**Experiment 4**

**Aim:** To understand Continuous Integration; to install and configure Jenkins with Maven / Ant / Gradle to setup a build Job

**Installation:**

Running the installer:



A screenshot of a computer

Description automatically generated

Initial login:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Installing plugins:

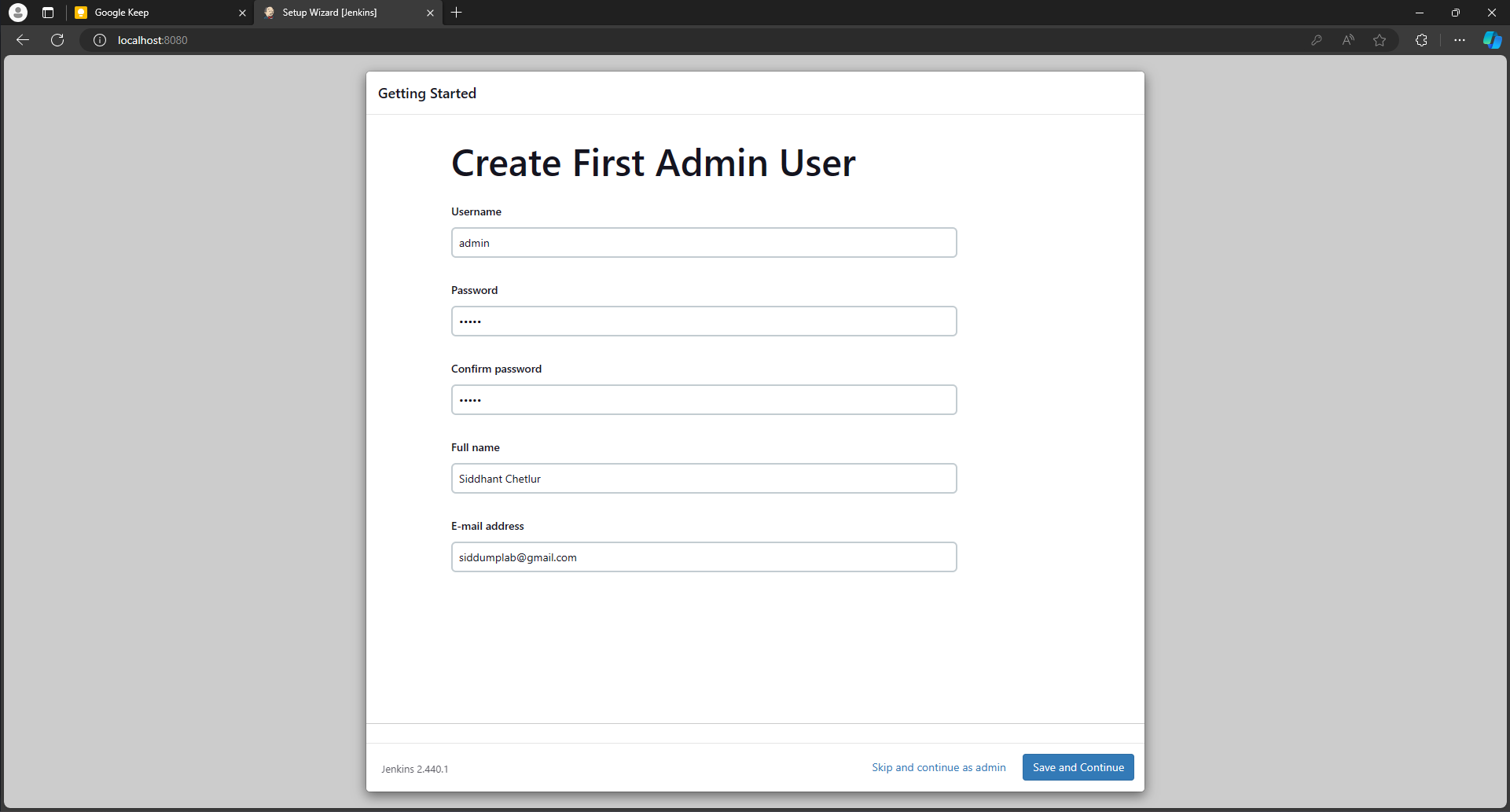
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Creating first admin user:



Finishing setup:

A screenshot of a computer

Description automatically generated

Launching dashboard:

A screenshot of a computer

Description automatically generated

**Programming in Jenkins:**

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.” In simple way, Continuous integration (CI) is the practice of frequently building and testing each change done to your code automatically.

Jenkins is a self-contained, open-source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Our first job will execute the shell commands. The freestyle project provides enough options and features to build the complex jobs that you will need in your projects.

Example 1

Example 1.1: Deploying a freestyle app in Jenkins

Creating a job:

A close-up of a white background

Description automatically generated

Naming the job and setting it as freestyle:

A screenshot of a computer

Description automatically generated

Selecting build type as “Execute shell”:

A screenshot of a computer program

Description automatically generated

Entering a simple command for the shell execution:

A blue rectangle with a white background

Description automatically generated

Applying and saving the project configuration:

A green rectangle with white background

Description automatically generated

Building the project:



Console output (after building):

A close-up of a white card

Description automatically generated

Example 1.2: Taking parameters through files

Contents of script example1.cmd:

A screenshot of a computer

Description automatically generated

Executing script example1.cmd on the terminal:

A screenshot of a computer

Description automatically generated

Modifying the Jenkins project to execute the script while supplying required parameters:

A screenshot of a computer

Description automatically generated

Console output after building the modified project:

A close-up of a letter

Description automatically generated

Example 2

Example 2.1: Running a Java program under Jenkins

Creating a simple Java program:

A screenshot of a computer

Description automatically generated

Compiling and running the program on the terminal:

A screenshot of a computer

Description automatically generated

Creating a new freestyle project:

A screenshot of a computer

Description automatically generated

Configure new project:

A screenshot of a computer

Description automatically generated

Console output after building:



Example 3

Example 3.1: Parameterise build

Creating a new freestyle project:

A screenshot of a computer

Description automatically generated

Enabling parameterisation and adding a String parameter:

A screenshot of a computer

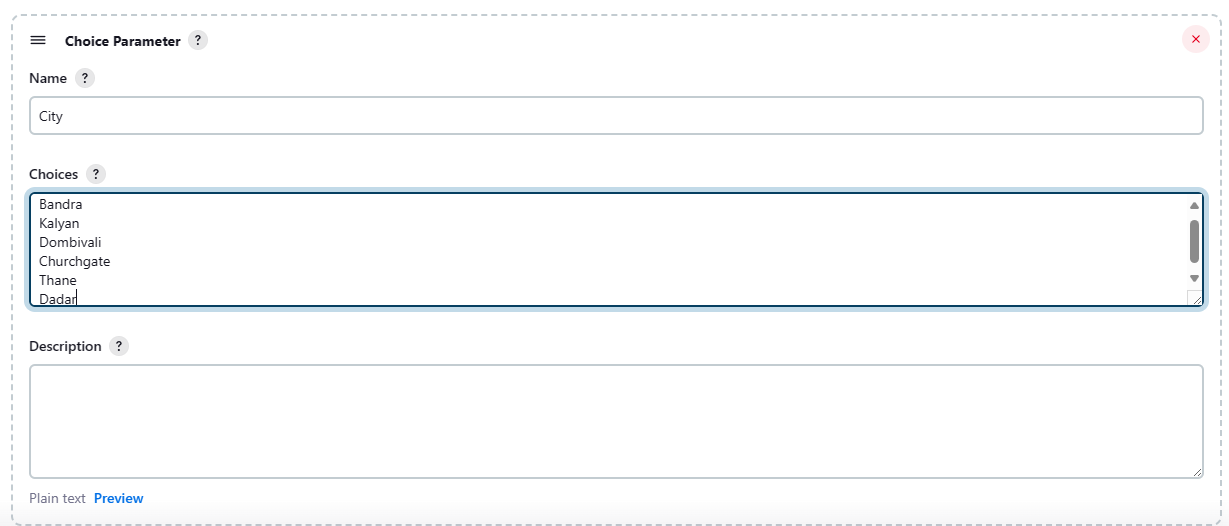
Description automatically generated

Configuring the string parameter as Fname:

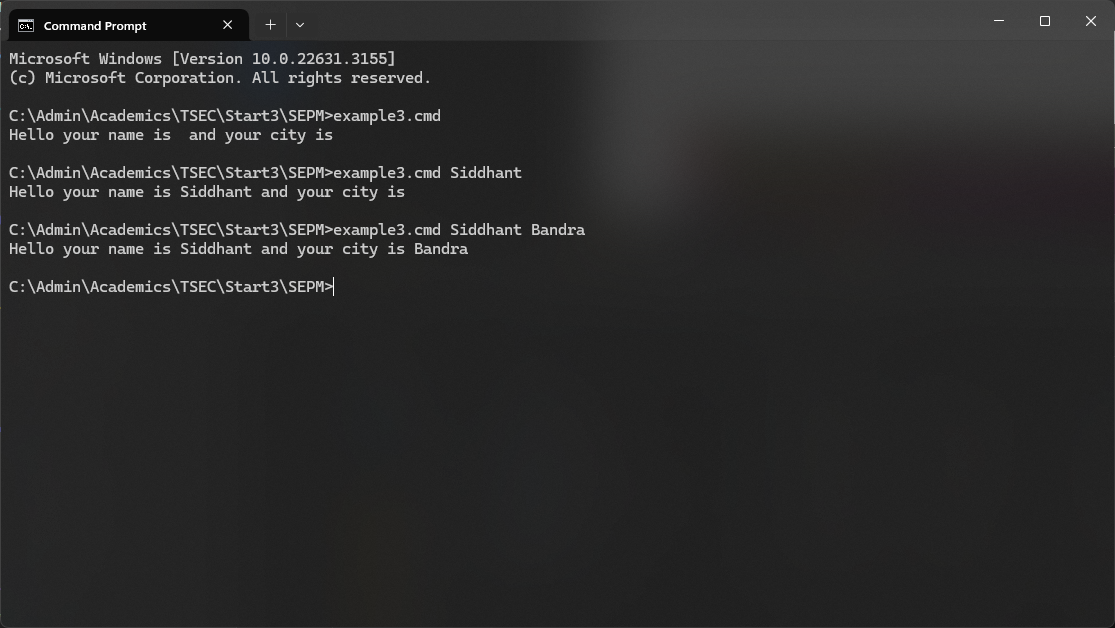
A white paper with lines

Description automatically generated with medium confidence

Adding a choice parameter and configuring it as City with the following choices:



Creating a script which takes 2 arguments for name and city:



Configuring build steps:

A screenshot of a computer

Description automatically generated

Entering parameters for build:

A blue and white line

Description automatically generated

Console output after building:

A close-up of a white rectangular object

Description automatically generated

Example 3.2: Running a Java program with parameters

Creating a Java program with an input argument:

A screenshot of a computer

Description automatically generated

Testing the program on the terminal:

A screenshot of a computer

Description automatically generated

Creating a new freestyle project:

A screenshot of a computer

Description automatically generated

Parameterise the project by adding a string parameter as follows:

A blue line on a white background

Description automatically generated

Configure the build steps:

A screenshot of a computer

Description automatically generated

Entering the parameter for the build:

A blue and white rectangle

Description automatically generated

Console output after building:

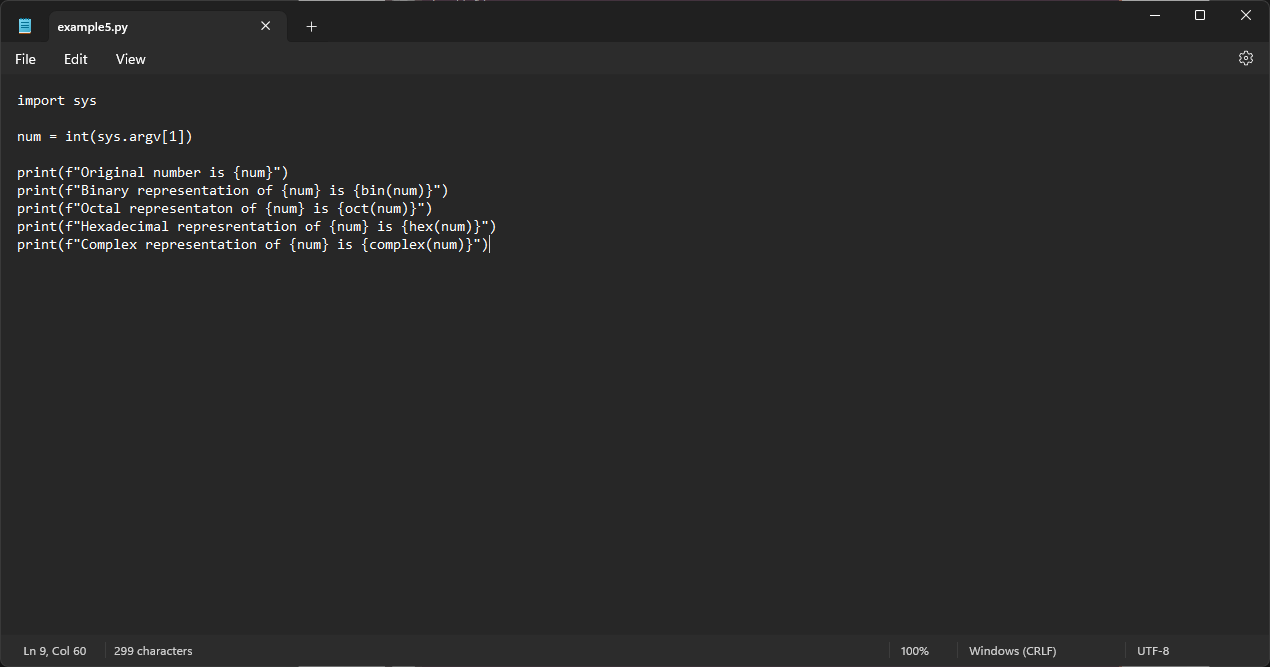
A close-up of a white screen

Description automatically generated

Example 5

Example 5.1: Running a Python program

Creating a simple Python script:



Running the Python script on the terminal:

A screenshot of a computer

Description automatically generated

Creating a new freestyle project:

A screenshot of a computer

Description automatically generated

Parameterising the project with a string parameter as follows:

A blue line on a white background

Description automatically generated

Configuring the build steps:

A screenshot of a computer

Description automatically generated

Setting the parameter for the build:

A blue and white rectangle

Description automatically generated

Console output after building:

A close-up of a computer screen

Description automatically generated

**Conclusion:** Thus, we have successfully studied Continuous Integration and installed, configured, and understood programming with Jenkins.