

## EXPERIMENT-01

### **AIM: Write a code for simple User Registration form for an event.**

<!-- HTML CODE WITH INLINE CSS FOR REGISTRATION FORM -->

## Objective

To develop a basic user registration form using HTML and JavaScript, allowing users to submit their details for event registration.

---

## Software Requirements

- Code editor (VS Code or any)
  - Web browser (Chrome, Firefox, etc.)
- 

## Procedure

### 1. Create HTML File for User Registration Form

- Create a file named registration.html.
- Add the following code:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>HTML Registration Form</title>

    <script src="script.js" defer></script>

<
s
t
y
l
e
>
```

```
b
o
d
y
{
    font-family:
    Arial, sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content:
    center;      align-
    items: center;

    height: 100vh;
}

.main {
background-
color: #fff;
border-radius:
15px;

    box-shadow: 0 0 20px
    rgba(0, 0, 0, 0.2);
    padding: 20px;      width:
    300px;
}

.main h2 {
    color: #4caf50;
    margin-bottom: 20px;
}

label {
display:
block;
margin-
bottom:
5px;
color: #555;
font-weight:
bold;
}
```

```
input[type="text"],  
input[type="email"],  
input[type="password"],  
select {  
width: 100%;  
margin-  
bottom: 15px;  
padding: 10px;  
box-sizing:  
border-box;  
border: 1px  
solid #ddd;  
  
border-radius: 5px;  
  
} button {  
padding: 15px;  
border-radius:  
10px;  
border: none;  
background-color:  
#4caf50;  
  
color:  
white;  
cursor:  
pointer;  
width:  
100%;  
  
font-size: 16px;  
}  
</style>  
</head>  
<body>  
<div class="main">  
<h2>Registration Form</h2>  
<form action="" method="">
```

```
<label for="firstName">First Name</label>
<input type="text" id="first" name="first" required />

<label for="lastName">Last Name</label>
<input type="text" id="last" name="last" required />
<label for="email">Email</label>
<input type="email" id="email" name="email" required />
<label for="password">Password</label>
<input type="password" id="password"
name="password" pattern="^(?=.*\d)(?=.*[a-zA-Z])(?=.*[^a-zA-Z0-9])\S{8,}$" title="Password"
must contain at least one number, one
alphabet, one symbol, and be at
least 8 characters long" required />
<label for="repassword">Re-type Password</label>
<input type="password" id="repassword" name="repassword" required
/>
<label for="mobile">Contact</label>
<input type="text" id="mobile" name="mobile" maxlength="10" required
/>
<label for="gender">Gender</label>
<select id="gender" name="gender" required>
<option value="male">
    Male
</option>
<option value="female">
    Female
</option>
<option value="other">
    Other
</option>
</select>
```

```

        <button type="button" onclick="submitForm(); return
false;">Register</button>      <p id="confirmation"></p>
    </form>
</div>
</body>
</html>

```

---

## Create JavaScript File for Form Handling

- Create a file named script.js.
- Add the following code:

```

function submitForm() {
    var firstName =
document.getElementById("first").value;  var
lastName =
document.getElementById("last").value;  var
email =
document.getElementById("email").value;
var mobile =
document.getElementById("mobile").value;
var gender =
document.getElementById("gender").value;

    var password =
document.getElementById("password").value;  var
repassword =
document.getElementById("repassword").value;

    if (!firstName || !lastName || !email || !password || !repassword ||
!mobile || !gender) {      alert("Please fill all the fields.");      return;
}

    // Password
match check  if
(password !==
repassword) {

    alert("Passwords do not
match!");      return;
}

```

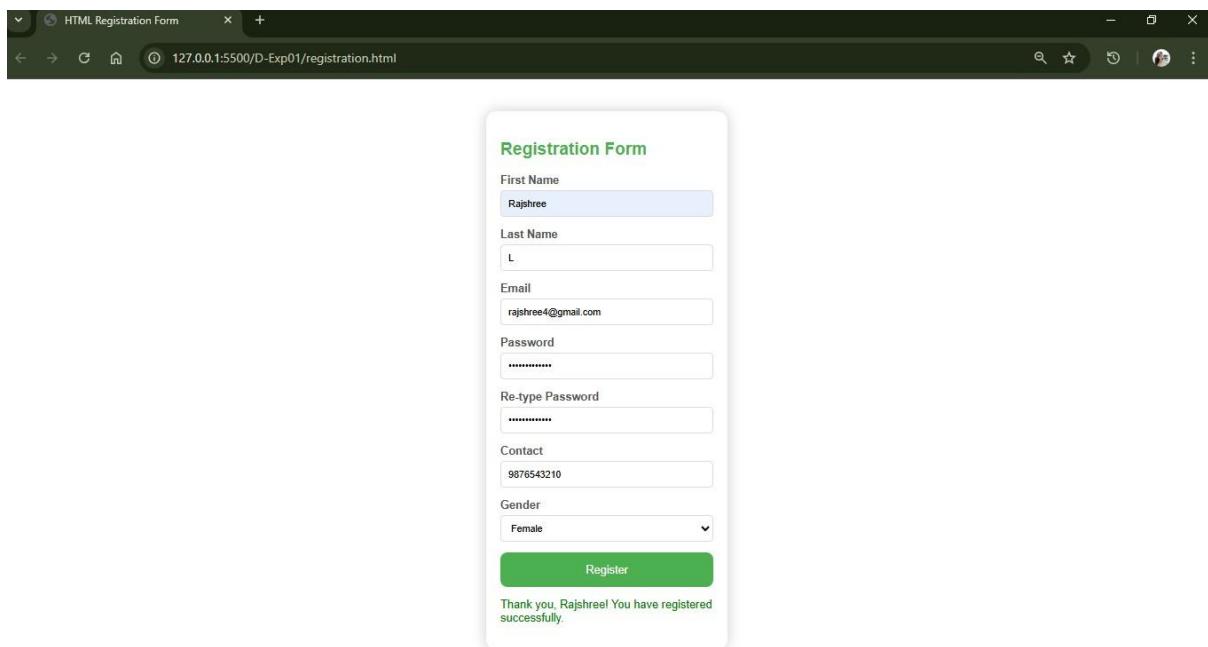
```
document.getElementById('confirmation').innerText =  
    'Thank you, ' + firstName + '! You have registered successfully.';  
  
    console.log('Registration Details:', { Name: firstName, Email: email, Phone:  
mobile  
});  
  
    document.getElementById("confirmation").style.color = "green";  
}
```

---

## Run and Test

- Open the registration.html file in a web browser.
  - Fill in user details and click **Register**.
  - Verify registration confirmation message is displayed.
- 

## Output



The screenshot shows a web browser window titled "HTML Registration Form". The address bar indicates the URL is 127.0.0.1:5500/D-Exp01/registration.html. The main content is a registration form with the following fields and values:

- First Name: Rajshree
- Last Name: L
- Email: rajshree4@gmail.com
- Password: \*\*\*\*\*
- Re-type Password: \*\*\*\*\*
- Contact: 9876543210
- Gender: Female

A large green "Register" button is centered at the bottom of the form. Below the form, a success message is displayed in green text: "Thank you. Rajshree! You have registered successfully."

## EXPERIMENT-02

AIM: Explore Git and GitHub Commands

---

### Objective

To explore and practice basic Git and GitHub commands for version control in a collaborative environment.

---

### Software Requirements

- Git installed (<https://git-scm.com/downloads>)
  - GitHub account (<https://github.com/>)
  - Code editor (VS Code or any)
- 

## VERSION CONTROL SYSTEM

A **Version Control System (VCS)** is a tool that helps manage changes to files, particularly source code, over time. It tracks modifications, allows multiple people to collaborate on the same project, and ensures that a complete history of changes is maintained. VCS is essential for software development, but it can also be used for other types of documents or files.

### GIT

**Git** is a distributed version control system (VCS) designed to manage and track changes in source code during software development. Created by Linus Torvalds in 2005, Git is widely used by developers to collaborate on projects, keep track of changes, and manage multiple versions of their codebase.

### GITHUB

**GitHub** is a web-based platform and service that provides hosting for software development and version control using Git. It offers a collaborative environment where developers can manage and share their projects, track issues, and work together on code. GitHub is widely used in the software development community for both open-source and private projects.

---

## Git Commands: Working With Local Repositories

### 1. git init

- The command `git init` is used to create an empty Git repository.
- After the `git init` command is used, a `.git` folder is created in the directory with some subdirectories. Once the repository is initialized, the process of creating other files begins.

**Command:** `git init`

### 2. git status

- The `git status` command tells the current state of the repository.
- The command provides the current working branch. If the files are in the staging area, but not committed, it will be shown by the `git status`. Also, if there are no changes, it will show the message `no changes to commit, working directory clean`.

**Command:** `git status`

The `git status` command shows us details about:

- **modified files** (files that are changed but not staged).
- **untracked files** (files that Git is not tracking).
- **staged files** (files that are staged and ready to be committed).

### 3. git config

- The `git config` command is used initially to configure the `user.name` and `user.email`. This specifies what email id and username will be used from a local repository.
- When `git config` is used with `--global` flag, it writes the settings to all repositories on the computer.

**Command:** `git config --global user.name "User-name"`  
`git config --global user.email "User-Email"`

### 4. git add

- Add command is used after checking the status of the files, to add those files to the staging area.
- Before running the commit command, "`git add`" is used to add any new or modified files.

**Command:** `git add <file-name>`

`git add . // to add all files`

## 5. git commit

- The commit command makes sure that the changes are saved to the local repository.
- The command "git commit -m <message>" allows you to describe what has happened and help others understand.

**Command:** git commit -m "Message"

## 6. git push

- The command git push is used to transfer the commits or pushing the content from the local repository to the remote repository.
- The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.

**Command:** git push -u origin master

## 7. git branch

- The git branch command is used to determine what branch the local repository is on.
- The command enables adding and deleting a branch.

**Command:** git branch <branch-name> // Create a new

branch  
git branch -a //List all remote or local  
branches  
git branch -d // Delete a branch

## 8. git checkout

- The git checkout command is used to switch branches, whenever the work is to be started on a different branch.
- The command works on three separate entities: files, commits, and branches.

**Command:** git checkout <branch-name> // Checkout an existing branch  
git checkout -b <new-branch> // Checkout and create a new branch with that name.

## 9. git merge

- The git merge command is used to integrate the branches together. The command combines the changes from one branch to another branch.
- It is used to merge the changes in the staging branch to the stable branch.

**Command:** git merge <branch-name>

---

# Git Commands: Working With Remote Repositories

## 1. git remote

- The git remote command is used to create, view, and delete connections to other repositories.
- The connections here are not like direct links into other repositories, but as bookmarks that serve as convenient names to be used as a reference.  
**Command:** `git remote add origin <repo-link>`   `git remote -v` // List all currently configured remote repositories.

## 2. git clone

- The git clone command is used to create a local working copy of an existing remote repository.
- The command downloads the remote repository to the computer. It is equivalent to the Git init command when working with a remote repository. **Command:** `git clone <remote-URL>`

## 3. git pull

- The git pull command is used to fetch and merge changes from the remote repository to the local repository.
- The command "git pull origin master" copies all the files from the master branch of the remote repository to the local repository.  
**Command:** `git pull <branch-name> <remote-URL>`

## 4. git log - View commit history

---

# Procedure

## 1. Configure Git (First Time

**Setup)** `git config --global user.name`  
"Your Name"

```
git config --global user.email "your-email@example.com" git
config -list
```

---

## 2. Initialize Local Repository

```
mkdir devops-lab
```

```
cd devops-lab
```

```
git init
```

---

## 3. Add Files and Commit

```
Changes echo "Hello  
DevOps" > readme.md  
git status  
git add readme.md  
git commit -m "Initial commit: added readme"
```

---

## 4. View Commit History

```
git  
lo  
g  
git  
lo  
g -  
-  
on  
eli  
ne
```

---

## 5. Push to GitHub

1. Create a repository on GitHub.

2. Connect local repo to GitHub: git remote add origin

```
https://github.com/your-username/your-repo-name.git git branch -M main  
git push -u origin main
```

---

## 6. Clone Repository

```
git clone https://github.com/your-username/your-repo-name.git
```

---

7. Check  
Remote  
URL git  
remote -v

---

## 8. Create and Manage Branches

```
git branch  
feature-1  
git  
checkout  
feature-1  
  
# or  
using  
git  
switch  
featur  
e-1
```

---

## 9. Update Files and Merge

```
Branch echo "Adding feature  
work" > readme.md git add .  
git commit -m "Updated readme in feature-1  
branch" git checkout main git merge feature-1
```

---

10. Pull  
Latest  
Changes git  
pull origin  
main

---

## OUTPUT

```
MINGW64:/c/Users/User/Desktop/GitDemo/devops-lab
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo (main)
$ git config --global user.name "student-rajshree"
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo (main)
$ git config --global user.email "landagerajshree294@gmail.com"
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo (main)
$ mkdir devops-lab
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo (main)
$ cd devops-lab/
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git init
Initialized empty Git repository in C:/Users/User/Desktop/GitDemo/devops-lab/.git/
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ echo "DevOps Project Demo" > README.md
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git status
On branch main

No commits yet

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

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

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
```

```
MINGW64:/c/Users/User/Desktop/GitDemo/devops-lab
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git commit -m "Initial Commit"
[main (root-commit) 151dfb1] Initial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git status
On branch main
nothing to commit, working tree clean
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git log
commit 151dfb1bcccf505ee544e7a4de41621f9e6c362f (HEAD -> main)
Author: student-rajshree <landagerajshree294@gmail.com>
Date:   Sun Jul 27 14:54:28 2025 +0530
    Initial Commit
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git remote add origin "https://github.com/student-rajshree/devops-lab.git"
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git remote -v
origin https://github.com/student-rajshree/devops-lab.git (fetch)
origin https://github.com/student-rajshree/devops-lab.git (push)
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 242 bytes | 80.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/student-rajshree/devops-lab.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

The screenshot shows a GitHub repository named 'devops-lab'. The repository is private and has 1 branch and 0 tags. The main file is 'README.md', which contains the text 'DevOps Project Demo'. The repository has 1 commit from 'student-rajshree' made 40 minutes ago. The repository has 0 stars, 0 forks, and 0 watching. It also has 0 releases published.

```
MINGW64:/c/Users/User/Desktop/GitDemo/devops-remote-lab
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo (main)
$ git clone "https://github.com/student-rajshree/devops-remote-lab.git"
Cloning into 'devops-remote-lab'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo (main)
$ cd devops-remote-lab/

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ echo "Create new file" > index.html

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ git add index.html
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   index.html
```

```
MINGW64/c/Users/User/Desktop/GitDemo/devops-remote-lab

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  index.html

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ git commit -m "Created and added new file"
[main cb1d8d2] Created and added new file
 1 file changed, 1 insertion(+)
 create mode 100644 index.html

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 103.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/student-rajshree/devops-remote-lab.git
  8673061..cb1d8d2 main -> main
branch 'main' set up to track 'origin/main'.

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/devops-remote-lab (main)
$ |
```

student-rajshree/devops-remote-lab

student-rajshree / devops-remote-lab

Code Issues Pull requests Actions Projects Security Insights Settings

devops-remote-lab Private

main 1 Branch 0 Tags

student-rajshree Created and added new file cb1d8d2 · 7 minutes ago 2 Commits

README.md Initial commit 12 minutes ago

index.html Created and added new file 7 minutes ago

README

devops-remote-lab

This is a Remote repo to clone in local machine

About

This is a Remote repo to clone in local machine

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

## EXPERIMENT-03

**AIM:** Practice Source code management on GitHub. Experiment with the source code Written in exercise 1.

---

**Objective:** To practice source code management activities such as adding, committing, pushing, pulling, branching, merging, and collaboration using GitHub.

---

### Software Requirements

- Git installed (<https://git-scm.com/downloads>)
- GitHub account (<https://github.com/>)
- Code editor (VS Code or any)

---

### Procedure

**1. Configure Git (First Time Setup)** git config --global user.name "Your Name"  
git config --global user.email "your-email@example.com" git config --list

---

**2. Create or Clone Repository**

- **Create a new repository on GitHub** or
- **Clone an existing repository:** git clone <https://github.com/your-username/your-repo-name.git> cd your-repo-name

---

**3. Initialize Local Repository (If Not Cloned)**

mkdir source-code-management

cd source-code-management

git init

---

**4. Add Source Code Files** echo "print('Hello DevOps') > main.py  
git status  
git add main.py

---

**5. Commit Changes**

git commit -m "Added main.py with Hello DevOps script"

---

**6. Create and Manage Branches**

git branch  
feature-  
branch git  
switch

feature-  
branch

---

**7. Modify Files and Commit** echo "print('Feature work added')" >> main.py  
git add main.py  
git commit -m "Updated main.py in feature branch"

---

**8. Merge Feature Branch into Main Branch** git  
switch main  
git merge feature-branch

---

**9. Connect to GitHub and Push Code** git remote  
add origin https://github.com/your-  
username/your-repo-name.git  
git branch -M main  
git push -u origin main

---

**10. Pull Updates from GitHub** git pull origin main

---

## **11. Collaboration Tasks**

- Collaborators clone repository.
  - Create new branches.
  - Make code changes.
  - Push to GitHub.
  - Raise pull requests (PR). □ Review and merge PRs.
- 

## **OUTPUT**

```
MINGW64/c/Users/User/Desktop/GitDemo/D-Exp01
User@DESKTOP-CSH05HD MINGW64 ~ (main)
$ cd Desktop/
User@DESKTOP-CSH05HD MINGW64 ~/Desktop (main)
$ cd GitDemo/
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo (main)
$ cd D-Exp01/
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git config --global user.name "student-rajshree"
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git config --global user.email "landagerajshree294@gmail.com"
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git init
Reinitialized existing Git repository in C:/Users/User/Desktop/GitDemo/D-Exp01/.
git/
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git remote add origin main https://github.com/student-rajshree/D-Exp01.git
usage: git remote add [<options>] <name> <url>
      -f, --[no-]fetch      fetch the remote branches
      --[no-]tags          import all tags and associated objects when fetching
                           or do not fetch any tag at all (-no-tags)
      -t, --[no-]track <branch>
                           branch(es) to track
      -m, --[no-]master <branch>
                           master branch
      --[no-]mirror[=(push|fetch)]
                           set up remote as a mirror to push to or fetch from

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git branch feature
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git checkout feature
Switched to branch 'feature'
```

```
MINGW64/c/Users/User/Desktop/GitDemo/D-Exp01
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git branch feature
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git checkout feature
Switched to branch 'feature'
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git branch
* feature
  main
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git add .
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git status
On branch feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   registration.html
    modified:   script.js

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git commit -m "Added Reset button"
[feature a15a5d0] Added Reset button
 2 files changed, 4 insertions(+), 4 deletions(-)

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git status
On branch feature
nothing to commit, working tree clean
```

```
MINGW64:/c/Users/User/Desktop/GitDemo/D-Exp01
User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git push -u origin feature
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 440 bytes | 62.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:     https://github.com/student-rajshree/D-Exp01/pull/new/feature
remote:
To https://github.com/student-rajshree/D-Exp01.git
 * [new branch]      feature -> feature
branch 'feature' set up to track 'origin/feature'.

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git diff main
diff --git a/registration.html b/registration.html
index 09198ad0..34cf5e2 100644
--- a/registration.html
+++ b/registration.html
@@ -90,7 +90,7 @@ 
    </select>
    <button type="button" onclick="submitForm(); return false;">Registe
r</button>
    <p id="confirmation"></p>
-    <!-- <button type="reset" onclick="clickForm()">Reset</button> -->
+    <button type="reset" onclick="clickForm()">Reset</button>
</form>
</div>
</body>
diff --git a/script.js b/script.js
index fe35d4d..91d7b4c 100644
--- a/script.js
+++ b/script.js
@@ -21,6 +21,6 @@ function submitForm() {
    console.log('Registration Details:', { Name: firstName, Email: email, P
hone: mobile });
    document.getElementById("confirmation").style.color = "green";
}

```

```
MINGW64:/c/Users/User/Desktop/GitDemo/D-Exp01
--- a/script.js
+++ b/script.js
@@ -21,6 +21,6 @@ function submitForm() {
    console.log('Registration Details:', { Name: firstName, Email: email, P
hone: mobile });
    document.getElementById("confirmation").style.color = "green";
}

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git checkout feature
Already on 'feature'
Your branch is up to date with 'origin/feature'.

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (feature)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git merge feature
Updating 4494e7f..a15a5d0
Fast-forward
  registration.html | 2 ++
  script.js        | 6 +++---
  2 files changed, 4 insertions(+), 4 deletions(-)

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$ git push -u origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/student-rajshree/D-Exp01.git
  4494e7f..a15a5d0  main -> main
branch 'main' set up to track 'origin/main'.

User@DESKTOP-CSH05HD MINGW64 ~/Desktop/GitDemo/D-Exp01 (main)
$
```

This screenshot shows the GitHub repository page for 'student-rajshree/D-Exp01'. The repository name is 'D-Exp01' and it is private. The current branch selected is 'feature'. The repository has 2 branches and 0 tags. There are 3 commits in the 'feature' branch:

- student-rajshree space removed (88086ae - 14 minutes ago)
- de Added registration and script file for register button (36 minutes ago)
- registration.html space removed (14 minutes ago)

The repository also contains a 'script.js' file which was added 27 minutes ago. A 'README' file is present but empty. The repository has 0 stars, 0 forks, and 0 releases. It is part of the 'DevOps Lab Experiment 1' organization.

This screenshot shows the same GitHub repository page for 'student-rajshree/D-Exp01', but with the 'main' branch selected. The repository has 2 branches and 0 tags. There are 3 commits in the 'main' branch:

- student-rajshree space removed (88086ae - 15 minutes ago)
- de Added registration and script file for register button (37 minutes ago)
- registration.html space removed (15 minutes ago)

The repository also contains a 'script.js' file which was added 28 minutes ago. A 'README' file is present but empty. The repository has 0 stars, 0 forks, and 0 releases. It is part of the 'DevOps Lab Experiment 1' organization.

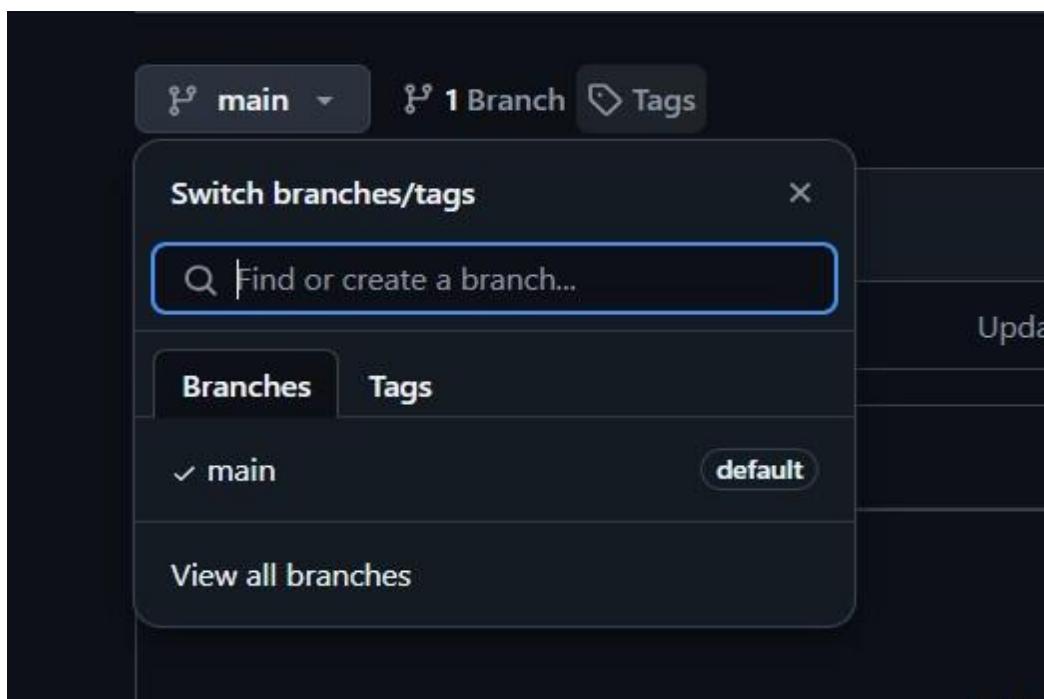
A screenshot of a web browser showing an "HTML Registration Form". The URL in the address bar is 127.0.0.1:5500/registration.html. The form is titled "Registration Form" and contains the following fields:

- First Name: An input field.
- Last Name: An input field.
- Email: An input field.
- Password: An input field.
- Re-type Password: An input field.
- Contact: An input field.
- Gender: A dropdown menu set to "Male".

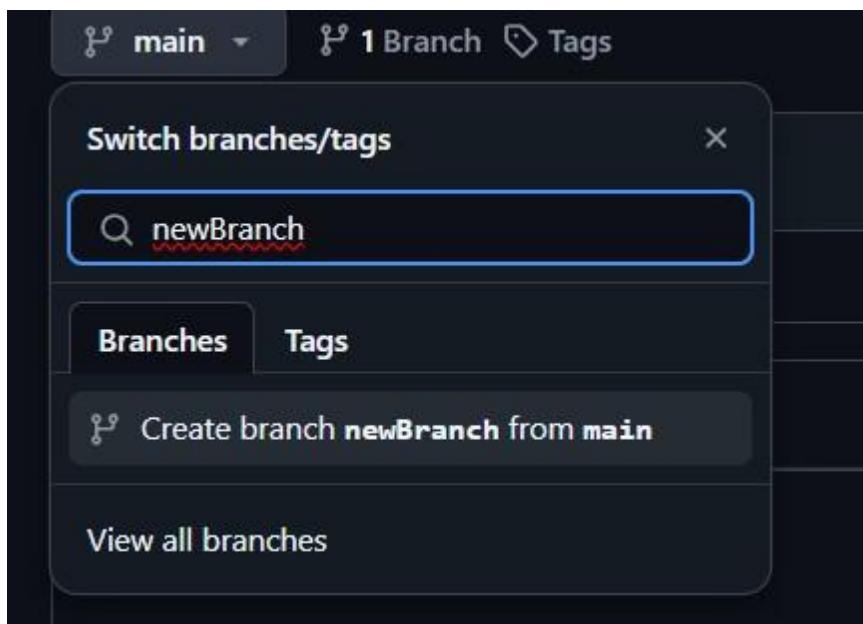
At the bottom are two green buttons: "Register" and "Reset".

## PERFORMING BRANCHING DIRECTLY THROUGH GIT HUB

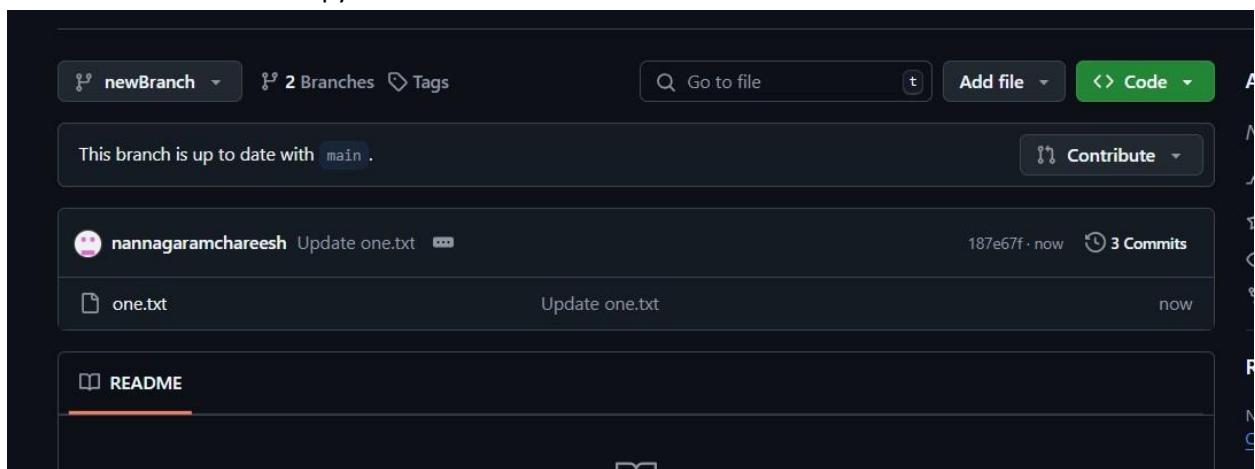
1. First go to your repository where you want to create a new branch
2. Now on the top left you will find the current branch, select it



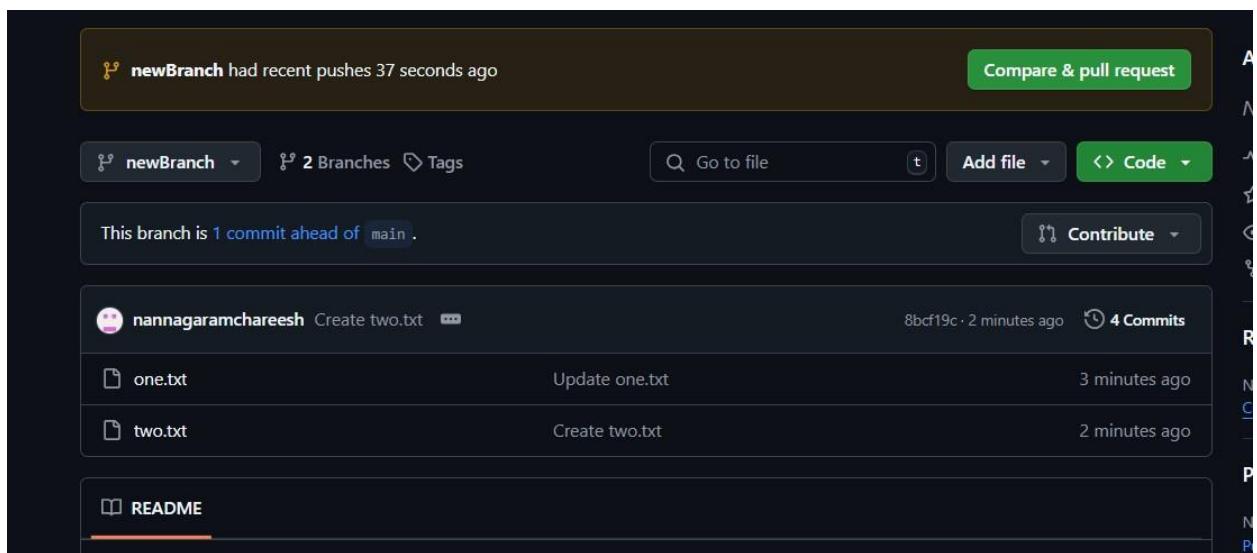
3. Now write the branch name that you want to create in the search bar. If it does not exist then you will see an option saying Create branch newBranch from main.



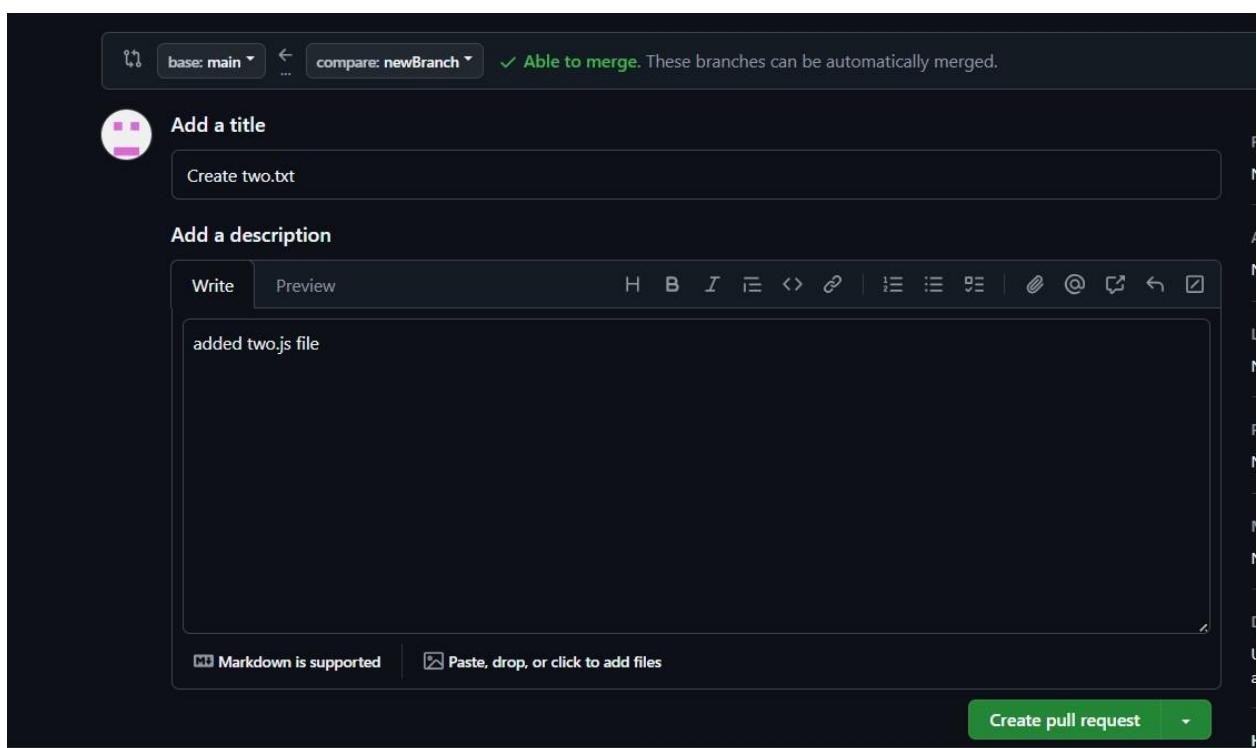
4. After clicking on Create branch, the newBranch will be created and you will automatically get switched to the new branch and all the files in the main branch will be available as a copy in this newBranch



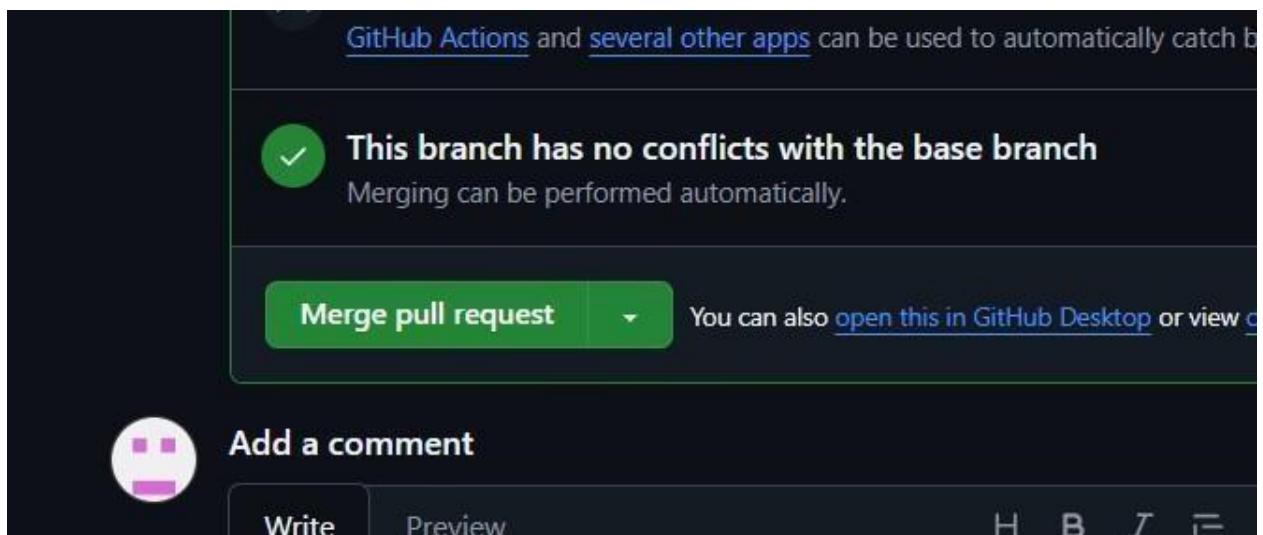
5. Create a new file in this branch write something in it and commit changes
6. Now to merge the changes in this newBranch to the main branch, click on compare & pull request



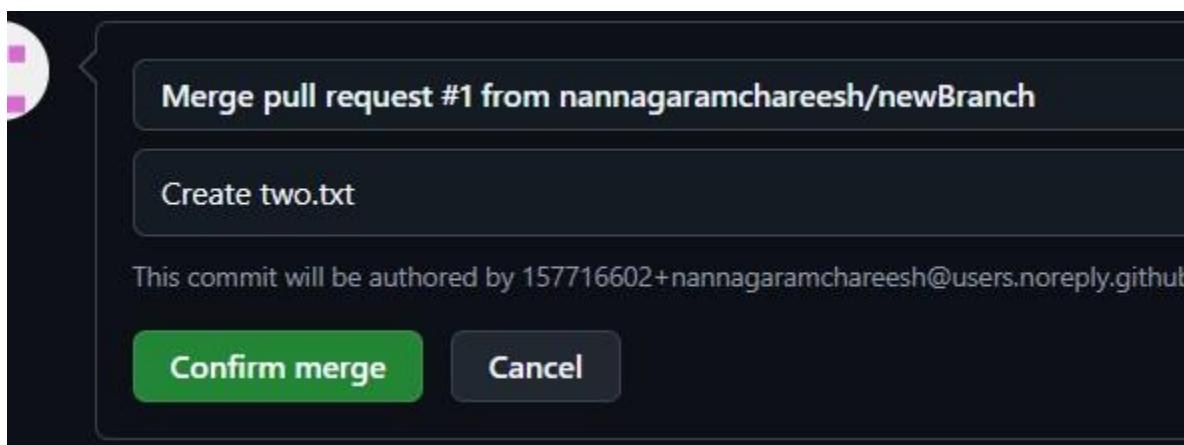
7. Now add a commit message and click on create pull request



8. Then click on Merge pull request in order to reflect the changes made in newBranch to main branch



9. Then click on confirm merge



10. After this if you go to main branch you will see that the changes made in newBranch will be reflected in the main branch So this is how branching can be done in git hub.

## CLONING A REPOSITORY

Inorder to clone a repository use the git clone command.

1. Go ahead and copy the repository of the project that you want to clone
2. Now create a new folder and open it in vs code
3. Now open terminal in vs code and execute the command: git clone "repository\_url"

## FORKING

Forking in GitHub is the process of cloning a repository of someone else's to your own GitHub account where you can modify it freely.

Steps to Fork and then make changes in your local repo-

**1. Go to the repository:** Visit the GitHub page of the repository you want to fork.

**2. Click "Fork":** On the top-right corner of the repository page, you will see a "Fork" button. Click on it.

The screenshot shows a GitHub repository page for 'chaitanya-sahu-123/StudentHub'. At the top right, there are buttons for 'Watch 1', 'Fork 0', and 'Star 0'. Below these, a prominent button says 'Fork your own copy of chaitanya-sahu-123/StudentHub'. The main area displays commit history with four entries:

| Commit Message           | Date         |
|--------------------------|--------------|
| node modules deleted     | last month   |
| This is the first commit | 4 months ago |
| cleared typo             | 4 months ago |
| This is the first commit | 4 months ago |

On the right side, there's an 'About' section with the message 'No description, website, or topics provided.' and various repository statistics: Readme, Activity, 0 stars, 1 watching, 0 forks, and a 'Report repository' link.

## Experiment-04

### AIM: Jenkins Installation and Setup: Explore the Environment

---

**Objective:** To install Jenkins, perform initial setup, and explore the Jenkins environment for automation tasks.

---

#### Software Requirements

- Java installed (JDK 11 or higher)
  - Jenkins package (<https://www.jenkins.io/download/>)
  - Web browser
- 

### Lab Procedure

#### Install Java (If Not

**Installed)** sudo apt

update

sudo apt install openjdk-11-jdk

```
jav  
a –  
ver  
sio  
n  
St  
ep  
s  
to  
ins  
tal  
l  
jd  
k
```

- search for jdk download
- go to the first link of oracle and click jdk 21 and download x64 installer
- give the permissions and install it
- go to command prompt and click java –version to know the version of java

```

Select Command Prompt
Microsoft Windows [Version 10.0.18362.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User>java -version
java version "21.0.8" 2025-07-15 LTS
Java(TM) SE Runtime Environment (build 21.0.8+12-LTS-250)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.8+12-LTS-250, mixed mode, sharing)

C:\Users\User>_

```

## Steps to install and setup Jenkins

1. search for Jenkins download
2. go to the first link ([Download and deploy \(jenkins.io\)](#))
3. now click on java generic package and download it
4. Now to run Jenkins go to command prompt and navigate to the folder where Generic java package.war (Jenkins.war) is downloaded 5. Now run this command  
java -jar jenkins.war

```

Command Prompt - java -jar jenkins.war

E:\Software\Jenkins>java -version
java version "21.0.8" 2025-07-15 LTS
Java(TM) SE Runtime Environment (build 21.0.8+12-LTS-250)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.8+12-LTS-250, mixed mode, sharing)

E:\Software\Jenkins>java -jar jenkins.war
Running from: E:\Software\Jenkins\jenkins.war
webroot: C:\Users\User\jenkins\war
2025-08-08 08:15:08.733+0000 [id=1] INFO  winstone.Logger#logInternal: Beginning extraction from war file
2025-08-08 08:15:12.310+0000 [id=1] WARNING o.e.j.ee9.nested.ContextHandler#setContextPath: Empty contextPath
2025-08-08 08:15:12.450+0000 [id=1] INFO  org.eclipse.jetty.server.Server#doStart: jetty-12.0.22; built: 2025-06-02T15:25:31.946Z; git: 335c9ab44a5591f0ea941bf350
e139b8c4f5537c; jvm 21.0.8+12-LTS-250
2025-08-08 08:15:13.923+0000 [id=1] INFO  o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.ee9.jsp.JettyJspServlet
2025-08-08 08:15:14.057+0000 [id=1] INFO  o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2025-08-08 08:15:15.307+0000 [id=1] INFO  hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\User\.jenkins found at: $user.home/.jenkins
2025-08-08 08:15:15.509+0000 [id=1] INFO  o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.ContextHandler$CoreContextHandler@3fa76c61[Jenkins v2.516.1,,b=file:///C:/Users/User/.jenkins/war/,a=AVAILABLE,h=oeje9n.ContextHandler$CoreToNestedHandler@2d5f7182[STARTED]]
2025-08-08 08:15:15.549+0000 [id=1] INFO  o.e.j.server.AbstractConnector#doStart: Started ServerConnector@ec1b2e4[HTTP/1.1, (http/1.1){0.0.0.0:8080}]
2025-08-08 08:15:15.582+0000 [id=1] INFO  org.eclipse.jetty.server.Server#doStart: Started oejs.Server@75d2da2d[STARTING]{[12.0.22,sto=0]} @8266ms
2025-08-08 08:15:15.586+0000 [id=35] INFO  winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2025-08-08 08:15:15.893+0000 [id=34] INFO  jenkins.model.Jenkins#<init>: Starting version 2.516.1
2025-08-08 08:15:16.176+0000 [id=43] INFO  Jenkins.InitReactorRunner$1#onAttained: Started initialization
2025-08-08 08:15:16.208+0000 [id=46] INFO  Jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2025-08-08 08:15:18.557+0000 [id=45] INFO  Jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2025-08-08 08:15:18.574+0000 [id=44] INFO  Jenkins.InitReactorRunner$1#onAttained: Started all plugins
2025-08-08 08:15:18.590+0000 [id=41] INFO  Jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2025-08-08 08:15:19.079+0000 [id=41] INFO  Jenkins.InitReactorRunner$1#onAttained: System config loaded
2025-08-08 08:15:19.083+0000 [id=45] INFO  Jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-08-08 08:15:19.087+0000 [id=48] INFO  Jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-08-08 08:15:19.094+0000 [id=43] INFO  Jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-08-08 08:15:19.185+0000 [id=62] INFO  hudson.util.Retryer#start: Attempt #1 to do the action check updates server
2025-08-08 08:15:26.073+0000 [id=48] INFO  jenkins.install.SetupWizard#init:
*****
*****Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
e37202ce18de4653843416798e48e719
This may also be found at: C:\Users\User\.jenkins\secrets\initialAdminPassword

```

6. Now a password will be generated, Copy that password

## Access Jenkins Web Interface

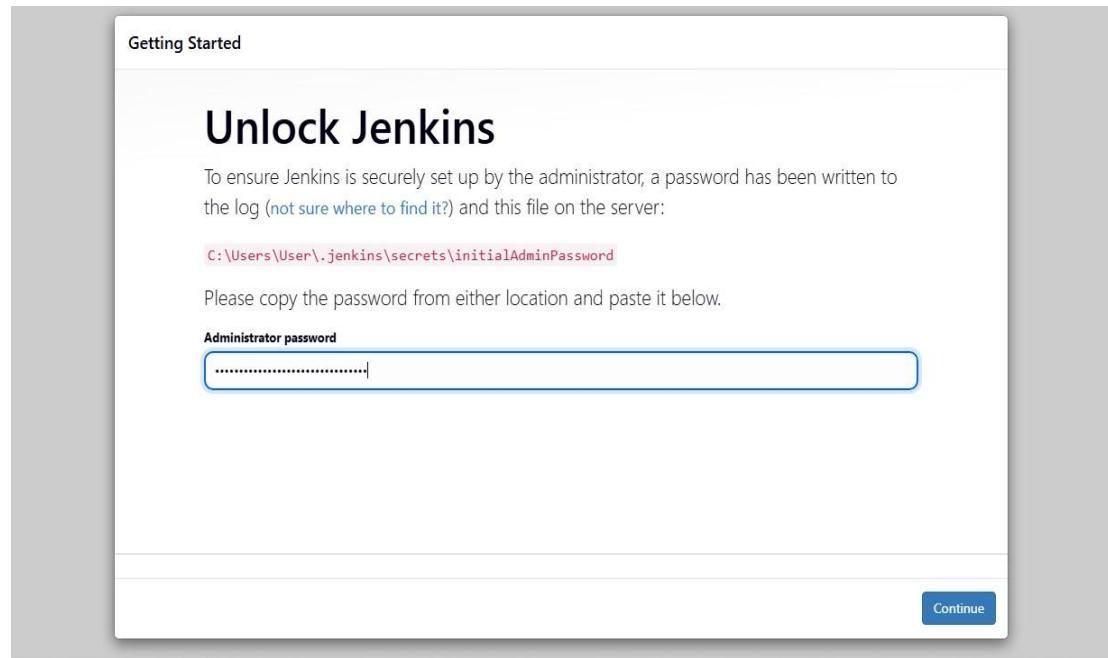
- Open a web browser and navigate to:

<http://localhost:8080>

---

## Unlock Jenkins

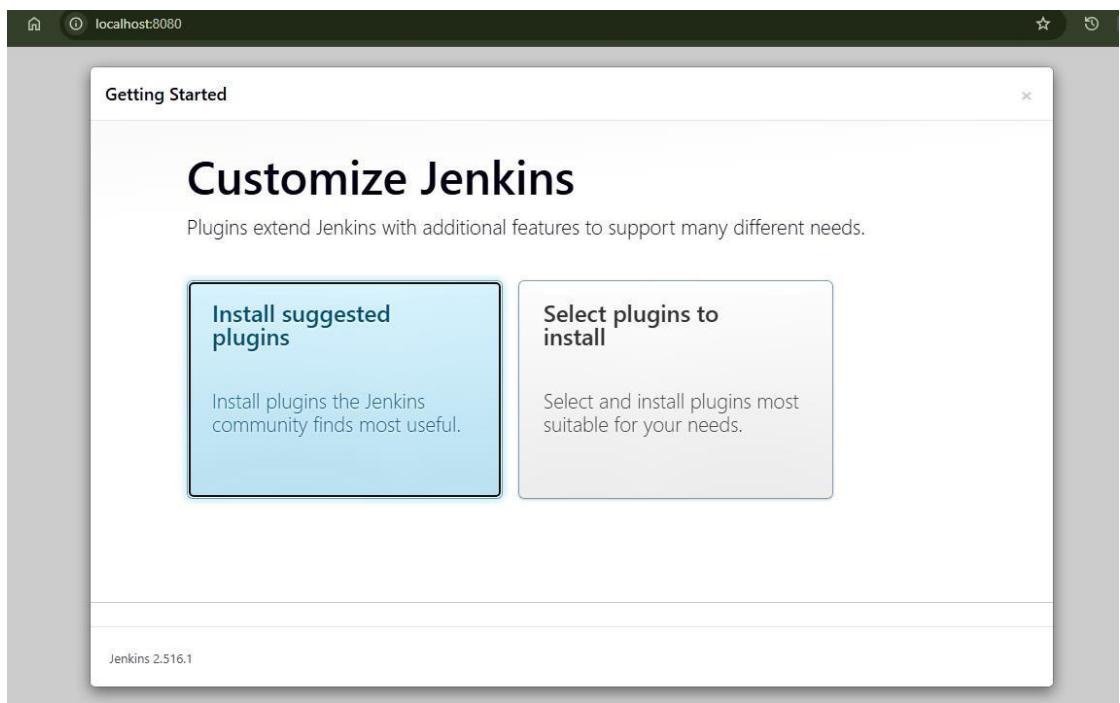
- Retrieve the administrator password:  
`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`
- Paste the password into the Jenkins setup wizard.

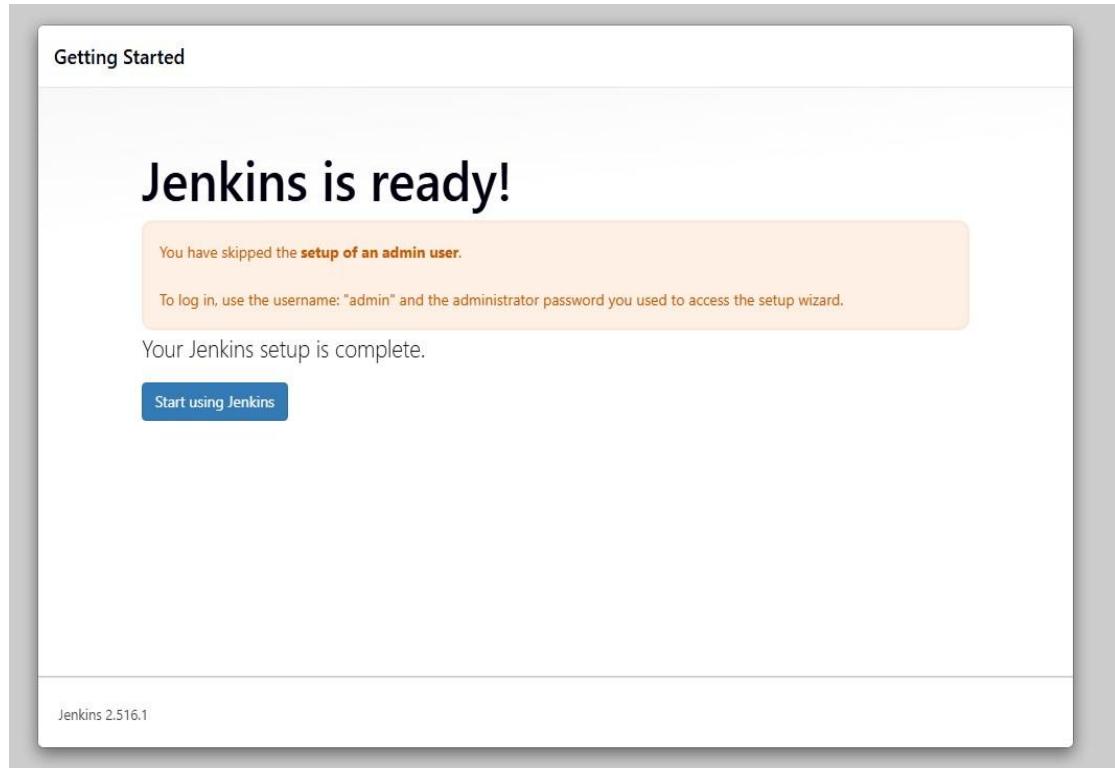
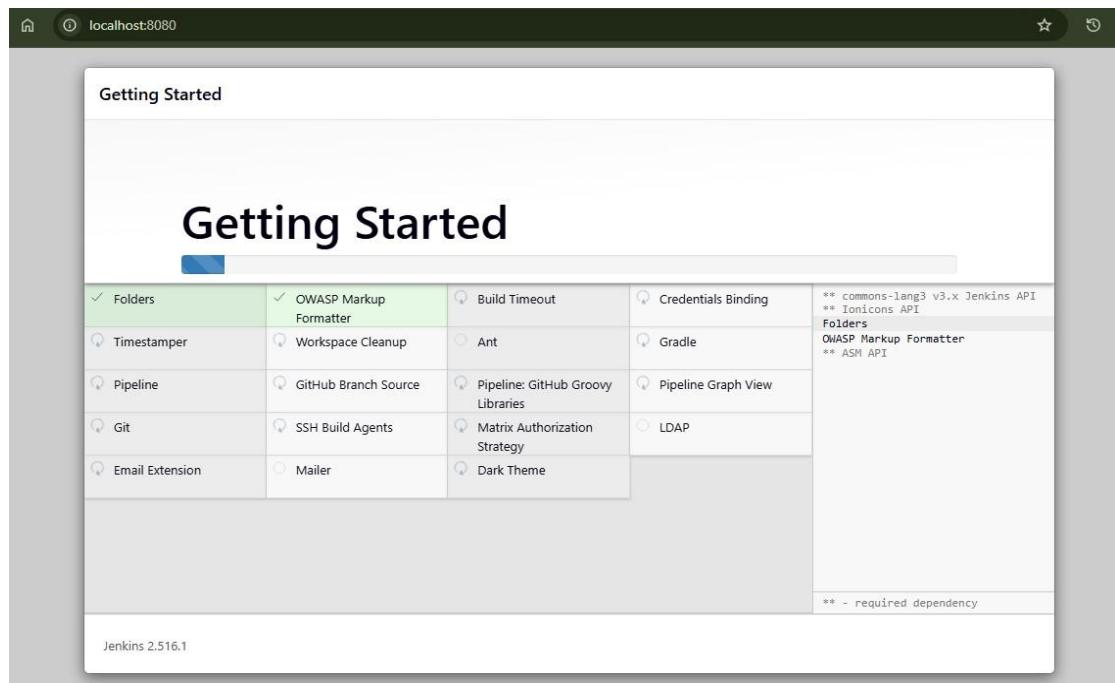


---

## 9. Install Suggested Plugins

- Choose **Install suggested plugins** when prompted.
- Wait for installation to complete.



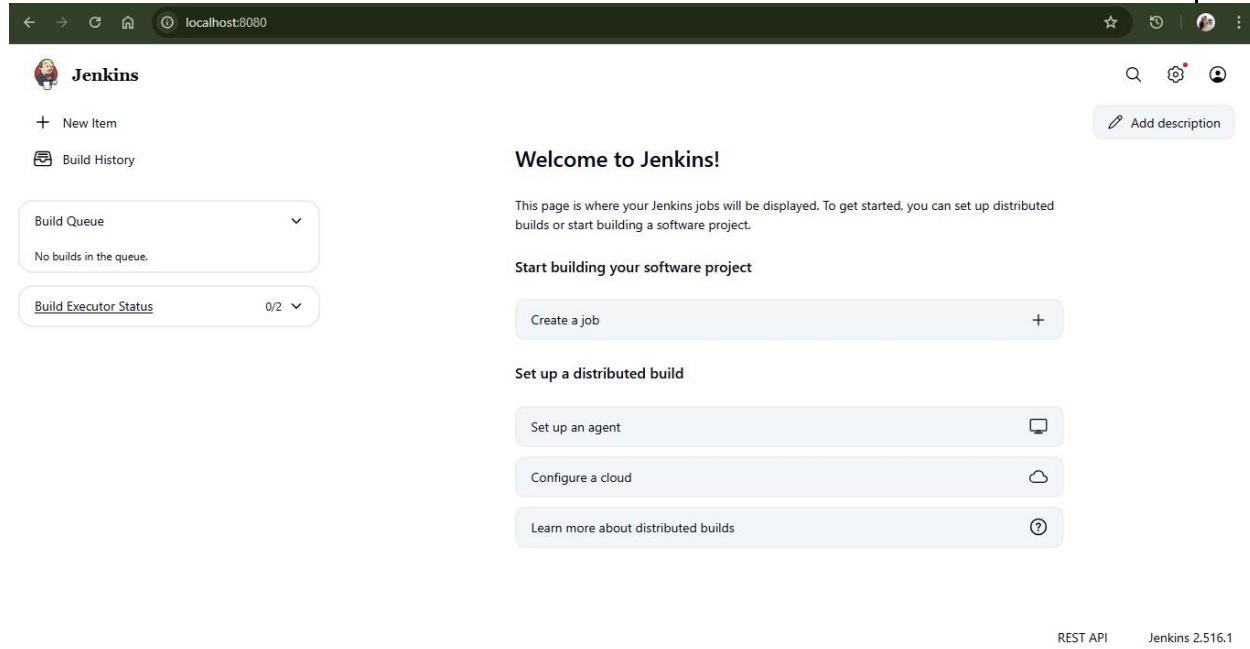


## 10. Create First Admin User

- Enter details like username, password, full name, and email address.

## 11. Explore Jenkins Dashboard □ View Jenkins dashboard interface.

- Explore options like **New Item**, **Manage Jenkins**, **Build History**, and **Credentials**.
- 



## 12 . Verify Installation

- Create a simple freestyle project.
  - Run the project to verify Jenkins build functionality.
- 

## Install Jenkins

### For Ubuntu/Debian:

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add - sudo sh -c  
'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list' sudo  
apt update sudo apt install Jenkins
```

---

### For RedHat/CentOS:

Follow instructions from the official Jenkins website.

Start Jenkins

Service sudo

systemctl start

jenkins sudo

systemctl

```
enable jenkins  
sudo systemctl  
status jenkins
```

---

## Conclusion

- Students should be able to install and configure Jenkins.
  - Explore the Jenkins environment for automating builds and tasks.
  - Understand Jenkins dashboard and management interface.
-

## EXPERIMENT-05

**AIM:** Demonstrate Continuous Integration and Development Using Jenkins

**Objective:** To demonstrate continuous integration and continuous development (CI/CD) by automating build and deployment processes using Jenkins.

---

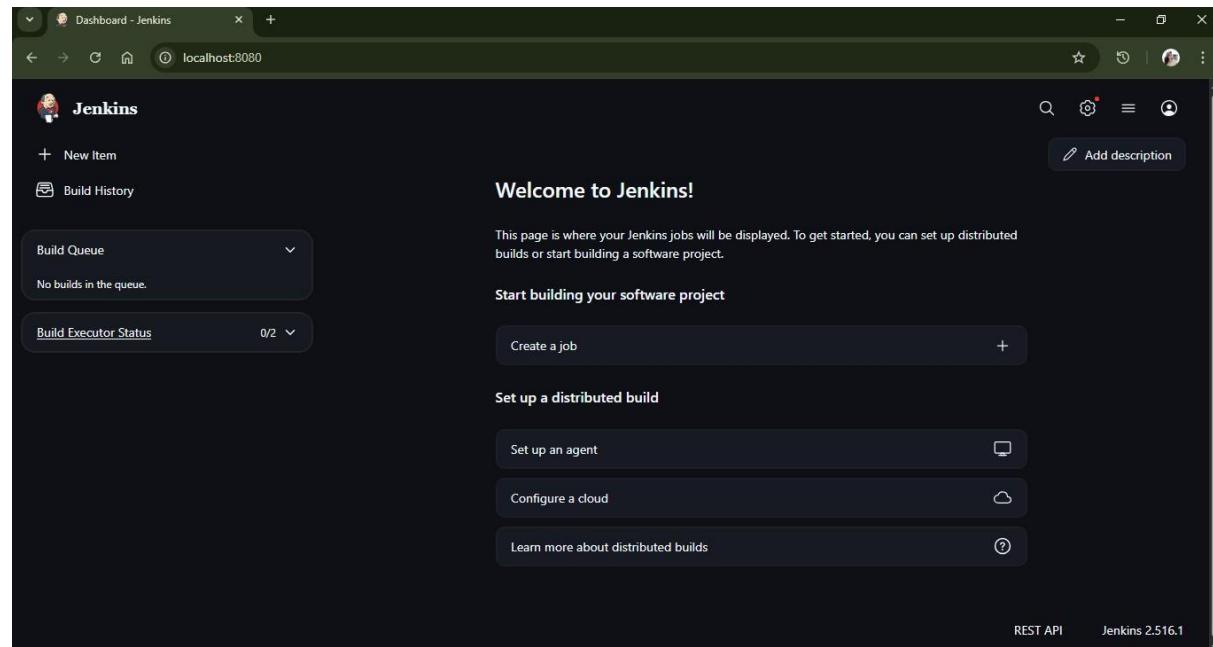
## Software Requirements

- Jenkins installed and configured (<https://www.jenkins.io/download/>)
  - Java installed (JDK 11 or higher)
  - Git installed (<https://git-scm.com/downloads>)
  - A sample project (Java/Python/Node.js etc.)
  - Web browser
- 

## Procedure

### 1. Configure Jenkins for CI/CD

- Ensure Jenkins is installed and accessible via `http://localhost:8080`.
- Log in to Jenkins dashboard.

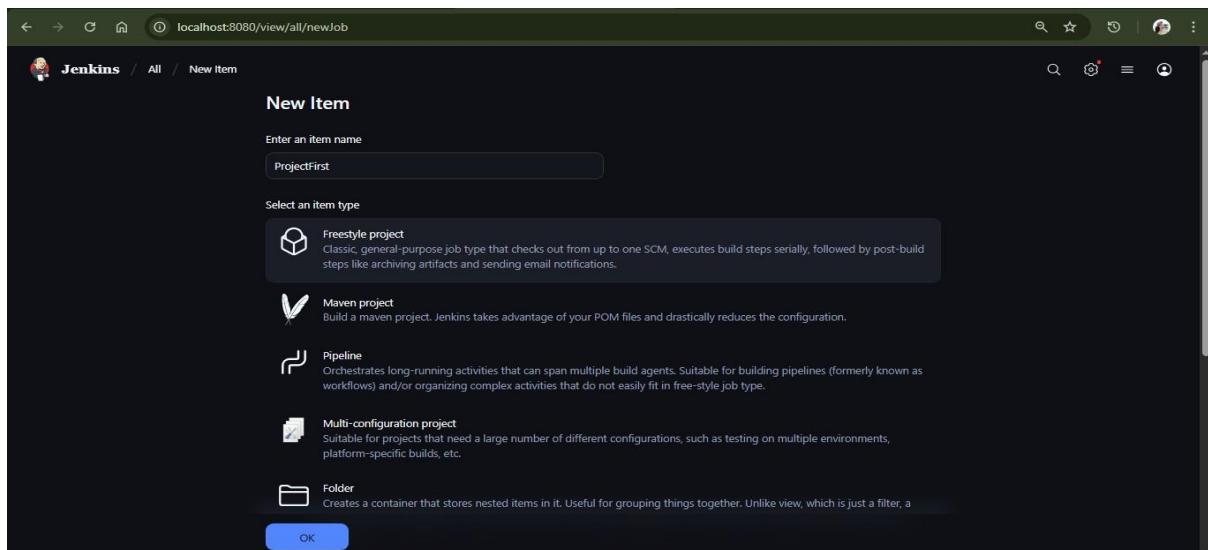


## 2. Install Required Plugins

- Go to **Settings > Manage Jenkins > Plugins.**
  - Install the following plugins if not already installed:
    - Git Plugin
    - Pipeline Plugin
    - Any relevant build tool plugin (Maven, Gradle, etc.)
- 

## 3. Create New Freestyle Project

- Click on **New Item**.
- Enter project name and select **Freestyle project**. □ Click **OK**.



## 4. Configure Source Code Management

- In the project configuration, go to **Source Code Management**.
- Select **Git**.
- Enter repository URL (e.g., <https://github.com/your-username/sampleproject.git>).

The screenshot shows the Jenkins configuration interface for a project named 'FirstProject'. The left sidebar has 'Source Code Management' selected. Under 'Source Code Management', 'Git' is chosen, and the 'Repository URL' is set to 'https://github.com/student-rajshree/integrationDemo.git'. There are options for 'Credentials' (set to '- none -') and 'Advanced' settings. At the bottom are 'Save' and 'Apply' buttons.

## 5. Configure Build Triggers

- Enable **Poll SCM** or **Build periodically** as needed.
- Alternatively, configure **GitHub webhook** for automatic builds.

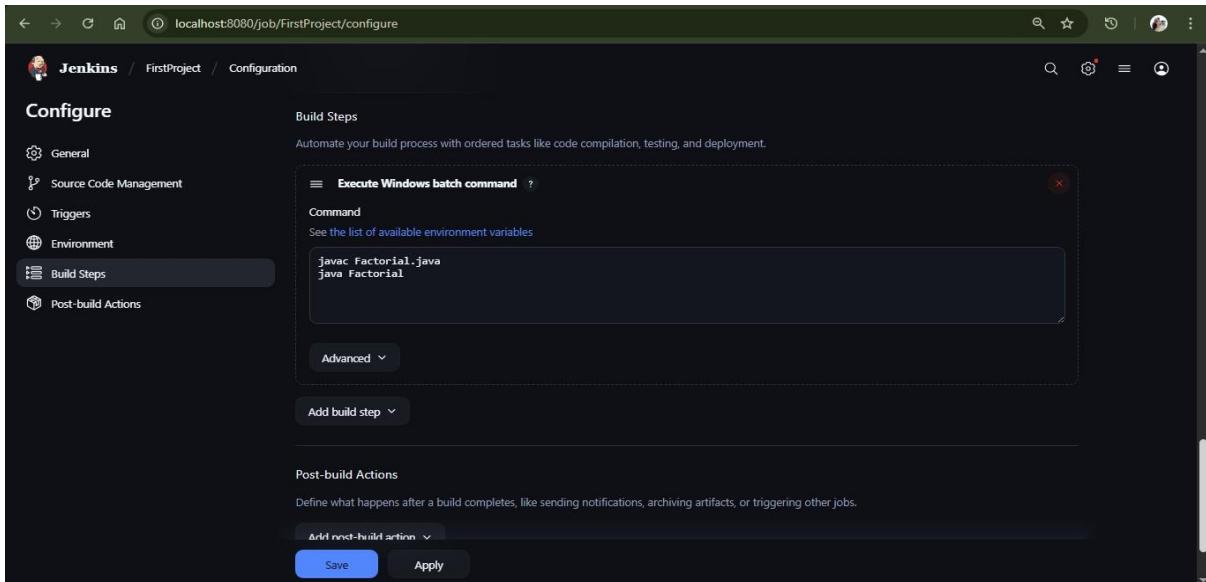
The screenshot shows the Jenkins configuration interface for 'FirstProject'. The left sidebar has 'Triggers' selected. Under 'Triggers', 'Build periodically' is checked. A 'Schedule' field contains '\*\*\*\*\*', which is highlighted with a warning message: '⚠ Do you really mean "every minute" when you say \*\*\*\*\*? Perhaps you meant "H \* \* \* \*" to poll once per hour'. Below the schedule are checkboxes for 'GitHub hook trigger for GITScm polling' and 'Poll SCM'. At the bottom are 'Save' and 'Apply' buttons.

## 6. Define Build Steps

- Go to **Build** section.
- Select appropriate build step (e.g., Execute shell, Invoke Maven targets).
- Example build command:  
mvn clean install

```
javac filename.java
```

```
java filename
```



The screenshot shows the Jenkins configuration interface for a job named 'FirstProject'. The 'Build Steps' section is active, displaying a single step: 'Execute Windows batch command'. The command entered is:  
`javac Factorial.java  
java Factorial`

## 7. Optional: Post-build

**Actions**  Add post-

build steps like:

Archive

artifacts

Email

notifications

ns

## 8. Save and Build

- Save the project.
- Click **Build Now**.
- Monitor the build process in **Build History**.

The screenshot shows the Jenkins interface for the 'FirstProject'. The left sidebar contains links like Status, Changes, Workspace, Build Now, Configure, Delete Project, Favorite, Open Blue Ocean, and Rename. The main area displays the 'Builds' section with a table showing the last five builds. The table includes columns for build number (#1 to #5), start time (e.g., 3:00 PM), and duration (e.g., 29 sec ago). All builds are marked as successful (green checkmark).

| Build | Start Time | Duration   |
|-------|------------|------------|
| #5    | 3:00 PM    | 29 sec ago |
| #4    | 2:59 PM    | 29 sec ago |
| #3    | 2:58 PM    | 29 sec ago |
| #2    | 2:57 PM    | 29 sec ago |
| #1    | 2:56 PM    | 29 sec ago |

## 9. View Console Output

- Click on build number.
- View **Console Output** to check build logs.

The screenshot shows the Jenkins 'Console Output' page for build #5. The left sidebar includes links for Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#5', Timings, Git Build Data, Lockable resources, Open Blue Ocean, and Previous Build. The main content area displays the command-line output of the build process. It shows the git clone, fetch, and checkout steps, followed by the compilation and execution of a Java program named 'Factorial.java'.

```
Started by timer
Running as SYSTEM
Building in workspace C:\Users\User\.jenkins\workspace\FirstProject
The recommended git tool is: NONE
No credentials specified
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\Users\User\.jenkins\workspace\FirstProject\.git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/student-rajshree/integrationDemo.git # timeout=10
Fetching upstream changes from https://github.com/student-rajshree/integrationDemo.git
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.50.1.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/student-rajshree/integrationDemo.git +refs/heads/*:refs/remotes/origin/*
# timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 405c88982da72c70b95c7349dd42b2323f764c7 (refs/remotes/origin/main)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f 405c88982da72c70b95c7349dd42b2323f764c7 # timeout=10
Commit message: "Delete Integration.java"
> C:\Program Files\Git\bin\git.exe rev-list --no-walk 405c88982da72c70b95c7349dd42b2323f764c7 # timeout=10
[FIRSTPROJECT] $ cmd /c call C:\Users\User\AppData\Local\Temp\jenkins7849028522153200510.bat

C:\Users\User\.jenkins\workspace\FirstProject>java Factorial.java

C:\Users\User\.jenkins\workspace\FirstProject>java Factorial
Factorial of 10 = 3628800
C:\Users\User\.jenkins\workspace\FirstProject>exit 0
Finished: SUCCESS
```

## 10. Automate Deployment (Optional)

- Add shell scripts or deployment tasks as build steps to automate deployment after build.

## EXPERIMENT-06

**AIM:** Explore Docker commands for content management.

**Objective:** To explore and understand Docker commands for managing images, containers, files, and volumes (content management).

---

## Software Requirements

- **Docker installed and configured** (Docker Desktop for Windows/macOS or Docker Engine for Linux) – <https://www.docker.com/get-started>
  - **Operating System:** Linux (Ubuntu 20.04 or later) / Windows 10 or later with WSL2 / macOS (latest)
  - **Command-line terminal** (Bash / PowerShell) for executing Docker commands
  - **Internet connectivity** to pull images from Docker Hub or other registries
  - **Optional text editor** (VS Code) to create and edit files inside containers
- 

**Docker** is an open-source platform that allows developers to easily build, deploy, and run applications in containers.

Docker allows developers to create a container image that includes all the dependencies needed for an application to run, regardless of the underlying operating system or infrastructure.

Docker also provides tools for managing container images, such as Docker Hub, a public registry of images, and Docker Compose, a tool for defining and running multi-container applications.

## Docker Commands

1. **docker:** This command is used to list all docker commands.
2. **docker –version:** This command is used to check docker version.
3. **docker pull ImageName:** This command is used to download a Docker image from a registry or from docker hub.
4. **docker run ImageName:** This command is used to create a new Docker container from an image.
  - i. **docker run –d ImageName:** This command is used to build a container in background.
  - ii. **docker run ImageName: Version:** This command is used to pull and create a specific version of an image.

- iii. **docker run -it ImageName:** This command is used to access terminal of that container in our local machine in the interactive mode.
  - iv. **docker run -name containerName -d ImageName:** This command is used to give custom name to a container.
- 5. **docker images:** This command is used to list all the Docker images that are currently available on your system.
- 6. **docker rm ContainerName:** This command is used to remove a Docker container.
- 7. **docker rmi ImageName:** This command is used to remove an docker image.
- 8. **docker start ContainerName or ContainerID:** This command is used to restart an existing container.
- 9. **docker stop ContainerName or ContainerID:** This command is used to stop a running container.
- 10. **docker ps:** This command is used to list all the running Docker containers.
- 11. **docker ps -a:** This command is used to list all the Docker containers, including running and stopped containers.
- 12. **docker logs ContainerID or ContainerName:** This command is used to check the logs of the container which can help to identify the root cause of the problem.
- 13. **docker exec -it ContainerID or ContainerName: bash/terminal/sh** This command is used to execute additional commands in a running container. We can access that containers bash/terminal.
- 14. **docker compose -f FileName.yaml up -d:** This command is used to manage multi-container Docker applications.
  - f: Option for file name **up -d:** Whenever the containers are defined inside the yaml file and if we want to create and start them in the detached mode we will choose up.
- 15. **docker compose -f FileName.yaml down:** Whenever the containers are defined inside the yaml file and if we want to delete them then we will choose down.
- 16. **docker build -t ImageName: version . :** This command is used to build docker image.

**-t or—tag:** Version of images **dot**

**(.):** Build context (Current  
Directory)

**17.docker login –u UserName:** The Docker login command will help you to authenticate with the Docker hub by which you can push and pull your images.

**18.docker push Image Name:** This command is used to push docker image to dockerhub.

**19.docker volume create VolumeName:** This command is used to create a volume.

**20.docker volume ls:** This command is used to list all volumes available.

**21.Docker run –it –v VolumeName: ContainerPath:** This command is used to attach the volume to a path inside the container.

**22.docker volume rm VolumeName:** This command is used to remove the volume.

**23.docker volume prune:** This command is used to remove unused volumes.

## Some important flags

**-d:** To run any container in background.

**-e:** To set environment variables

**-p:** To bind the host port with the container port

## EXPERIMENT-07

**AIM:** Develop a simple containerized application using Docker.

**Objective:** To understand how to create, containerize, and run a simple application using Docker.

---

### Software Requirements

- **Docker installed and configured** (Docker Desktop for Windows/macOS or Docker Engine for Linux) – <https://www.docker.com/get-started>
  - **Operating System:** Linux (Ubuntu 20.04 or later) / Windows 10 or later with WSL2 / macOS (latest)
  - **Command-line terminal** (Bash / PowerShell) for executing Docker commands
  - **Git** for source control
  - **Internet connectivity** to pull images from Docker Hub or other registries
  - **Optional text editor** (VS Code) to create and edit files inside containers
- 

### Lab Procedure

1. Clone the getting-started-app repository using following command:

```
git clone https://github.com/docker/getting-started-app.git
```

2. View the contents of the cloned repository. You should see the following files and sub-directories.

```
├── getting-started-app/
|   ├── .dockerignore
|   ├── package.json
|   ├── README.md
|   ├── spec/
|   └── src/
        └── yarn.lock
```

3. Create a Docker file in the same directory and add the following code

```

FROM node:18-alpine

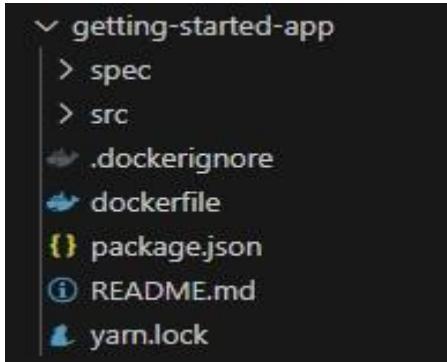
WORKDIR
/app COPY ..

RUN yarn install --production

CMD ["node", "src/index.js"]

```

**EXPOSE 3000**



In the terminal, make sure you're in the getting-started-app directory. Replace /path/to/getting-started-app with the path to your getting-started-app directory.

*cd /path/to/getting-started-app*

#### 4. Build the image

Run the following command in the terminal

**docker build -t getting-started-app .**

```

E:\CMRCET20240105\III Yr\Exp-07\getting-started-app>docker build -t getting-started-app .
[+] Building 88.3s (10/10) FINISHED
--> [internal] load build definition from dockerfile
--> => transferring dockerfile: 154B
--> [internal] load metadata for docker.io/library/node:18-alpine
--> [auth] library/node:pull token for registry-1.docker.io
--> [internal] load .dockerignore
--> => transferring context: 66B
--> [1/4] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4a28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
--> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4a28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
--> => sha256:25ff2da83641908f65c3a74d0409d6b1b62ccfaab220b9ea70b80df5a2e0549 446B 0.2s
--> => sha256:1e5a4c89ceec5c0826c540ab0ed4b6b491c96eda01837f430bd47f0d26702d6e1 1.26MB 0.1s
--> => sha256:dd71dde834b5c203d162902eebb8994cb2309ae049aeabc4fea161b2b5a3d0e 40.01MB 5.3s
--> => sha256:f1b232174bc91741fdf3da96d885011092101a032a93a388b79e99e69c2d5c870 3.64MB 0.05
--> => extracting sha256:f1b232174bc91741fdf3da96d885011092101a032a93a388b79e99e69c2d5c870 2.2s
--> => extracting sha256:dd71dde834b5c203d162902eebb8994cb2309ae049aeabc4fea161b2b5a3d0e 5.4s
--> => extracting sha256:1e5a4c89ceec5c0826c540ab0ed4b6b491c96eda01837f430bd47f0d26702d6e3 0.2s
--> => extracting sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549 0.1s
--> [internal] load build context: 6.42MB 0.05
--> [2/4] WORKDIR /app 36.0s
--> [3/4] COPY . . 0.35
--> [4/4] RUN yarn install --production 29.8s
--> exporting to image 12.1s
--> exporting layers 0.1s
--> => exporting manifest sha256:b63b3f3a68e65cb64253f6a6b9a0b625a6f3c81ea6df4a69c94ed018eba2b41e 0.05
--> => exporting config sha256:d3768a2e73203287cd1b5438f8d0492fd91bcb5a8b70e34a6fde06ee80415a7 0.05
--> => exporting attestation manifest sha256:4e1ee567225f4e4db1e841e0295d913f8fb8066c8d9bf8b030dc5e8258f5fc2 0.1s
--> => exporting manifest list sha256:3d2d0fbba0956550b1796f52eb5d94f0d83f01957a0a073e61841ed99a0e0cc 0.1s
--> => naming to docker.io/library/getting-started-app:latest 0.05
--> => unpacking to docker.io/library/getting-started-app:latest 17.3s
WARNING: current commit information was not captured by the build: git was not found in the system: exec: "git.exe": executable file not found in %PATH%
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/fzvxoji8j0872ifoxhrq7rhcf
  
```

5. Verify the image is created or not by using **docker images** command

```
E:\CMRCET20240105\III Yr\Exp-07\getting-started-app>docker images
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
getting-started-app    latest   3d2d9fbb0a95  12 hours ago  346MB
```

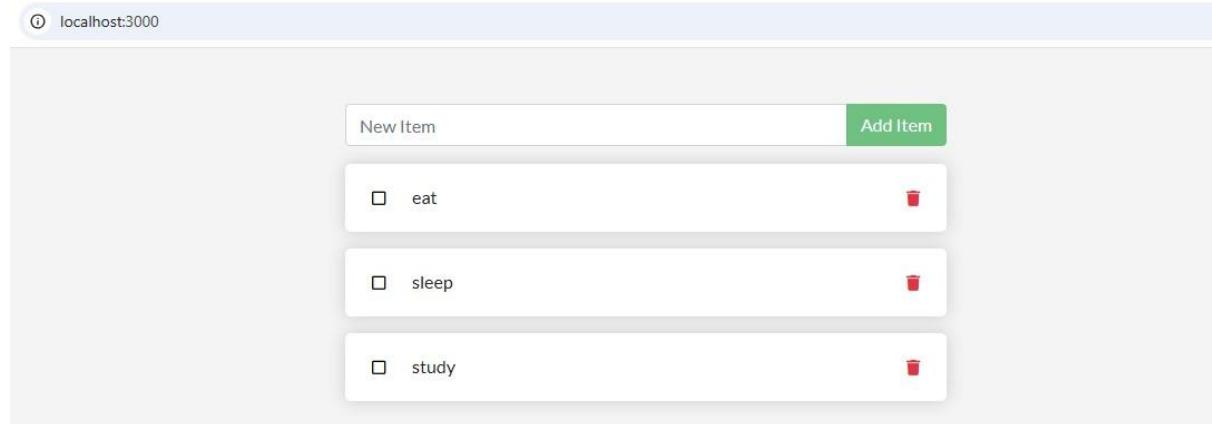
6. Run the image using following command `docker run -dp 127.0.0.1:3000:3000 getting-started-app`

```
E:\CMRCET20240105\III Yr\Exp-07\getting-started-app>docker run -dp 127.0.0.1:3000:3000 getting-started-app
a151d35c8c8247ebf44ac3508d9612d5b47253fb9665498469445e6fc120301b

E:\CMRCET20240105\III Yr\Exp-07\getting-started-app>docker images
REPOSITORY          TAG      IMAGE ID   CREATED        SIZE
getting-started-app    latest   3d2d9fbb0a95  58 minutes ago  346MB
rajshreel023/getting-started-app  latest   3d2d9fbb0a95  58 minutes ago  346MB
mysql                latest   91447968e669  2 weeks ago   1.26GB
ubuntu               latest   353675e2a41b  4 weeks ago   117MB
mysql                8       70fe679fe469  4 weeks ago   1.07GB
nginx               latest   d5f28ef21aab  8 weeks ago   279MB

E:\CMRCET20240105\III Yr\Exp-07\getting-started-app>docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED             STATUS              PORTS               NAMES
a151d35c8c82   getting-started-app   "docker-entrypoint.s..."   About a minute ago   Up About a minute   127.0.0.1:3000->3000/tcp   serene_spence
```

7. Open browser and type <https://localhost:3000> to check the output the image is running on port 3000



## EXPERIMENT-08

### **AIM: Integrate Kubernetes and Docker**

**Objective:** To integrate Kubernetes with Docker for orchestrating containerized applications using Pods, Deployments, and Services.

---

### **Software Requirements**

- Docker installed (<https://www.docker.com/products/docker-desktop/>)
  - Kubernetes installed (Minikube or Docker Desktop with Kubernetes enabled)
  - kubectl command-line tool
  - Code editor (VS Code or any)
  - Sample containerized application
- 

The Kubernetes server runs as a single or multi-node cluster, within Docker container(s). This lightweight setup helps you explore Kubernetes features, test workloads, and work with container orchestration in parallel with other Docker functionalities.

Docker Desktop includes a standalone Kubernetes server and client, as well as Docker CLI integration that runs on your machine.

The Kubernetes server runs locally within your Docker instance, is not configurable, and is a single-node cluster. It runs within a Docker container on your local system, and is only for local testing.

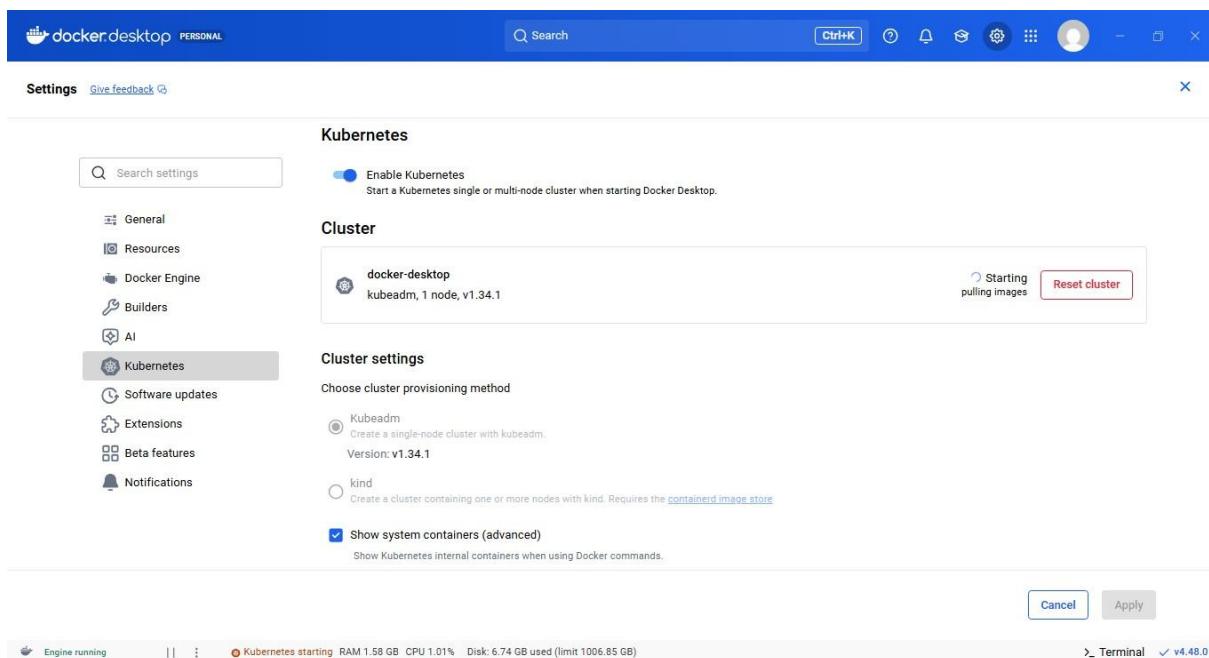
Turning on Kubernetes allows you to deploy your workloads in parallel, on Kubernetes, Swarm, and as standalone containers. Turning on or off the Kubernetes server does not affect your other workloads.

---

### **Lab Procedure**

#### **Install and turn on Kubernetes**

1. Open the Docker Desktop Dashboard and navigate to Settings.
2. Select the Kubernetes tab.
3. Toggle on Enable Kubernetes.
4. Choose your [cluster provisioning method](#).
5. Select Apply to save the settings.



This sets up the images required to run the Kubernetes server as containers, and installs the `kubectl` command-line tool on your system at:

`/usr/local/bin/kubectl` (Mac) or

`C:\Program Files\ Docker\ Docker\resources\bin\kubectl.exe` (Windows).

Docker Desktop for Linux does not include `kubectl` by default. You can install it separately by following the [Kubernetes installation guide](#). Ensure the `kubectl` binary is installed at `/usr/local/bin/kubectl`.

When Kubernetes is enabled, its status is displayed in the Docker Desktop Dashboard footer and the Docker menu.

Docker Desktop does not upgrade your Kubernetes cluster automatically after a new update. To upgrade your Kubernetes cluster to the latest version, select **Reset Kubernetes Cluster**

You can check which version of Kubernetes you're on with: **kubectl version**

```
C:\Users\User>kubectl version
Client Version: v1.34.1
Kustomize Version: v5.7.1
Server Version: v1.34.1
```

## Using the `kubectl` command

Kubernetes integration automatically installs the Kubernetes CLI command at:  
**/usr/local/bin/kubectl** on Mac and

**C:\Program Files\ Docker\ Docker\ Resources\ bin\ kubectl.exe** on Windows.

This location may not be in your shell's **PATH** variable, so you may need to type the full path of the command or add it to the **PATH**.

If you have already installed **kubectl** and it is pointing to some other environment, such as **minikube** or a Google Kubernetes Engine cluster, ensure you change the context so that **kubectl** is pointing to **docker-desktop**:

```
kubectl config get-contexts kubectl
```

```
config use-context docker-desktop
```

```
C:\Users\User>kubectl config get-contexts
CURRENT      NAME          CLUSTER          AUTHINFO          NAMESPACE
*           docker-desktop  docker-desktop  docker-desktop
```

```
C:\Users\User>kubectl config use-context docker-desktop
Switched to context "docker-desktop".
```

#### Tip

If the **kubectl config get-contexts** command returns an empty result, try:

- Running the command in the Command Prompt or PowerShell.
- Setting the **KUBECONFIG** environment variable to point to your **.kube/config** file.

---

## Verify installation

To confirm that Kubernetes is running, list the available nodes: **kubectl get nodes**

```
C:\Users\User>kubectl get nodes
NAME          STATUS    ROLES          AGE     VERSION
docker-desktop  Ready    control-plane  6d1h   v1.34.1
```

---

If you installed **kubectl** using Homebrew, or by some other method, and experience conflicts, remove **/usr/local/bin/kubectl**.

---

## EXPERIMENT-09

**AIM:** Automate the process of running containerized application developed in exercise 7 using Kubernetes

**Objective:** To automate the process of building, running, and managing containerized applications using Docker and scripting techniques.

---

## Software Requirements

- Docker installed (<https://www.docker.com/products/docker-desktop/>)
  - Kubernetes installed (Minikube or Docker Desktop with Kubernetes enabled)
  - kubectl command-line tool
  - Code editor (VS Code or any)
  - Sample containerized application
- 

### Step1:

1. Clone this repository to your local repository <https://github.com/shiv4j/kube>
2. Make sure that cloned repository consist of “my-kube1-deployment.yaml”, “my-kube1-service.yaml” files in folder.
3. Push this local repository to github (or) you can fork that repository from <https://github.com/shiv4j/kube>
4. After completion of pushing or forking of kube1 folder into your github repository you should able to see like this that contains all the files.

The screenshot shows a GitHub repository named "mykube-01". The repository details are as follows:

- Code**: 1 Branch, 0 Tags
- Commits**: 13 Commits (by student-rajshree, 3 hours ago)
- Activity**: 0 stars, 0 forks, 0 watching
- Releases**: No releases published. Create a new release.
- Packages**: No packages published. Publish your first package.

| File / Commit            | Description                      | Time                  |
|--------------------------|----------------------------------|-----------------------|
| jenkinsfile              | modified jenkinsfile             | 39766f9 · 3 hours ago |
| node_modules             | initial commit                   | last year             |
| Dockerfile               | updated files                    | 3 hours ago           |
| Jenkinsfile              | modified jenkinsfile             | 3 hours ago           |
| app.js                   | updated files                    | 3 hours ago           |
| my-kube1-deployment.yaml | updated my-kube1-deployment.yaml | 3 hours ago           |
| my-kube1-service.yaml    | initial commit                   | last year             |
| package-lock.json        | initial commit                   | last year             |
| package.json             | initial commit                   | last year             |

### Step2:

Push your github repository to the Jenkins.

Click on New Item



Enter a name and select Pipeline project and click on Ok.

A screenshot of the Jenkins 'New Item' creation dialog. At the top, it says 'New Item'. Below that is a text input field with the placeholder 'Enter an item name' containing a single character 'I'. A red error message below the field says '» This field cannot be empty, please enter a valid name'. Underneath is a section titled 'Select an item type' with four options: 'Freestyle project', 'Maven project', 'Pipeline', and 'Multi-configuration project'. The 'Pipeline' option is highlighted with a light gray background. Its description says 'Orchestrates long running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.' At the bottom of the dialog is a blue 'OK' button.

- In the configure, go to pipeline tab-> Select Definition as **Pipeline script from SCM-> Select SCM as Git** and paste your repository url
- Specify your branch whether it is main or master based on your github repository.
- Click on Apply and Save.

The screenshot shows the Jenkins Pipeline configuration page. On the left, there's a sidebar with tabs: General, Triggers, Pipeline (which is selected and highlighted in blue), and Advanced. The main area is titled "Pipeline" with the sub-instruction "Define your Pipeline using Groovy directly or pull it from source control." Below this, the "Definition" section is set to "Pipeline script from SCM". Under "SCM", the provider is chosen as "Git". The "Repositories" section contains one repository with the URL "https://github.com/student-rajshree/kube.git" and credentials "student-rajshree/\*\*\*\*\*".

This screenshot shows the continuation of the Jenkins Pipeline configuration. It includes sections for "Branches to build" (with a "Branch Specifier" of "\*/\*main") and "Script Path" (set to "Jenkinsfile"). At the bottom, there are "Save" and "Apply" buttons.

- After creation of your Jenkins project build it-> Click on **Build Now** The build should be shown in green tick mark.

Jenkins / Experiment-09

Status    **Experiment-09**    Add description

</> Changes    Build Now    Configure    Delete Pipeline    Full Stage View    Favorite    Open Blue Ocean    Stages    Rename    Pipeline Syntax

**Stage View**

| Declarative: Checkout SCM                          | Build | Test     | Deploy |    |
|--|-------|----------|--------|----|
| Average stage times:<br>(full run time: ~2min 30s) | 10s   | 1min 59s | 1s     | 1s |
| Oct 29<br>10:03<br>1 commit                        | 10s   | 1min 48s | 1s     | 1s |
| Oct 29<br>09:50<br>No Changes                      | 10s   | 2min 10s | 1s     | 1s |

**Permalinks**

- Last build (#3), 1 hr 7 min ago
- Last stable build (#3), 1 hr 7 min ago
- Last successful build (#3), 1 hr 7 min ago
- Last completed build (#3), 1 hr 7 min ago

You will get docker image for this project, like showed in the below

Docker Desktop PERSONAL

Images / kube:latest

kube:latest [IN USE]

CREATED 1 hour ago SIZE 1.36 GB Recommended fixes Run

Layers (20)

|   |   |           |   |
|---|---|-----------|---|
| 0 | ADD file:40953ed6e6f96703b2e0c13288437c2aaafb3...               | 130.11 MB | ? |
| 1 | CMD ["bash"]  | 0 B       | ? |
| 2 | set -eux; apt-get update; apt-get install -y --no-install-re... | 18.92 MB  | ? |
| 3 | set -ex; if ! command -v gpg > /dev/null; then apt-get u...     | 18.78 MB  | ? |
| 4 | apt-get update && apt-get install -y --no-install-recomm...     | 160.46 MB | ? |
| 5 | set -ex; apt-get update; apt-get install -y --no-install-rec... | 543.22 MB | ? |
| 6 | groupadd -gid 1000 node && useradd --uid 1000 --gid ...         | 405.5 KB  | ? |
| 7 | ENV NODE_VERSION=14.21.3  | 0 B       | ? |
| 8 | ARCH= && dpkgArch="\$(dpkg --print-architecture)" && ...        | 117.58 MB | ? |
| 9 | ENV YARN_VERSION=1.22.19  | 0 B       | ? |

Vulnerabilities Packages

Analyzed by docker scout Give feedback

This image couldn't be analyzed  
It may have an unsupported architecture, or exceed the maximum size

Retry

Engine running    Kubernetes running RAM 1.71 GB CPU 79.83% Disk: 13.60 GB used (limit 1006.85 GB)    Terminal v 4.48.0

### Step3:

- Push the docker image into dockerhub
- open command prompt and run the command “docker login”
- Tag your iamge using this syntax

**docker tag <local-image-name>:<tag> yourusername/image-name:<tag>**

Ex: docker tag kube:latest rajshree023/kube1:latest

Here username is your dockerhub account username

- Push the image to dockerhub **docker push yourusername/image-name:<tag>**

## Ex: docker push rajshree1023/kube1:latest

```
E:\CMRCET20240105\III Yr\Exp-09\kube>docker tag kube:latest rajshree1023/kube1:latest
E:\CMRCET20240105\III Yr\Exp-09\kube>docker push rajshree1023/kube1:latest
The push refers to repository [docker.io/rajshree1023/kube1]
305dd7593bcf: Pushed
d9a8df589451: Pushed
5f32ed3c3f27: Pushed
b253aeafeaa7: Pushed
0c8cc2f24a4d: Pushed
2ff1d7c41c74: Pushed
1de76e268b10: Pushed
3d2201bd995c: Pushed
b1bafef8cac: Pushed
6f51ee005dea: Pushed
0d27a8e86132: Pushed
e267854aef7: Pushed
b60c8eb4f55f: Pushed
82705cc4112d: Pushed
latest: digest: sha256:fbb1b51d3a8ec06709dde30525c936a29b5f23166f00615ef53eabc43126c9e6 size: 856
```

## Step4:

Start the Kubernetes

Syntax: minikube start

```
C:\Users\vijayrubika>minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4391 Build 22631.4391
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Restarting existing docker container for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
* Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
- Using image docker.io/kubernetesui/dashboard:v2.7.0
- Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
* Some dashboard features require the metrics-server addon. To enable all features please run:
  minikube addons enable metrics-server
* Enabled addons: storage-provisioner, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

- Apply my-kube1-deployment.yaml file **kubectl apply -f my-kube1-deployment.yaml**

```
E:\CMRCET20240105\III Yr\Exp-09\kube>kubectl apply -f my-kube1-deployment.yaml
deployment.apps/my-kube1-deployment created
```

- Apply my-kube1-service.yaml file

Syntax: **kubectl apply -f my-kube1-service.yaml**

```
E:\CMRCET20240105\III Yr\Exp-09\kube>kubectl apply -f my-kube1-service.yaml
service/my-kube-deployment created
```

That will apply to the deployments and services

- To check that type command “**kubectl get pods**” and “**kubectl get service**”
- To open the Kubernetes dashboard type = “**minikube dashboard**”

That will open Kubernetes dashboard automatically on your default primary browser

The screenshot shows the Kubernetes Dashboard interface. At the top, there are three tabs: 'Wilsonbolledula/kube' (active), 'kube [Jenkins]', and 'Kubernetes Dashboard'. The URL in the address bar is '127.0.0.1:51408/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/#/workloads?namespace=default'. The main content area is titled 'Workloads' and displays 'Workload Status' with three green circles representing 'Running' counts: 3 for Deployments, 5 for Pods, and 5 for Replica Sets. Below this, the 'Deployments' section lists two entries:

| Name                | Images                          | Labels            | Pods  | Created    |
|---------------------|---------------------------------|-------------------|-------|------------|
| my-kube1-deployment | wilsonbolledula/my-kube1:latest | app: my-kube-app1 | 2 / 2 | a.day.ago  |
| my-kube-deployment  | wilsonbolledula/maven:3         | app: my-kube-app  | 2 / 2 | 2.days.ago |

- Finally you can checkout your deployments and pods here.

## EXPERIMENT-10

### **AIM: Install and Explore Selenium for Automated Testing**

**Objective:** To install Selenium and explore its environment for automating web application testing.

---

## Software Requirements

- Python installed (<https://www.python.org/downloads/>)
  - Selenium library installed
  - Web browser (Chrome, Firefox, etc.)
  - WebDriver for the browser (e.g., ChromeDriver)
  - Code editor (VS Code or any)
- 

## Lab Procedure

### **1. Install Python**

- Download and install Python.
- Verify installation:

```
python --version
```

### **2. Install Selenium Library**

- Use pip to install Selenium:

```
pip install selenium
```

### **3. Download WebDriver**

- Download ChromeDriver from
- (<https://sites.google.com/chromium.org/driver/>) or appropriate WebDriver for your browser.
- Place the WebDriver executable in your system PATH or project directory.



chrome driver for selenium

X | Microphone | Camera | Search

AI Mode All Videos Images Short videos Shopping Forums More Tools

Chrome for Developers  
<https://developer.chrome.com/Docs/ChromeDriver>

## Downloads | ChromeDriver - Chrome for Developers

17 Dec 2024 — Resolved issue 3515: selenium.common.exceptions.WebDriverException: Message:  
unknown error: bad inspector message. For more details, see the ...

### Version selection

Version selection is the process of matching a Chrome binary of a ...

### Get started with ChromeDriver

ChromeDriver is a separate executable that Selenium ...

Open the 1<sup>st</sup> link.

[ChromeDriver](#)

Contribute

Design docs

### Download ChromeDriver

Chrome version 115+

#### Earlier Chrome

Stable Releases

Canary Releases

Version selection

### Get started

Desktop

Home > Docs > ChromeDriver

Was this helpful?

## Downloads

**Warning:** If you're using Chrome version 115 or newer, consult the [Chrome for Testing availability dashboard](#). On this dashboard, you'll find [JSON endpoints](#) to download specific ChromeDriver versions.

### Earlier Chrome versions

For earlier versions of Chrome, here are the versions of ChromeDriver that support it.

For more information on selecting the right version of ChromeDriver, see the [Version Selection](#) page.

ChromeDriver 114.0.5735.90

Supports Chrome version 114

Click on version selection and check your version in chrome by  
`chrome://settings/help`

The screenshot shows the Google Chrome settings interface. On the left, there's a sidebar with various settings categories like Privacy and security, Performance, AI innovations, Appearance, Search engine, Default browser, On startup, Languages, Downloads, Accessibility, System, Reset settings, Extensions, and About Chrome. The 'About Chrome' option is highlighted with a blue bar at the bottom of the sidebar. The main content area is titled 'About Chrome' and features the Google Chrome logo. It displays a message: 'Nearly up to date! Relaunch Chrome to finish updating. Version 141.0.7390.107 (Official Build) (64-bit)' with a 'Relaunch' button. Below this are links for 'Get help with Chrome', 'Report an issue', and 'Privacy policy'. At the bottom, it says 'Google Chrome Copyright 2025 Google LLC. All rights reserved.' and includes links for 'Chromium', 'open source software', and 'Terms of Service'.

Click on the Chrome for Testing (CfT) availability dashboard.

The screenshot shows the 'chrome for developers' website at developer.chrome.com. The top navigation bar includes links for Docs, Case studies, Blog, and New in Chrome, along with a search bar and a globe icon. The main content area is titled 'Version selection' with a dropdown menu. A descriptive paragraph explains that version selection is the process of matching a Chrome binary to a compatible ChromeDriver binary. It notes that starting with M115, the release processes are integrated, and the latest Chrome + ChromeDriver releases per channel (Stable, Beta, Dev, Canary) are available at the [Chrome for Testing \(CfT\) availability dashboard](#). It also mentions the use of CFT JSON endpoints for automated downloads. Below this, another section discusses version selection for older versions, mentioning the 'latest-patch-versions-per-build' JSON endpoint and the 'latest-versions-per-milestone' endpoint as alternatives.

Click on Stable

## Chrome for Testing availability

This page lists the latest available cross-platform Chrome for Testing versions and assets per Chrome release channel.

Consult [our JSON API endpoints](#) if you're looking to build automated scripts based on Chrome for Testing release data.

Last updated @ 2025-10-23T09:10:11.440Z

| Channel                       | Version                               | Revision | Status  |
|-------------------------------|---------------------------------------|----------|---|
| <a href="#"><u>Stable</u></a> | <a href="#"><u>141.0.7390.122</u></a> | r1509326 |    |
| <a href="#"><u>Beta</u></a>   | <a href="#"><u>142.0.7444.52</u></a>  | r1522585 |    |
| <a href="#"><u>Dev</u></a>    | <a href="#"><u>143.0.7475.7</u></a>   | r1530564 |    |
| <a href="#"><u>Canary</u></a> | <a href="#"><u>143.0.7488.0</u></a>   | r1533544 |  |

| Binary                | Platform  | URL   | HTTP status |
|-----------------------|-----------|---|-------------|
| chrome                | linux64   | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/linux64/chrome-linux64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/linux64/chrome-linux64.zip</a>                                       | 200         |
| chrome                | mac-arm64 | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-arm64/chrome-mac-arm64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-arm64/chrome-mac-arm64.zip</a>                               | 200         |
| chrome                | mac-x64   | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-x64/chrome-mac-x64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-x64/chrome-mac-x64.zip</a>                                       | 200         |
| chrome                | win32     | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win32/chrome-win32.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win32/chrome-win32.zip</a>   | 200         |
| chrome                | win64     | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win64/chrome-win64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win64/chrome-win64.zip</a>   | 200         |
| chromedriver          | linux64   | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/linux64/chromedriver-linux64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/linux64/chromedriver-linux64.zip</a>                           | 200         |
| chromedriver          | mac-arm64 | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-arm64/chromedriver-mac-arm64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-arm64/chromedriver-mac-arm64.zip</a>                   | 200         |
| chromedriver          | mac-x64   | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-x64/chromedriver-mac-x64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-x64/chromedriver-mac-x64.zip</a>                           | 200         |
| chromedriver          | win32     | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win32/chromedriver-win32.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win32/chromedriver-win32.zip</a>                                   | 200         |
| chromedriver          | win64     | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win64/chromedriver-win64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win64/chromedriver-win64.zip</a>                                   | 200         |
| chrome-headless-shell | linux64   | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/linux64/chrome-headless-shell-linux64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/linux64/chrome-headless-shell-linux64.zip</a>         | 200         |
| chrome-headless-shell | mac-arm64 | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-arm64/chrome-headless-shell-mac-arm64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-arm64/chrome-headless-shell-mac-arm64.zip</a> | 200         |
| chrome-headless-shell | mac-x64   | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-x64/chrome-headless-shell-mac-x64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/mac-x64/chrome-headless-shell-mac-x64.zip</a>         | 200         |
| chrome-headless-shell | win32     | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win32/chrome-headless-shell-win32.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win32/chrome-headless-shell-win32.zip</a>                 | 200         |
| chrome-headless-shell | win64     | <a href="https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win64/chrome-headless-shell-win64.zip">https://storage.googleapis.com/chrome-for-testing-public/141.0.7390.122/win64/chrome-headless-shell-win64.zip</a>                 | 200         |

- Select according to your Operating System and Copy and paste chromedriver link in chrome it will download automatically.
  
- The file will be downloaded, extract the files and set the path of chromedriver.exe in system environment variables

- Verify chromeDriver installation **chromeDriver –version**
- Open a VS code and create a python file and Add the following code

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

# Specify the correct path to your ChromeDriver chrome_service
= Service("C:\Program
Files\chromedriverwin64\chromedriver.exe")

# Set up the WebDriver for Chrome
driver = webdriver.Chrome(service=chrome_service)

try:
    # Step 1: Open Google in the browser
    driver.get("https://www.google.com")

    # Step 2: Locate the search box using the name attribute
    search_box = driver.find_element(By.NAME, "q")

    # Step 3: Enter the search term and press Enter
    search_box.send_keys("https://cmrcet.ac.in/")
    search_box.send_keys(Keys.RETURN)

    # Optional: Wait for a few seconds to see the search results
    time.sleep(20)

finally:
    # Step 4: Close the browser
    driver.quit()
```

**Make sure to change the path of the driver in the code to avoid errors**

- Run the code

## **OUTPUT**

The screenshot shows a Google search results page for the query "cmrcet.ac.in". The top result is a link to the CMR College of Engineering & Technology website, which includes the college's logo, name, address, and contact information. Below it, there are links to "CMRCET Results" and "Admissions 2025-26". The search interface includes standard Google controls like AI Mode, All, Images, News, Videos, Short videos, Shopping, More, Tools, and Sign in.

https://cmrcet.ac.in/ - Google Search

Google https://cmrcet.ac.in/

AI Mode All Images News Videos Short videos Shopping More Tools Sign in

cmrcet.ac.in https://cmrcet.ac.in › about

About – CMR College of Engineering & Technology

CMR College of Engineering & Technology Kandlakoya(v), Medchal Road Hyderabad, Telangana, India - 501401 · principal@cmrcet.ac.in · +91 9248727210. Copyright ...

CMRCET Results https://results.cmrcet.ac.in

CMRCET Results

CMR College of Engineering and Technology. Examination Results. CMRCET Results. Exam Year, All, 2025, 2024, 2023. Filters: All; B.Tech; MBA; M.Tech; M.Tech. All

cmrcet.ac.in https://cmrcet.ac.in › admissions-2025-26

[Admissions 2025-26 – CMR College of Engineering ...](#)

CMR College of Engineering & Technology Kandlakoya(v), Medchal Road Hyderabad, Telangana, https://cmrcet.ac.in/admissions-2025-26/ ac.in · +91 9248727210. Copyright

## EXPERIMENT-11

**AIM:** Write a simple program in JavaScript and perform testing using Selenium.

**Objective:** To develop automated test cases using Selenium for validating the functionality of a previously containerized web application.

---

### Software Requirements

- Python or Node.js environment (as per project)
  - Selenium WebDriver installed
  - Web browser (Chrome, Firefox, etc.)
  - WebDriver for the browser (e.g., ChromeDriver)
  - Docker installed and running
  - Code editor (VS Code or any)
  - Sample containerized web application running
- 

### Lab Procedure

1. Open VS code and create html and javascript

files 2. Create html file and paste this code

#### **index.html**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sum Calculator</title>
    <
    s
    t
    y
    l
    e
    >
    b
```

```
o
d
y
{
text-align: center;
}
</style>
</head>
<body>
<h1>Sum Calculator</h1>
<input type="number" id="num1" placeholder="Enter first number">
<input type="number" id="num2" placeholder="Enter second number">
<button id="add">Add</button>
<p>Result: <span id="result">0</span></p>
<script>  function
calculateSum(a, b)
{ return a + b;
} document.getElementById('add').addEventListener('click', function() { const num1
= parseInt(document.getElementById('num1').value, 10); const num2 =
parseInt(document.getElementById('num2').value, 10); const result =
calculateSum(num1, num2); document.getElementById('result').textContent =
result; });
</script>
</body>
</html>
```

---

```
script.js import { fileURLToPath } from 'url'; import { dirname } from 'path'; import { Builder, By,
until } from 'selenium-webdriver'; import assert from 'assert';
```

```
// Define __dirname manually const __filename
= fileURLToPath(import.meta.url); const
__dirname = dirname(__filename);
```

```
async function runTest() { // Define driver here let driver =  
await new Builder().forBrowser('chrome').build();  
  
try {  
    // Use __dirname (not dirname) await  
    driver.get('file://' + __dirname + '/index.html');  
  
    // Find input elements const num1 = await  
    driver.findElement(By.id('num1')); await  
    num1.sendKeys('50');  
  
    const num2 = await driver.findElement(By.id('num2'));  
    await num2.sendKeys('10');  
  
    // Click the "Add" button const addButton = await  
    driver.findElement(By.id('add')); await  
    addButton.click();  
  
    // Verify the result const result = await  
    driver.findElement(By.id('result')); const text = await  
    result.getText();  
  
    assert.strictEqual(text, '60', 'Sum calculation is incorrect');  
    console.log('✓ Test passed: Sum is correct');  
} catch (error) {  
    console.error('✗ Test failed:',  
    error);  
} finally { console.log('Press  
any key to exit...');  
process.stdin.setRawMode(true  
); process.stdin.resume();  
process.stdin.on('data', async ()
```

```
=> {    await driver.quit();
process.exit(0);
});

}

}

r
u
n
T
e
s
t
(
)
;
```

---

### 3. Open terminal

#### 4. npm install selenium.webdriver

```
E:\CMRCET20240105\III Yr\Devops Lab Exp Final\Exp-11>npm install selenium.webdriver
up to date, audited 18 packages in 6s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

#### 5. node script.js

```
E:\CMRCET20240105\III Yr\Devops Lab Exp Final\Exp-11>node script.js
(node:18864) [MODULE_TYPELESS_PACKAGE_JSON] Warning: Module type of file:///E:/CMRCET20240105/III%20Yr/Devops%20Lab%20Exp%20Final/Exp-11/script.js is not specified and it doesn't parse as CommonJS.
Reparsing as ES module because module syntax was detected. This incurs a performance overhead.
To eliminate this warning, add "type": "module" to E:\CMRCET20240105\III Yr\Devops Lab Exp Final\Exp-11\package.json.
(Use "node --trace-warnings ..." to show where the warning was created)

DevTools listening on ws://127.0.0.1:51648/devtools/browser/b1b48b41-ebb4-4c3e-94a2-d307ea682426
Test passed: Sum is correct
Press any key to exit...
```

---

## OUTPUT

Sum Calculator

File E:/CMRCET20240105/III%20Yr/Devops%20Lab%20Exp%20Final/Exp-11/index.html

## Sum Calculator

49      10      Add

Result: 59

---

## EXPERIMENT-12

**AIM:** Develop test cases for the above containerized application using selenium.

**Objective:** To validate the functional, UI, authentication, and integration behavior of the web application running inside Docker by automating end-to-end tests using Selenium (remote WebDriver / Selenium container).

---

### Software Requirements

- Code editor (VS Code or any)
  - Docker Desktop
  - Python (for running scripts locally if needed)
  - Selenium (pip install selenium)
  - Web browser (Chrome, Firefox, etc.)
- 

### Lab Procedure

Clone this repository to your local repository <https://github.com/Srivaishnavi08/tests>

**Or follow the below steps**

**Step-1:**

Create a directory named selenium-test and navigate to the current directory path. tests/

```
|-  
Dockerfile  
|-  
index.html  
ml  
|- SeleniumTest.py  
|- docker-compose.yml
```

---

**Step 2:** Create the index.html file

This is your sample web page that Selenium will interact with.

**index.html**

```
<!DOCTYPE html>
```

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Selenium Login Example</title>
</head>
<body>
    <!-- Homepage -->
    <div id="homepage">
        <h1>Welcome!!!!!!</h1>
        <!-- Get Started Free Button -->
        <a href="#loginPage" id="get-started" onclick="navigateToLogin()">Get
started</a>
    </div>

    <!-- Login Page -->
    <div id="loginPage" style="display: none;">
        <h2>Login to the Page</h2>
        <form onsubmit="return validateLogin()">
            <label for="user_email_login">Email:</label>
            <input type="email" id="user_email_login" name="user_email_login"
required>
            <br><br>
            <label for="user_password">Password:</label>
            <input type="password" id="user_password" name="user_password"
required>
            <br><br>
            <button type="submit" name="commit">Login</button>
        </form>
        <p id="error-message" style="color: red; display: none;">Invalid credentials,
please try again.</p>
    </div>
</body>
</html>
```

```
</div>

<!-- Dashboard Section (only shown after successful login) -->
<div id="dashboard" style="display: none;">
    <h2>Welcome to Your Dashboard!</h2>
    <p>This is the dashboard area you see after a successful login.</p>  </div>

<script>
    // Function to navigate to the login page      function
navigateToLogin() {
    document.getElementById('homepage').style.display = 'none';
    document.getElementById('loginPage').style.display = 'block';
}

    // Function to validate login credentials      function validateLogin() {
const email = document.getElementById('user_email_login').value;
const password = document.getElementById('user_password').value;

    if (email === "abc@gmail.com" && password === "password") {
        // Hide login page and display dashboard
        document.getElementById('loginPage').style.display = 'none';
        document.getElementById('dashboard').style.display = 'block';      return
false; // Prevent actual form submission
    } else {
        // Show error message if credentials are incorrect
        document.getElementById('error-message').style.display = 'block';      return
false; // Prevent actual form submission
    }
}

</script>
</body>
</html>
```

---

### **Step 3: Create the dockerfile**

This dockerfile sets up a simple HTTP server to serve your index.html file.

#### **Dockerfile**

```
# Use the official Python image as the base image
```

```
FROM python:3.9
```

```
# Set the working directory in the container
```

```
WORKDIR /app
```

```
# Copy the index.html file to the container
```

```
COPY index.html .
```

```
# Expose port 8000 for the HTTP server
```

```
EXPOSE 8000
```

```
# Start a simple HTTP server to serve the index.html file
```

```
CMD ["python", "-m", "http.server", "8000"]
```

---

### **Step-4: Create the Selenium Test Script (seleniumTest.py)**

This script will automate testing of your HTML page using Selenium.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time

print("Test Execution Started")
options = webdriver.ChromeOptions()
options.add_argument('--ignore-ssl-errors=yes')
options.add_argument('--ignore-certificate-errors')
```

```
# Start the Selenium WebDriver driver =
webdriver.Remote(
    command_executor='http://localhost:4444/wd/h
ub', options=options
)

# Maximize the window size driver.maximize_window() time.sleep(10)
driver.get("http://host.docker.internal:8000") # Access the local server
time.sleep(10)

try:
    # Wait for the "Get started free" link to be clickable
    link = WebDriverWait(driver, 30).until(
        EC.element_to_be_clickable((By.LINK_TEXT, "Get started"))
    )
    link.click() # Click the link time.sleep(10) #
    Wait for any resulting page to load

    WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "user_email_login"))
    )
    WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.ID,
"user_password")))
)

# Enter login credentials username =
driver.find_element(By.ID, "user_email_login") password =
driver.find_element(By.ID, "user_password") login_button
= driver.find_element(By.NAME, "commit")

username.send_keys("abc@gmail.com") # Replace with actual username
password.send_keys("password") # Replace with actual password
login_button.click()
```

```

# Check for a post-login element (adjust to your page's unique element for
logged-in users) try:
    error_message = WebDriverWait(driver, 10).until(
        EC.visibility_of_element_located((By.ID, "error-message"))
    )
    time.sleep(10)      print("Login
failed: Incorrect credentials")  except:
    # No error message found, proceed with checking for dashboard
    WebDriverWait(driver, 10).until(
        EC.visibility_of_element_located((By.ID, "dashboard")) # Replace with actual
post-login element ID
    )
    print("Login Successful!")

except Exception as e:
    print(f"An error occurred while trying to click the link: {e}")
finally:
    # Ensure the browser quits after
execution  driver.quit()  print("Test
Execution Completed!")

```

### **Step 5:** Create the docker-compose.yml File

This file defines two services: your HTML server and the Selenium Chrome container.

```

(docker-
compose.y
ml)
services:
app:
build:
context: .

```

```
dockerfile: Dockerfile  
container_name: html-server  
ports:  
- "8000:8000"
```

```
selenium:  
image: selenium/standalone-chrome  
container_name: selenium-chrome  
ports:  
- "4444:4444" depends_on:  
- app
```

---

## Step-6:

- Build and Run Your Docker Containers
- In terminal use the command

**docker compose -f docker-compose up --build**

This command will:

- Build the Docker image for your HTML server.
- Pull the Selenium standalone Chrome image.

```
E:\CMRCET20240105\III Yr\Devops Lab Exp Final\Exp-12\tests>docker compose -f docker-compose.yml up --build  
[+] Building 59.4s (11/11) FINISHED  
=> [internal] load local bake definitions  
=> => reading from stdin 564B  
=> [internal] load build definition from Dockerfile  
=> => transferring dockerfile: 396B  
=> [internal] load metadata for docker.io/library/python:3.9  
=> [auth] library/python:pull token for registry-1.docker.io  
=> [internal] load .dockignore  
=> => transferring context: 28  
=> [1/3] FROM docker.io/library/python:3.9@sha256:61c518a24fa2c5e6c2ead2b29bc0c81ff7691d6f36459d2e399d4e25ddc0db38  
=> => resolve docker.io/library/python:3.9@sha256:61c518a24fa2c5e6c2ead2b29bc0c81ff7691d6f36459d2e399d4e25ddc0db38  
=> [internal] load build context  
=> => transferring context: 328  
=> CACHED [2/3] WORKDIR /app  
=> CACHED [3/3] COPY index.html .  
=> exporting to image  
=> => exporting layers  
=> => exporting manifest sha256:18dc5ae8632caeab87ebe9338ad9fcbe1f5204f3fa1404eed6c1738767c57e01  
=> => exporting config sha256:cd57edbd8d681ab6fed02b7e93fd48a487c1521f71405276f4aa/efef671b7  
=> => exporting attestation manifest sha256:373fd5b1425368273f5e8447643e08fb72621bc267f8535de450076421113  
=> => exporting manifest list sha256:b42a91e6f2e9efb02fe913c58fd9352079209ebc7e37ddfe148673c189887d  
=> => naming to docker.io/library/tests-app:latest  
=> => unpacking to docker.io/library/tests-app:latest  
=> resolving provenance for metadata file  
[+] Running 2/2  
  tests-app          Built  
  Container html-server Recreated  
Attaching to html-server, selenium-chrome  
selenium-chrome | Virtual environment detected at /opt/venv, activating...  
selenium-chrome | Python 3.12.3  
selenium-chrome | 2025-10-28 16:40:58,888 INFO Included extra file "/etc/supervisor/conf.d/chrome-cleanup.conf" during parsing  
selenium-chrome | 2025-10-28 16:40:58,895 INFO Included extra file "/etc/supervisor/conf.d/recorder.conf" during parsing  
selenium-chrome | 2025-10-28 16:40:58,896 INFO Included extra file "/etc/supervisor/conf.d/selenium.conf" during parsing  
selenium-chrome | 2025-10-28 16:40:58,899 INFO Included extra file "/etc/supervisor/conf.d/uploader.conf" during parsing  
selenium-chrome | Unlinking stale socket /tmp/supervisor.sock  
selenium-chrome | 2025-10-28 16:40:59,539 INFO RPC interface 'supervisor' initialized  
selenium-chrome | 2025-10-28 16:40:59,541 INFO supervisor started with pid 10  
selenium-chrome | 2025-10-28 16:41:00,551 INFO spawned: 'xvfb' with pid 11  
selenium-chrome | 2025-10-28 16:41:00,598 INFO spawned: 'vnc' with pid 12  
selenium-chrome | 2025-10-28 16:41:01,087 INFO spawned: 'novnc' with pid 13  
selenium-chrome | 2025-10-28 16:41:01,293 INFO spawned: 'selenium-standalone' with pid 15  
selenium-chrome | 2025-10-28 16:41:01,611 INFO success: xvfb entered RUNNING state, process has stayed up for > than 0 seconds (startsecs)
```

- Start both services.
- Press v to navigate to docker.

|              |              |                      |       |                  |        |         |  |  |  |  |
|--------------|--------------|----------------------|-------|------------------|--------|---------|--|--|--|--|
| tests        | -            | -                    | 3.18% | 195.24MB / 3.63G | 10.51% | 160.5Mi |  |  |  |  |
| selenium-chr | 85ad964df47d | selenium/s 4444:4444 | 3.15% | 181.7MB / 1.81Gi | 9.78%  | 142MB   |  |  |  |  |
| html-server  | f641b54bf357 | tests-app 8000:8000  | 0.03% | 13.54MB / 1.81Gi | 0.73%  | 18.5MB  |  |  |  |  |

- Click on the links

**8000:8000**



**4444:4444**

URI: http://172.18.0.3:4444

Stereotypes: v.141.0

Sessions: 1 Max. Concurrency: 1

Help - All rights reserved - Software Freedom Conservancy 2025.

Queue size: 0

### Step 7:

Run the Selenium Test Script

While your containers are running, open a new terminal and run:

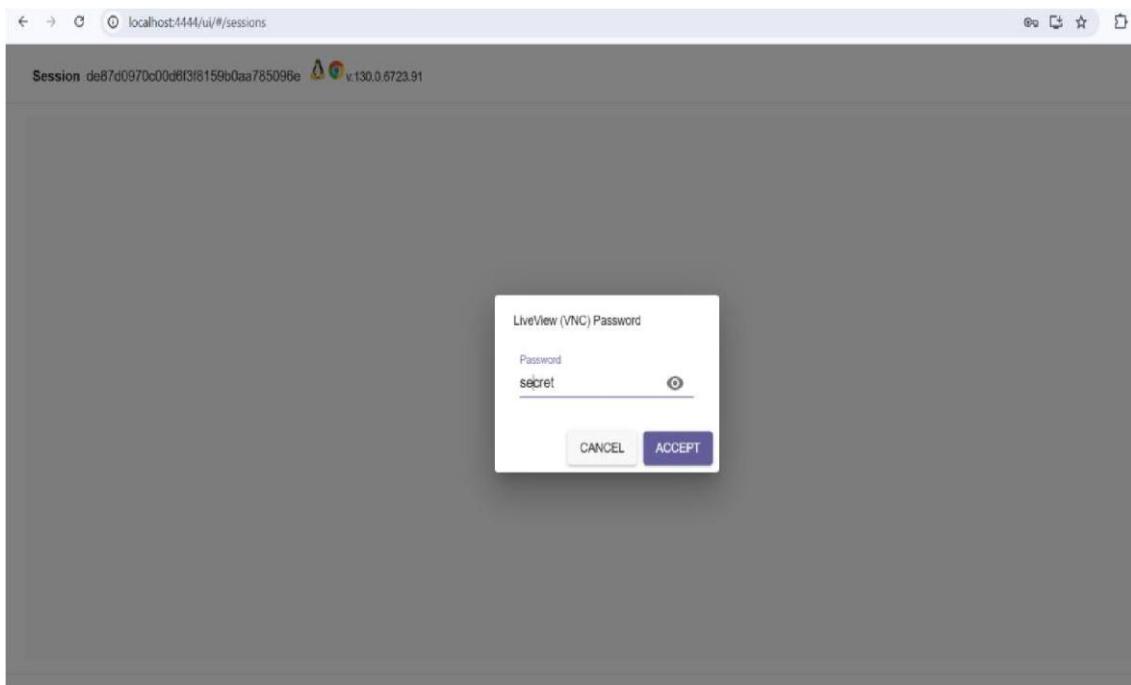
**python seleniumTest.py**

Go to sessions, click on video icon.

| Session                          | Capabilities    | Start time          | Duration | Node URI               |
|----------------------------------|-----------------|---------------------|----------|------------------------|
| de87d0970c00d6f3f8169b0aa785096e | v.130.0.6723.91 | 13/11/2024 17:26:37 | 2.5s     | http://172.18.0.3:4444 |

Concurrency  
100%  
1 / 1

Enter the password “secret” to view the video of automatic testing.



#### Step 8:

Stop the Docker Containers Once you are done testing.

Stop the containers using: **docker compose -f docker-compose down**

This command stops and removes the containers, networks, and any volumes associated with the services.

After running the test and clicking the video icon, the video will show how the test interacts with the web page (clicking buttons, entering passwords, etc.)