Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

S004

Continuous Assessment for Laboratory / Assignment sessions

Academic Year 2024-25

Name: Aditi Ambasta SAP ID: 60018220113 Course: **DevOps**

Course Code: **DJS22ADC5014** **Year:** T. Y. B. Tech Sem: V Batch: A1 & A2

Department: Artificial Intelligence(AI) & Data Science

Performance Indicators (Any no. of Indicators) (Maximum 5 marks per indicator)	1	2	3	4	5	6	7	8	A1
Course Outcome	1	2	3	3	4	4	5	5	1,2,3
1. Knowledge (Factual/Conceptual/Procedural/ Metacognitive)	5	5	5	5	5	5	5	5	5
2. Describe (Factual/Conceptual/Procedural/ Metacognitive)	5	5	5	5	5	5	5	5	5
3. Demonstration (Factual/Conceptual/Procedural/ Metacognitive)	5	5	5	5	5	5	5	5	5
4. Strategy (Analyse & / or Evaluate) (Factual/Conceptual/ Procedural/Metacognitive)	4	4	4	4	4	4	4	5	5
5. Interpret/ Develop (Factual/Conceptual/ Procedural/Metacognitive)	-	-	-	-	-	-	-	-	
6. Attitude towards learning (receiving, attending, responding, valuing, organizing, characterization by value)	4	4	4	4	4	4	4	4	5
7. Non-verbal communication skills/ Behaviour or Behavioural skills (motor skills, hand-eye coordination, gross body movements, finely coordinated body movements speech behaviours)	-	-	-	-	-	-	-	-	
Total	23	23	23	23	23	23	23	24	25
Signature of the faculty member									

Outstanding (5), Excellent (4), Good (3), Fair (2), Needs Improvement (1)

Total Term-work (25) =	24
Sign of the Student:	Aditi

Signature of the Faculty member:
Name of the Faculty member:Date:

Signature of Head of the Department

EXPERIMENTNO.1

SEMESTER:V
SUBJECT:DevopsLaboratory(DJS22ADL5014)
NAME OF THE STUDENT: Aditi Ambasta ROLLNO.: S004/60018220113

Aim:Writecodeforasimpleuserregistrationformforanevent.
ToStudyDevOps:Principles,Practices,andDevOpsEngineerRoleandResponsibilities. Post

Lab Question:

WhataretheEssentialSkillsrequiredforaDevOps Engineer?
DescribeDevOpsprincipleindetail.

DevOpsPrinciples

In short, the main principles of DevOps are automation, continuous delivery, and fast reaction to feedback. You can find a more detailed explanation of DevOps pillars in the CAMS acronym:

Culturerepresentedbyhumancommunication,technicalprocesses,andtools

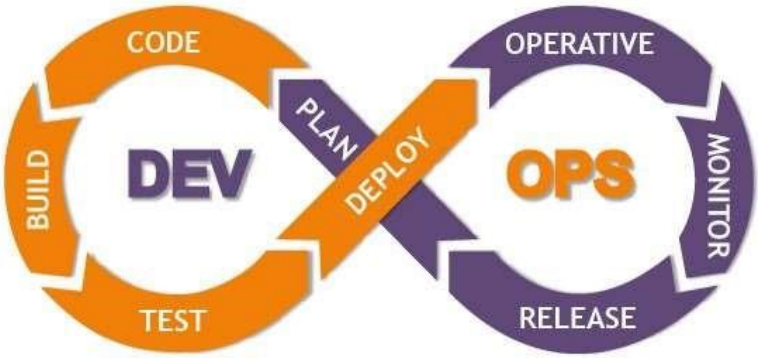
Automationofprocesses

MeasurementofKPIs

Sharingfeedback,bestpractices,and knowledge

Adherence to these principles is achieved through a number of DevOps practices that include continuous delivery, frequentdeployments,QAautomation,validatingideasasearly as possible, and in-team collaboration.

DevopsPractices



- Plan – In DevOps planning plays an important role. In this stage, all the requirements of the project and everything regarding the project like time for eachstage,cost.etcarediscussed.Thiswillhelpeveryoneinteamstogeta brief idea about the project.
- Code – In this Stage the code is written over here according to the client's requirements. Here the code is divided into small codes called Units. This is done to get a clear picture of the code. For example, if the team is doing a project on an online -Ekart application then the login part is divided as one unit,afterloginthepagewhichshowsallthecategoriesisdividedasanother unit, user profile as another unit, etc.

Some of the examples of the tools used are Git, JIRA

- Build–InthisstageBuildingoftheunitsisdone.Someoftheexamplesofthe tools used are maven, Gradle.
- Test – Testing of all units is done in this stage. So we will get to know where exactlythecodeishavingbugsandiftherearemistakesfounditisreturned. Some of the examples of the tools used are Selenium, PYtest
- Release– At this point, the build is prepared to be deployed in the operational environment. The DevOps department prepares updates or sends several versions to production when the build satisfies all checks basedontheorganizationaldemands.
- . An example of the tool used is Jenkins.
- Deploy–Inthisstage,thecodeisdeployedontheclient'senvironment. Some of the examples of the tools used are AWS, Docker.
- Operate–Operationsareperformedonthecodeifrequired.Someofthe examples of the tools used are Kubernetes, open shift.
- Monitor–Inthisstagemonitoringoftheapplicationisdoneoverhereinthe client's environment. Some of the examples of the tools used are Nagios, elastic stack.

DevOpsEngineerRolesand Responsibilities
ADevOpsEngineerisalinkbetweensoftwaredevelopmentandIToperations.Theyare

responsiblefordesigning,implementing,andmaintainingthesoftwaredevelopmentprocess and ensuring that the software is deployed efficiently and reliably.

TheDevOpsEngineerrolesandresponsibilitiesmay include:

- Automation:
  DevOpsEngineersareresponsibleforautomatingasmuchofthesoftware developmentprocessaspossible.Thisincludes automating code testing, building, deployment,andmonitoring.Automationhelpstoreduceerrorsandimprovethe efficiencyofthesoftwaredevelopmentprocess.
- InfrastructureasCode:
  DevOps Engineers are responsible for developing and maintaining the infrastructure ascode.TheyusetoolslikeAnsible,Puppet,orCheftomanageinfrastructure,and theyworkwithcloudcomputingplatformslikeAWS,Azure,orGoogleCloud.
- ContinuousIntegrationandDelivery:
  DevOpsEngineersareresponsibleforimplementingandmaintainingContinuous IntegrationandContinuousDelivery(CI/CD)Pipelines.Thisinvolvesautomatingthe buildingandtestingofcode,ensuringthatthesoftwareisalwaysinareleasable state.
- MonitoringandTroubleshooting:
  DevOps Engineers are responsible for monitoring thesoftwaredevelopmentprocess and identifying and resolving issues. They use tools like Prometheus, Grafana, or Datadogtomonitorthehealthofthesystem,andtheyworkwithdeveloperstо troubleshootissues.
- Collaboration:
  DevOps Engineers work closely with development and operations teams to ensurethatthesoftwaredevelopmentprocessiscollaborativeandefficient.They communicatewithstakeholdersandensurethateveryoneisalignedwiththegoalsof the project.
- Security:
  DevOpsEngineersareresponsibleforensuringthatthesoftwaredevelopment processissecure.Thisincludesimplementingsecuritymeasuresthroughoutthe softwaredevelopmentlifecycle,conductingsecurityaudits,andensuringthat securitypoliciesandproceduresare followed.

EssentialSkillsforaDevOps Engineer

DevOps is an approach to software engineering that combines software development with operations, typically to enable faster delivery of software and services. DevOps engineers are responsible for managing the development, deployment, and maintenance of software applicationsandservices.Assuch,theyneedtopossessawiderangeofskillstobe successful.

HerearesomeEssentialSkillsrequiredforaDevOpsEngineer:

- AutomationSkills:

  Puppet, or Chef.Theyshouldbeabletoautomatethesoftwaredevelopmentprocess as much as possible to reduce errors and improve efficiency.
- Cloud Computing Skills:
  DevOpsEngineersshouldbeproficientincloudcomputingplatformslikeAWS,Azure, or Google Cloud. They should have experience with designing and maintainingcloudbased infrastructure, including networking, storage, and security.
- ContainerizationSkills:
  DevOps Engineers should be familiar with containerization technologies like Docker (Docker is a platform for developing, packaging, and deploying applications in containers) and Kubernetes (Kubernetes is an open-source container orchestration platform for automating deployment, scaling, and management of containerized applications). They should be able to create and manage container-based deployments and troubleshoot issues.
- ContinuousIntegrationandDelivery(CI/CD)Skills:
  DevOpsEngineersshouldbeexperiencedwithimplementingandmaintainingCI/CD pipelines.Theyshouldhaveagoodunderstandingofthesoftwaredevelopment processandbeabletoautomatecodetesting,building,deployment,and monitoring.
- MonitoringandTroubleshootingSkills:

DevOps Engineers should be able to monitor the software development process and identify and resolve issues quickly. They should have experience with monitoring toolslikePrometheus,Grafana,orDatadogandbeabletotroubleshootissueswith developers.

- CollaborationSkills:
  DevOpsEngineersshouldbeabletoworkcollaborativelywithdevelopmentand operations teams. They should have excellent communication and problem-solving skillsandbeabletofosteracultureofcollaborationand efficiency.
- Security Skills:
  DevOpsEngineersshouldbeabletoensurethatthesoftwaredevelopmentprocessis secure.Theyshouldhaveexperiencewithimplementingsecuritymeasures throughoutthesoftwaredevelopmentlifecycle,conductingsecurityaudits,and ensuringthatsecuritypoliciesandproceduresare followed.
- Orchestration:
  DevOpsengineersshouldbeabletoorchestratecomplex,distributedsystemsto ensurethatallcomponentsofthesystemarefunctioningcorrectly.
- CommunicationSkill:
  DevOps Engineer should possessexcellentcommunicationskills.Theyshouldbeable to collaboratewithdevelopmentandoperationsteamstoidentifyandresolveissues, andtheyshouldbeable to communicate the status of the software development process to stakeholders effectively.

CODE:

Frontend:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Register</title>
  <link
href="https://fonts.googleapis.com/css2?family=Rubik:wght@300;400;500;700&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css">
  <style>
    :root {
      --bg-dark: #121212;
      --card-dark: #1E1E1E;
      --primary-color: #FF6B6B;
      --text-light: #FFFFFF;
      --text-muted: #B0B0B0;
    }

    * {
      box-sizing: border-box;
    }

    body {
      height: 100vh;
      background-color: black;
      background-image:
        linear-gradient(to right, rgba(255, 255, 255, 0.13) 1px, transparent 1px),
        linear-gradient(to bottom, rgba(255, 255, 255, 0.101) 1px, transparent 1px);
      background-size: 80px 80px; /* Size of grid squares */
      color: white;
      font-family: 'Rubik', Arial, sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      margin: 0;
      padding: 20px;
```

```
}

.auth-container {
  width: 100%;
  max-width: 500px;
}

.auth-card {
  background-color: var(--card-dark);
  border-radius: 16px;
  padding: 40px;
  box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
}

h2 {
  text-align: center;
  color: var(--primary-color);
  margin-bottom: 30px;
}

.form-group {
  margin-bottom: 20px;
}

label {
  display: block;
  margin-bottom: 8px;
  color: var(--text-muted);
}

input {
  width: 100%;
  background-color: var(--bg-dark);
  border: 1px solid #333;
  color: var(--text-light);
  padding: 12px 15px;
  border-radius: 8px;
  transition: border-color 0.3s ease;
  margin-bottom: 10px;
}
```

```
input:focus {
   outline: none;
   border-color: var(--primary-color);
}

.avatar-preview-container {
   display: flex;
   justify-content: center;
   margin-bottom: 20px;
}

.avatar-img {
   width: 120px;
   height: 120px;
   border-radius: 50%;
   object-fit: cover;
   border: 3px solid var(--primary-color);
}

.btn-primary {
   width: 100%;
   background-color: var(--primary-color);
   color: var(--bg-dark);
   border: none;
   padding: 12px;
   border-radius: 8px;
   cursor: pointer;
   font-weight: 600;
   transition: background-color 0.3s ease;
}

.btn-primary:hover {
   background-color: #FF5252;
}
.input-icon {
   position: relative;
}
.input-icon i {
   position: absolute;
   left: 15px;
   top: 40%;
```

```
        transform: translateY(-50%);
        color: var(--text-muted);
      }
      input.icon-input {
        padding-left: 40px;
      }
    </style>
</head>
<body>
    <div class="auth-container">
      <div class="auth-card">
        <h2>Register</h2>
        <form method="POST">
          <div class="input-icon">
            <i class="fas fa-user"></i>
            <input type="text" name="username" id="username" placeholder="Username"
class="icon-input" required>
          </div>
          <div class="input-icon">
            <i class="fas fa-envelope"></i>
            <input type="email" name="email" id="email" placeholder="Email" class="icon-
input" required>
          </div>
          <div class="input-icon">
            <i class="fas fa-lock"></i>
            <input type="password" name="password" id="password" placeholder="Password"
class="icon-input" required>
          </div>

          <div class="form-group avatar-preview-container">
            <img id="avatar-preview" src="https://api.multiavatar.com/default.svg" alt="Avatar
Preview" class="avatar-img">
          </div>

          <button type="submit" class="btn-primary">
            <i class="fas fa-user-plus"></i> Register
          </button>
        </form>
      </div>
    </div>

    <script>
```

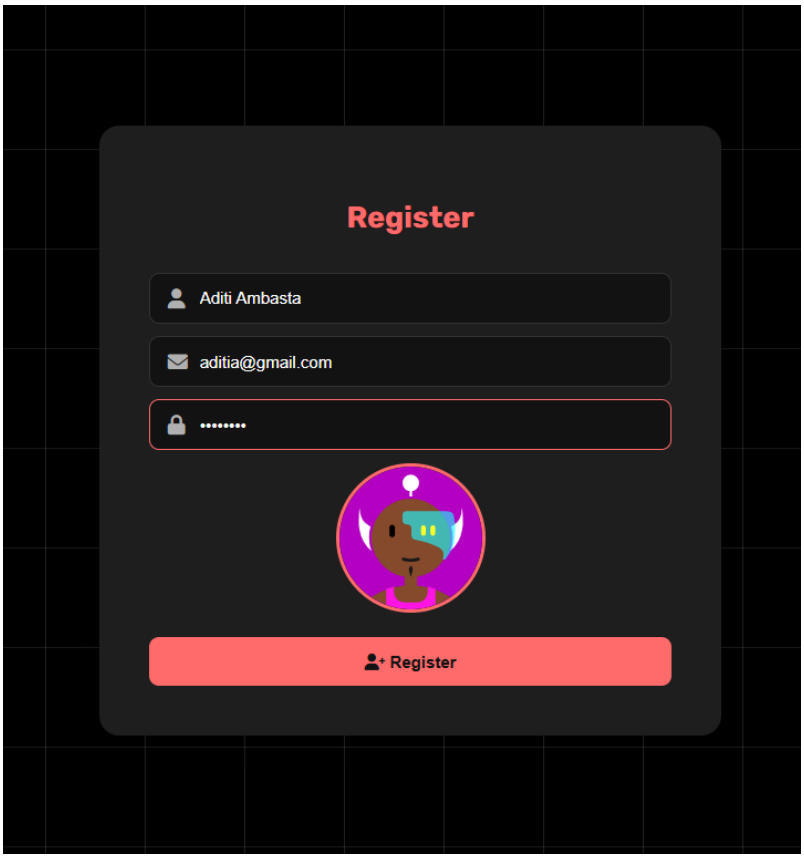
```
    document.getElementById('username').addEventListener('input', function() {
      const username = this.value;
      const avatarPreview = document.getElementById('avatar-preview');
      if (username) {
        avatarPreview.src = `https://api.multiavatar.com/${username}.svg`;
      } else {
        avatarPreview.src = `https://api.multiavatar.com/default.svg`;
      }
    });
  </script>
</body>
</html>
```

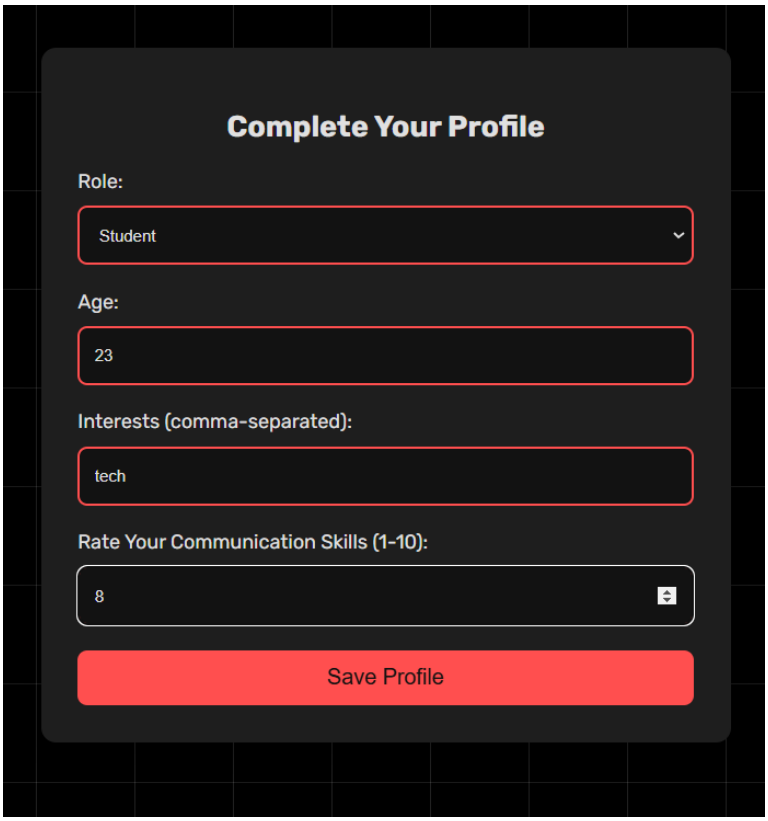
BACKEND:

```
from flask import Flask, render_template, request, jsonify
from flask_cors import CORS
from flask_mysqldb import MySQL
from flask_bcrypt import Bcrypt

app = Flask(__name__)
CORS(app)
bcrypt = Bcrypt(app)

# MySQL configuration
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'speakboost_user'
app.config['MYSQL_PASSWORD'] = 'ipd_password'
app.config['MYSQL_DB'] = 'speakboost'

mysql = MySQL(app)

@app.route('/register', methods=['GET', 'POST'])
def register():
  if request.method == 'POST':
    username = request.form['username']
    email = request.form['email']
    password = request.form['password']

      # Generate a unique avatar URL using Multiavatar API (using username for uniqueness)
    avatar_url = f"https://api.multiavatar.com/{username}.svg"  # Unique avatar URL

    # Hash password
    hashed_password = bcrypt.generate_password_hash(password).decode('utf-8')
```

```
        # Insert data into the database without role
        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO users (username, email, password, avatar_url, role) VALUES
(%s, %s, %s, %s, %s)",
                    (username, email, hashed_password, avatar_url, None))  # Leave role as None
        mysql.connection.commit()
        cur.close()

        flash("Registration successful! Please log in.", "success")
        return redirect(url_for('login'))  # Redirect to login page after registration

    return render_template('register.html')
```

OUTPUT:





id	username	email	password	created_at	role	avatar_url	age	interests	communication_rating
1	aditi	aditi@gmail.com	$2b$12$2v5FCsgIjOxe39YNwQuqauPnH9XeSh...	2024-11-19 14:02:37	Student	NULL	44	teach	4
2	adi	adi@gmail.com	$2b$12$r1IxYTyt57Ebwh7ZiDKZTen6FzG0qoBN...	2024-11-19 14:16:06	NULL	NULL	NULL	NULL	NULL
3	shreya	sh@gmail.com	$2b$12$A7sBoSTjw0Vp1oyw4MqUc.kfb34KDYy...	2024-11-19 15:03:39	Student	NULL	44	teach	5
4	new	new@gmail.com	$2b$12$fTemQ4tINxyM1yUn.A9SkevWO4TuQ...	2024-11-19 16:32:44	Podcaster	https://api.multiavatar.com/new.svg	NULL	NULL	NULL
5	mahi	mahi@gmail.com	$2b$12$w1uefHwG1H5ZDp0pIQAhfO845S7LW...	2024-11-19 16:47:16	Podcaster	https://api.multiavatar.com/mahi.svg	NULL	NULL	NULL
6	mira	mira@gmail.com	$2b$12$hPij2GoniEgCBxf3KnAQHeWK1XoA7ex...	2024-11-20 20:37:53	Student	https://api.multiavatar.com/mira.svg	55	teach	5
7	mike	mike@gmail.com	$2b$12$aKmT/sINnkvkn996xI2p7OkPa2CUNIm...	2024-11-21 16:19:26	Student	https://api.multiavatar.com/mike.svg	23	technology	6
8	test	test@gmail.com	$2b$12$/f9OS/ec3jmBpJv/pXjnuetuA14Rv1nqc...	2024-11-22 23:17:20	Social Media Influencer	https://api.multiavatar.com/test.svg	23	music	7
9	Aditi Ambasta	aditia@gmail.com	aditia@gmail.com 5.I3ESQ5AD6ReO0cXFYBBtf...	2024-11-26 20:03:24	NULL	https://api.multiavatar.com/Aditi Ambasta.svg	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Conclusion : Hence we study and understand devops concepts in terms of principle and skill requirement. Also create a single web page using different web based language.

Aditi Ambasta
600182213
S004 - AIDS

Devops Exp - 01

1. What are the essential skills required for a Devops Engineer

→ Devops is an approach to software engineering that combines software development to enable faster delivery of software & services. Devops Engineer are responsible for managing the development, deployment & maintainence of software applications and services. As such they need to possess a wide range of skills to be successful.

Here are some essential skills:

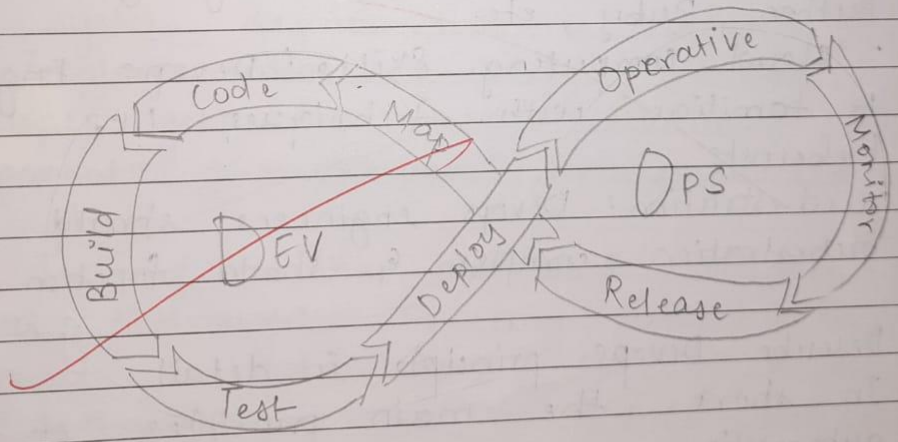
- Automation skills: Devops engineers should have strong automation skills, including experience with scripting language such as Python, Ruby, etc.
- Cloud computing skills: Devops Engineers should be familiar with techniques like Docker & Kubernets.
- Orchestration: Devops engineers should be able to orchestration complex & shauld function correctly.

2. Describe Devops principle in detail

→ In short, the main principles of Devops are automation, continuous delivery & fast reaction to feedback. You can find a more detailed explained of Devops pillars in the CAMS acronym.

- Culture represented by human communication, technical process & tools.
- Automation of process
- Measurement of KPI's
- Sharing feedback, best practices & knowledge.

Adherence to these principles is achieved through a number of Devops practices that include continuous delivery, frequency development, deployment, QA automation, validating ideas as early as possible, and in team collaboration.



23

23/9/24.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

EXPERIMENT NO. 2

SEMESTER:V **SUBJECT: Devops Laboratory(DJS22ADL5014)**
NAME OF THE STUDENT: Aditi Ambasta **SAP ID: S004/60018220113**

Aim:To carry out Version Control System / Source Code Management, install git andcreate a GitHub account.

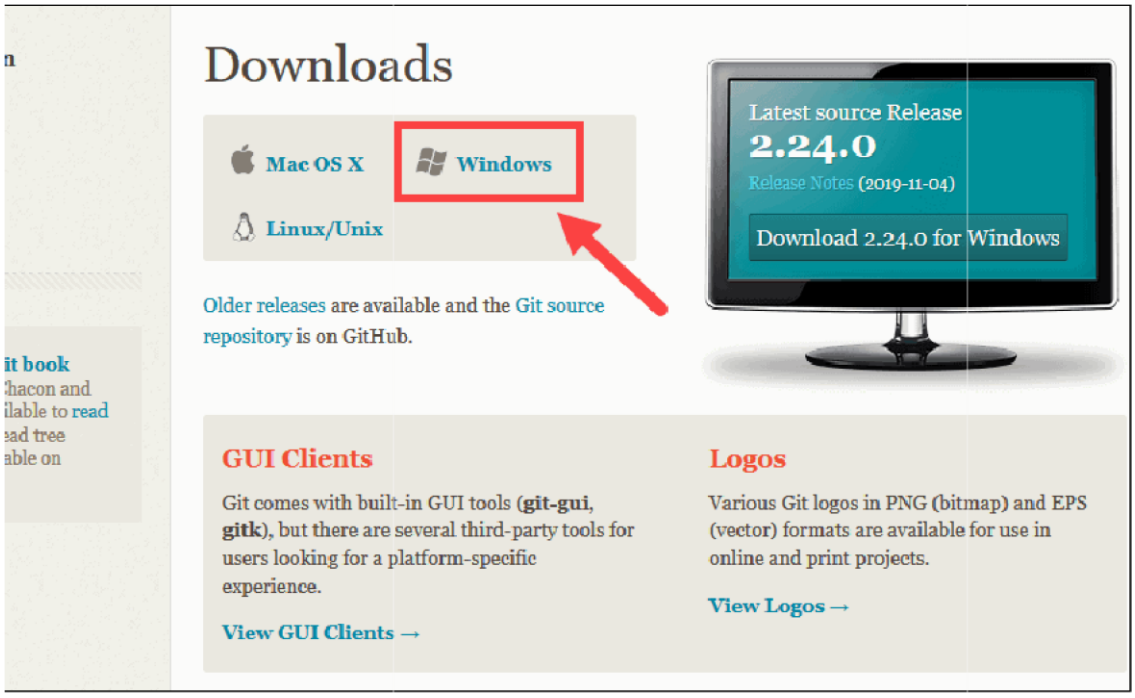
Theory:

Post Lab Question:
Explain SDLC in details with Diagram
ExplainDifferencebetweenGit,Github,GithubAction,GitLab..(Anythree)

Practical:
Prerequisites

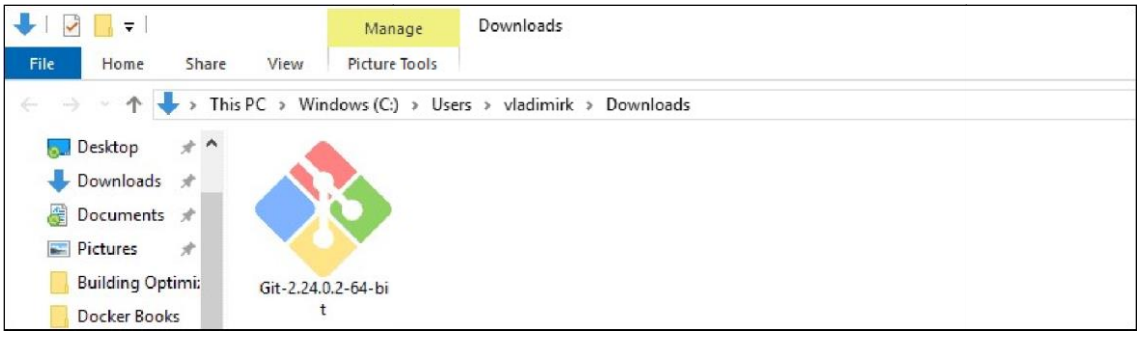
- Create Username and password for the Github website (https://github.com/login)

Download Git for Windows
1. BrowsetotheofficialGitwebsite: https://git-scm.com/downloads
2. ClickthedownloadlinkforWindowsandallowthedownloadto complete.

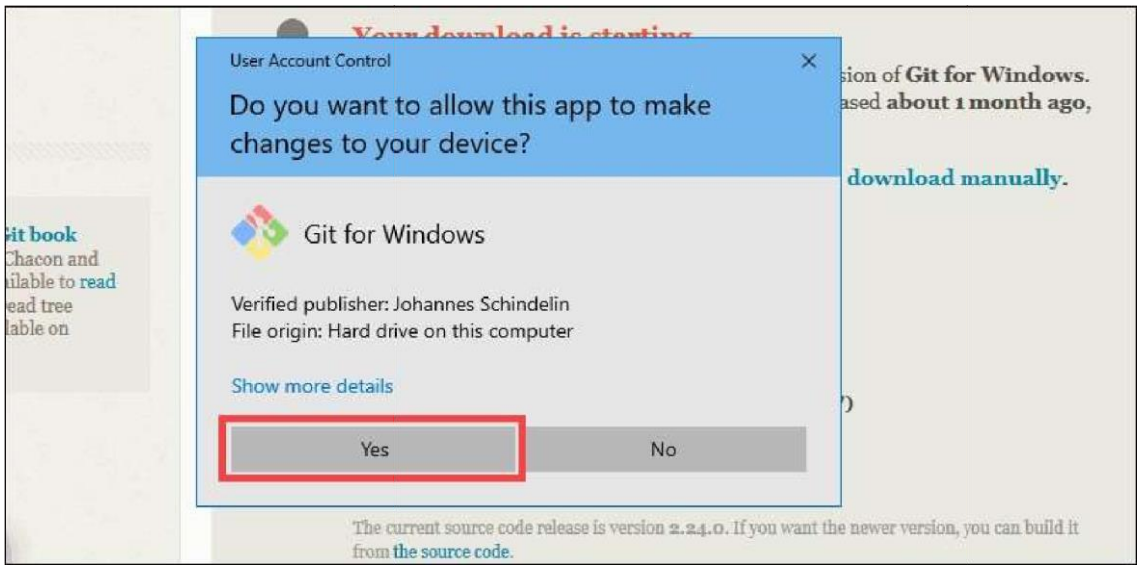


Extract and Launch Git Installer

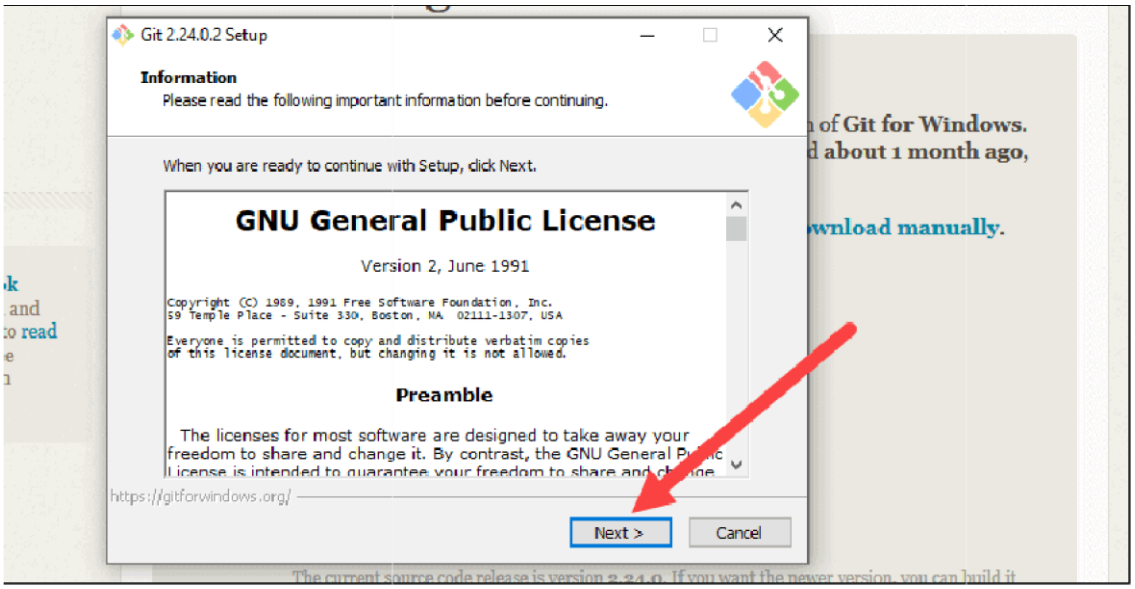
3. Browsetothedownloadlocation(orusethedownloadshortcutinyourbrowser). Double-click the file to extract and launch the installer.



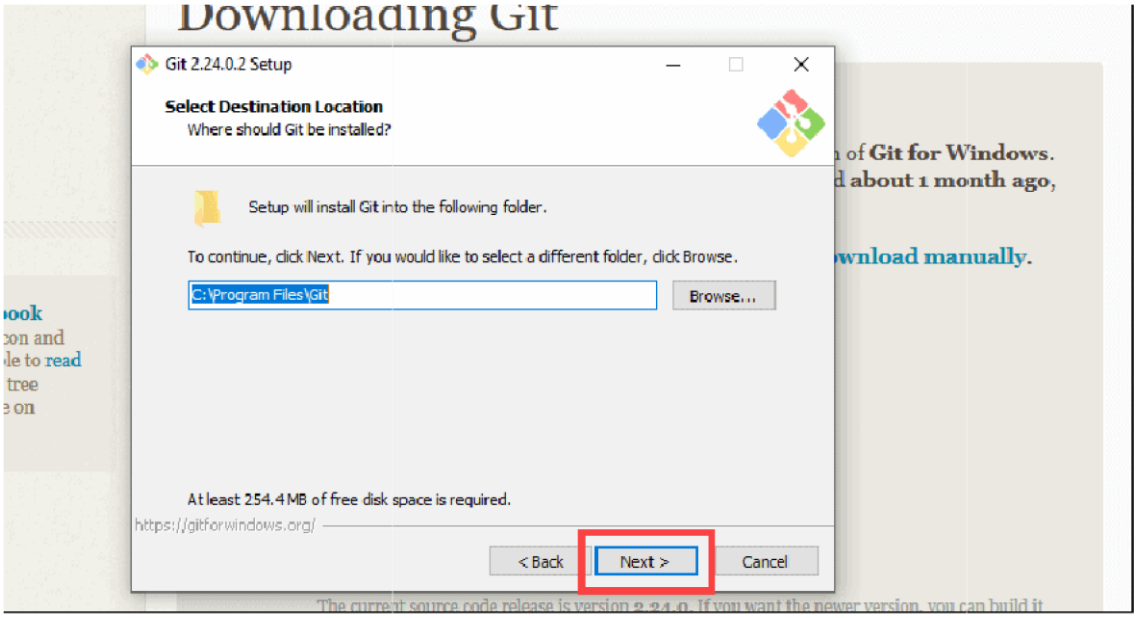
4. Allowtheapptomakechangestoyourdevicebyclicking YesontheUserAccount Control dialog that opens.



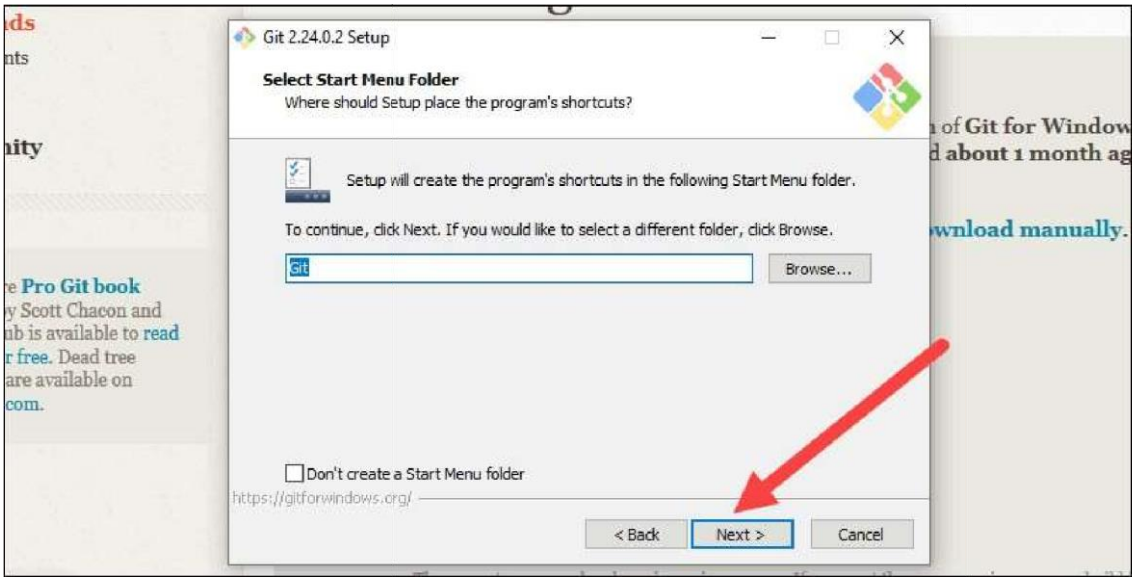
5. Review the GNU General Public License, and when you're ready to install, click Next.



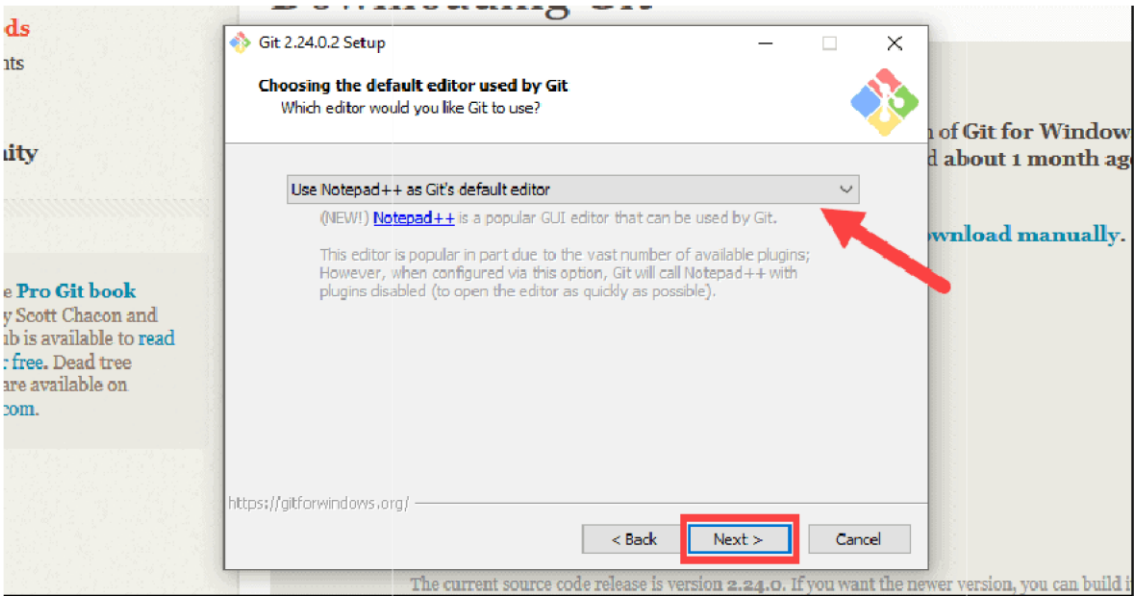
6. Theinstallerwillaskyoufor an installation location. Leave the default, unless you have reason to change it, and click Next.



7. Acomponentselectionscreenwillappear.Leavethe defaults unless you have a specific need to change them and click Next.

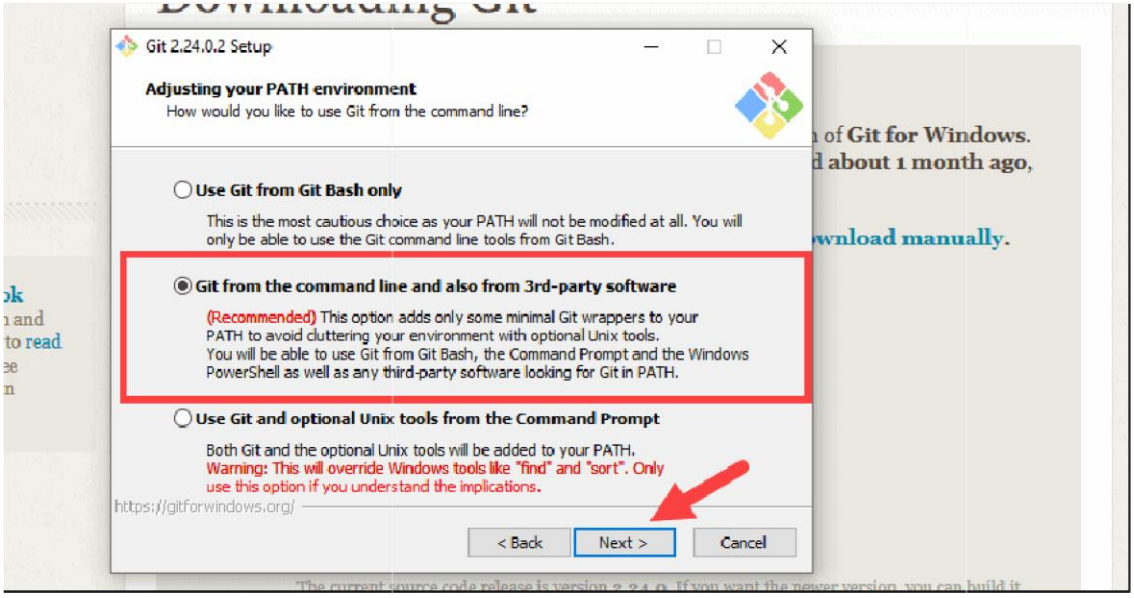


9. Selectatexteditoryou'dliketousewithGit.Usethedrop-downmenutoselect Notepad++ (or whichever text editor you prefer) and click Next.



10. The next stepallowsyoutochooseadifferentnameforyourinitialbranch.Thedefaultis 'master.' Unless you're working in a team that requires a different name, leave the default option and click Next.

11. This installation step allows you to change the PATH environment. The PATH is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click Next.

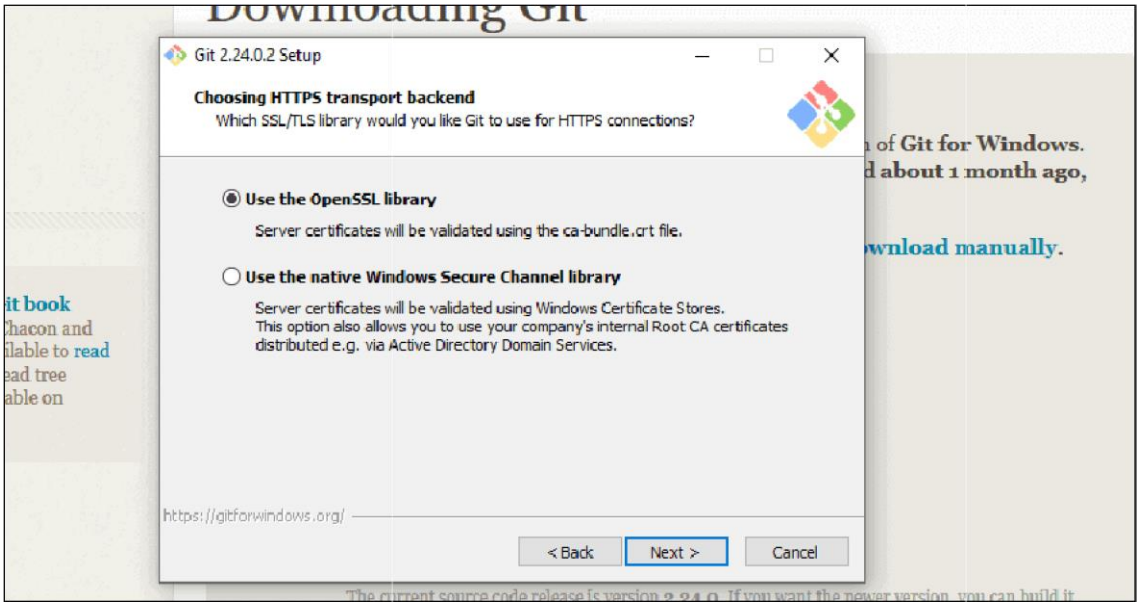


ServerCertificates,LineEndingsandTerminal Emulators

12. The installer now asks which SSH client you want Git to use.Gitalreadycomeswithits own SSH client, so if you don't need a specific one, leave the default option and click Next.

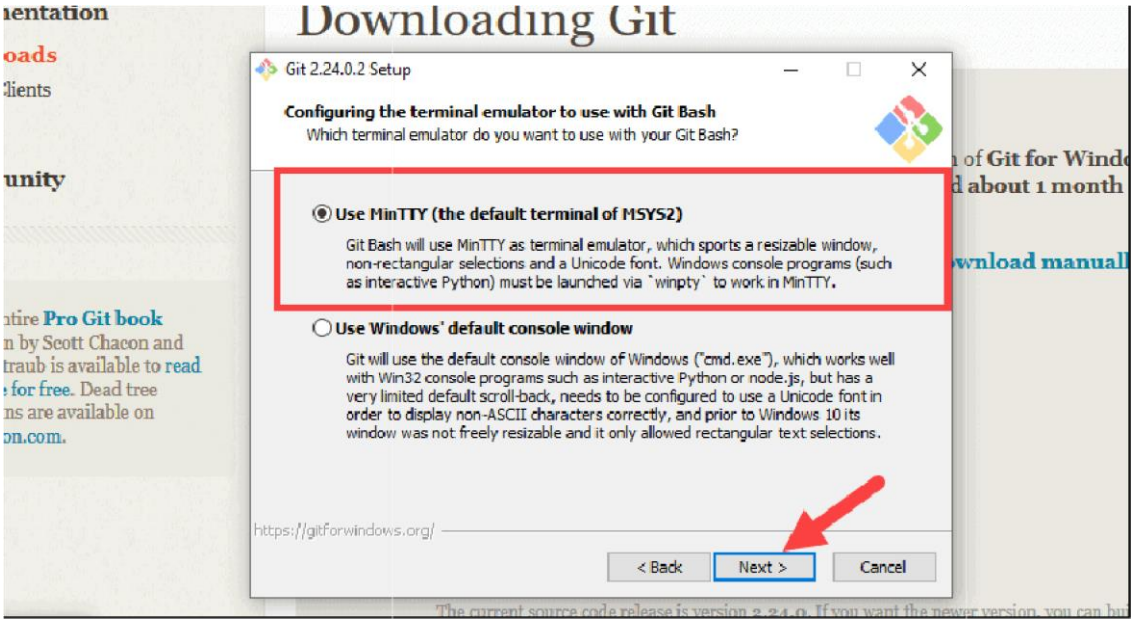
Note:CheckoutourcomparisonofSSHandHTTPS for Git and which one you should use.

13. The next option relates to server certificates. Most users shouldusethedefault.Ifyou're working in an Active Directory environment, you may need to switch to Windows Store certificates. Click Next.



14. The next selection converts line endings. It is recommended that you leave the default selection. This relates to the way data is formatted and changing this option may cause problems. Click Next.

15. Choose the terminal emulatoryouwanttouse.ThedefaultMinTTYisrecommended,for its features. Click Next.



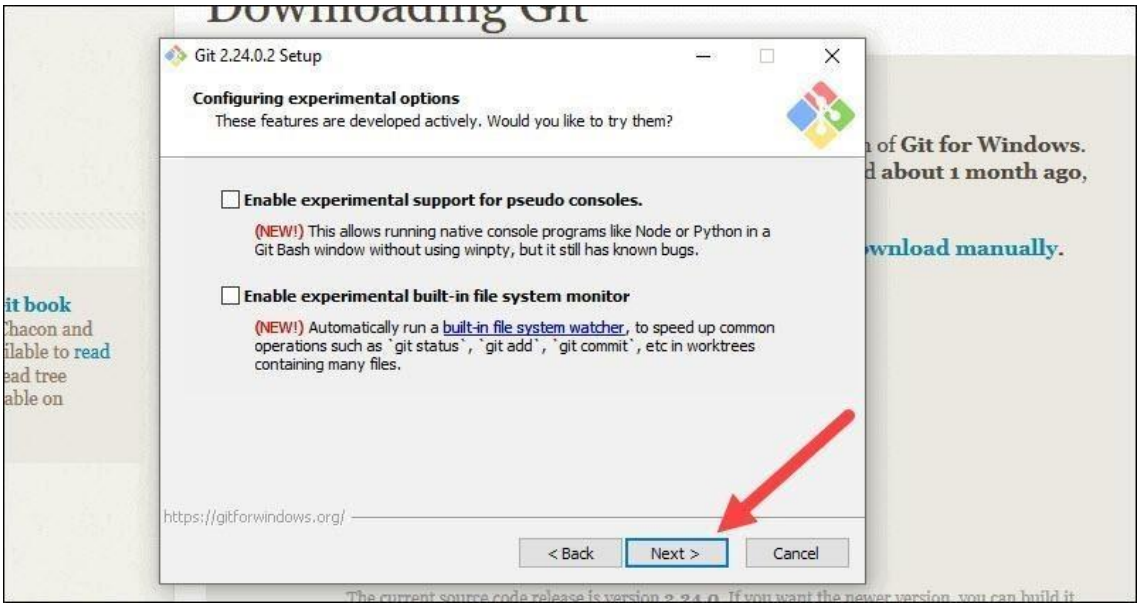
16. The installer now asks what the git pull command should do. The default option is recommended unless you specifically need to change its behavior. Click Next to continue with the installation.

17. Next you should choose which credential helper to use. Git uses credential helpers to fetch or save credentials. Leave the default option as it is the most stable one, and click Next.

Additional Customization Options

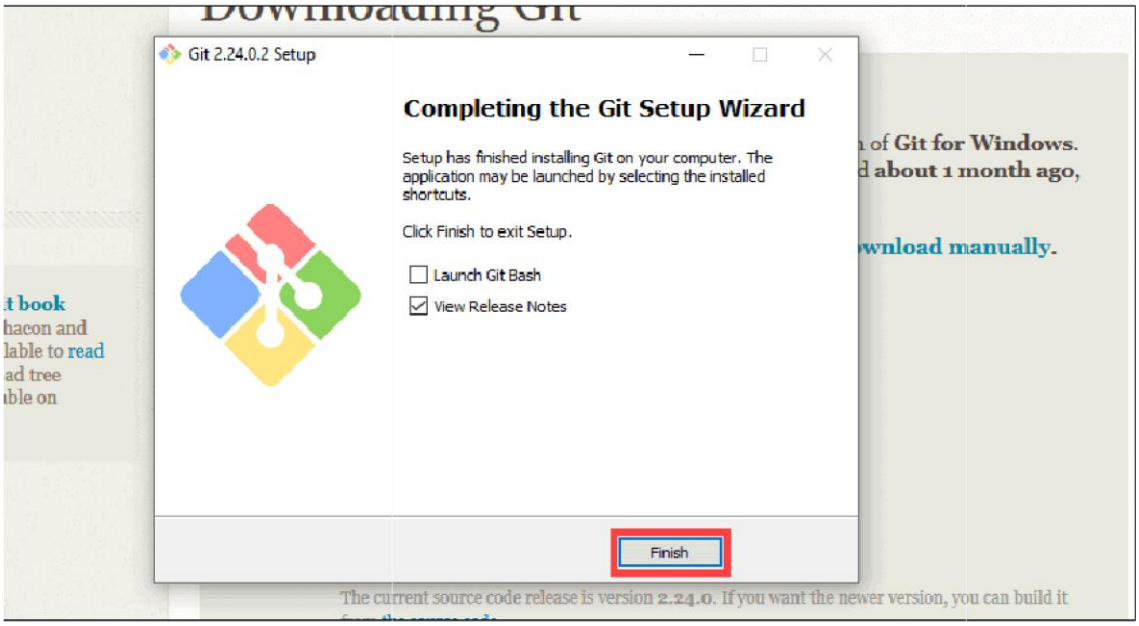
18. Thedefaultoptionsarerecommended,howeverthisstepallowsyoutodecidewhichextra option you would like to enable. If you use symbolic links, which are like shortcuts for the command line, tick the box. Click Next.

19. Depending on the version of Git you're installing, it may offer to install experimental features. At the time this article was written, the options to include support for pseudo controls and a built-in file system monitor were offered. Unlessyouarefeelingadventurous, leave them unchecked and click Install.



Complete Git Installation Process

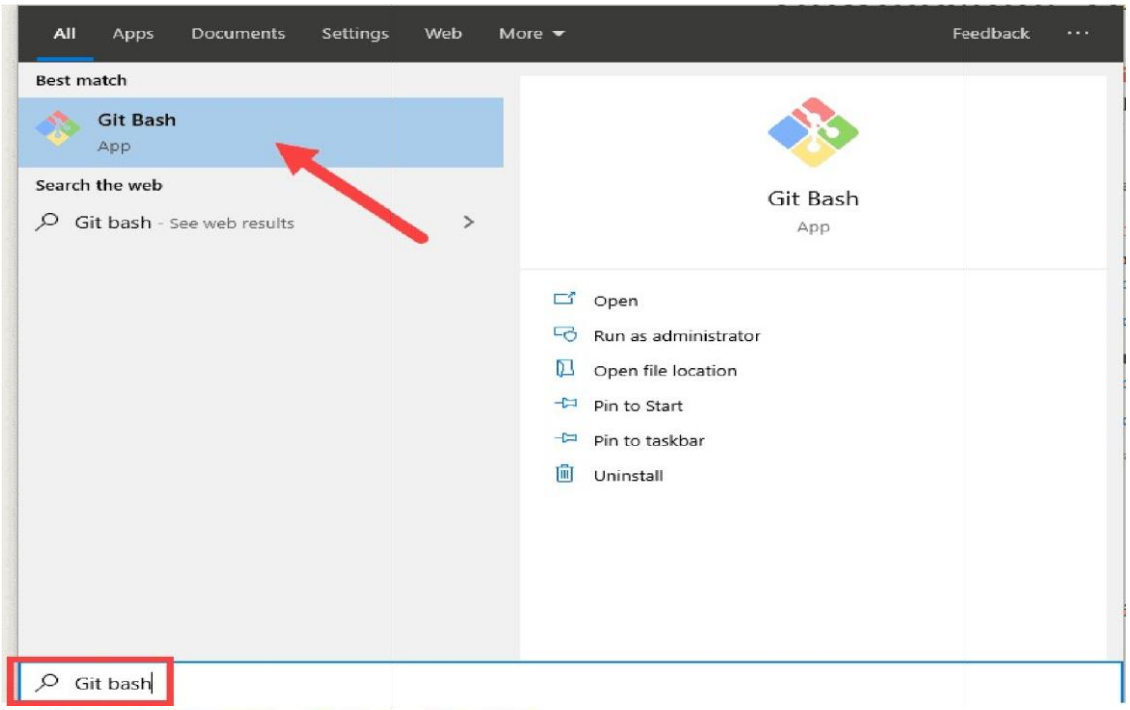
20. Once the installation is complete, tick theboxestoviewtheReleaseNotesorLaunchGit Bash, then click Finish.



Note:LearnthedifferencesbetweenCLIandGUI. How to

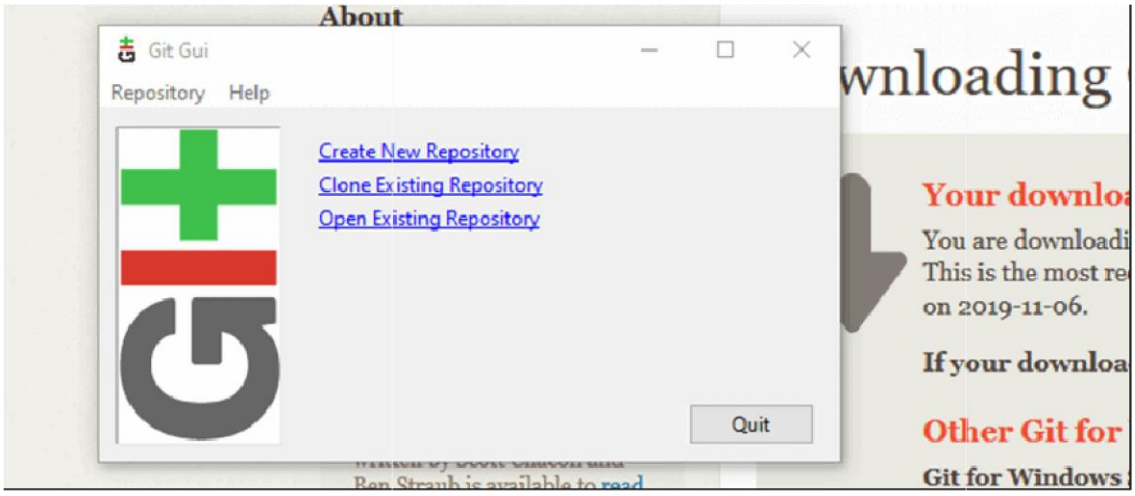
Launch Git in Windows
Githastwomodesofuse–abashscriptingshell(orcommandline)andagraphical user interface (GUI).

Launch Git Bash Shell
To launch Git Bash open the WindowsStartmenu,typegitbashandpressEnter(orclickthe application icon).



Launch Git GUI

TolaunchGit GUI open the Windows Start menu, type git gui and press Enter (or click the application icon).



Connecting to a Remote Repository

YouneedaGitHubusernameandpasswordforthisnext step.

Conclusion:hencewepracticallystudytheinstallationofGitandcreatinggithub accounts.

Aditi Ambasta
60018220113
S004 - AI&DS

Devops Exp-2

1. Explain SDLC in detail with diagram.

→ SDLC i.e Software Development Life Cycle is a process that defines various stages involved in the development of software for delivering a high quality product. SDLC covers the complete life cycle of a software i.e from inception to retirement.

SDLC Phases:

(i) Requirement gathering & analysis :- All relevant info is collected from customer to develop a product as per their exception.

(ii) Design :- Requirement gathered in SRS document is used as an input & software architecture that is used for implementing the system development is observed.

(iii) Implementation or Coding :- Starts once the developer gets the design document. All the components of software are implemented in this phase.

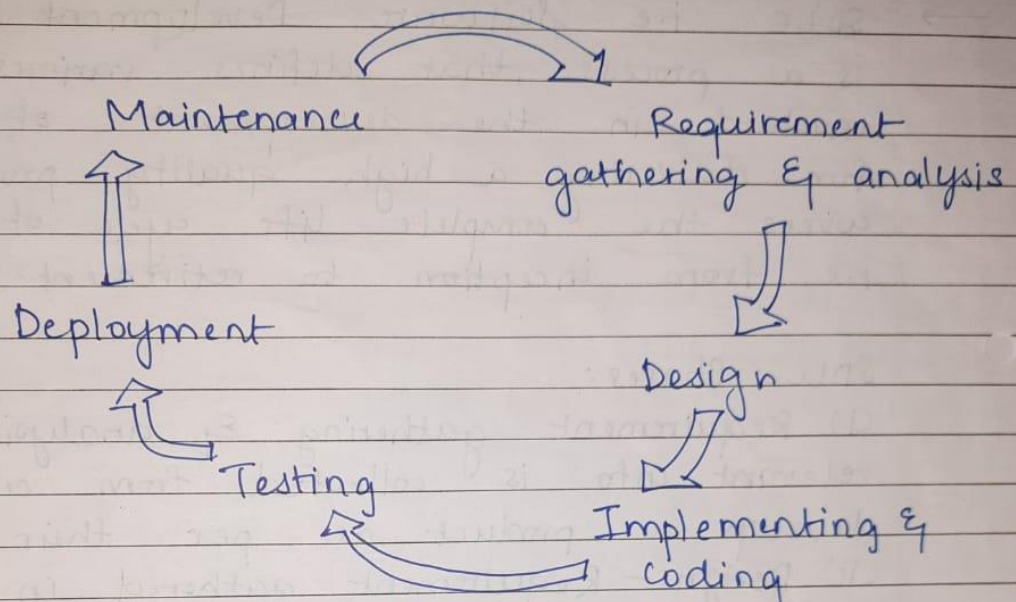
(iv) Testing :- The developer software is tested thoroughly & any defects found are assigned to developers to fix them.

(v) Deployment :- After testing, it is deployed in the production environment or UAT (User acceptance Testing) is done depending on customer expectation.

(vi) Maintenance :- Maintainance of the product is done i.e if any issue comes up it

23/9/24 (23)

is said to be taken case by the developers.



2. Difference between Git, Github & Gitlab.

→

	Git	GitHub	Gitlab
(i)	Version control syst	Cloud Based git repo hosting service	Complete Devops Platform
(ii)	Source code management	Repository hosting	Repo management
(iii)	local or remote repository	collaboration, CI/CD Cloud based repository.	CI/CD, Devops tools. Self hosted repository.
(iv)	Basic branching & merging	Pull requests, code reviews	Merge requests.
(v)	Command line interface.	Web Based GUI.	Web based GUI.
(vi)	Issue tracking not included.	Integrated issue tracking	Integrated issue tracking with boards.

Conclusion: Thus, we studied about SDLC phases & understand difference b/w GIT, Github & Gitlab.

EXPERIMENT NO. 3

Name: ADITI AMBASTA **Roll: S004** **SAP:60018220113**

Brach: AI & DS **Subject: Devops Laboratory (DJS22ADL5014)** **Sem: V**

Aim: Perform various GIT operations on local and Remote repositories

Practical:

- Step 1:
  To launch Git Bash open the Windows Start menu, type git bash and press Enter (or click the application icon).

- Step 2:
  Check the Git version: $ git --version

- Step 3:
  For any help, use the following command: $ git help config
  provides a manual from the help page for the command just following it (here, it's config).

  Another way to use the same command is as follows: $ git config --help

- Step 4:
  Create a local directory using the following command: $ mkdir AIDS
  $ cd AIDS

- Step 5:
  The next step is to initialize the directory: $ git init

- Step 6:
  Go to the folder where "AIDS" is created and upload exp.no.1's login page "Loginpage.html"

- Step 7:
  Enter the Git bash interface and type in the following command to check the status:
  $ git status

- Step 8:
  Add the "loginpage" to the current directory using the following command:
  $ git add login.html

- Step 9: (first time user tobe followed step.10)
  Next, make a commit using the following command:
  $ git commit -m "committing a text file"

- Step 10: if a github account is not available then create a new one.. Link the Git to a Github Account:

- git config --global user.name "github_username"

- git config --global user.email "email_address"

- Step 11:
  Open your Github account and create a new repository with the name "AIDS_demo" or any name and click on "Create repository." This is the remote repository. Next, copy the link of "URL"

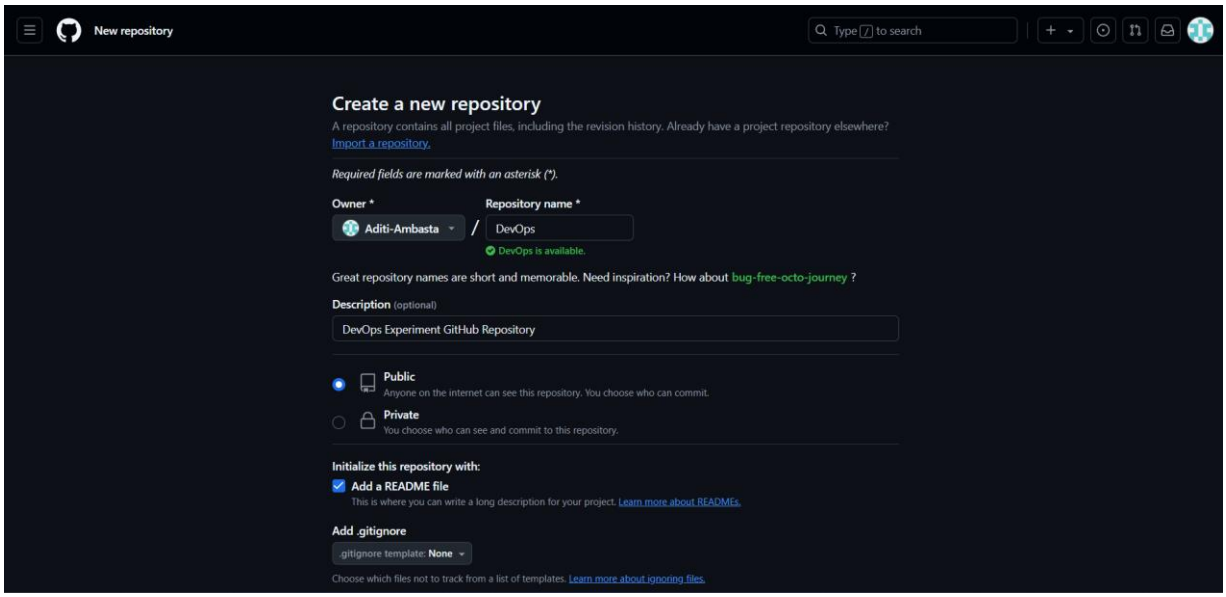
- Step 12:
  Go back to Git bash and link the remote and local repository using the following command:

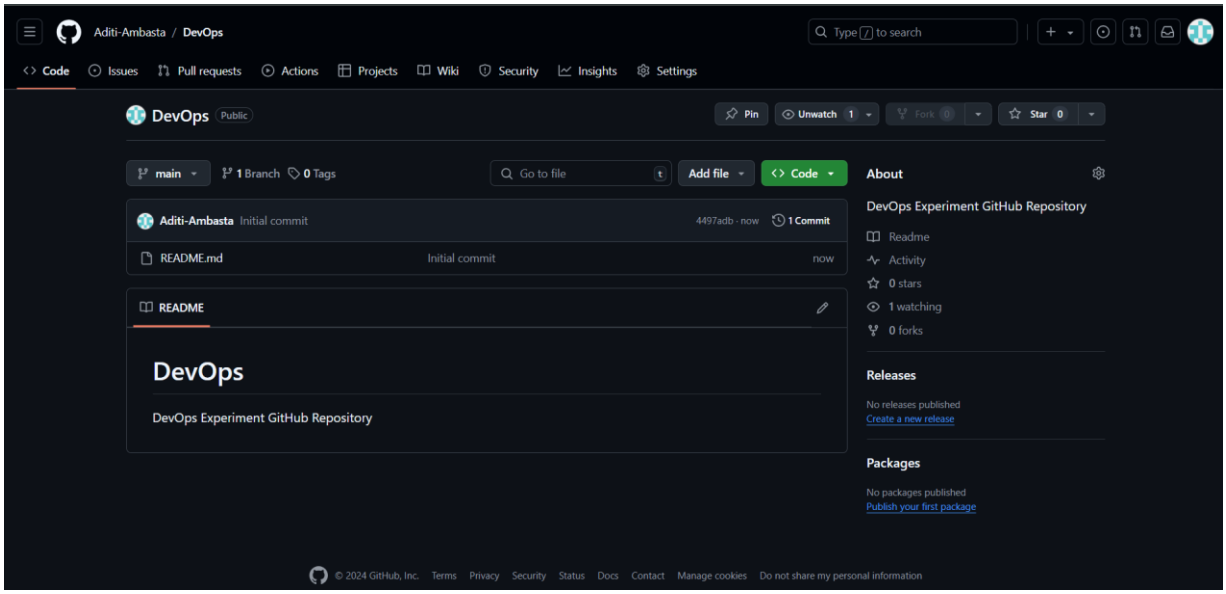
  $ git remote add origin <URL>

- Step 13: Push the local file onto the remote repository using the following command:

  $ git push origin master

- Step 14: Move back to Github and click on "AIDS_demo" and check if the local file "Login.html" is pushed to this repository





Conclusion: Hence we practically studied the installation of Git and creating github accounts.

Aditi Ambaste
60018220113
S004 - AI&DS

Devops Exp-03

Aim : To perform various GIT operations on local & remote repositories.

Theory :

Q. Explain the following :

1) Git Add :

Purpose - This command adds changes from your working directory to the staging area (also known as the index). It prepares the files for a commit.

Usage : git add <file>, adds a specific file
git add, adds all changes in the current directory.

Eg :- git add index.html.

2) git commit

Purpose : This command records the changes in staging area to the local repo. Each commit is a snapshot of your project at a particular point in time.

Usage : git commit -m "Your message" creates a commit with a message. git commit without -m opens the default text editor to write a commit message.

3) Git push

Purpose : This command uploads your local repo

changes to a remote repository. This typically includes changing sending commits to a brand on the remote.
Usage: git push origin main pushes changes to the main branch on the engine remote.
Eg :- git push origin main

4) git pull
Purpose: This command fetches changes from a remote repository on your local merges into your local branch. It's a combination of git fetch & git merge.
Usage: git pull origin main feteches & merges changes from the main branch on the origin.
Eg :- git pull origin main

5) git done
Purpose: This command creates a copy of a remote repository on your local machine. It also automatically sets up the remote connection, so you can push & pull changes.
Usage: git done <repo url> clones the repo located at given URL.
Eg :- git clone https://github/user/repo.git

Conclusion: We learnt the commands using git which are crucial for tracking & sharing experimental changes efficiently.

23/9/2024

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEE
(Autonomous College Affiliated to the University of Mumbai)

EXPERIMENT NO. 4

Name: ADITI AMBASTA **Roll: S004** **SAP: 60018220113**
Brach: AI & DS **Subject: Devops Laboratory (DJS22ADL5014)** **Sem: V**

Aim: Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to set up a build Job.

Practical:
**Note: Install Java 11 software before installation of jenkins server.

Install Java JDK
Java JDK is required to run Hadoop, so if you haven't installed it, install it.
1.download it from for example here
2.Run the installation file and the default installation directory will be C:\Program Files\Java\jdk11.2.0\bin

Configure environment variables

1. Open the Start Menu and type in 'environment' and press enter.
2. A new window with System Properties should open up.
3. Click the Environment Variables button near the bottom right.

JAVA_HOME environment variable

1. From step 3, find the location of where you installed Java. In this example, the default directory is C:\Program Files\Java\jdk11.2.0
2. Create a new User variable with the variable name as JAVA_HOME and the value as C:\Program Files\Java\jdk11.2.0\bin

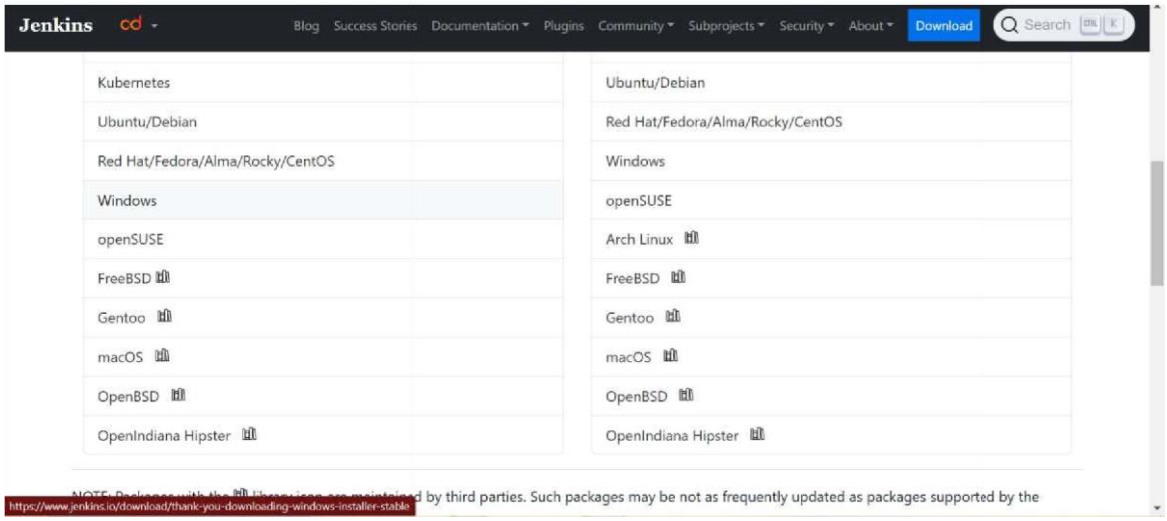
System Variable

1.Select Path →Edit→ add C:\Program Files\Java\jdk11.2.0

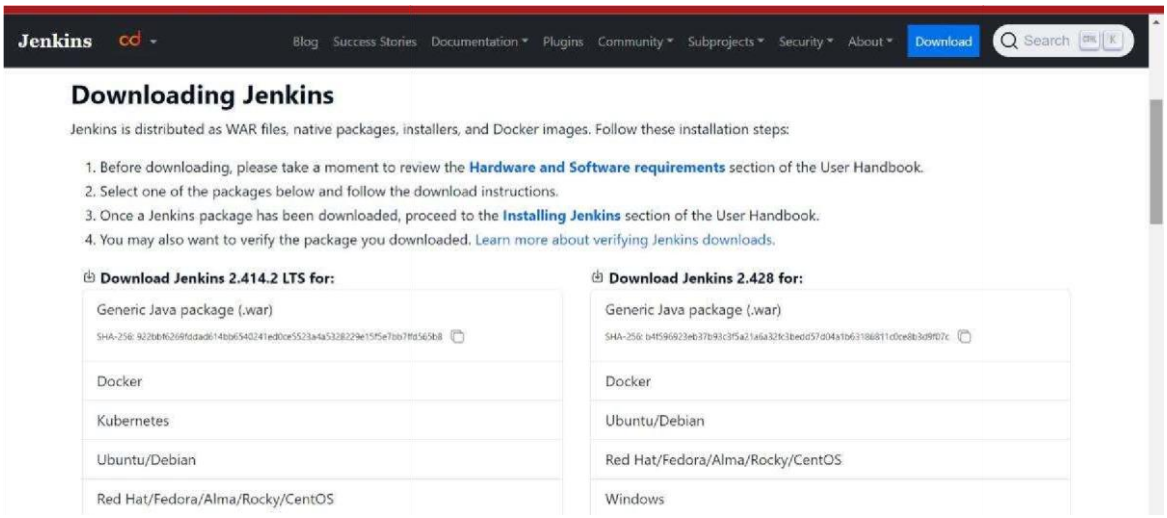
After installation, open up CMD or Powershell and confirm Java is installed:

$ java -version java
version "11.2.0"
Java(TM) SE Runtime Environment (build 11.2.0)
Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)

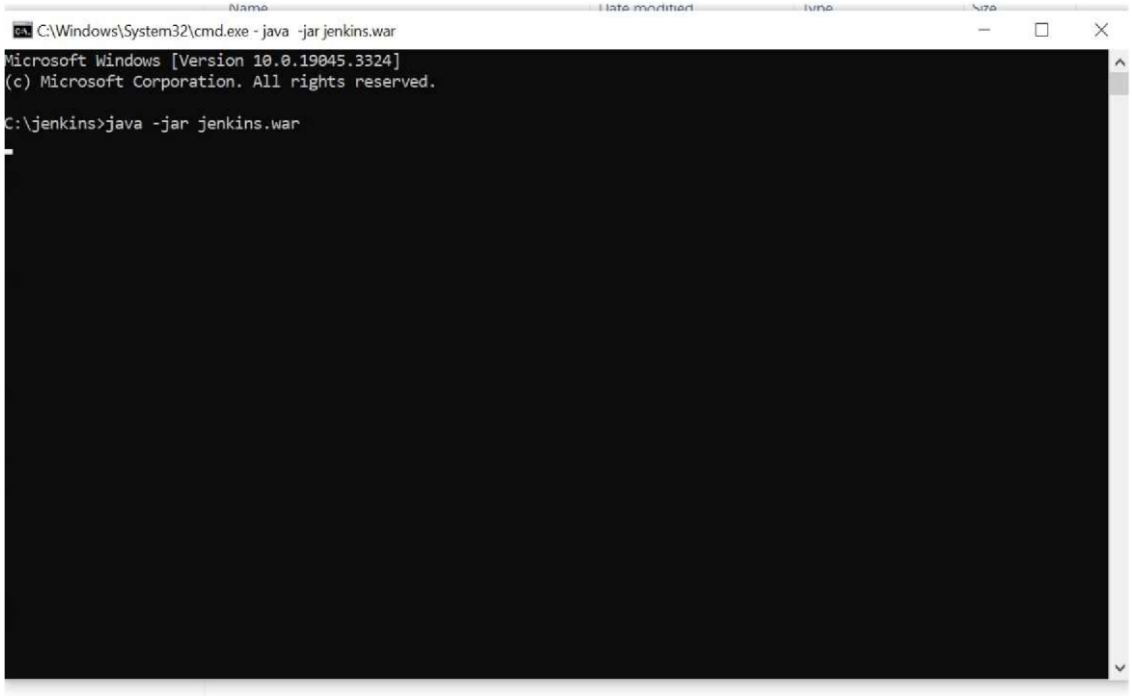
Step1:Go to the URL https://jenkins.io/download/ to download the windows version or jenkins.war file..



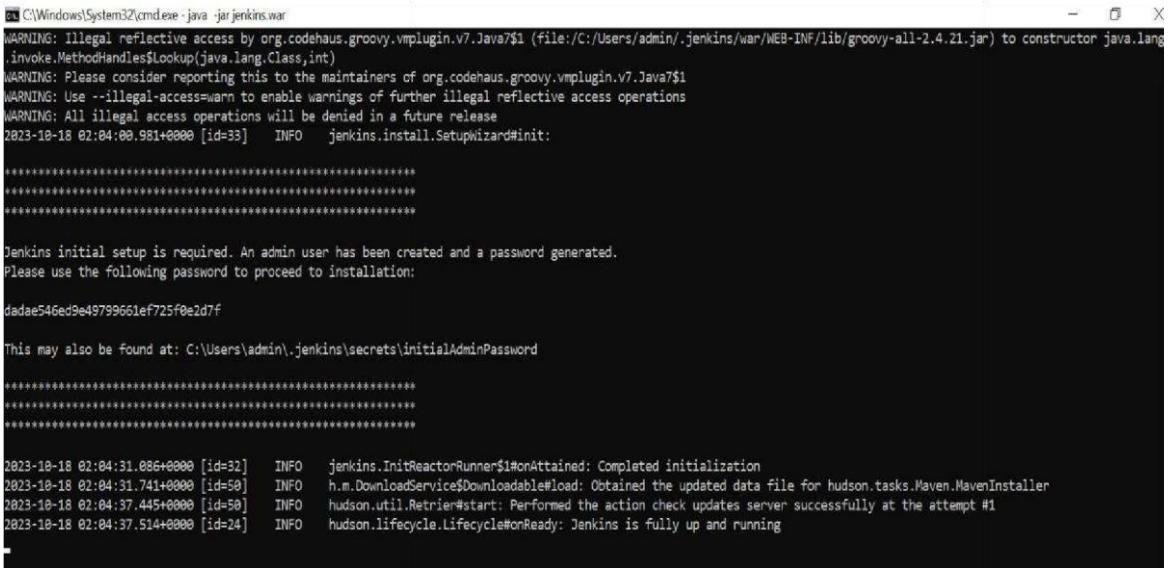
Step2:select Download jenkins 2.14.4.2 LTS for:



Step3: Go to downloaded jenkins.war file and select url enter cmd → open cmd prompt.---> enter command → java -jar jenkins.war


C:\Windows\System32\cmd.exe - java -jar jenkins.war
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.
C:\jenkins>java -jar jenkins.war

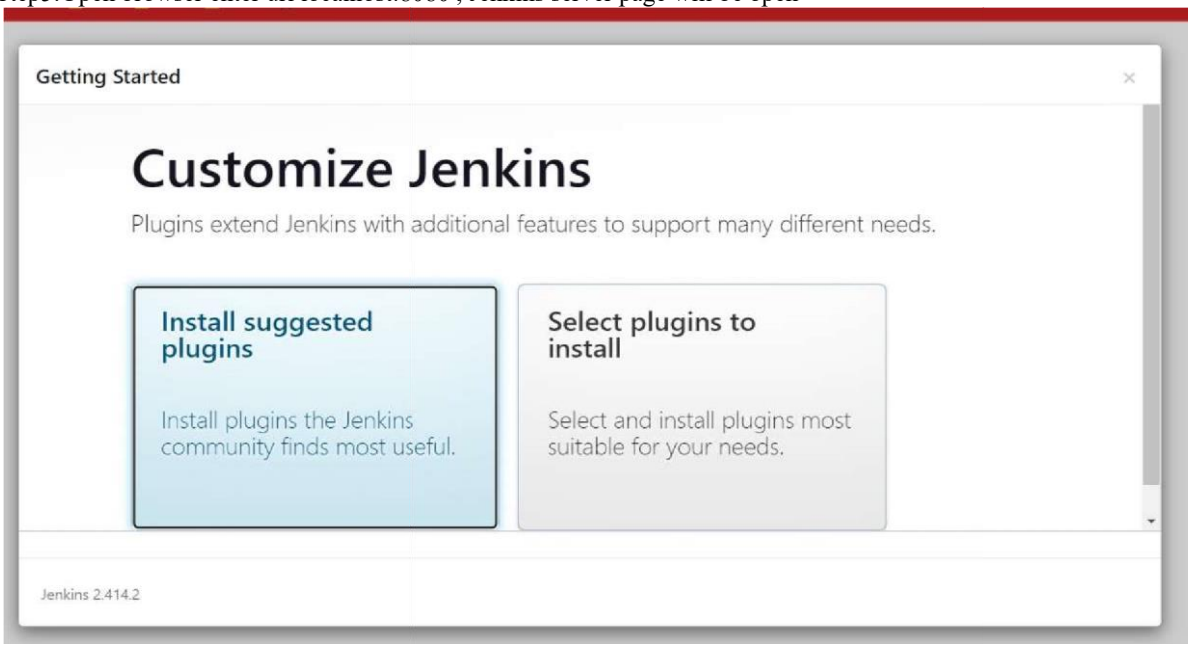
Step4: Jenkins server will start running and it will print password, note down it



Note: password will be available on follow path..

C:\Users\admin\.jenkins\secrets\initialAdminPassword

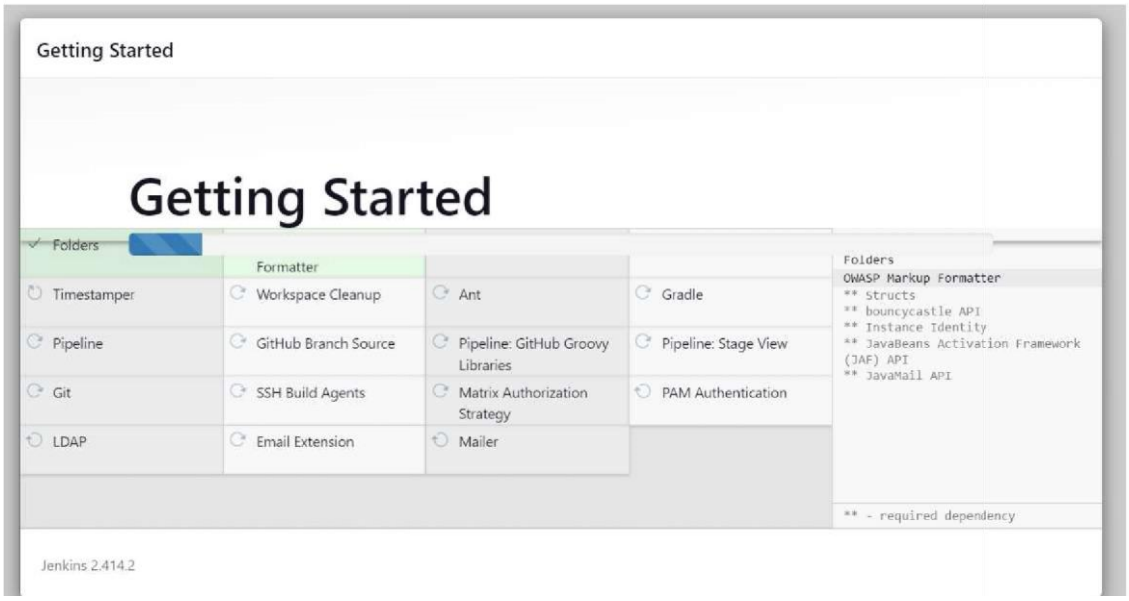
Step5:Open browser enter url localhost:8080 , Jenkins server page will be open



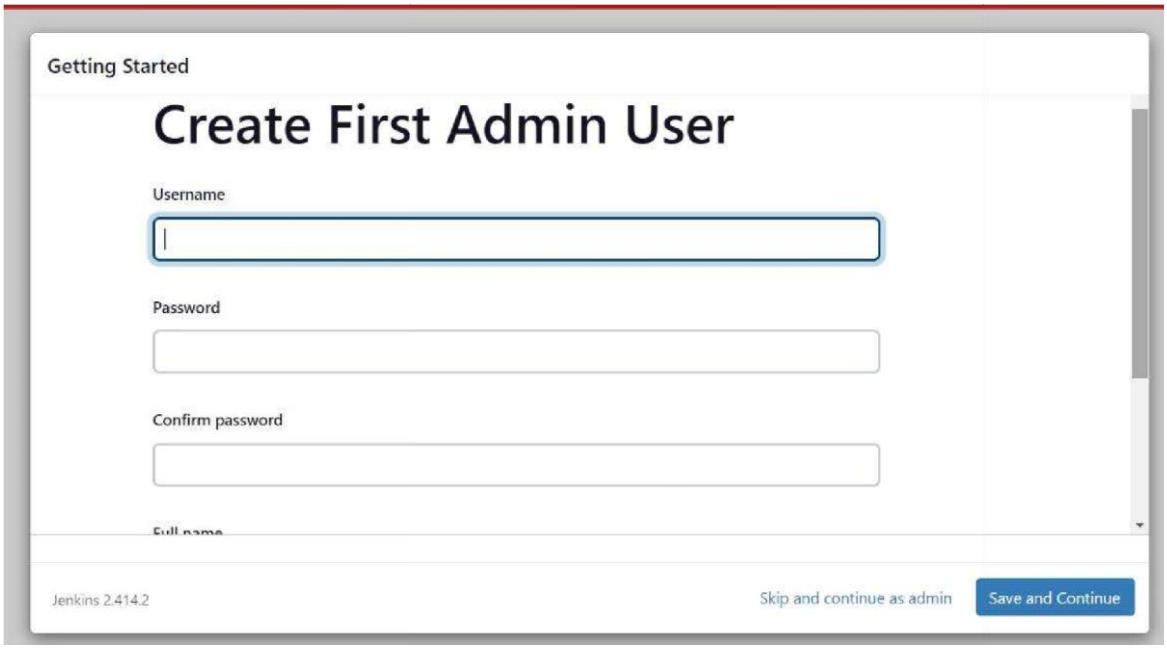
Step7:enter password which was noted down by you.

Step8:It would take time for configuration

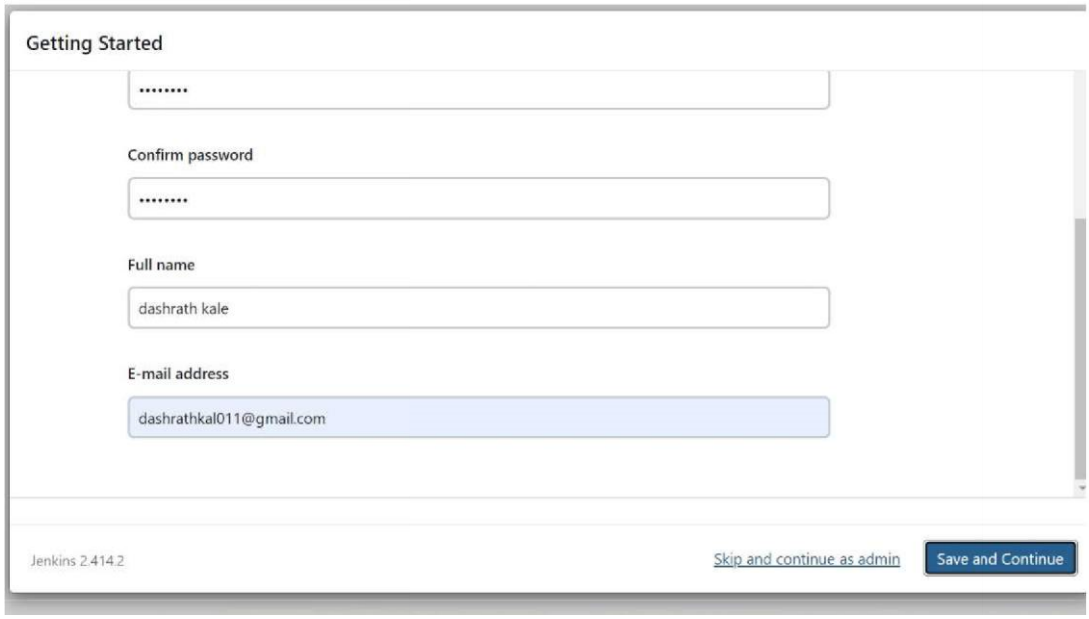
Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)



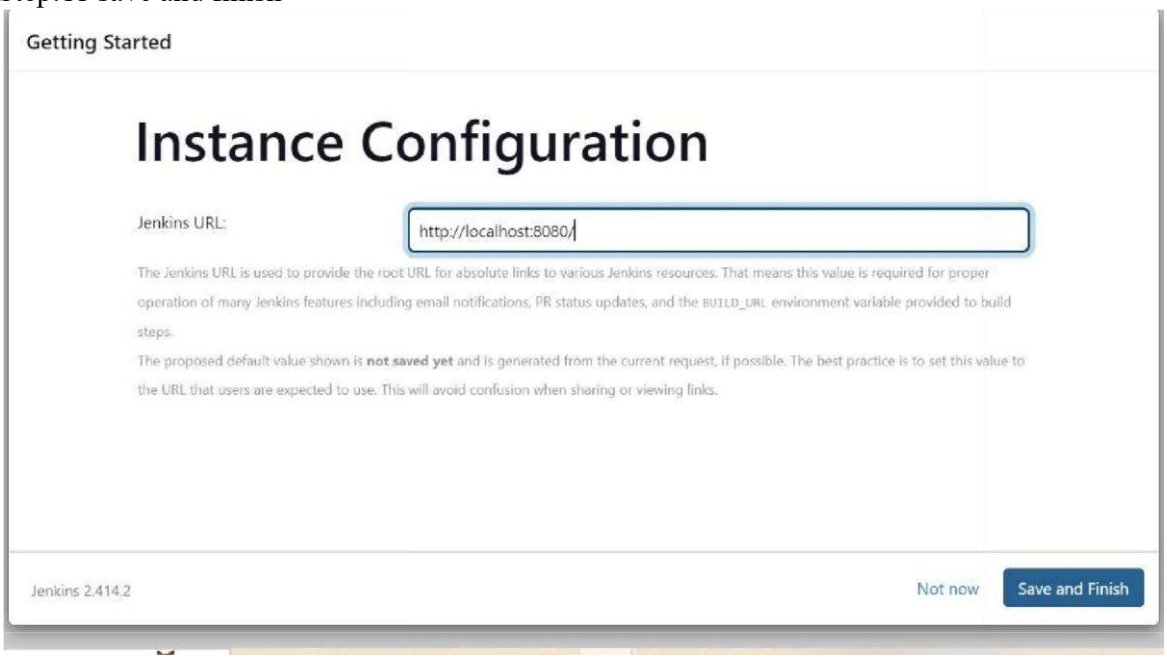
Step:9 if you want to create a user under admin you enter details otherwise default admin may get selected..


Getting Started
Create First Admin User
Username
Password
Confirm password
Full name
Jenkins 2.414.2
Skip and continue as admin
Save and Continue

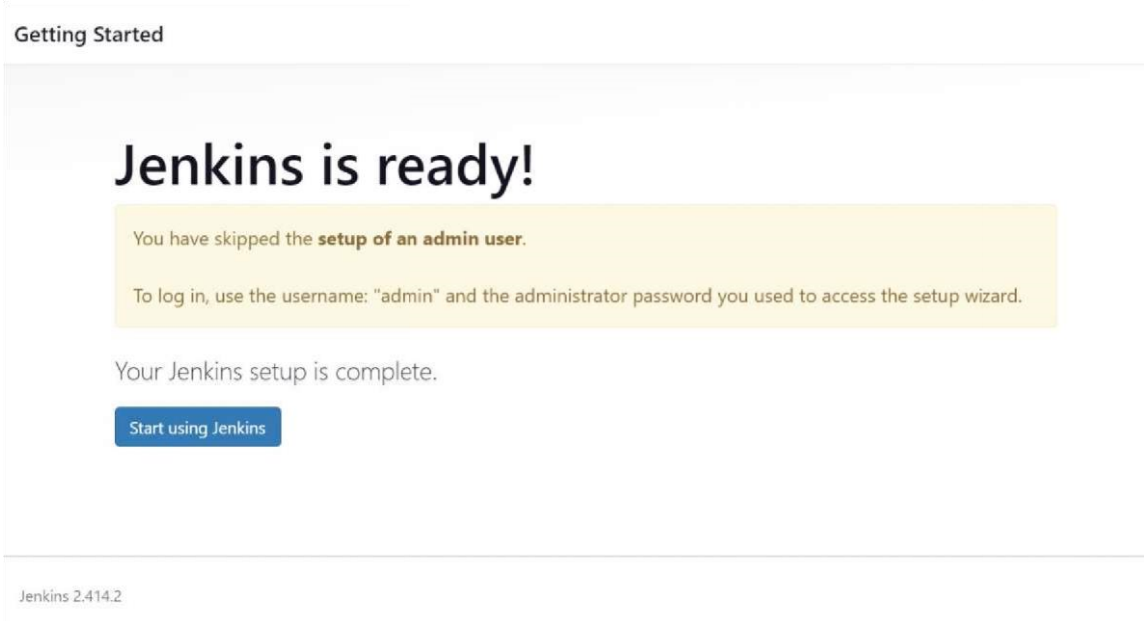
Step10:



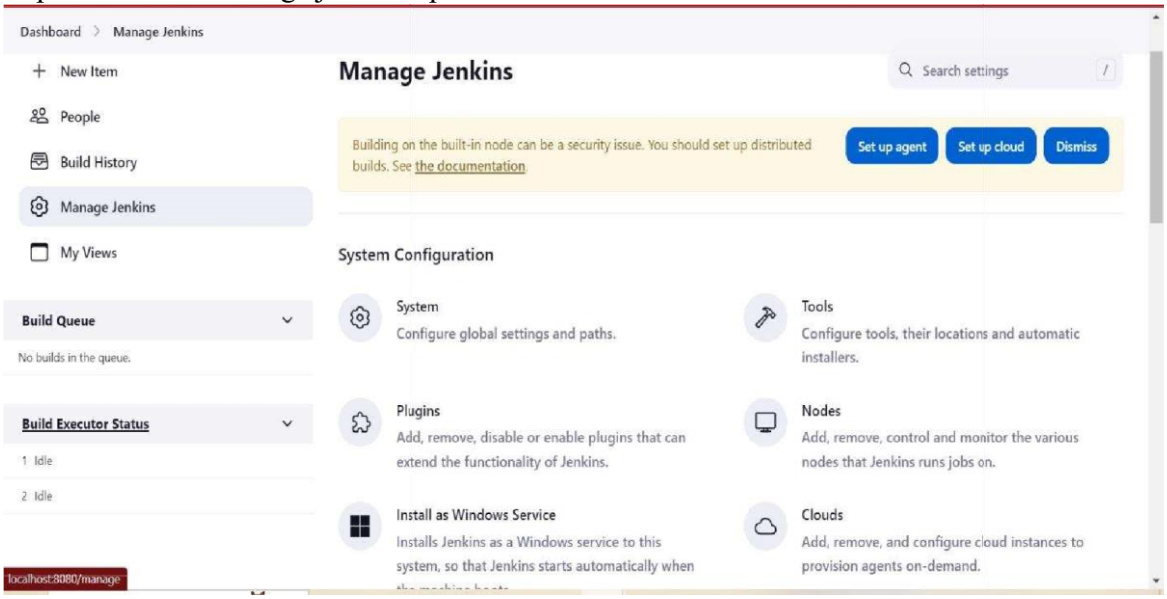
Step:11 save and finish



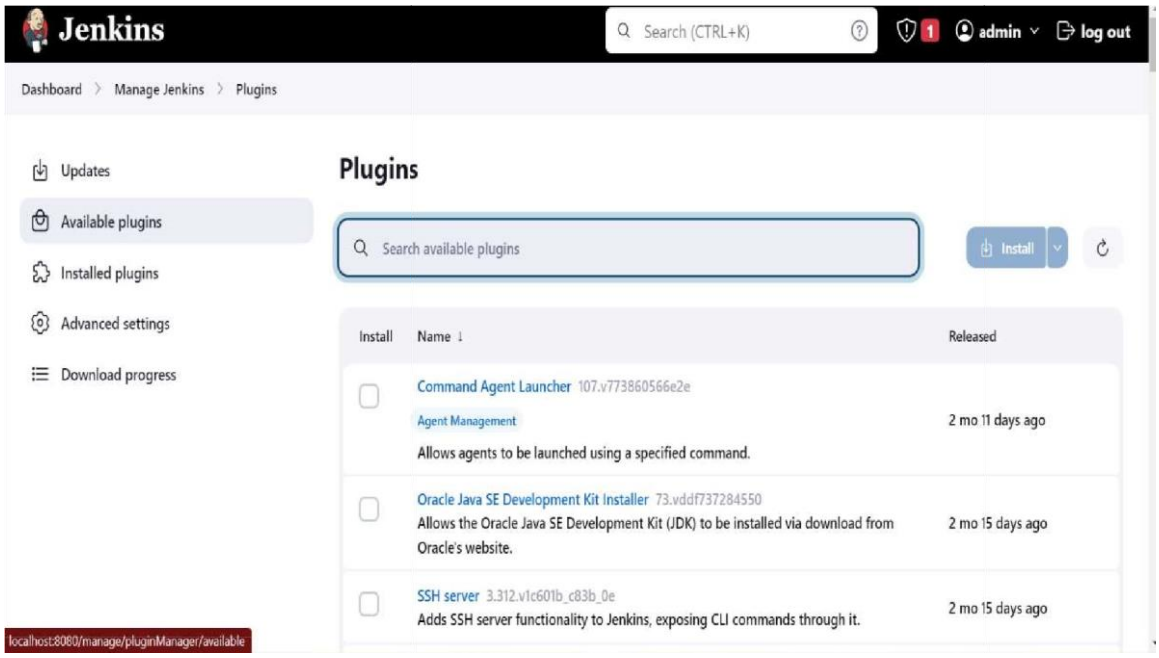
Step:12



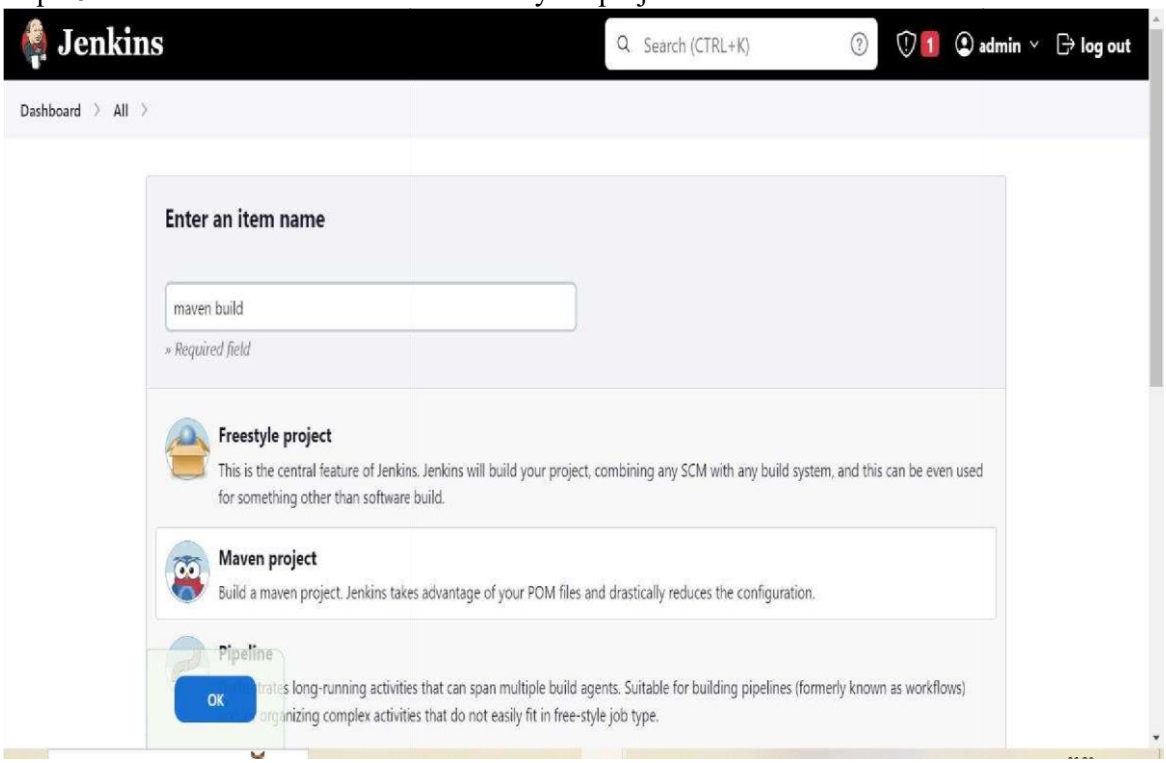
Step:13 Click on Manage jenkins option

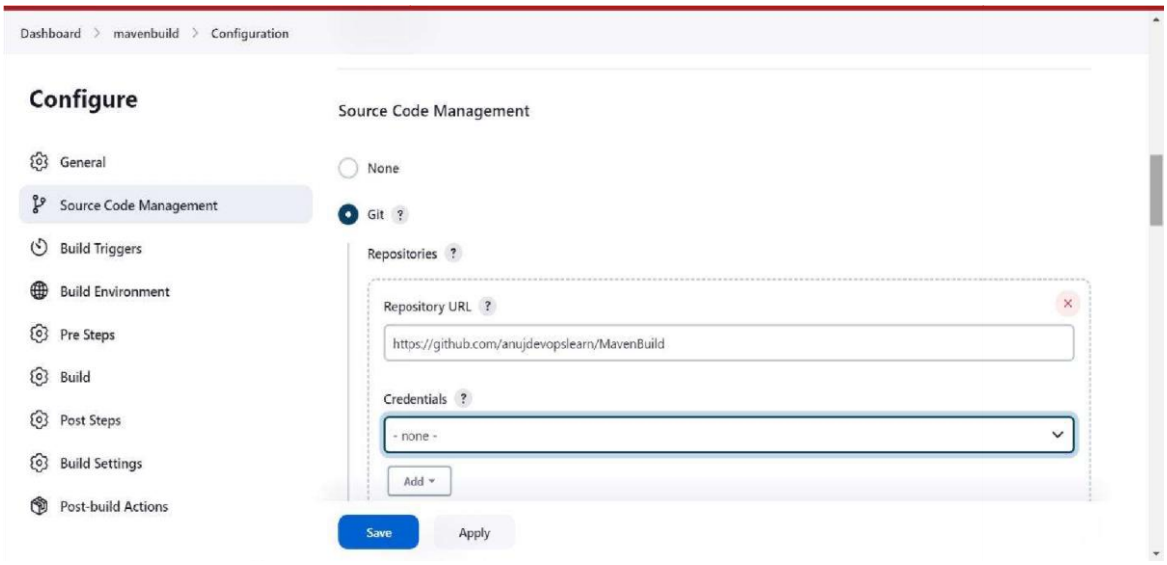


Step:14 click on available plugin search maven and git and install it…

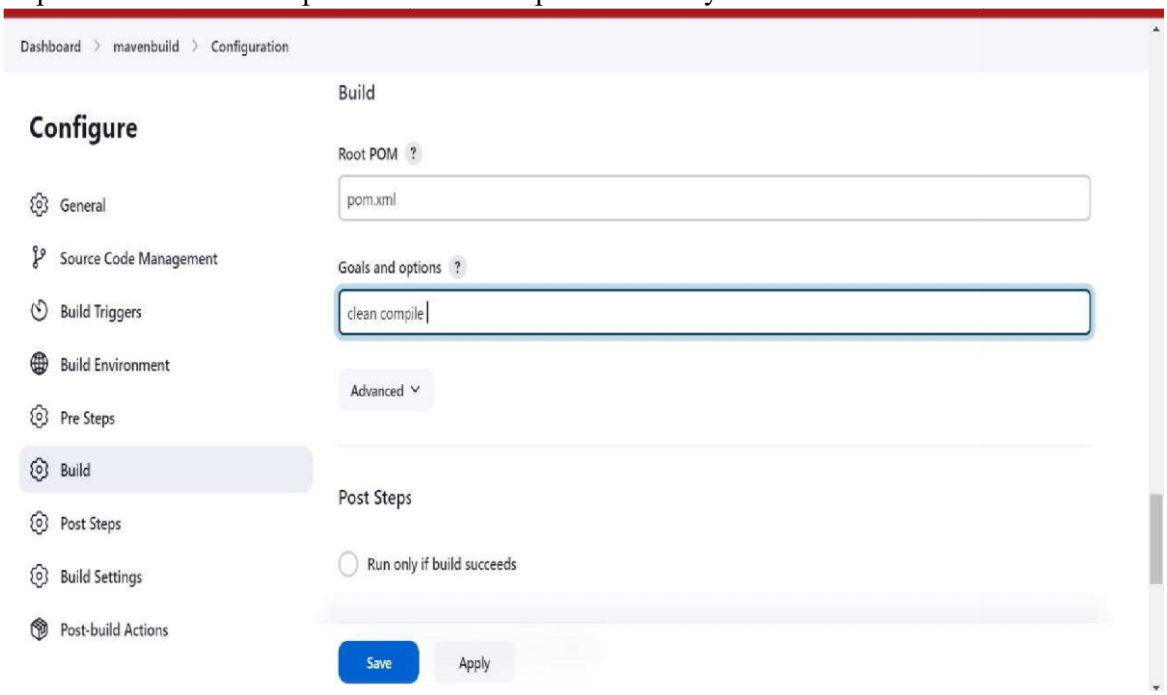


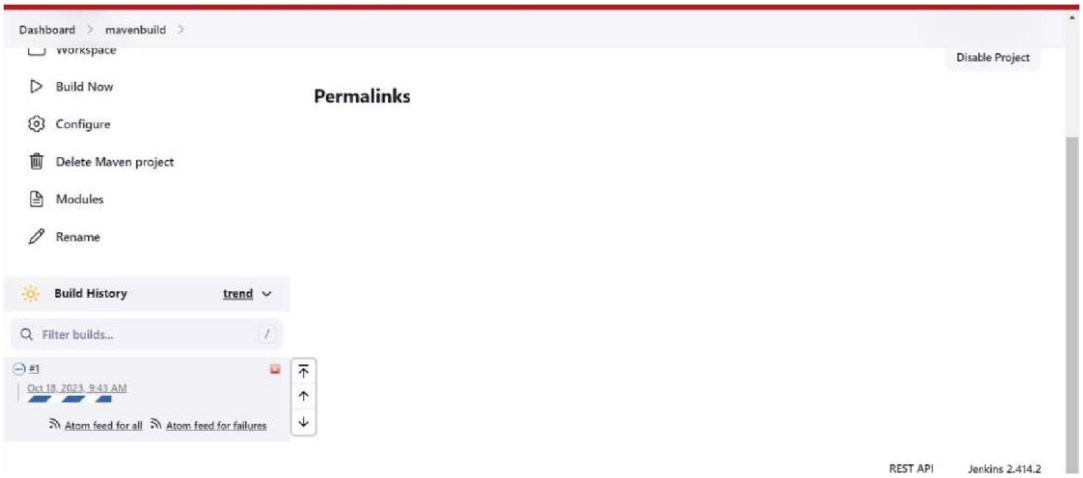
Step:15 Now click on a new item and enter your project name…





Step:18 enter Goal and option i.e clean compile test finally click on save button





Step:21 check job's build successfully executed

Conclusion : Hence we study continuous integration with help of git ,maven and build one job..

Devops Exp-4

Aim : Continous Integration

Theory :

1) Explain architecture of CI with every phrases & software tools

→ Continuous Integration is a practice in software development where developers frequently integrate their code into a shared repository, typically multiple times a day. The main goal of CI are to detect & fix integration issues early.

Architecture of Continuous Integration

(1) Build :
Tools :- Jenkins, Travis CI, Circle CI, Bamboo
Phase :- The CI Server triggers the build process after a code change is committed This step involves preparing the environment & dependencies needed to compile.

(2) Compile :
Tools :- Maven, Gradle, Ant
Phase :- Source code is compiled into executable code. This step translates the high level code into machine readable format.

③ Code Review
Tools :- Github, Gitlab
Phase :- Before or after the build, code is reviewed by peers or automated tools to ensure it adheres to coding standards.

④ Unit Testing
Tools :- JUNIT, NUNIT, Test NG
Phase :- Automated tests are run on individual components or units of the codebase to ensure they function as expected.

⑤ Integration Testing
Tools :- Selenium, JUNIT, Cucumber
Phase : After unit testing, integration tests are performed to ensure that different components of the application work together correctly.

⑥ Package (WAR, JAR, etc)
Tools : Maven, Gradle, Ant
Phase : Once the code has passed all tests, it is packaged into deployable artifacts such as WAR, JAR or other formats. These artifacts are then stored in an artifact repository for deployment.

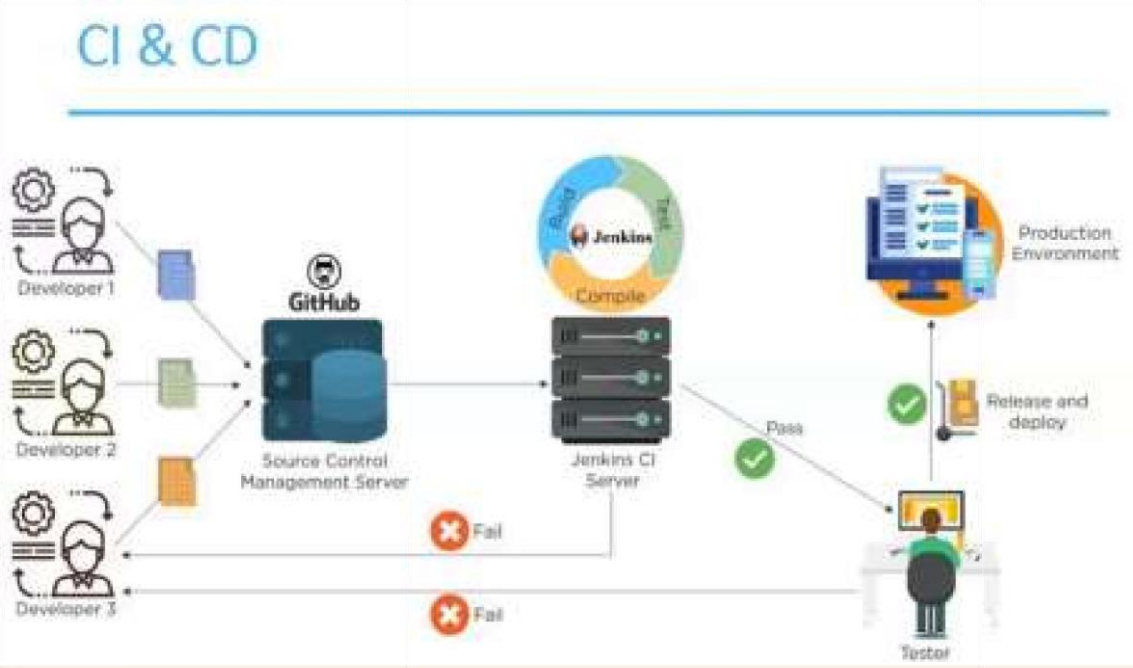
Conclusion: CI automates code integration, building, testing & deployment.

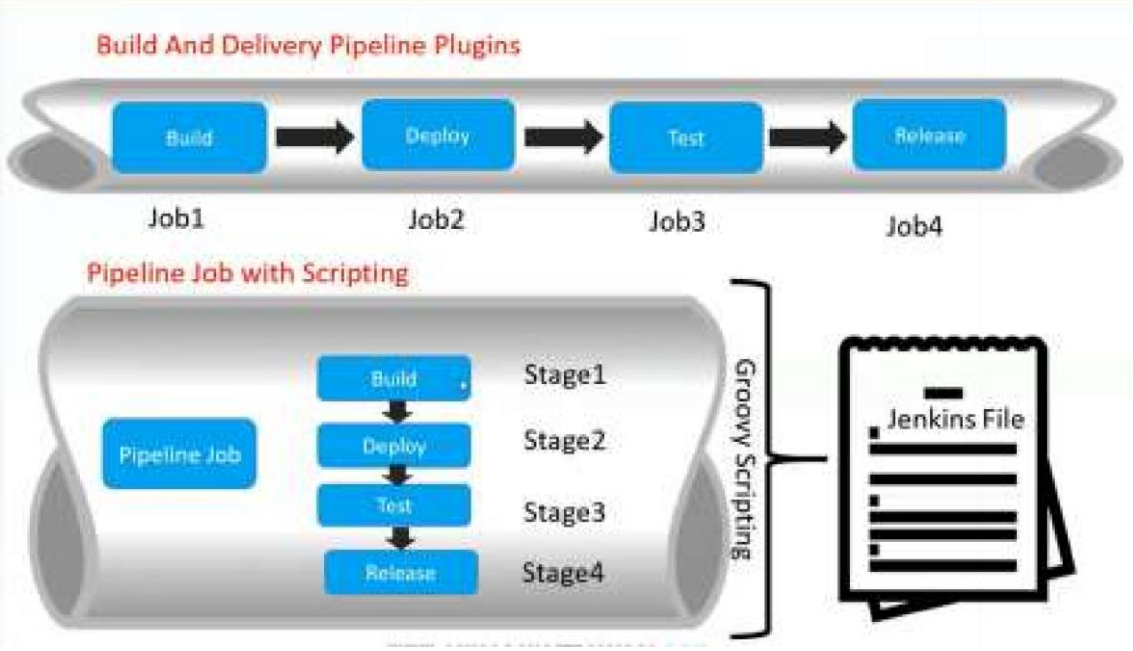
EXPERIMENT NO. 5

Name: Aditi Ambasta **Roll: S004** **SAP: 60018220113**
Brach: AI & DS **Subject: Devops Laboratory (DJS22ADL5014)** **Sem: V**

Aim: To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application.

Practical:

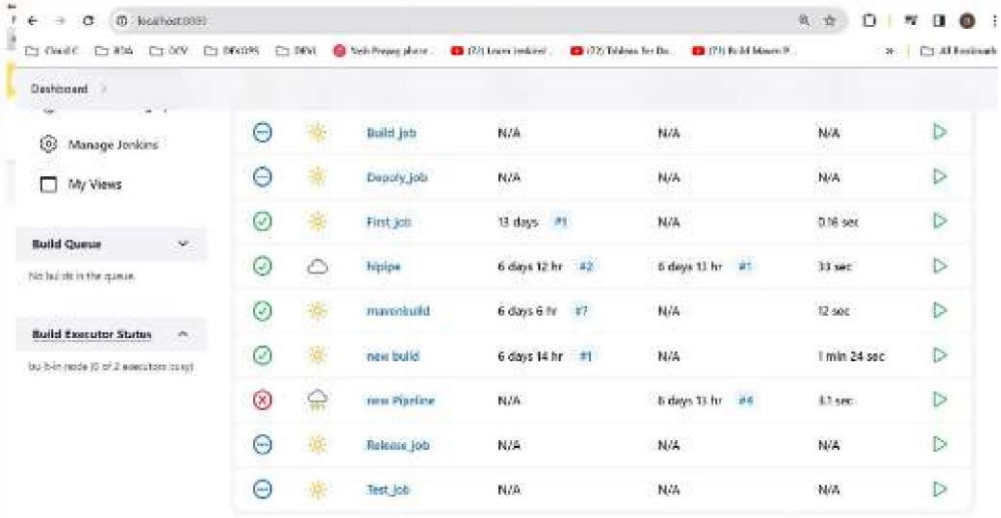




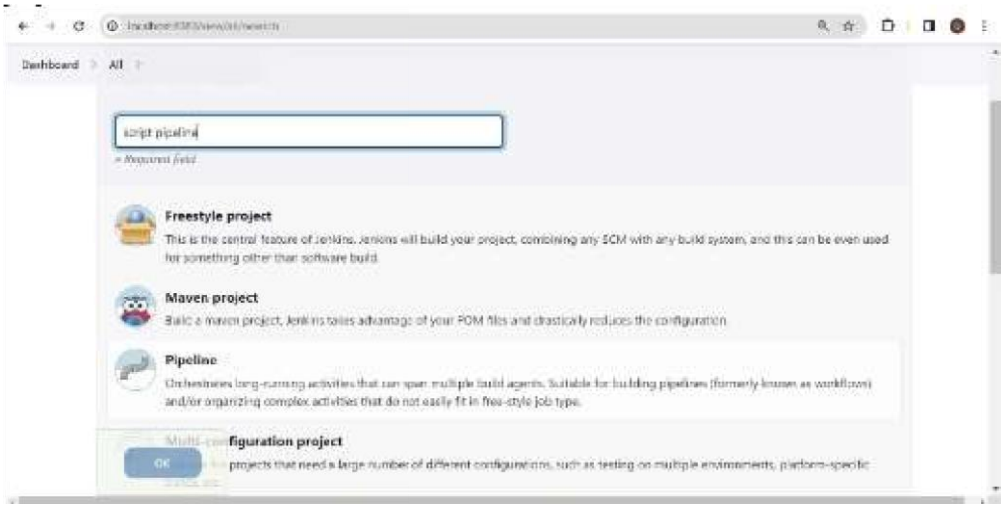
How many Ways we can create Pipeline

- We can Create Jenkins Pipeline in 2 Ways
- 1) Using **Build And Delivery Pipeline Plugins**
- **2) Using Groovy Script on the Fly(Here we use Jenkins file)**
  - Scripted
  - Declarative

STEP:1 Click on new item on page

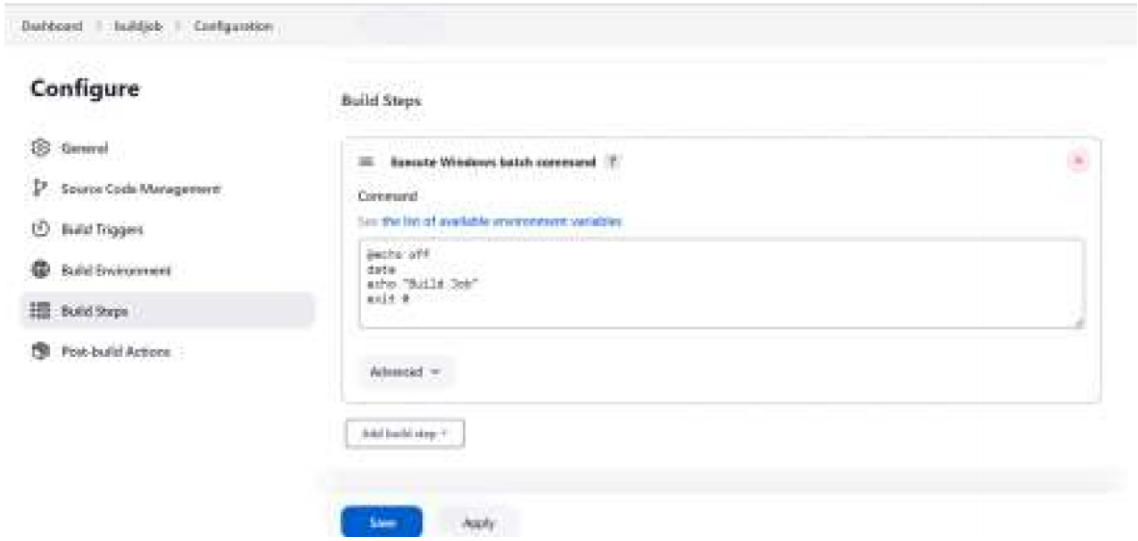


Step 2: Enter Build name you want and select project type ..here we are selecting maven project.. Then click the OK btn.

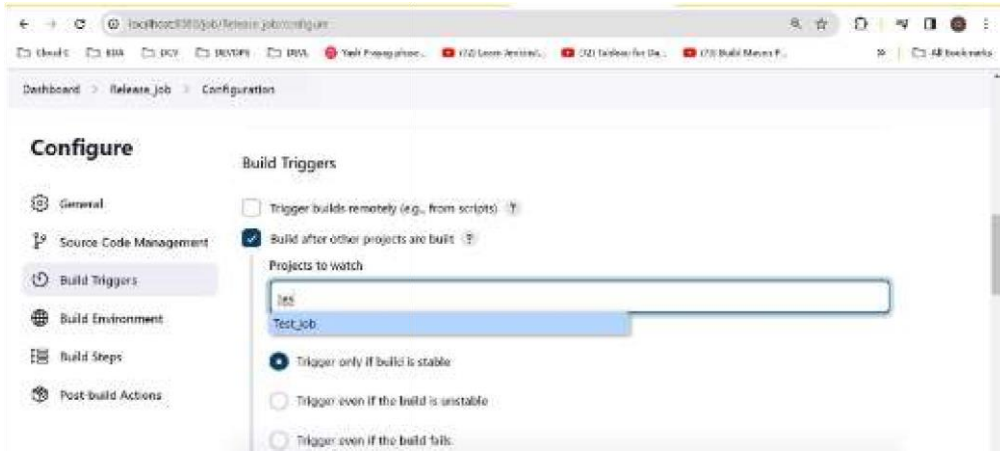


Step3: Build trigger → Windows batch command

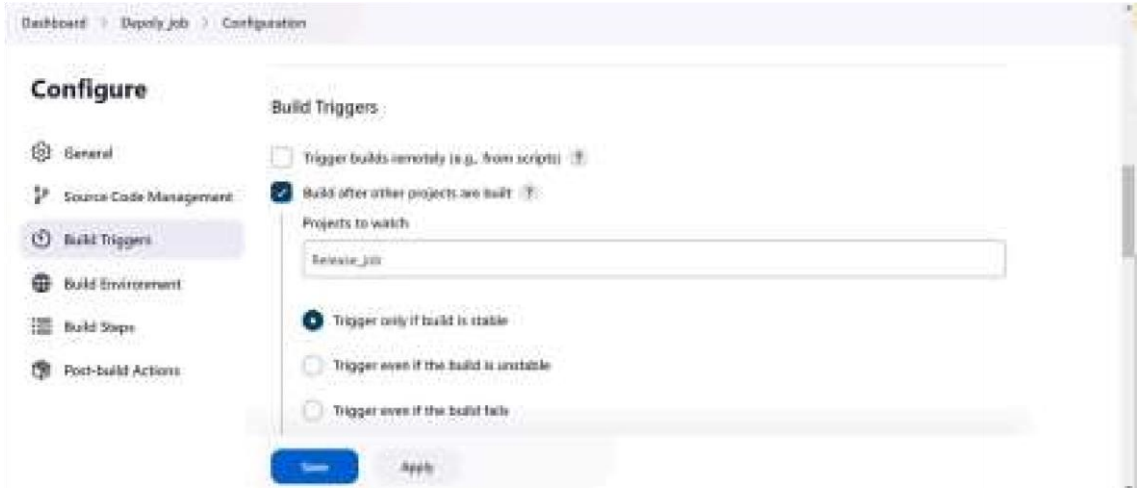
@echo off date



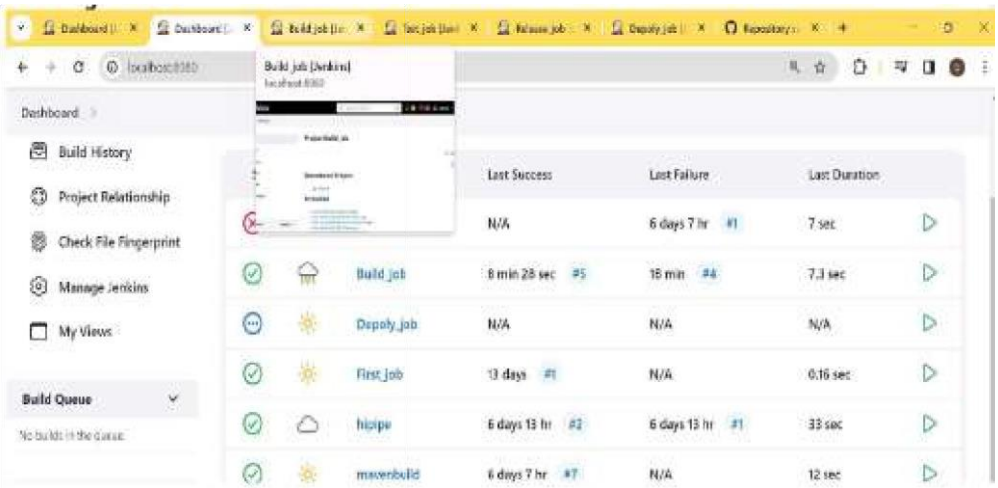
STEP:4 like buildjob create testjob,deployjob,releasejob



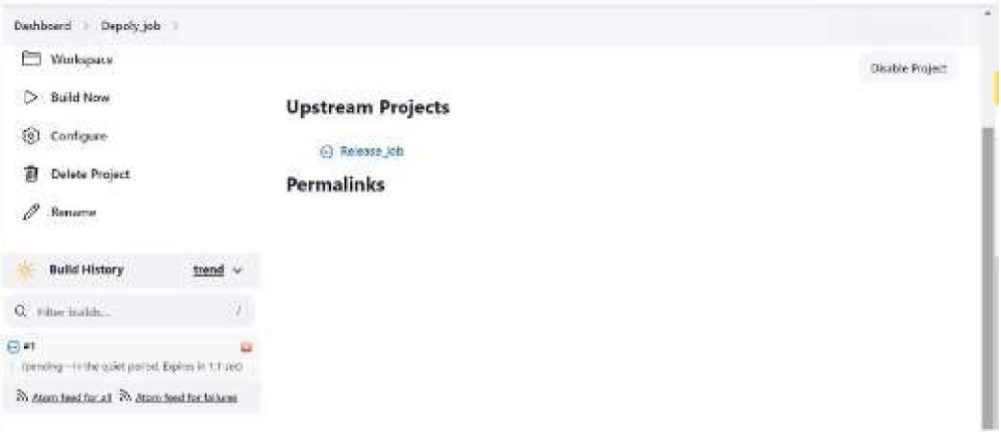
STEP:5 all job are interconnected using manual pipeline



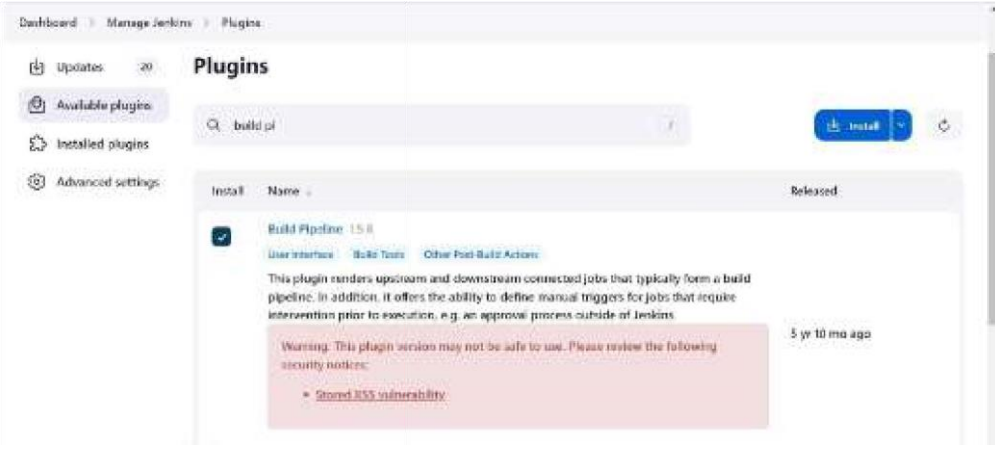
Step 6: Right click on all build open new browser.e.g .. Build job,Test_job , Deploy job.



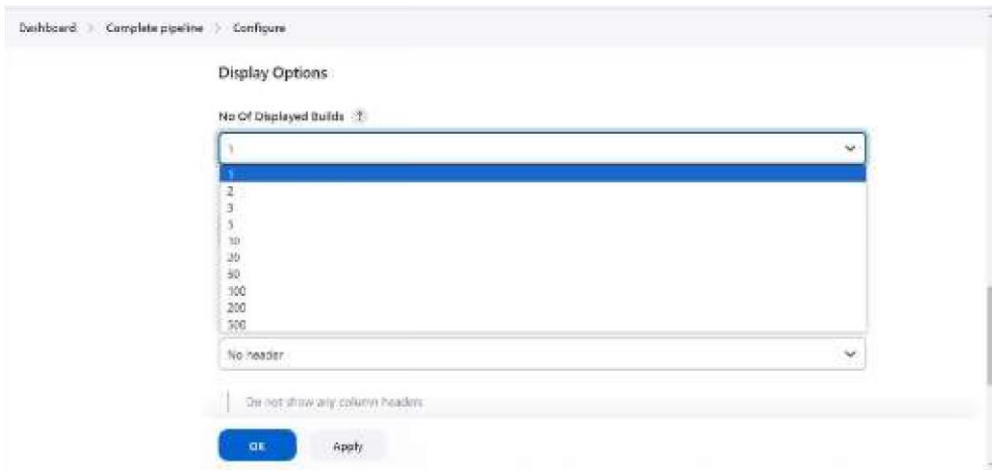
STEP:7 open Build_job first and click on build now button.. It will run the remaining build automatically …



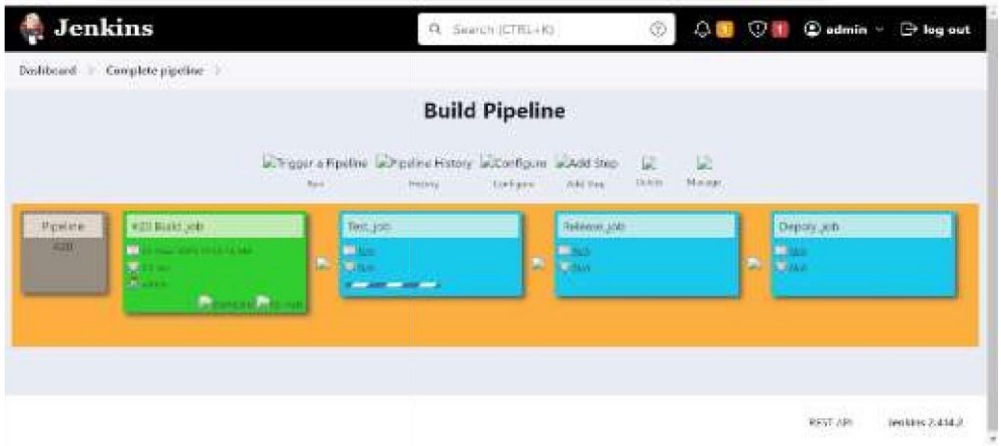
STEP:8 Above steps used for manual pipeline.now we want to automate pipeline for that install build pipeline and deliver pipeline plaguing from jenkins manage option..



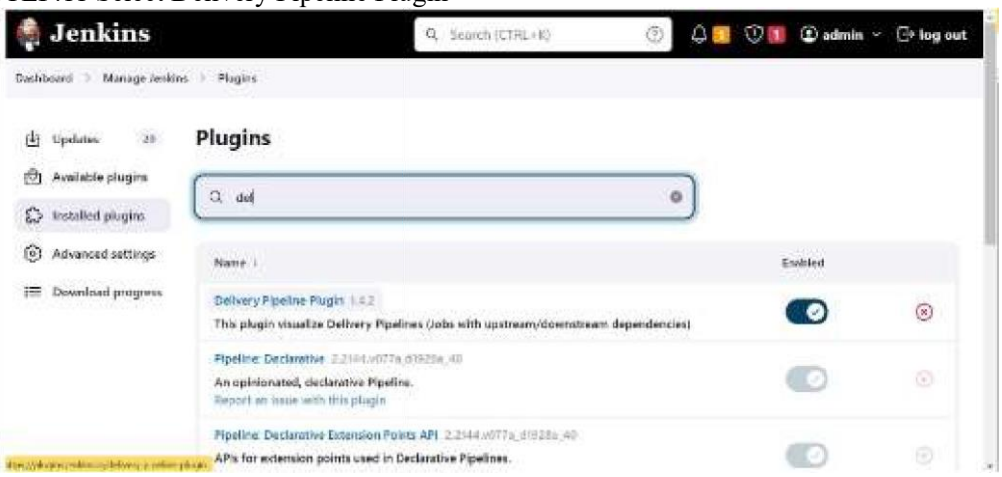
STEP:9 How many times you want to run the pipeline.



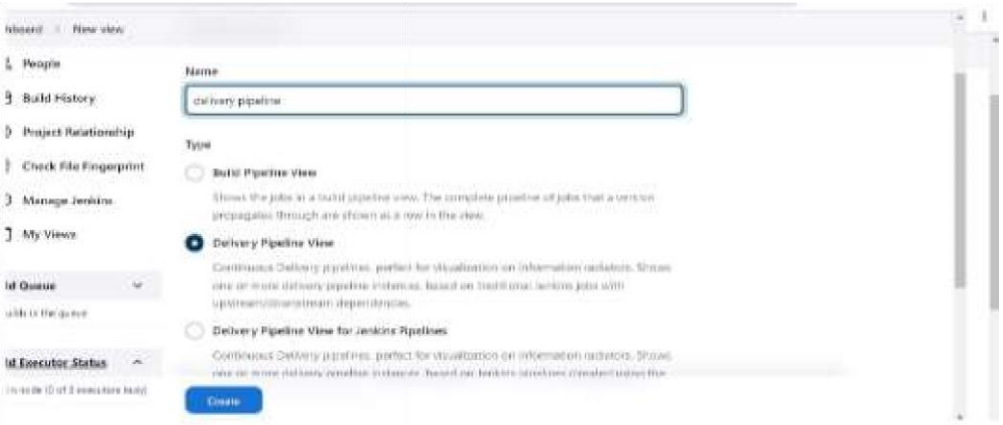
STEP:10 Build Pipeline



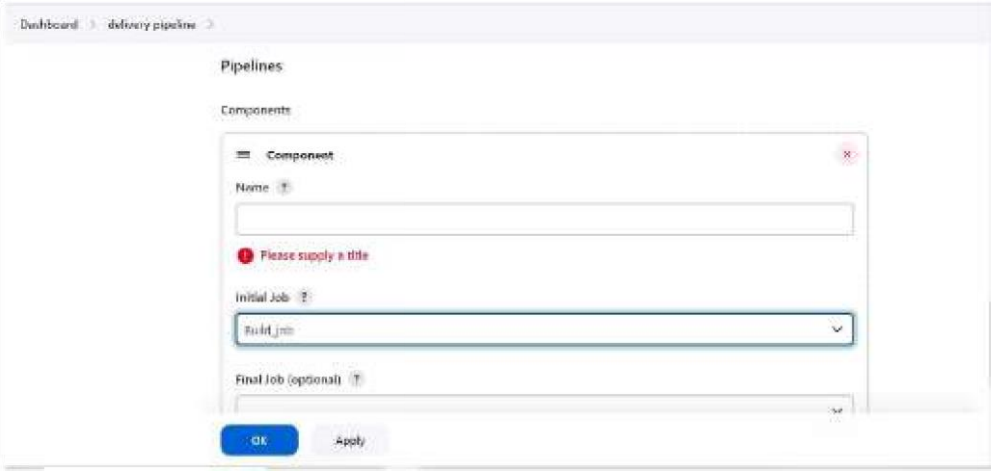
STEP:11 Select Delivery Pipeline Plugin



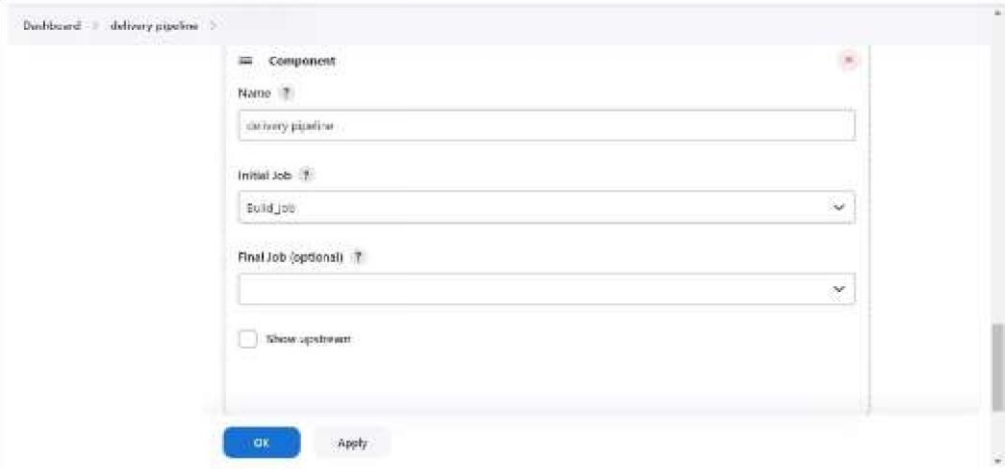
STEP:12 Delivery the pipeline



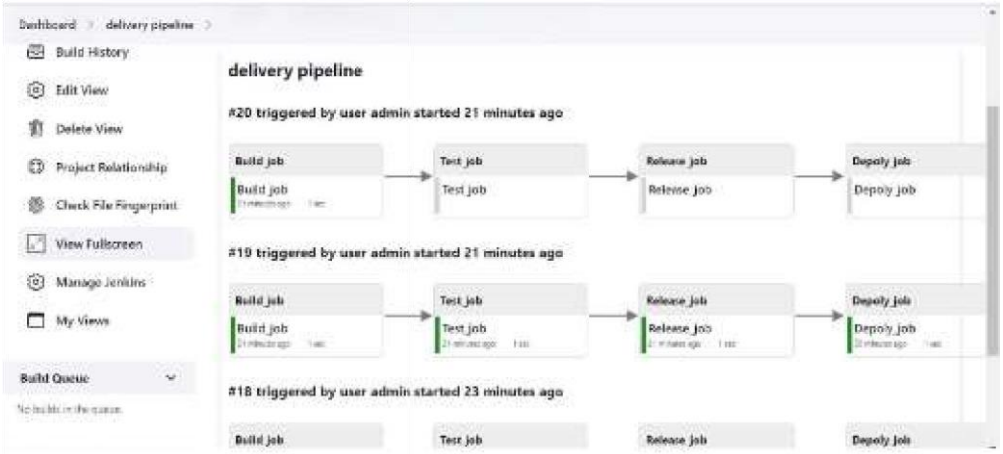
STEP:13 Give the Initial Job name “Build_job”.



STEP:14.Same procedure for Delivery Pipeline.

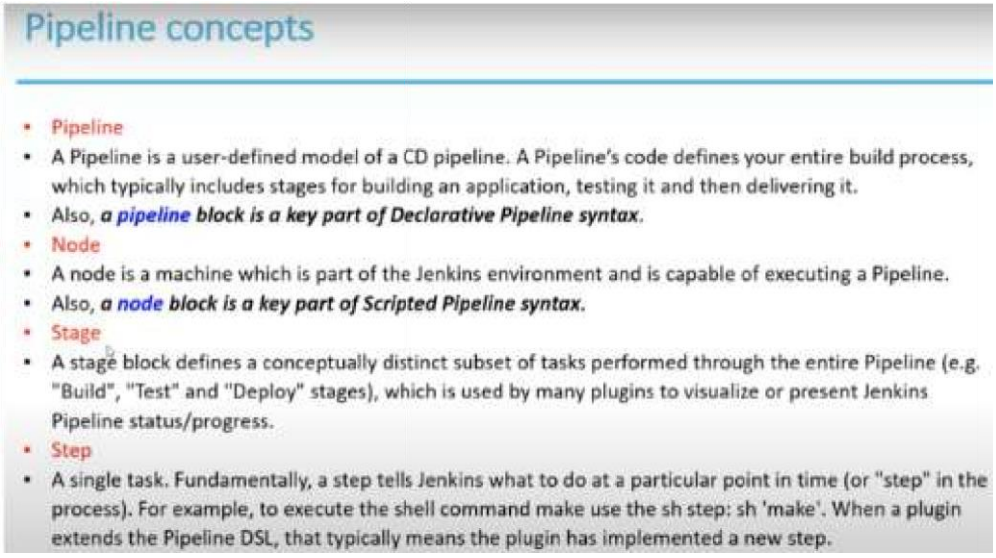


STEP 15:



STEP 16:

STEP 17:



OR
Script pipeline method two Method
1.Scripted

Step1: create project using Pipeline project

2.Pipeline option → Pipeline

```
node{ stage('Build') { echo 'Want to Build
          project'
          }
          stage('Test') { echo 'Test the
                    Project'
          }
          stage('Deploy') { echo 'deploy the
                    project'
          }
}
```

4.Click on Build pipeline

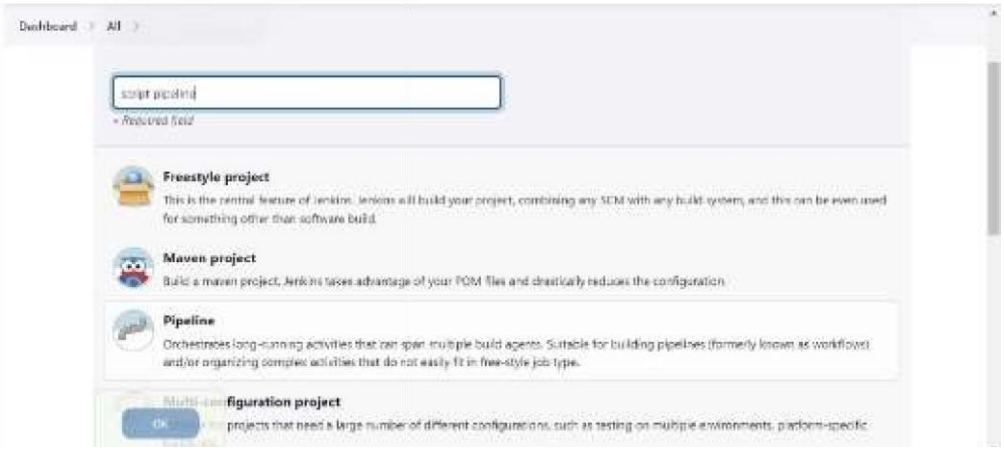
```
node{ stage('Build') { echo 'Want to Build
          project'
          }
          stage('Test') { echo 'Test the
                    Project'
          }
          stage('Deploy') { echo 'deploy the
                    project'
          }
}
```

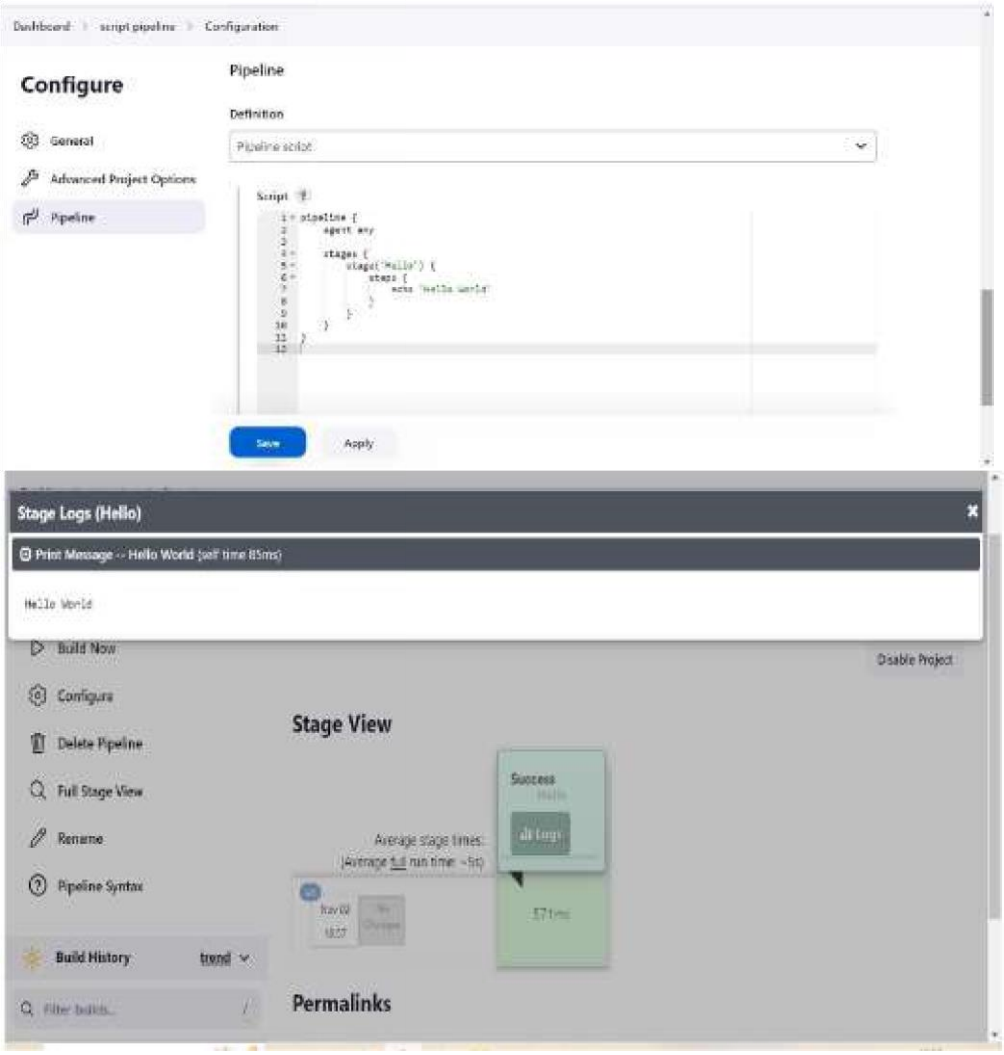
Declarative Pipeline

Jenkinsfile (Declarative Pipeline)

```
pipeline {
    agent any 1
    stages {
        stage('Build') { 2
            steps {
                // 3
            }
        }
        stage('Test') { 4
            steps {
                // 5
            }
        }
        stage('Deploy') { 6
            steps {
                // 7
            }
        }
    }
}
```

1. Execute this Pipeline or any of its stages, on any available agent.
2. Defines the "Build" stage.
3. Perform some steps related to the "Build" stage.
4. Defines the "Test" stage.
5. Perform some steps related to the "Test" stage.
6. Defines the "Deploy" stage.
7. Perform some steps related to the "Deploy" stage.





Conclusion : Hence we study CI/CD using jenkins server build test deploy jobs.

Devops Exp-5

1) What is the use of maven, gradle, Ant ?

→ • Maven : Manages projects builds, dependencies and documentation.
Features - Uses XML configuration, centralized repositories.
When to use : Maven is great for larger projects where dependency management in a structured build process is essential.

• Gradle : A flexible and efficient build system that combines the best of Maven & Ant.
Features : More flexible & efficient than Maven. Allows for incremental builds & parallel task execution.

• Ant : A basic build tool for automating tasks like compiling, packaging & testing
features : XML based configurations, highly flexible. Does not have built in dependency management.
When to use : Ant is good for smaller projects or those where a very custom build process is needed.

2) What is pipeline in Jenkins server ?

→ A pipeline in Jenkins is a set of automated

processes that defines the steps to build, test, and deploy software. It represents your CI/CD workflows as code. Jenkins pipelines help automate tasks & enable continuous delivery of software by allowing developers to define the entire lifecycle of an application in a structured, reproducible way.

Key concepts of Jenkins Pipelines:

i) Pipeline as code
ii) Stages and steps
iii) Types of pipelines
iv) Pipeline stages
v) Pipeline features

Benefits

i) Automation
ii) Version control
iii) Customization
iv) Scalability

Conclusion: By using Jenkins pipeline teams can streamline their CI/CD processes ensuring faster more reliable releases.

23

4/11/24

EXPERIMENT NO. 6

Name: Aditi Ambasta Roll: S004 SAP: 60018220113

Brach: AI & DS Subject: Devops Laboratory (DJS22ADL5014) Sem: V

Aim: Setup selenium web driver using Eclipse software

Practical:

Selenium WebDriver Installation

Selenium installation is a 3-step process:
Step 1: Install Java SDK
Step 2: Install Eclipse
Step 3: Install Selenium Webdriver Files
Step 4: Configure Eclipse IDE with WebDriver
In this tutorial, we will learn how to install Selenium Webdriver. Below is the detailed process

NOTE: The versions of Java, Eclipse, Selenium will keep updating with time. But the installation steps will remain the same. Please select the latest version and continue the installation steps below

Step 1 – Install Java Software Development Kit (JDK)

Install Java

Follow below steps to complete your Java installation.

■ Go to the Java Downloads Page and click on the option for Java Platform (JDK).

Java SE Development Kit 11.0.1

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.

○ **Accept License Agreement** ◉ **Decline License Agreement**

Product / File Description	File Size	Download
Linux	147.4 MB	jdk-11.0.1_linux-x64_bin.deb
Linux	154.09 MB	jdk-11.0.1_linux-x64_bin.rpm
Linux	171.43 MB	jdk-11.0.1_linux-x64_bin.tar.gz
macOS	166.2 MB	jdk-11.0.1_osx-x64_bin.dmg
macOS	166.55 MB	jdk-11.0.1_osx-x64_bin.tar.gz
Solaris SPARC	186.8 MB	jdk-11.0.1_solaris-sparcv9_bin.tar.gz
Windows	150.98 MB	jdk-11.0.1_windows-x64_bin.exe
Windows	170.99 MB	jdk-11.0.1_windows-x64_bin.zip

■

In the next page, select the Accept License Agreement radio button, accept it and click the download link against your matching system configuration.

■ You can run the installer once the download is over and follow onscreen instructions.

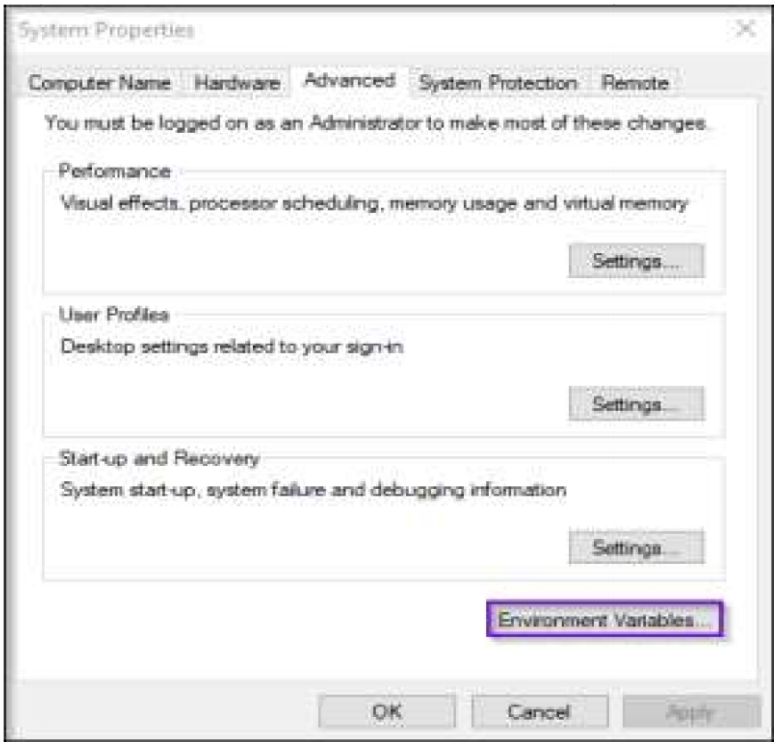
○ Go to start and search for 'System'

○ Click on 'System'

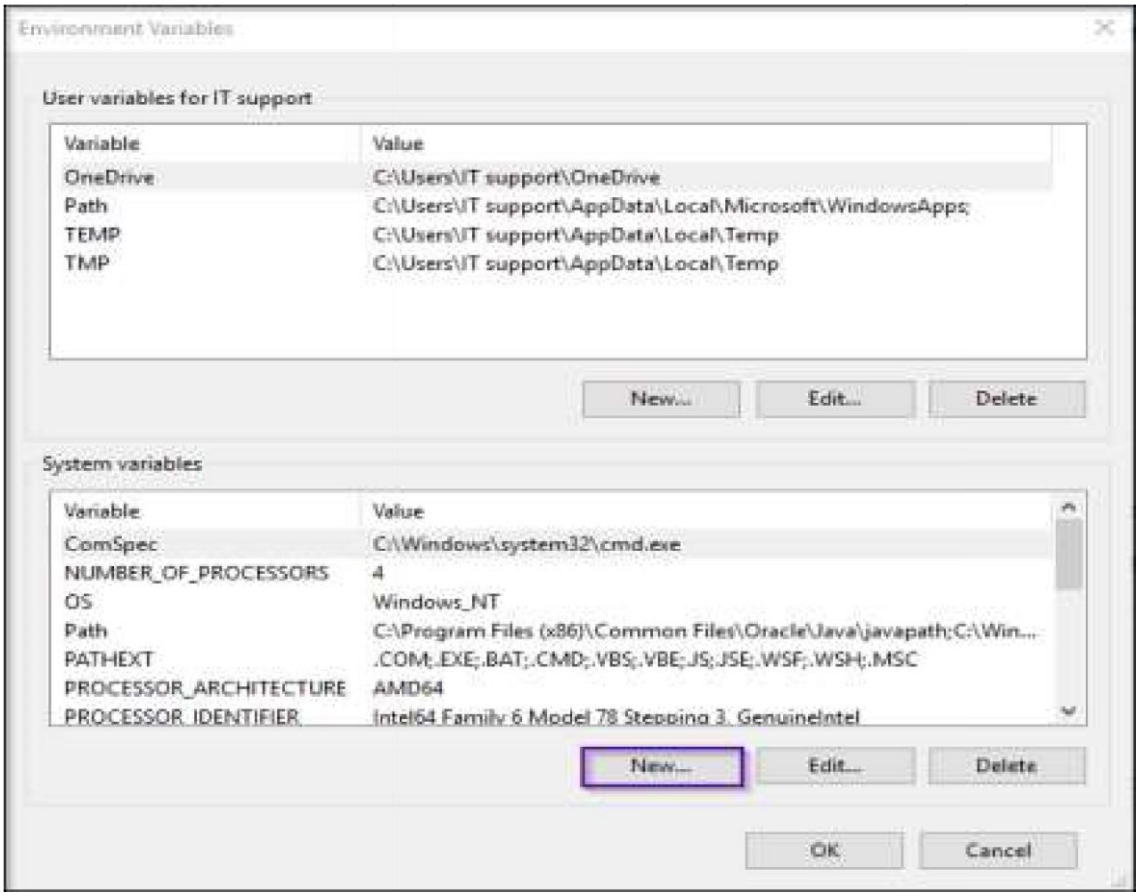
○ Click on 'Advanced system settings'

○ Click on 'Environment Variables' under 'Advanced' tab as shown

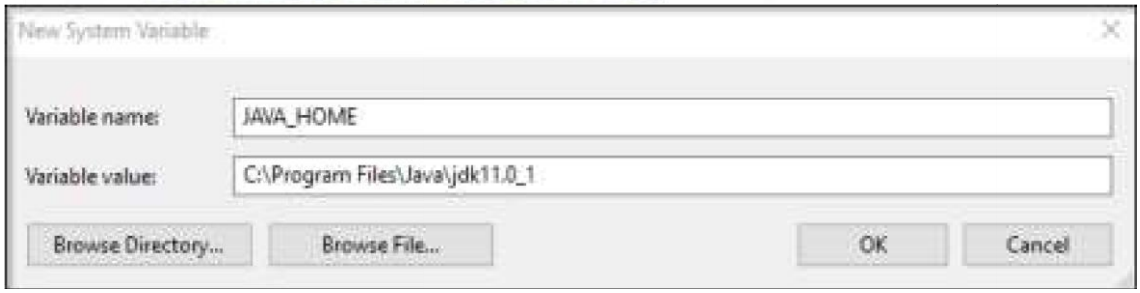
- below:



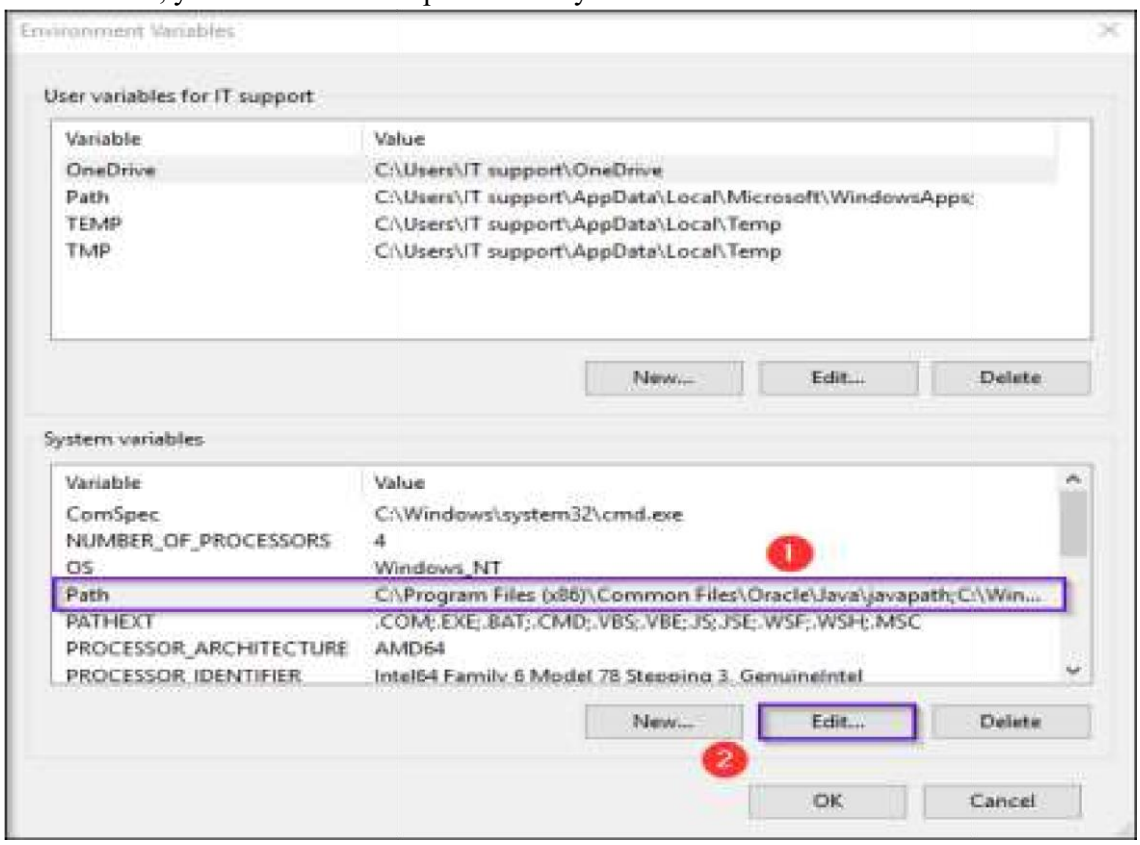
- Next, under system variables choose new and enter the variable name as 'JAVA_HOME' and the full path to Java installation directory as per your system as shown below:



- Below figure depicts the configuration of environment variable name and value.



■ Next thing that you have to do is to configure your environment variables. Let's see how to do that. Here, you have to edit the path of the system variable as shown below.



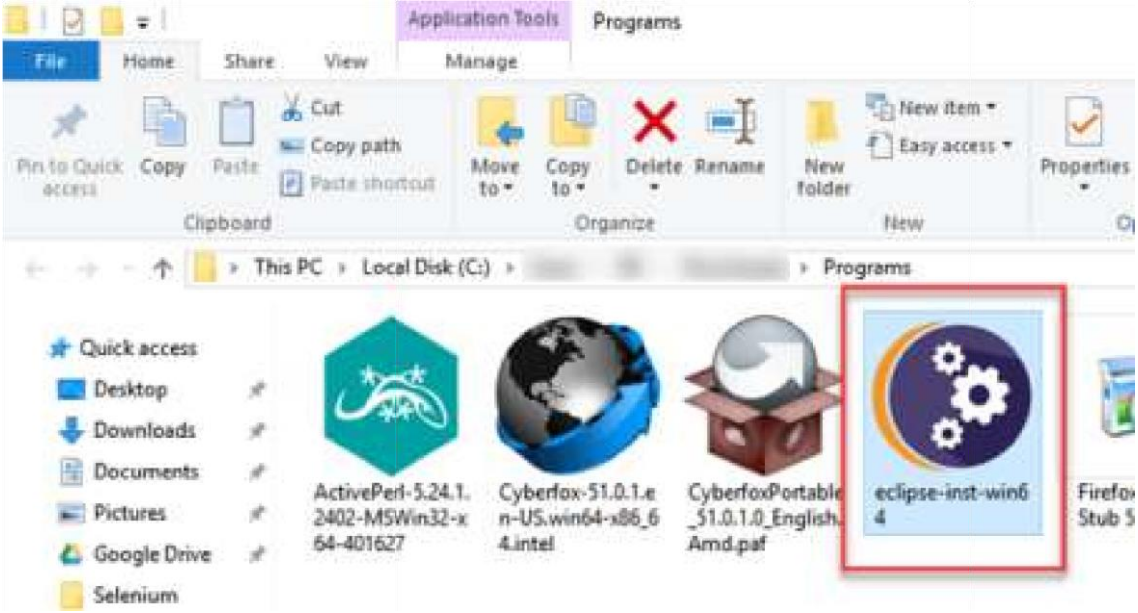
■ Under 'Variable value', at the end of the line, enter the following path – %JAVA_HOME%bin;
Now, you can click 'OK' and you are done.
■ Now to cross-check the installation, just run following command in cmd – java -version. It should display the installed version of Java in your system.

Step 2 – Install Eclipse IDE

Download the latest version of "Eclipse IDE for Java Developers" here. Be sure to choose correctly between Windows 32 Bit and 64 Bit versions.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

You should be able to download an exe file named “eclipse-inst-win64” for Setup.



Double-click on a file to Install the Eclipse. A new window will open. Click Eclipse IDE for Java Developers.

After that, a new window will open which click button marked 1 and change path to "C:\eclipse". Post that Click on the Install button marked 2

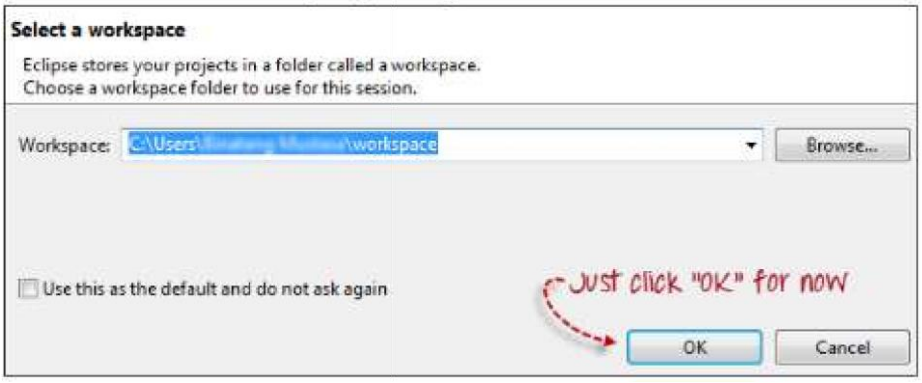
After successful completion of the installation procedure, a window will appear. On that window click on Launch.

This will start the eclipse neon IDE for you.
Step 3 – Selenium WebDriver Installation

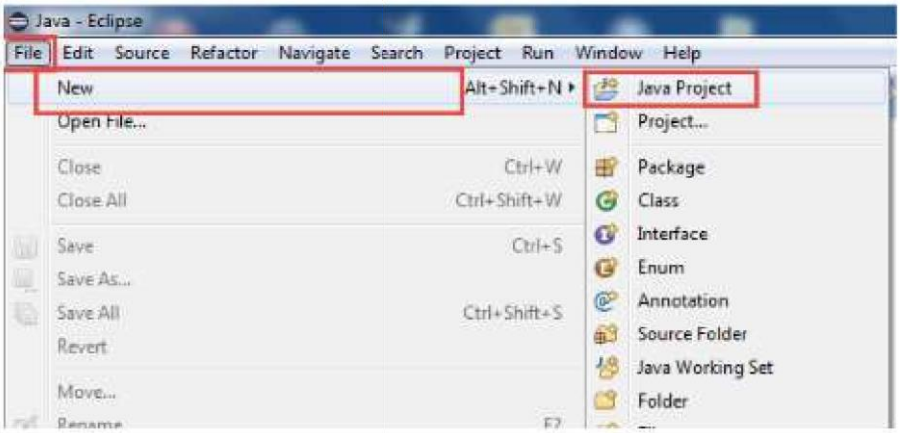
You can download Selenium Webdriver for Java Client Driver here. You will find client drivers for other languages there, but only choose the one for Java.

This download comes as a ZIP file named "selenium-3.14.0.zip". For simplicity of Selenium installation on Windows 10, extract the contents of this ZIP file on your C drive so that you would have the directory "C:\selenium-3.14.0\". This directory contains all the JAR files that we would later import on Eclipse for Selenium setup.
Step 4 – Configure Eclipse IDE with WebDriver

1.Launch the "eclipse.exe" file inside the "eclipse" folder that we extracted in step 2. If you followed step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe.
2.When asked to select a workspace, just accept the default location.

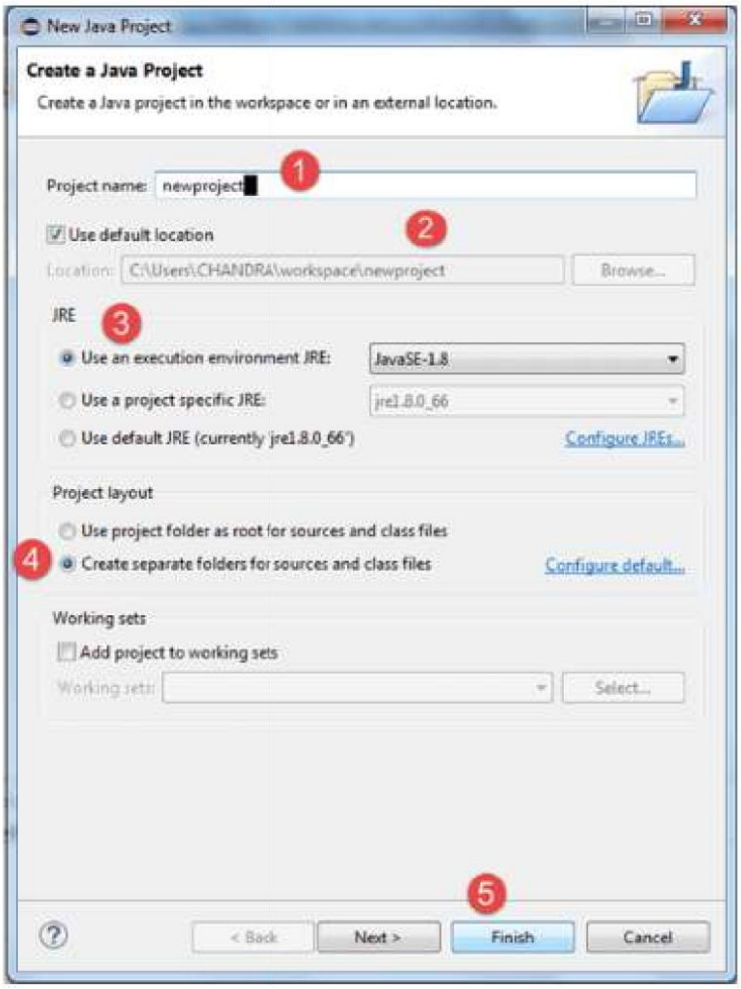


3. Create a new project through File > New > Java Project. Name the project as “newproject”.



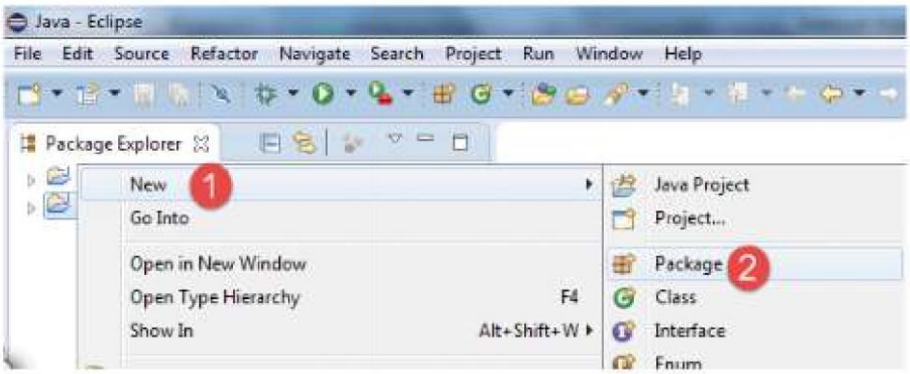
A new pop-up window will open. Enter details as follow

1. Project Name
2. Location to save a project
3. Select an execution JRE
4. Select the layout project option
5. Click on the Finish button



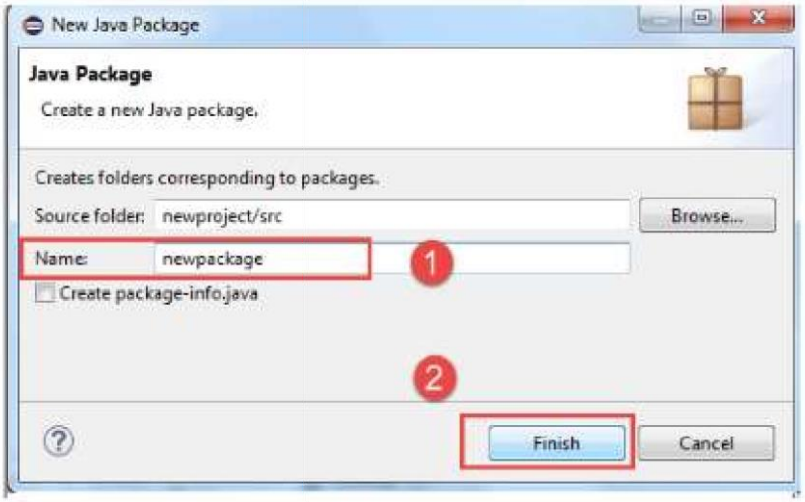
4. In this step,

1. Right-click on the newly created project and
2. Select New > Package, and name that package as "newpackage".



A pop-up window will open to name the package,

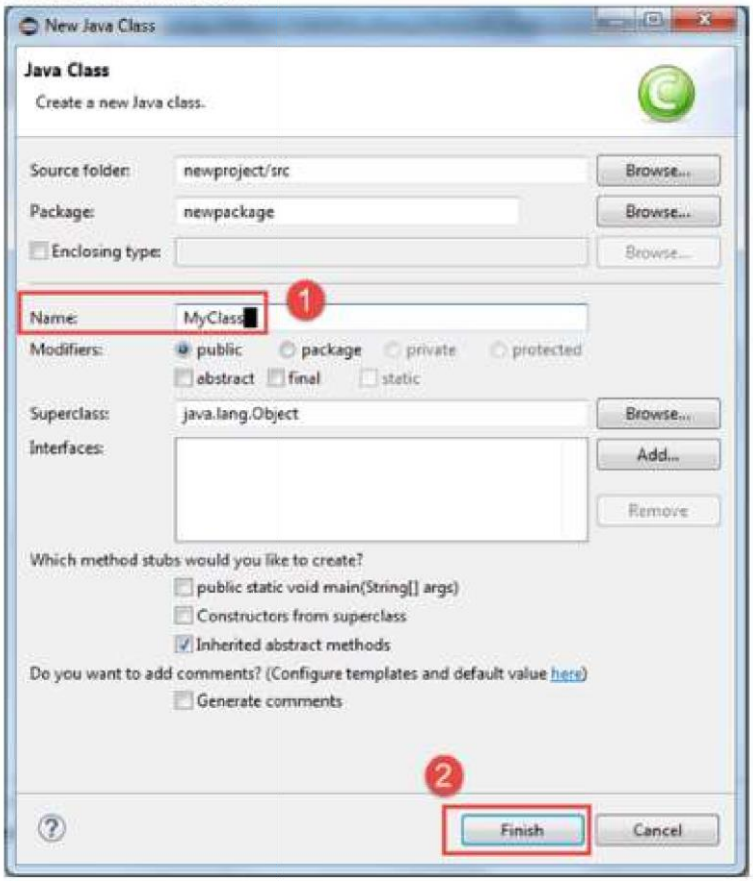
1. Enter the name of the package
2. Click on the Finish button



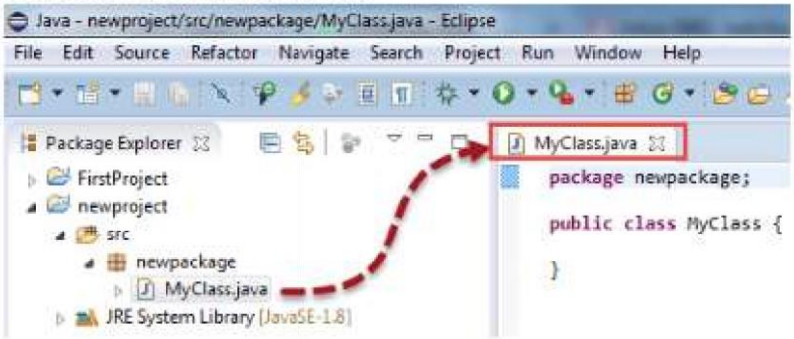
5. Create a new Java class under newpackage by right-clicking on it and then selecting- New > Class, and then name it as "MyClass". Your Eclipse IDE should look like the image below.

When you click on Class, a pop-up window will open, enter details as

1. Name of the class
2. Click on the Finish button



This is what it looks like after creating class.

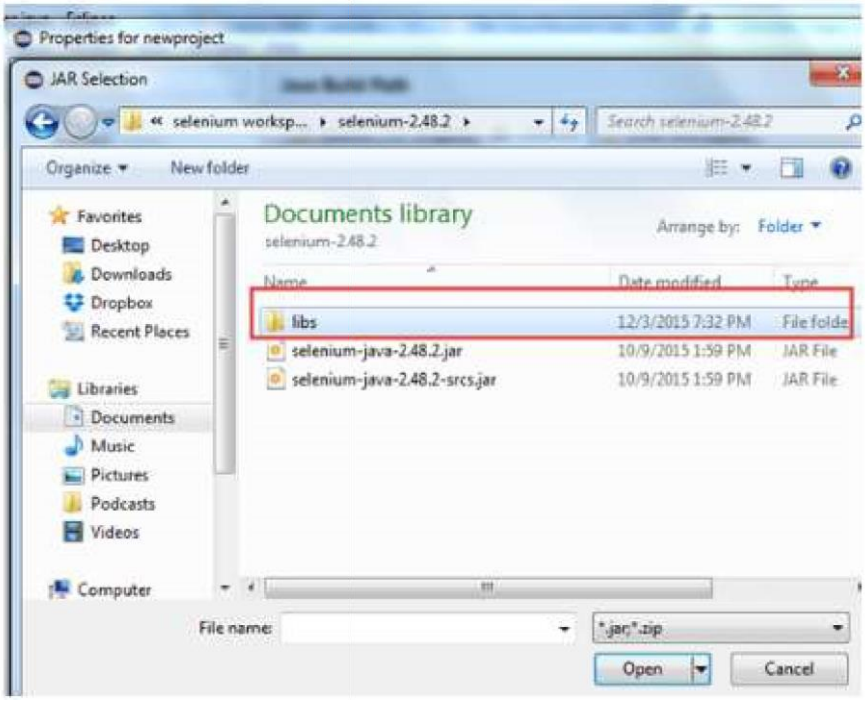


Now selenium WebDriver's into Java Build Path

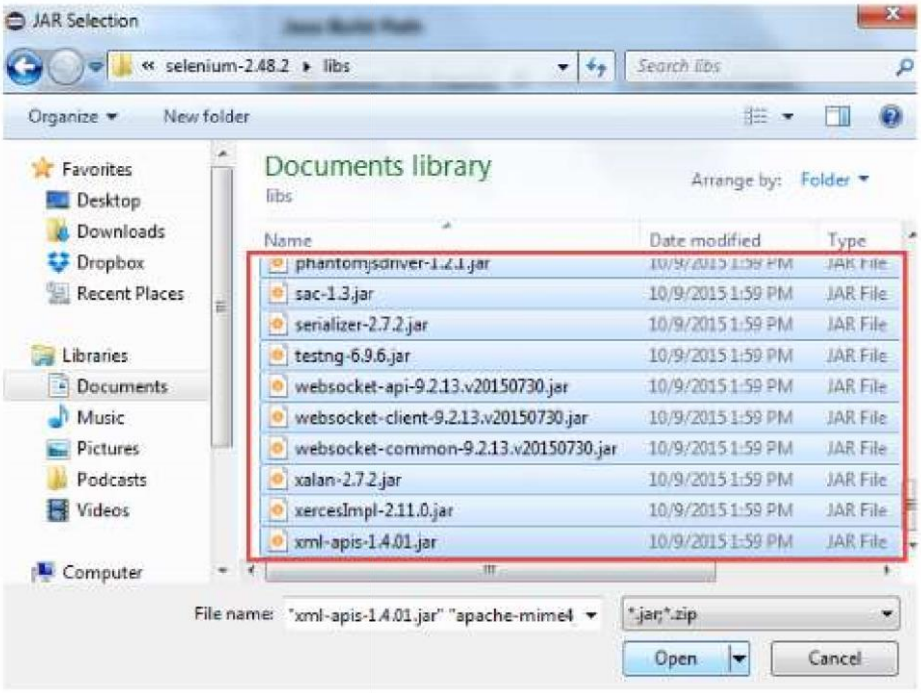
In this step,

1. Right-click on "newproject" and select Properties.
2. On the Properties dialog, click on "Java Build Path".
3. Click on the Libraries tab, and then
4. Click on "Add External JARs.."

When you click on "Add External JARs.." It will open a pop-up window. Select the JAR files from the selenium folder you want to add.

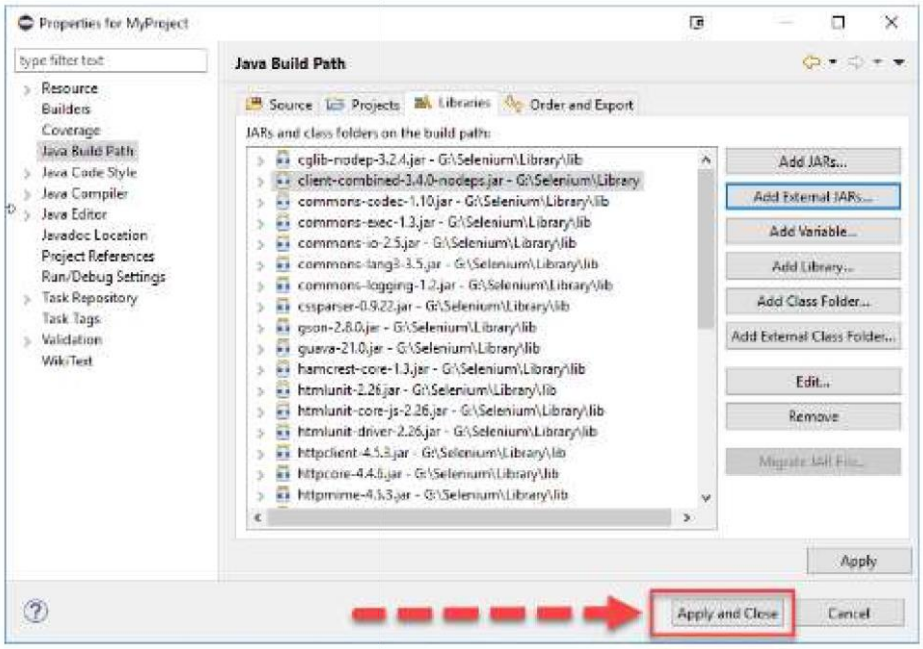


After selecting jar files, click on the OK button. Select all files inside the lib folder.

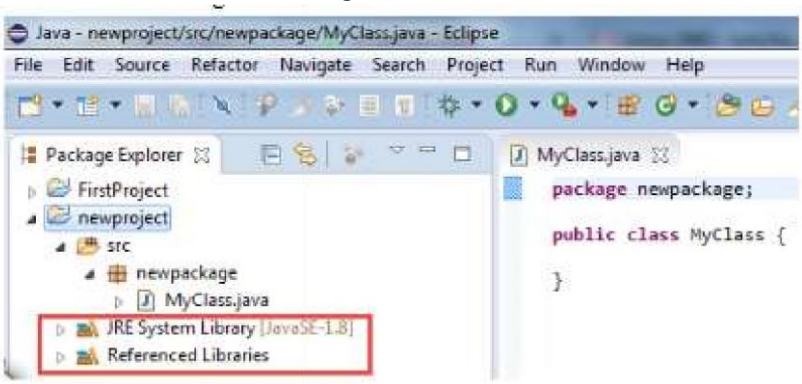


Select files outside lib folder

Once done, click "Apply and Close" button



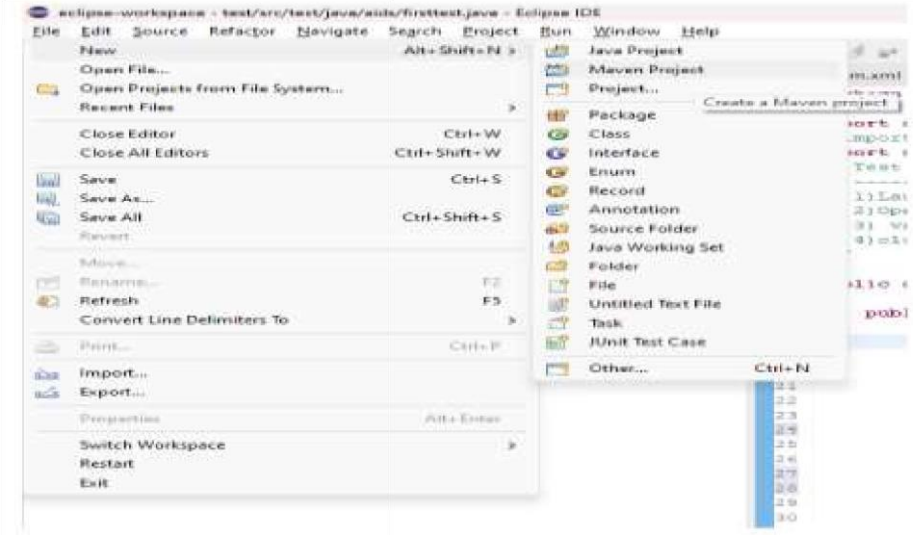
6. Add all the JAR files inside and outside the "libs" folder. Your Properties dialog should now look similar to the image below.



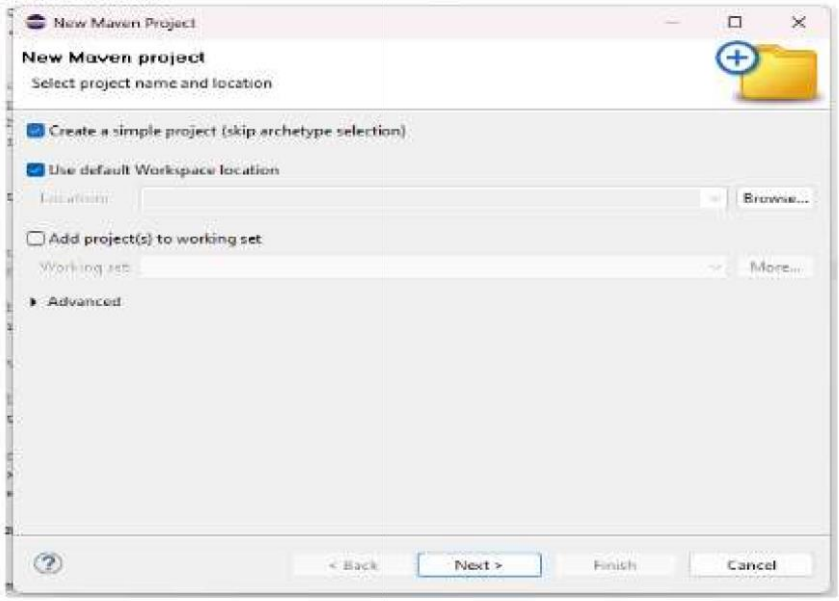
7. Finally, click OK and we are done importing Selenium libraries into our project.

2.Creating Maven Project

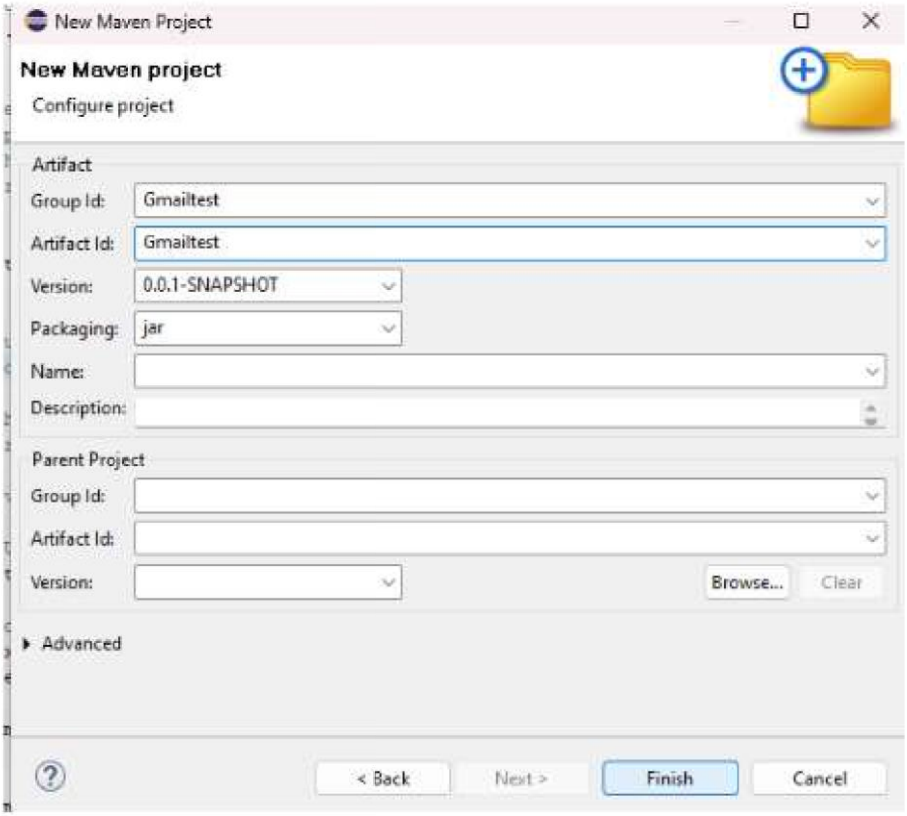
Step1:Create new project



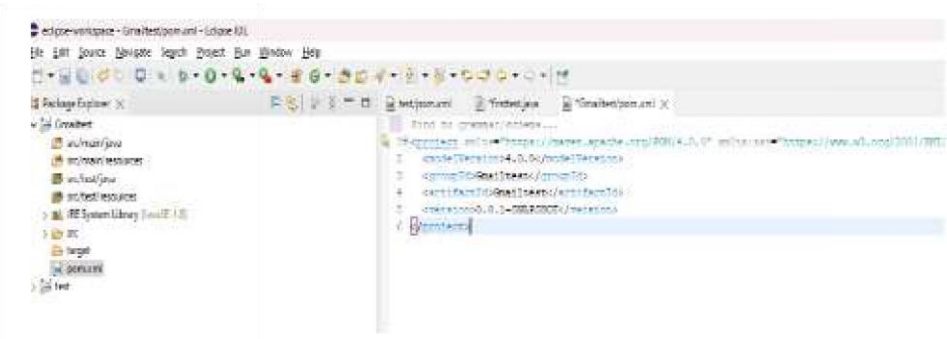
Step2: Click next



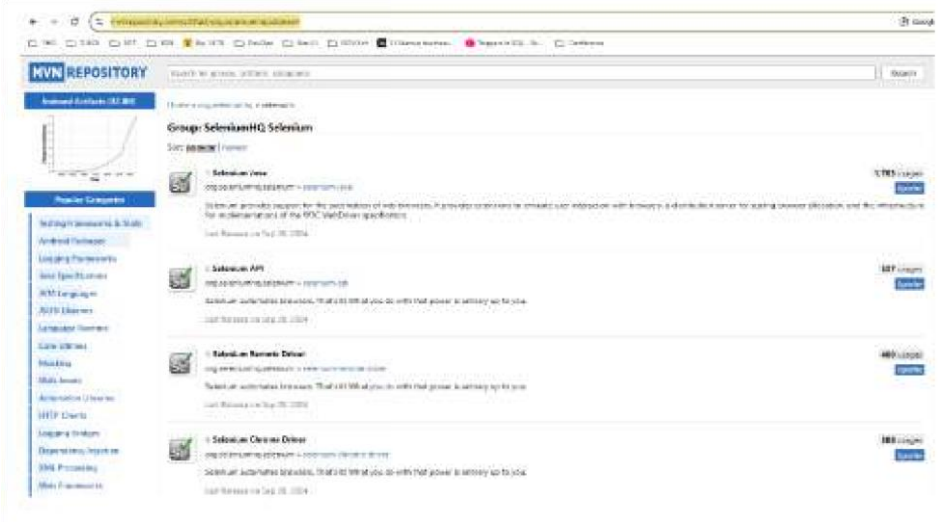
Step3:Give name of Group id and click finish



Step4:Open pom.xml and in first line replace http to https



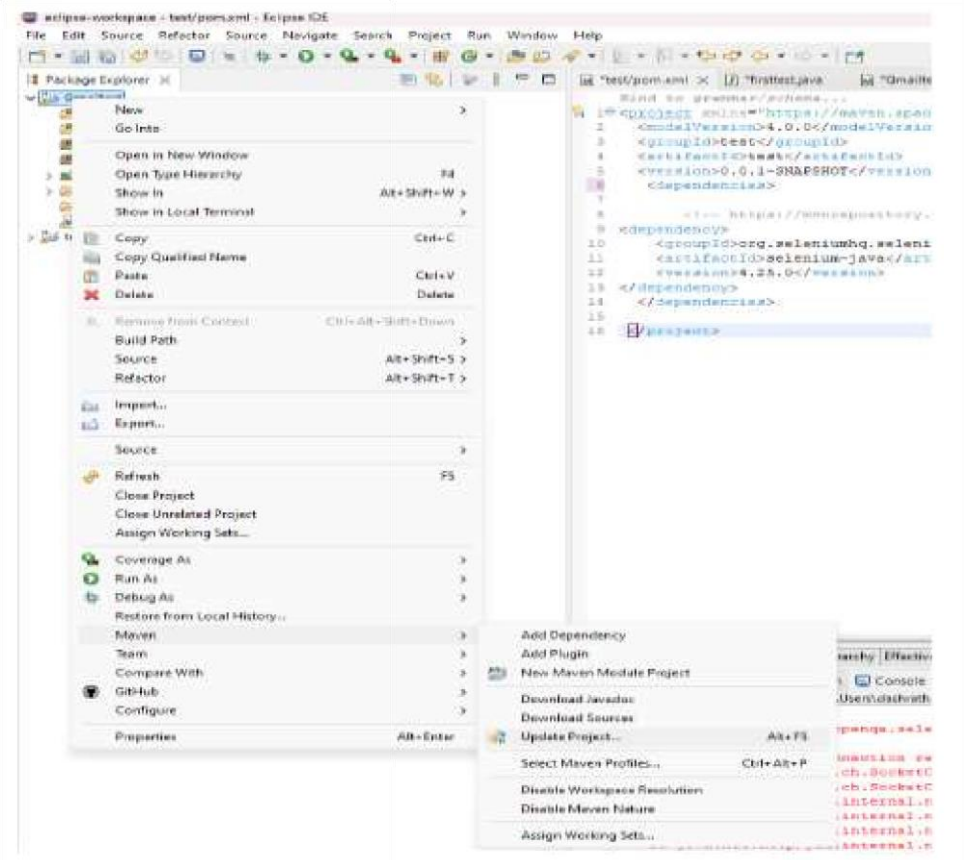
Step5:Configure with Selenium JAR file with eclipse Latest version. Search the "Selenium Java" in the text box.



Step6: Copy the dependency code and paste pom.xml

```
<dependencies>

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.25.0</version>
</dependency>

</dependencies>
```

Step7: save project next →Right clk on project –maven→ update project



Step8:Right Clk on src/test/java→ create package

Step9: Right Clk on package→ create class website

Conclusion: hence we study setup selenium web driver with eclipse editor

Devops Exp-6

Aim : Setup Selenium webdriven using eclipse software

Theory :

1) Features of Selenium

i) Language Support :
Supports multiple programming language such as Java, Python

ii) Multi Browser Capability :
Supports browsers such as chrome, firefox allowing easy testing

iii) Flexible : can be executed in linux, MAC etc.

iv) Community Support : Large Documentation & resource available

v) Open Source : freely available for everyone to use.

vi) Performance & Speed : Web drivers automate web testing allowing to execute test case quitely.

2) Explain different software testing automation tools

→ i) Browser stack Automate: Web testing product which helps run application tests on all desktop browsers. It includes automate scale & test management, accessibility testing to ensure that you have an end to end solution for all software

ii) Selenium: Developed by Jason Higgins in 2004, which is open source web user interface used for testing software. Software is developed on Linux, MAC & windows.

iii) Bug Bug: Web testing tool for test automation unlike conventional testing tools. technical & non technical users. It offers an intuitive interface for creating, editing & executing list without coding.

Conclusion: Hence, here we have successfully learnt about the different devops tools used during testing & successfully setup selenium webserver using eclipse edition.

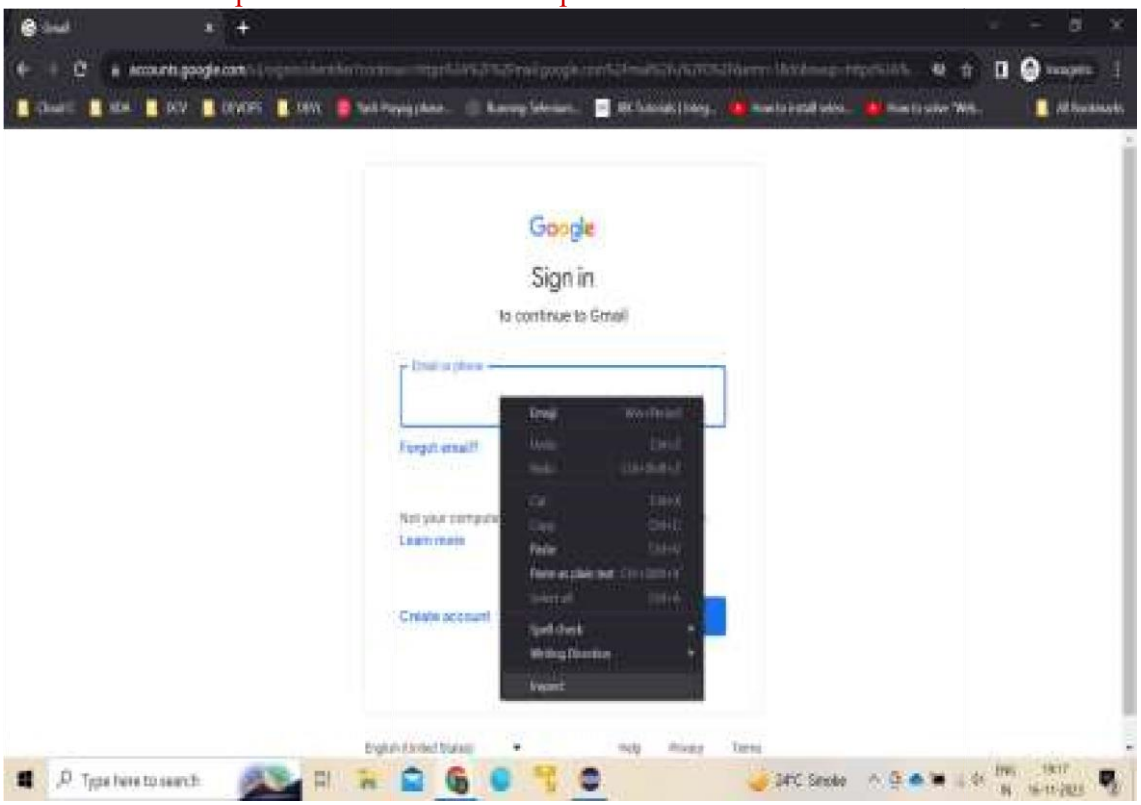
29

EXPERIMENT NO. 7

Name: Aditi Ambasta **Roll: S004** **SAP: 60018220113**
Brach: AI & DS **Subject: Devops Laboratory (DJS22ADL5014)** **Sem: V**

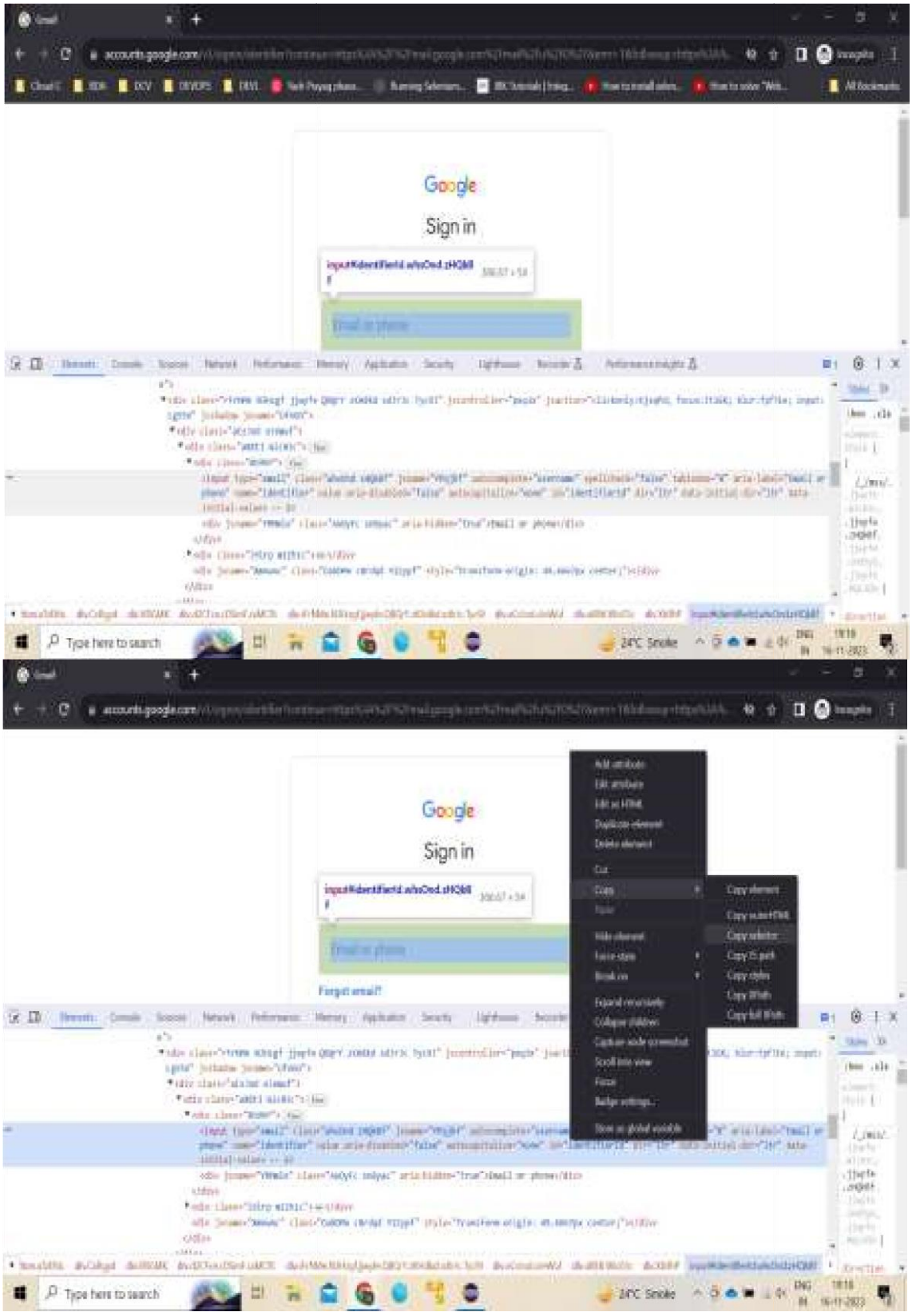
Aim: Run Gmail Signing Page test script using Selenium.
Practical:

Two methods to implement selenium test Script.

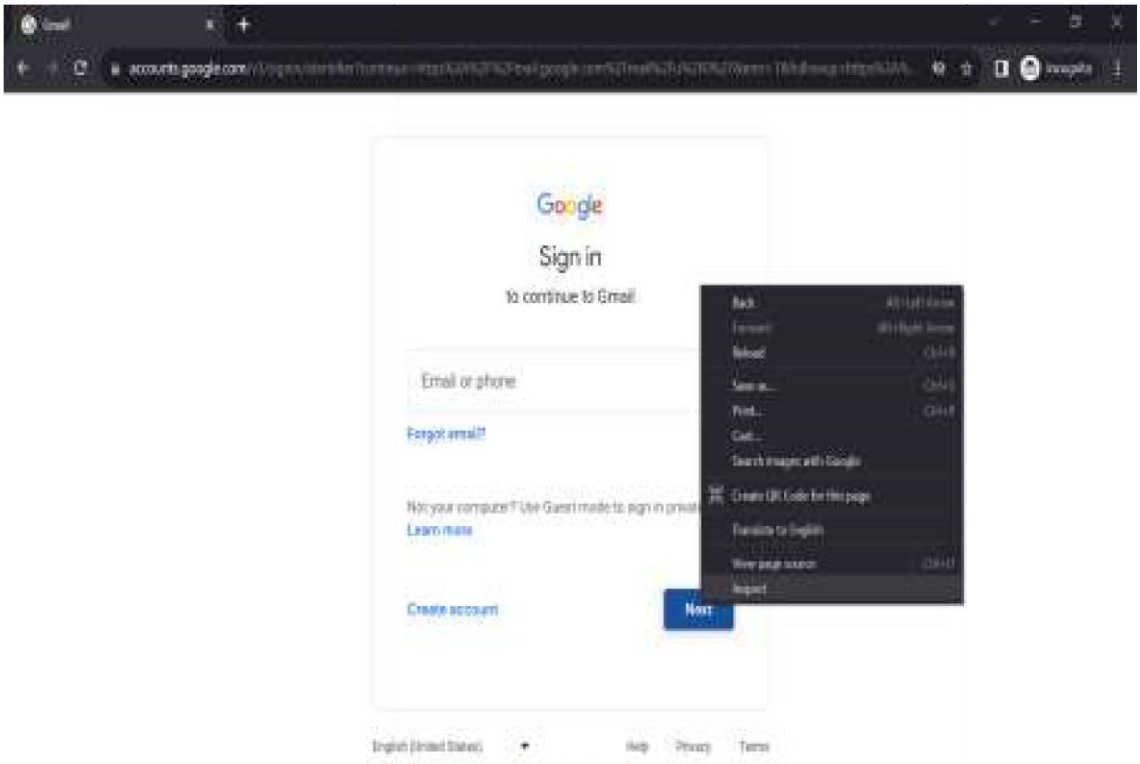


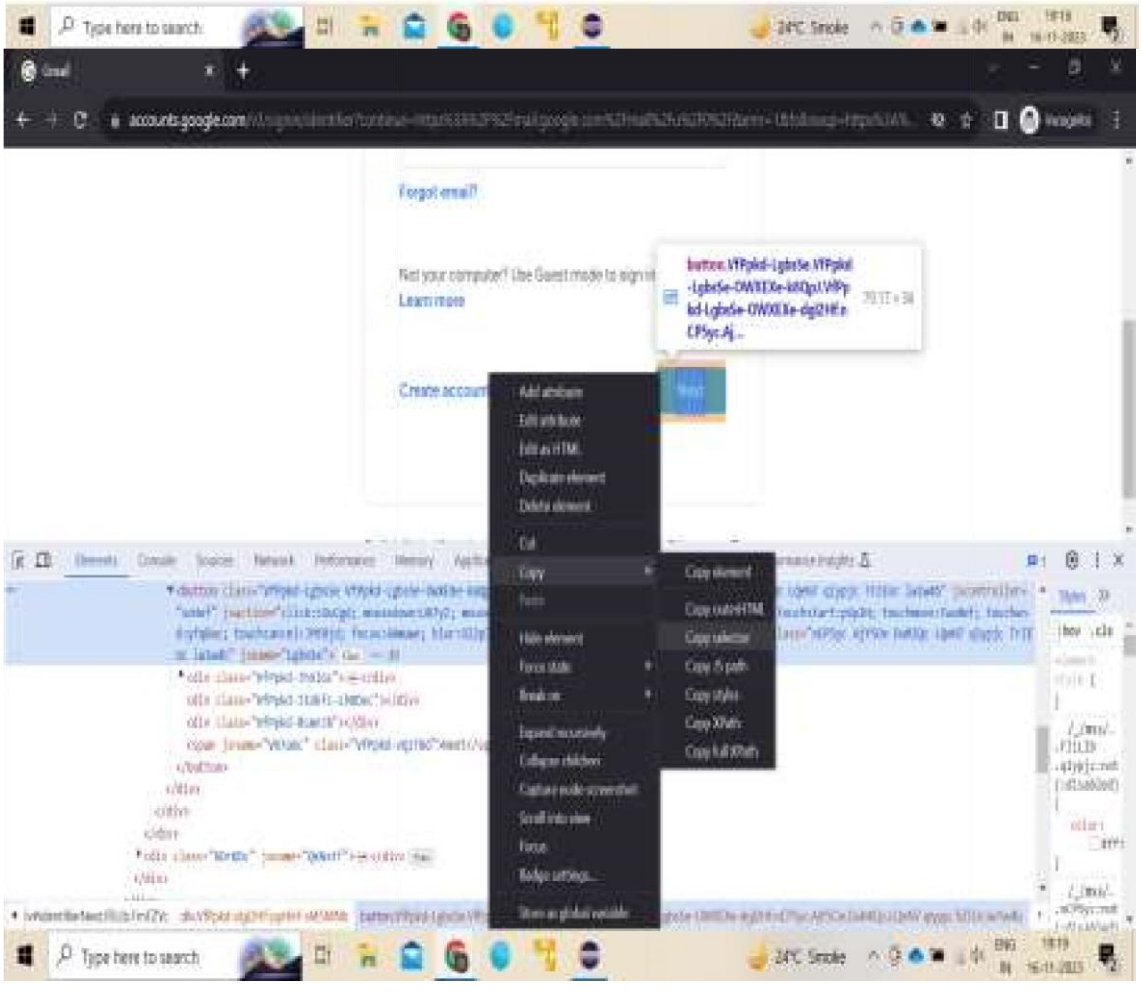
1.Downloading JAR files and attaching them to java projects(manually) – Not recommended.

Step1. Install Latest webdriver available in given link
Step2: Provide WebElement value(either id or name) for Email or phone textbox



Step:3 Click on Next Button and fidget element value


Google
Sign in
to continue to Gmail
Email or phone
Forgot email?
Learn more
Create account
Next
Back
Reload
Save as...
Print...
Cast...
Search images with Google
Create QR Code for this page
Translate to English
View page source
Inspect


Forgot email?
Learn more
Create account
Next
Add attribute
Edit attribute
Edit as HTML
Duplicate element
Delete element
Cut
Copy
Paste
Hide element
Force state
Break on
Expand recursively
Collapse children
Capture node screenshot
Scroll into view
Focus
Badge settings...
Store as global variable
Copy element
Copy outerHTML
Copy selector
Copy JS path
Copy styles
Copy XPath
Copy full XPath

Step4: Write given code in Eclipse editor

```
package aids;
import java.util.concurrent.TimeUnit;  import
org.openqa.selenium.By;  import
org.openqa.selenium.WebDriver;  import
org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.*;
public class gmail1 {  public static void main(String[] args) {
                        // TODO Auto-generated method stub
                        //System.setProperty("webdriver.chrome.driver","C:\\Web
Driver\\chromedriver-win64\\chromedriver.exe");  WebDriver driver= new
                        ChromeDriver();  driver.get("https://www.gmail.com");
                        System.out.println(driver.getTitle());


driver.findElement(By.id("identifierId")).sendKeys("softesting001@gmail.com
");
                         driver.findElement(By.id("identifierNext")).click();
                        driver.findElement(By.id("password")).sendKeys("aids@123");
                        driver.findElement(By.id("passwordNext")).click();  driver.quit();  driver.close();
            }
}
```

Step5: Click on run Button

2.Creating Maven Project

Reference of Experiment no6

Step9: Right Clk on package→ create class website(firsttest)

```
package aids;
import org.openqa.selenium.chrome.ChromeDriver;
//import org.openqa.selenium.edge.EdgeDriver;  import
org.openqa.selenium.WebDriver;
/* Test Case
*   --------
*   1)Launch Browser(chrome)
*   2)Open URL https://www.msbte.com
*   3) validate the title should be "Home |
              ई" * 4)close browser
*/  public class firsttest {  public static void main(String[] args) {
                        // TODO Auto-generated method stub
                        //System.out.println("dashrath kale");

                        //1)Launch Browser(chrome)
                        //ChromeDriver driver=new ChromeDriver();  WebDriver driver=new
                        ChromeDriver();
                    // WebDriver driver=new EdgeDriver();
```

```
                //2)Open URL https://www.msbte.com
                driver.get("https://www.iitb.ac.in/");

                //3) validate the title should be "Home |
                ई"  String text=driver.getTitle();  if
                (text.equals("Home |
                    ई"))
{
                                System.out.println("Test Case Pass");
                }  else {
                                System.out.println("Test Case Fail");
                        }  driver.close();
                //driver.quit();

}
}
```

Conclusion: Hence we study how run test script with help of Selenium web driver

Devops - Exp 7

1. What types of testing are done in CD on the test server in production & explain any one

→ On a Continous Deployement (CD) pipeline, testing on a test server in a production-like environment is critical for ensuring that the application is ready for release without manual intervention. The key types of testing done in CD on test server are:

① Smoke testing
② Sanity testing
③ End-to-End testing
④ User acceptance testing
⑤ Performance testing

User Acceptance Testing (UAT) is a type of testing perform in the final phase of the CD pipeline, typically on test server that mimics the production environment. The primary purpose of UAT is to validate whether the software meets the business requirements & is ready for release to production.

Detailed explanation of UAT.

① End to end user focus
② Business requirement validation
③ Real world scenarios.
④ Approval for release

Advantages,

① Ensures that the product meets the expectations of users.

(2) Reduces the risk of post-release issues as actual business scenarios are tested.

2. What types of testing are done in CI build?

→ In CI build, various types of testing are done to ensure the code is functional, reliable & maintainable. Key types of testing are,

① Unit testing
② Integration testing
③ Acceptance testing
④ Performance testing
⑤ Smoke testing

Unit testing is the process of testing individual units or components of a software application to ensure that each part of the code functions as expected. Frameworks & tools are JUnit and pytest. Advantages are early bug detection & documentation.

Conclusion: Thus CI & CD play important role for testing.

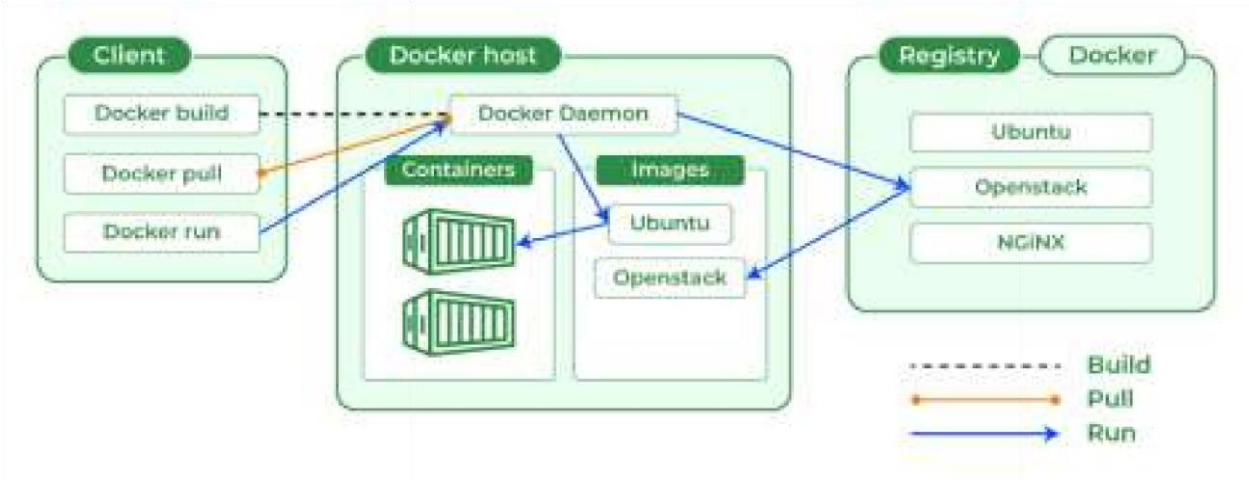
Dec 23

EXPERIMENT NO. 8

Name: Aditi Ambasta Roll: S004 SAP: 60018220113 Brach: AI-DS Subject: Devops Laboratory (DJS22ADL5014) Sem: V

AIM: Create Docker Image for to implement ETL pipeline with mysql workbench using python.

Practical:



Install Docker Desktop

Step 1: Download Docker Desktop

1. Go to the official Docker website: Docker Desktop for Windows.
2. Click Download Docker Desktop for Windows.

Step 2: Install Docker Desktop

1. Locate the downloaded .exe file in your Downloads folder and double-click to start the installation.
2. Follow the installer's instructions:
   - ○ Click OK to continue.
   - ○ The installer will download and install the necessary components.
3. After the installation completes, restart your system.

Step 3: Launch Docker Desktop

1. After restarting, find Docker Desktop in your Start Menu and launch it.
2. During the first run, Docker Desktop may ask for administrative privileges to configure components. Grant the permission.
3. Docker Desktop should now launch and show a window indicating that it's running.

Step 4: Verify Docker Installation
Open Command Prompt or PowerShell and run the following command to verify Docker is running:

docker --version
You should see something like:

Docker version 24.0.0, build abcdef1 Test

Docker by running a simple container:

docker run hello-world

1. If Docker is installed correctly, you should see a message saying that the Docker installation is working.

```
C:\code>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d211f485f2dd1dee407a80973c8f129f00d54604d2c90732e8e320e5038a0348
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

Step 5:Prerequisites

1. Install Python 3.12: Ensure that Python 3.12 is installed on your Windows machine. You can download it from the official Python website. Remember to check the option to add Python to your PATH during installation.
2. Install MySQL Server: Download and install MySQL Server from MySQL's official site. During installation, set up your root password and remember it.
3. Install MySQL Workbench: Download and install MySQL Workbench from here. 4. Install Required Python Libraries: You will need mysql-connector-python for connecting to MySQL and possibly pandas for data transformation. You can install them using pip:

Step 6: Create database
CREATE DATABASE exp10;
USE exp10;

```
CREATE TABLE transformed_data ( id INT
AUTO_INCREMENT PRIMARY KEY, name
VARCHAR(100),
value INT,
transformed_value INT
);
SELECT * FROM transformed_data;
```

Step 7:

pip install mysql-connector-python pandas pip install pyarrow #will improve the performance

of certain Pandas operations. Step 8: Set Up

Your MySQL Database

1. Open MySQL Workbench.
2. Connect to the MySQL Server using the root user or any other user you've created. 3. Create a New Database:

○ In the Navigator panel, right-click on Schemas and select Create Schema. ○ Name your schema (e.g., etl_pipeline) and click Apply.

4. Create Tables: You need to create tables for storing your extracted and transformed data. You can execute SQL commands in the SQL Editor.

Step 9: Extract Data from Source

You can extract data from various sources like CSV files, APIs, or other databases. For this example, we'll extract data from a CSV file.

Create a Sample CSV File: Create a CSV file (e.g., 1.csv) with the following content:

```
name,value
Ram,100
Sham,150
Radha,200
Durga,150
```

Step 10: give file name ETL.py

```
import mysql.connector import
pandas as pd

# Extract
#Windows path df = pd.read_csv(r'C:\Users\admin\Downloads\1.csv') df =
pd.read_csv('/project/data/1.csv') # docker unable to access local file so that option to
#mount your local directory into the container.

# Transform
```

```
df['transformed_value'] = df['value'] * 2


# Load
connection = mysql.connector.connect(
# host='localhost',
host='host.docker.internal', # Docker not support direct local host
user='root', # Replace with your MySQL username  password='das@11',
# Replace with your MySQL password  database='exp10'
)

cursor = connection.cursor()

for index, row in df.iterrows():
cursor.execute("INSERT INTO transformed_data (name, value, transformed_value) VALUES
(%s, %s, %s)",(row['name'], row['value'], row['transformed_value']))  query

= "SELECT * FROM transformed_data"

# Step 3: Use pandas to execute the query and store the result in a DataFrame df =
pd.read_sql(query, connection)

# Step 4: Print the DataFrame or work with the data  print(df)


connection.commit() # Commit the transaction
cursor.close()  connection.close()
```

Step 11: Create dockerfile with .txt extension in notepad

```
# Use the official Python Alpine image  FROM
python:alpine

# Set the working directory in the container
WORKDIR /project

# Copy the current directory contents into the container at /project
COPY . /project

# Install dependencies if you have a requirements.txt file  RUN
pip install -r requirements.txt

# Run the Python script when the container launches  CMD
["python", "1.py"]
```

Step 12: Create requirements.txt file in notepad Mentioned

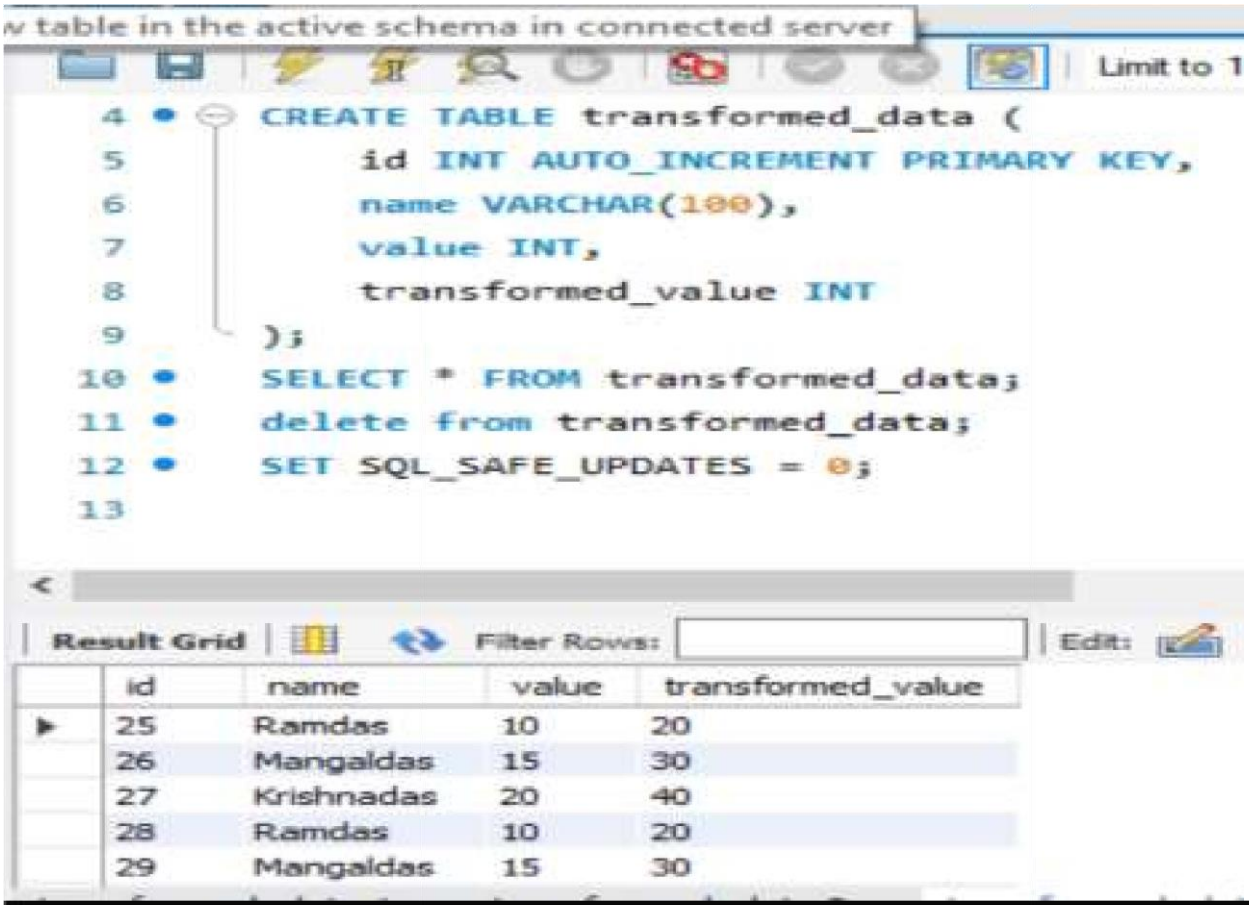
below given python library files

mysql-connector-python pandas

Step 13:Go on CMD prompt Run command
C:\code>docker build -t djimage -f dockerfile . #docker build the image name djimage

C:\code>docker run djimage # it will throws error bcz 1.csv yet to mount in container C:\code>docker run -v C:/Users/admin/Downloads:/project/data djimage

Step 14: Check data extract from 1.csv and transformed, loaded to SQL server




```
C:\code>docker run -v C:/Users/admin/Downloads:/project/data djimage
/project/1.py:32: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection
) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. P
lease consider using SQLAlchemy.
  df = pd.read_sql(query, connection)
   id        name  value  transformed_value
0  25      Ramdas     10                 20
1  26   Mangaldas     15                 30
2  27  Krishnadas     20                 40
```

Docker images

```
C:\code>docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
djimage       latest    c7db28a68a7c   2 hours ago      234MB
<none>        <none>    979e5f00ae35   2 hours ago      234MB
<none>        <none>    7b68f5958186   2 hours ago      234MB
<none>        <none>    9f13b766c472   2 hours ago      56.1MB
<none>        <none>    3ff9de44f365   2 hours ago      44.6MB
hello-world   latest    d2c94e258dcb   17 months ago    13.3kB
```

$sudo service docker stop
$sudo service docker start

Conclusion: Hence we study how to install docker and deploy projects in any environment.

Devops - Exp 8

1. Describe the architecture of Docker & explain every component in detail.

→ Docker architecture is a client_server architecture consisting of three main components. It is built to simplify the creation, deployment & management of applications in lightweight containers.

① Docker client : The user-facing interface that sends commands to Docker Daemon (via CLI or Rest API) common commands include docker run, docker pull & docker build.

② Docker Daemon (Docker Engine) : the background service that does the heavy lifting like creating, running & managing Docker containers. It listens to requests from the Docker client & handles container orchestration, image management & handles functions.

③ Docker objects : Images : Immutable templates for creating containers

Volumes : Persistent storage that containers can use to store & share data.

Networks : Allow communication between Docker containers & external systems.

2. Explain Nagios architecture in Detail.

→ Nagios architecture is based on a client-server model & includes following key components:

① Nagios core: The central server component responsible for scheduling & processing checks. It uses plugins to gather data from monitored devices. The core processes this data & generates alerts & reports.

② Plugins: Small progs that run on Nagios server to collect info about states of services.

③ Nagios Remote Pluggin Executes (NRPE): A pluggin that allows Nagios to execute plugins on remote hosts for services that can't be monitored directly.

④ Web interface: A user interface that provides access to system status, reports & logs.

⑤ Notification System: It sends all of alerts (via emails, SMS, etc.) when issues are detected.

24

Aditi Ambasta
60018220113
8004

DevOps
Assignment - 01

1. What is configuration management?

→ Configuration management (CM) tools help automate the process of tracking, management controlling software configuration across different environment. They ensure that software remains functional & convinient when moved from one environment to another.

Popular CM tools

i) Ansible :- Open source automation tool used for CM and application deployement

ii) Chef :- Automates infrastructure management through code

iii) Salt Stack :- CM and Orchestration tool with real time event driven automation

iv) Terraform :- Infrastructure tool that manages cloud & resources.

2.

Iterative Model	Incremental Model
• focuses on repeating the development cycles to refine and enhance the product	• focuses on developing and delivering portion of the product seperately.
• Each iteration goes through entire SDLC	• Each increment is a stand alone part of final product.

- Feedback is incorporated in each iteration
- Delivery is often done at end of iteration cycle.
- Cycle refinement of the entire system

- feedback only integrated after increment.
- Delivery occurs with each increment & can affect subsequent increments.
- Linear development in small increments.

3. Containers

Lightweight portable encapsulation of app & their dependencies.

Shares the host OS

More efficient, uses less overhead

Significantly much faster

Less startup time

Virtualization

- Technology that allows multiple storage operating system to run on single machine
- Provides full isolation with its own OS.
- less efficient due to the more overhead due to running full OS instances.
- Slower to execute as the programs are heavier.
- Extremely high startup time due to the OS Boottime.

4.

Merge	Release
• Retains the original commit history, creating a merge commit	• Rewrite the commit history to create a linear history.
• Can handle conflicts during merge process & creates a merge commit that reflects conflict resolution.	• Conflict can be resolved for each individual commit when rebasing.
• Easier for collaborative workflows, especially with multiple developers	• Requires more care especially in shared more care, especially in shared branch as it rewrite commit history
• Branch history remains diverged, showing where branches split & then are re-merged	• creates a clean, linear commit structure, hiding the original branch divergence.
• Best for maintaing clean records of branch merges, preserving the history	• Best for creating a linear history in private branch or solo work.

5. Diff. Srum & Extreme Programming (XP).

Srum	Extreme Programming
• Agile framework for managing and completing projects	• Mainly focused on best engineering practices.

- Uses Sprint - typically 2-4 weeks long.
- Defines specific rules (Product Manager, owner), scrum master, team developers)
- Emphasizes Scrum ceremonies (Daily standups, Sprint reviews)
- Minimal documentation focuses mainly on the successful implementation of a working project

- Uses short iterations, often 1-2 weeks.
- fewer emphasized roles, emphatizes team work.
- Emphasizes technical practices (pair-programming test-driven optimization)
- Minimal with emphasis, on focus on code and even clear communication.

6.

Fetch	Pull
• Downloads changes from a remote repository to local repository without merging them.	• Downloads changes from a remote repository and even merges changes with local branch.
• To update local repo with remote changes without affecting dir.	• Update local repo while simultaneously integrating changes from remote repo
• No direct impact requires manual merge.	• Directly updates and even merge with local branch.
• Allows us to review changes before merge.	• combinations fetching and merging into single step
• eg:- git fetch origin	• git pull origin main.