Assignment Number:5

Name: Divya Shah Branch: IT/ V Roll No.:115 Date:24/08/2023

Aim: To build Java program using Jenkins.

LO mapped: LO1, L03

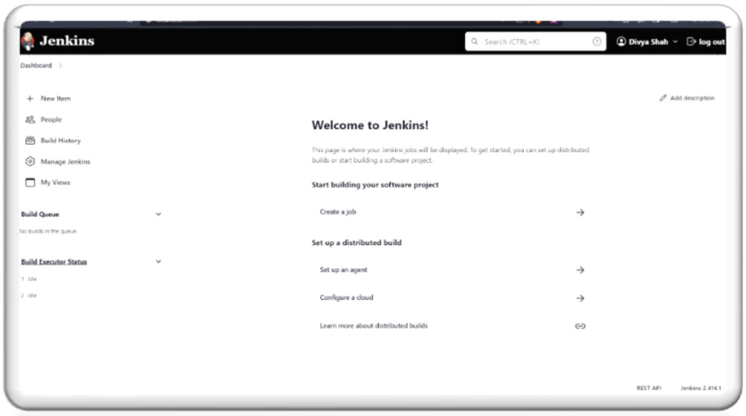
Theory:

Jenkins is an open-source automation server that facilitates continuous integration (CI) and continuous delivery (CD) of software projects. It helps automate the build, test, and deployment processes, allowing development teams to streamline their development workflow and deliver high-quality software more efficiently. Jenkins provides an extensible plugin architecture that enables integration with various tools and technologies.

Steps to Build a Java Program Using Jenkins:

To build a Java program using Jenkins, you'll need to set up a Jenkins job that defines the build process. Below are the steps to accomplish this:

I. Install Jenkins: Download and install Jenkins on your system.



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

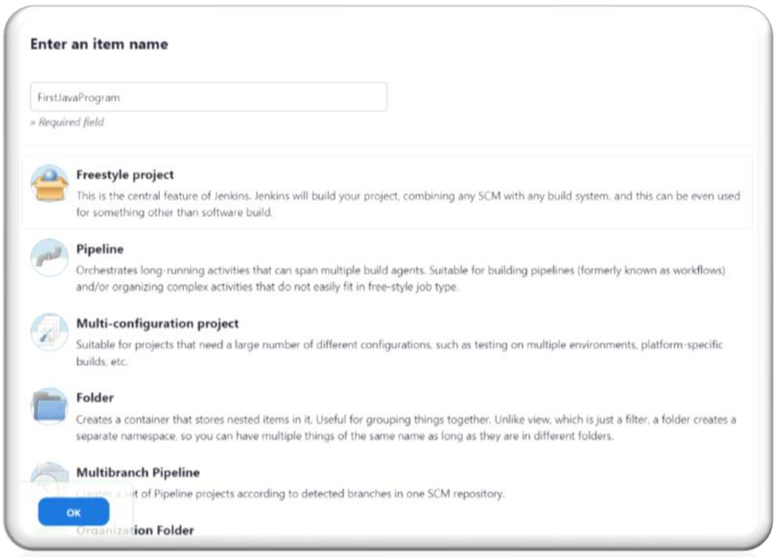
Start building your software project

Set up a distributed build

No builds in the queue.

Fig.1 Dashboard of Jenkins

II. Create Job: Create a new freestyle project job in Jenkins.



Enter an item name

FirstJavaProgram

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

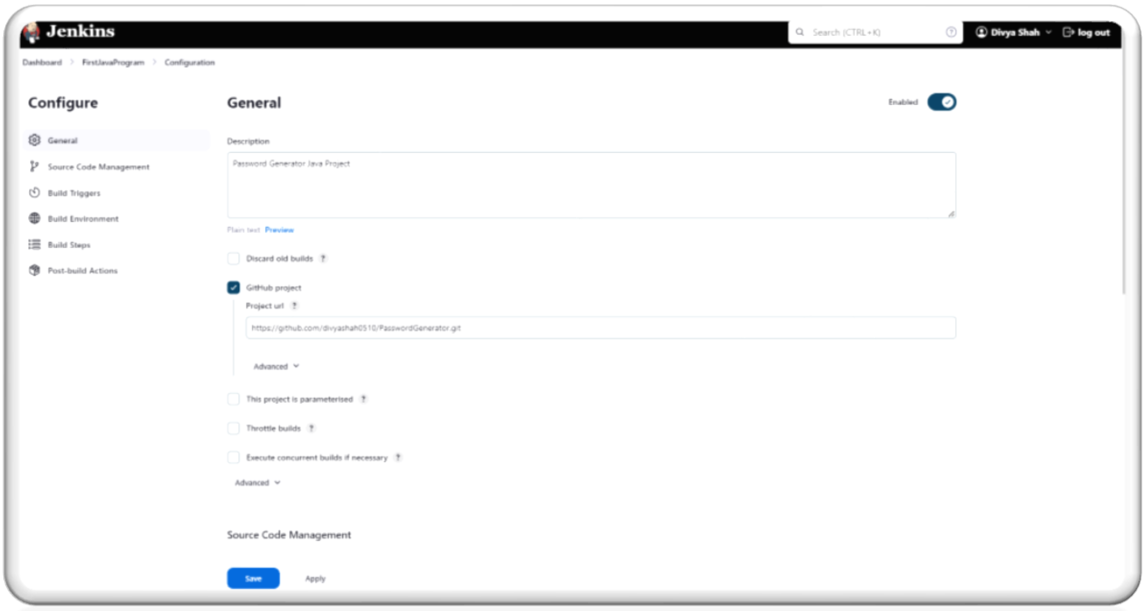
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder

Fig.2

III. Source Code: Configure source code management (e.g., Git repository URL).



General

Enabled

Description

Password Generator Java Project

Plain text Preview

☐ Discard old builds

☑ GitHub project

Project url

https://github.com/divyashah0510/PasswordGenerator.git

Advanced

☐ This project is parameterised

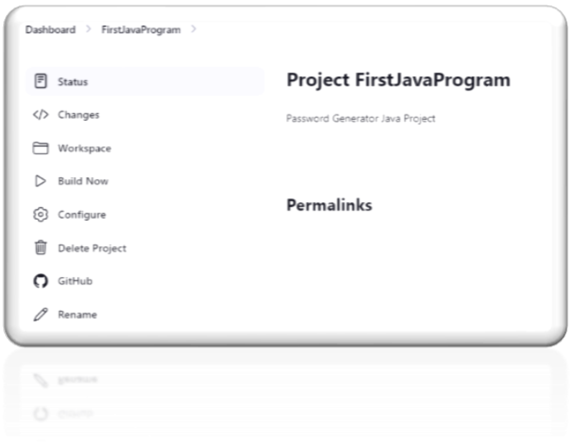
☐ Throttle builds

☐ Execute concurrent builds if necessary

Advanced

Source Code Management

Fig.3 GitHub Project copy



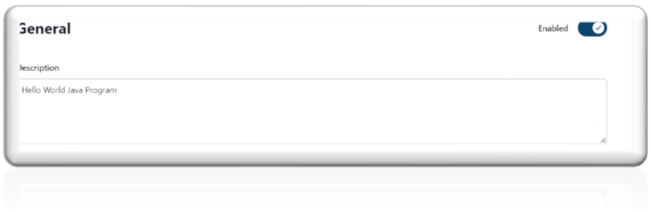
Dashboard > FirstJavaProgram >

Project FirstJavaProgram

Password Generator Java Project

Permalinks

Fig.3.1 GitHub Project copy



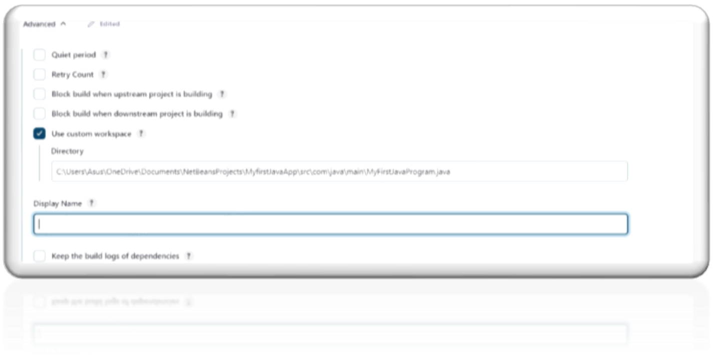
General

Enabled

Description

Hello World Java Program

Fig.3.2



Advanced ^ Edited

- [ ] Quiet period
- [ ] Retry Count
- [ ] Block build when upstream project is building
- [ ] Block build when downstream project is building
- [x] Use custom workspace

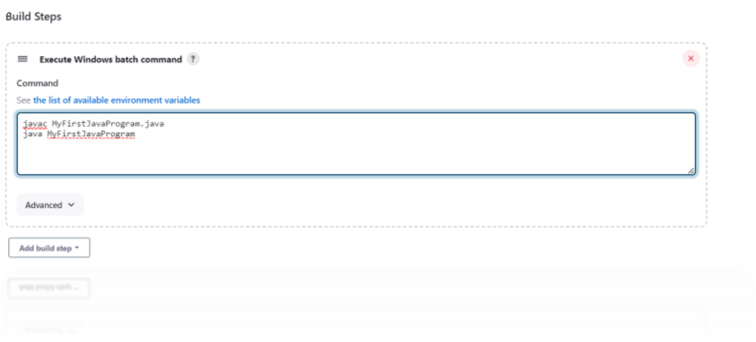
Directory

C:\Users\Asus\OneDrive\Documents\NetBeansProjects\MyfirstJavaApp\src\com\java\main\MyFirstJavaProgram.java

Display Name

- [ ] Keep the build logs of dependencies

Fig.3.3



Build Steps

Execute Windows batch command

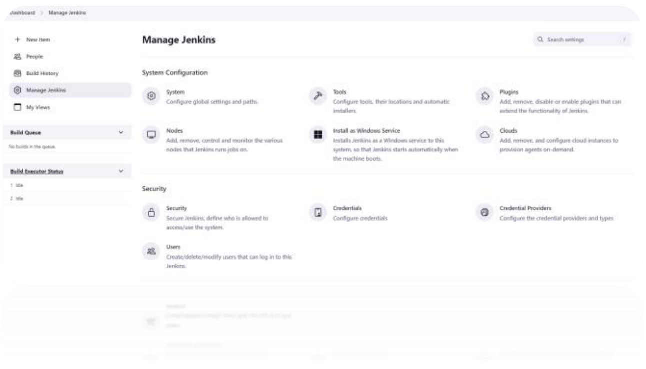
Command

See the list of available environment variables

```
javac MyFirstJavaProgram.java
java MyFirstJavaProgram
```

Fig.3.4 In this step choose “Execute windows bash commands”

IV. Build Trigger: Set up build triggers, like periodic or repository changes.


Dashboard > Manage Jenkins

New Item
People
Build History
Manage Jenkins
My Views

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

Manage Jenkins

Search settings

System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Install as Windows Service
Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically when the machine boots.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Security

Security
Secure Jenkins; define who is allowed to access/use the system.

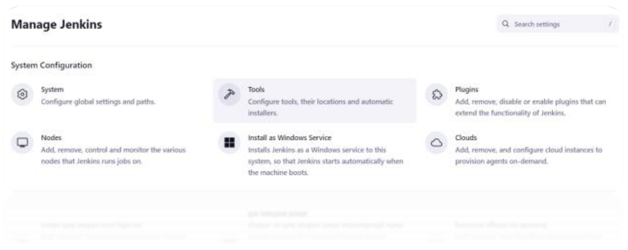
Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Users
Create/delete/modify users that can log in to this Jenkins.

Fig.4

V. Build Steps: Add build step, e.g., "Invoke top-level Maven targets" for Maven projects with goals like clean install.


Manage Jenkins

Search settings

System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

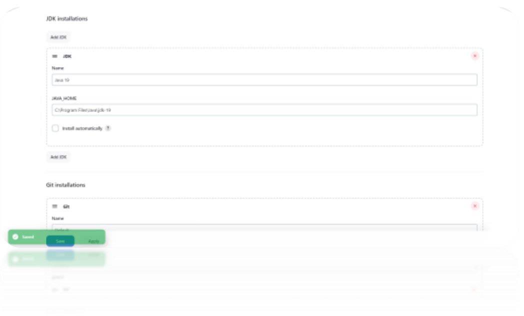
Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Install as Windows Service
Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically when the machine boots.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Fig.5


JDK installations

Add JDK

JDK

Name
Java 19

JAVA_HOME
C:\Program Files\Java\jdk-19

Install automatically

Add JDK

Git installations

Git

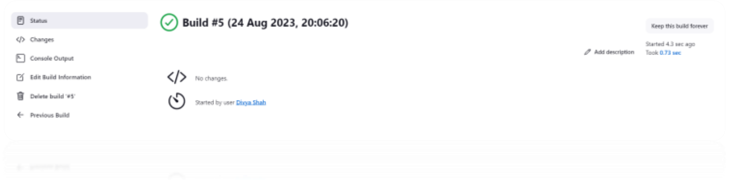
Name

Saved

Save Apply

Fig.5

VI. Post-Build Actions: Optionally configure actions after the build, like archiving artifacts or notifications.


Status
Changes
Console Output
Edit Build Information
Delete build '#5'
Previous Build

Build #5 (24 Aug 2023, 20:06:20)

Keep this build forever

Add description

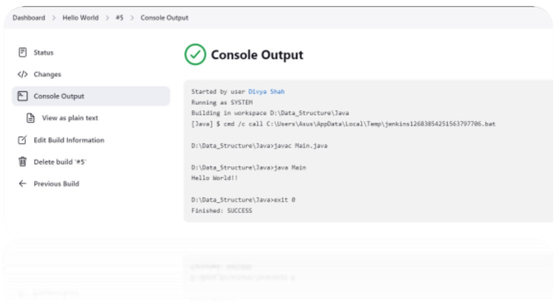
Started 4.3 sec ago
Took 0.73 sec

No changes.

Started by user Divya Shah

Fig.6

VII. Save and Run: Save job settings and manually trigger the job.

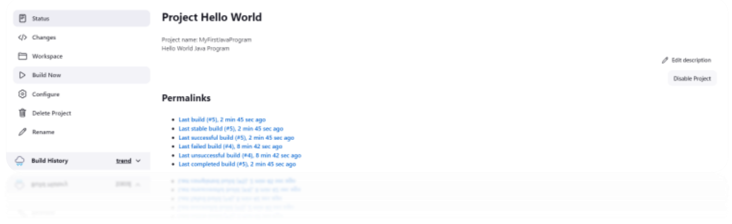


Console Output

```
Started by user Divya Shah
Running as SYSTEM
Building in workspace D:\Data_Structure\Java
[Java] $ cmd /c call C:\Users\Asus\AppData\Local\Temp\jenkins12683854251563797706.bat

D:\Data_Structure\Java>javac Main.java

D:\Data_Structure\Java>java Main
Hello World!!

D:\Data_Structure\Java>exit 0
Finished: SUCCESS
```

Fig.7

VIII. Monitor Progress: Watch the Jenkins dashboard for build progress.



Project Hello World

Project name: MyFirstJavaProgram
Hello World Java Program

Permalinks

- Last build (#5), 2 min 45 sec ago
- Last stable build (#5), 2 min 45 sec ago
- Last successful build (#5), 2 min 45 sec ago
- Last failed build (#4), 8 min 42 sec ago
- Last unsuccessful build (#4), 8 min 42 sec ago
- Last completed build (#5), 2 min 45 sec ago

Fig.8

IX. View Artifacts: Access build artifacts if configured.

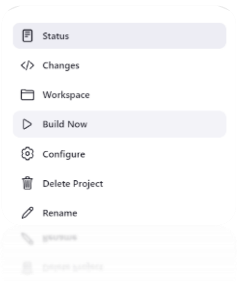


Fig.9

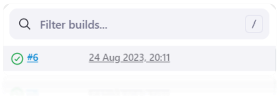
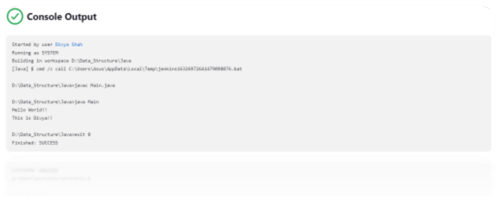


Fig.10


Console Output

```
Started by user Divya Shah
Running as SYSTEM
Building in workspace D:\Data_Structure\Java
[Java] $ cmd /c call C:\Users\Asus\AppData\Local\Temp\jenkins16326872641d79098076.bat

D:\Data_Structure\Java>javac Main.java

D:\Data_Structure\Java>java Main
Hello World!!
This is Divya!!

D:\Data_Structure\Java>exit 0
Finished: SUCCESS
```

Fig.11

Conclusion: By this assignment we learned how to build a java program using Jenkins.