EXP2:

sudo apt update

su root

sudo apt-get install git

git –version

git config –global user.name “Ananya\_Mishra22”

git config –global user.email “ananyamishra22105@gmail.com”

git config –list

next go to git and create new repo: it should be public and tick create README.md

back to terminal create new folder

mkdir ananya

cd ananya

git init

git clone your\_repo\_url

ls

cd hello\_test/

ls

cat README.md

touch hello.py

gedit hello.py

print(“Hello World”)

git status

get add hello.py

git status

get commit -m “First commit”

git status

git remote set-url origin <https://tokenhere@github.com/user_name/repo_name>

git push -u origin main

git log

git show

git diff

git revert<secondcommitID>

EXP 3:

Prerequisite: of 2nd exp

git branch feature

git branch

git checkout feature

git branch

echo “This text was added orginally while in feature branch” >> hello.py

cat hello.py

git status

git add hello.py

git status

git commit -m “Added a line in feature branch”

git log

git checkout main

git branch

git merge feature

cat hello.py

git remote set-url origin <https://tokenhere@github.com/user_name/repo_name>

git push -u origin main

git branch

git branch –d feature

git branch

EXP4:

sudo apt-get update

sudo systemctl start jenkins.service

sudo systemctl status jenkins

sudo ufw status

sudo ufw allow 8080

sudo ufw enable

Go to browser localhost:8080

Goto new item and give a name and select freestyle project

Goto general write anything in description

In general go to source code management and write this url:

<https://github.com/sujataoak799/HelloWorld_29072024.git>

Click on "Add build step" and select the "Execute Shell".

In execute shell write this:

javac Simple.java

java Simple

click on Build Now option.

Click on the build number #2 in the Build History section.

EXP 6:

In jenkins dashboard ,goto manage jenkins ->manage nodes

Select new node,enter a name,select permanent agent,click OK

Goto terminal pwd,you will get a directory

su root

find / -type f -name java

Find for openjdk-amd64

Fill in the details in nodes and then

Under ‘Node Properties’, provide jdk path(amd-64),save it

goto Manage Jenkins -> Security

Allow the port number provided

sudo ufw allow 50000/tcp

Nodes -> agent2 there will be command provided copy and paste them in terminal

After this this is ready for connection,goto new item create a new project(fresstyle)

Then in general write description anything and tick restrict where this project can be run,

and write the node name below it,in execute shell write echo “Anything” and save

Click build now and console output you will get success

Goto Jenkins Dashboard->Manage Jenkins->Nodes->agent2

EXP 7:

Goto firefox into browser search for selenium extension

Click on add

Next click on the puzzel icon in the right side of the browser

Click on selenium

Click on Create a new project

Give a name and click ok

Create a basic test case in Selenium ide.

Give a name

Then click on start recoding

Add <https://google.com/>

And hit ok

Search something and browse around the stop recoding

Then hit run test cases in the top bar of selenium ide

Click save on top menu bar

EXP 8:

sudo su

# curl -fsSL https://get.docker.com -o get-docker.sh

# sudo sh get-docker.sh

ls

cat get-docker.sh

# docker –version

#docker login

#docker run hello-world

#docker run -it ubuntu bash

# docker images

docker pull <image name>

#docker pull mysql

docker ps

docker ps –a

Now to get the container running : #docker run -it ubuntu bash

In New terminal: docker ps –a

In First Terminal: ls

Now get exit from ubuntu container

In New terminal: docker ps –a

So You see the container has exited 3 minutes ago.

docker exec -it <container id> bash

#docker rm <container-name/container-id>

#docker rmi <image-name/image-id>

Commands related to containers:

docker start <container-id> : Start the stop container

docker stop <container-id> : Stop the running container

docker pause <container-id> : Pause the processes in running container

docker kill <container id> : Kill the container.

#docker pull mysql

#docker images

docker run --name test-mysql -e MYSQL\_ROOT\_PASSWORD=strong\_password -d mysql

In New Terminal:

$ docker exec -it container\_name bash

Open the bash terminal of test-mysql:

$ docker exec -it test-mysql bash

$ mysql -u root -p

Enter password: ...

mysql>

mysql> SELECT ‘hello-world!!!’;

mysql> show databases;

Run few mysql commands