

## EXPERIMENT NO. 3

**Date Of Performance :**

**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:

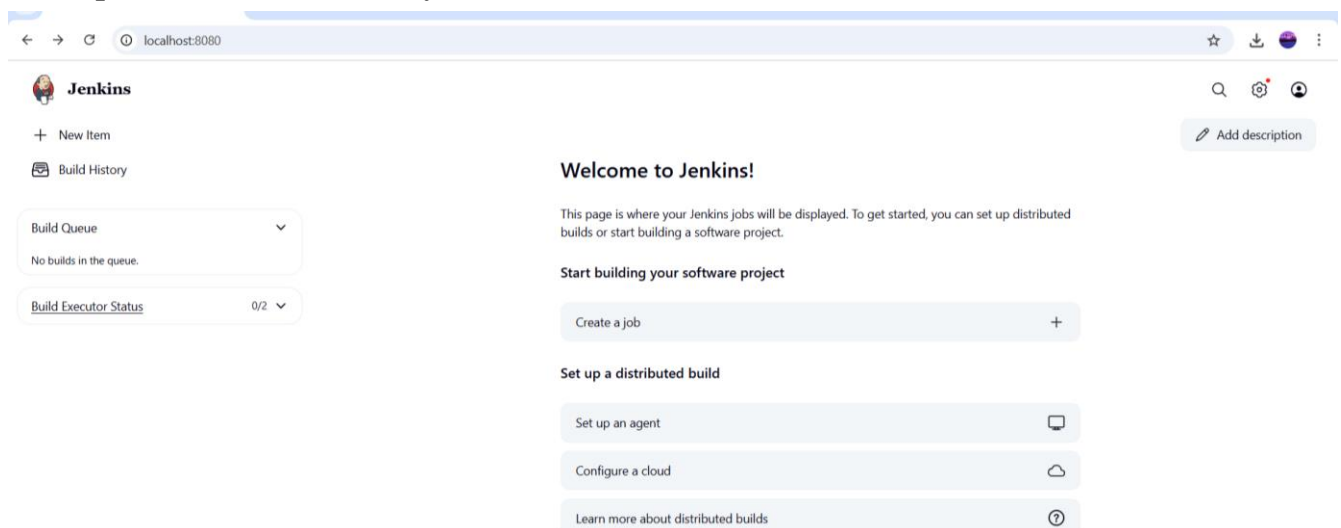
- Gradle
- Maven
- And more

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

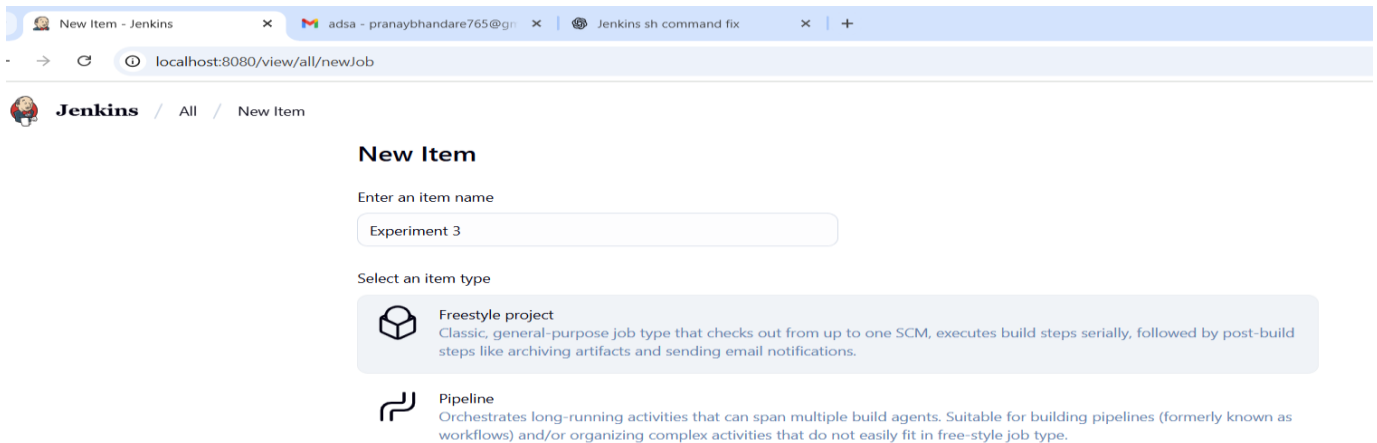
- Nose2
- PyTest
- Robot
- Selenium
- And more

### TO PRACTISE/EXECUTE SHELL PROGRAMS USING JENKINS

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

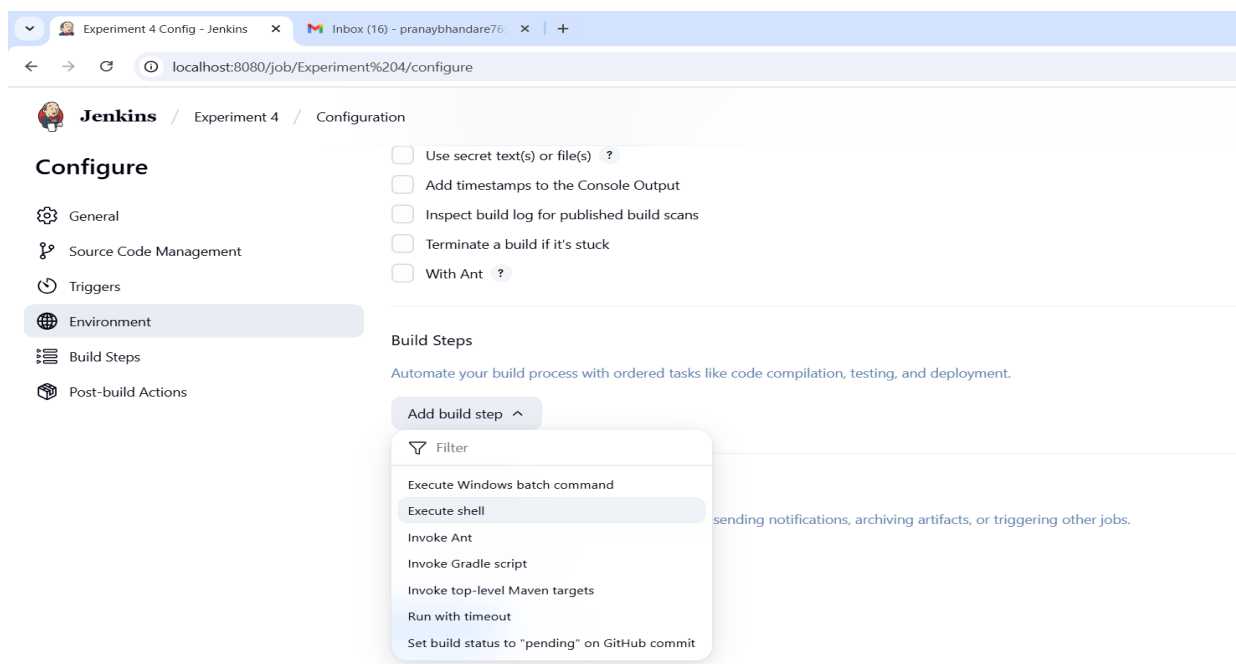


**Step 2:** Give a name to project as “P1”, select Option “Free style project” and click on OK button.



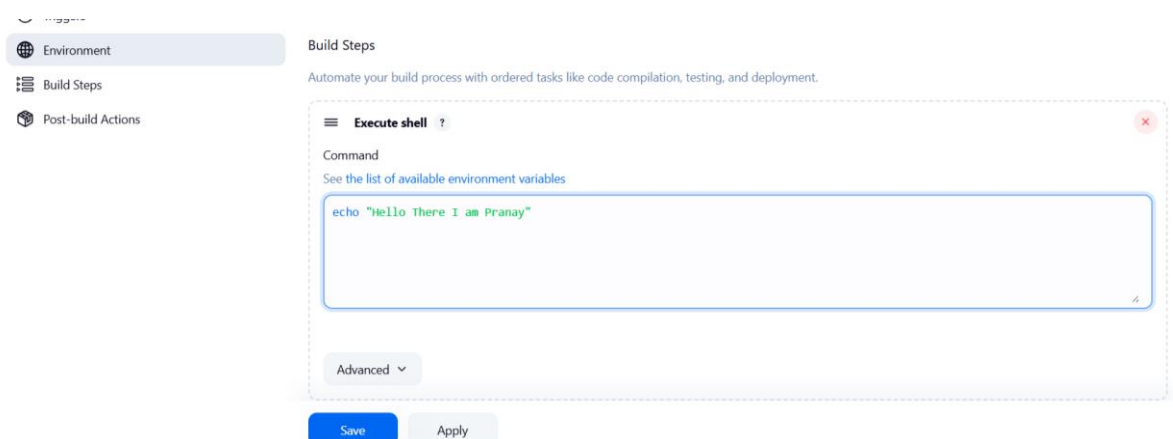
The screenshot shows the Jenkins 'New Item' page in a web browser. The browser tabs include 'New Item - Jenkins', 'adsa - pranaybhandare765@gm', and 'Jenkins sh command fix'. The address bar shows 'localhost:8080/view/all/newJob'. The Jenkins logo and navigation links 'All' and 'New Item' are at the top. The main heading is 'New Item'. Below it, there is a text input field for 'Enter an item name' with the value 'Experiment 3'. Under 'Select an item type', two options are shown: 'Freestyle project' (selected) and 'Pipeline'. The 'Freestyle project' description states: '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.' The 'Pipeline' description states: '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.



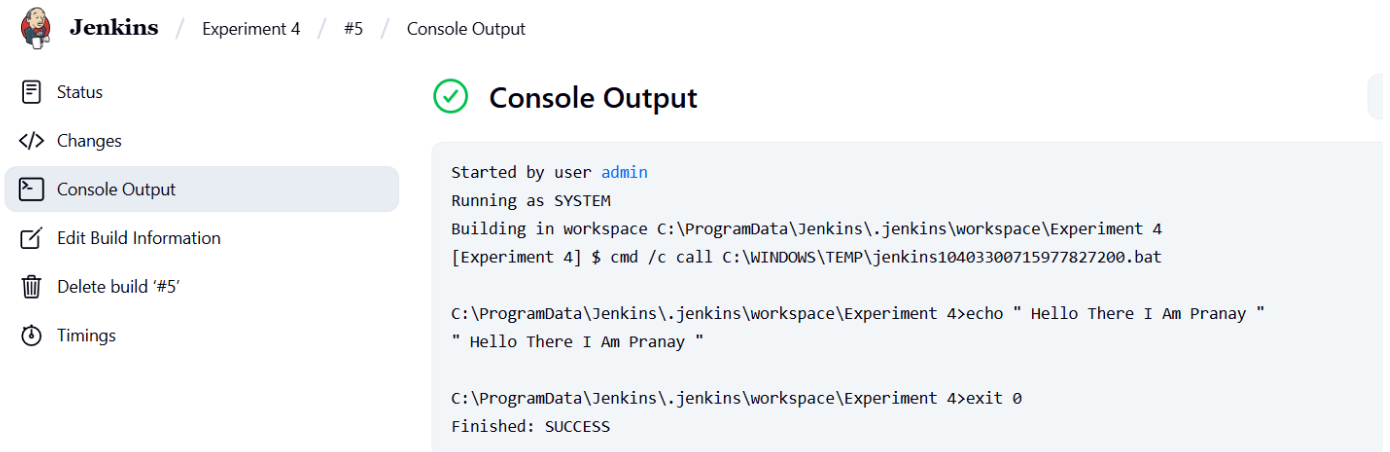
The screenshot shows the Jenkins 'Configure' page for 'Experiment 4'. The browser tabs include 'Experiment 4 Config - Jenkins' and 'Inbox (16) - pranaybhandare76'. The address bar shows 'localhost:8080/job/Experiment%204/configure'. The Jenkins logo and navigation links 'Experiment 4' and 'Configuration' are at the top. The main heading is 'Configure'. On the left, there is a sidebar with links: 'General', 'Source Code Management', 'Triggers', 'Environment' (selected), 'Build Steps', and 'Post-build Actions'. The main content area has a section for 'Build Steps' with a description: 'Automate your build process with ordered tasks like code compilation, testing, and deployment.' Below this, there is a button 'Add build step ^'. A dropdown menu is open, showing a list of build steps: 'Filter', 'Execute Windows batch command', 'Execute shell' (highlighted), 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. There are also several checkboxes for configuration options: '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', and 'With Ant'.

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



The screenshot shows the 'Execute shell' configuration dialog in Jenkins. The dialog has a title bar 'Execute shell ?' and a close button. Inside, there is a section for 'Command' with a link 'See the list of available environment variables'. Below this is a text area containing the command 'echo "Hello There I am Pranay"'. At the bottom, there is a button 'Advanced' and two buttons 'Save' and 'Apply'.

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



The screenshot shows the Jenkins interface for a job named 'Experiment 4' at build #5. The 'Console Output' tab is selected, displaying the following text:

```
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

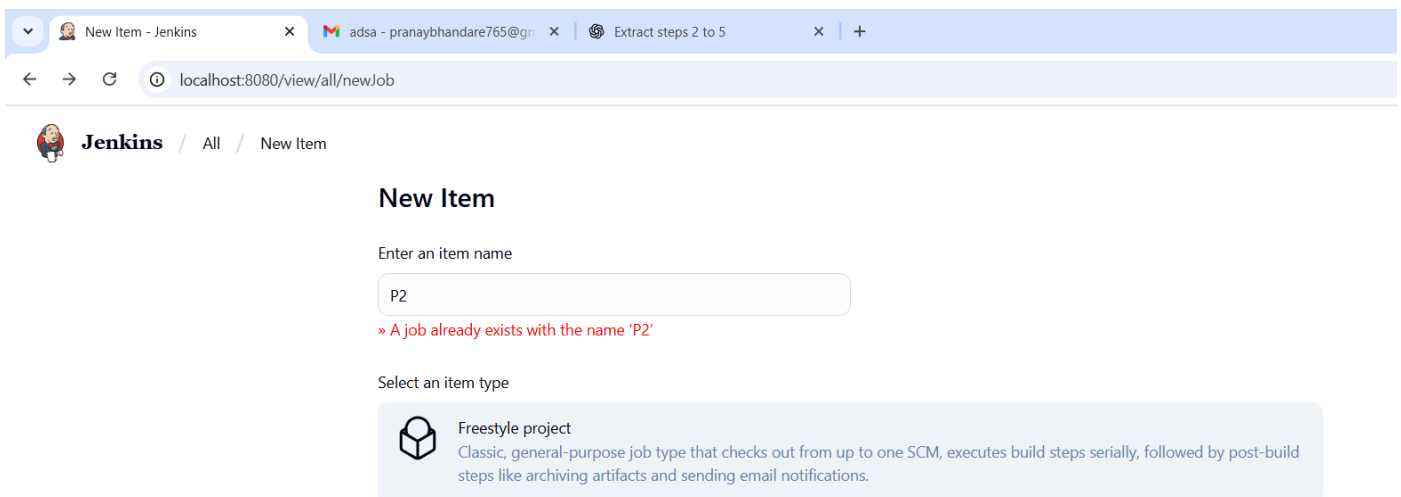
C:\ProgramData\Jenkins\.jenkins\workspace\Experiment 4>echo " Hello There I Am Pranay "
" Hello There I Am Pranay "

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

The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build '#5', and Timings.

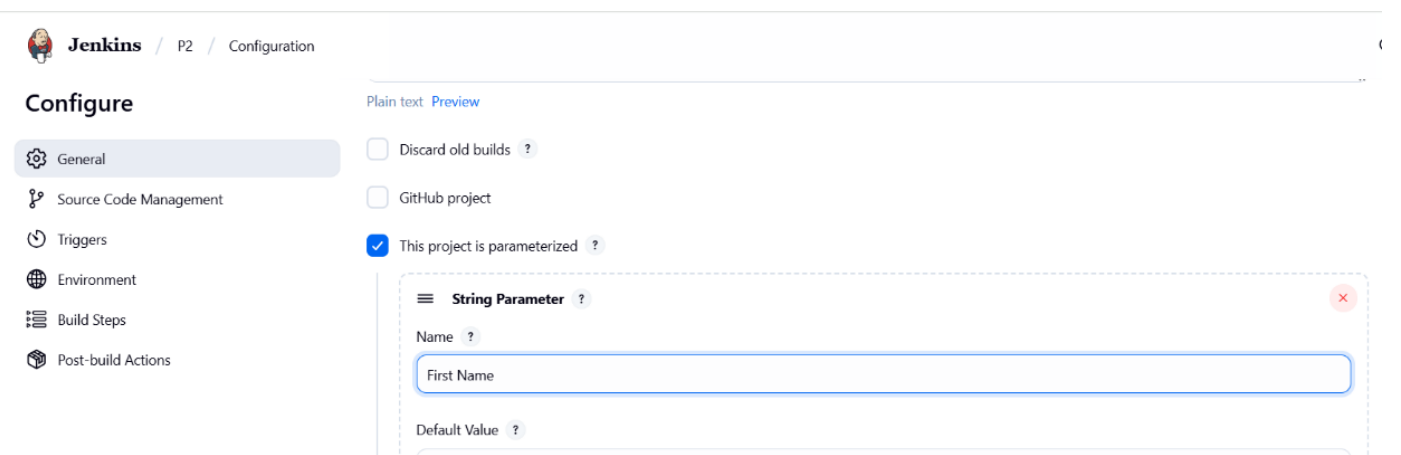
## Practise / Execute Parameterised JAVA Programs Using JENKINS.

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



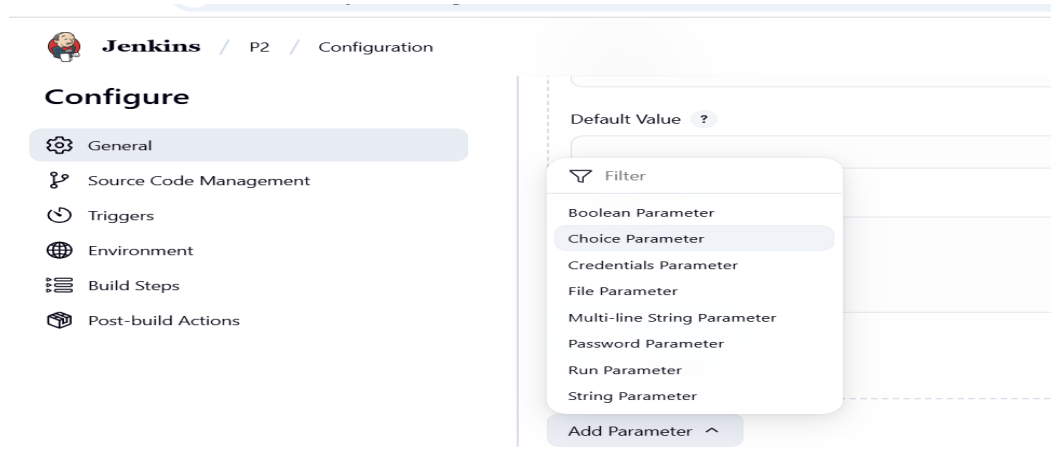
The screenshot shows the 'New Item' page in Jenkins. The browser address bar indicates the URL is localhost:8080/view/all/newJob. The page title is 'New Item - Jenkins'. The breadcrumb navigation shows 'Jenkins / All / New Item'. The main heading is 'New Item'. Below it, there is a text input field for 'Enter an item name' with the value 'P2'. A red error message below the field states: '» A job already exists with the name 'P2''. Below this, there is a section 'Select an item type' with a list of options. The 'Freestyle project' option is selected, with a description: '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”

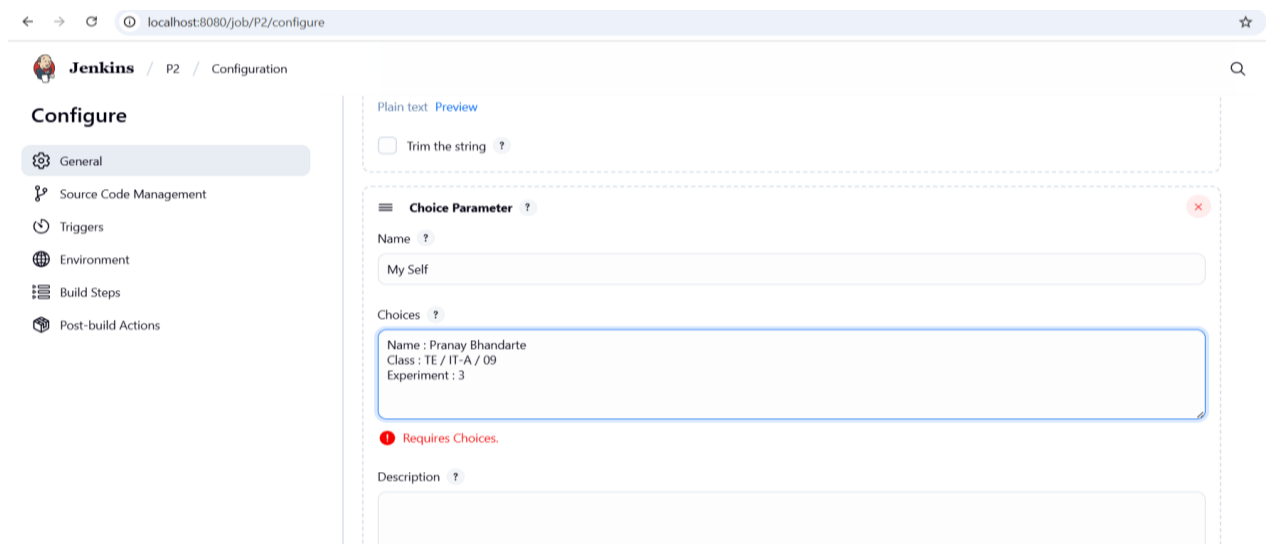


The screenshot shows the 'Configure' page for the Jenkins job 'P2'. The breadcrumb navigation shows 'Jenkins / P2 / Configuration'. The left sidebar contains links for General (selected), Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The main heading is 'Configure'. Below it, there is a section 'Plain text Preview'. The 'Discard old builds' checkbox is unchecked. The 'GitHub project' checkbox is unchecked. The 'This project is parameterized' checkbox is checked. Below this, there is a section 'String Parameter' with a 'Name' field containing 'First Name' and a 'Default Value' field.

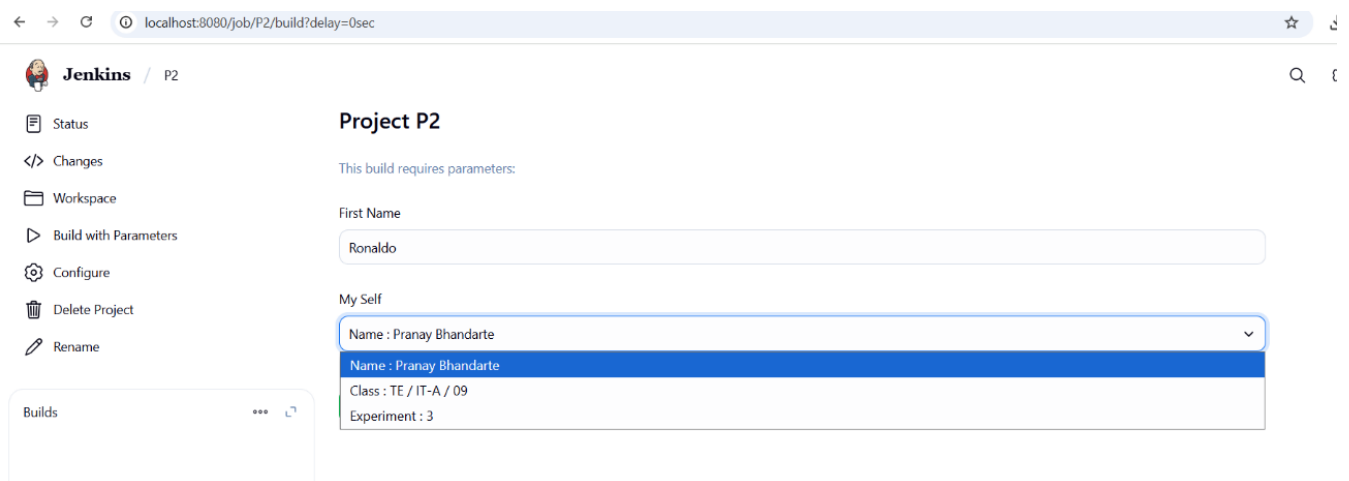
**Step 3:** Click on add parameter and select choice parameter. Take second parameter as choice parameter



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



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





← → ↻ ⓘ localhost:8080/job/P2/1/console




**Jenkins** / P2 / #1 / Console Output


 Status

 Changes

 Console Output

 Edit Build Information

 Delete build '#1'

 Parameters



## Console Output

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.