

## Experiment 1:

Demonstrate use of git commands to push and pull 3 html files from Github account.

<https://chat.openai.com/share/1dfd8076-b6f5-4ee6-8d56-369060865a52>

### Files to be created:

index.html

about.html

contact.html

### After pushing, code to change in index.html:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

```
</html>
```

**Experiment 2:**

Demonstrate Building java program (addition of 2 numbers) through Jenkins.

<https://chat.openai.com/share/0f86002e-89c0-4054-82a3-1d459830d54d>

File name: add.java

Code:

```
public class add {  
    public static void main(String[] args) {  
        int number1 = 5; // You can change these numbers  
        int number2 = 3; // You can change these numbers  
        int sum = number1 + number2;  
        System.out.println("Sum: " + sum);  
    }  
}
```

Code to run in jenkins:

```
javac add.java  
java add
```

### Experiment 3:

Demonstrate Continuous Integration process in Jenkins. Build a java program every 2 minutes (addition of two numbers) residing in Github repository.

<https://chat.openai.com/share/c9806fea-c1c9-4cbf-b7f7-1941d846dc6d>

File name: add.java

Code:

```
public class add {  
    public static void main(String[] args) {  
        int number1 = 5; // You can change these numbers  
        int number2 = 3; // You can change these numbers  
        int sum = number1 + number2;  
        System.out.println("Sum: " + sum);  
    }  
}
```

In the Schedule box, enter `H/2 * * * *` to poll the repository every 2 minutes.

Under Source Code Management, where you entered the repository URL, change the branch specifier from `*/master` to `*/main`.

Code to run in jenkins:

```
javac add.java  
java add
```

#### Experiment 4:

Demonstrate Continuous Integration process in Jenkins. Build a java program every 2 minutes (addition of two numbers) residing in Github repository.

<https://chat.openai.com/share/ca666849-1df8-4712-b188-d557a8ddc448>

#### Files to be created:

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Home Page</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This is the home page. You can find more information on other pages.</p>
</body>
</html>
```

about.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>About Us</title>
</head>
<body>
  <h1>About Us</h1>
  <p>This page tells you about our website and the services we offer.</p>
</body>
</html>
```

contact.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contact Us</title>
</head>
<body>
  <h1>Contact Us</h1>
  <p>Here you can find how to reach us:</p>
</body>
</html>
```

Pipelining code:

```
pipeline {
  agent any

  stages {
    stage('Clone Repository') {
      steps {
        git branch: 'main', url: 'your repo link here' // Replace with your actual GitHub repository URL
      }
    }

    stage('Deploy to XAMPP') {
      steps {
        script {
          // Assuming XAMPP is installed at C:\xampp\htdocs on a Windows server
          bat "xcopy /Y /I /E /F /C *.html C:\\xampp\\htdocs\\"
        }
      }
    }
  }
}
```

### Experiment 5:

Demonstrate CI/CD in Jenkins. Consider 3 html files in Github repository and write a declarative pipeline to deploy on Xampp server.

<https://chat.openai.com/share/35b72747-995d-4e54-9d58-aa29b1fee659>

#### Files to be created:

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Home Page</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This is the home page. You can find more information on other pages.</p>
</body>
</html>
```

about.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>About Us</title>
</head>
<body>
  <h1>About Us</h1>
  <p>This page tells you about our website and the services we offer.</p>
</body>
</html>
```

contact.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contact Us</title>
</head>
<body>
  <h1>Contact Us</h1>
  <p>Here you can find how to reach us:</p>
</body>
</html>
```

### Pipelining Code:

```
pipeline {
  agent any

  stages {
    stage('Checkout') {
      steps {
        // Replace 'your repo link here' with your actual repository URL
        git branch: 'main', url: 'your repo link here'
      }
    }

    stage('Build') {
      steps {
        echo 'Building...'
      }
    }

    stage('Deploy') {
      steps {
        echo 'Deploying...'
        bat "xcopy /Y /I /E /F /C *.html C:\\xampp\\htdocs\\"
      }
    }
  }
}
```

### **Experiment 6:**

Write a Selenium script to perform automated testing.

<https://chat.openai.com/share/fe2a80d9-a2e5-440f-9822-caf9d75dcabd5>

File name: main.py

Terminal: pip install selenium

Code:

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

chrome_options = webdriver.ChromeOptions()
chrome_options.add_experimental_option("detach", True)

driver = webdriver.Chrome(chrome_options)
driver.get(url="https://www.google.com/")

time.sleep(3)
search = driver.find_element(By.XPATH, value='//*[@id="APjFqb"]')
search.send_keys("What is Selenium?", Keys.ENTER)

time.sleep(3)
driver.quit()

print("Selenium automated testing successful")
```



## Experiment 7:

Create an image of php project and push on Dockerhub repository.

<https://chat.openai.com/share/4573ef5c-bce0-40eb-b7a2-1cb8ac3beb90>

File Name: index.php

PHP Code:

```
<?php
echo "Hello, Docker!";
?>
```

File Name: dockerfile

dockerfile Code:

```
FROM php:7.4-apache
WORKDIR /var/www/html
COPY . /var/www/html
EXPOSE 80
```

## Open terminal from your folder:

1st command:

```
docker build -t "your-folder-name" .
```

2nd command:

```
docker run -p 8090:80 -d "your-folder-name"
```

3rd command:

```
docker login
```

4th command:

```
docker tag "your-folder-name" "your-docker-username"/"your-folder-name":latest
```

5th command:

```
docker push "your-docker-username"/"your-folder-name":latest
```

### Experiment 8:

Demonstrate use of docker commands to pull ubuntu official image, create a file and push updated image on Dockerhub.

<https://chat.openai.com/share/dccd02cb-871c-481e-938f-df0e490ec840>

### Open command prompt:

#### 1st command:

docker login

#### 2nd command:

docker pull ubuntu

#### 3rd command:

docker run -it --name myubuntu ubuntu bash

(Above command will open docker terminal in command prompt only. So don't change anything, just keep pasting the commands in the correct order as I have given. **Khud ka dimag maat lagao.**)

#### 4th command:

echo "Hello from Docker!" > hello.txt

#### 5th command:

cat hello.txt

#### 6th command:

exit

#### 7th command:

docker commit myubuntu "your-docker-username"/ubuntu-custom

#### 8th command:

docker push "your-docker-username"/ubuntu-custom

```
Command Prompt
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sarve>docker login
Authenticating with existing credentials...
Login Succeeded

C:\Users\sarve>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
fdcaa7e87498: Pull complete
Digest: sha256:562456a05a0dbd62a671c1854868862a4687bf979a96d48ae8e766642cd911e8
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu

C:\Users\sarve>docker run -it --name myubuntu ubuntu bash
root@80d55a9f02cf:/# echo "Hello from Docker!" > hello.txt
root@80d55a9f02cf:/# cat hello.txt
Hello from Docker!
root@80d55a9f02cf:/# exit
exit

C:\Users\sarve>docker commit myubuntu sarvesh2109/ubuntu-custom
sha256:e723cc8ab8e7e107fe6c89d17a5efd2fb2c4e7f06a38ee3334f1be1777d66943

C:\Users\sarve>docker push sarvesh2109/ubuntu-custom
Using default tag: latest
The push refers to repository [docker.io/sarvesh2109/ubuntu-custom]
cbf336c6fb88: Pushed
3e1ed584ae0e: Mounted from library/ubuntu
latest: digest: sha256:331a4aa4b373020a02ed8baeb0485ba9dde6619f819b85e200b1bcd8b1f6442 size: 736

C:\Users\sarve>
```