

Practical - 4

Collaborating and Cloning Using GitHub

Step 1. Clone a public repository:

```
git clone https://github.com/bhargavikeni12/Demo.git
```

```
cd Demo
```

```
HP@DESKTOP-PNISSJS MINGW64 ~
$ git clone https://github.com/bhargavikeni12/Demo.git
Cloning into 'Demo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

HP@DESKTOP-PNISSJS MINGW64 ~
$ cd Demo
```

Step 2: Create a New Branch

```
git checkout -b update-readme
```

```
HP@DESKTOP-PNISSJS MINGW64 ~/Demo (main)
$ git checkout -b update-readme
Switched to a new branch 'update-readme'

HP@DESKTOP-PNISSJS MINGW64 ~/Demo (update-readme)
$ echo "Added a line for practical" >> README.md

HP@DESKTOP-PNISSJS MINGW64 ~/Demo (update-readme)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
```

```
HP@DESKTOP-PNISSJS MINGW64 ~/Demo (update-readme)
$ git config --global user.name "Bhargavi Keni"

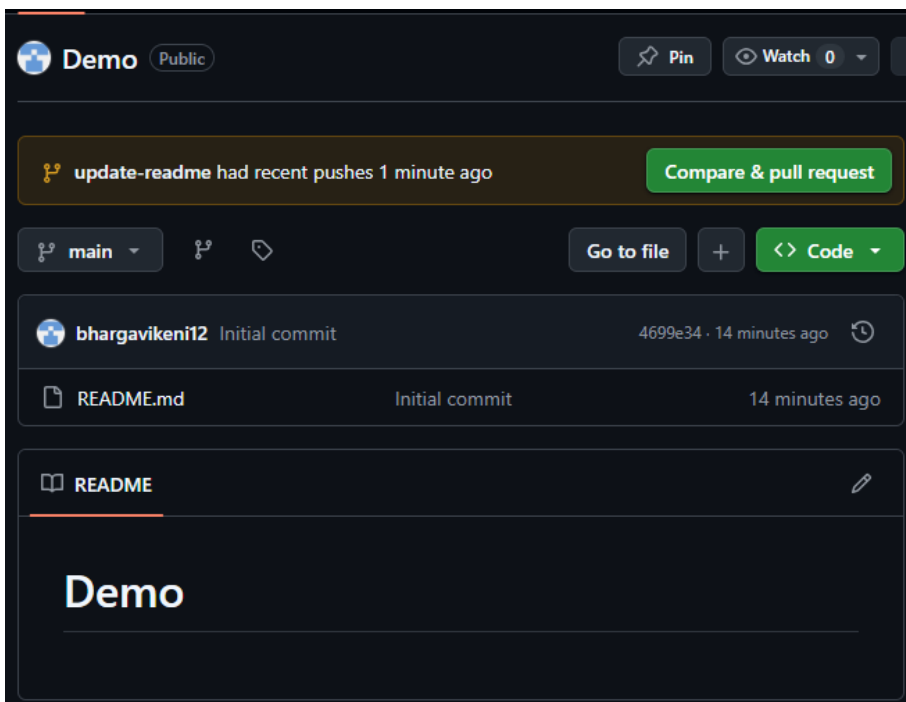
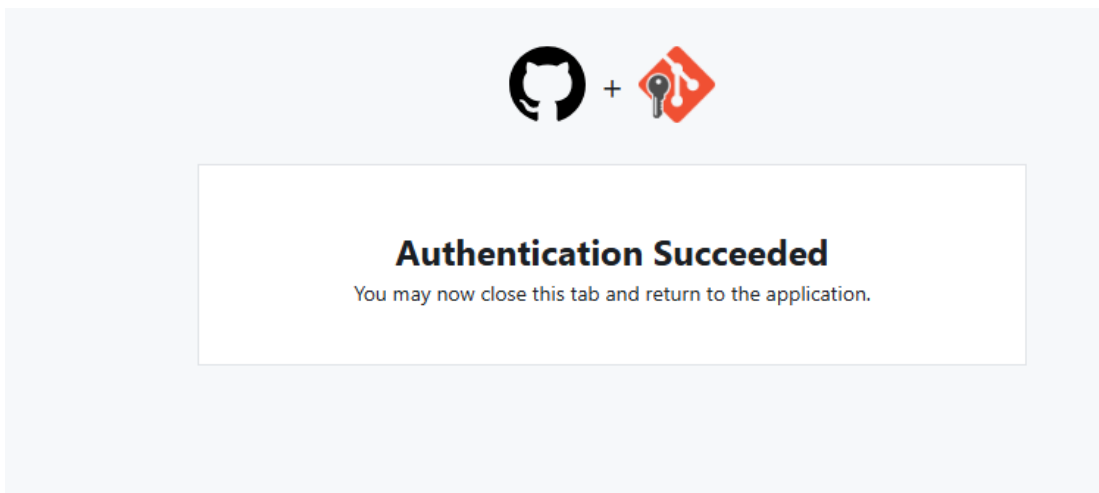
HP@DESKTOP-PNISSJS MINGW64 ~/Demo (update-readme)
$ git config --global user.email "bhargavikeni12@gmail.com"

HP@DESKTOP-PNISSJS MINGW64 ~/Demo (update-readme)
$ git commit -m "Updated README for practical"
[update-readme 4586019] Updated README for practical
1 file changed, 1 insertion(+), 1 deletion(-)
```

Step 3: Push and Open pull request:

`git push origin update-readme`

```
HP@DESKTOP-PNISSJS MINGW64 ~/Demo (update-readme)
$ git push origin update-readme
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'update-readme' on GitHub by visiting:
remote:   https://github.com/bhargavikeni12/Demo/pull/new/update-readme
remote:
To https://github.com/bhargavikeni12/Demo.git
 * [new branch]      update-readme -> update-readme
```



Practical No – 5

Using GitLab Web IDE

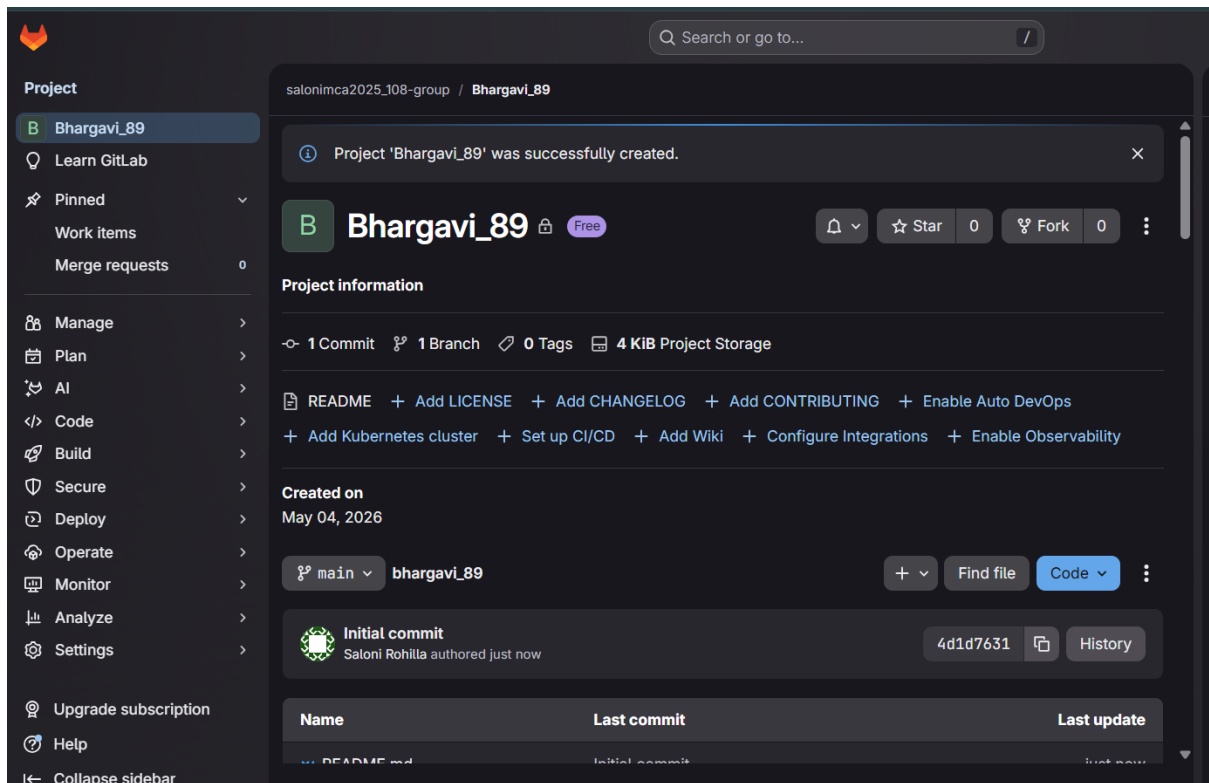
GitLab Web IDE

Steps:

1. Sign up at <https://gitlab.com>
2. Create a project.
3. Click on Web IDE in your repository.

The screenshot displays the GitLab interface for creating a new project. On the left, a sidebar lists navigation options: Home, Projects (selected), Groups, Work items, Merge requests, To-Do List, Milestones, Snippets, Activity, Import history, Workspaces, Environments, Operations, and Security. Below these are links for Upgrade subscription, Help, and Collapse sidebar. The main content area is titled 'Create blank project' and includes a search bar at the top. The breadcrumb trail reads 'Your work / Projects / New project / Create blank project'. The form fields are as follows:

- Project name:** A text input field containing 'Bhargavi_89'. Below it, a note states: 'Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.'
- Project URL:** A text input field containing 'https://gitlab.com/'. To its right is a dropdown menu labeled 'Pick a group or namespace'.
- Project slug:** A text input field containing 'bhargavi_89'.
- Project deployment target (optional):** A dropdown menu with the text 'Select the deployment target'.
- Visibility Level:** Three radio button options: 'Private' (with a lock icon), 'Internal' (with a lock icon and a person icon), and 'Public' (with a globe icon). The 'Public' option is selected. Below these options, a note states: 'The project can be accessed without any authentication.'
- Project Configuration:** A section with a checked checkbox labeled 'Initialize repository with a README'.



4. Create a file (index.html):

```
<html>
```

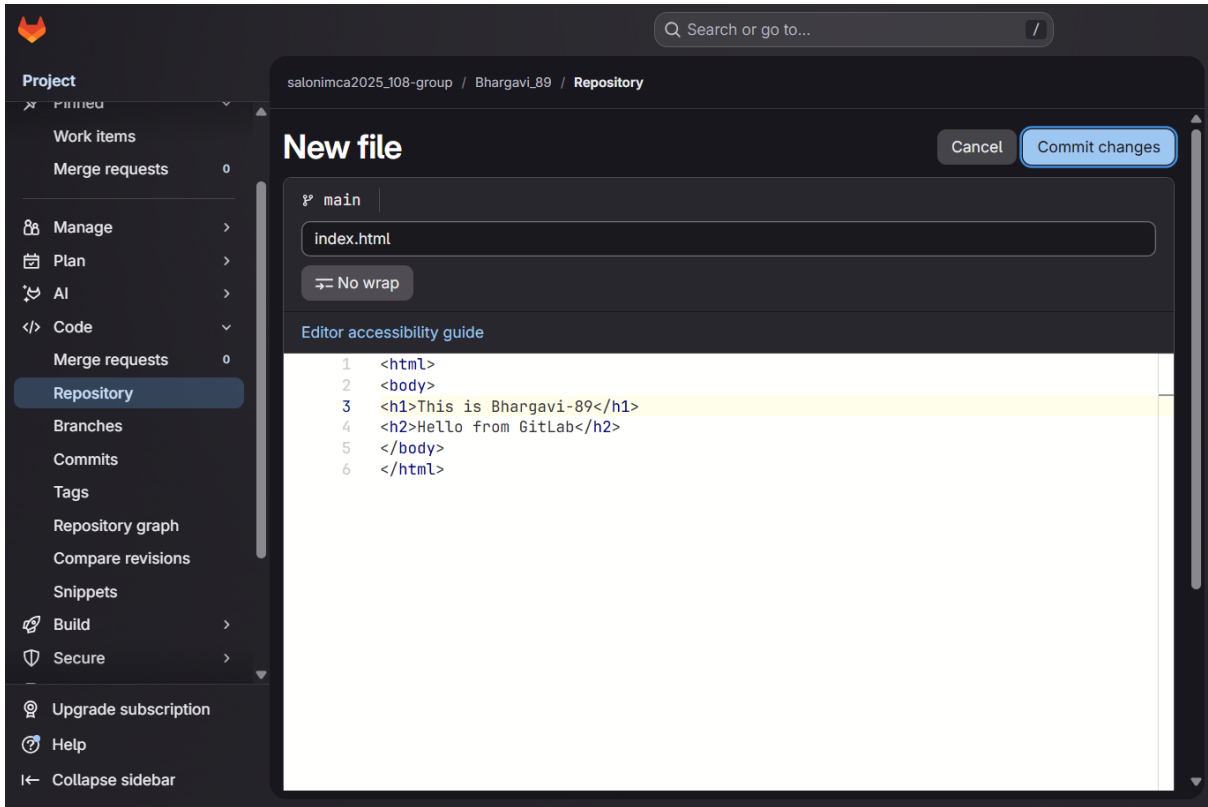
```
<body>
```

```
  <h1>Hello Prince</h1>
```

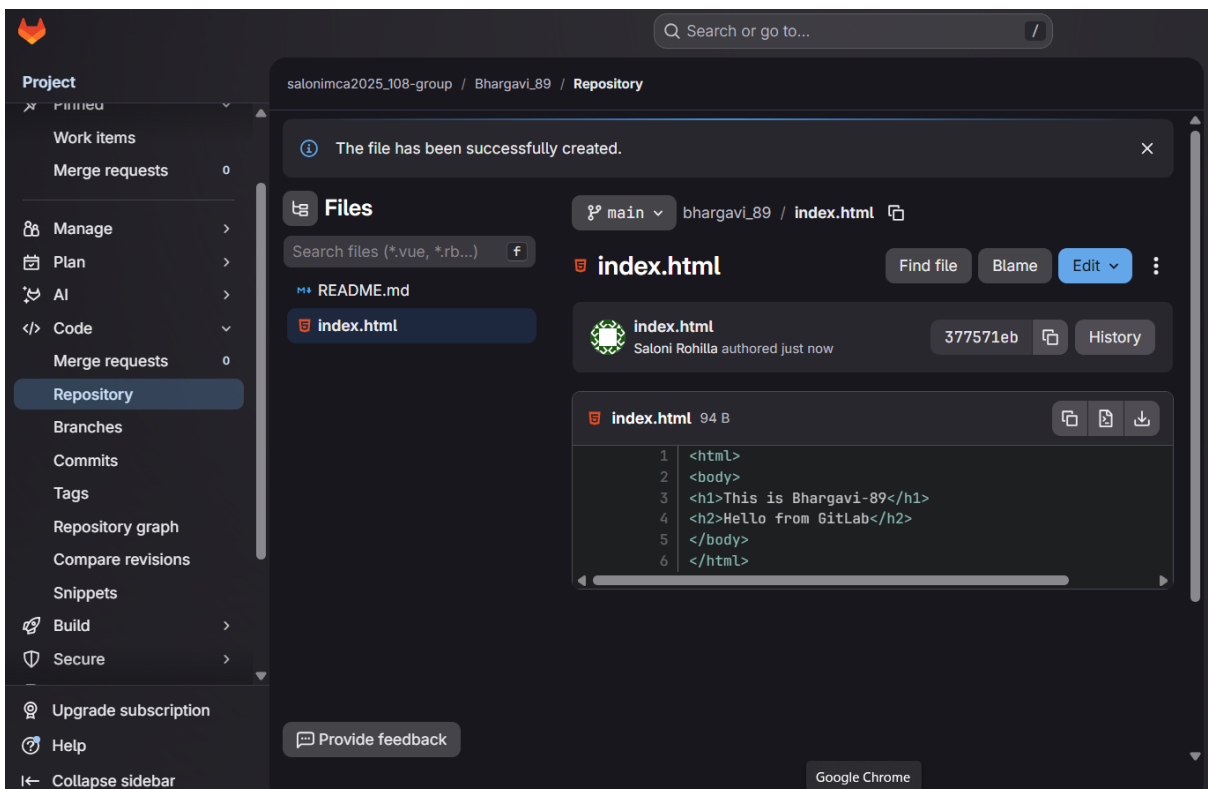
```
  <h2>Working on Gitlab</h2>
```

```
</body>
```

```
</html>
```



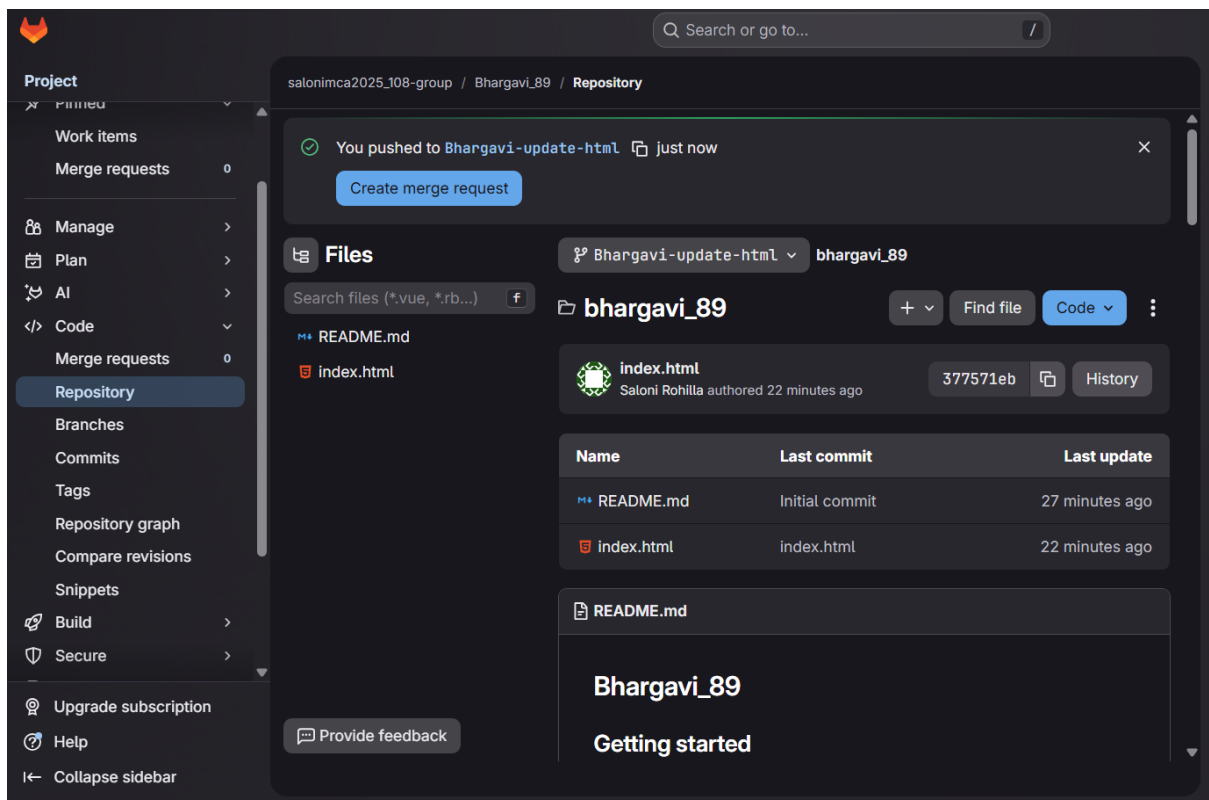
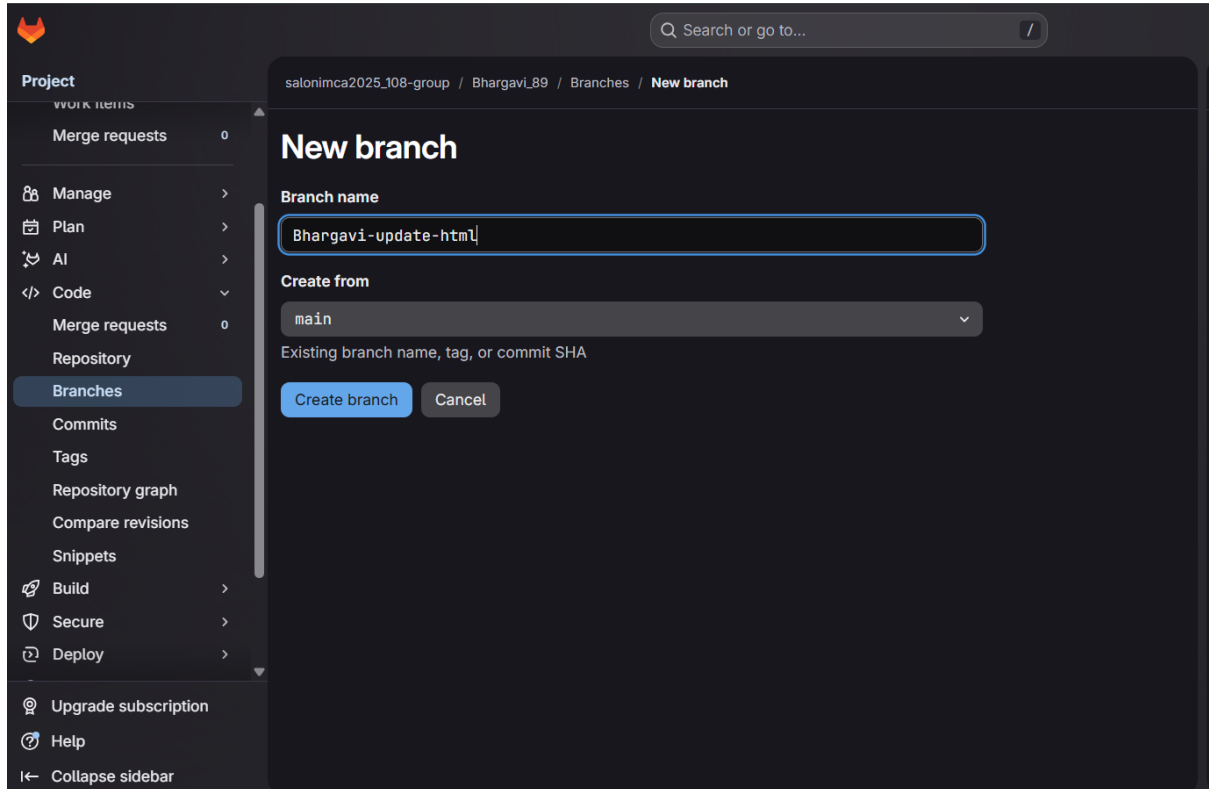
5. Click Commit and push changes



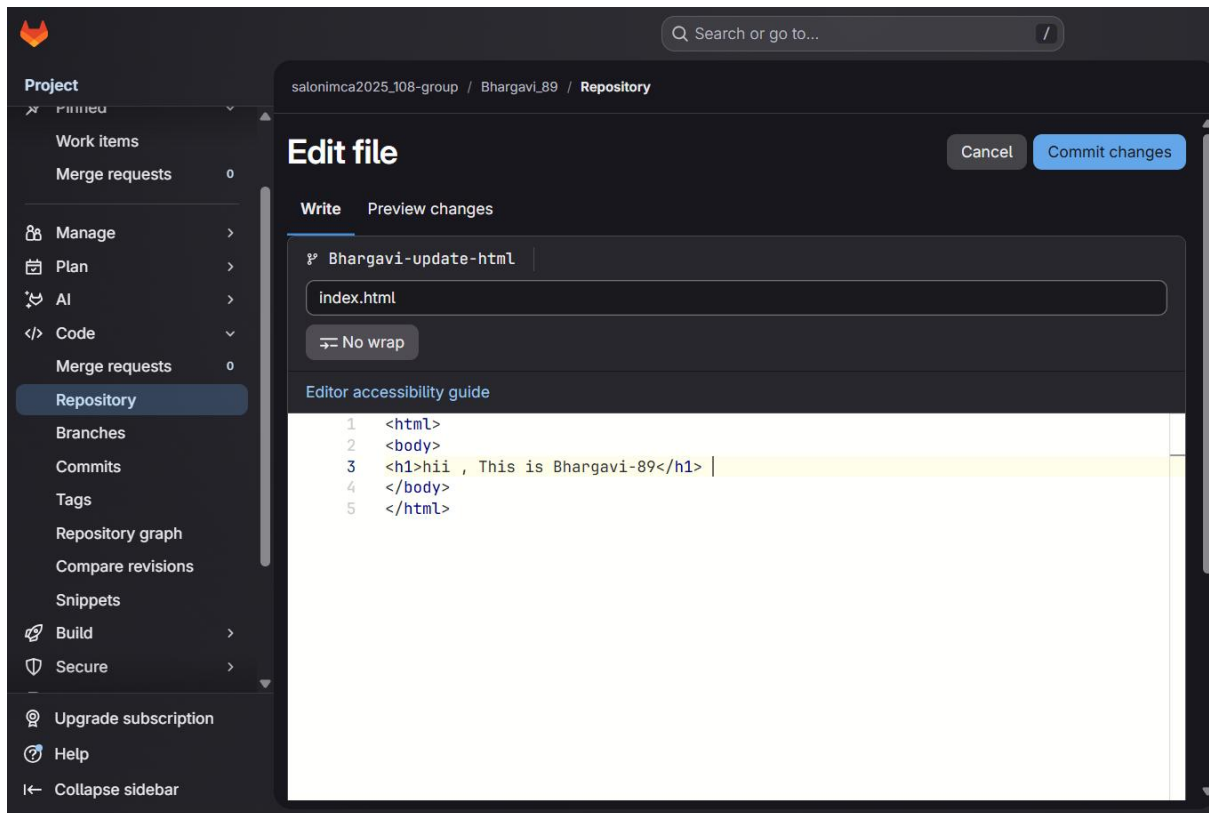
Practical – 6

Performing Merge Requests Using GitLab

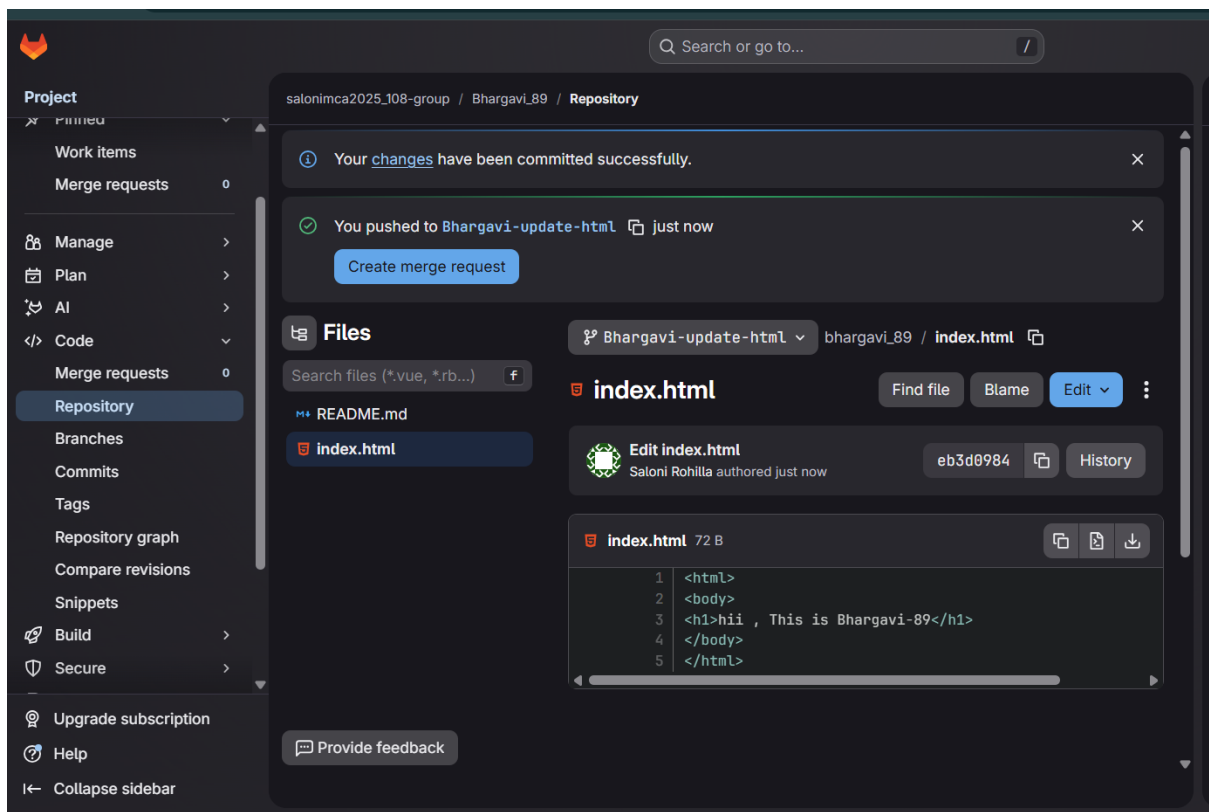
Step 1. Create a new branch in Web IDE

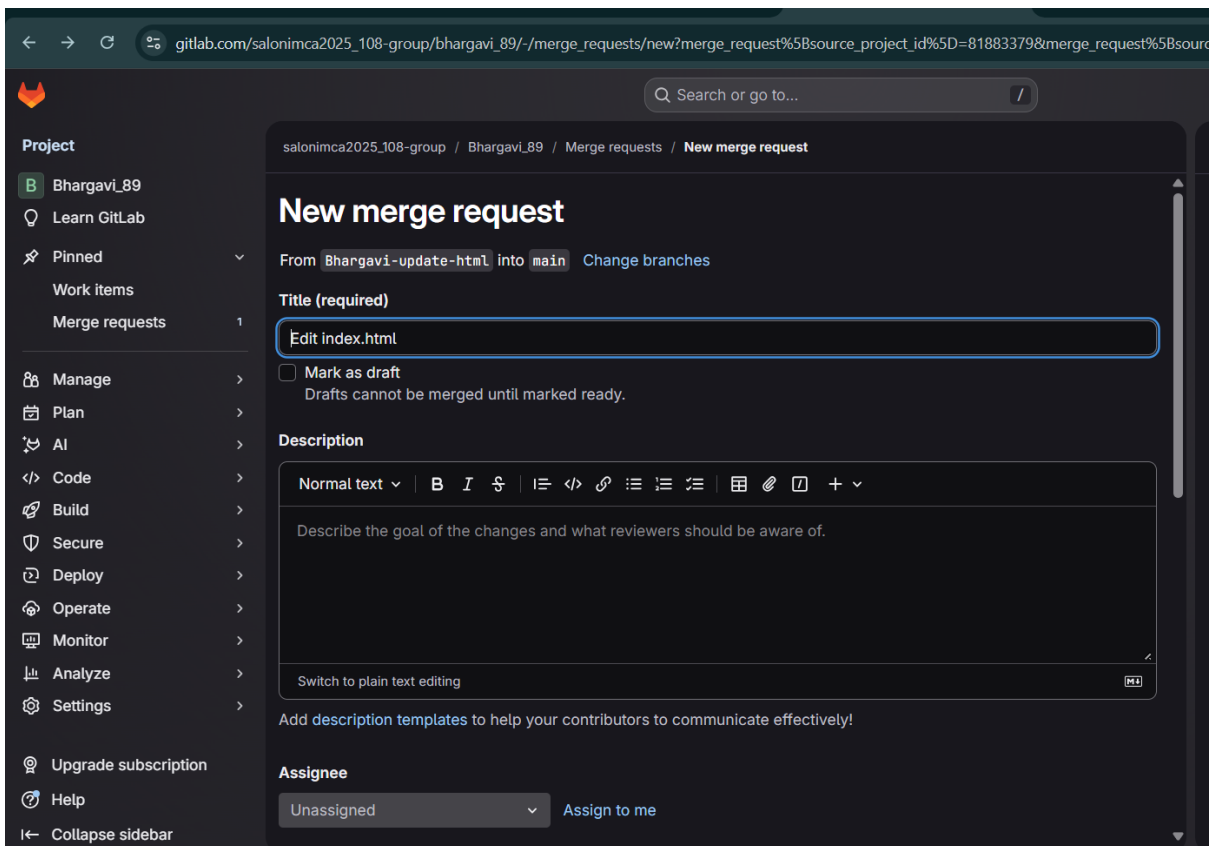
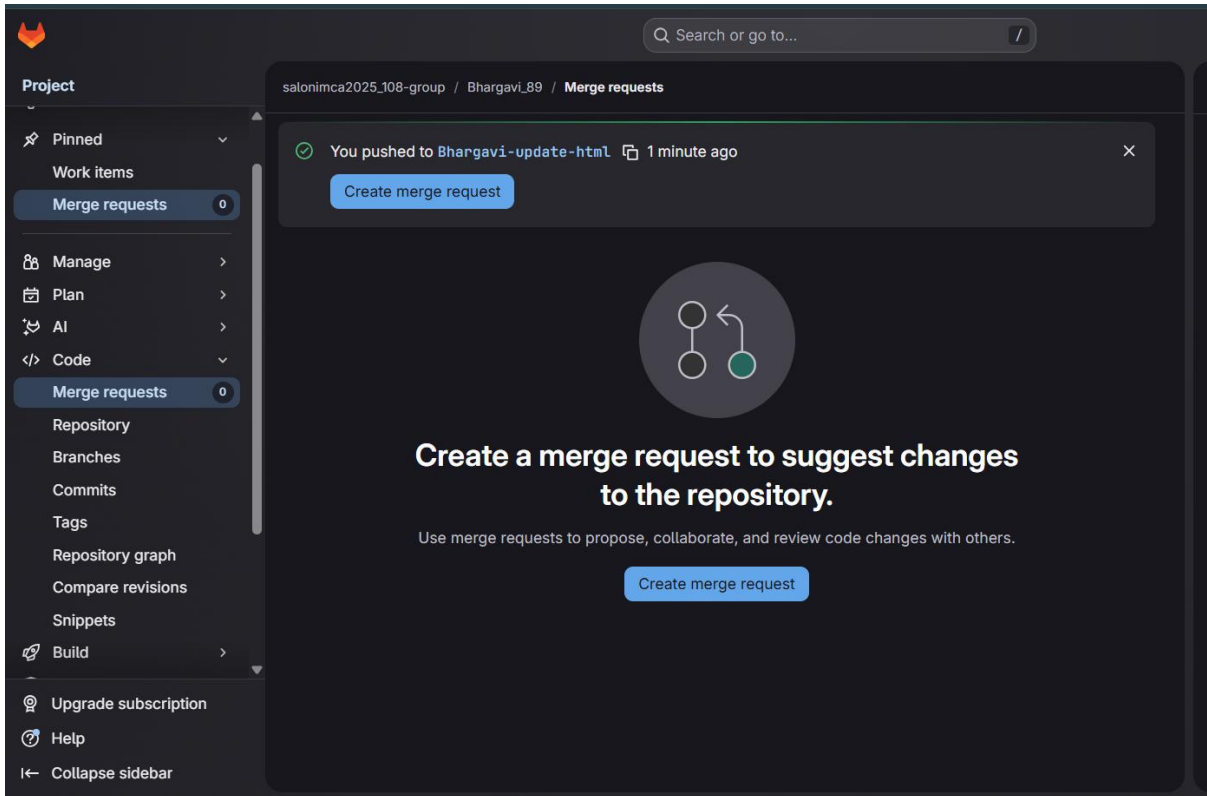


2. Add/edit a file and commit



3. Click on Merge Requests > New Merge Request.





4. Select source and target branches

5. Submit and merge after review

This screenshot shows the 'New merge request' page in GitLab. The breadcrumb navigation at the top reads 'salonimca2025_108-group / Bhargavi_89 / Merge requests / New merge request'. The left sidebar contains a 'Project' menu with options like 'Bhargavi_89', 'Learn GitLab', 'Pinned', 'Work items', 'Merge requests', 'Manage', 'Plan', 'AI', 'Code', 'Build', 'Secure', 'Deploy', 'Operate', 'Monitor', 'Analyze', 'Settings', 'Upgrade subscription', and 'Help'. The main content area is titled 'New merge request' and features two dropdown menus for 'Source branch' (set to 'salonimca2025_108-group/bhargavi_89') and 'Target branch' (set to 'main'). Below these, two commit entries are shown: 'Edit index.html' by Saloni Rohilla (commit hash eb3d0984) and 'index.html' by Saloni Rohilla (commit hash 377571eb). A blue button labeled 'Compare branches and continue' is positioned below the commit list.

This screenshot displays the details of a merged merge request in GitLab. The breadcrumb navigation is 'salonimca2025_108-group / Bhargavi_89 / Merge requests / 12'. The left sidebar is expanded to show 'Merge requests' with a notification badge. The main content area shows a status bar indicating the merge is 'Merged' by 'Bhargavi' from the 'Bhargavi-update-html' branch into the 'main' branch. Below this, there are tabs for 'Overview', 'Commits', 'Pipelines', and 'Changes'. A message from GitLab CI/CD is visible, along with a dropdown menu for 'Approval is optional'. The 'Merged by' section shows 'Saloni Rohilla' merged the request 18 seconds ago, with options to 'Revert' or 'Cherry-pick'. The 'Merge details' section lists 'Changes merged into main with 78099d6a' and 'Deleted the source branch.'. An 'Activity' section shows a recent mention by 'Saloni Rohilla' in commit '78099d6a' and the merge action itself. At the bottom, there is a rich text editor for adding comments.

Practical No – 7

Workflow Management in GitLab

Steps1: In your repo, create .gitlab-ci.yml:

stages:

- build
- test

build-job:

stage: build

script:

- echo "Building..."

test-job:

stage: test

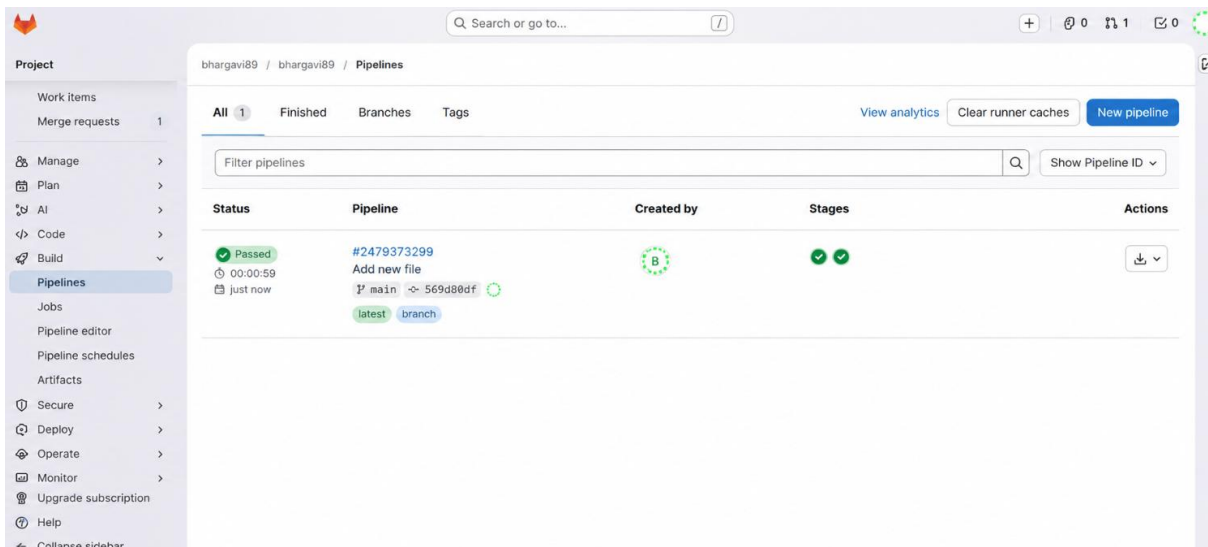
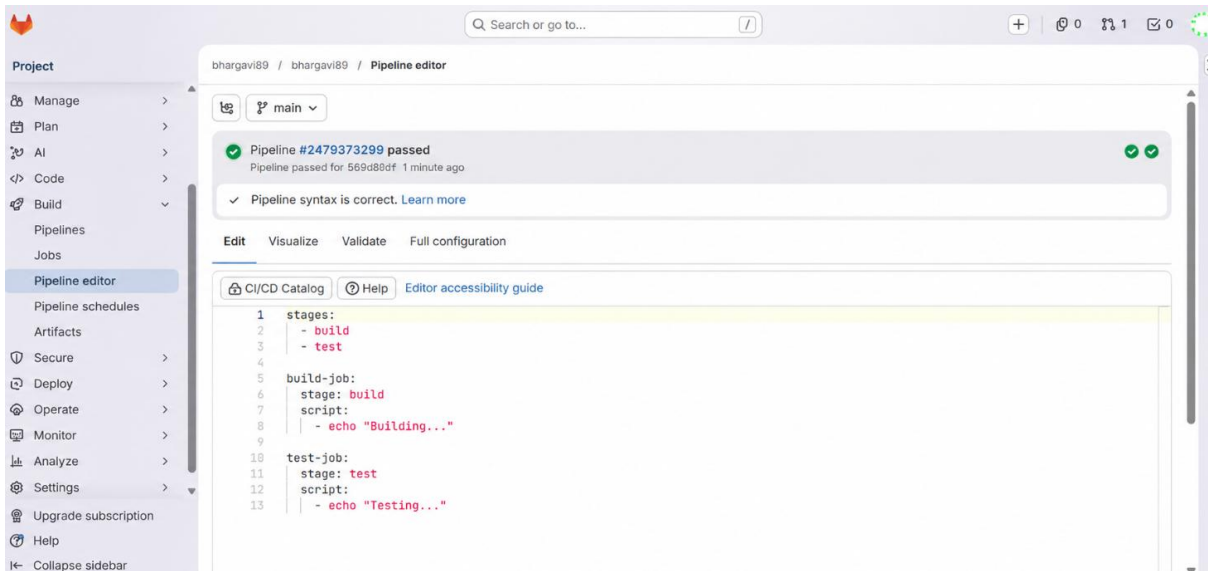
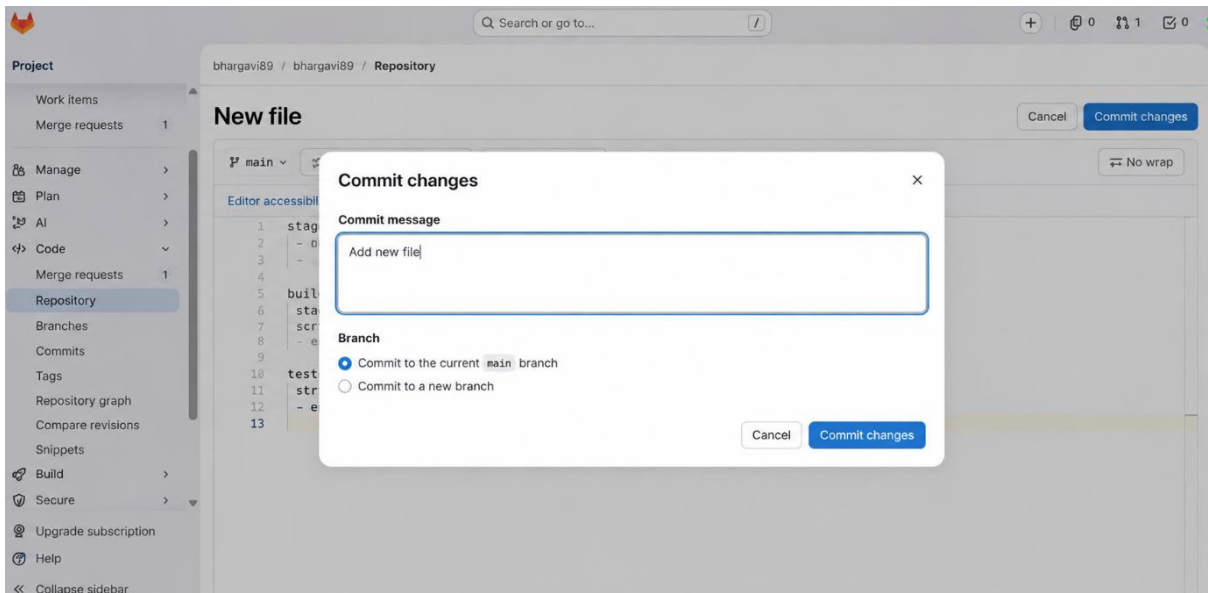
script:

- echo "Testing..."

The screenshot shows the GitLab web interface for a repository named 'bhargavi89 / bhargavi89 / Repository'. A 'New file' editor is open for the file '.gitlab-ci.yml'. The editor contains the following YAML code:

```
1 stages:
2   - build
3   - test
4
5 build-job:
6   stage: build
7   script:
8     - echo "Building..."
9
10 test-job:
11   stage: test
12   script:
13     - echo "Testing..."
```

2. Commit and push.



3. Go to CI/CD > Pipelines and view the build/test stages.

The screenshot displays a web interface for a CI/CD pipeline. On the left is a sidebar with navigation options: Project, Work Items, Merge requests (1), Manage, Plan, AI, Code, Build, Pipelines (selected), Jobs, Pipeline editor, Pipeline schedules, Artifacts, Secure, Deploy, Operate, Monitor, Upgrade subscription, Help, and Collapse sidebar. The main content area shows the pipeline details for commit #2479373299. The pipeline is marked as 'Passed' and was created 3 minutes ago by 'Bhargavi89', finishing 2 minutes ago. It is for commit 569d88df and is on the 'main' branch. The pipeline consists of two stages: 'build' and 'test'. The 'build' stage contains one job, 'build-job', and the 'test' stage contains one job, 'test-job'. Both jobs are shown as completed with green checkmarks. The pipeline summary indicates 2 jobs and 0 tests.

Practical No – 8

Continuous Integration and Development Using Jenkins

Steps:

1. Install Jenkins (visit <https://www.jenkins.io>)

2. Run Jenkins: <http://localhost:8080>

Getting Started

Getting Started

<input checked="" type="checkbox"/> Folders	<input checked="" type="checkbox"/> OWASP Markup Formatter	<input checked="" type="checkbox"/> Build Timeout	<input type="checkbox"/> Credentials Binding	** commons-lang3 v3.x Jenkins API ** Ionicons API Folders OWASP Markup Formatter ** ASM API ** JSON Path API ** Structs ** Pipeline: Step API ** commons-text API ** Token Macro Build Timeout
<input type="checkbox"/> Timestamper	<input type="checkbox"/> Workspace Cleanup	<input type="checkbox"/> Ant	<input type="checkbox"/> Gradle	
<input type="checkbox"/> Pipeline	<input type="checkbox"/> GitHub Branch Source	<input type="checkbox"/> Pipeline: GitHub Groovy Libraries	<input type="checkbox"/> Pipeline Graph View	
<input type="checkbox"/> Git	<input type="checkbox"/> SSH Build Agents	<input type="checkbox"/> Matrix Authorization Strategy	<input type="checkbox"/> LDAP	
<input type="checkbox"/> Email Extension	<input type="checkbox"/> Mailer	<input type="checkbox"/> Dark Theme		

Jenkins 2.555.1

3. Create new Freestyle Project: CD-Bhargavi89

Jenkins / All / New Item

New Item

Enter an item name

Select an item type

- Pipeline**
Build, test, and deploy using pipelines. Supports stages, parallel work, and running on multiple agents.
- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

OK

4. Under Source Code Management, choose Git and enter your repo URL.

5. Add Build Step > Execute Windows batch command:

echo Roll No.: Bhargvi89

echo Building Project...

echo Run tests...

Jenkins / CD-Bhargavi89 / Configure

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute Windows batch command

Command

See the list of available environment variables

```
echo Roll No: Bhargvi89
echo Building Project...
echo Running Tests...
```

Advanced

+ Add build step

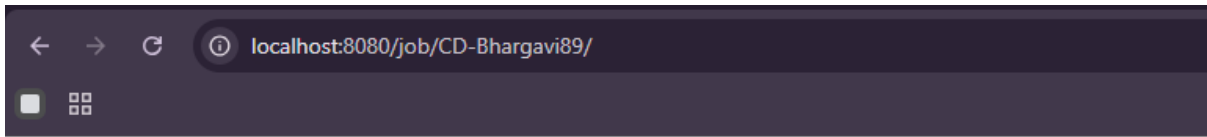
Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

+ Add post-build action

Save Apply

6. Save and click Build Now.



Jenkins / CD-Bhargavi89

Status

CD-Bhargavi89

Changes

Permalinks

Workspace

Build Now

- [Last build \(#1\), 1 min 22 sec ago](#)
- [Last stable build \(#1\), 1 min 22 sec ago](#)
- [Last successful build \(#1\), 1 min 22 sec ago](#)
- [Last completed build \(#1\), 1 min 22 sec ago](#)

Configure

Delete Project

Rename

Builds >

Filter

Today

#1 11:31 PM



Jenkins / CI-prince127

Status

CI-prince127

Changes

Permalinks

Workspace

Build Now

Configure

Delete Project

Rename

Builds >

Filter

Today

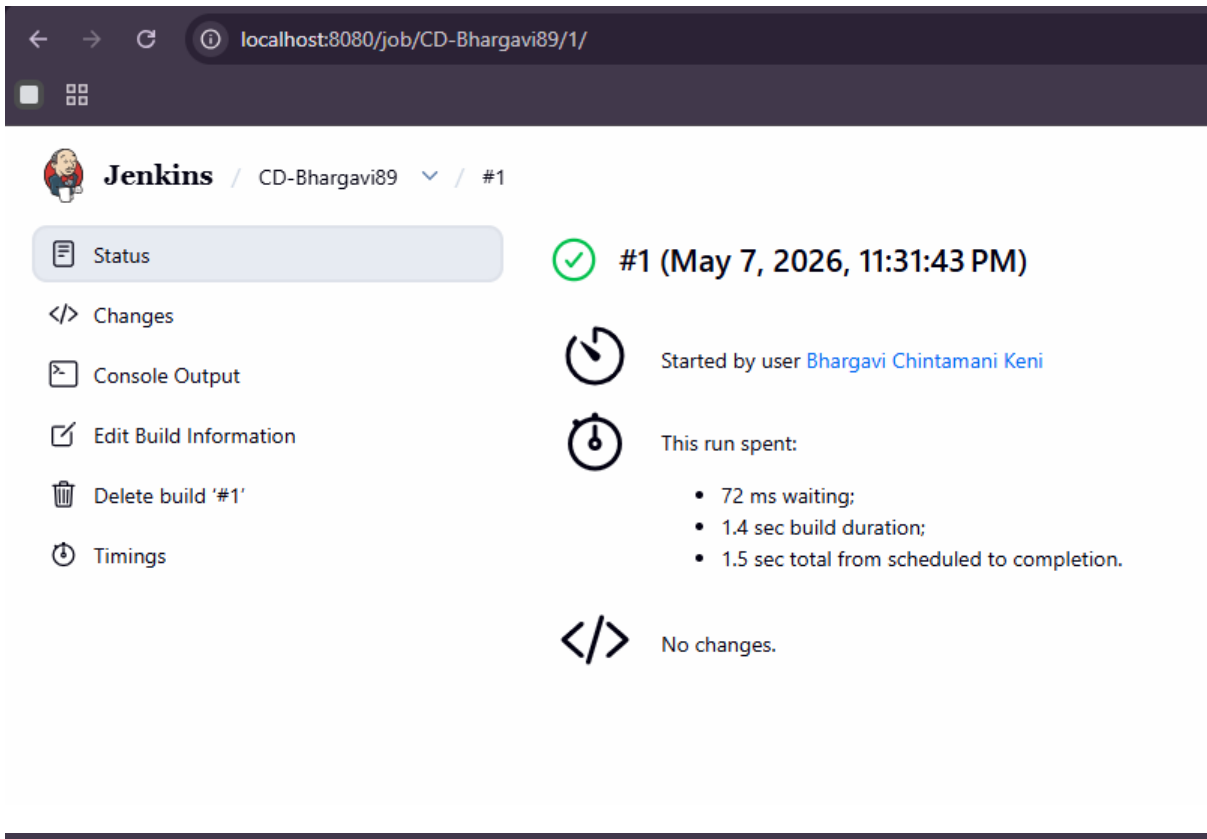
#4 9:23 pm

- [Last build \(#4\), 5 min 47 sec ago](#)
- [Last stable build \(#4\), 5 min 47 sec ago](#)
- [Last successful build \(#4\), 5 min 47 sec ago](#)
- [Last completed build \(#4\), 5 min 48 sec ago](#)

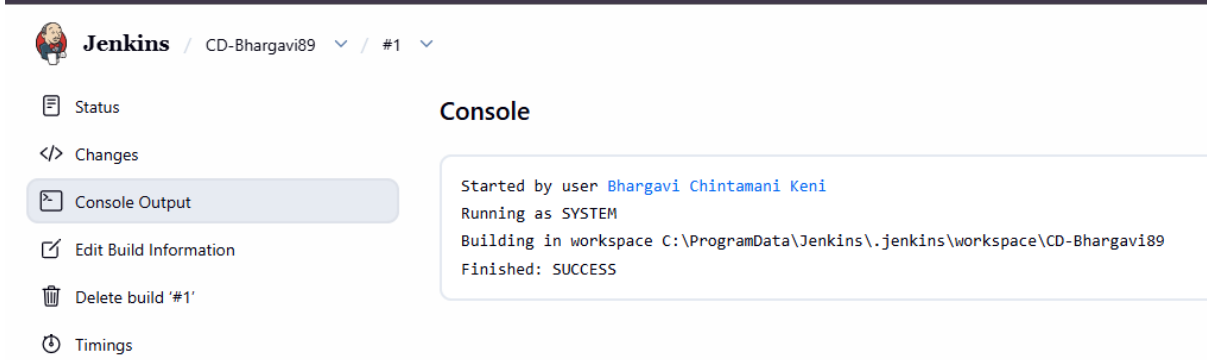
Search, Settings, Help icons

Add description

7. Check output in Console Output.



The screenshot shows the Jenkins web interface for a build. The browser address bar indicates the URL is localhost:8080/job/CD-Bhargavi89/1/. The Jenkins logo and breadcrumb navigation 'Jenkins / CD-Bhargavi89 / #1' are visible. On the left, a sidebar contains menu items: Status (selected), Changes, Console Output, Edit Build Information, Delete build '#1', and Timings. The main content area shows a green checkmark icon followed by '#1 (May 7, 2026, 11:31:43 PM)'. Below this, it states 'Started by user Bhargavi Chintamani Keni'. A clock icon indicates 'This run spent:' followed by a list of metrics: '72 ms waiting;', '1.4 sec build duration;', and '1.5 sec total from scheduled to completion.'. At the bottom, a code icon is followed by 'No changes.'



The screenshot shows the Jenkins web interface for the same build, but with the 'Console Output' menu item selected in the sidebar. The breadcrumb navigation is 'Jenkins / CD-Bhargavi89 / #1'. The 'Console' section is expanded, displaying the following text: 'Started by user Bhargavi Chintamani Keni', 'Running as SYSTEM', 'Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\CD-Bhargavi89', and 'Finished: SUCCESS'.

Practical 9

Explore Docker Commands for Content Management

1. Check Docker Version

```
docker version
```

2. Pull Nginx Image

```
docker pull nginx
```

```
C:\Users\HP>docker --version
Docker version 29.4.2, build 055a478

C:\Users\HP>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
4f55086f7dd0: Pull complete
d5e71e642bf5: Download complete
Digest: sha256:f9078146db2e05e794366b1bfe584a14ea6317f44027d10ef7dad65279026885
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

2. Pull Nginx Image

```
docker pull nginx
```

```
C:\Users\HP>docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
85c66128325a: Pull complete
ff048f1f2159: Pull complete
801a1ad15b4e: Pull complete
4677c2a9a3d4: Pull complete
677c63196868: Pull complete
ce776bbcda0d: Pull complete
3531af2bc2a9: Pull complete
54c38c75806e: Download complete
69989ccd189b: Download complete
Digest: sha256:6e23479198b998e5e25921dfff8455837c7636a67111a04a635cf1bb363d199dc
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview nginx
```

3. List Docker Images

```
docker images
```

4. Run Nginx Container

```
docker run -d -p 8081:80 --name mynginx nginx
```

```
C:\Users\HP>docker run -d -p 8081:80 --name mynginx nginx
13832fcb85425215306b6458d0d6604699054e11eaf6b51d12540ed0e6599efd
```

5. List Running Containers

```
docker ps
```

```
C:\Users\HP>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
13832fcb8542   nginx    "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:8081->80/tcp, [::]:8081->80/tcp mynginx
```

6. Copy File from Host to Container

```
docker cp "C:\Users\HP\Documents\bhargavi89.txt"
mynginx:/usr/share/nginx/html/index.html
```

```
C:\Users\HP>docker cp "C:\Users\HP\Documents\bhargavi89.txt" mynginx:/usr/share/nginx/html/index.html
Successfully copied 139B (transferred 2.05kB) to mynginx:/usr/share/nginx/html/index.html
```

7. Copy File from Container to Host

```
docker cp mynginx:/usr/share/nginx/html/index.html
"C:\Users\HP\Documents\bhargavi89.txt"
```

```
C:\Users\HP>docker cp "C:\Users\HP\Documents\bhargavi89.txt" mynginx:/usr/share/nginx/html/index.html
Successfully copied 139B (transferred 2.05kB) to mynginx:/usr/share/nginx/html/index.html
```

8. Create Docker Volume

```
docker volume create mydata
```

```
C:\Users\HP>docker volume create mydata89
mydata89
```

9. List Docker Volumes

```
docker volume ls
```

```
C:\Users\HP>docker volume ls
DRIVER      VOLUME NAME
local      mydata89
```

Step 10: Remove Container

```
docker rm -f mynginx
```

```
C:\Users\HP>docker rm -f mynginx  
mynginx
```

11. Remove Docker Image

```
docker rmi -f nginx
```

```
C:\Users\HP>docker rmi -f nginx  
Untagged: nginx:latest  
Deleted: sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635cf1bb363d199dc
```

Output

Welcome to nginx!

If you see this page, nginx is successfully installed and working. Further configuration is required for the web server, reverse proxy, API gateway, load balancer, content cache, or other features.

For online documentation and support please refer to nginx.org.
To engage with the community please visit community.nginx.org.
For enterprise grade support, professional services, additional security features and capabilities please refer to f5.com/nginx.

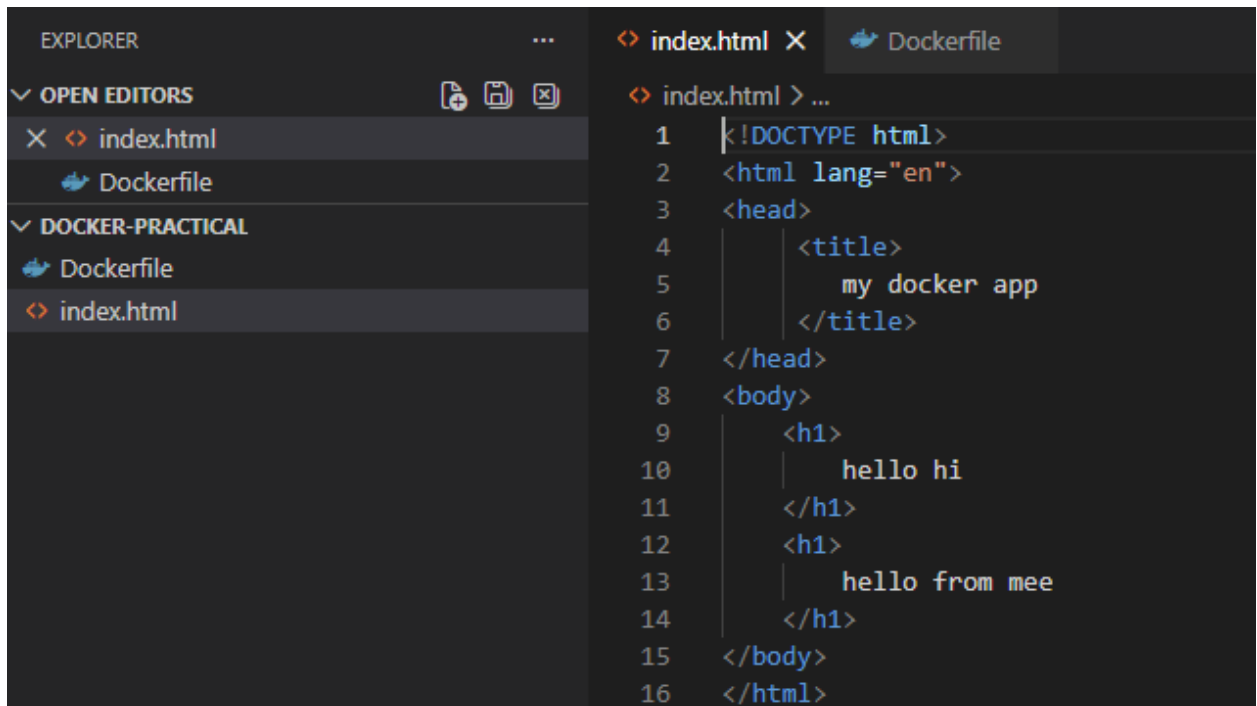
Thank you for using nginx.

← ↻ ⓘ File C:/Users/HP/Documents/bhargavi89.html

Hello Docker

Practical – 10

Aim: Develop a Simple Containerized Application using Docker



The screenshot shows the VS Code editor with the Explorer sidebar on the left and the Editor view on the right. The Explorer sidebar shows the 'DOCKER-PRACTICAL' folder containing 'index.html' and 'Dockerfile'. The Editor view shows the content of 'index.html' with line numbers 1 through 16. The code is as follows:

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>
5          my docker app
6      </title>
7  </head>
8  <body>
9      <h1>
10         hello hi
11      </h1>
12      <h1>
13         hello from mee
14      </h1>
15  </body>
16  </html>

```

Dokerfile



The screenshot shows the VS Code editor with the Explorer sidebar on the left and the Editor view on the right. The Explorer sidebar shows the 'DOCKER-PRACTICAL' folder containing 'index.html' and 'Dockerfile'. The Editor view shows the content of 'Dockerfile' with line numbers 1 through 4. The code is as follows:

```

1  FROM nginx:latest
2
3  FROM nginx:latest
4  COPY index.html |c:\Users\USER\docker-practical\index.html

```

```
cd docker-practical
```

```
docker build -t my-docker-webapp .
```

```
docker run -d -p 8081:80 --name webapp-container my-docker-webapp
```

```
docker ps
```

```
docker stop webapp-container
```

```
docker rm webapp-container
```

```
docker rmi webapp-container
```

```

PS C:\Users\USER\docker-practical> docker build -t my-docker-webapp .
[*] Building 1.3s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 135B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 32B
=> [stage-1 1/2] FROM docker.io/library/nginx:latest@sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635cf1bb363d199dc
=> => resolve docker.io/library/nginx:latest@sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635cf1bb363d199dc
=> CACHED [stage-1 2/2] COPY index.html C:\Users\USER\docker-practical\index.html
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:bd4b7fc2fa6de26ac8d356f063072c55fe1f88b5efdd60924e5788937f8c74b1
=> => exporting config sha256:c08a3a2bffc58a80e9827fbf4e0ed3fc6c64d411e3e747cb6bc7699cd7f5d8e
=> => exporting attestation manifest sha256:d0fc8bb3145f5b0009c7383d92aaf3eff56c16881cf2b160c0cacea80d63718f
=> => exporting manifest list sha256:215755742cc58bd69c0c7a535f8613b8cd28ccbe83573d712ada5e2f65c85e01
=> => naming to docker.io/library/my-docker-webapp:latest
=> => unpacking to docker.io/library/my-docker-webapp:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/pxta4vabxd67cqr5k723khjn
PS C:\Users\USER\docker-practical> docker run -d -p 8081:80 --name webapp-container my-docker-webapp
2340ef4f2119412b0f5d87f22518483eedc96098891555d2c878d7c4e7205803
PS C:\Users\USER\docker-practical> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
2340ef4f2119   my-docker-webapp  "/docker-entrypoint.…"   8 seconds ago  Up 7 seconds  0.0.0.0:8081->80/tcp, [::]:8081->80/tcp  webapp-container
PS C:\Users\USER\docker-practical> docker stop webapp-container
webapp-container
PS C:\Users\USER\docker-practical> docker rm webapp-container
webapp-container
PS C:\Users\USER\docker-practical> docker rmi my-docker-webapp
Untagged: my-docker-webapp:latest
Deleted: sha256:215755742cc58bd69c0c7a535f8613b8cd28ccbe83573d712ada5e2f65c85e01
PS C:\Users\USER\docker-practical>

```

Image Name	Tag	Image ID	Created	Size	Actions
ubuntu	latest	72297040430d	1 year ago	117.35 MB	
docker/welcome-to-docker	latest	eedaff45e3c7	2 years ago	29.46 MB	
ai/mistral	latest	74c6ad24f914	N/A	4.37 GB	
my-docker-webapp	latest	215755742cc5	9 minutes ago	237.17 MB	