Experiment ①:-

⇒ Install JDK-17, Maven, Gradle, IDE & set up procedure. Differences b/w Maven & gradle.
Run a simple Java program.

Steps:- Maven Installation

Chrome → Search for maven download → Binary Zip archive for (apache maven) window

→ download the file in desktop → double click & open → Install (if not extracted, extract it) [C-drive]/anywhere u want

Copy the path ~~[illegible]~~ & goto → properties → Advance s/m settings

Note:- ① Make sure u have installed Java.
cmd prompt ⇒ java --version.

② check mvn --version.

→ Environment variables → s/m variables → new →
Variable name ⇒ MAVEN_HOME (uppercase)
Variable value ⇒ paste the path where u have apache maven folder.

→ new → M2_HOME ⇐ Variable name

copy the same path as for Maven ⇐ Variable value

Search for the [path] variable → [edit] → copy the path of the bin folder & paste the path in the [path] (open) variable.

Gradle Installation

Browser → gradle download → Installation Gradle → Releases → download : [binary-only] ← click on it (it downloads)

Now create a folder in C-drive → paste the downloaded zip file → extract

Note! - In the folder u have created say for example ⇒ [Gradle] folder → gradle-8.13 (extracted)

Now go to [gradle-8.13] folder → open bin → now copy the path

rt click
This PC → properties → Advanced S/m Setth. → Environ variables → path
new → paste
or
edit → new → paste the path → ok → ok → ok

Now go to cmd ⇒ gradle -v.

will get version of gradle

cmd — java — version
mvn — version
gradle — version

Experiment (2) :-

Install Maven, create a Maven Project, POM file
IDE

IntelliJ Idea → file → new → Project → Maven Archetype
→ Name : hep2
Location : default let it be
Build S/m : Maven
JDK : 17

[Create]

→ POM file will be created.
(XML)

→ HTML, CSS, LOGO files need to be given for students

→ push to git hub & generate URL

→ Make use of [Maven resources plugin]

```
<build> <plugins>
                (Here paste maven resources plugin will be given copy & paste it here
</build> <plugins>
```

// TO CREATE DOCS FOLDER.

[/docs]

→ Terminal

| mvn clean install |

after running this command /docs folder will be created.

→ git push -u origin master

→ settings → pages → Deploy from a branch

| master | | ~~root~~ /docs | | save |

Now URL will be generated i.e

Ur site is live on that URL...! ☺

Experiment (3) :-

→ Install gradle, IDE, JAVA, DSL, Build a Project (Both the languages).

Intellij Idea → Create New project → Select Gradle. (under Java/kotlin) →

Set the Group ID (Eg:- com.example) → click finish

language (Java)	Gradle DSL — Groovy

→ Here Build.gradle is important file

→ Modify the build.gradle, delete all the code & replace it with the code provided in the Github.

~~Src → main →~~

Main class should be in Org.example

→ terminal Gradle run or rt. side Gradle → tasks → application → run (double click)

→ Copy the docs folder to the project created

→ Create repo in Git hub & push

→ ~~Create~~ Create URL

→ Task → verification → test [rt. corner side of the window]
or
terminal gradle test ↵

→ gradle ~~Jarra~~ -jar.

Jarra -jar ./build/libs/(Name of the file Pres Tab) ↵

↑ current directory

o/p Hello from Gradle - - - !

→ Gradle lifecycle

① Initialize project → Creates a new gradle project (gradle init)

② Compile code → compiles the project

③ Run application → (gradle run) Runs the application

④ Clean Build files → Deletes the old build files (gradle clean)

⑤ Run tests → Executes unit tests (gradle test)

⑥ Generate JAR → packages the project into a JAR (gradle Jar)

⑦ Deploy to Gitlab pages → pushes website to Gitlab (git push origin main)

Experiment (5)

What is Jenkins?
Installing steps of Jenkins.
Configure Jenkins for first use.

→ It is a CI-CD tool.

→ CI-CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development

→ The main concepts of CI-CD are → Continuous Integration
→ " delivery
→ " deployment

~~192.168.33.10 web~~
~~192.168.33.11 db~~

→ Show the source code in IDE run, & push to Git hub, generate URL, same to be migrated to Jenkins & execute.

→ Need to write installation steps of Jenkins

Exp (4)

Build & Run a JAVA application with Maven & Migrate the same to gradle

As given in manual

Experiment (6)

Same as (5)

Experiment (7)

What is Ansible?
Why Ansible?
What is Inventory file?
What is Playbook.yml?
Commands to execute
Show the o/p in Browser

Experiment (8)

→ Manual creation of Java project for selenium test

→ New project → selenium ~~test~~ Automation test → JAVA
src → (delete main file) → new package → org.test → rt click → Java class → FirstSelenium.Java → paste the code.
search for

→ Go to Browser → selenium.dev → downloads → select selenium server (Grid) → latest stable version 4.29.0
↑ click on this
→ download & save it.

→ file → project structure → libraries → + → Java → ~~sel~~ got project → lib → JAR file (select) → add → add → OK

→ JAR – Java Archive → it is a pkg. of collecting several classes & Interfaces. which help us to perform some specific tasks.

→ Here we are automating web application for that we are making use of Automation framework called "Selenium" (It is open source)

→ we need selenium Jar file which will interact with the web application (Browser)

→ In the same browser related to selenium go down & select [+ Browsers for chrome] → Click on documentation → select stable release for chrome → download → save.

⋮ → help → about → now u can see ~~your~~ ur browser version

→ rt. click on project → new directory → create lib → copy Jar file there → ok.

→ rt-click on project → new directory → Create drivers → copy chrome driver → ok