|  |  |  |
| --- | --- | --- |
| **INDEX** | | |
| **SL NO.** | **PROGRAMS** | **PAGENO.** |
| 01 | Explain Git tool in Devops and install git tool and its implementation part. | 1 - 15 |
| 02 | Explain Jenkins tool in Devops and install Jenkins tool and explain the features of Jenkins and explain the features of Jenkins and implementation of Jenkins | 16 - 39 |
| 03 | Explain selenium tools in devops and install selenium tool and explain the features of selenium and implementation of selenium | 40 - 52 |
| 04 | Explain puppet tool in devops and its advantages and features of puppet | 52 -53 |
| 05 | Explain Ansible tool in devops and its advantages and features of Ansible | 53 |
| 06 | Explain Docker tool in devops and its advantages and features of Docker | 54 |
| 07 | Explain saltstack tool in devops and its advantages and features of Saltstack | 55 |
| 08 | Explain Chef tool in devops and its advantages and features of chef | 55-56 |

**1. Explain Git tool in DevOps and install git tool and its implementation part.**

Git is an open-source distributed version control system that is freely available for everyone. It is designed to handle minor to major projects with speed and efficiency. It is developed to ordinate the work among programmers.

The version control allows you to track and work together with your team members in the same workspace. It is used as a critical distributed version-control for the DevOps tool.

# Features

* It is a free open source tool.
* It allows distributed development.
* It supports the pull request.
* It enables a faster release cycle.
* Git is very scalable.
* It is very secure and completes the tasks very fast. **Step 1:**



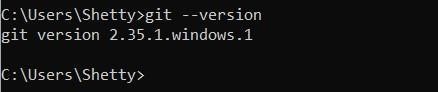
* Click first link (git-scm.com) [**https://git-scm.com/downloads**](https://git-scm.com/downloads)



* Click Windows option (Depending on your operating system)
* Click 64-bit Git for windows setup
* Download the software exe file

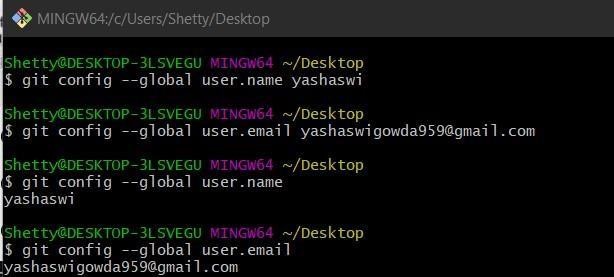


* Open Software exe file
* Click next – next for all the process
* Checking git version in command prompt



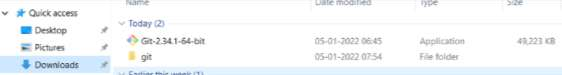
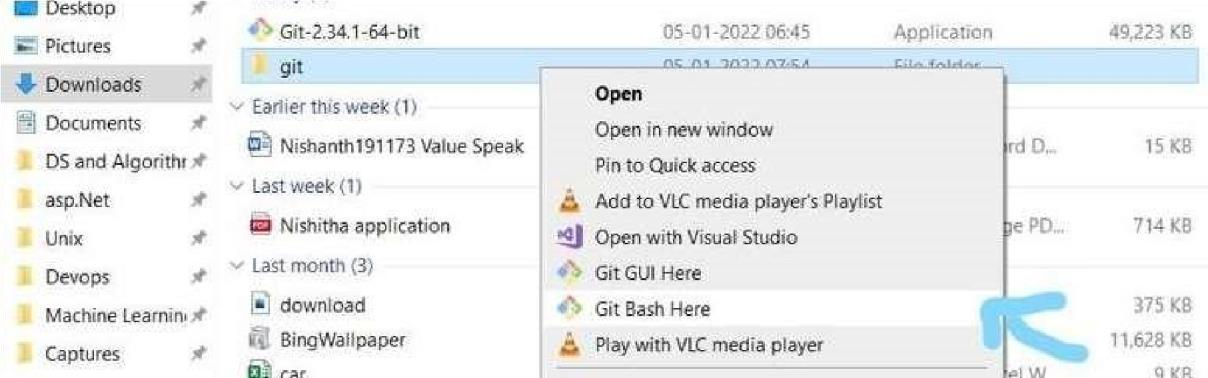
**Step 2:**

* Setting author name and email



**Step 3:**

* Create a file
* Open folder in Git Bash



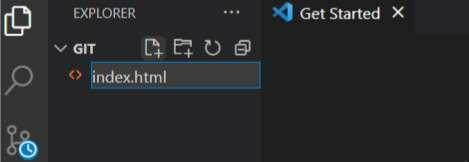
**Step 4:**

* Open folder in VS code



**Step 5:**

* Create a new file



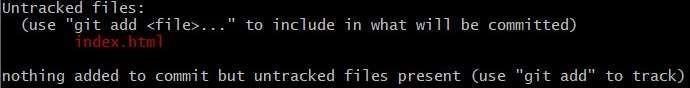
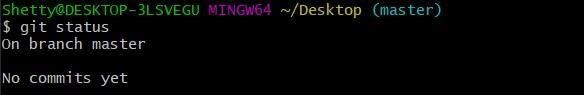
**Step 6:**

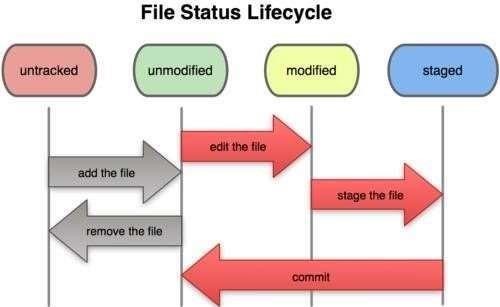
* Add Data
* Git init
* Initialize empty repository



**Step 7:**

* Git Status





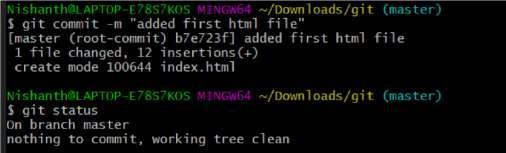
**Step 8:**

* Add file to the staged area Git add
* git add -A [ to add all files to the staged area]



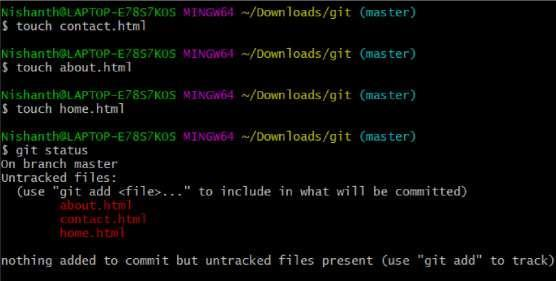
**Step 9:**

* Git Commit [Save]



**Step 10 :**

* Create a file in git
* Touch



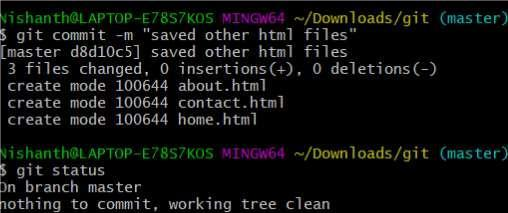
**Step 11:**

* Add all files to staged Area



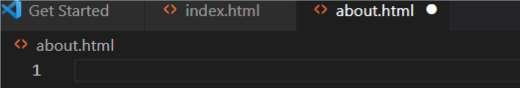
**Step 12:**

* Save all files [Commit]



**Step 13:**

* Git notifies



* when data changes in the file



**Step 14:**

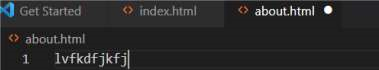
Git checkout command

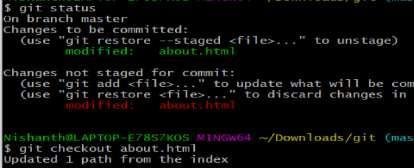
* it matches last commit changes
* git checkout -f [ to commit all the files]



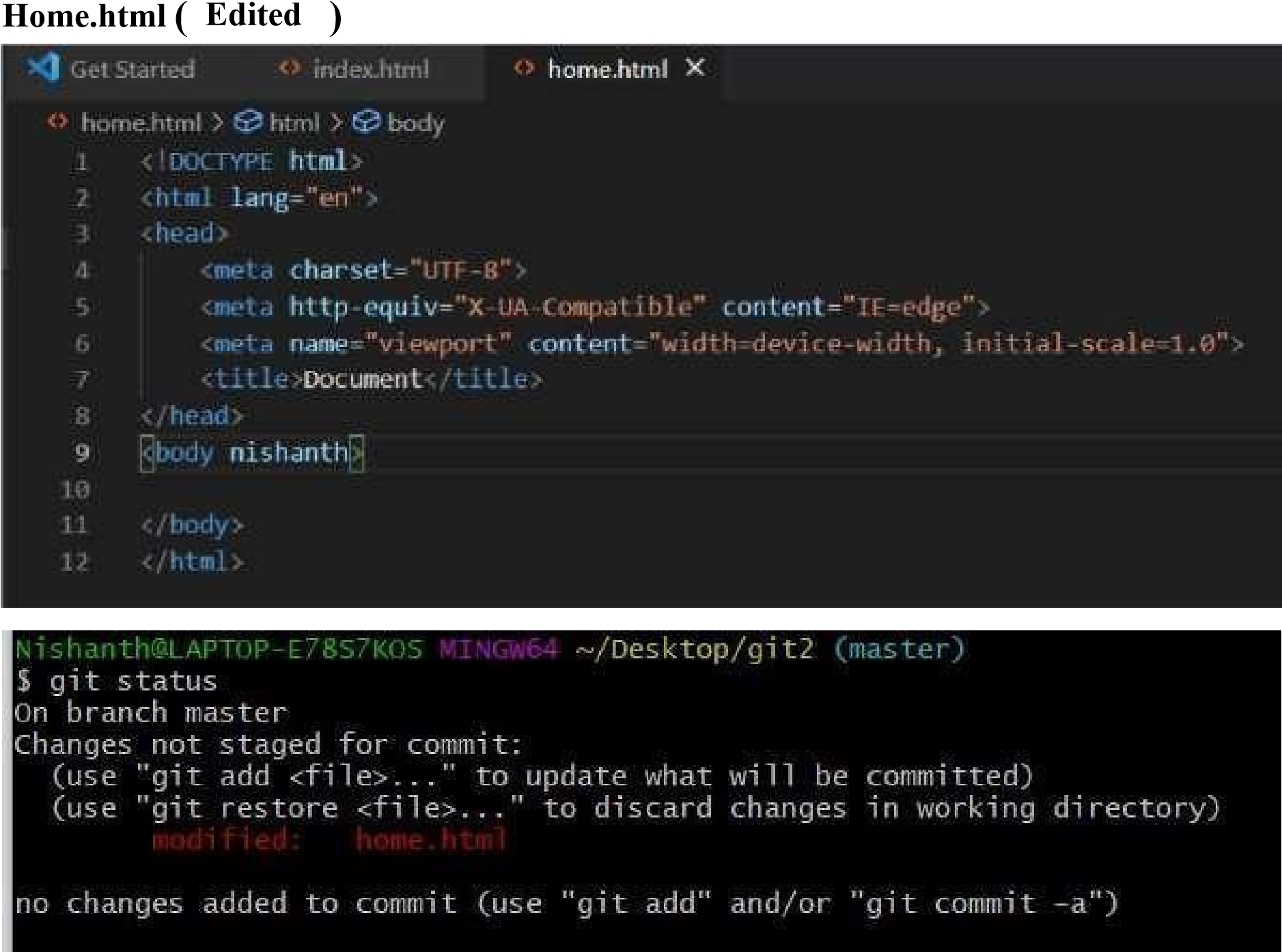
# Git diff

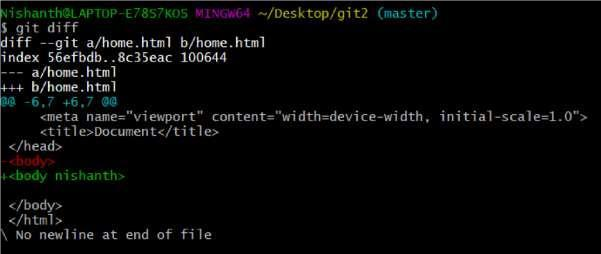
* displays the differences between files in two commits or between a commit and your current repository
* git-diff - Show changes between commits, commit, and working tree. Home.html







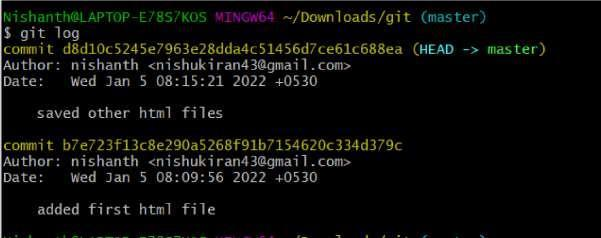


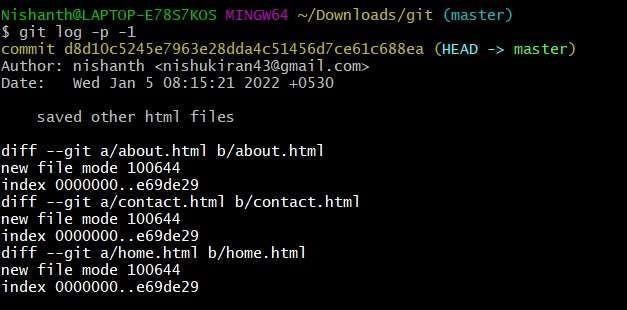


**Step 15:**



Git log





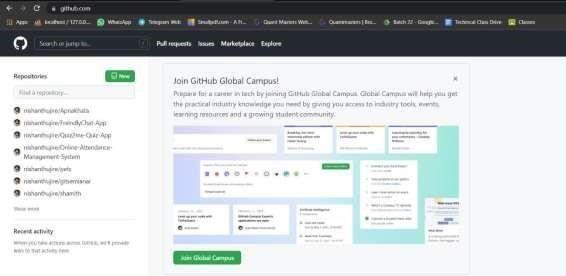
**Step 16:** ls

* it shows all the files in the directory



**Step 17:**

# GitHub

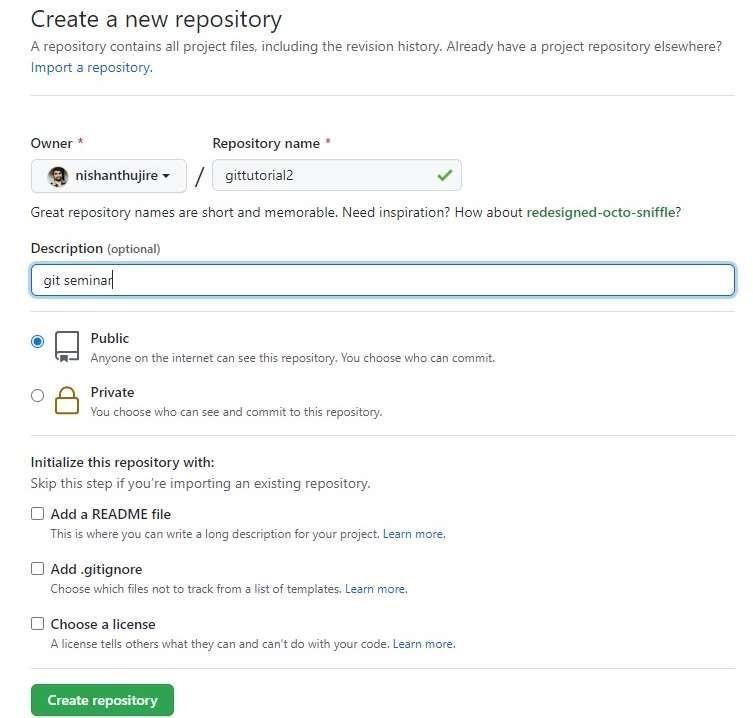


**Step 18:**

* Click plus symbol to create a new repository

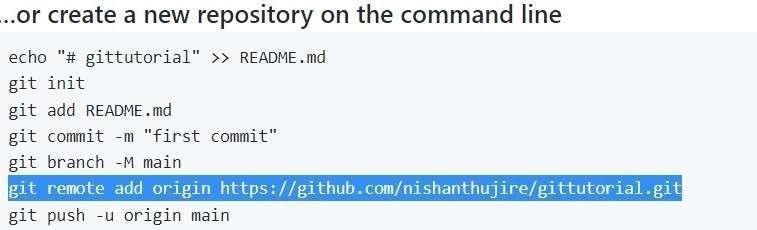


* Click Create repository button to create a gittutorial2 repository



**Step 19:** copy repo

URL





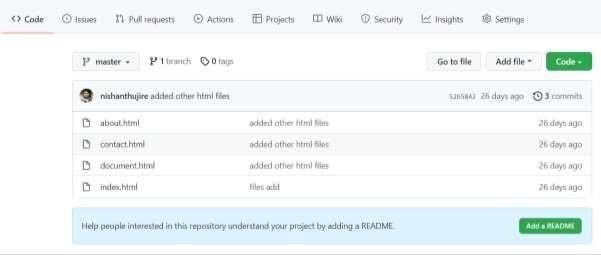
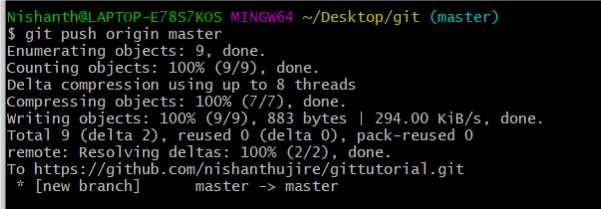
**Step 20:**

* Connecting local repository to remote repository

**Step 21:**

Git push

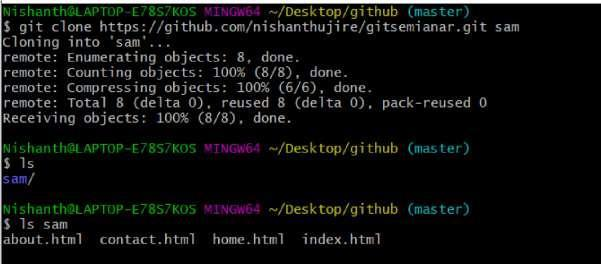
* The git push command is used to upload local repository content to a remote repository



**Step 22:**



Git Clone



# 2. Explain Jenkins tool in DevOps and install Jenkins tool and explain the features of Jenkins and explain the features of Jenkins and implementation of Jenkins

Jenkins is an open-source automation tool written in Java programming language that allows continuous integration.

Jenkins builds and tests our software projects which continuously makes it easier for developers to integrate changes to the project, and makes it easier for users to obtain a fresh build. It also allows us to continuously deliver our software by integrating with a large number of testing and deployment technologies. **Features**

* It is an open-source tool.
* It is free of cost.
* It does not require additional installations or components. Means it is easy to install.
* Easily configurable.
* It supports 1000 or more plugins to ease your work. If a plugin does not exist, you can write the script for it and share it with the community. • It is built in java and hence it is portable.
* It is platform-independent. It is available for all platforms and different operating systems. Like OS X, Windows or Linux.
* Easy support, since its open-source and widely used.
* Jenkins also supports cloud-based architecture so that we can deploy Jenkins in cloudbased platforms.

•

**Step 1:**

# Install Java Version 8

To download the Java [**Click here**. S](https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)elect the file according to your platform.



## Download Jenkins war File

* This war is required to install Jenkins.
* The official website for Jenkins is<https://jenkins.io/>



Click on the Download button.



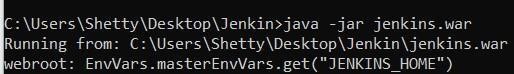
* Click on Generic Java Package (.war) to download the Jenkins war file.

**Step 2:**

* Open the command prompt and go to the directory where the Jenkins.war file is located.

And then run the following command:

* C:/Java -jar Jenkins.war
* When you run this command, various tasks will run, one of which is the extraction of the war file which is done by an embedded webserver called winstone.



Once the processing is complete without major errors, the following line will come in the output of the command prompt.

INFO: Jenkins is fully up and running



Click on **Allow access** button to allow access.

**Step 3:**

Unlocking Jenkins

* When you first access a new Jenkins instance, you are asked to unlock it using an automatically generated password.
* Browse to localhost:8080 (or whichever port you configured for Jenkins when installing it) and wait until the Unlock Jenkins page appears.
* From the Jenkins console log output, copy the automatically-generated alphanumeric password (between the 2 sets of asterisks).
* On the Unlock Jenkins page, paste this password into the Administrator password field and click Continue.



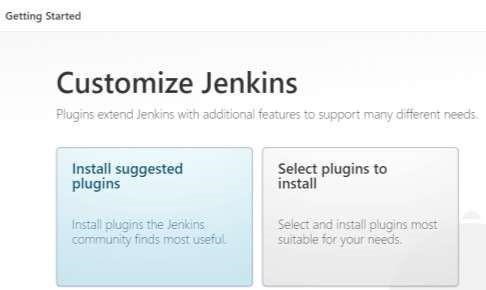
Then click on continue.

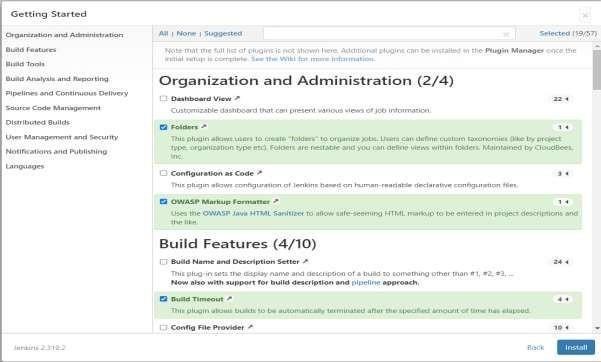
**Step 4:**  Customizing Jenkins with plugins

After unlocking Jenkins, the Customize Jenkins page appears. Here you can install any number of

Click one of the two options shown:

* Install suggested plugins - to install the recommended set of plugins, which are based on the most common use cases.
* Select plugins to install - to choose which set of plugins to initially install. When you first access the plugin selection page, the suggested plugins are selected by default.





Click on install

**Step 5:**

## Creating the first administrator user

Finally, after customizing Jenkins with plugins, Jenkins asks you to create your first administrator user.

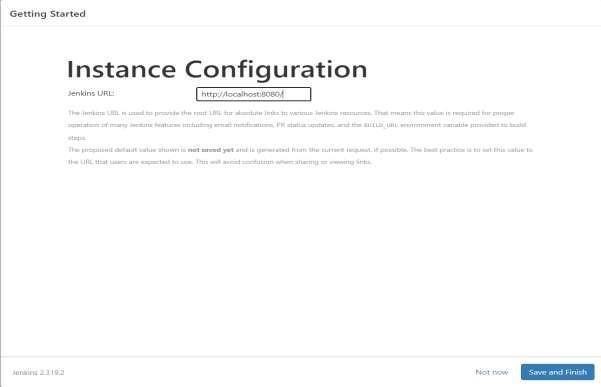
1. When the Create First Admin User page appears, specify the details for your administrator user in the respective fields and click Save and Finish.
2. When the Jenkins is ready page appears, click Start using Jenkins. Notes:

* This page may indicate Jenkins is almost ready! instead and if so, click Restart.
* If the page does not automatically refresh after a minute, use your web browser to refresh the page manually.

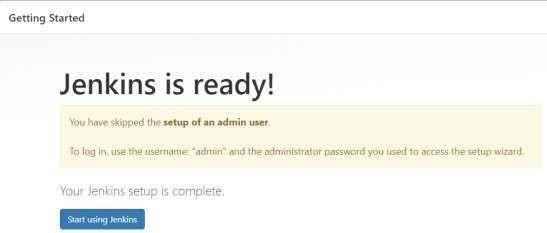
3. If required, login to Jenkins with the credentials of the user you just created and you are ready to start using Jenkins!



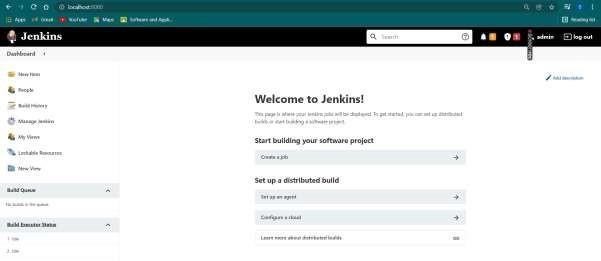
* If you create a new user you will be continued as a new user or Skip this and continue as admin.



* Click on save and finish



* Click on start using Jenkins



* Jenkins home page.



**Jenkins home directory:** Back up the folder, and you’ve backed up the entire server. Delete this folder and the next time you run the Jenkins WAR file it will assume it’s a brand new installation and configure itself from scratch. The importance and significance of the Jenkins

Home folder can not be.jenkins home directory path

C:\Users\Owner\.jenkins

Example : C:\Users\shetty\.jenkins



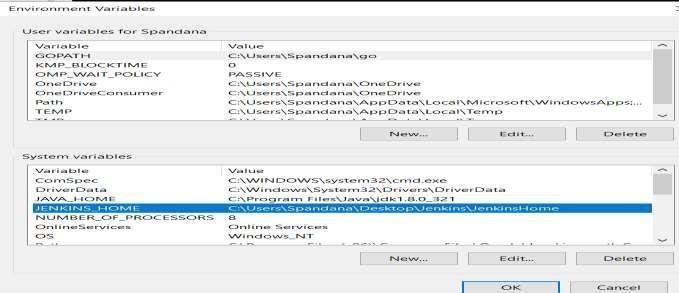
**How to change the home directory?**

Step 1: Check your current home directory

* 1. Navigate to your folder where Jenkins.war is present.
  2. Java -jar jenkins.war

Step 2: Create a new folder (Which will be a new home directory) Step 3:Copy all data from the old directory to the new directory.

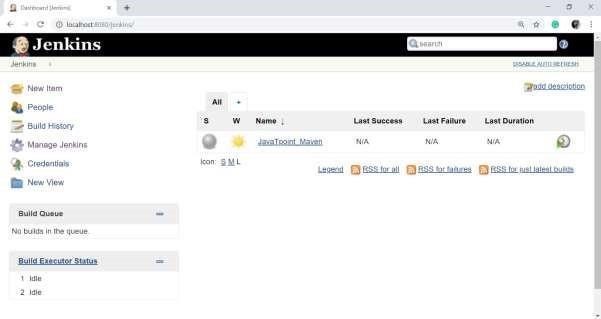
Step 4: Change environment variable – JENKINS\_HOME and set to new dir



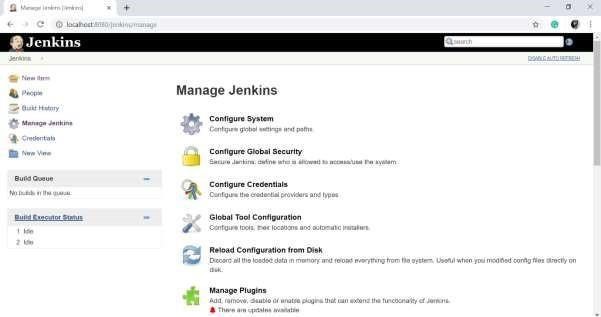
Restart the Jenkins

Jenkins Configuration

To configure the Jenkins, click on the 'Manage Jenkins' menu option from the left-hand side of the Jenkins Dashboard screen.

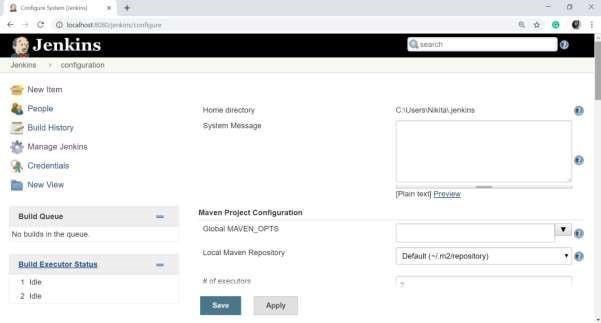


On the Manage Jenkins page, you will see the following options:



# Configure System

Click on the 'Configure System'.



The configure system page is a critical configuration part. This screen represents a variety of sections, each correlating to a different configuration area from generic Jenkins settings, global environment variables, and most installed plugins are configured on this page.

## Home directory

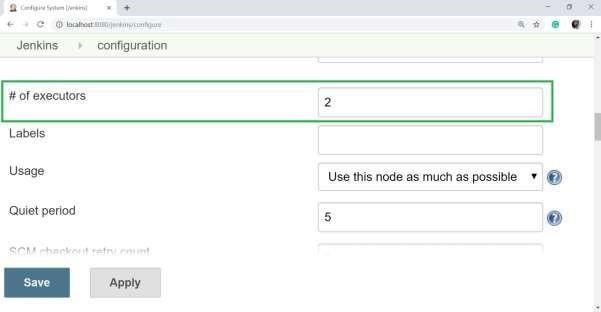
Jenkins requires some disk space to perform builds and keep archives. You can check this location from the configuration screen of Jenkins.

By default, this is set to ~/.jenkins, and this location will be initially stored within your user profile (such as C:\Users\Nikita\.jenkins) location.



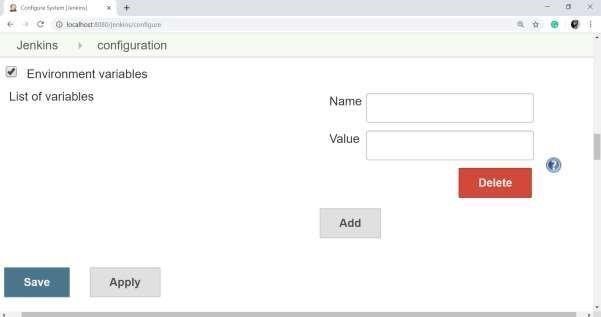
## # of executors

This option refers to the total number of concurrent job executions that can take place on the Jenkins machine. This can be changed based on the requirements. I recommended to you to keep this number the same as the number of CPUs on the systems for better performance.



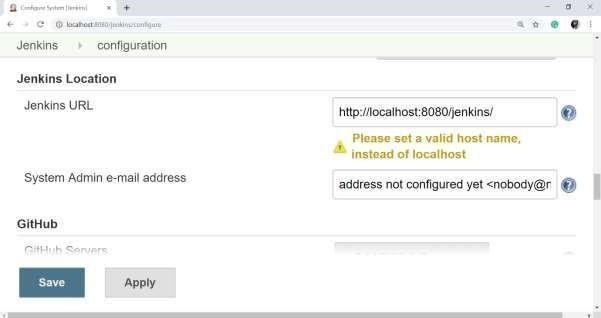
## Environment Variables

This option is used to add custom environment variables that will apply to all the jobs. Environment variables are key-value pairs and can be accessed and used in Builds wherever required. (For example SLACK\_TOKEN, SAUCE\_API\_KEY ).



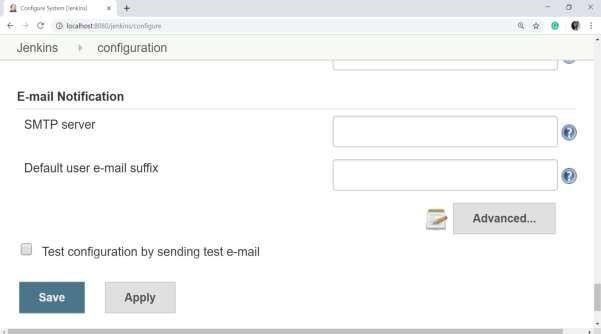
## Jenkins URL

By default, the Jenkins URL is set to the localhost. If you have a DNS (domain name setup) for your machine, set this to the domain name else overwrite localhost with the IP of the machine. This will help in setting up slaves (nodes) and while sending out links using the email as you can directly access the Jenkins URL using the environment variable JENKINS\_URL which can be accessed as ${JENKINS\_URL}.



## Email Notification

In the Email Notification section, you can configure the SMTP settings for sending out emails. This needs for Jenkins to connect to the SMTP mail server and send out emails to the recipient list.



# Getting starting with the Jobs

Jobs are the heart of Jenkins's build process. A job can be considered as a particular task to achieve a required objective in Jenkins. Every time you run a job, Jenkins compiles the job configuration inside the project workspace to perform the defined steps. Each run of this job is called a build and each step is called a build step.

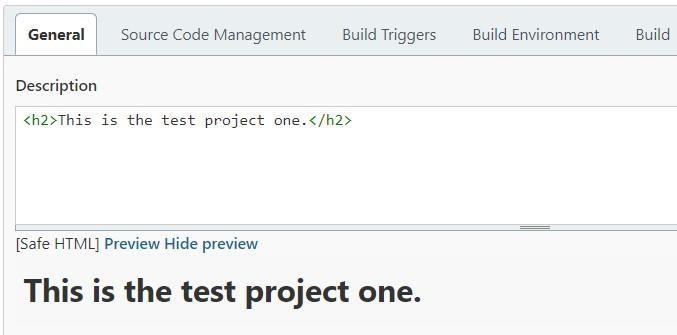
**Step 1** − Go to the Jenkins dashboard and click on New Item



**Step 2** − In the next screen, enter the Item name, in this case, we have named it Test1. Choose the ‘Freestyle project option’.



Step 3 − The following screen will come up in which you can specify the details of the job.



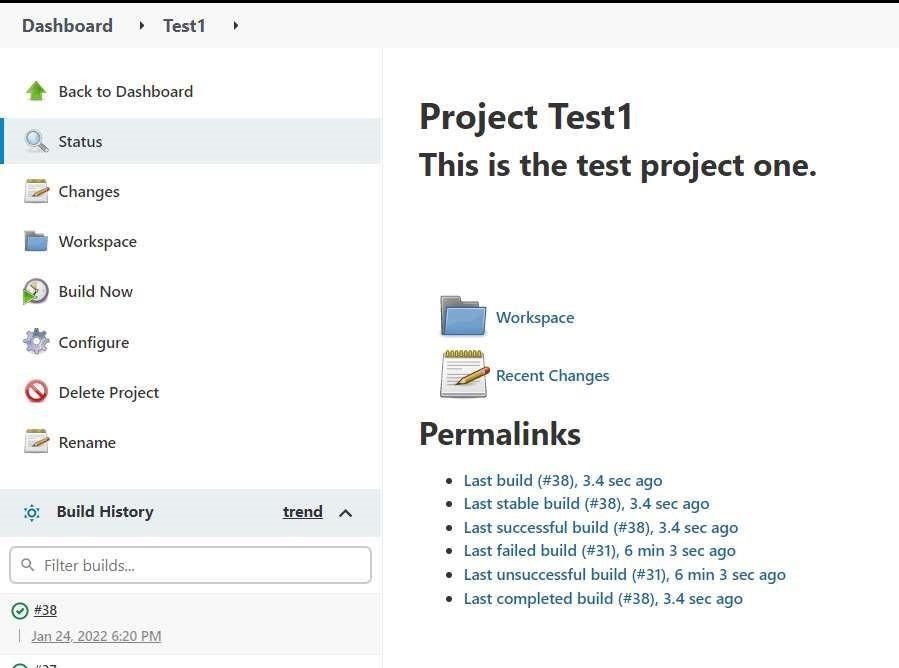
* Build takes place every 1 minute.



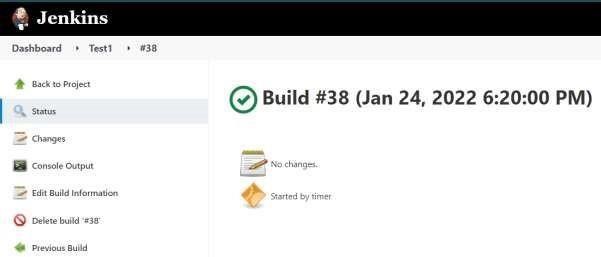
Green tick mark indicates that build is successful.



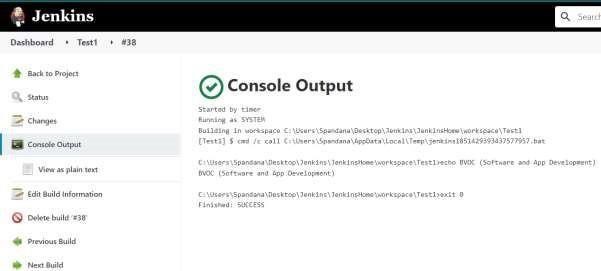




* Click on the build history.



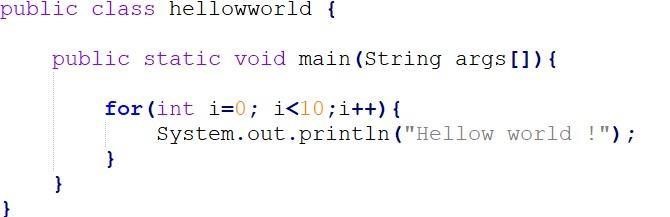
* Click on Console Output.



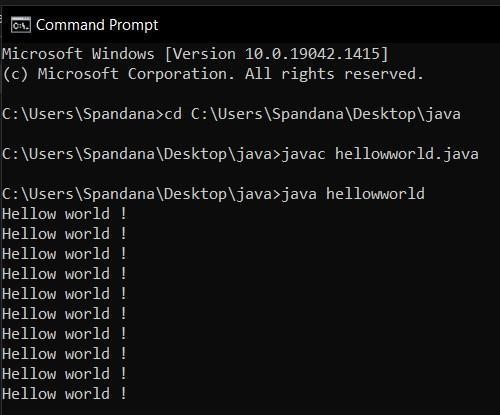
# Jenkins Integration with Git(SCM)

Step 1: Create a java program.

Example



* Create the new folder and place the java program file.
* Let us run the java program in the command prompt for testing purposes.



Step 2: Install Plugins and create a Jenkins job to run the program. Plugins to install

Open your dashboard -> Manage Jenkins -> Manage Plugins

* GIT Plugin
* GitHub Branch Source Plugin

To change the httpport use --httpPort=PORT\_NUMBER



Click on save.



Step 3: Add project to Git and GitHub.

1. Create a new repository on GitHub.

In Terminal, change the current working directory to your local project.

1. Initialize the local directory as a Git repository.

git init

Add the files to your new local repository. This stages them for the first commit.

git add . or:

git add --all

Commit the files that you've staged in your local repository.

git commit -m 'First commit'

Copy the remote repository URL field from your GitHub repository, in the right sidebar, copy the remote repository URL.

In Terminal, add the URL for the remote repository where your local repository will be pushed.

git remote add origin <remote repository URL> Sets

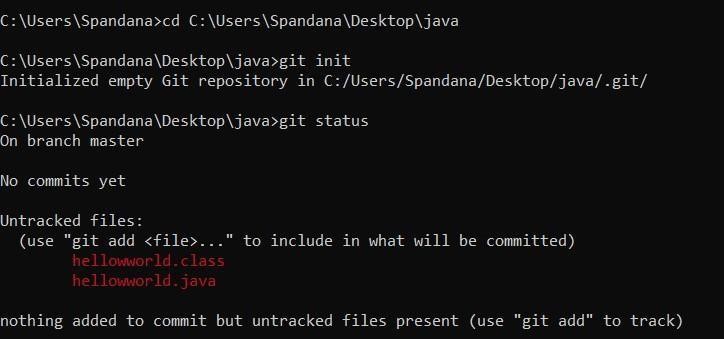
the new remote:

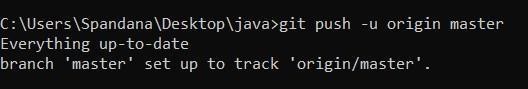
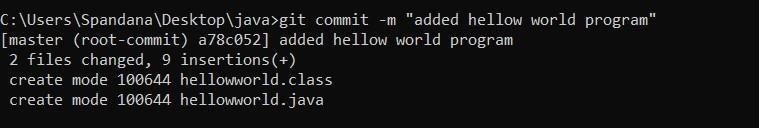
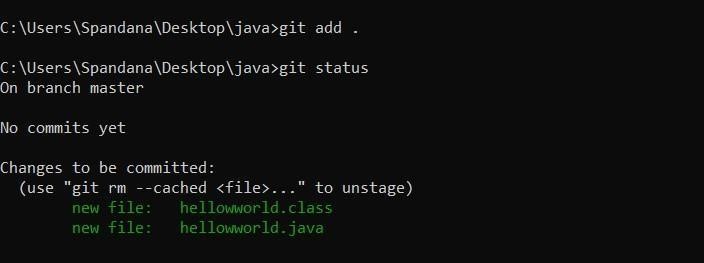
git remote -v

Push the changes in your local repository to GitHub.

git push origin master

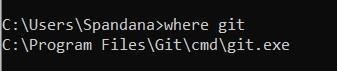
Pushes the changes in your local repository up to the remote repository you specified as the origin



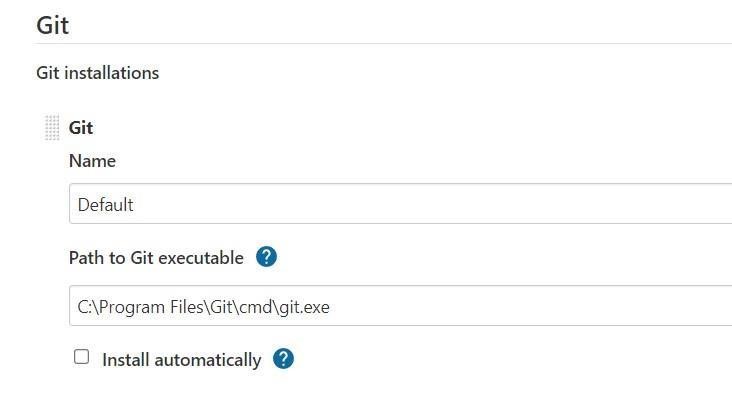


Step 4: Configure the project.

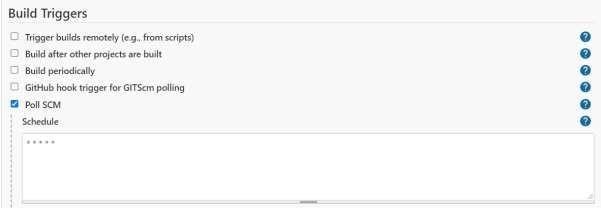
* Install the git plugin.
* Go to Manage Jenkins -> Manage Plugins -> install
* Run where git command on command prompt



* Go to Manage Jenkins -> Global Tool Configuration -> Git -> Path to Git executable > set the git.exe path.



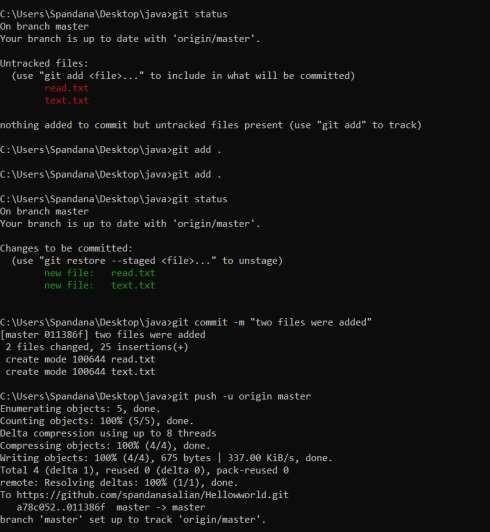
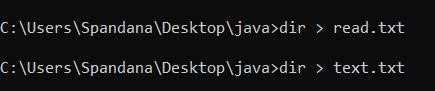
* Configure the Hellowworld project.



**Step 23:**

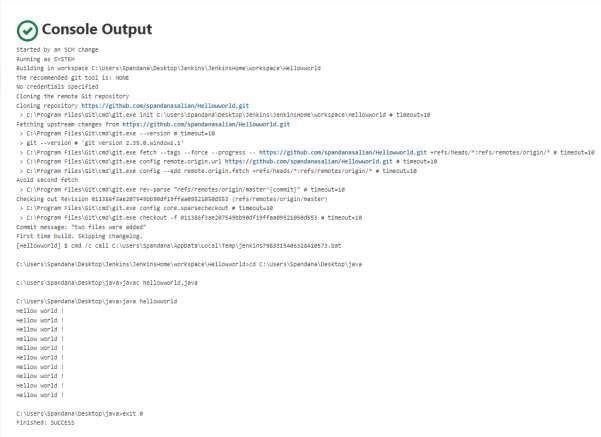
For testing

* Create a file in the Java folder in which the hellowworld java program is present.





* Build took place due to the changes made to the source code.



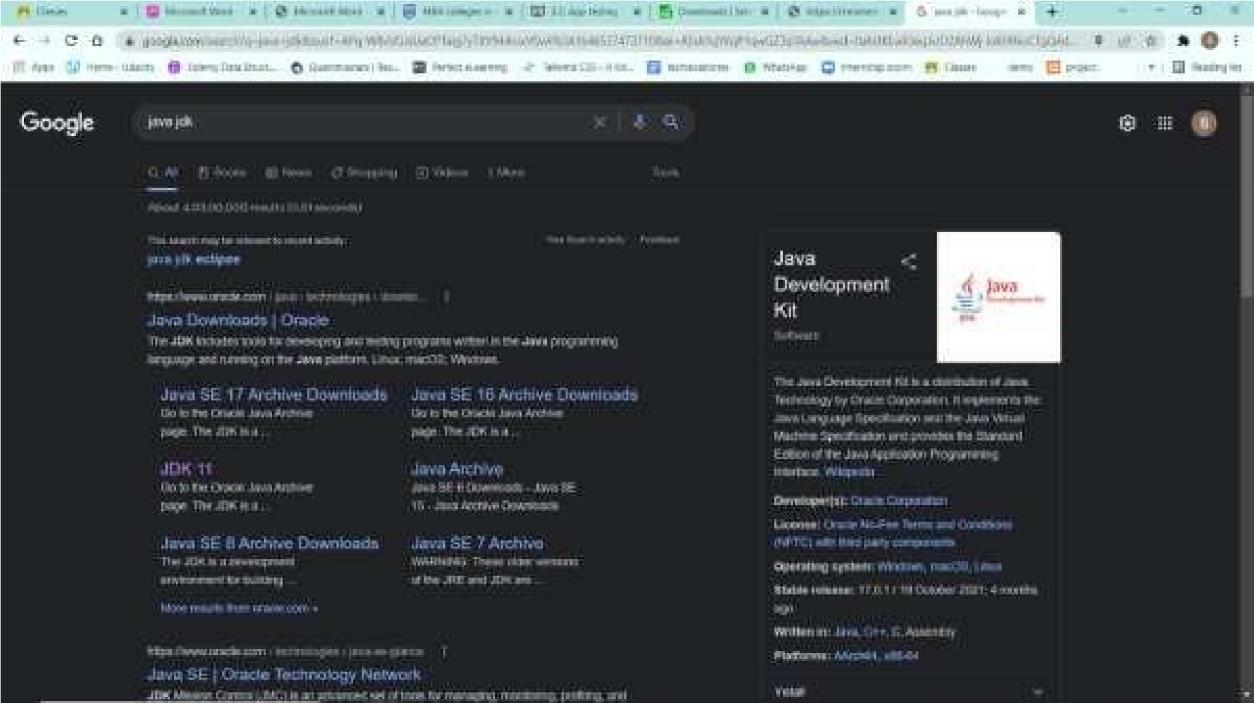
## 3.Explain selenium tools in DevOps and install selenium tool and explain the features of selenium and implementation of selenium

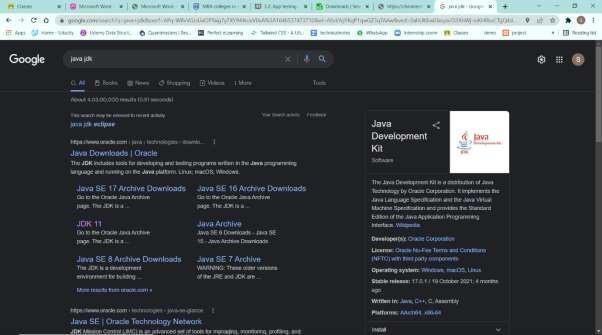
Selenium is a portable software testing framework for web applications. It provides an easy interface for developing automated tests. **Features**

* It is a free open source tool.
* It supports multiplatform for testing, such as Android and ios.
* It is easy to build a keyword-driven framework for a WebDriver. • It creates robust browser-based regression automation suites and tests.

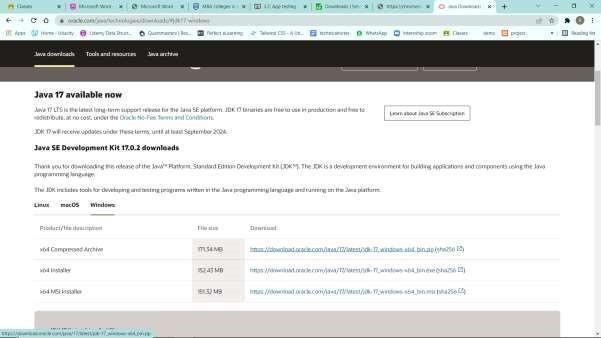
**Step 1.**

* Install java in our system
* Search as java JDK in your favorite browser

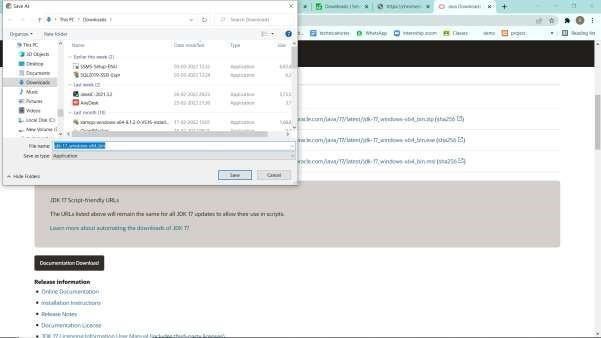




* Click the link mentioned above and you will be redirected to the below page.
* In the windows, section downloads the X64 windows downloader.

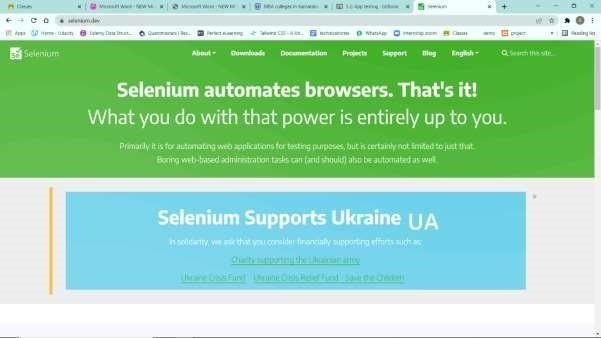


* Save the file and install the JDK.



**Step 2:**

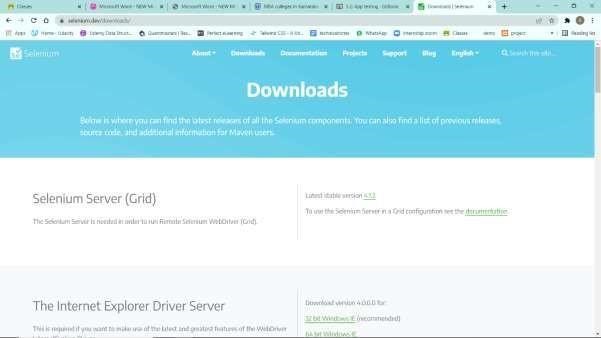
* Install the Selenium Grid :
* Open the selenium dev in your favorite browser and click on downloads.



* In downloads click the below-mentioned link and download the file and place it in the c folder.



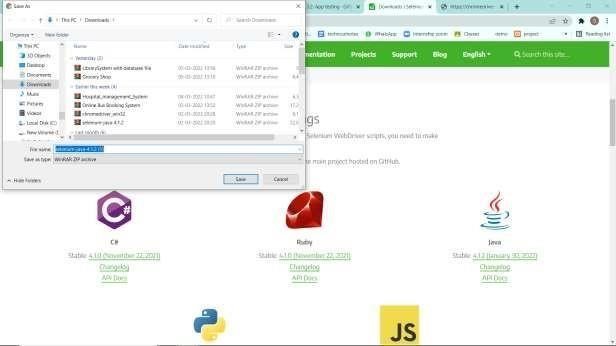
**Step 3 :**



* Download the java client programing language for the selenium.
* On the same page scroll down and click the below-mentioned link.

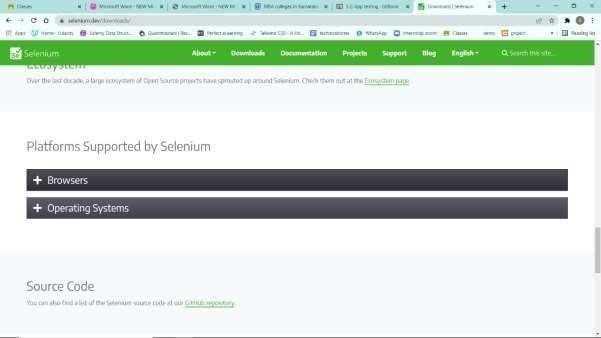


* Download the file and place it in the c folder same as of selenium grid as extracted.



**Step 4 :**

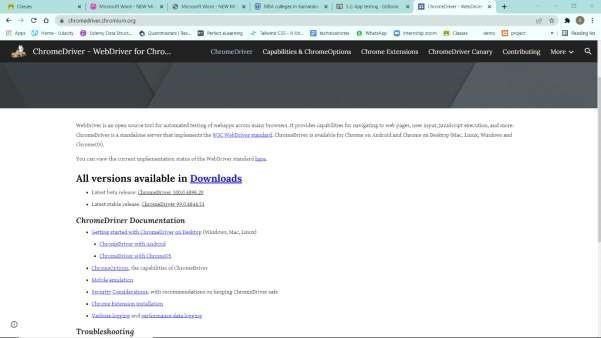
* Download the web driver for the selenium.
* Scroll down a bit on the same page and click the link below the mentioned figure.



* Download the driver of your choice and place it in the same folder as of selenium grid.



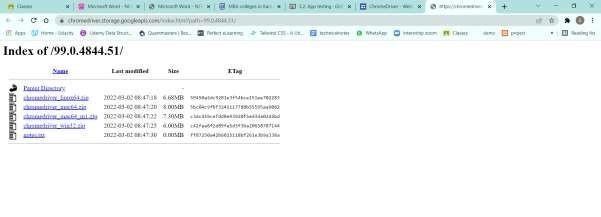
* Check your chrome version and download the respective web driver from the mentioned list



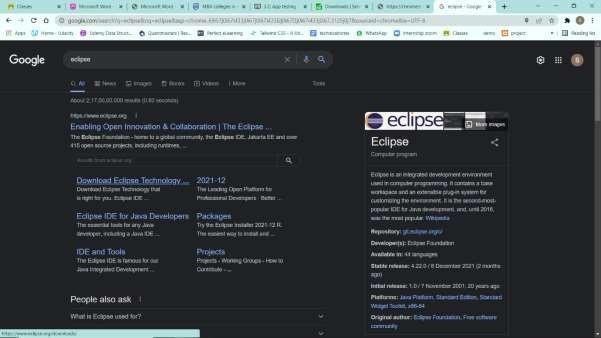
Download the zip file and extract to the selenium folder.



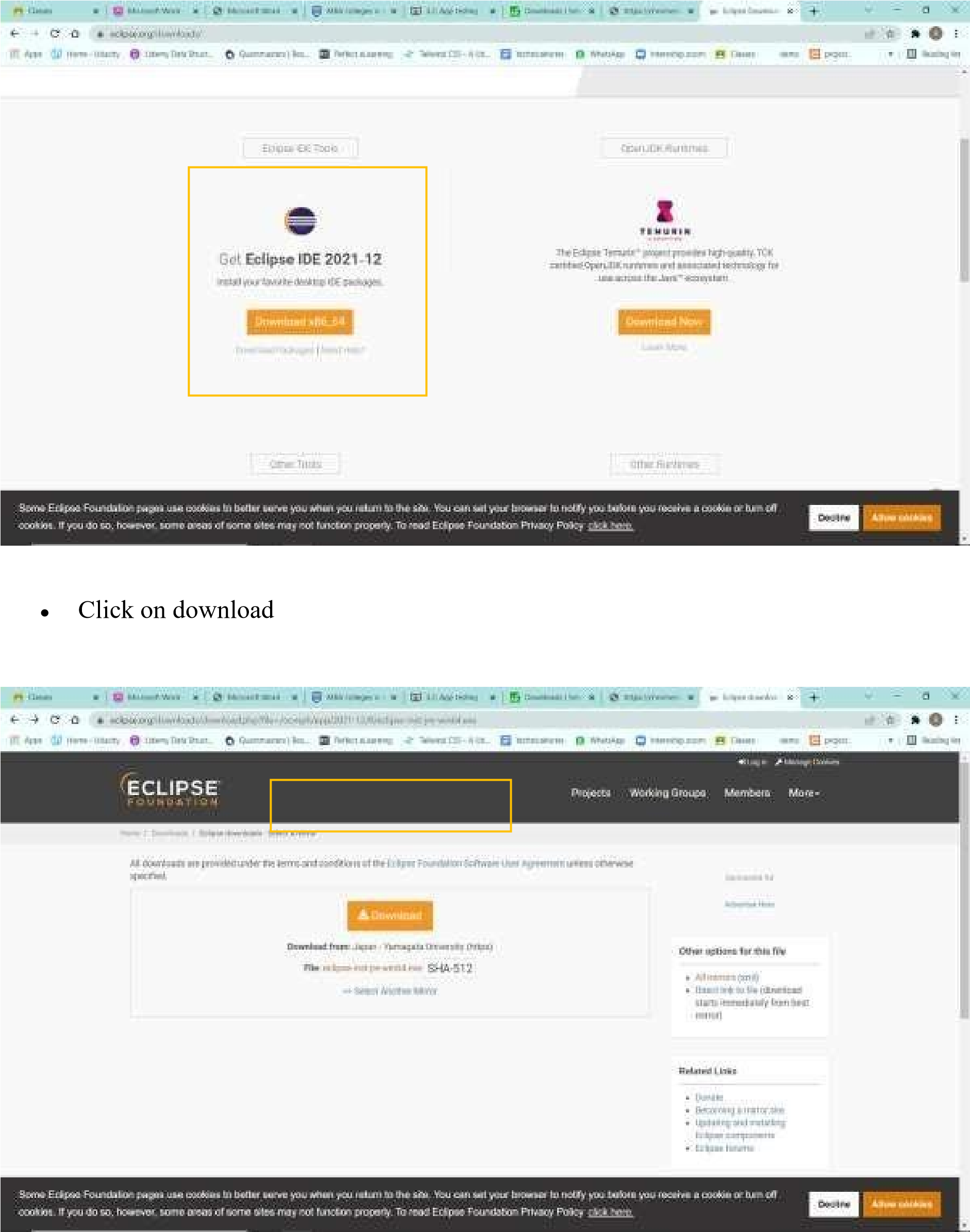
**Step 5:**



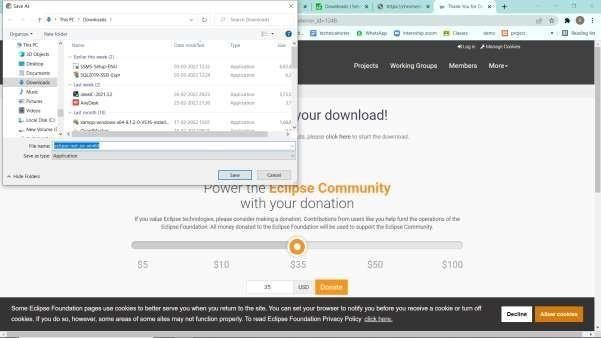
* Installation of the eclipse.
* Search the eclipse in your favorite browser.



Click the official website of the eclipse and click on the download X64 for windows.

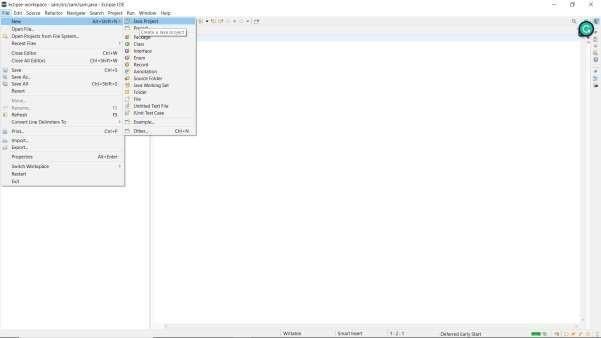


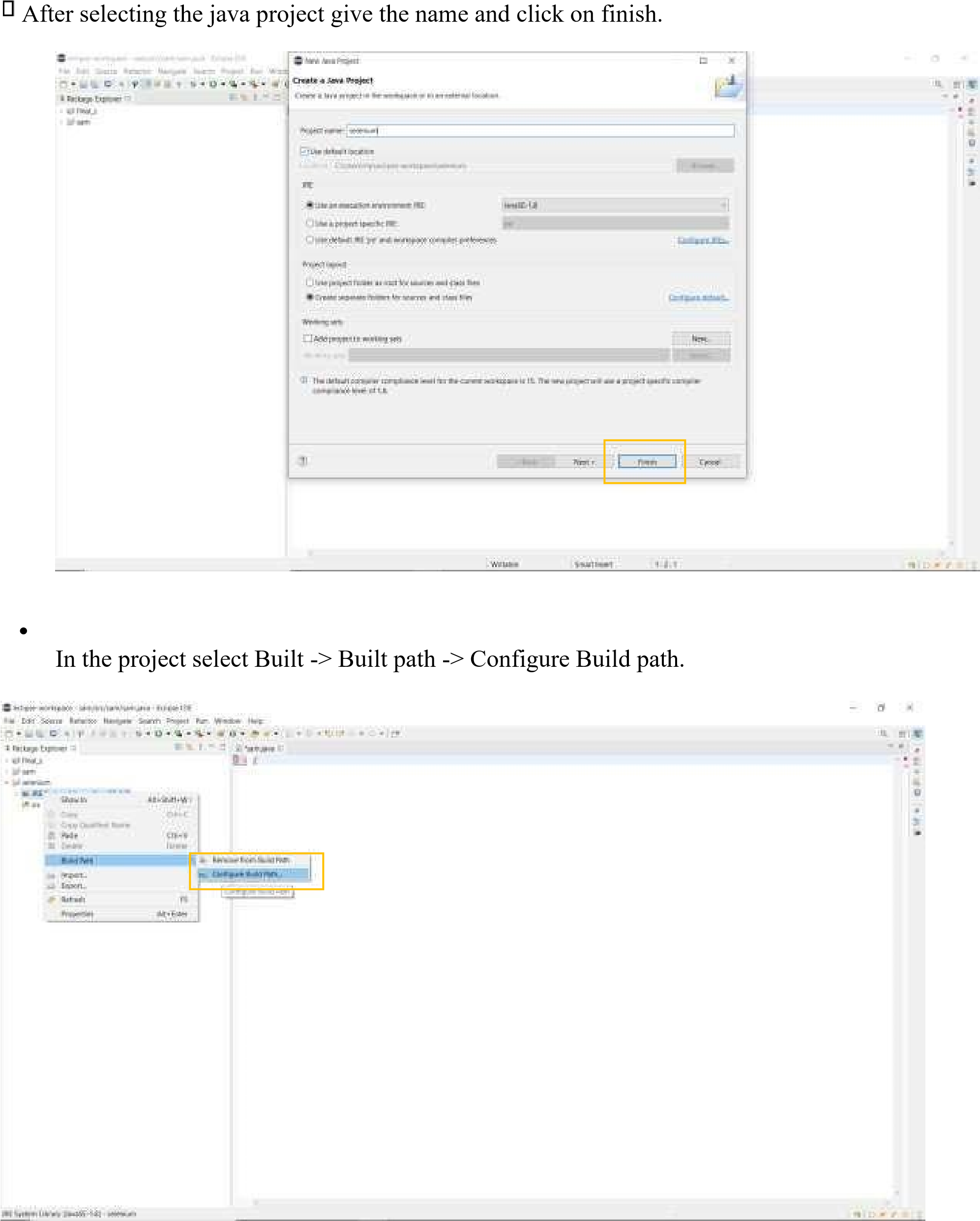
Specify the relevant path and install the ide in your system.



**Step 6 :**

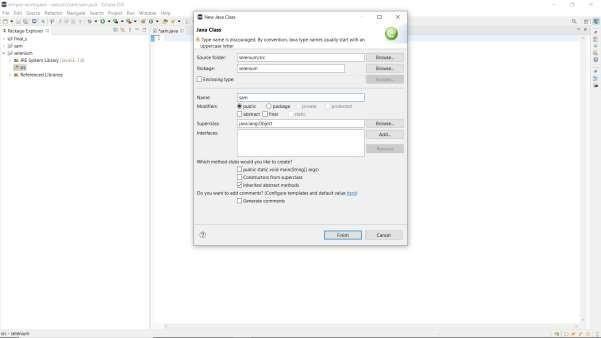
* Running the project.
* Open the ide and create the new project with installed java plugins and create a new project.







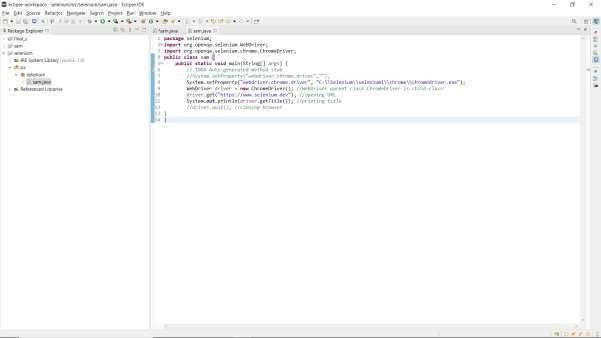
* Create a new class.



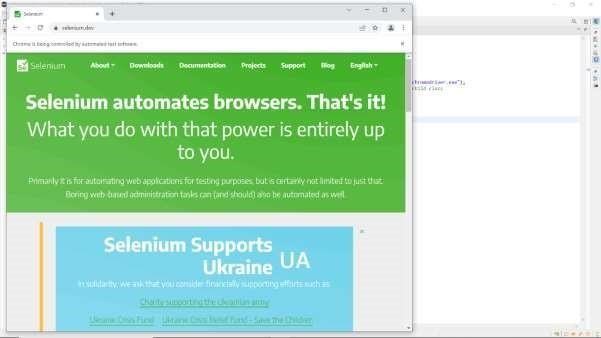
•



Apply the below code in the class.



* Run the project to automate the browser.



**4. Explain puppet tool in DevOps and its advantages and features of puppet**

* Puppet is the most widely used DevOps tool.
* It allows the delivery and release of the technology to change quickly and frequently. It has features of versioning, automated
* Testing, and continuous delivery. It enables to manage entire infrastructure as code without expanding the size of the team.

# Features

* Real-time context-aware reporting.
* Model and manage the entire environment.
* Defined and continually enforce infrastructure.
* Desired state conflict detection and remediation.
* It inspects and reports on packages running across the infrastructure.
* It eliminates manual work for the software delivery process.
* It helps the developer to deliver great software quickly.

**5. Explain Ansible tool in DevOps and its advantages and features of Ansible**

* Ansible is a leading DevOps tool.
* Ansible is an open-source IT engine that automates application deployment, cloud provisioning, and other IT tools. It makes it easier for DevOps teams to scale automation and speed up productivity.
* Ansible is easy to deploy because it does not use any **agents** or **custom** **security** infrastructure on the client-side, and by pushing modules to the clients. These modules are executed locally on the client-side, and the output is pushed back to the

Ansible server.

## Features

* It is easy to use to open source deploy applications.
* It helps in avoiding complexity in the software development process.
* It eliminates repetitive tasks.
* It manages complex deployments and speeds up the development process.

**6. Explain Docker tool in DevOps and its advantages and features of Docker**

Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

## Features

* It configures the system more comfortable and faster.
* It increases productivity.
* It provides containers that are used to run the application in an isolated environment.
* It routes the incoming request for published ports on available nodes to an active container. This feature enables the connection even if there is no task running on the node. • It allows saving secrets into the swarm itself.

### 7.Explain saltstack tool in DevOps and its advantages and features of Saltstack

Stackify is a lightweight DevOps tool. It shows real-time error queries, logs, and more directly into the workstation. SALTSTACK is an ideal solution for intelligent orchestration for the software-defined data center.

### Features

* It eliminates messy configuration or data changes.
* It can trace detail of all the types of web requests.
* • It allows us to find and fix the bugs before production.

**8. Explain Chef tool in DevOps and its advantages and features of chef**

A chef is a useful tool for achieving scale, speed, and consistency. The chef is a cloud-based system and open-source technology. This technology uses Ruby encoding to develop essential building blocks such as recipes and cookbooks. The chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.

 Chef has got its convention for different building blocks, which are required to manage and automate infrastructure.

### Features

* It maintains high availability.
* It can manage multiple cloud environments.
* It uses popular Ruby language to create a domain-specific language.
* The chef does not make any assumptions about the current status of the node. It uses its mechanism to get the current state of the machine.