**EXERCISE-6**

**Aim:** Write a build script to build the application using a build automation tool like Maven. Create a folder structure that will run the build script and invoke the various software development build stages. This script should invoke the static analysis tool and unit test cases and deploy the application to a web application server like Tomcat.

**What is Maven?**

Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout anda default build lifecycle.

In case of multiple development teams environment, Maven can set-up the way to work as per standards in a very short time. As most of the project setups are simple and reusable, Maven makes life of developer easy while creating reports, checks, build and testing automation setups. Maven provides developers ways to manage the following –

•Builds

•Documentation

•Reporting

•Dependencies

•SCMs

•Releases

•Distribution

•Mailing list

**What is TomCat?**

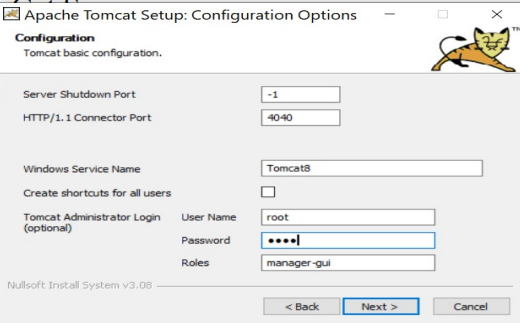
It is an open-source Java servlet container that implements many Java Enterprise Specs such as the Websites API, Java-Server Pages and last but not least, the Java Servlet. The complete name of Tomcat is "Apache Tomcat" it was developed in an open, participatory environment and released in 1998 for the very first time. It began as the reference implementation for the very first Java-Server Pages and the Java Servlet API. However, it no longer works as the reference implementation for both of these technologies, but it is considered as the first choice among the users even after that. It is still one of the most widely used java-sever due to several capabilities such as good extensibility, proven core engine, and well-test and durable. Here we used the term "servlet" many times, so what is java servlet; it is a kind of software that enables the webserver to handle the dynamic(java-based) content using the Http protocols.

**Installation of TomCat server:**

1.Open any web Browser and search for Tomcat 8 download. GO to the first link that appears on the webpage. Or click on the below link to download TomCat Server

**32-bit/64-bit Windows Service Installer (pgp, sha512)**

2.Once it has downloaded, double click on the .exe file and click on Yes then Next-> i agree -> next. When the following window appears give the port number as 4040 and username as root and password as root and then click on Next.



Once you have installed successfully go to any web browser and in the search bar give **localhost:4040** and click **enter**.

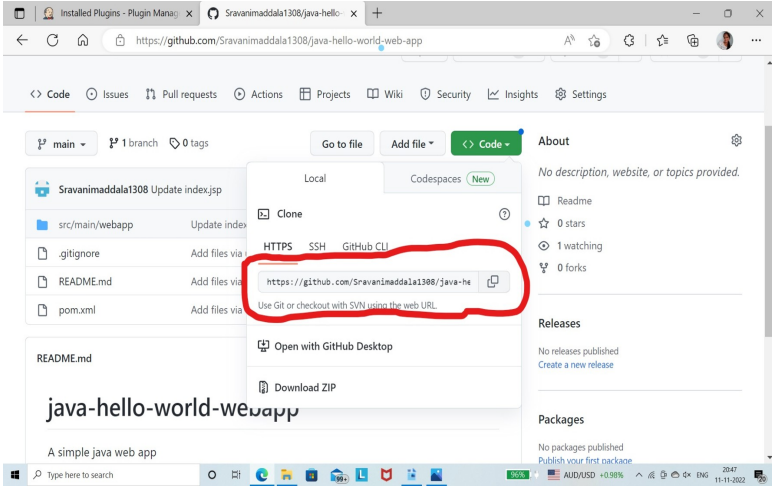
**In GitHub:**

1.Login to your GitHub account and create a New repository named as **java-hello-world-webapp**.

prior to this download the file java-hello-world-webapp from your mail and extract it.

2. Now upload the above extracted files to your new repository and commit changes.

3.copy the link as shown below:

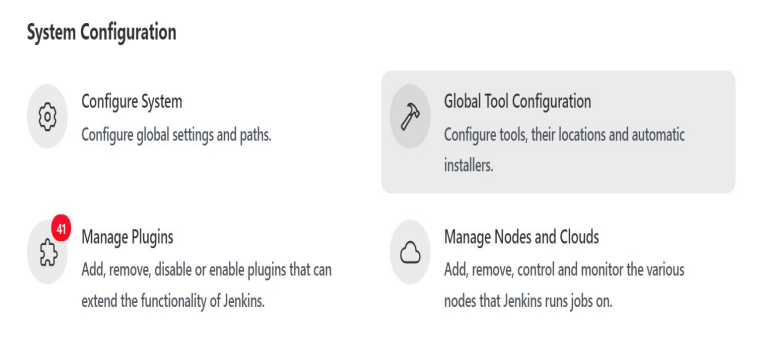


**Now in Jenkins**

**Follow the below given steps carefully:**

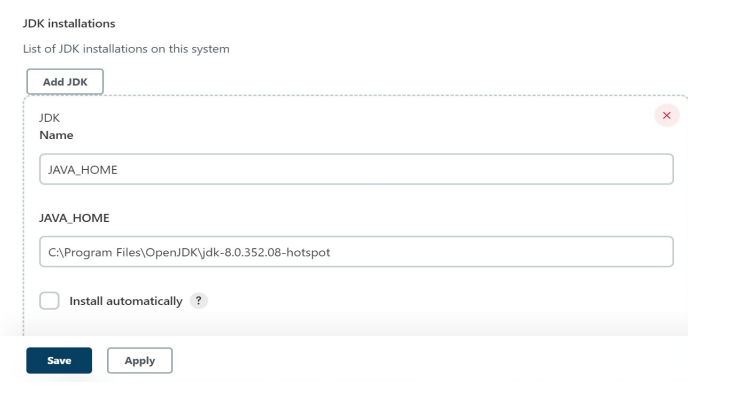
1.Open Jenkins Dashboard

2.In the left click on **Manage Jenkins** and click on **Global configure** **Tools**

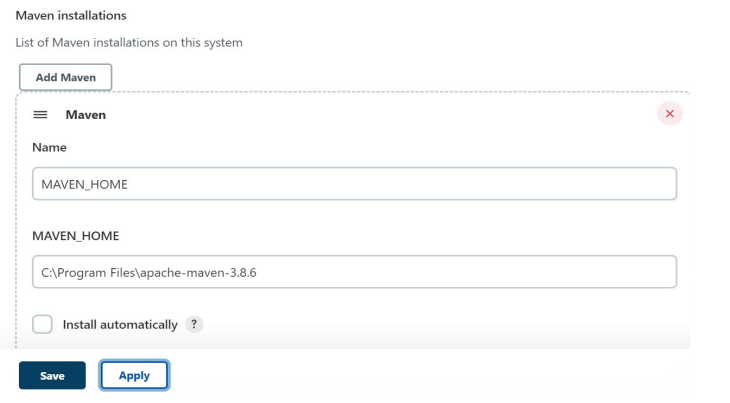


3.Now Scroll down and under **jdk installations** click on **add** and fill the details as shown in the below picture.

**Note** that the path specified here is the path where you have installed Java in your system.



4.Now go to the bottom of the page and under Maven click on add. The path specified here is the location of Maven where you have installed in your system.



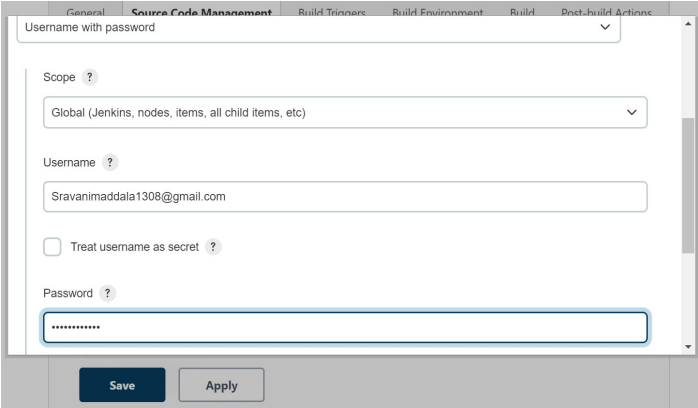
5.Now click on Apply and then click on save.

6.Now go to **plugin Manager(click on Manage Jenkins->Manage plugins).**

7.click on Available and search for **“deploy to container”** and click on **install without restart**. Also search for **Maven Integration** and **install**.

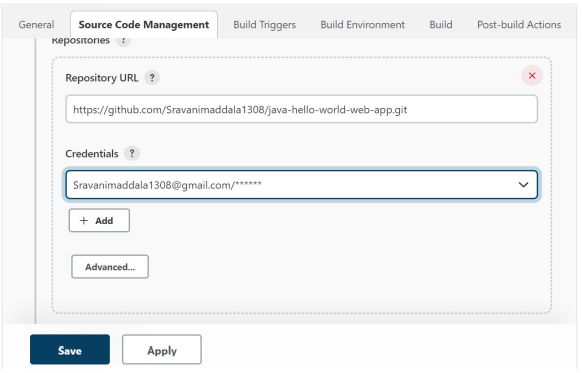
8. create a new job named **“Maven-webapp\_development”** and select **Freestyle project** while you are creating the job.

9.under SCM select **Git** and in the repository Url give the above copied link. under credentials click on add-->Jenkins and give the details as follows

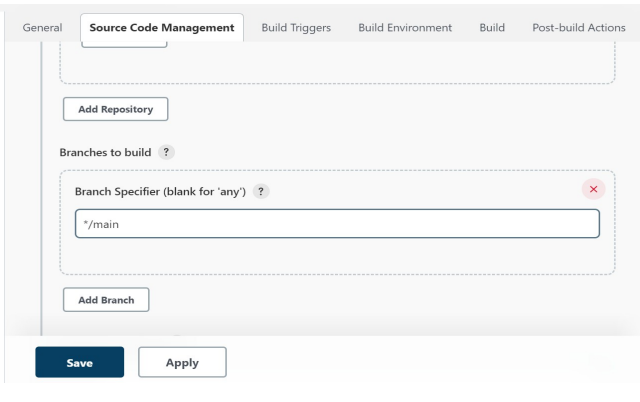


Give the username and password that you have used for logging to GitHub.Scroll down and click on “add”.

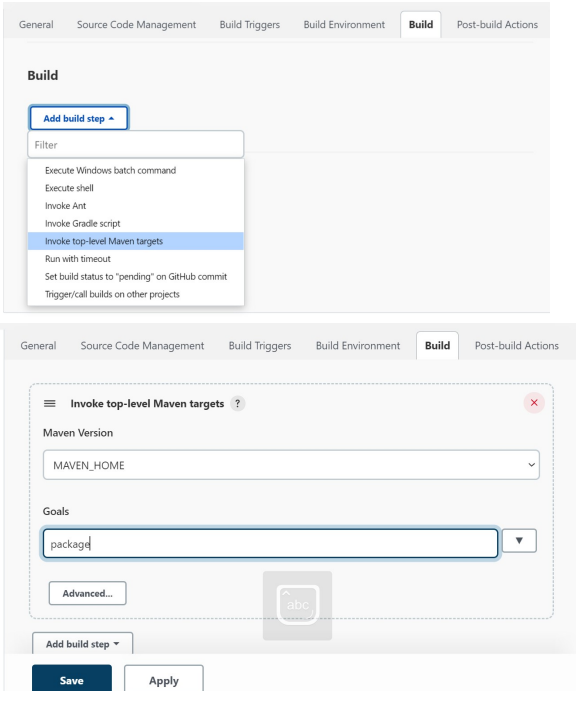
10.After the credentials successfully, select the one that is created above then the screen looks as follows



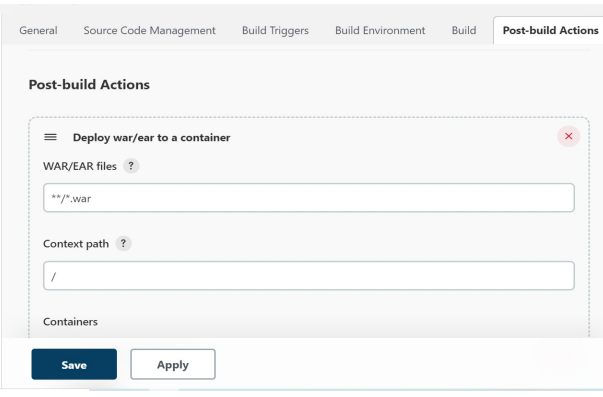
11. Specify the branch that the code in your GitHub has(main or master)

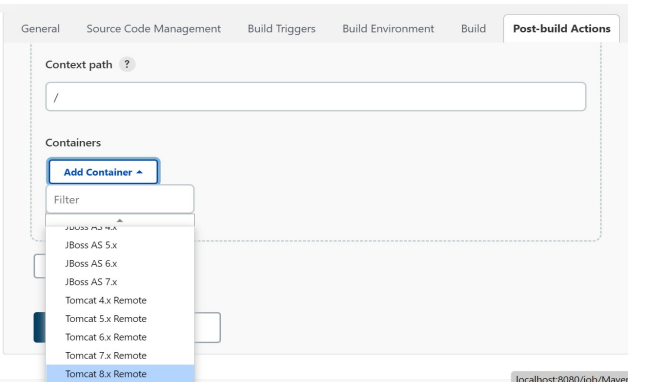


12.Under Build, click on **Add Build step** and **select invoke toplevel Maven targets** and specify the details as shown

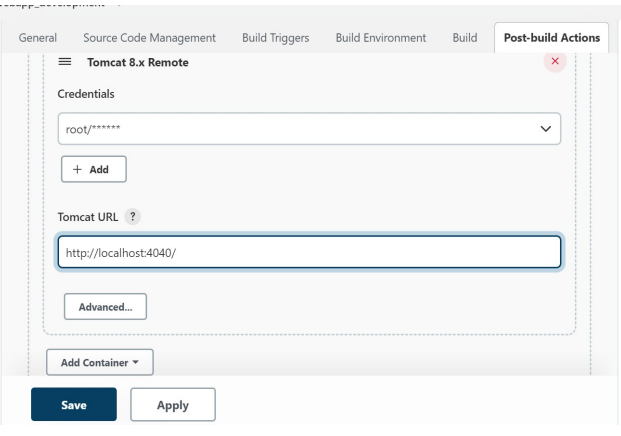


12.under **Post-build Actions** click on **Add container** and select **TomCat 8x remote** and give the details as follows





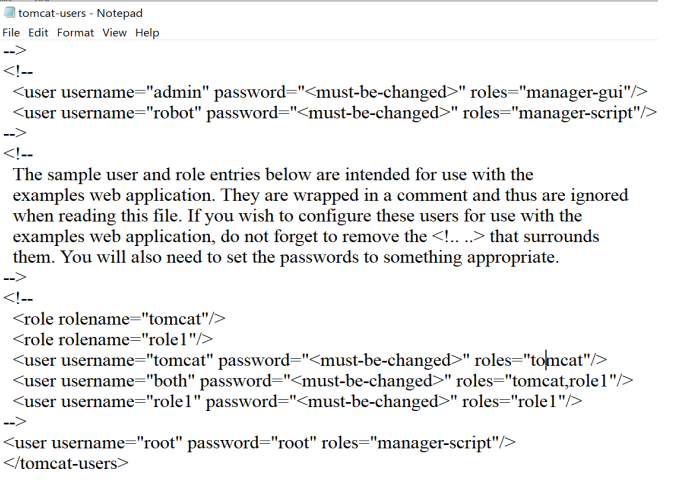
13.Under credentials click on Add-->Jenkins and give the username and password as root.Click on Add.Now select the above added credential from the dropdown displayed.



Click on Apply and save.

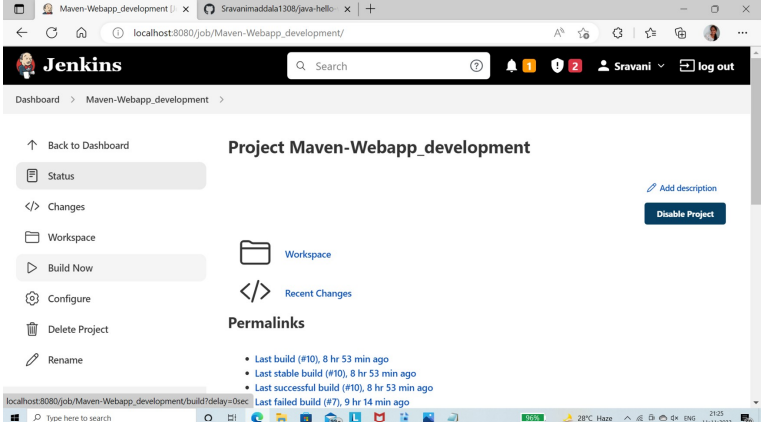
Now go to C:\Program Files\Apache Software Foundation\Tomcat 8.5\ conf in them open the file **tomcat-users** with notepad and insert the following line just above the last line

<user username=”root” password=”root” roles=”manager-scripts”



save the file.

Now click on “Build Now”



After the build has completed successfully open any web browser and search **localhost:4040** you will see the output of the HTML file as follows.

