JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

UNIVERSITY COLLEGE OF ENGINEERING SCIENCE AND TECHNOLOGY
HYDERABAD, KUKATPALLY, HYDERABAD – 500085

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

This is to certify that _____ of CSE(IDP) III year I Semester bearing the Hall-Ticket number _____ has fulfilled his DevOps Lab record for the academic year 2024-2025.

_____   _____

Signature of the Head of the Department   *Signature of the Staff Member*

Date of Examination_____

_____   _____

Internal Examiner   *External Examiner*

Contents

1.Write code for a simple user registration form for an event.

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Registration Form</title>
</head>
<body>
<form id="registrationForm">
  <label for="fullName">Full Name:</label><br>
  <input type="text" id="fullName" name="fullName" required><br>
  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email" required><br>
  <label for="password">password:</label><br>
  <input type="password" id="password" name="password" required><br>
  <br>
  <input type="submit" value="Register">
</form>
<div id="successMessage" style="display: none;">
  <p>Registration Successful!</p>
</div>
<script>
  document.getElementById("registrationForm").addEventListener("submit", function(event) {
    event.preventDefault();
    document.getElementById("successMessage").style.display = "block";
    document.getElementById("registrationForm").reset();
  });
```
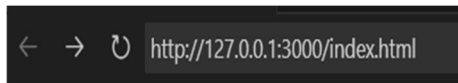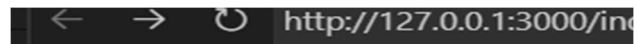
```
</script>


</body>

</html>
```


Full Name:

Email:

password:

Register

Full Name:
alle karthik
Email:
alle223@gmail.com
password:
••••

Register

Registration Successful!

2.Explore Git and GitHub commands.

Git and GitHub are two of the most popular tools used for version control and collaboration in software development.

Here are some common Git and GitHub commands.

- Initializing a git repository: $git init

```
C:\Users\allek>git init
Initialized empty Git repository in C:/Users/allek/.git/
```

-
- Checking the status of your repository: $ git status

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

-
- Adding files to the stage: $ git add

- Committing changes: $ git commit -m "commit message"

- Checking the commit history: $ git log

```
C:\Users\allek>git log
fatal: your current branch 'master' does not have any commits yet
```

- Undoing changes: $ git checkout

- Creating a new branch: $ git branch

- Switching to a different branch: $ git checkout

- Merging two branches: $ git merge

- Pushing changes to a remote repository: $ git push origin

- Cloning a repository from GitHub: $ git clone

```
C:\Users\allek>git clone
fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

    -v, --[no-]verbose      be more verbose
    -q, --[no-]quiet        be more quiet
    --[no-]progress         force progress reporting
    --[no-]reject-shallow   don't clone shallow repository
    -n, --no-checkout       don't create a checkout
    --checkout              opposite of --no-checkout
    --[no-]bare             create a bare repository
    --[no-]mirror           create a mirror repository (implies --bare)
    -l, --[no-]local        to clone from a local repository
    --no-hardlinks          don't use local hardlinks, always copy
    --hardlinks             opposite of --no-hardlinks
    -s, --[no-]shared       setup as shared repository
    --[no-]recurse-submodules[=<pathspec>]
                            initialize submodules in the clone
    --[no-]recursive ...    alias of --recurse-submodules
    -j, --[no-]jobs <n>     number of submodules cloned in parallel
```

• Creating a pull request on GitHub: Go to the repository on GitHub, select the branch you want to merge and click the "New pull request" button.

3.Practice Source code management on GitHub. Experiment with the source code written in exercise 1

Here a step by step procedure to practice source code management on github using source code written in exercise 1

1:sign up/login to github

If you don't have a github account signup for one

If you already have an account login to github

2:create new repository

Click on "+" sign on the top right corner of the hithub page

Select new repository

Fill the repository name ,descriotion and other details

Optionally choose to initialize the repository with a readme file

Click on create repository

3:open any text editor like vs code

Open vs code ->terminal->check whether git command is exists or not .just type git on terminal

If git is not recoginised by vs code

->first of see whether git is installed on your pc or not if not installed then install git

->after installing the git set the path in environment variables

->then restart your system ,then again type git

```
PS C:\Users\allek\OneDrive\Desktop\registation>
PS C:\Users\allek\OneDrive\Desktop\registation> git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone      Clone a repository into a new directory
   init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add        Add file contents to the index
   mv         Move or rename a file, a directory, or a symlink
   restore    Restore working tree files
   rm         Remove files from the working tree and from the index
```
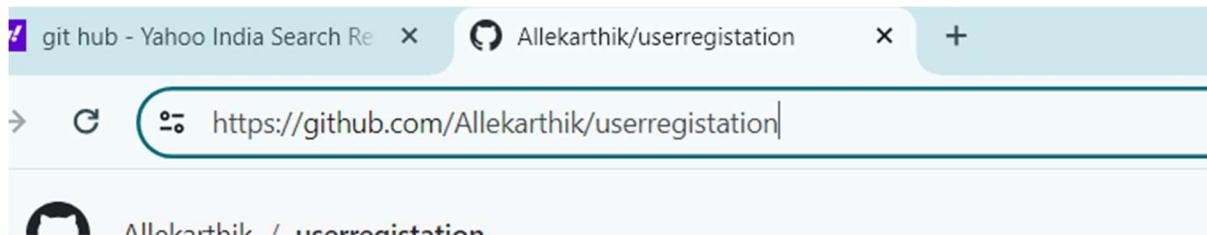
4:whatever the files that we created for simple user registration form add them to your git repository

5:after adding open terminal and type git clone https://github.com/Allekarthik/userregistation

Here after git clone it should be ur link

```
git hub - Yahoo India Search Re   X      ☉ Allekarthik/userregistation     X      +

→   C    ⊙⁚  https://github.com/Allekarthik/userregistation|

         Allokarthik / userregistation
```

-

6:then you can access all your files in github from vs code
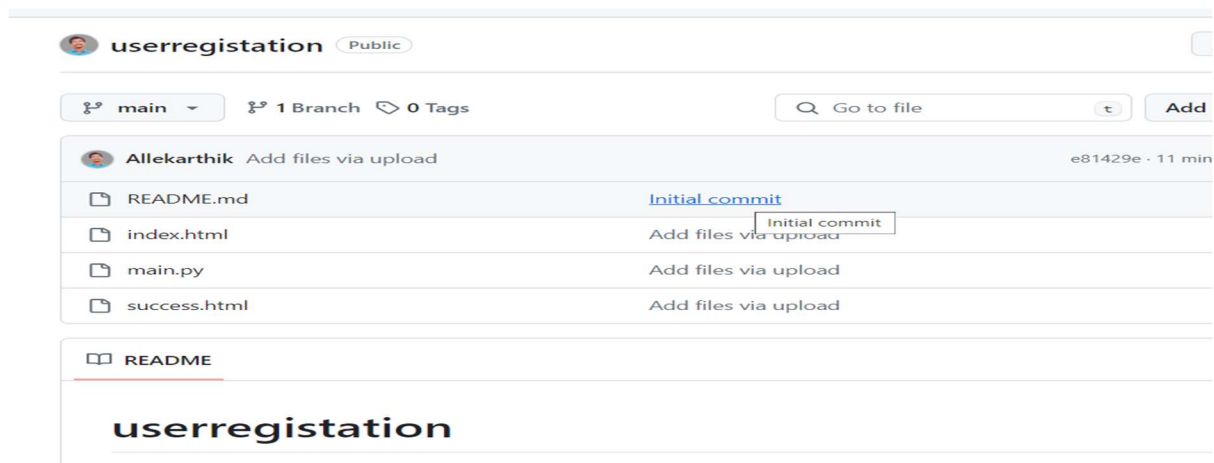
```
PS C:\Users\allek\OneDrive\Desktop\registation> git clone https://github.com/Allekarthik/userregistation
Cloning into 'userregistation'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (12/12), 4.25 KiB | 1.06 MiB/s, done.
Resolving deltas: 100% (2/2), done.
PS C:\Users\allek\OneDrive\Desktop\registation>
```
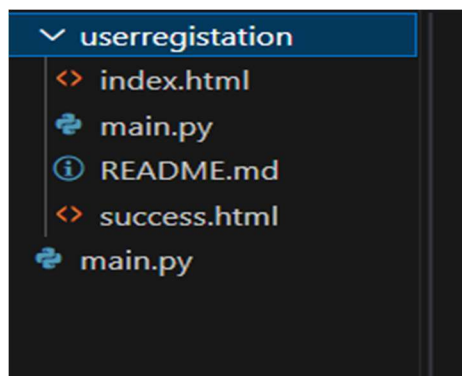

4. Jenkins installation and setup, explore the environment.

Install java jdk-21

Set environment varaiable for JDK

Download and install Jenkins

Run Jenkins on local host http://localhost:6969/

Username :admin

Password:5cfe93da2d2a444ebb1485b86c2c95ed

Note:it is different from user to user you have set up this after installation

6

Steps to run simple python program:

Go to dashboard->new item,then enter an item name choose free style project

Click on ok

Then open general->description(eg:this is my first program)

General

Description

this is my 2nd program

Advanced->use custom workspace->enter the path where your python file has saved eg: C:\Users\allek\Downloads\jenkins programs

Go to build steps ->execute windows batch command (you can get get this path from environment variables)

Build Steps

≡   Execute Windows batch command   ?   ×
Command
See the list of available environment variables

C:\Users\allek\AppData\Local\Programs\Python\Python311\python.exe main.py

Advanced ∨

📋   **main.py**   ×   +

File   Edit   View

print("hello world")|

↻   🖥   >   Downloads   >   jenkins programs

⌻   ⎘   🄰   ↗   🗑   ↑↓ Sort ∨   ≡ View ∨   •••

Name	Date modified	Type	Size
∨ Today			
📄 main	08-06-2024 10:34	Python Source File	1 KB

Then click on save

Go to build now,after this you can find build history open link(with date&time) and click on console output

Output will be displayed

Status

</> Changes

☐ Workspace

▷ Build Now

⚙ Configure

🗑 Delete Project

✎ Rename

☀ Build History   trend ∨

🔍 Filter...   /

⊘ #1
| Jun 8, 2024, 10:36 AM

🔊 Atom feed for all 🔊 Atom feed for failures

⊘ hello

this is my 2nd program

Permalinks

- Last build (#1), 8 min 20 sec ago
- Last stable build (#1), 8 min 20 sec ago
- Last successful build (#1), 8 min 20 sec ago
- Last completed build (#1), 8 min 20 sec ago

⊘ **Console Output**

```
Started by user admin
Running as SYSTEM
Building in workspace C:\Users\allek\Downloads\jenkins programs
[jenkins programs] $ cmd /c call C:\WINDOWS\TEMP\jenkins42318965722706176334.bat

C:\Users\allek\Downloads\jenkins
programs>C:\Users\allek\AppData\Local\Programs\Python\Python311\python.exe main.py
hello world

C:\Users\allek\Downloads\jenkins programs>exit 0
Finished: SUCCESS
```

5.Demonstrate continuous integration and development using Jenkins.(build pipeline)

Project url :github.com/Allekarthik/integration

Repository url :https://github.com/Allekarthik/integration.git

For integration of Jenkins with github first of all we need to create a repository in that place any file lets say eg:index.html

Then create new item ->general ->add any description

In github project add project url

Then in git add repository url

Then select main because my github is stored under main

Then select repository browser as githubweb

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)   ?
- ☐ Build after other projects are built   ?
- ☐ Build periodically   ?
- ☑ GitHub hook trigger for GITScm polling   ?
- ☐ Poll SCM   ?

Then tick the github hook trigger for GITScm polling

Then output will be displayed on the screen .

⊘ **Console Output**

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\github
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\github\.git # time
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/Allekarthik/integration.git # timeout=10
Fetching upstream changes from https://github.com/Allekarthik/integration.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.45.2.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/Allekarthik/integration.git +refs/
 > git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 5a35d63f38bf5118e5266923697fbbdd91653eb4 (refs/remotes/origin/main)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 5a35d63f38bf5118e5266923697fbbdd91653eb4 # timeout=10
Commit message: "Add files via upload"
 > git.exe rev-list --no-walk 5a35d63f38bf5118e5266923697fbbdd91653eb4 # timeout=10
Finished: SUCCESS
```

Steps to build pipeline:

Create 3 jobs like job1,job2,job3

We can create by new item->name->apply->save

Go to manage jeenkins->pulgins->available pulgins->then install build pipeline

Afer successful installation of pipeline

U can find "+" in main page click on it

Give any name eg:karthik - > select build pipeline view - > create
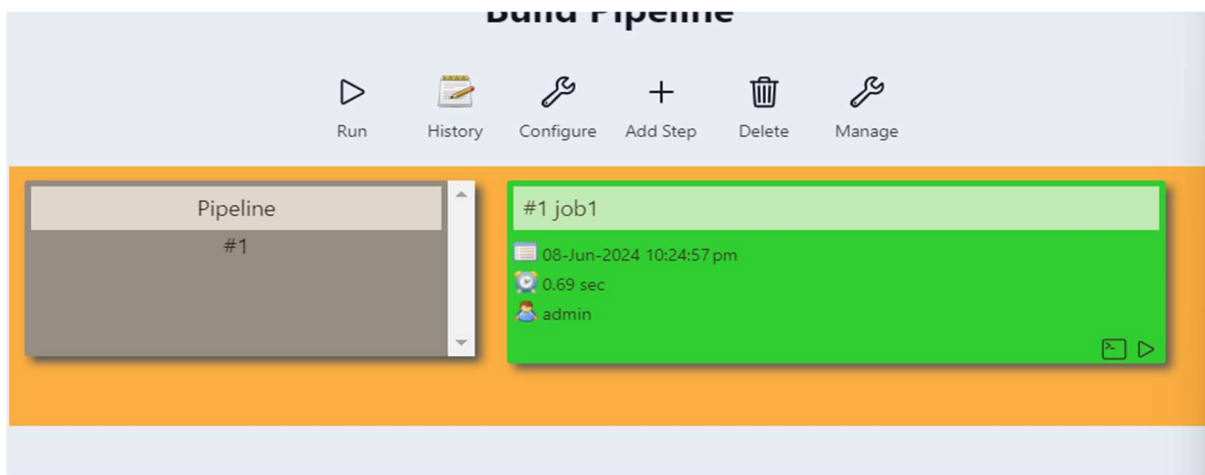
Then select the initial job eg: job1

Upstream / downstream config

Select Initial Job ?

job1

Then click on apply -> ok

Then click on run

6. Explore Docker commands for content management.

Docker is a powerful platform for developing, shipping, and running applications in containers. Content management within Docker involves managing images, containers, volumes, and networks. Here are some essential Docker commands for content management

Docker Commands for Content Management

1. Docker run

- **Description:** Runs a command in a new container. It's one of the most used Docker commands because it creates and starts a new container.
- **Syntax:** docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
- **Example:** $ docker run --name mycontainer -it ubuntu:16.04 /bin/bash
  - o **Explanation:**
    - --name mycontainer: Assigns the name "mycontainer" to the container.
    - -it: Combines -i (interactive) and -t (pseudo-TTY) options to keep the container running interactively.

2. Docker start

- **Description:** Starts one or more stopped containers. It does not create a new container but starts an existing one.
- **Syntax:** docker start [OPTIONS] CONTAINER [CONTAINER...]
- **Example:** $ docker start mycontainer
  - o **Explanation:** Starts the container named "mycontainer".

3. Docker stop

- **Description:** Stops one or more running containers. It sends a SIGTERM signal to the main process inside the container, allowing it to exit gracefully.
- **Syntax:** docker stop [OPTIONS] CONTAINER [CONTAINER...]
- **Example:** $ docker stop mycontainer
  - o **Explanation:** Stops the container named "mycontainer".

4. Docker rm

- **Description:** Removes one or more containers. The container must be stopped before it can be removed.
- **Syntax:** docker rm [OPTIONS] CONTAINER [CONTAINER...]
- **Example:** $ docker rm mycontainer
  - o **Explanation:** Removes the container named "mycontainer".

```
PS C:\Users\allek> docker start mycontainer
mycontainer
PS C:\Users\allek> docker stop mycontainer
mycontainer
PS C:\Users\allek> docker rm mycontainer
mycontainer
```

5. Docker ps

- **Description:** Lists containers. By default, it shows only running containers.
- **Syntax:** docker ps [OPTIONS]
- **Example:** $ docker ps
    - **Explanation:** Lists all currently running containers. To list all containers, including stopped ones, use docker ps -a.

6. Docker images

- **Description:** Lists images. It shows all the Docker images available on the local host.
- **Syntax:** docker images [OPTIONS] [REPOSITORY[:TAG]]
- **Example:** $ docker images
    - **Explanation:** Lists all images stored locally on the host.

```
PS C:\Users\allek> docker ps
CONTAINER ID   IMAGE                            COMMAND                CREATED        STATUS         PORTS                   NAMES
175c732d06ed   docker/welcome-to-docker:latest  "/docker-entrypoint…"  6 minutes ago  Up 6 minutes   0.0.0.0:8088->80/tcp    welcome-to-docker
PS C:\Users\allek> docker ps -a
CONTAINER ID   IMAGE                            COMMAND                CREATED         STATUS          PORTS                   NAMES
175c732d06ed   docker/welcome-to-docker:latest  "/docker-entrypoint…"  10 minutes ago  Up 10 minutes   0.0.0.0:8088->80/tcp    welcome-to-docker
PS C:\Users\allek> docker images
REPOSITORY                         TAG       IMAGE ID       CREATED       SIZE
docker/welcome-to-docker           latest    c1f619b6477e   7 months ago  18.6MB
karthikalle/welcome-to-docker      latest    c1f619b6477e   7 months ago  18.6MB
ubuntu                             16.04     b6f507652425   2 years ago   135MB
PS C:\Users\allek> docker pull ubuntu:16.04
16.04: Pulling from library/ubuntu
Digest: sha256:1f1a2d56de1d604801a9671f301190704c25d604a416f59e03c04f5c6ffee0d6
Status: Image is up to date for ubuntu:16.04
docker.io/library/ubuntu:16.04
```

7. Docker pull

- **Description:** Pulls an image or a repository from a registry. It downloads the image from a Docker registry like Docker Hub.
- **Syntax:** docker pull [OPTIONS] NAME[:TAG|@DIGEST]
- **Example:** $ docker pull ubuntu:16.04
    - **Explanation:** Pulls the Ubuntu 16.04 image from the Docker Hub registry.

8. Docker push

- **Description:** Pushes an image or a repository to a registry. It uploads the image to a Docker registry.
- **Syntax:** docker push [OPTIONS] NAME[:TAG]
- **Example:** $ docker push myimage
    - **Explanation:** Pushes the image named "myimage" to the Docker Hub registry.

```
PS C:\Users\allek> docker push karthikalle/welcome-to-docker
Using default tag: latest
The push refers to repository [docker.io/karthikalle/welcome-to-docker]
7f216224e911: Mounted from docker/welcome-to-docker
01e36c0e0b84: Mounted from docker/welcome-to-docker
901e6dddcc99: Mounted from docker/welcome-to-docker
f126bda54112: Mounted from docker/welcome-to-docker
38067ed663bf: Mounted from docker/welcome-to-docker
854101110f63: Mounted from docker/welcome-to-docker
81fdcc81a9d0: Mounted from docker/welcome-to-docker
cc2447e1835a: Mounted from docker/welcome-to-docker
latest: digest: sha256:2a6094f1c4b71cead4eb234b11f4a7e6bb5bc988b86e78017949abdab13a16b2 size: 1986
PS C:\Users\allek>
```

7. Develop a simple containerized application using Docker.

Here's an example of how you can develop a simple containerized application using Docker: Choose an application:

Before that create a folder in your file manager eg: 22507_Docker ->python_image ->Dockerfile ,app.py

Note:Docker should be opened first

• Choose a simple application that you want to containerize. For example, a Python script that prints "Hello World".

• Create a file named "Dockerfile" in the same directory as the application. In the Dockerfile, specify the base image, copy the application into the container, and specify the command to run the application.

Here's an example Dockerfile for a Python script:

Dockerfile

```
# myfirstprogram
FROM python
WORKDIR /app
COPY . /app
CMD ["python3", "app.py"]
```

app.py

```
print("hello world")
```

• Build the Docker image: Run the following command to build the Docker image: $ docker build -t myfirstprogram .

```
PS C:\Users\allek\Downloads\22507_Docker> cd .\python_image\
PS C:\Users\allek\Downloads\22507_Docker\python_image> build -t myfirstprogram .
```

```
PS C:\Users\allek\Downloads\22507_Docker\python_image> docker build -t myfirstprogram .
[+] Building 2.9s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
         0.0s

View build details: docker-desktop://dashboard/build/default/default/ff61p0a3ii050ryhl21q0qfz6

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

This command builds a new Docker image using the Dockerfile and tags the image with the name "myfirstprogram".

•Run the Docker container: Run the following command to start a new container based on the image: $ docker run – myfirstprogram

```
    View a summary of image vulnerabilities and recommendations → docker scout quickview
  PS C:\Users\allek\Downloads\22507_Docker\python_image> docker run myfirstprogram
  hello world
```

You can see our created container in docker apploication

	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions		
☐	objective_ritchie 3923cd77d844	myfirstprogram	Exited	N/A		2 hours ago	▶	⋮	🗑

You can check in Windows powershell also

```
PS C:\Users\allek> docker ps -a
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS                      P
ORTS                 NAMES
e6667136970f   myfirstprogram                 "python3 app.py"         7 minutes ago   Exited (0) 7 minutes ago
                     great_mayer
3923cd77d844   d895c1a6b484                   "python3 app.py"         2 hours ago     Exited (0) 2 hours ago
                     objective_ritchie
175c732d06ed   docker/welcome-to-docker:latest  "/docker-entrypoint.…"  3 hours ago     Exited (255) 8 minutes ago   0
.0.0.0:8088->80/tcp  welcome-to-docker
PS C:\Users\allek> |
```

This is a simple example of how you can use Docker to containerize an application. In a real-world scenario, you would likely have more complex requirements, such as running multiple containers, managing network connections, and persisting data. However, this example should give you a good starting point for using Docker to containerize your applications.

8. Install and Explore Selenium for automated testing or Write a simple program in JavaScript and perform testing using Selenium.

Prerequired:

    Download and install Node.js

     Download and install vs code

Install selenium-webdriver and install mocha

https://storage.googleapis.com/chrome-for-testing-public/125.0.6422.141/win64/chrome-win64.zip

->for installing selenium web driver go to any web browser->selenium web driver install ->Download selenium->javascript stable 4.21->chromedriver.exe->win64

Steps:

Create a folder called newfolder in your downloads /or any other main folder

Open this folder in vs code and go to termina->new terminal then type the below command i.e npm init where it atomatically creates package.json file

```
{
  "name": "new-folder",
```

```json
"version": "1.0.0",
"main": "index.js",
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"author": "",
"license": "ISC",
"description": "",
"dependencies": {
  "selenium-webdriver": "^4.21.0"
},
"devDependencies": {
  "chromedriver": "^125.0.3",
  "geckodriver": "^4.4.1"
},
"mocha":"^10.2.0" //here u need to add this extra line
}
```

```
PS C:\Users\allek\Downloads\22507_selenium> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.
```

```
Press ^C at any time to quit.
package name: (22507_selenium)
version: (1.0.0)
description:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\allek\Downloads\22507_selenium\package.json:

{
  "name": "22507_selenium",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "selenium-webdriver": "^4.21.0"
  },
  "mocha": "^10.2.0",
  "devDependencies": {
    "chromedriver": "^125.0.3",
    "geckodriver": "^4.4.1"
  },
  "description": ""
}

Is this OK? (yes) yes
```

Then next step is to install selenium

After installing it will create an package-lock.json file

```
PS C:\Users\allek\Downloads\22507_selenium> npm install selenium-webdriver

up to date, audited 109 packages in 2s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\allek\Downloads\22507_selenium>
```

After succesfull installation of selenium we need to install chromedriver

```
PS C:\Users\allek\Downloads\22507_selenium> npm install chromedriver geckodriver --save-dev

added 92 packages, and audited 109 packages in 17s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\allek\Downloads\22507_selenium> node test.js
node:internal/modules/cjs/loader:1148
  throw err;
  ^

Error: Cannot find module 'C:\Users\allek\Downloads\22507_selenium\test.js'
    at Module._resolveFilename (node:internal/modules/cjs/loader:1145:15)
    at Module._load (node:internal/modules/cjs/loader:986:27)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:174:12)
    at node:internal/main/run_main_module:28:49 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}
```

Home.js

```
const { Builder, By, Key, until } = require('selenium-webdriver');

const chrome = require('selenium-webdriver/chrome');


(async () => {
    const driver = await new Builder()
        .forBrowser('chrome')
        .setChromeOptions(new chrome.Options())
        .build();


    try {
        await driver.get('https://www.google.com');
        await driver.findElement(By.name('q')).sendKeys('Selenium', Key.RETURN);
        await driver.wait(until.titleContains('Selenium'), 100000000000);
    } catch (error) {
        console.error('Test failed:', error);
    } finally {
        await driver.quit();
    }
```

})();

```
PS C:\Users\allek\Downloads\22507_selenium> node home.js
[17580:21120:0607/214821.992:ERROR:sandbox_win.cc(910)] Sandbox cannot access executable. Check filesystem permissions are valid. See https://bit.ly/31yqMJR.: A
ccess is denied. (0x5)

DevTools listening on ws://127.0.0.1:52041/devtools/browser/40b92a58-5a07-42df-b061-0f2037857e59
[17580:27540:0607/214822.274:ERROR:network_service_instance_impl.cc(600)] Network service crashed, restarting service.
PS C:\Users\allek\Downloads\22507_selenium>
```

Output:
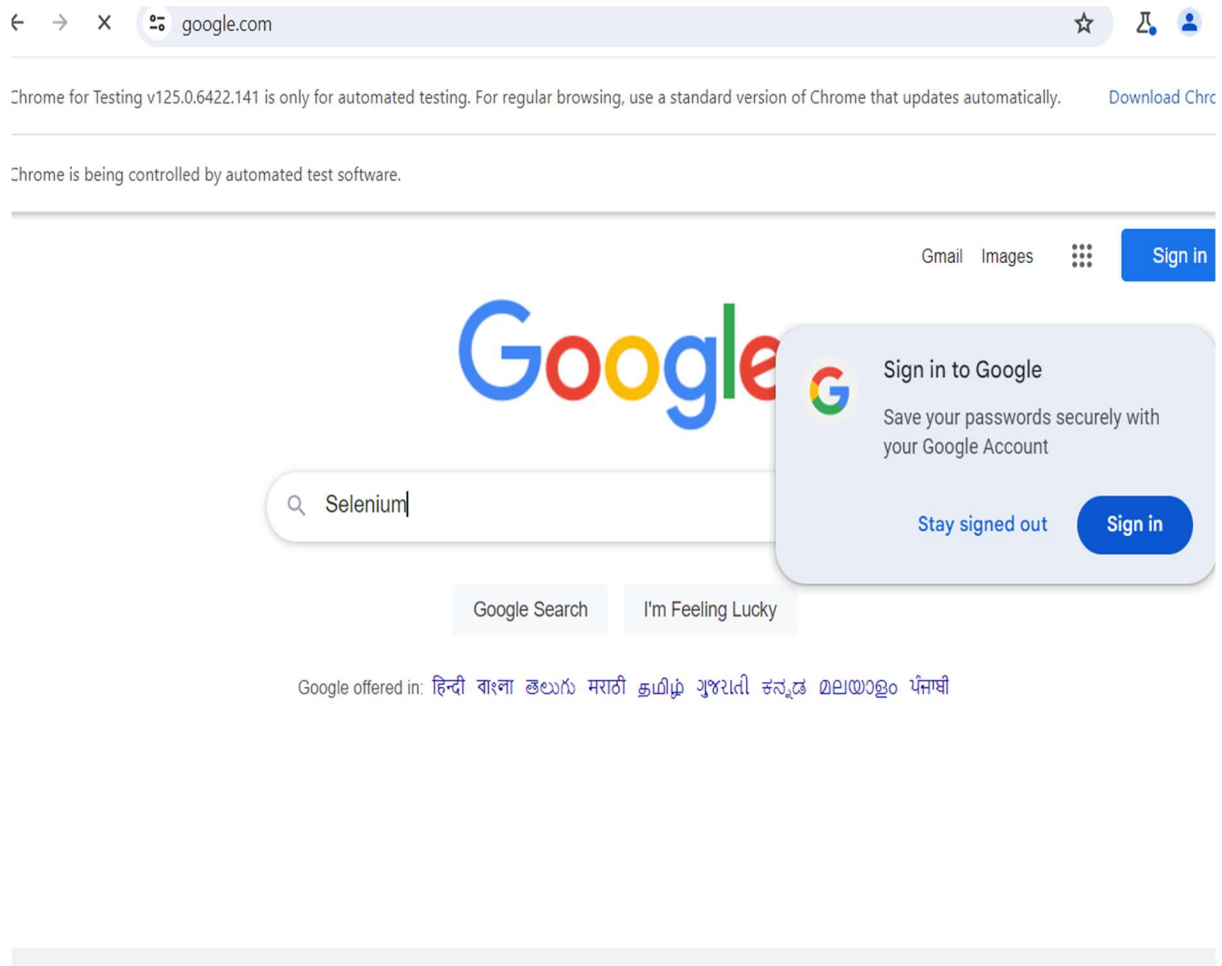
9. Develop test cases for the above containerized application using selenium.

```javascript
const { Builder, By, Key, until } = require('selenium-webdriver');
const chrome = require('selenium-webdriver/chrome');

(async () => {
  const driver = await new Builder()
    .forBrowser('chrome')
    .setChromeOptions(new chrome.Options())
    .build();

  try {
    // Test Case 1: Navigate to Google and verify title
    await driver.get('https://www.google.com');
    await driver.wait(until.titleContains('Google'), 10000);
    console.log('Test Case 1 Passed: Title contains "Google"');

    // Test Case 2: Search for "Selenium" on Google
    await driver.findElement(By.name('q')).sendKeys('Selenium', Key.RETURN);
    await driver.wait(until.titleContains('Selenium'), 10000);
    console.log('Test Case 2 Passed: Title contains "Selenium"');

    // Test Case 3: Verify search results
    const searchResults = await driver.findElements(By.css('div.g'));
    console.log(`Test Case 3 Passed: Found ${searchResults.length} search results`);
```

```
    // Test Case 4: Verify the presence of the search input box

    const searchInput = await driver.findElement(By.name('q'));

    const isSearchInputDisplayed = await searchInput.isDisplayed();

    console.log(`Test Case 4 Passed: Search input box is displayed:
${isSearchInputDisplayed}`);


  } catch (error) {

    console.error('One or more test cases failed:', error);

  } finally {

    await driver.quit();

  }
})();
```

```
PS C:\Users\allek\Downloads\New folder> node app.js
[11328:20764:0608/014027.409:ERROR:sandbox_win.cc(910)] Sandbox cannot access executable. Check filesystem permissions are valid. See https://bit.ly/31yqMJR.: A
ccess is denied. (0x5)


DevTools listening on ws://127.0.0.1:57147/devtools/browser/a7157ec8-2ac0-455a-ba9c-672c5201cdb4
[11328:22088:0608/014027.730:ERROR:network_service_instance_impl.cc(600)] Network service crashed, restarting service.
Test Case 1 Passed: Title contains "Google"
Test Case 2 Passed: Title contains "Selenium"
Test Case 3 Passed: Found 12 search results
Test Case 4 Passed: Search input box is displayed: true
PS C:\Users\allek\Downloads\New folder>
```