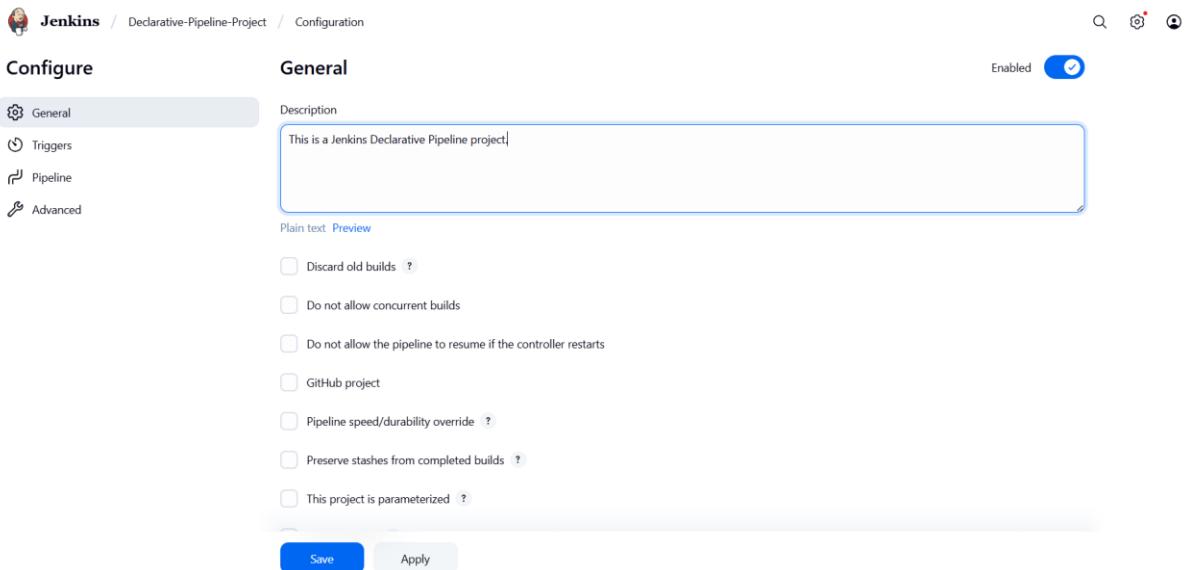
Pipeline speed/durability override ?

Preserve stashes from completed builds ?

This project is parameterized ?

Enabled

Save Apply



Jenkins / prgm6 / Configuration

Configure Pipeline

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/dhanuja1709/SBS-lab.git

Credentials ?

- none -

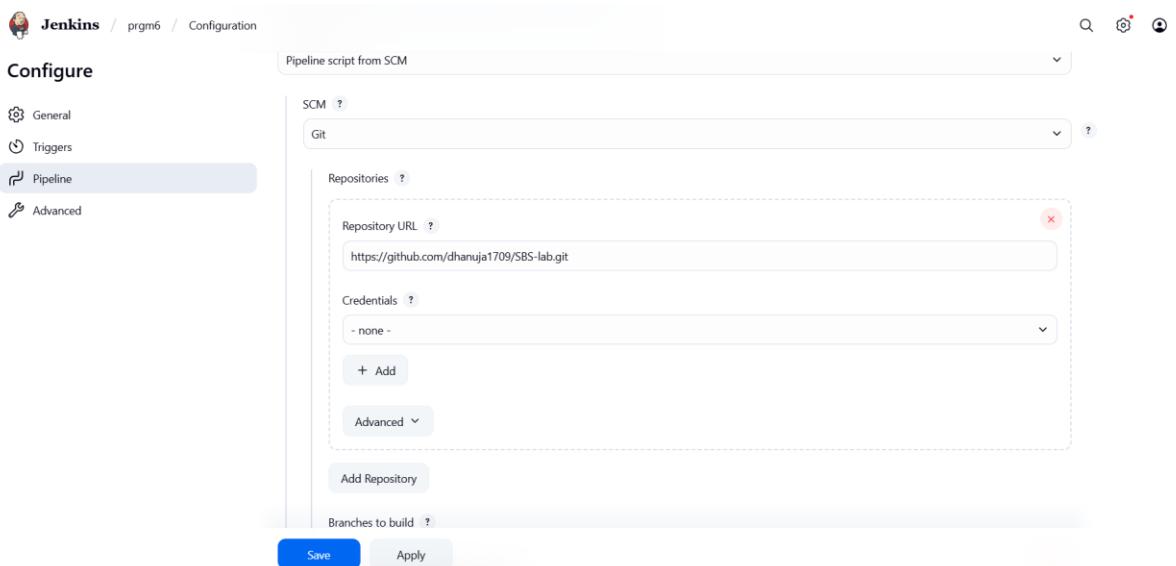
+ Add

Advanced ▾

Add Repository

Branches to build ?

Save Apply



Jenkins / Declarative-Pipeline-Project / Configuration

Configure Advanced

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

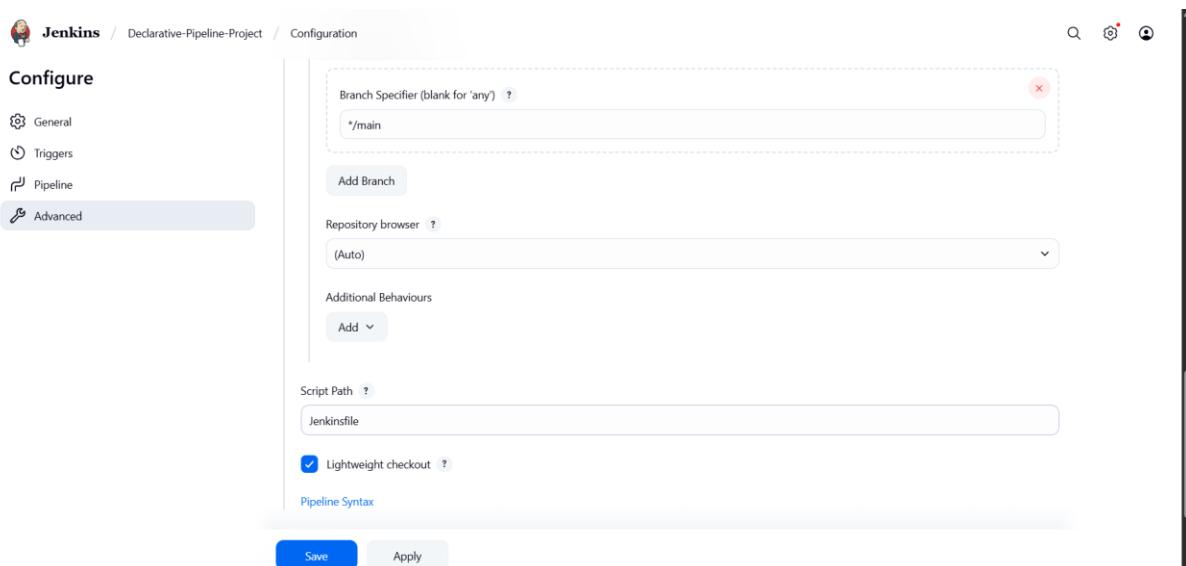
Script Path ?

Jenkinsfile

Lightweight checkout ?

[Pipeline Syntax](#)

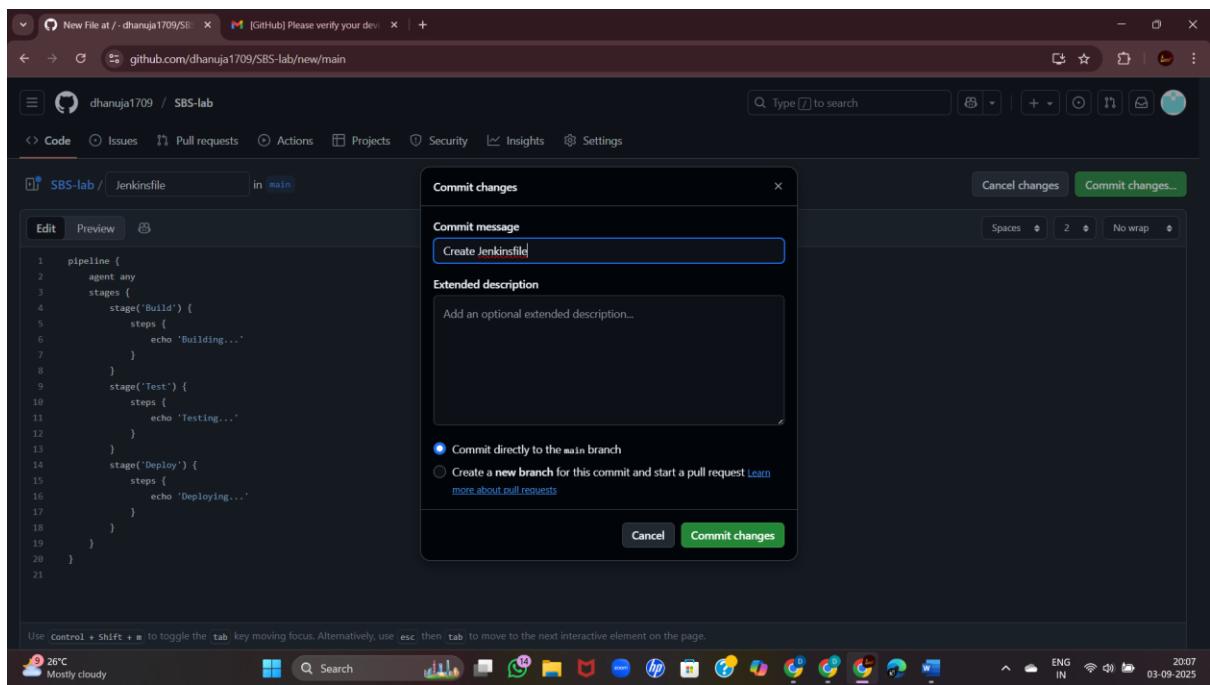
Save Apply



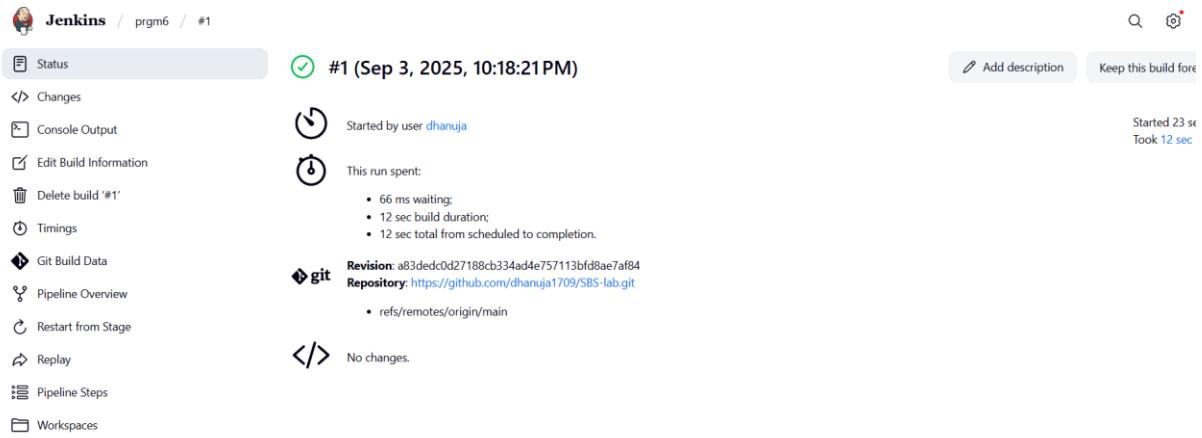
A screenshot of a web browser displaying a GitHub repository page. The URL is github.com/dhanuja1709/SBS-lab. The page shows the `Jenkinsfile` in the `main` branch. The code content is as follows:

```
1 pipeline {
2   agent any
3   stages {
4     stage('Build') {
5       steps {
6         echo 'Building...'
7       }
8     }
9     stage('Test') {
10    steps {
11      echo 'Testing...'
12    }
13  }
14  stage('Deploy') {
15    steps {
16      echo 'Deploying...'
17    }
18  }
19 }
20 }
```

The GitHub interface includes standard navigation buttons like back, forward, and search, along with repository-specific tabs for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. A toolbar at the top right provides options for canceling or committing changes.



OUTPUT :



The screenshot shows the Jenkins interface for a build named '#1' (Sep 3, 2025, 10:18:21 PM). The build was started by user 'dhanuja'. It took 12 seconds and completed 23 seconds ago. The pipeline overview shows a single stage with no changes or workspace modifications.

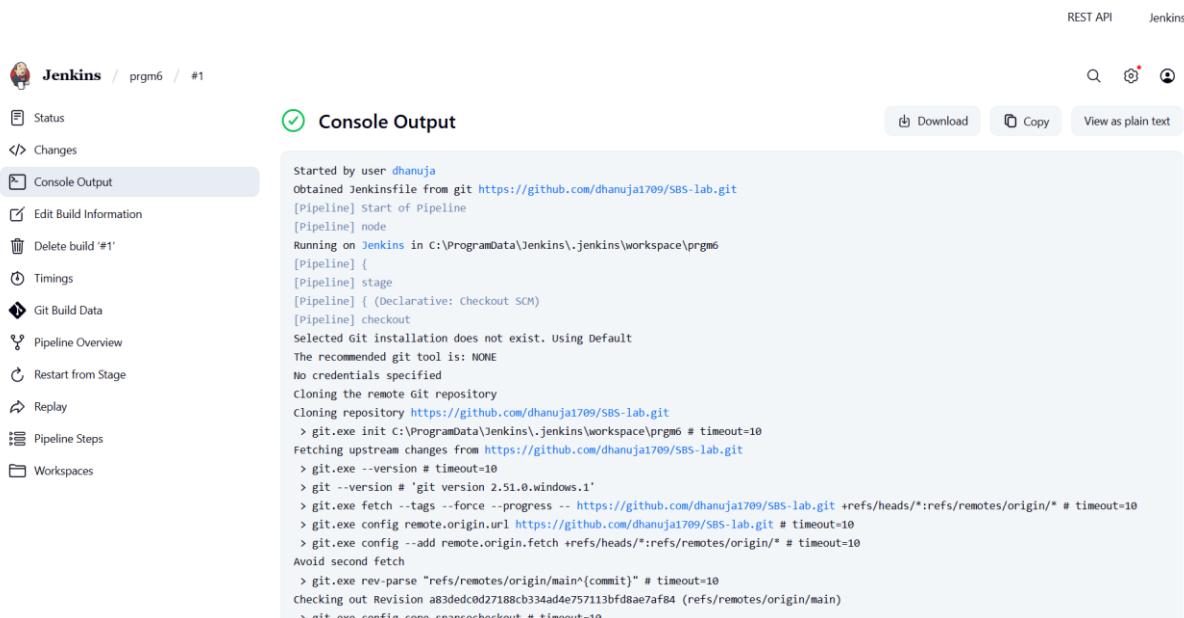
Build Information:
Started by user [dhanuja](#)
This run spent:

- 66 ms waiting;
- 12 sec build duration;
- 12 sec total from scheduled to completion.

Revision: a83dedc0d27188cb334ad4e757113bfd8ae7af84
Repository: <https://github.com/dhanuja1709/SBS-lab.git>
refs/remotes/origin/main

Console Output:
</> No changes.

Build Steps:
Status, Changes, Console Output (selected), Edit Build Information, Delete build '#1', Timings, Git Build Data, Pipeline Overview, Restart from Stage, Replay, Pipeline Steps, Workspaces.



The screenshot shows the Jenkins console output for build '#1'. The output details the execution of a Jenkinsfile from a GitHub repository. It includes the start of the pipeline, node selection, and the execution of a declarative pipeline script. The log shows the cloning of the repository, fetching upstream changes, and performing a git checkout to revision a83dedc0d27188cb334ad4e757113bfd8ae7af84.

```
Started by user dhanuja
Obtained Jenkinsfile from git https://github.com/dhanuja1709/SBS-lab.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\prgm6
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/dhanuja1709/SBS-lab.git
> git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\prgm6 # timeout=10
Fetching upstream changes from https://github.com/dhanuja1709/SBS-lab.git
> git.exe --version # timeout=10
> git --version # 'git version 2.51.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/dhanuja1709/SBS-lab.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/dhanuja1709/SBS-lab.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision a83dedc0d27188cb334ad4e757113bfd8ae7af84 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
```



Jenkins / prgm6 / #1



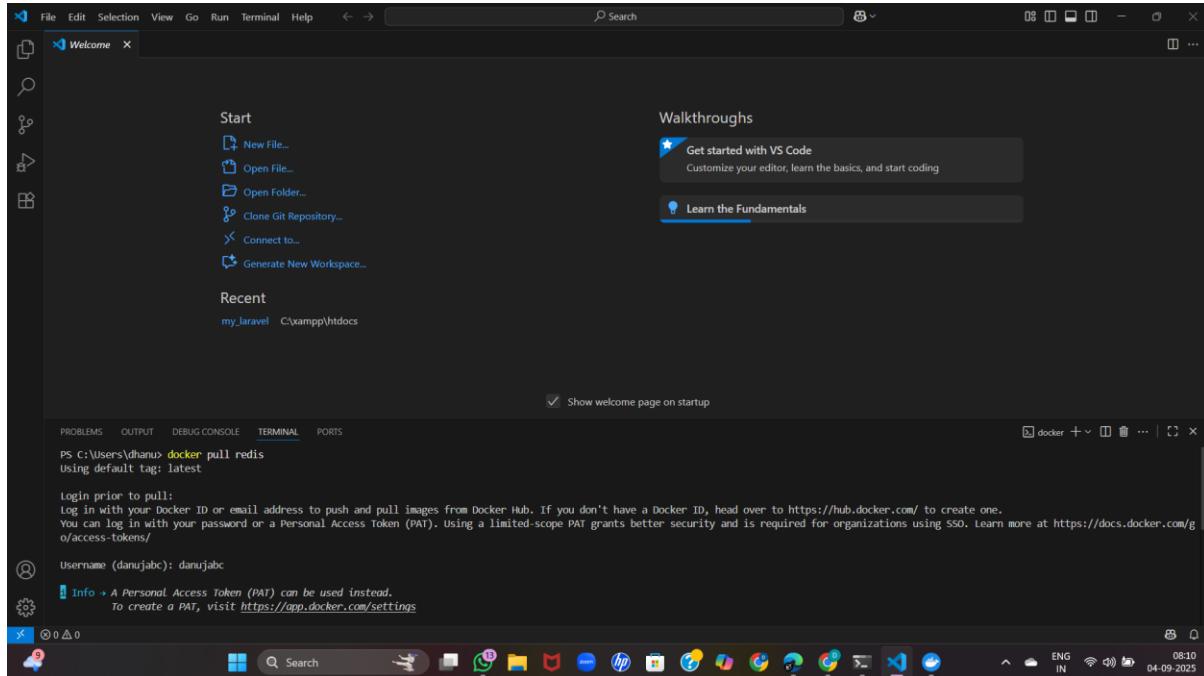
```
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision a83dedc0d27188cb334ad4e757113bfd8ae7af84 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f a83dedc0d27188cb334ad4e757113bfd8ae7af84 # timeout=10
Commit message: "Create Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Building...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] echo
Testing...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] echo
Deploying...
[Pipeline] }
Final Pipeline // et...aa
```



Jenkins / prgm6 / #1

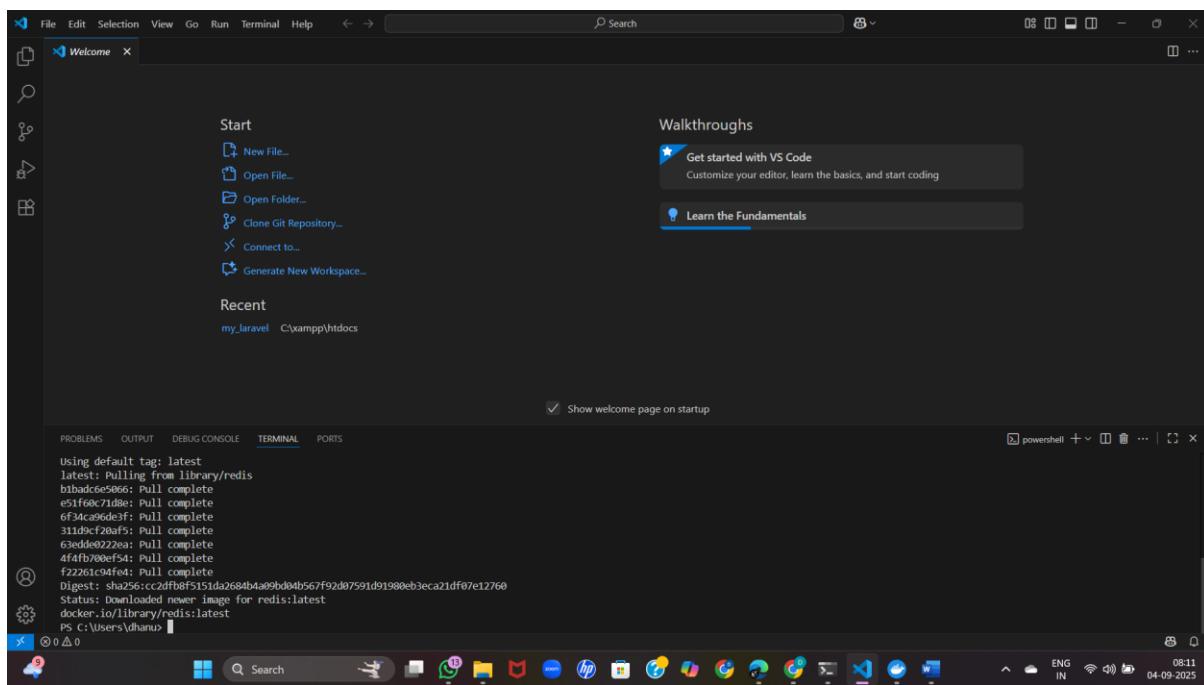
```
[Pipeline] { (Build)
[Pipeline] echo
Building...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] echo
Testing...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] echo
Deploying...
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

PROGRAM 7: PULL , RUN AND DELETE A DOCKER IMAGE IN WINDOWS



```
PS C:\Users\dharu> docker pull redis
Using default tag: latest

Login prior to pull:
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/g
o/access-tokens/
```



```
Using default tag: latest
Latest: Pulling from library/redis
b1badc6e5066: Pull complete
e51f66c71d8e: Pull complete
6f34ca96de1f: Pull complete
311d9cf20af5: Pull complete
63edded0226a: Pull complete
4f4fb700ef54: Pull complete
f22261c94fe4: Pull complete
Digest: sha256:c2dfb8f5151da2684b4a09bd04b567f92d07591d91980eb3eca21df07e12760
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\Users\dharu>
```

```
my_laravel C:\xampp\htdocs

Show welcome page on startup

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Digest: sha256:cc2dfb8f5151da2684b4a09bd04b567f92d07591d91980eb3eca21df07e12760
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
PS C:\Users\dhanu> docker run -d --name my-redis -p 6379:6379 redis
>>
052fa9a843440c3b3bd1694272abda02a5db3ed975cb4c0eb7e6f46a071f01db
PS C:\Users\dhanu> docker ps
>>
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
052fa9a84344 redis "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp my-redis
PS C:\Users\dhanu> docker exec -it my-redis redis-cli
>>
127.0.0.1:6379> set name "danu lakshmi"
```

```
Show welcome page on startup

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

>>
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
052fa9a84344 redis "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp my-redis
PS C:\Users\dhanu> docker exec -it my-redis redis-cli
>>
127.0.0.1:6379> set name "danu lakshmi"
OK
127.0.0.1:6379> get name
"danu lakshmi"
127.0.0.1:6379> exit
PS C:\Users\dhanu> docker stop my-redis
>> docker rm my-redis
>>
```

```
Show welcome page on startup

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\dhanu> docker stop my-redis
>> docker rm my-redis
>>
my-redis
my-redis
PS C:\Users\dhanu> docker rmi redis
>>
Untagged: redis:latest
Deleted: sha256:cc2dfb8f5151da2684b4a09bd04b567f92d07591d91980eb3eca21df07e12760
PS C:\Users\dhanu> docker images
>>
REPOSITORY TAG IMAGE ID CREATED SIZE
PS C:\Users\dhanu>
```

```
Show welcome page on startup

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

>>
my-redis
my-redis
PS C:\Users\dhanu> docker rmi redis
>>
Untagged: redis:latest
Deleted: sha256:cc2dfb8f5151da2684b4a09bd04b567f92d07591d91980eb3eca21df07e12760
PS C:\Users\dhanu> docker images
>>
REPOSITORY TAG IMAGE ID CREATED SIZE
PS C:\Users\dhanu> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS C:\Users\dhanu>
```

Reference :

Program: Pull, Run, Test, and Delete a Docker Image (Redis)

Aim

To demonstrate the process of pulling, running, testing, stopping, and deleting a Docker image (Redis) using Windows, Docker Desktop, and VS Code.

Algorithm

1. Start Docker Desktop and open VS Code terminal (PowerShell).
2. Login to Docker Hub
3. **docker login**
 - Enter Docker Hub username and password (or Personal Access Token if 2FA enabled).
4. Pull Redis image from Docker Hub
docker pull redis
5. Verify image is pulled
6. **docker images**
 - Redis should appear in the list.
7. Run a Redis container
docker run -d --name my-redis -p 6379:6379 redis
8. Check running containers
9. **docker ps**
 - my-redis should be running.
10. Access Redis CLI inside container
11. **docker exec -it my-redis redis-cli**
 - Run test commands:
 - **set name "danu lakshmi"**
 - **get name**
 - **exit**
 - Output should confirm data is stored and retrieved.
12. Stop the container
13. **docker stop my-redis**

14. Remove the container

15. docker rm my-redis

16. Delete the Redis image

17. docker rmi redis

18. Verify removal

19. docker images

20. docker ps -a

- Both container and image lists should be empty.
-

Docker Commands & Meaning

- docker login → login to Docker Hub
- docker pull redis → download Redis image
- docker images → list all downloaded images
- docker run -d --name my-redis -p 6379:6379 redis → run Redis container in background
- docker ps → show running containers
- docker exec -it my-redis redis-cli → open Redis CLI inside container
- set name "danu lakshmi" → stores data in Redis
- get name → retrieves stored data
- exit → leave Redis CLI
- docker stop my-redis → stop the container
- docker rm my-redis → remove the container
- docker rmi redis → delete the Redis image
- docker ps -a → show all containers (running + stopped)