# EXPERIMENT NO. 3

**Date Of Perfomance :**

**Date Of Submission :**

**AIM** : a) To practice/execute shell programs using Jenkins.
b) To practice/execute parameterised Java programs using Jenkins.

## THEORY

Jenkins is an open-source server that is written entirely in Java. It lets you execute a series of actions to achieve the continuous integration process, that too in an automated fashion. This CI server runs in servlet containers such as Apache Tomcat.

Jenkins facilitates continuous integration and continuous delivery in software projects by automating parts related to build, test, and deployment. This makes it easy for developers to continuously work on the betterment of the product by integrating changes to the project. Jenkins automates the software builds in a continuous manner and lets the developers know about the errors at an early stage.

A strong Jenkins community is one of the prime reasons for its popularity. Jenkins is not only extensible but also has a thriving plugin ecosystem.

**Software builds using Jenkins** are possible using the build systems such as:

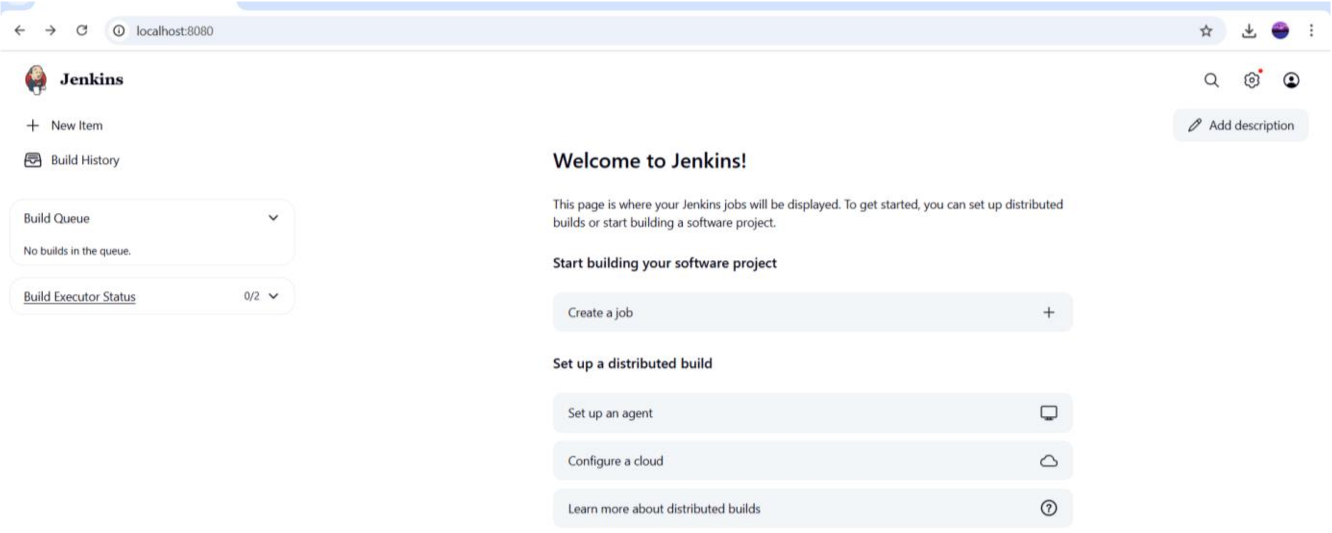
- Gradle
- Maven
- And more

**Automation testing** using test frameworks such as:

- Nose2
- PyTest
- Robot
- Selenium
- And more

## TO PRACTISE/EXECUTE SHELL PROGRAMS USING JENKINS

**Step 1:** Click on *Create new jobs*

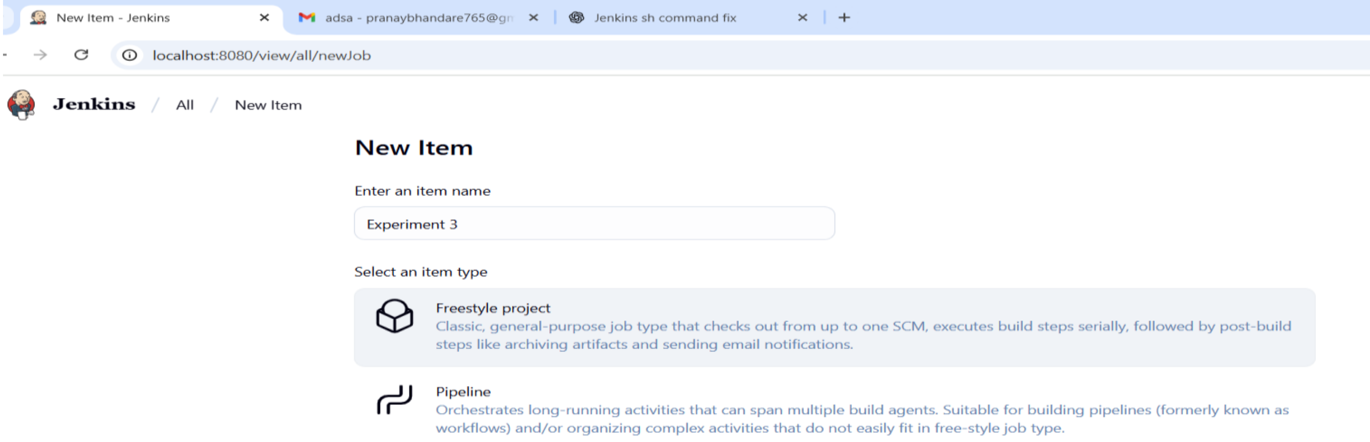


Welcome to Jenkins!

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.

No builds in the queue.

**Step 2:** Give a name to project as “P1”, select Option “Free style project” and click on OK button.



New Item

Enter an item name

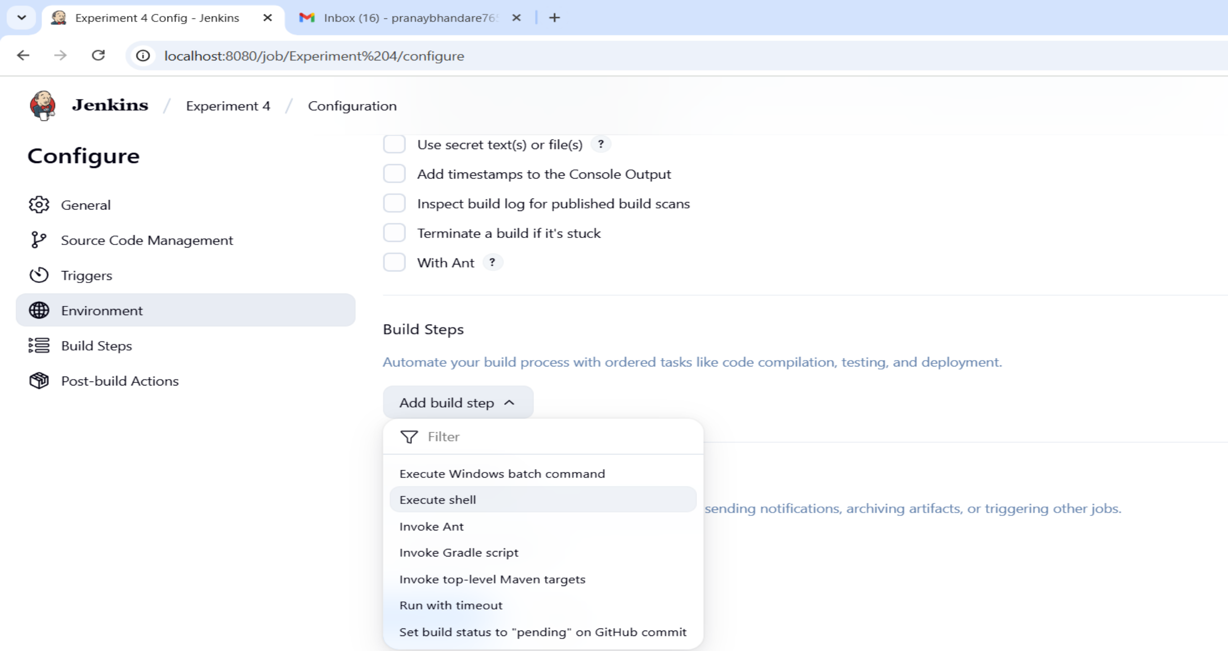
Experiment 3

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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.

**Step 3:** To run simple shell scripts on Jenkins click on Build option select the Execute script from dropdown menu.


Experiment 4 Config - Jenkins
Inbox (16) - pranaybhandare76
localhost:8080/job/Experiment%204/configure
Jenkins / Experiment 4 / Configuration
Configure
General
Source Code Management
Triggers
Environment
Build Steps
Post-build Actions
Add build step
Filter
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit

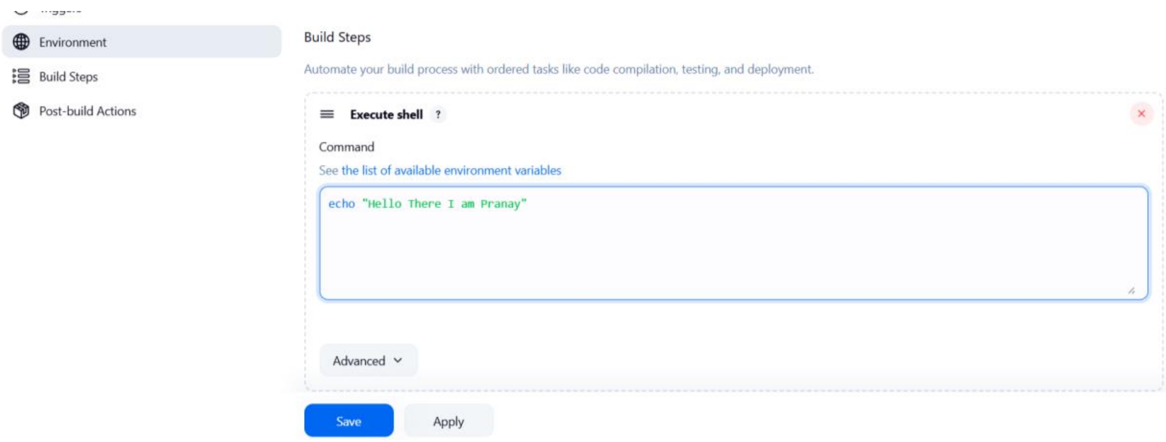
□ Use secret text(s) or file(s) ?
□ Add timestamps to the Console Output
□ Inspect build log for published build scans
□ Terminate a build if it's stuck
□ With Ant ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

sending notifications, archiving artifacts, or triggering other jobs.

**Step 4:** Write a simple shell command and click on apply followed by save button.



Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

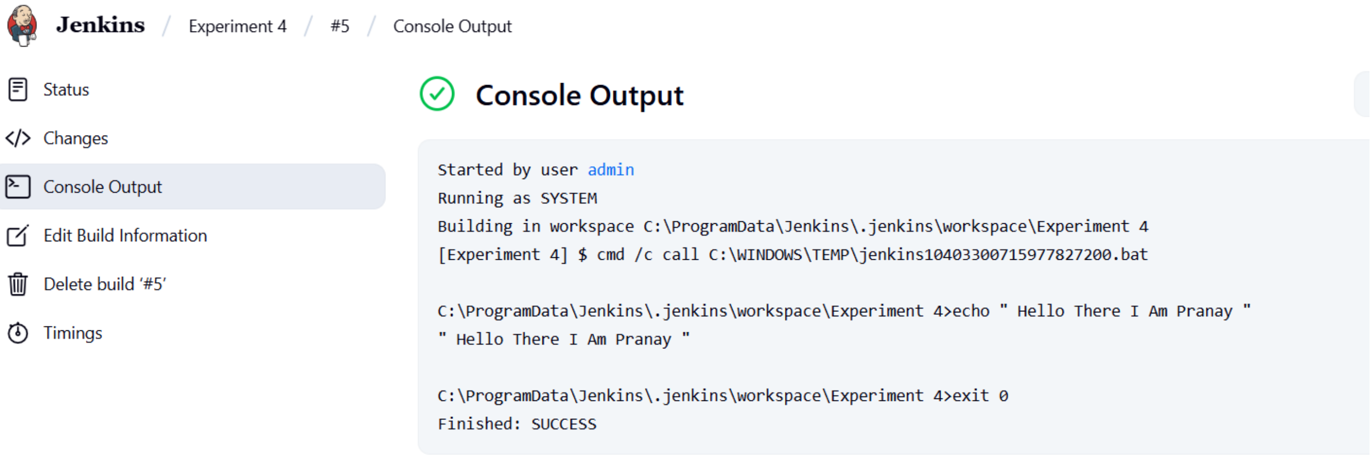
Execute shell ?

Command

See the list of available environment variables

```
echo "Hello There I am Pranay"
```

**Step 5:** Click on first build "1" followed by console output to see the output.



Console Output

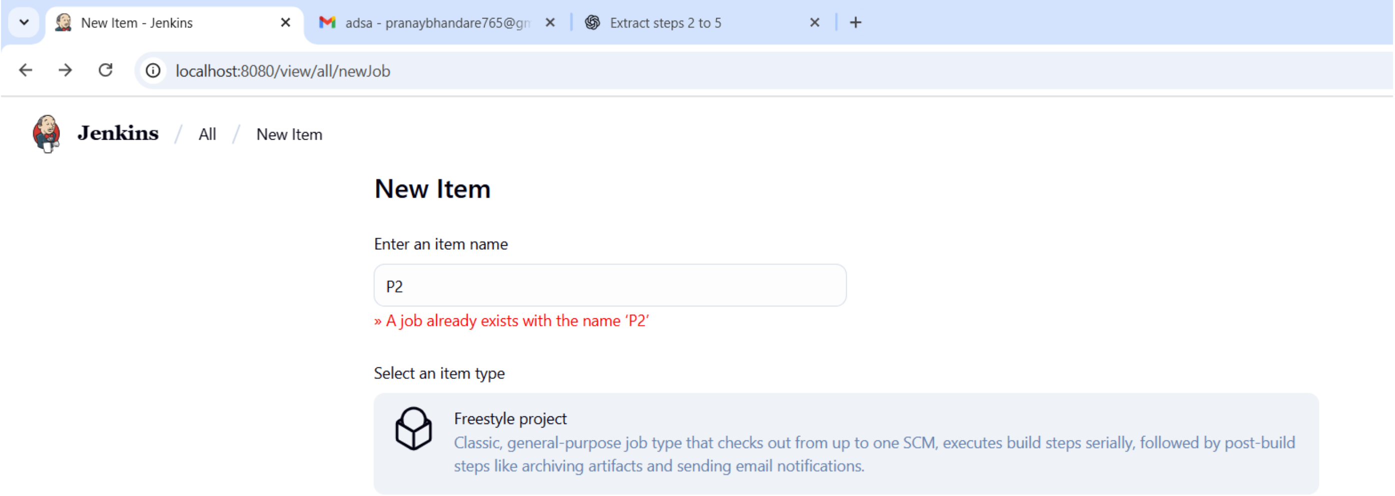
```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Experiment 4
[Experiment 4] $ cmd /c call C:\WINDOWS\TEMP\jenkins10403300715977827200.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Experiment 4>echo " Hello There I Am Pranay "
" Hello There I Am Pranay "

C:\ProgramData\Jenkins\.jenkins\workspace\Experiment 4>exit 0
Finished: SUCCESS
```

## Practise / Execute Parameterised JAVA Programs Using JENKINS.

**Step 1:** Create a freestyle project P2 in Jenkins.



New Item

Enter an item name

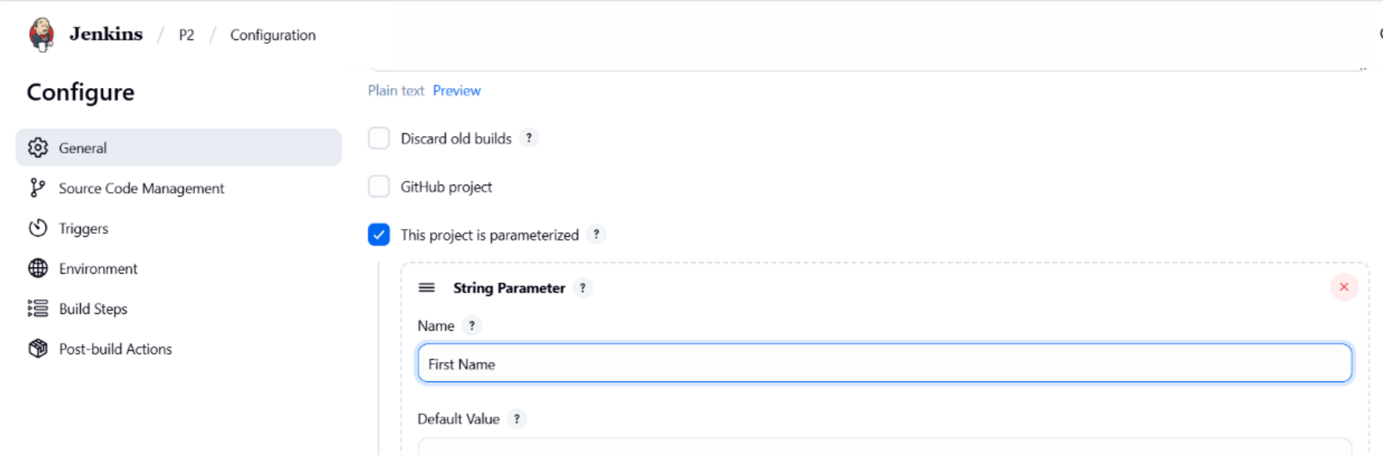
P2

» A job already exists with the name 'P2'

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

**Step 2:** Click on general menu and select option this project is parameterize. Select String parameter and specify name as "First-Name"



□ Discard old builds ?

□ GitHub project

☑ This project is parameterized ?

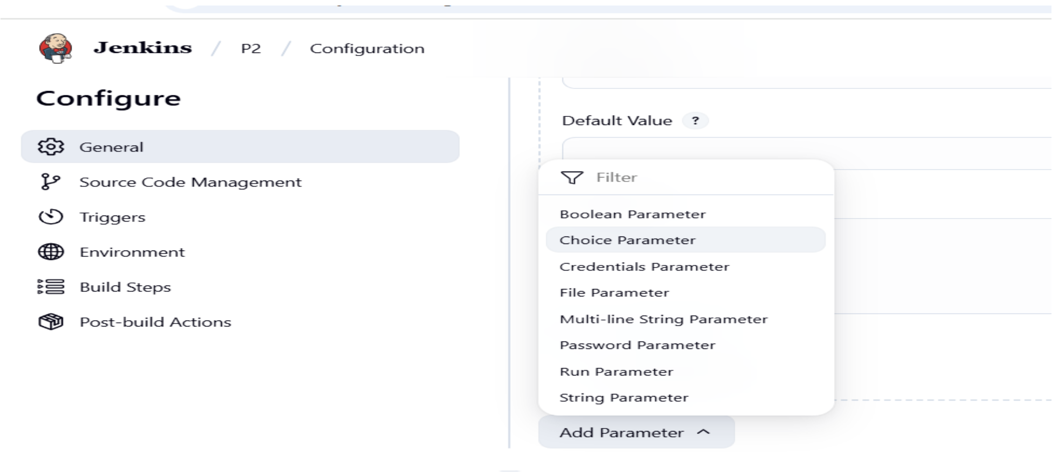
String Parameter ?

Name ?

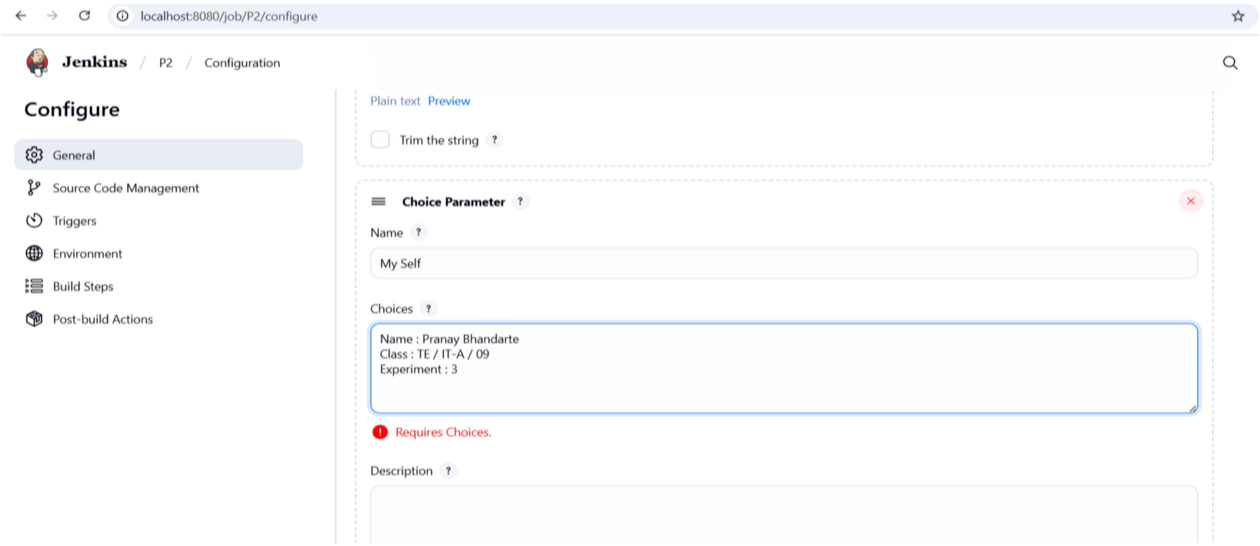
First Name

Default Value ?

**Step 3:** Click on add parameter and select choice parameter. Take second parameter as choice parameter



**Step 4:** Specify name as “City” and add the choices in each line

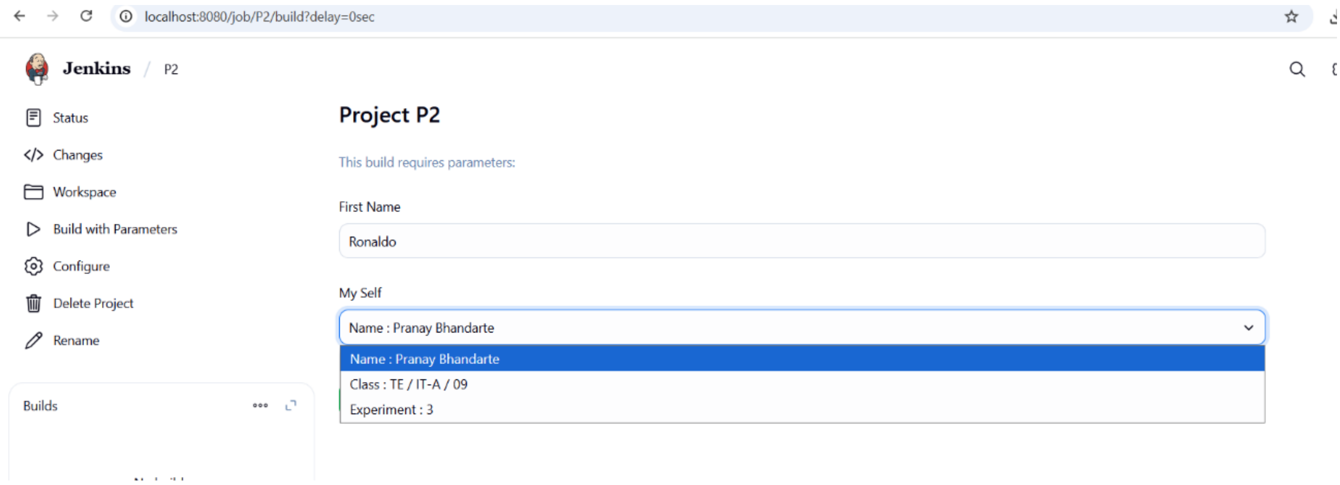


Name: My Self

Choices:

```
Name : Pranay Bhandarte
Class : TE / IT-A / 09
Experiment : 3
```

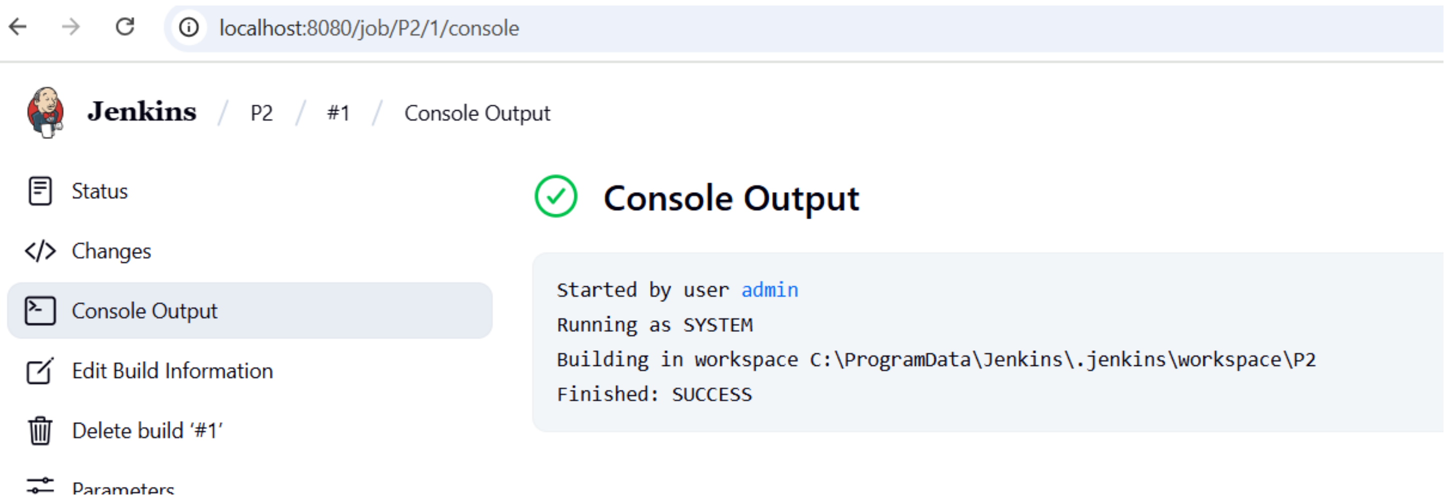
**Step 5:** Click on build with parameters and specify the values



First Name: Ronaldo

My Self: Name : Pranay Bhandarte

- Name : Pranay Bhandarte
- Class : TE / IT-A / 09
- Experiment : 3



```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\P2
Finished: SUCCESS
```

**Conclusion:** Hence we can conclude that we have learned and implemented shell programs and parameterized Java programs using Jenkins.