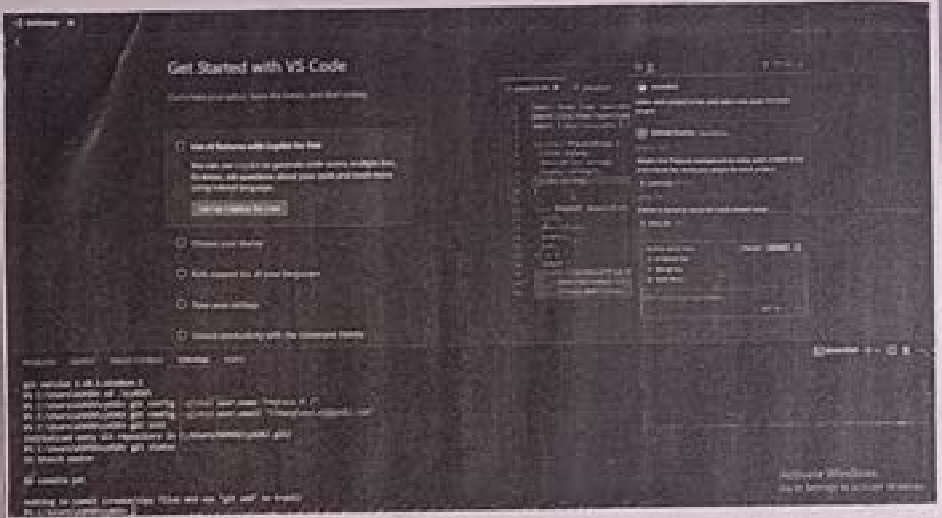


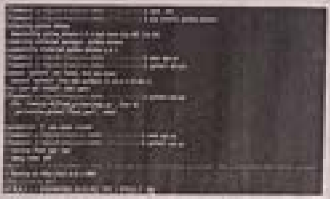
15/02/2025

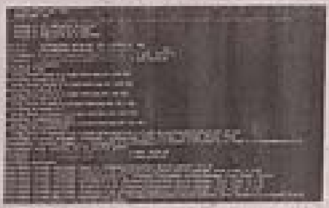
1 Explore basic git and github commands

Git is used for version control

* git --version
  - This will tell the version if its installed
* mkdr cy21
* cd cy21
  - make/create a directory in terminal and enter into it
* git config --global user.name "name"
* git config --global user.email "email"
  - These set the username and email
* git init
  - To initialize the repository within the folder
* git status
  - This specifies in which branch we are and commits
* Create a file with some content, save and execute git status again
* git add filename
* - This will add file
* git commit -m "comment"
  - To save this file/track file in our repository
* Next add other line of content in file and save it
* Retain change in file execute all above commands once again
* If we check git status it will be in green colour
* git commit it.

- git branch branch-name
  - To create a new branch
- git checkout branch-name
  - To enter a branch
- git checkout master
  - To enter master branch
- git fetch
  - To merge master and other branch
- git merge branch_name
- git log
  - To know commit history
  - We get new commit hash value
- git branch -d branch-name
  - To delete branch
- git checkout commit hash
  - To goto first version of our file
- create new repo in github, create file
- clone this on local repo
- git clone <<http link>>
- To push local repo to github acc
  - git remote add origin <http-->
  - git push -u origin master





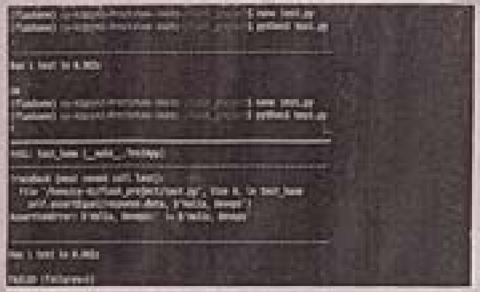
2. Implement, code, build, test, configure and monitor the software app in DevOps using Flask

```
i) sudo apt update
ii) python3 --version
iii) sudo apt install python3-pip -y
iv) pip --version
v) pip install flask
vi) sudo apt install python3-venv -y
vii) mkdir flask-project
viii) cd flask-project
ix) python3 -m venv flaskenv
x) source flaskenv/bin/activate
xi) flask --version
xii) nano app.py
```

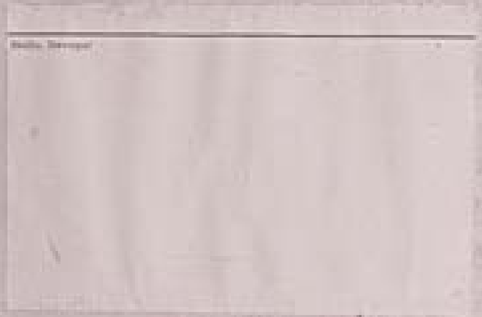
Implementation:

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def home():
    return "Hello, DevOps!"

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000)
```

```
xiii) python3 app.py
```



```
[illegible]
[illegible]

Ran 1 test in 0.001s

OK
[illegible]
[illegible]

FAIL: test_home (__main__.TestApp)

Traceback (most recent call last):
  [illegible]
  [illegible]
AssertionError: [illegible]

Ran 1 test in 0.001s

FAILED (failures=1)
```



Hello, Devops!

D D M M Y Y Y Y

* Build:
- shell script build.sh

```
#!/bin/bash
echo "Setting up environment"
python3 -m venv venv
source ./venv/bin/activate
pip install flask
echo "Environment setup complete. Run the application:
            python3 app.py"
```

run bash build.sh

* Tests
- create test.py file

```
import unittest
from app import app
class TestApp(unittest.TestCase):
    def test_home(self):
        tester = app.test_client()
        response = tester.get('/')
        self.assertEqual(response.status_code, 200)
        self.assertEqual(response.data, b"Hello DevOps!")

if __name__ == "__main__":
    unittest.main()
```

* Configure:

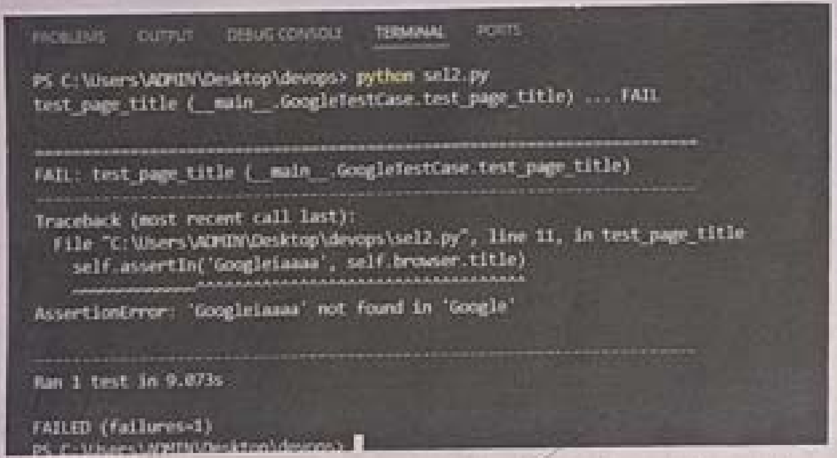
```
create .env file
nano .env and update
flask-port = 5000
pip install python-dotenv
```

- Change in app.py only

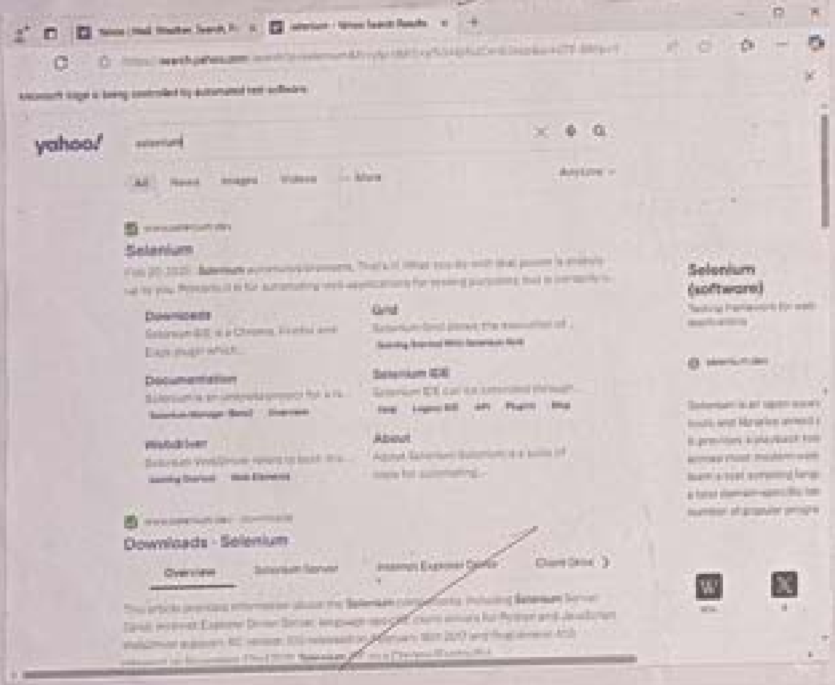
```
import os
from flask import Flask
app = Flask(__name__)
@app.route('/')
def home():
    return "Hello Devops!"

if __name__ == "__main__":
    port = int (os.getenv('flask-port', 5000))
    app.run (port=port)
```

* Monitor:

```
@app.route('/health')
def health():
    return {"status": "up"}, 200
```



```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\ADMIN\Desktop\devops> python sel2.py
test_page_title (__main__.GoogleTestCase.test_page_title) ... FAIL

======================================================================
FAIL: test_page_title (__main__.GoogleTestCase.test_page_title)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\ADMIN\Desktop\devops\sel2.py", line 11, in test_page_title
    self.assertIn('Googleiaaaa', self.browser.title)
AssertionError: 'Googleiaaaa' not found in 'Google'

----------------------------------------------------------------------
Ran 1 test in 9.073s

FAILED (failures=1)
PS C:\Users\ADMIN\Desktop\devops>
```

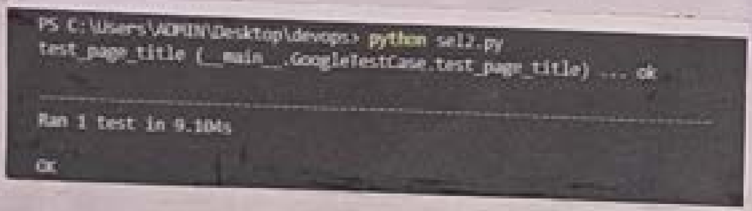


13 03 2025

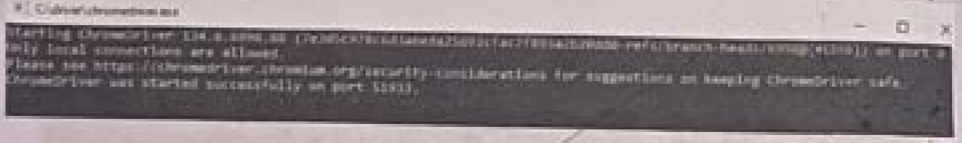
3 Install and Explore selenium for automated testing

- Selenium: Open source for automated web browser

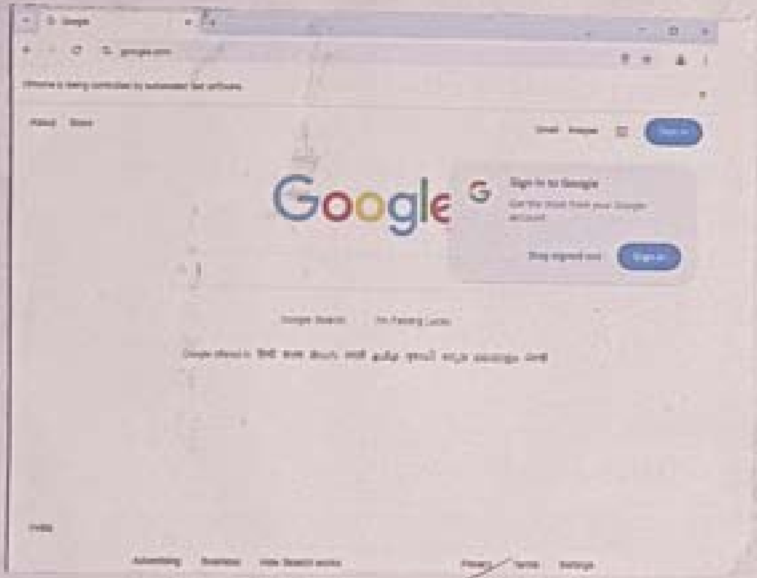
```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

browser = webdriver.Firefox()
browser.get('http://www.yahoo.com')
assert 'yahoo' in browser.title
elem = browser.find_element(By.NAME, 'p')
elem.send_keys('selenium' + Keys.RETURN)
time.sleep(10)
browser.quit()


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

chrome_driver_path = "c:\driver\chromedriver.exe"
service = Service(executable_path = chrome_driver_path)
```



```
PS C:\Users\ADMIN\Desktop\devops> python sel2.py
test_page_title (__main__.GoogleTestCase.test_page_title) ... ok

----------------------------------------------------------------------
Ran 1 test in 9.104s

OK
```



```
C:\driver\chromedriver.exe
Starting ChromeDriver 124.0.[illegible] ([illegible]) on port 0
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully on port [illegible].
```



D.S.C.E.

DD MM YYYY

```
driver = webdriver.chrome (service = service)
driver.get ("https://www.google.com")

assert "Google" in driver.title

time.sleep(5)
driver.quit()
```

* Unittest Framework

```
import unittest
from selenium import webdriver

class GoogleTestCase(unittest.TestCase):
    def setUp(self):
        self.browser = webdriver.Firefox()
        self.addCleanup(self.browser.quit)
    def test_page_title(self):
        self.browser.get("http://www.google.com")
        self.assertIn('Google', self.browser.title)

if __name__ == "__main__":
    unittest.main(verbosity=2)
```

* NOTE:
- Install chromedriver 64 bit
- extract the file and copy the chromedriver.exe file
- Paste it in the C drive with new folder driver then run it

[illegible]

D.S.C.E.

```
[illegible]
```

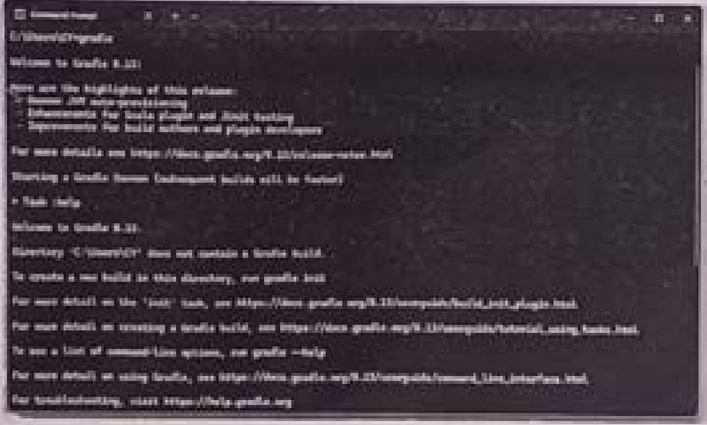
4. Setting up a gradle project understanding build scripts (Groovy), dependency management and task automation:

- Gradle:

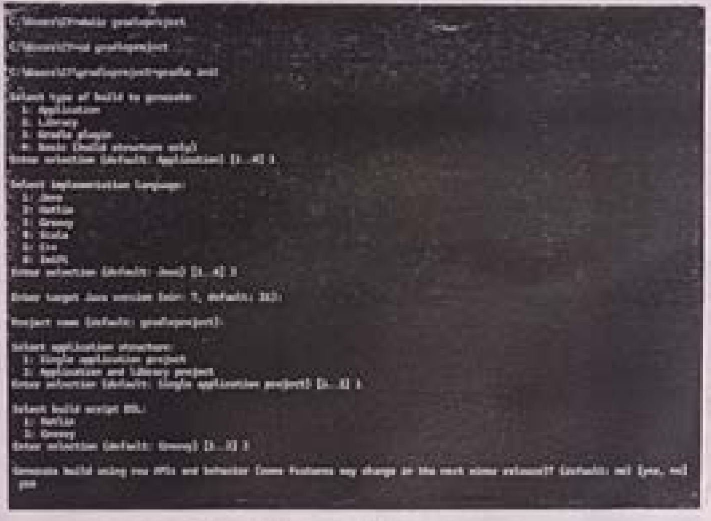
Gradle is an open source build automation tool used for java projects, supporting Groovy and Kotlin. It automates tasks accelerating the agile process like compiling code, running tests, and creating JAR files. Gradle's incremental builds reduce build time by compiling only changed code. It offers efficient dependency management and supports multi-project builds. With Groovy or Kotlin DSL, it provides flexibility for customization. Gradle is widely used in Continuous Integration (CI) pipelines for automated testing and deployment. Its build cache enhances performance by reusing previous outputs, making it a preferred choice for large-scale applications.

- Install Extensions: Gradle
  Extension pack for java.

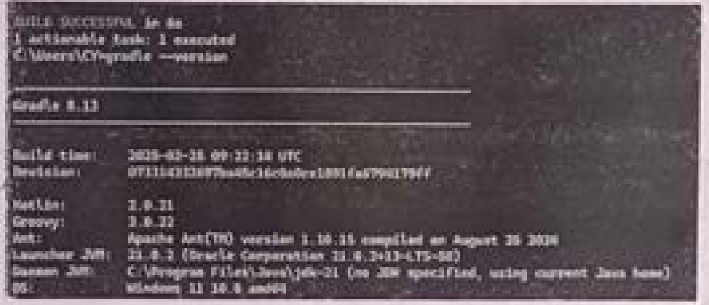
- First way to install and use:
  Includes directly loading the required libraries and selecting the language as Groovy.



```
C:\Users\CY>gradle

Welcome to Gradle 8.13!

Here are the highlights of this release:
 - Daemon JVM auto-provisioning
 - Enhancements for Scala plugin and JUnit testing
 - Improvements for build authors and plugin developers

For more details see https://docs.gradle.org/8.13/release-notes.html

Starting a Gradle Daemon (subsequent builds will be faster)

> Task :help

Welcome to Gradle 8.13.

Directory 'C:\Users\CY' does not contain a Gradle build.

To create a new build in this directory, run gradle init

For more detail on the 'init' task, see https://docs.gradle.org/8.13/userguide/build_init_plugin.html

For more detail on creating a Gradle build, see https://docs.gradle.org/8.13/userguide/tutorial_using_tasks.html

To see a list of command-line options, run gradle --help

For more detail on using Gradle, see https://docs.gradle.org/8.13/userguide/command_line_interface.html

For troubleshooting, visit https://help.gradle.org
```



```
C:\Users\CY>mkdir gradleproject

C:\Users\CY>cd gradleproject

C:\Users\CY\gradleproject>gradle init

Select type of build to generate:
  1: Application
  2: Library
  3: Gradle plugin
  4: Basic (build structure only)
Enter selection (default: Application) [1..4] 1

Select implementation language:
  1: Java
  2: Kotlin
  3: Groovy
  4: Scala
  5: C++
  6: Swift
Enter selection (default: Java) [1..6] 1

Enter target Java version (min: 7, default: 21):

Project name (default: gradleproject):

Select application structure:
  1: Single application project
  2: Application and library project
Enter selection (default: Single application project) [1..2] 1

Select build script DSL:
  1: Kotlin
  2: Groovy
Enter selection (default: Groovy) [1..2] 2

Generate build using new APIs and behavior (some features may change in the next minor release)? (default: no) [yes, no]
 yes
```



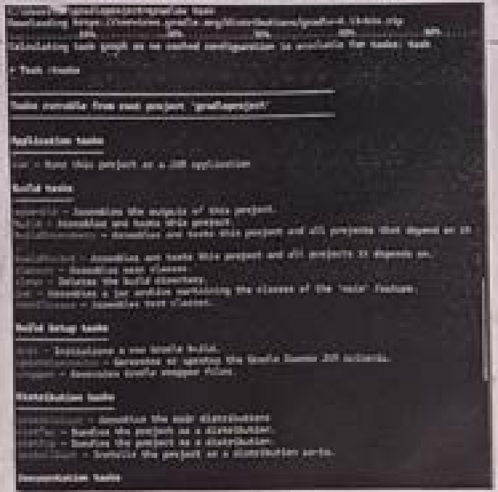
```
BUILD SUCCESSFUL in 4s
1 actionable task: 1 executed
C:\Users\CY>gradle --version

------------------------------------------------------------
Gradle 8.13
------------------------------------------------------------

Build time:    2025-02-25 09:22:14 UTC
Revision:      [illegible]

Kotlin:        2.0.21
Groovy:        3.0.22
Ant:           Apache Ant(TM) version 1.10.15 compiled on August 25 2024
Launcher JVM:  21.0.2 (Oracle Corporation 21.0.2+13-LTS-58)
Daemon JVM:    C:\Program Files\Java\jdk-21 (no JDK specified, using current Java home)
OS:            Windows 11 10.0 amd64
```

D D M M Y Y Y Y

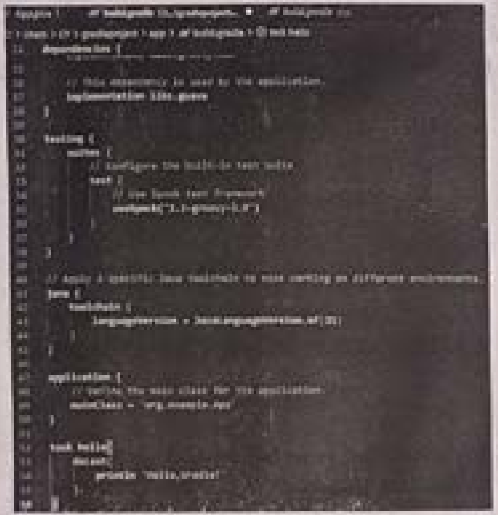
Program:

```
package gradleproj;
import java.awt.Desktop;
import java.net.URL;

public class App {
    public static void main(String[] args) {
        try {
            String url = "https://www.google.com";
            if(Desktop.isDesktopSupported()) {
            {   Desktop desktop = Desktop.getDesktop();
                desktop.browse(new URI(url));
            }
            else{
                System.out.println("Desktop is not supported");
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
```

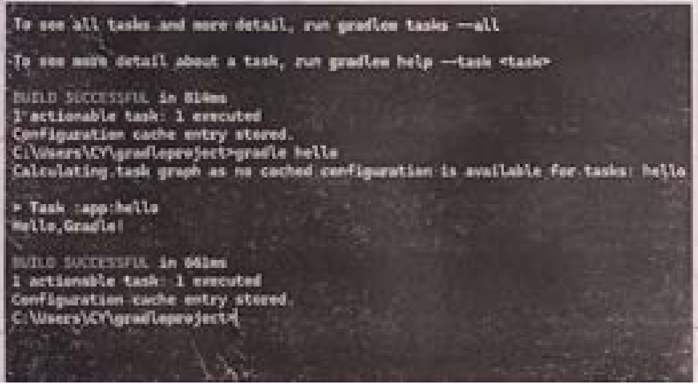
* Second way to install Gradle:
1) Install gradle from internet
2) Extract file and copy it to program files
3) Open bin folder and copy the path
4) Set the path in environmental variables



```
C:\Users\CY\gradleproject>gradle task
Downloading https://services.gradle.org/distributions/gradle-[illegible]-bin.zip
.............10%.............20%.............30%.............40%.............50%
Calculating task graph as no cached configuration is available for tasks: task

> Task :tasks

------------------------------------------------------------
Tasks runnable from root project 'gradleproject'
------------------------------------------------------------

Application tasks
-----------------
run - Runs this project as a JVM application

Build tasks
-----------
assemble - Assembles the outputs of this project.
build - Assembles and tests this project.
buildDependents - Assembles and tests this project and all projects that depend on it.

buildNeeded - Assembles and tests this project and all projects it depends on.
classes - Assembles main classes.
clean - Deletes the build directory.
jar - Assembles a jar archive containing the classes of the 'main' feature.
testClasses - Assembles test classes.

Build Setup tasks
-----------------
init - Initializes a new Gradle build.
updateDaemonJvm - Generates or updates the Gradle Daemon JVM criteria.
wrapper - Generates Gradle wrapper files.

Distribution tasks
------------------
assembleDist - Assembles the main distributions
distTar - Bundles the project as a distribution.
distZip - Bundles the project as a distribution.
installDist - Installs the project as a distribution as-is.

Documentation tasks
```



```
{ } Welcome        build.gradle [illegible] ●        build.gradle [illegible]
C: > Users > CY > gradleproject > app > build.gradle > [illegible]
dependencies {
    [illegible]

    // This dependency is used by the application.
    implementation libs.guava
}

testing {
    suites {
        // Configure the built-in test suite
        test {
            // Use Spock test framework
            useSpock('2.3-groovy-3.0')
        }
    }
}

// Apply a specific Java toolchain to ease working on different environments.
java {
    toolchain {
        languageVersion = JavaLanguageVersion.of(21)
    }
}

application {
    // Define the main class for the application.
    mainClass = 'org.example.App'
}

task hello{
    doLast{
        println 'Hello,Gradle'
    }
}
```



```
To see all tasks and more detail, run gradlew tasks --all

To see more detail about a task, run gradlew help --task <task>

BUILD SUCCESSFUL in 814ms
1 actionable task: 1 executed
Configuration cache entry stored.
C:\Users\CY\gradleproject>gradle hello
Calculating task graph as no cached configuration is available for tasks: hello

> Task :app:hello
Hello,Gradle!

BUILD SUCCESSFUL in 664ms
1 actionable task: 1 executed
Configuration cache entry stored.
C:\Users\CY\gradleproject>
```

* Commands to be executed in Gradle:

- gradle
- mkdir gradleproject
- cd gradleproject
- gradle init
- select type of build to generate: 1: Application
- Select implementation language: Groovy
- Enter selection 3
- select application structure: 1: single application project
- select build script DSL: 2: Groovy

→ BUILD SUCCESSFUL

→ gradle task

→ BUILD successful

- Once the gradle project is created
- navigate into the folder
- Open the folder in vscode
- Open the build gradle file
- Make the following changes / additions

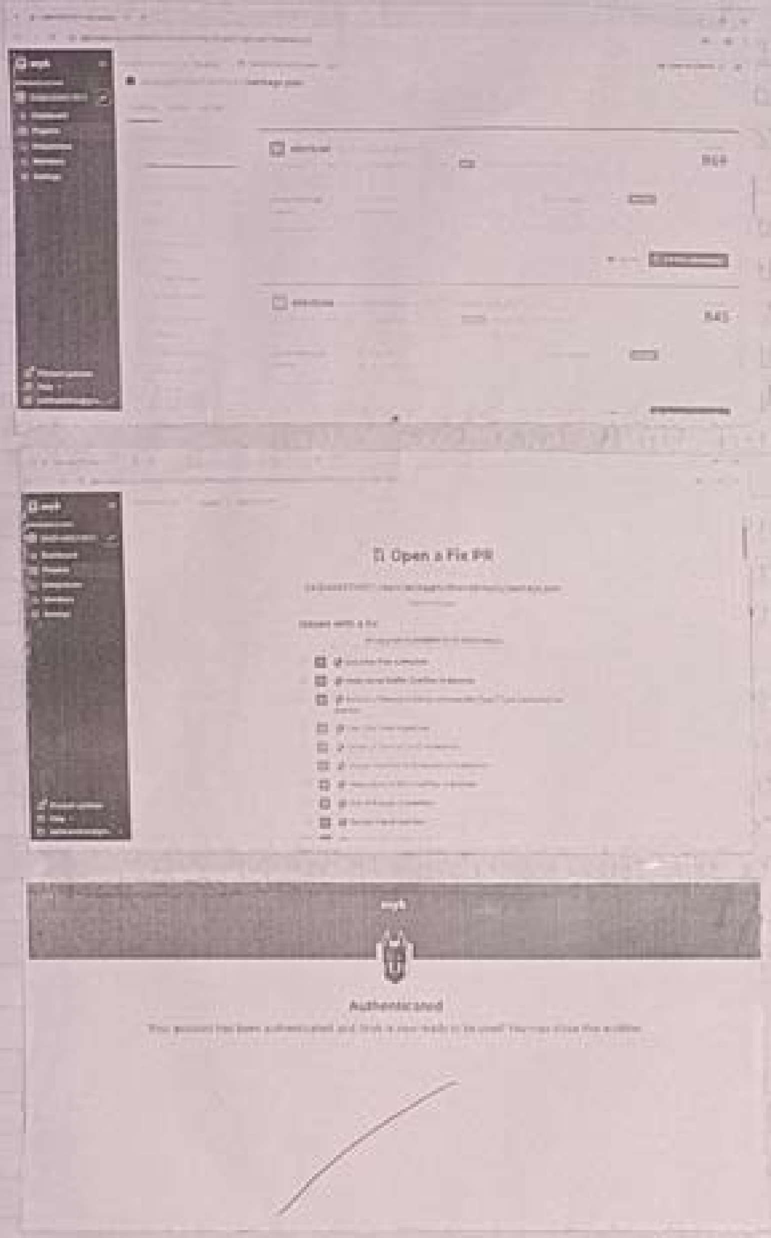
```
task hello{
    dolast{
        println "Hello, Gradle!" }
    }
}
```

Run the file and see the output

Output shows

Hello, Gradle!

[illegible]



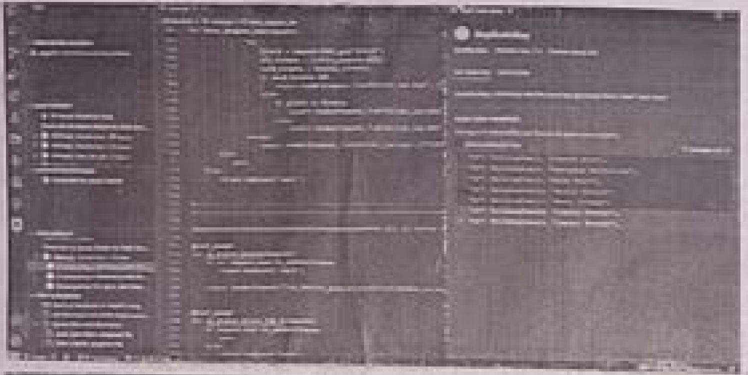
05 04 2025

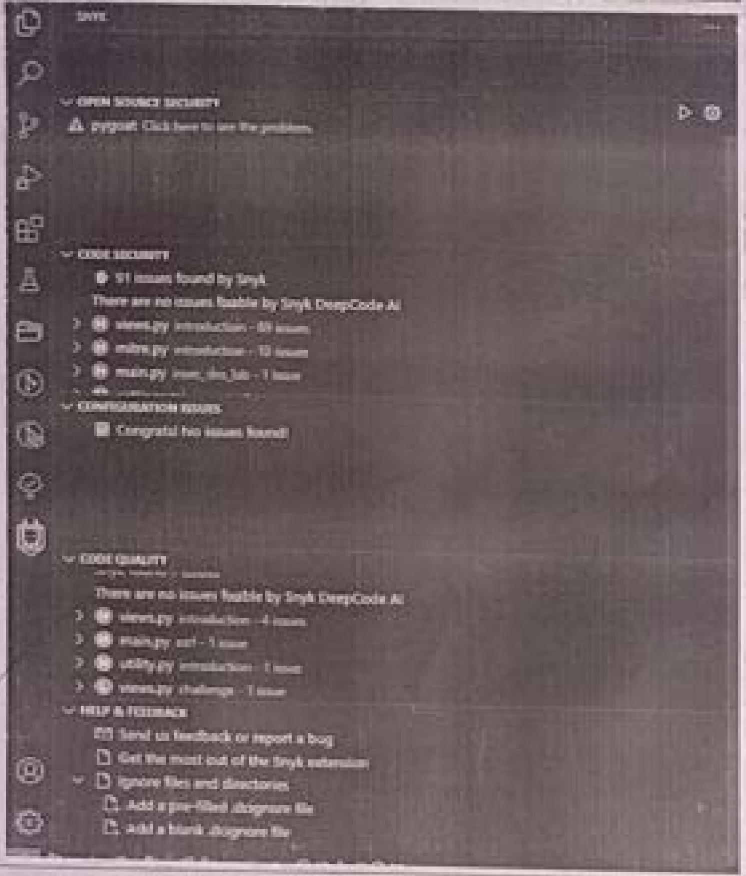
Experiment 5: Implementing continuous security with snyk

Snyk is a security tool used to find and fix vulnerabilities in code, dependencies, container image and infrastruct-ure as a code.
Snyk scans multiple content types for security issues:
Snyk Open source: find and automatically fix open-source vulnerabilities

1) Using Snyk website:
- go to snyk.io
- login with your Github
- then authorize snyk
- Go for your projects and you can check your project vulnerability under four sections

C	H	M	L
critical	High	Moderate	low

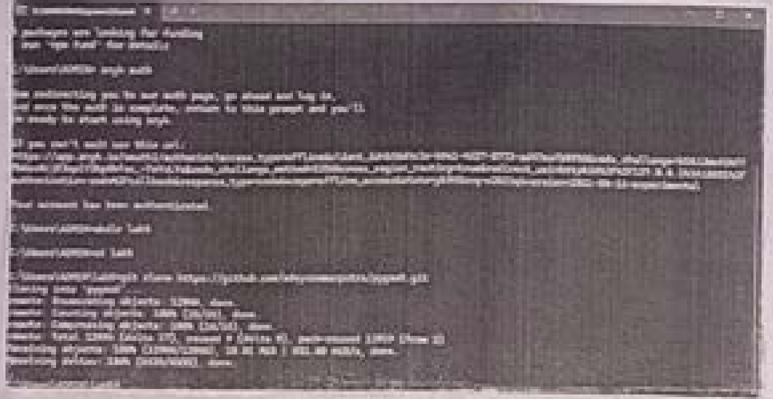




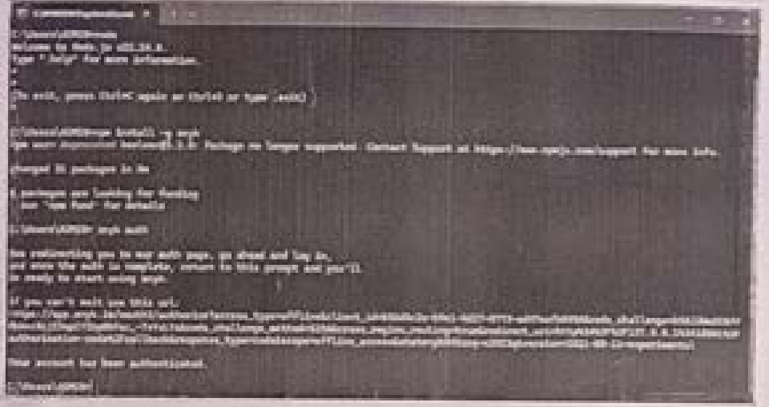
```
SNYK

∨ OPEN SOURCE SECURITY
  ⚠ pygoat Click here to see the problem.

∨ CODE SECURITY
    91 issues found by Snyk
    There are no issues fixable by Snyk DeepCode AI
  >  views.py introduction - 80 issues
  >  mitre.py introduction - 10 issues
  >  main.py insec_des_lab - 1 issue
∨ CONFIGURATION ISSUES
    Congrats! No issues found!

∨ CODE QUALITY
    There are no issues fixable by Snyk DeepCode AI
  >  views.py introduction - 4 issues
  >  main.py ssrf - 1 issue
  >  utility.py introduction - 1 issue
  >  views.py challenge - 1 issue
∨ HELP & FEEDBACK
    Send us feedback or report a bug
    Get the most out of the Snyk extension
  ∨ Ignore files and directories
      Add a pre-filled .dcignore file
      Add a blank .dcignore file
```

8) <u>Using Vscode:</u>

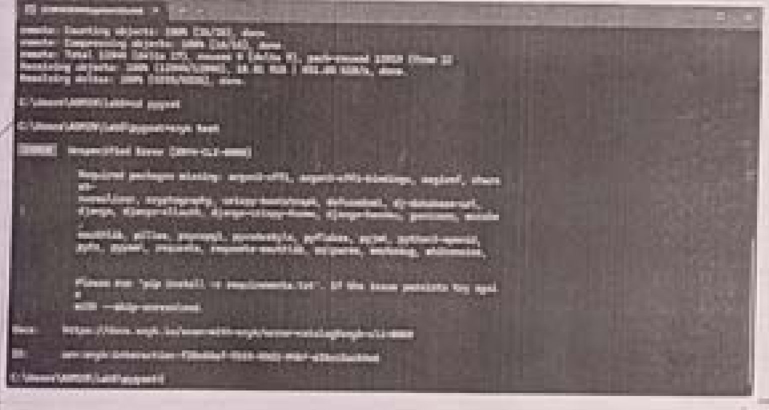
- Search for pygoat in chrome and copy the code of github and go to cmd, type:
    git clone <code>
- Then open the cloned folder in Vs code
- In Vs code go to extensions and install snyk security
- left panel you get to know vulnerabilities



```
[illegible] packages are looking for funding
  run `npm fund` for details

C:\Users\ADMIN>snyk auth

Now redirecting you to our auth page, go ahead and log in,
and once the auth is complete, return to this prompt and you'll
be ready to start using snyk.

If you can't wait use this url:
[illegible]

Your account has been authenticated.

C:\Users\ADMIN>[illegible]

C:\Users\ADMIN>cd lab

C:\Users\ADMIN\lab>git clone [illegible]
Cloning into '[illegible]'...
remote: Enumerating objects: [illegible], done.
remote: Counting objects: 100% [illegible], done.
remote: Compressing objects: 100% [illegible], done.
remote: Total [illegible]
Receiving objects: 100% [illegible], done.
Resolving deltas: 100% [illegible], done.
```



```
C:\Users\ADMIN>node
Welcome to Node.js [illegible].
Type ".help" for more information.
>
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>

C:\Users\ADMIN>npm install -g snyk
npm warn deprecated [illegible]: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.

changed [illegible] packages in [illegible]

[illegible] packages are looking for funding
  run `npm fund` for details

C:\Users\ADMIN>snyk auth

Now redirecting you to our auth page, go ahead and log in,
and once the auth is complete, return to this prompt and you'll
be ready to start using snyk.

If you can't wait use this url:
[illegible]

Your account has been authenticated.

C:\Users\ADMIN>
```

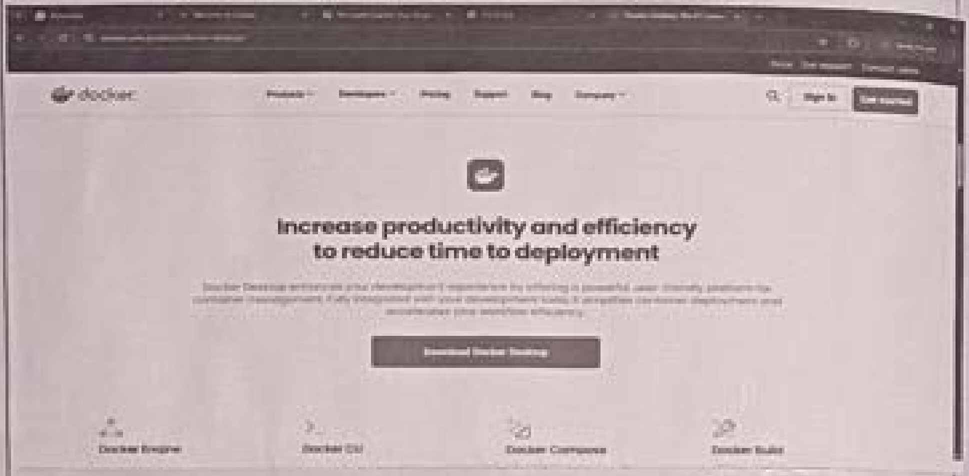


```
remote: Counting objects: 100% [illegible], done.
remote: Compressing objects: 100% [illegible], done.
remote: Total [illegible]
Receiving objects: 100% [illegible], done.
Resolving deltas: 100% [illegible], done.

C:\Users\ADMIN\lab>cd [illegible]

C:\Users\ADMIN\lab\[illegible]>snyk test

[illegible]   Unspecified Error (SNYK-CLI-0000)

        Required packages missing: [illegible]
        [illegible]

        Please run `pip install -r requirements.txt`. If the issue persists try again
        with --skip-unresolved.

Docs:   [illegible]

ID:     [illegible]

C:\Users\ADMIN\lab\[illegible]>
```

(ii) Using cmd to download snyk:

- Install node js .msi and open it
- go to cmd and type node
- open install -g snyk
- snyk auth
- give grant app access
- snyk test
- snyk monitor

[illegible]



Increase productivity and efficiency to reduce time to deployment

Docker Desktop enhances your development experience by offering a powerful, user-friendly platform for container management. Fully integrated with your development tools, it simplifies container deployment and accelerates your workflow efficiency.

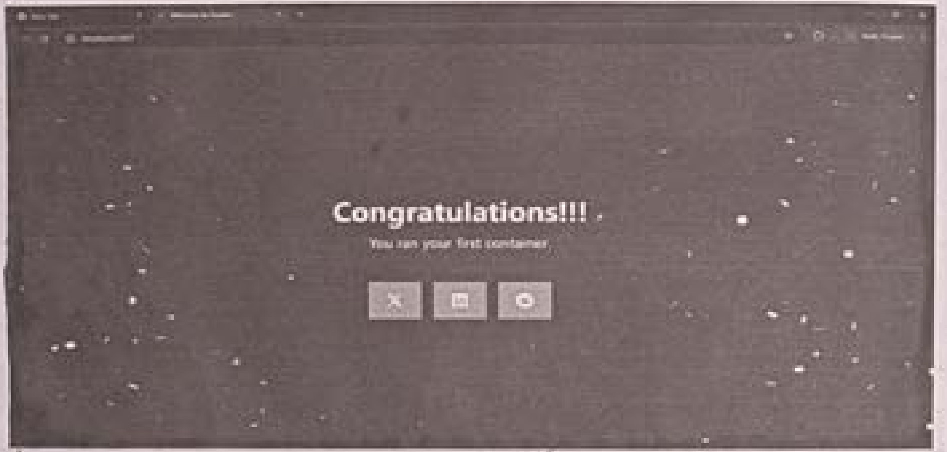
17 04 2025

Experiment 6: Develop a simple containarized application using docker

- Docker Desktop is an open source application that makes it easy to build, share and run containarized applications on your computer using Docker. It provides a user friendly interface and tools for developers to work with docker containers on Windows and macos
- It's a tool that is used to package your code, dependencies and runtime environment into a single container
- Docker Desktop Includes:

* Docker Engine: It's the core component that runs containers
* Docker CLI: Command-line tool to interact with Docker
* Docker Compose: Tool to define and run multi container applications
* GUI Dashboard: Visual Interface to manage containers, images, volumes, etc

- It's commonly used for local development and testing before deploying apps to cloud or production environments.



Congratulations!!!

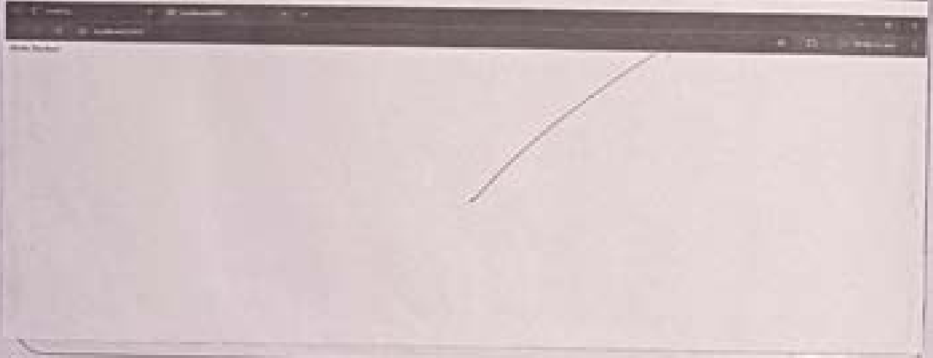
You ran your first container

D D M M Y Y Y Y

2) Using Application

- Go to the browser and install docker-desktop
- In command prompt type 'wsl' and check if the wsl is already installed, otherwise install by using command 'wsl --install'
- Turn on windows subsystem for linux in the control panel
- Go to downloads on the file explorer and install docker desktop.
- Accept the terms and conditions, select student in the role and click on continue
- Type https://github.com/docker/welcome-to-docker in the browser
- Copy the ~~code~~ url from the repository
- In cmd, create a directory and type cd <directory-name>
- Inside the directory clone the repository by typing command

  git clone <link>
- Get into the repository by typing cd welcome-to-docker
- To build: docker build -t welcome-to-docker .
- To run: docker run -p 5000:5000 welcome-to-docker.
- Go to docker-desktop and click on just appeared (or so) welcome-to-docker.



D D M M Y Y Y Y

9) **Manually**

- Create a directory in the command prompt
  Ex: mkdir dockerproj
- Enter into the directory : cd dockerproj
- type notepad app.py and type the following code.

```
from flask import flask
app = Flask(_name_)
@app.route ('/')
def hello-docker():
    return "Hello Docker!"

if __name__ == "__main__":
    app.run( host = '0.0.0.0', port=5000)
```

- type notepad requirements.txt and type the following

```
flask == 2.0.1
Werkzeug == 2.0.1
```

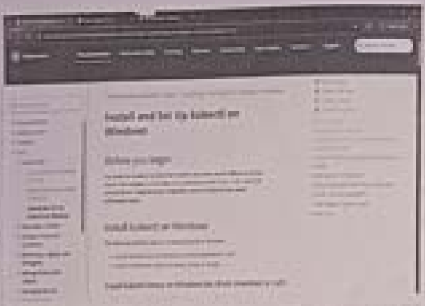
- type notepad Dockerfile and type the following code

```
FROM Python:3.9-slim
WORKDIR /app
COPY requirements.txt
RUN pip install --no-cache-dir -r
requirements.txt
```

container
COPY . .
EXPOSE 500
CMD ["python", "app.py"]

- Go to folder/directory location and type cmd
- > ren Dockerfile.txt Dockerfile
- To build : docker build -t dockerproj .
- To run : docker run -p 5000:5000 dockerproj.

- Go to docker desktop and open the file with your project name (dockerproj)
- The output will be Hello Docker!


Install and Set Up kubectl on Windows

Before you begin

Install kubectl on Windows

Install kubectl binary on Windows (via direct download or curl)

Installation

LAB-07

- Automate the process of running containerized application using Kubernetes:

* Kubernetes:
  It is an open-source container which automates the deployment, scaling and management of containerized applications. It allows us to efficiently manage and scale applications in a clustered environment with features like self-healing, load balancing, and automated rollouts.

* Minikube:
  It is a tool to create a local Kubernetes cluster on machine, allows to run Kubernetes in a single-node environment for development and testing.

- Install kubectl binary on windows using curl
  → curl.exe -LO "https://dl.k8s.io/release/v1.32.0/bin/windows/amd64/kubectl.exe"

- Validate the binary:
  Download the kubectl checksum file:
  → curl.exe -LO "https://dl.k8s.io/v1.32.0/bin/windows/amd64/kubectl.exe.sha256"

- Validate the kubectl binary against the checksum file:
  → CertUtil -hashfile kubectl.exe SHA256 type kubectl.exe.sha256

- Using powershell to automate the verification using the -eq operator to get a True or False result:
  → $(CertUtil -FileHash -Algorithm SHA256 .\kubectl.exe).Hash -eq $(Get-Content .\kubectl.exe.sha256)

- kubectl version --client

- Install Minikube

Minikube is local kubernetes, focusing on making it easy to learn and develop for kubernetes.

1. Download and run the installer for the latest release using the above command.
   → New-Item -Path 'C:\' -Name 'minikube' -ItemType Directory -Force
   → Invoke-WebRequest -OutFile 'C:\minikube\minikube.exe'
     -Uri 'https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe'
     - Use BasicParsing

[illegible]

2. Add the minikube.exe binary to your path
   → $oldPath = [Environment]::GetEnvironmentVariable('path', [EnvironmentVariableTarget]::Machine)

   → if ($oldPath.split(';') -inotcontains 'C:\minikube') { [Environment]::SetEnvironmentVariable('path', $('{0};C:\minikube' -f $oldPath), [Environment VariableTarget]::Machine)

3. Start your cluster
   → minikube start

4. Interact with your cluster
   → kubectl get po -A
   → minikube kubectl -- get po -A
   → alias kubectl="minikube kubectl --"
   → minikube dashboard

5. Deploy applications
   → kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
   → kubectl expose deployment hello-minikube --type=NodePort --port=8080
   → kubectl get services hello-minikube
   → minikube service hello-minikube
   → kubectl port-forward service/hello-minikube 7080:8

6. Manage your cluster
   → minikube pause
   → minikube stop

Getting Started

Instance Configuration

[illegible]

[illegible]

LAB-08

Demonstrate Creating Job using Jenkins:

Jenkins:
It is a free and open source automation server used for continuous integration and delivery
→ Using Jenkins applications are transferred to testing and production

Steps to Install Jenkins:

- Install latest version of JDK
- Go to Chrome browser and type Jenkins for windows
- Once it is downloaded open JDK and Install the JDK
- Go to C:drive → programs file → eclipse
  Adoptium → JDK bin
- Copy this path and edit it into environmental variables
- Check the Java version : Java --version
- Search local security policies and select it
- Select user right assignment and select log on as a service → add user → type administrator → select and apply
- Go to Downloads then double click on Jenkins

Getting Started

- To run service as a local or domain user since we don't know the password, select "Run service as local system"
- Click on test code. If failed give another port number
- Click on next & disable "start service" feature & select "other feature will be unavailable"

- Search service in start & find Jenkins right click on it and select start & close the window

- C:Drive → program files → Jenkins → Jenkins.exe. Check if server is running

- Go to chrome & type "localhost:8080"

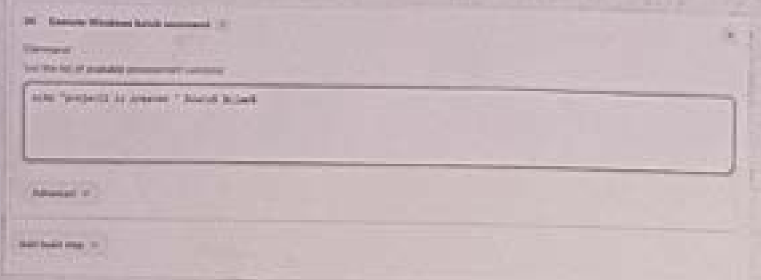
- Open Jenkins.err for password

- To customize Jenkins and select install suggested plugins

- To create first admin user enter the details and click on save and continue and the save & finish

- Click on start using Jenkins

- Click on new item, enter name select freestyle project and click on ok

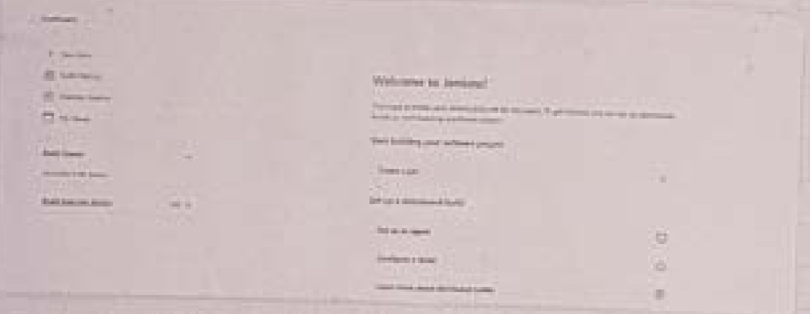


Execute Windows batch command

Command

[illegible]

```
[illegible]
```



Welcome to Jenkins!

[illegible]

D.S.C.E.

- Go to Build steps, click on execute windows batch command

- Type [echo "proj \ is created %date% %time%] and save it

- Build is successful

- Click on console output.

D.S.C.E.

```
[illegible]
```

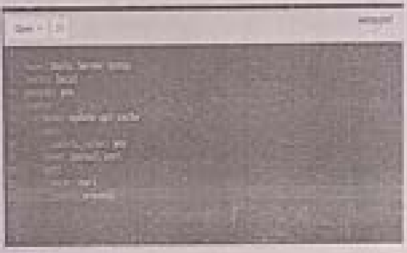
LAB-09

Configuration Management with Ansible

Ansible is an open-source IT automation engine that simplifies and automates various IT processes, including configuration. Developed by Michel Dettan and now maintained by the Ansible community and Ansible Inc., it is known for its agentless inc., it which means it does not require any software to be installed on the managed nodes. Instead, Ansible uses temporary remote connection via SSH or windows remote management (WinRM) to execute tasks. It is widely used for managing cloud and on premises infrastructure and is supported on multiple operating systems.

Ansible uses a simple language called YAML to define playbook, which are scripts that automate tasks. These playbook can be used to define the desired state of a system, such as installing software, configuration services, or deploying application.

```
[illegible]
```



```
[illegible]
```

Steps:

# This program to be executed on ubuntu

- In this initial step we need to update the ubuntu system through the below given command

```
sudo apt update
```

- Install ansible

```
sudo apt install ansible
```

- Check & confirm the installation

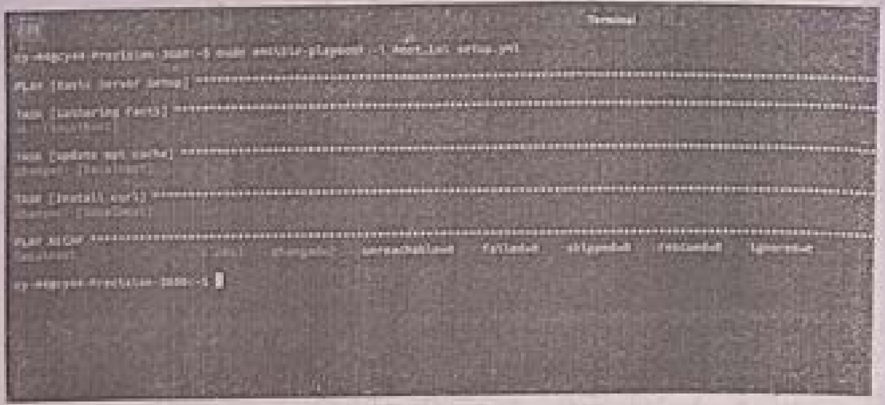
```
ansible --version
```

- Create a playbook by using the below given command

```
nano host.ini
[local]
localhost ansible.connection=local
```

- Create a Yaml file

```
nano setup.yaml
- name: Basic Server Setup
  hosts: local
  become: yes
  tasks:
    - name: update apt cache
  apt:
    update_cache: yes
```



```
[illegible]-Precision-[illegible]:~$ sudo ansible-playbook -i host_ini setup.yml

PLAY [Basic Server Setup] ******************************************************

TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [Update apt cache] ********************************************************
changed: [localhost]

TASK [Install curl] ************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost          : [illegible]   changed=[illegible]   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

[illegible]-Precision-[illegible]:~$
```

DD MM YYYY

```
- name: Install curl
  apt:
    name: curl
    state: present
```

- copy the above given yaml code and validate its syntax using yaml validator on chrome

- Run the program

  sudo ansible-playbook -i hosts.ini setup.yaml

```
[illegible]
```

```
[illegible]
```

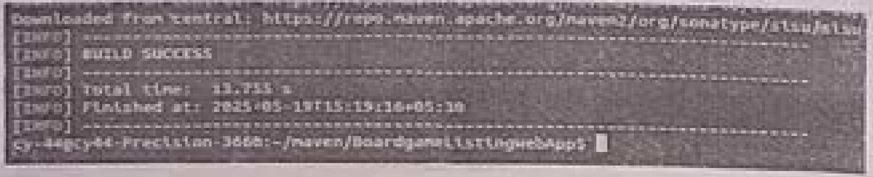
LAB 10

Creating Maven project, understanding POM file, dipendting management & plugins

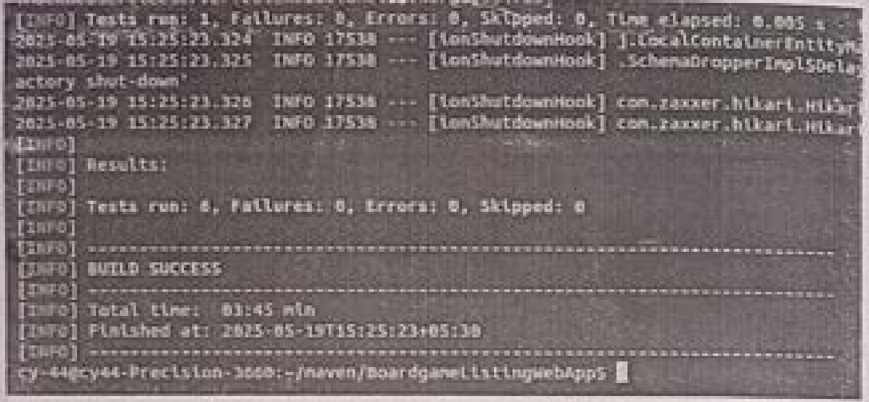
Maven is a building automation and project management tool developed using the java programming language, primarily used for java-based projects to manage the build process, including source code compilation, testing, packing and more.
It utilizes the project object model (POM), where the pom.xml file describe the projects configuration & depending management.
Maven Simplifies project depending management, build automation, and configuration handling, making it easier for teams to manage complex projects.
It also supports various plugins for tracks such as compilation, testing, packing, deployment, & documentation generation.

The build documentation contains

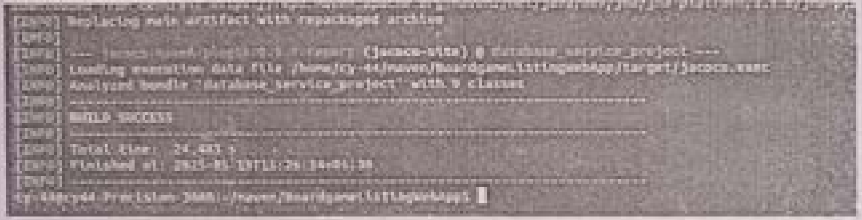
- Clean
- compile
- package
- Validate
- Test



```
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/sisu/sisu
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  13.755 s
[INFO] Finished at: 2025-05-19T15:19:16+05:30
[INFO] ------------------------------------------------------------------------
cy-44@cy44-Precision-3660:~/maven/BoardgameListingWebApp$ 
```



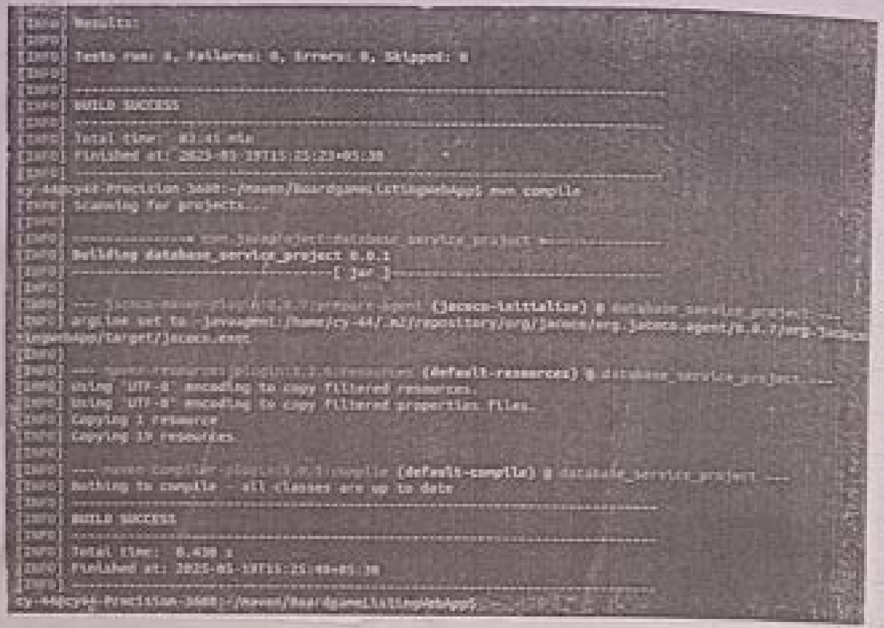
```
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.085 s -
2025-05-19 15:25:23.324  INFO 17538 --- [ionShutdownHook] j.LocalContainerEntityMa
2025-05-19 15:25:23.325  INFO 17538 --- [ionShutdownHook] .SchemaDropperImpl$Delay
actory shut-down'
2025-05-19 15:25:23.326  INFO 17538 --- [ionShutdownHook] com.zaxxer.hikari.Hikar
2025-05-19 15:25:23.327  INFO 17538 --- [ionShutdownHook] com.zaxxer.hikari.Hikar
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  03:45 min
[INFO] Finished at: 2025-05-19T15:25:23+05:30
[INFO] ------------------------------------------------------------------------
cy-44@cy44-Precision-3660:~/maven/BoardgameListingWebApp$ 
```



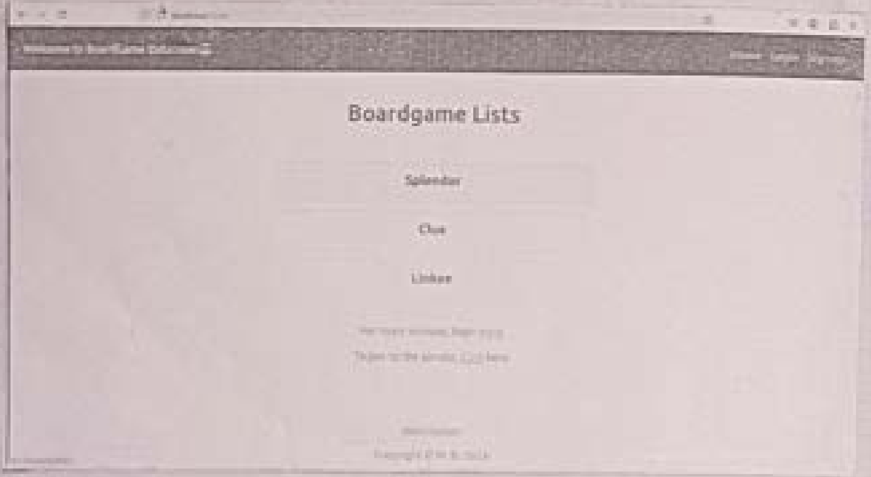
```
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- [illegible] (jacoco-site) @ database_service_project ---
[INFO] Loading execution data file /home/cy-44/maven/BoardgameListingWebApp/target/jacoco.exec
[INFO] Analyzed bundle 'database_service_project' with 9 classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  24.483 s
[INFO] Finished at: 2025-05-19T15:26:34+05:30
[INFO] ------------------------------------------------------------------------
cy-44@cy44-Precision-3660:~/maven/BoardgameListingWebApp$ 
```

Steps:

- This experiment will be preferred in ubuntu system
- Update the ubuntu system before starting the execution of this lab
  sudo apt update
- Install the open jdk by the below command:
  sudo apt install open jdk -11-jre-headless -y
- Install maven by the given command:
  sudo apt install maven -y
- Create a directory
  mkdir maven
  cd maven
- Install git
  sudo apt install git
- Clone a repo for execution
  git clone https://github.com/Krishpluto/Boardgame
  hosting webApp.git
- Navigate inside the directory called boardgame and execute the below commands
  mvn clean
  mvn validate



```
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  03:41 min
[INFO] Finished at: 2025-03-19T15:23:23+05:30
[INFO] ------------------------------------------------------------------------
cy-44@cy44-Precision-3660:~/maven/BoardgameListingWebApp$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -------------< com.javaproject:database_service_project >--------------
[INFO] Building database_service_project 0.0.1
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.7:prepare-agent (jacoco-initialize) @ database_service_project ---
[INFO] argLine set to -javaagent:/home/cy-44/.m2/repository/org/jacoco/org.jacoco.agent/0.8.7/org.jaco
tingWebApp/target/jacoco.exec
[INFO]
[INFO] --- maven-resources-plugin:3.2.0:resources (default-resources) @ database_service_project ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Using 'UTF-8' encoding to copy filtered properties files.
[INFO] Copying 1 resource
[INFO] Copying 19 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ database_service_project ---
[INFO] Nothing to compile - all classes are up to date
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.438 s
[INFO] Finished at: 2025-03-19T15:25:49+05:30
[INFO] ------------------------------------------------------------------------
cy-44@cy44-Precision-3660:~/maven/BoardgameListingWebApp$
```



Boardgame Lists

Splendor

Clue

Linkee

D.S.C.E.

mvn test
mvn compile
mvn package

- Move to target directory
  cd target
  java -jar database-service-project-0.0.1.jar

- We can see this webpage being hosted on our localhost
  localhost:8080

D.S.C.E.