

Register No: 21L3IA0505

Experiment No: 2

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
	Total	10	
Date		Signature of the Lab teacher	

AIM: To install and configure Jenkins to test, and to deploy Java or web applications using NetBeans or eclipse.

Jenkins is an open source automation tool written in java with plugins built in and on continuous integration purpose. Plugins allow integration of various devops stages.

JENKINS INSTALLATION:-

- Install Java Development Kit (JDK)
- Set the path for the environmental variables for JDK
- Install Jenkins

Register No. : []

Experiment No. : []

Date: []

- * Browse to the official Jenkins download page. Click the windows link to begin the download.
- * Once the download is complete, run the Jenkins.msi installation file.
- * The setup wizard starts click next to proceed
- * Select the install destination folder and click next to continue.
- * Then click on Run service as local system (Not-recommended)
- * Enter the port number you want Jenkins to run on. Click Testport to check is the selected port is available, then click next to proceed.
- * select the directory where Java installed on your system and click next to proceed.
- * Select the features you want to install with Jenkins and click

Register No : []

Experiment No : []

Date: []

next to continue.

- * Click install to start the installation process
- * Once the installation is complete, click to exit the install wizard
- * Once Jenkins is installed, explore it open the web browser and type "localhost:5000".
- * Enter the credentials and log in If you install Jenkins for the first time, the dashboard will ask you to install the recommended plugins.
- * Install all the recommended plugins
- * Open Jenkins → manage Jenkins → System
- * Open Jenkins → manage Jenkins → Security → continuous global security
first authentication strategy
- LDAP integration

Register No :

Experiment No :

Date:

Lets say we have created multiple Jenkins or pipelines in Jenkins. All these belong to different teams and organizations.

- * Open Jenkins → manage Jenkins → system configuration ↗
Manage plugins
5 options are available as
 - plugins which are available (Azure, AWS cloud)
 - plugins which are installed (already present)
 - plugins which are advanced
(Here you can create your own plugins)
- * How to create a job / item in Jenkins?
open Jenkins → click on the new item
→ job1 → Freestyle project → ok.
build → execute shell

Register No :

Experiment No :

Date:

Command → echo "hello from our save.
Build now

Register No: 2113140505

Experiment No: 3

Date:

S. No	Component	Max. Marks	Marks Secured
1	Preparedness	2	
2	Viva-Voce	2	
3	Experiment	3	
4	Analysis & Record	3	
	Total	10	
Date		Signature of the Lab teacher	

AIM: To perform version control on websites or software using different version control tools like RCS/CVS/GIT/Mercurial

GIT: It is a version control system that allows developers to track changes in their code.

GITHUB: Github is a web-based hosting service for git repositories.

REPOSITORY:

A directory (or) storage space where your projects can be local to a folder on your computer, or it could be a storage space on

Github

Register No :

Experiment No :

Date:

There are 2 types of repositories

- Central repository
- Local repository

GITHUB AND GIT

version control:

version control is the management of changes to documents, computer programs, large websites and other collection of it. These changes are usually termed as "versions".

Register No : []

Experiment No : []

Date: []

WHY VERSION CONTROL

→ Collaboration

→ storing versions

i) snapshots of all versions are properly documented and stored.

ii) versions are also named accurately.

→ Backup

In any case if your central server crashes, a backup is always available in your local

version control system tools

→ git

→ subversion (SVN)

→ CVS (Concurrent version system)

→ mercurial

but in centralized VCS local copy is not available, so, mostly everyone prefer distributed system where you can find local copy.

FEATURES OF GIT

→ distributed → speed

Register No : []

Experiment No : []

Date: []

- Compatible → open source
- Non-linear → reliable
- Branching → secure
- Lightweight → economic.

Installation of github & working

- First open github.com
- Signup for github
- Click on start a project.
- Repository name (git-github)
 - Public
- Initialize repository with a README
- Then click create repository
- If you want to change something in README file, click on it, Then click on edit, Then write something, then click on commit changes.
- Install git for windows
- open one folder in C drive as project.
- Right click, click on Git bash here

Register No: _____ Experiment No: _____ Date: _____

It will open one terminal, which is called as git bash emulator.

→ \$git init

git folder has been created which contain all kind of configuration.

→ \$git remote add origin "Central repository link"

→ \$git pull origin main

one readme file will be there in local repository.

→ If you want to change something in local repository, then you have to add those files in the index first.

→ open one text file or edul.txt write "Hello world" It will show the workspace but that edul.txt will not be in the local repository, because we have not commit that.

→ \$git status

edul.txt (untracked files means which are not added in index yet)

Register No : []

Experiment No : []

Date: []

→ \$ git add edul.txt (added)

→ \$ git status

changes to be committed.

→ \$ git commit -m "adding first commit in local repo"

so, this is the changes are finally committed in local repository.

→ Again, we will create two or more files as edu2.txt
edu3.txt

→ \$ git status
untracked files (edu2.txt, edu3.txt)

→ Now do some changes in first file
i.e. edutxt

→ \$ git status
modified: edul.txt
untracked:
files edu2.txt
 edu3.txt

→ \$ git add -A (to add multiple files)

→ \$ git status
all the files are added.

Register No :

Experiment No :

Date:

→ \$ git commit -a -m "adding 3 files to
(all the files are committed to gne
3 files changed, 3 insertions(+)

→ \$ git log

It gives exact time , year

BRANCHING

→ \$ git branch first branch ↴

now this first branch contains all the files
from master branch.

→ \$ git checkout first branch

It will switch to first branch

→ \$ ls

Readme.md edul.txt eduz.txt eduz3.txt