

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### 

### 

### 

### 

### CICD USING JENKINS – TIP CALCULATOR BY

### Team – A11

### **K.Harini(21JG1A0552) K.Akshaya(21JG1A0548)**

### **D.Vineetha(21JG1A0520) A.Kavya(21JG1A0505)**

Under the esteemed guidance of

**Dr.P.V.S.Lakshmi Jagadamba**

**Professor**

Head of CSE Department

## 

## **Department of Computer Science and Engineering**

## **GAYATRIVIDYAPARISHADCOLLEGEOFENGINEERINGFORWOMEN**

[Approved by AICTE NEW DELHI, Affiliated to JNTUK Kakinada]

[Accredited by National Board of Accreditation (NBA) for B.Tech. CSE, ECE & IT – Valid from 2019-20 to 2021-22]

Kommadi , Madhurawada, Visakhapatnam–530048

### 2022–2023

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



|  |
| --- |
|  |
|  |  |

## **CERTIFICATE**

This is to certify that the project report titled **“tip calculator”** is a bonafide work of following III B.Tech. students in the Department of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering for Women affiliated to JNT University, Kakinada during the academic year 2023-2024 Semester-1.

### **Name: A.Kavya(21JG1A0505) Name:K.Akshaya(21JG1A0548)**

### **Name: D.Vineetha(21JG1A0520) Name: K.Harini(21JG1A0552)**

***Dr.P.V.S.LakshmiJagadamba***

#### **Professor**

#### **(Head of the department)**

**TABLE OF CONTENTS**

TOPICS PAGE NO.

|  |  |  |
| --- | --- | --- |
| **1.** | **ABSTRACT** | **4** |
|  | * 1. Problem Statement   2. Features |  |
| **2.** | **APPLICATION**  2.1 Software/Hardware Requirements  2.2 Methodology  2.3 Modern Architecture  2.4 Implementation and Output | **5-11**  5  6  7  7 |
| **3.** | **CICD using Jenkins** | **11-35** |
|  | 3.1 Git | 11 |
|  | 3.2 Jenkins  3.2.1. Steps for installing Jenkins  3.2.2. Plugins used in Our Project  3.2.3. Create a job  3.2.4. Add a Post build action  3.2.5. Build the Project | 15  15  18  23  24  25 |
| **4.** | **CONCLUSION** | **37** |

### 

**Continuous Integration and continuous deployment using Jenkins**

Tip calculator

1) **ABSTRACT:**

The Tip Calculator offers users a straightforward and intuitive interface, allowing them to input their bill total, select their desired tip percentage, and even split the bill among multiple diners effortlessly. Moreover, the app includes a real-time currency converter for international travelers, ensuring that tipping is accurate and hassle-free regardless of the location.

FEATURES:

1)Intuitive Interface: The app's user-friendly design ensures quick and hassle-free bill total input and tip calculation.

2)Tip Customization: Users can easily select the desired tip percentage, catering to various service levels and preferences.

3)Effortless Bill Splitting: Going out with friends or dining with a group can often result in complicated bill-splitting scenarios. Tip Calc simplifies this process by allowing users to divide the bill fairly among multiple diners. No more pen and paper calculations; Tip Calc handles it efficiently.

4)Currency Converter: Real-time exchange rate information allows international travelers to tip in their preferred currency.

The " Tip Calculator" is a cutting-edge mobile application designed to revolutionize the way individuals calculate and manage tips during their dining experiences. In an era of digital innovation, this app combines user-friendly design with advanced features to ensure accuracy and convenience in the process of tipping.

A tip calculator is a handy tool designed to assist individuals in quickly and accurately determining the appropriate tip to leave when dining at a restaurant or partaking in other services where gratuities are customary. This user-friendly application streamlines the process by allowing users to input the total bill amount and select the desired tip percentage, ensuring a straightforward and efficient calculation. Additionally, it often offers features for splitting the bill among multiple diners and may include a currency converter for international travelers.

**2.) APPLICATION :**

**2.1 Software/Hardware requirements:**

2.1.1 SOFTWARE REQUIREMENTS-

A tip calculator is a relatively simple software application, and its software and hardware requirements are generally quite minimal. Here are the basic software and hardware requirements for a typical tip calculator:

1. Operating System: Most tip calculators are developed as mobile apps or web applications. As such, they may need to be compatible with various operating systems, including:

* For Mobile Apps: Android and iOS (for smartphones and tablets).
* For Web Apps: Cross-browser compatibility (e.g., Chrome, Firefox, Safari, Edge).

1. Development Tools: The software requirements include the tools and programming languages used for development. Common requirements include

* Development Environment: Integrated Development Environment (IDE) for the chosen platform (e.g., Android Studio, Xcode, Visual Studio Code).
* Programming Languages: Java (for Android), Swift (for iOS), HTML, CSS, and JavaScript (for web apps).

2.1.2 Hardware Requirements-

1)Processor and Memory: Tip calculators are lightweight applications, and they do not require significant processing power or memory. A typical smartphone or modern computer should suffice.

2)Storage Space: Tip calculator apps are small in size and do not consume much storage space on the device.

3)Internet Connection (for real-time features): If your tip calculator offers real-time currency conversion or other online features, an internet connection is required.

4)Screen Resolution and Size: The user interface of the tip calculator should be designed to fit different screen sizes and resolutions, from small mobile screens to larger desktop displays.

**2.2 Methodology:**

Developing a tip calculator is a relatively simple software project, and you can use an Agile methodology to guide the development process. Agile methodologies are well-suited for small to medium-sized software projects with changing or evolving requirements. Here's a basic methodology for creating a tip calculator:

1)Project Initiation:

Define the Scope: Clearly define the scope of the tip calculator. Identify the essential features, such as bill input, tip percentage selection, and bill splitting.

2)Team Setup:

Assemble a small development team with roles such as a developer, designer (for UI/UX), and a product owner or project manager.

1. User Stories and Requirements: Create user stories or use cases that describe the functionality from a user's perspective. For example:

* "As a user, I want to input the bill amount."
* "As a user, I want to select the tip percentage."
* "As a user, I want to split the bill among multiple people."

4)Testing:

Conduct thorough testing to identify and fix bugs and ensure the software works as expected.

Test scenarios should include different bill amounts, tip percentages, and bill-splitting options.

5)Documentation:

Maintain documentation that includes user manuals or in-app help to guide users on how to use the tip calculator effectively.

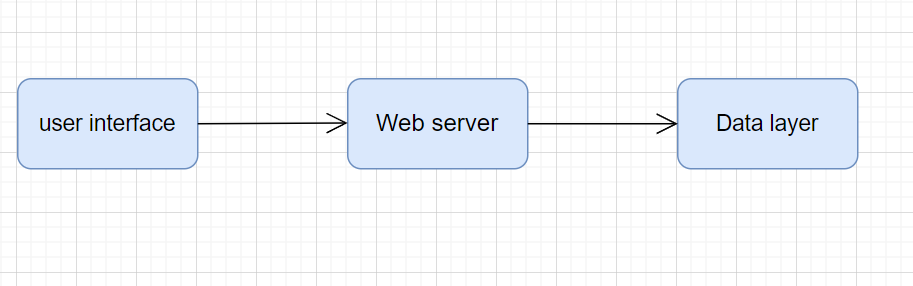
1. Deployment:

Once the software meets the desired functionality and quality standards, prepare it for deployment on the intended platform (e.g., mobile app stores, web hosting).

7)Maintenance:

Continuously monitor and maintain the software to address any issues, ensure compatibility with updated platforms or devices, and implement improvements.

**2.3 Modern Architecture:**

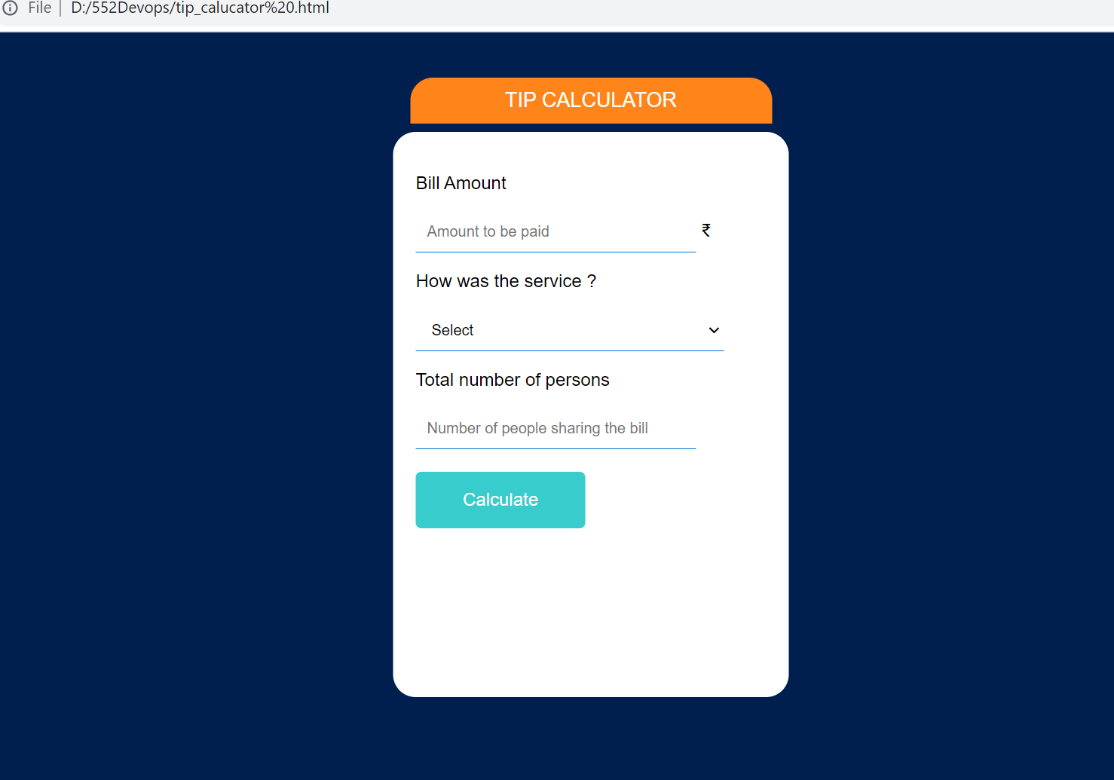
****

**2.4 Implementation:**

SAMPLE CODES:

|  |
| --- |
| <html>  <head>  <style>  body {  background-color: #001f4f;  font-family: 'Raleway', sans-serif;  }    .container {  width: 350px;  height: 500px;  background-color: #fff;  position: absolute;  left: 50%;  top: 50%;  transform: translateX(-50%) translateY(-50%);  border-radius: 20px;  }    h1 {  position: absolute;  left: 50%;  top: -60px;  width: 300px;  transform: translateX(-50%);  background-color: #FF851B;  color: #fff;  font-weight: 100;  border-top-left-radius: 20px;  border-top-right-radius: 20px;  font-size: 18px;  text-align: center;  padding: 10px;  }    .wrapper {  padding: 20px;  }    input,  select {  width: 80%;  border: none;  border-bottom: 1px solid #0074D9;  padding: 10px;  }    input:focus,  select:focus {  border: 1px solid #0074D9;  outline: none;  }    select {  width: 88% !important;  }    button {  margin: 20px auto;  width: 150px;  height: 50px;  background-color: #39CCCC;  color: #fff;  font-size: 16px;  border: none;  border-radius: 5px;  }    .tip {  text-align: center;  font-size: 18px;  display: none;  }  </style>  </head>  <body>  <div class="container">  <h1>TIP CALCULATOR</h1>  <div class="wrapper">  <p>Bill Amount</p>  <input type="text" id="amount" placeholder="Amount to be paid"> ₹  <p>How was the service ?</p>  <select id="services">  <option selected disabled hidden>Select</option>  <option value="0.25">25% - Top Notch</option>  <option value="0.20">20% - Excellent</option>  <option value="0.15">15% - Good</option>  <option value="0.10">10% - Bad</option>  <option value="0.05">5% - Worst</option>  </select>  <p>Total number of persons</p>    <input type="text" id="persons" placeholder="Number of people sharing the bill">  <button id="calculate">Calculate</button>  </div>  <div class="tip">  <p>Tip Amount</p>  <span id="total">0</span>₹  <span id="each">each</span>  </div>  </div>  <script>  //the function called when Calculate button is clicked.  window.onload = () =>  {  /\*calling a function calculateTip which  will calculate the tip for the bill.\*/  document.querySelector('#calculate').onclick =  calculateTip;  }    function calculateTip() {  /\*assign values of ID : amount, person and  service to variables for further calculations.\*/  let amount = document.querySelector('#amount').value;  let persons = document.querySelector('#persons').value;  let service = document.querySelector('#services').value;    console.log(service);  /\*if statement will work when user  presses calculate without entering values. \*/  //so will display an alert box and return.  if (amount === '' && service === 'Select') {  alert("Please enter valid values");  return;  }    //now we are checking number of persons  if (persons === '1')  //if there is only one person then we need not to display each.  document.querySelector('#each').style.display = 'none';  else  //if there are more than one person we will display each.  document.querySelector('#each').style.display = 'block';    /\*calculating the tip by multiplying total-bill and type of service;  then dividing it by number of persons.\*/  //fixing the total amount upto 2 digits of decimal  let total = (amount \* service) / persons;  total = total.toFixed(2);    //finally displaying the tip value  document.querySelector('.tip').style.display = 'block';  document.querySelector('#total').innerHTML = total;  }  </script>  </body>  </html> |

Output:



**3.) CICD USING JENKINS:**

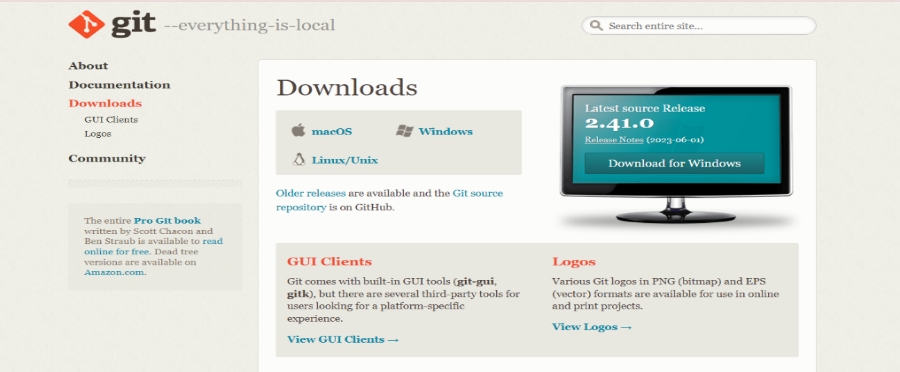
**3.1 Git Bash:**

Git Bash is an application for Microsoft Windows environments which provides an emulation layer for a Git command line experience. Bash is an acronym for Bourne Again Shell. A shell is a terminal application used to interface with an operating system through written commands.

**2.1.1 Installation and Setup of Git:**

1) Download GIT from the website below based on your system’s operating system and configuration.



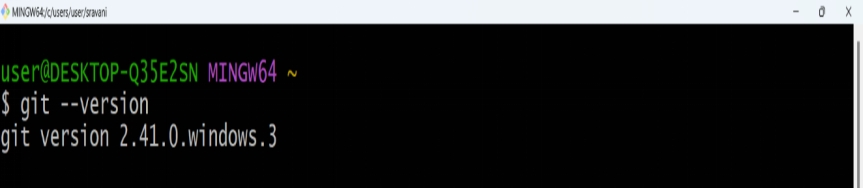




2) Install GIT by clicking next for a few times.

3) After installation is complete go to start menu and search for Git Bash and open it.

4) Check the Git version using the command git –version.



1. Create a local directory using the following commands:

**mkdir directory\_name**

**cd directory\_name**

6)The next step is to initialize the directory using the git init command as shown below.



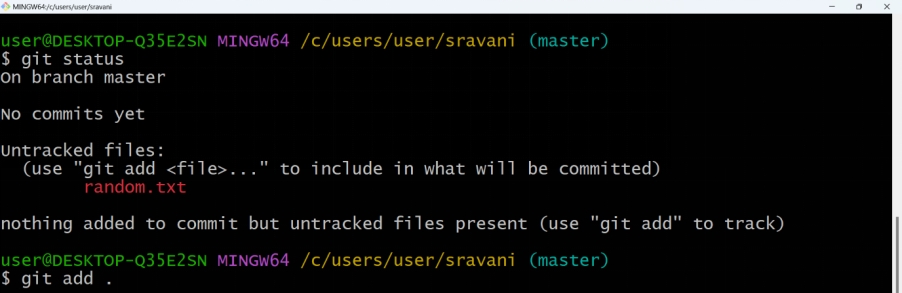
1. Create a text file inside the newly created local directory, write some content and save it.



1. Enter the Git bash interface and type in the following command to check the status of the files inside the local directory using: **git status**

1. **git add [file name]** or **git add .** tracks the respective files and are ready to be committed into the GitHub repository.

10)Every time you make modifications to a particular file, you need to add it using git add command to make it ready to be committed.

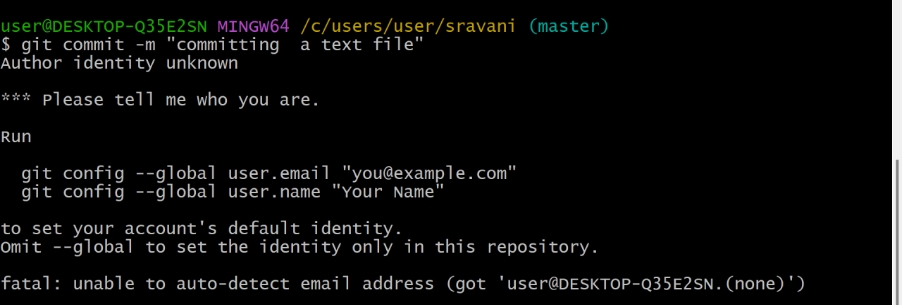


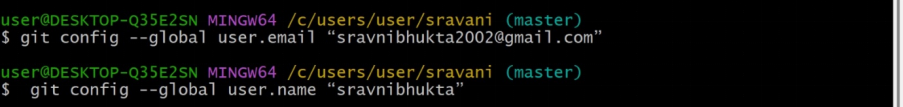
11) Next, make a commit using **git commit -m "committing message"** as shown below.

12) Link the Git to a GitHub Account using following commands as shown:

**git config --global user.email “your email linked to GitHub account”**

**git config --global user.name “your GitHub username”**





13) Open your GitHub account and create a **new repository** with any name of your choice.

14) **Public repositories** are accessible to everyone on the internet while the **Private repositories** are only accessible to you and the people you explicitly share access to.

15) You can add a **README file** to a repository to communicate important information about your project.

**3.2 Jenkins:**

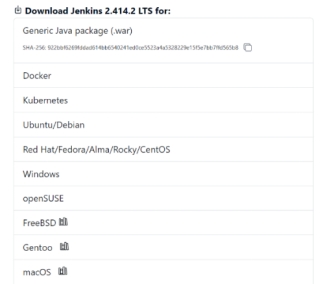
Jenkins is an open source continuous integration/continuous delivery and deployment (CI/CD) automation software DevOps tool written in the Java programming language

There are 2 steps involved in setting up Jenkins completely. They are,

* Download and installation of Jenkins software
* Unlocking Jenkins with localhost

**2.2.1 Steps to download and install Jenkins:**

1) Go to the website **<https://www.jenkins.io/download/>** where we can get Jenkins software for any type of Operating system (Windows, Linux, Mac, etc..). Select windows for Jenkins 2.414.2 LTS.

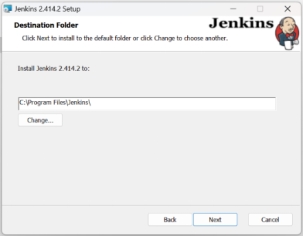


1. After selecting windows, Jenkins.exe file starts downloading.

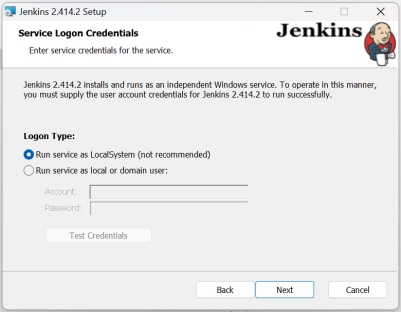
Go to Downloads -> Double click on Jenkins.exe file. Click on Next.



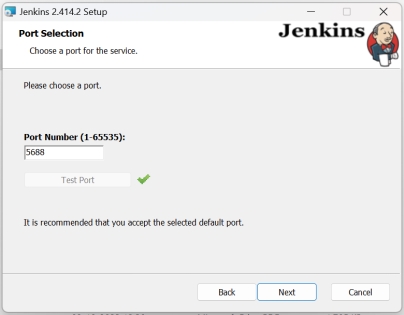
1. Now you will be asked to set the destination for the folder. There is no need to change the path. Click on Next.



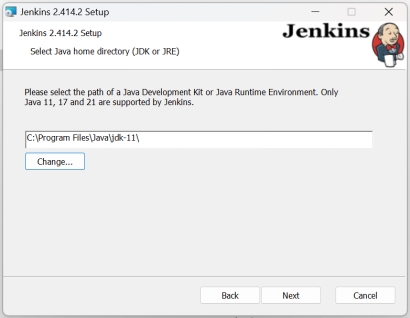
1. Now you will be asked to enter the services logon credentials. Select Run service as LocalSystem and click on Next.

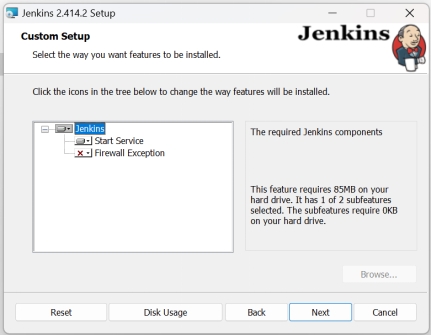


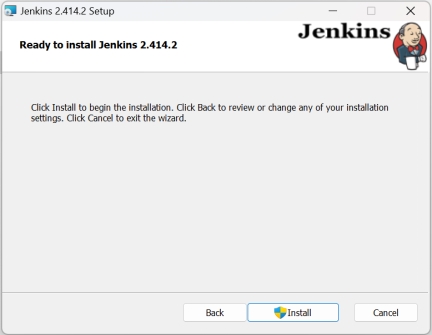
1. Now you will be asked to select a port number for localhost. By default it is 8080 if it is not available, then you have to choose one from the available ports in your device. You can use that number with localhost when the test post result is success. Click on Next.



1. As Jenkins runs on Java Environment, we must give Java path to Jenkins. Jenkins supports only Java 11, 17 or 21. After giving path of Java, Click on Next.



1. Now a custom setup window opens. Click on Next without making any changes in it.
2. Click on Install. Now the installation of Jenkins begins.

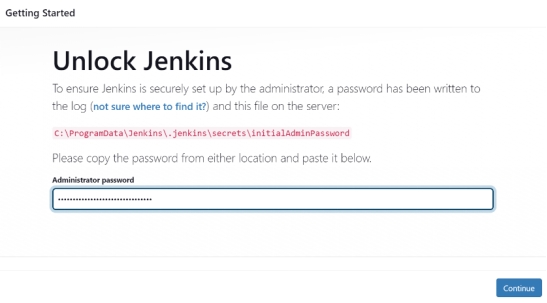


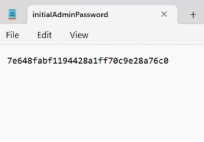
1. After the completion of Installation, Click on Finish.



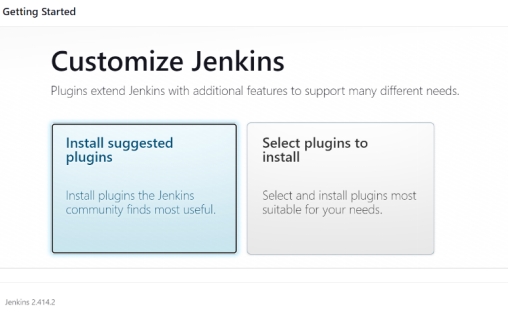
* + 1. **Steps to unlock the Jenkins after Installation:**

1. Open your web browser. Type localhost:5688 (your port number). After pressing Enter you will be opened with a Getting started window asking for a password. To find the password open the location mentioned and copy paste the code. Click on Continue.

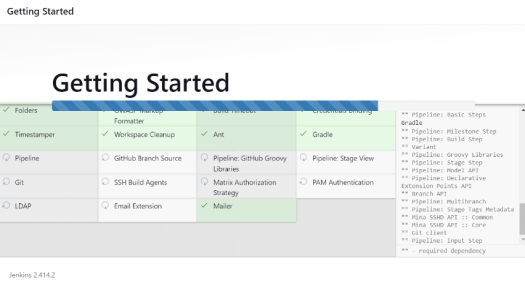
****

****

1. Select Install suggested plugins.

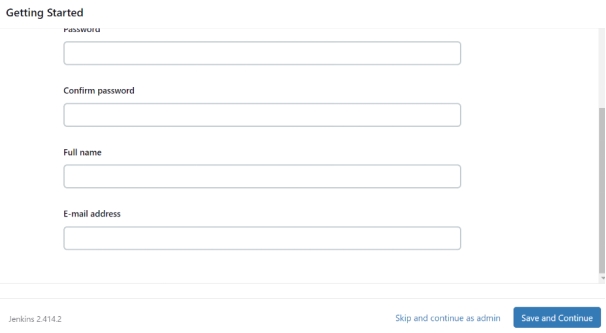
****

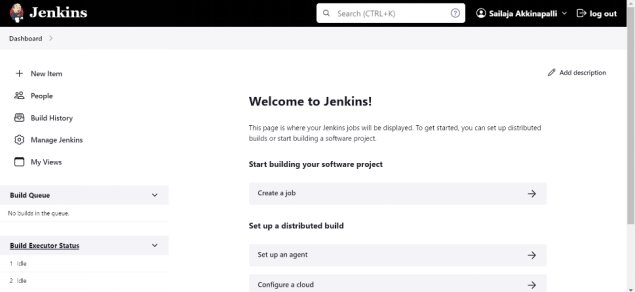
1. Now all the required plugins will be downloaded one by one.



1. Now create an account by filling all the required details. Click on Save and Continue after filling all the details.

****

****

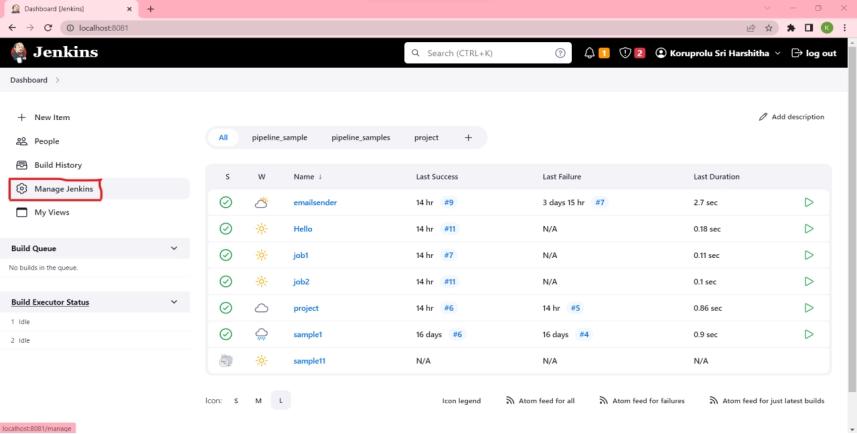
1. After clicking on Save and Continue, you successfully completed setting up your ****profile in Jenkins.

**3.2.2Plugins used in our project:**

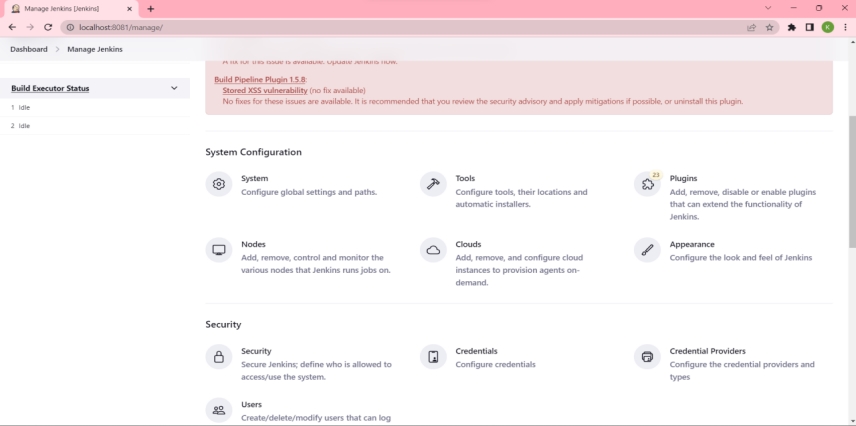
* **Post build task:** This plugin allows the user to execute a shell/batch task depending on the build log output. Java regular expression are allowed.
* **Simple Theme:** This plugin allows to customize Jenkin's appearance with custom CSS and JavaScript. It also allows to replace the Favicon. For a more user-centric approach to theming, take a look at the Theme Manager plugin

**Steps to install plugins:**

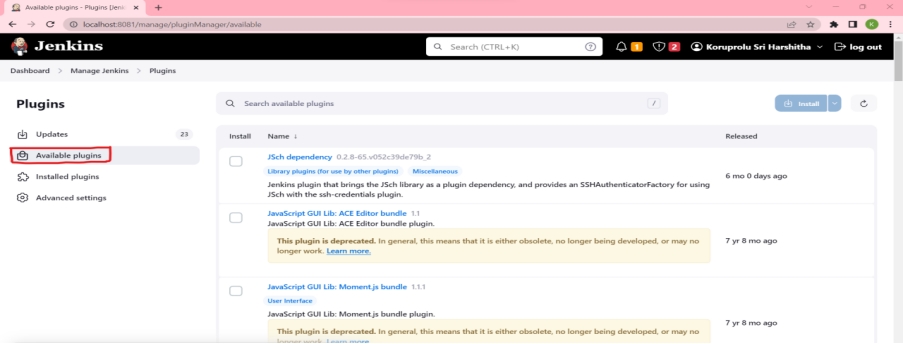
* In Jenkins dashboard click on manage Jenkins on the left hand side of Jenkins dashboard.



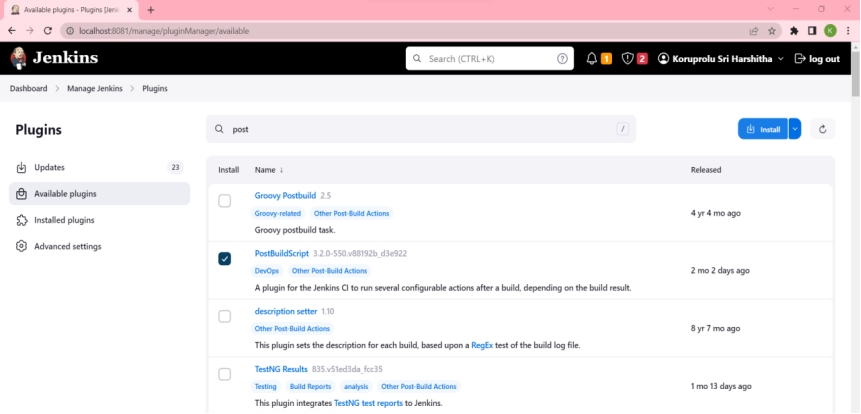
* Scroll down to plugins section and click on plugins.



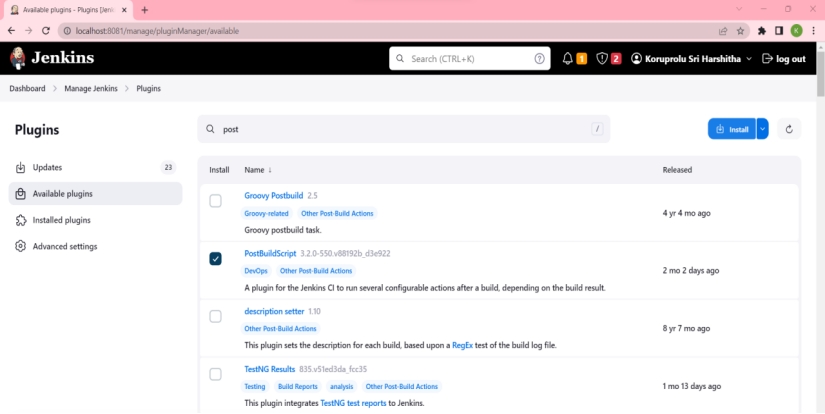
* After that click on “available plugins”.



* Search the plugin you want and enable the plugin.

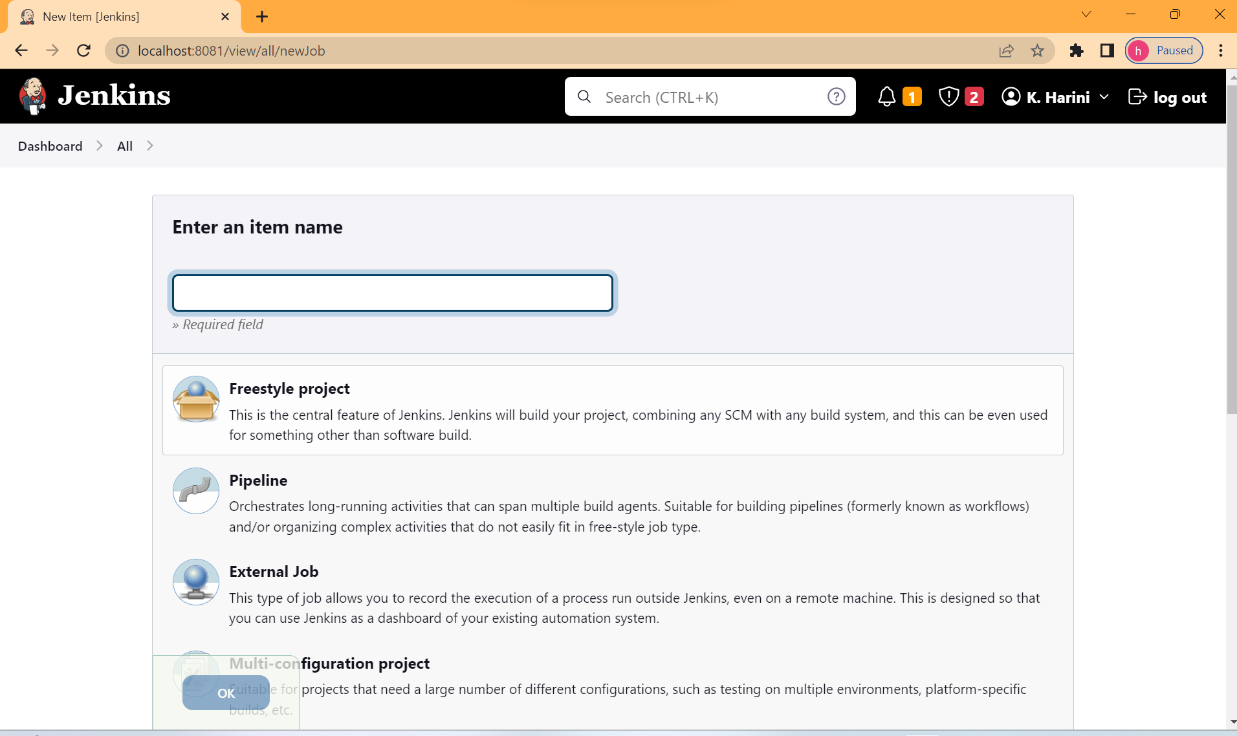


* Now click on install.

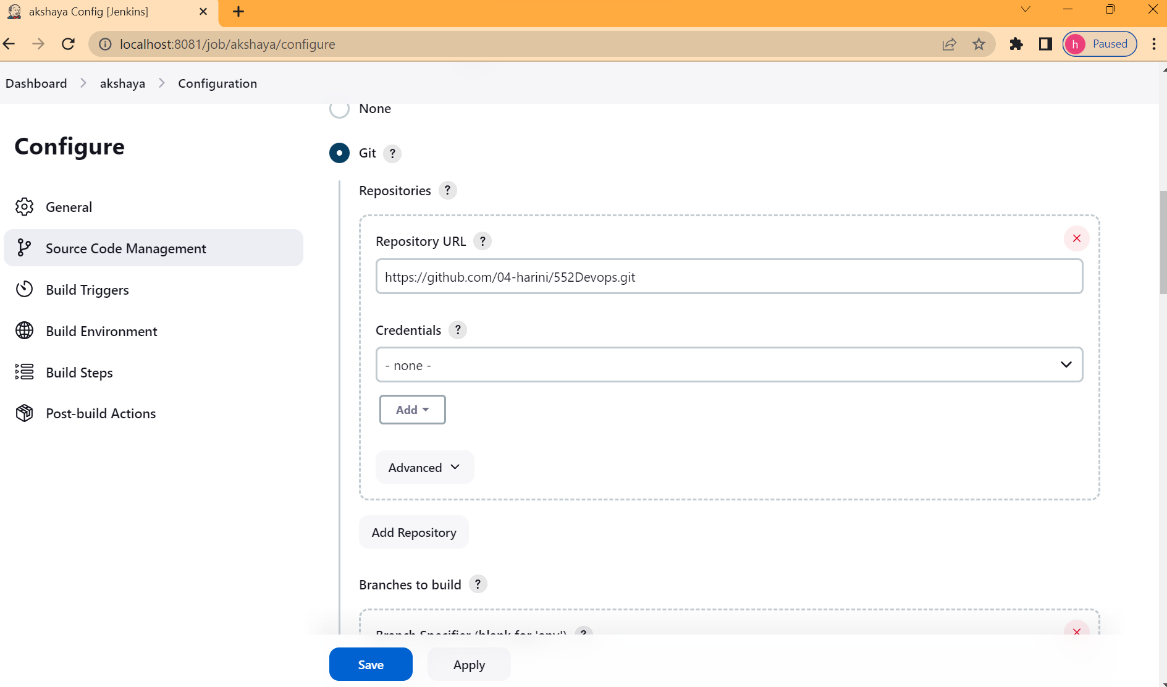


**3.2.3Create a job:**

* Click NEW ITEM link on the left hand side of Jenkins dashboard.
* Enter the new project’s name in the enter an item name field and select the Freestyleproject type. Click ok to create a job.

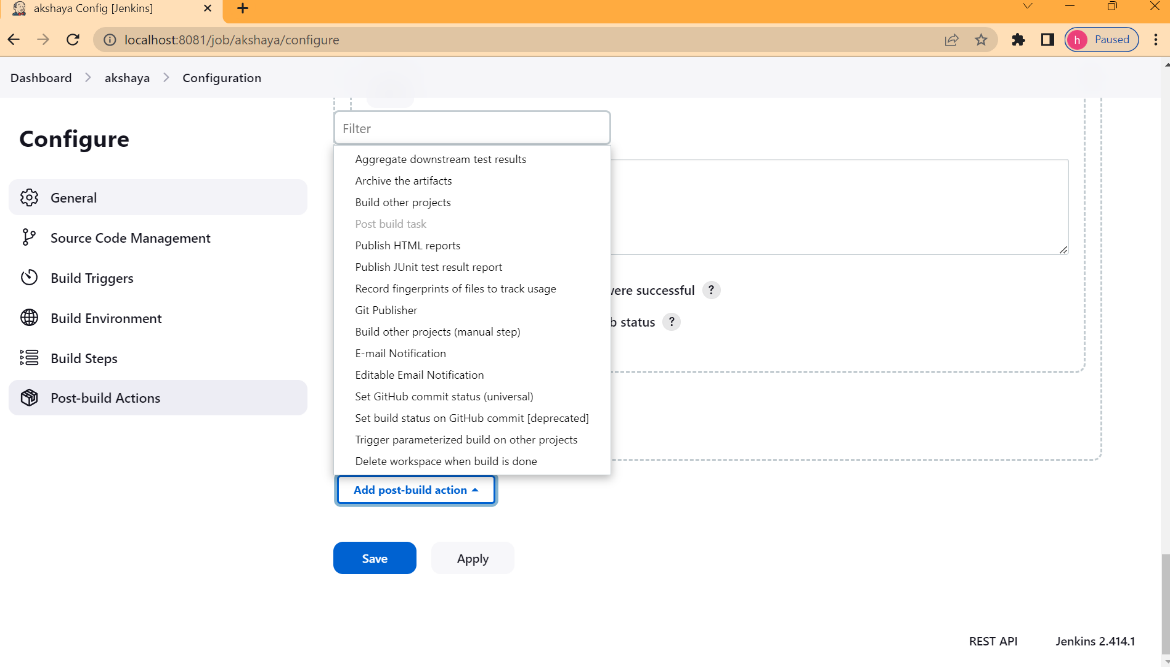


* Under general tab add project description in description field.
* Then scroll down to “Source code management”. In that click git and add git repository url of your project and add your project branch.

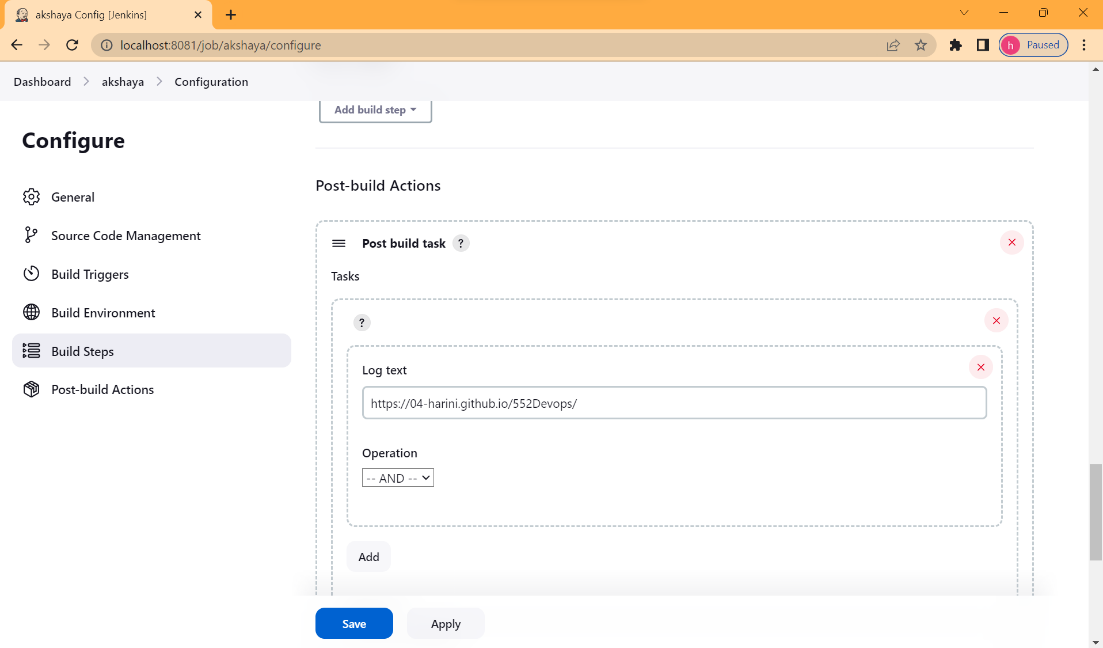


**3.2.4 Add a post build action:**

* Then scroll down to “post build action” section.
* Open the post build action drop-down menu and select post build task.

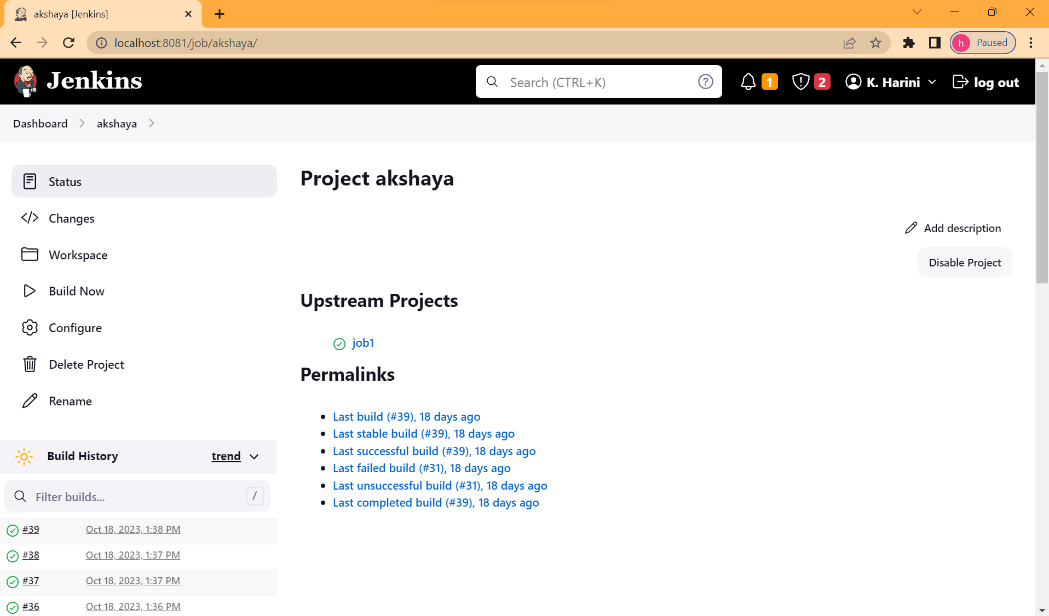


* Enter the log text and script of your project.
* Click “save” button to save the changes to the project.

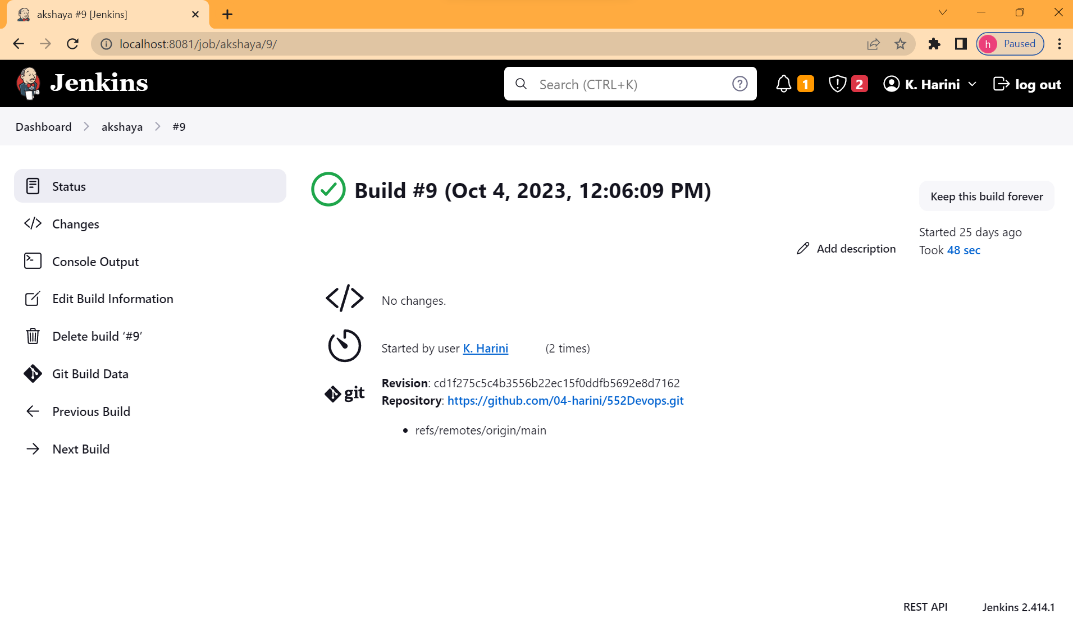


**3.2.5Build the project:**

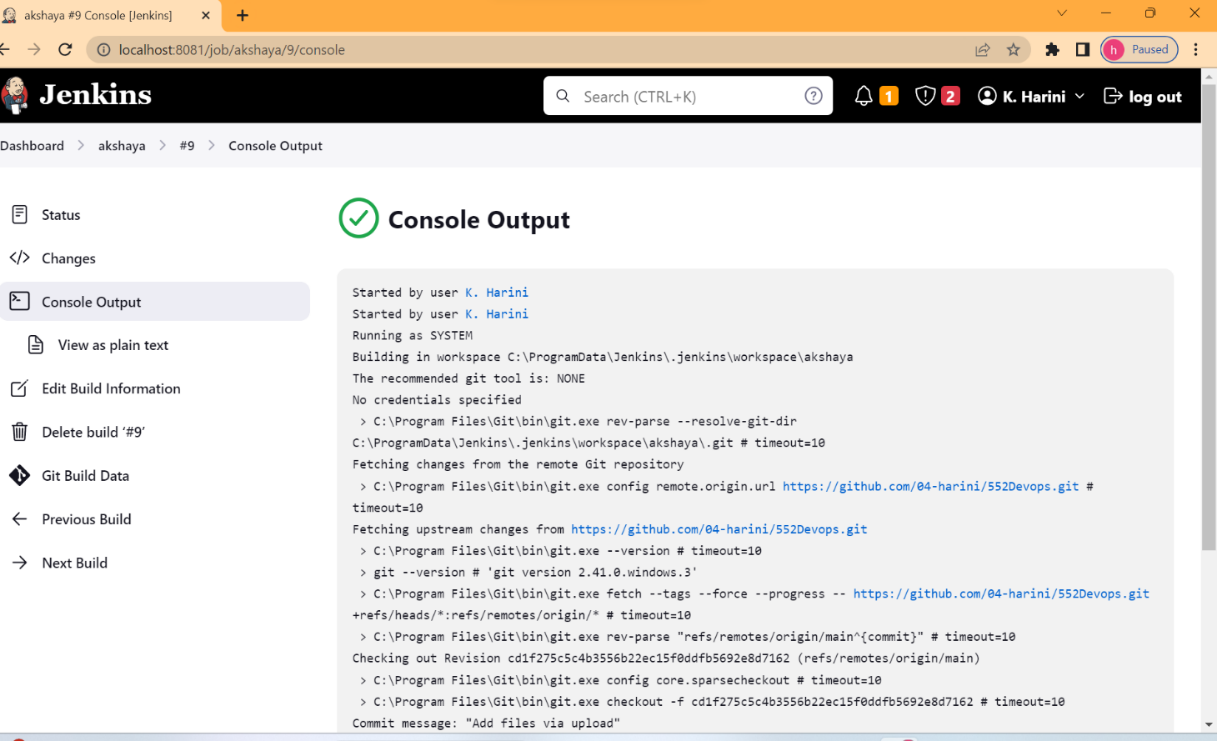
* Click the “Build now” link on the left-hand side of the new project page.



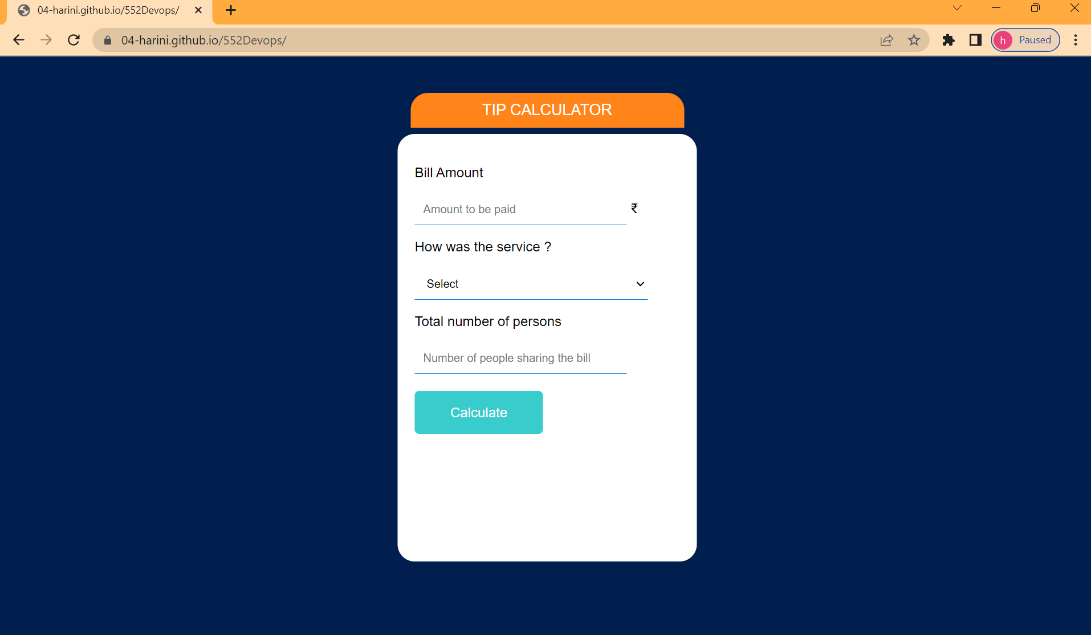
* Click the link to the latest project build in the Build history section.
* Click “console output” link on the left-hand side to display the output link.



* The console output indicates that Jenkins is successfully executed, displays the link of your project.
* Click on the link to display the output.



* Our output screen will look like this.



**2.2.6 Implementation of Jenkins Pipelines:**

**2.2.6.1 What is a pipeline in Jenkins ?**

Jenkins is a popular open-source automation server that allows you to automate various aspects of the software development and deployment process. Jenkins pipelines are a way to define and automate your continuous integration and continuous delivery (CI/CD) workflows within Jenkins. Jenkins pipelines provide a way to represent your build, test, and deployment processes as code, which makes it easier to version, manage, and share your automation workflows.

There are two main types of pipelines in Jenkins:

* **Scripted Pipeline:**

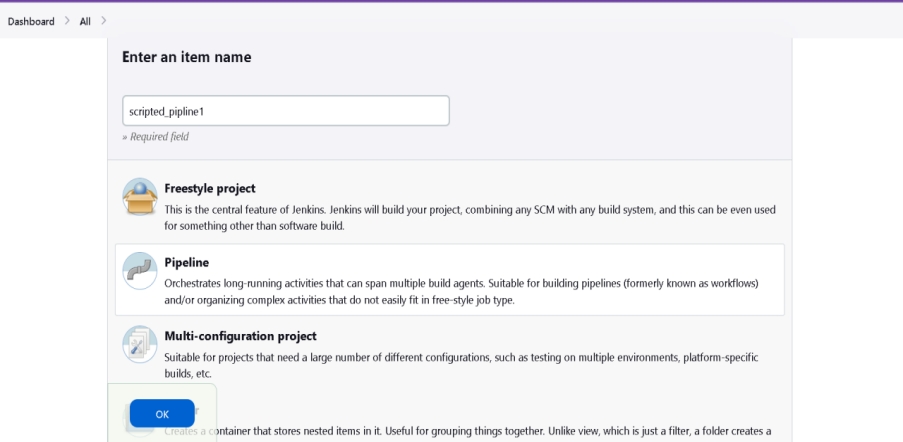
This is the traditional way of defining Jenkins pipelines using Groovy-based DSL (Domain Specific Language). Scripted pipelines are written in a Groovy script and provide great flexibility, but they can be complex to manage and maintain.

* **Declarative Pipeline:**

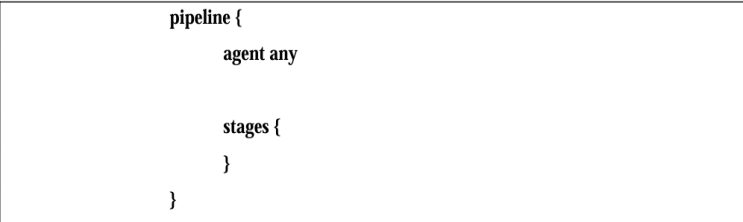
Declarative pipelines are a more structured and human-readable way to define your CI/CD workflows. They are defined using a simpler syntax and are recommended for those who want to get started quickly and maintain pipelines more easily.

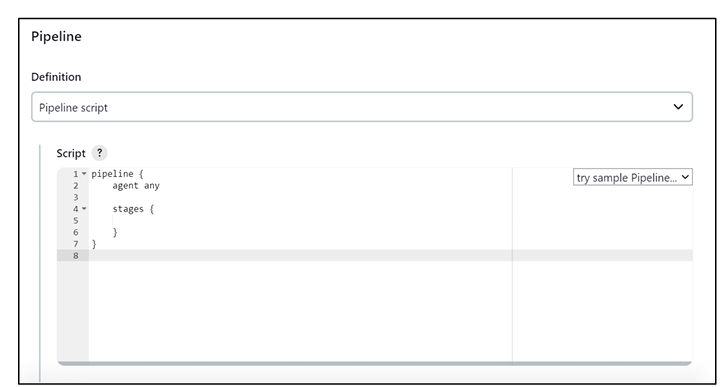
**Deploymemt of a job using Declarative Pipeline:**

* Open Jenkins Dashboard. Click on New Item.
* Enter a name for Pipeline. Select Pipeline and Click OK.



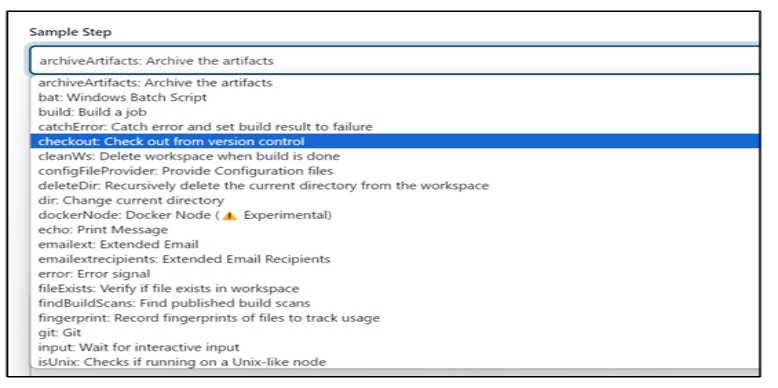
* Scroll down until you see Pipeline, type the following in the Script Box.

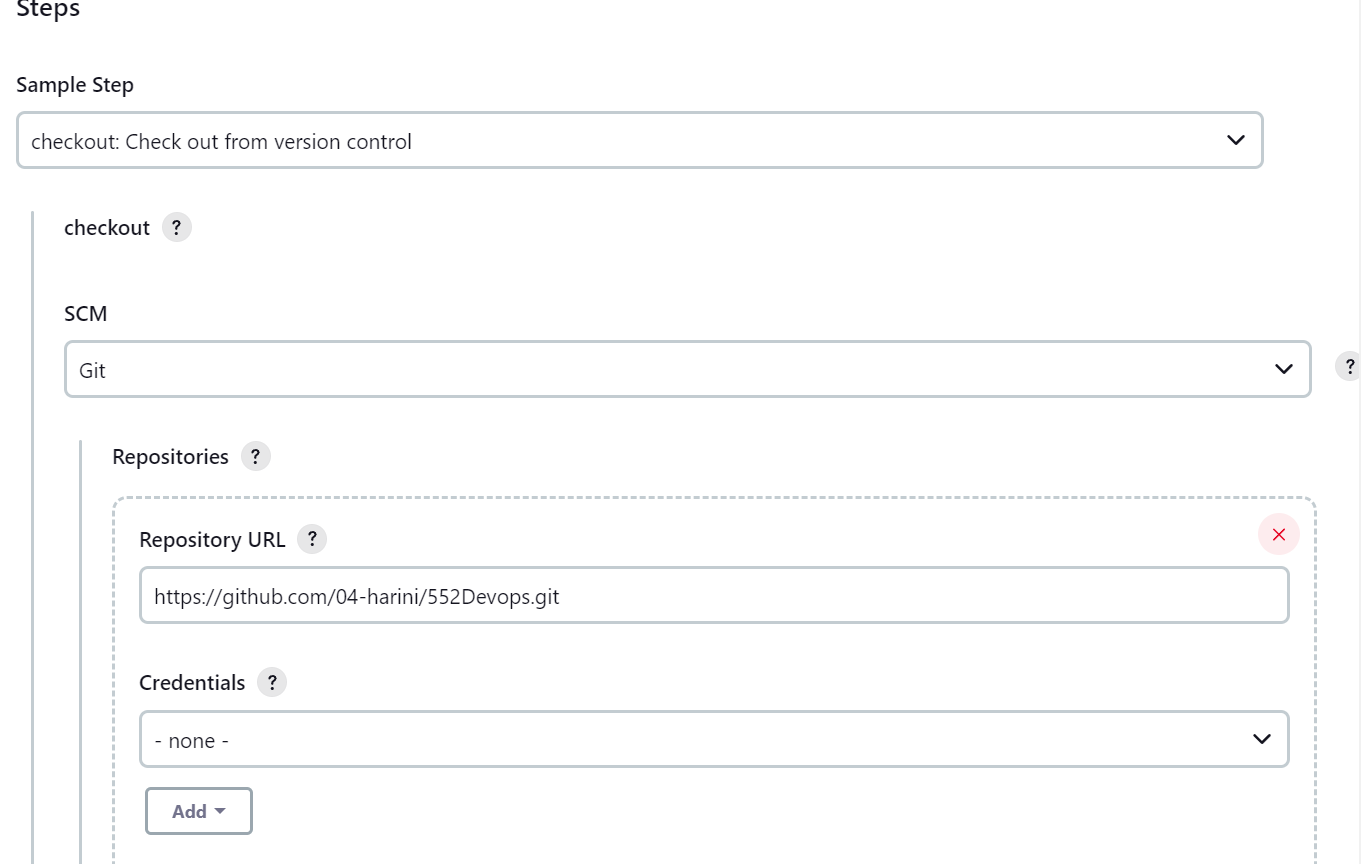




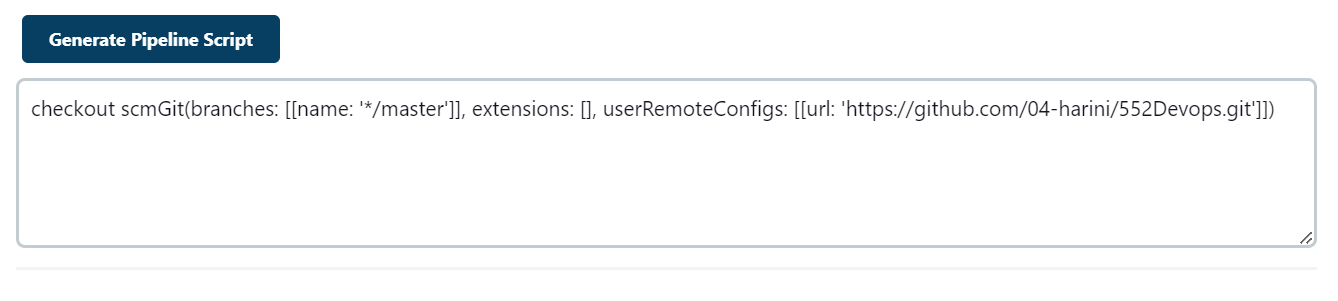
Now inside stages, we create different stages which are executed for their respective purposes. We must make sure that each stage name should be unique to avoid failed build.

➢ Stage-1: checkout Click on Pipeline Syntax which is just below the Script Box. In Sample Step dropdown, Select checkout, paste the Repository URL and branch name, Click on Generate Pipeline Script which is below the Sample Step.









After clicking on Generate Pipeline Script, you will get a statement. Copy paste the statement inside Checkout stage in the Script.

|  |
| --- |
| pipeline {  agent any  stages {  stage('checkout') {  steps {  checkout scmGit(branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/04-harini/552Devops.git']])  }  }  }  } |

Stage-2: git As we are building a project which is pushed into GitHub, we need git pipeline to connect our project from Git to Jenkins. Click on Pipeline Syntax which is just below the Script Box. In Sample Step dropdown, Select git, paste the Repository URL and branch name, Click on Generate Pipeline Script which is below the Sample Step.After clicking on Generate Pipeline Script, you will get a statement. Copy paste the statement inside Git stage in the Script.



* Stage-3: publishHTML As we are building a HTML web page in Jenkins publishHTML pipeline is important to get the output of our project. Click on Pipeline Syntax which is just below the Script Box. In Sample Step dropdown, Select publishHTML, paste the project folder name as HTML directory and first introduction page name as Index page, Click on Generate Pipeline Script which is below the Sample Step.
* After clicking on Generate Pipeline Script, you will get a statement. Copy paste the statement inside HTML stage in the Script.





Stage-4: Output

|  |
| --- |
| stage('output'){  steps{  script{  echo 'https://04-harini.github.io/552Devops/'  }  }  } |

Final code:

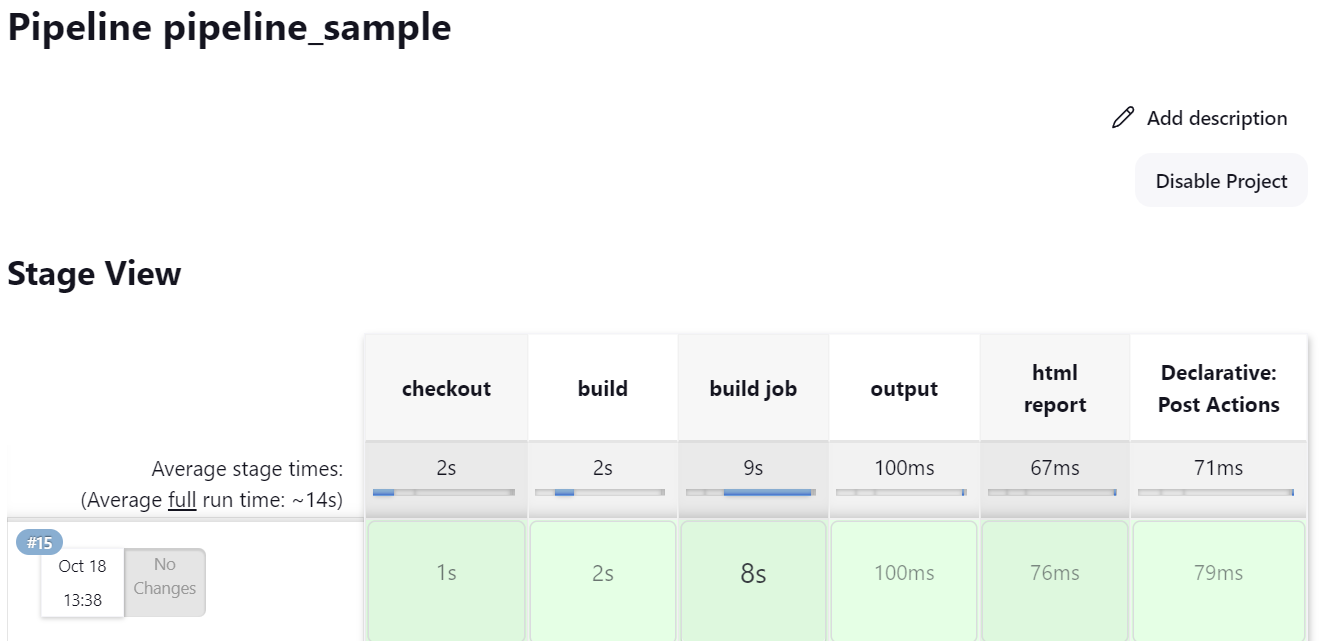
|  |
| --- |
| pipeline {  agent any  stages {  stage('checkout') {  steps {  checkout scmGit(branches: [[name: '\*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/04-harini/552Devops.git']])  }  }    stage('build'){  steps{  git branch: 'main', url: 'https://github.com/04-harini/552Devops.git'  }  }  stage('build job'){  steps{  build 'akshaya'  }  }  stage('output'){  steps{  script{  echo 'https://04-harini.github.io/552Devops/'  }  }  }  stage('html report'){  steps{ publishHTML([allowMissing:false,alwaysLinkToLastBuild: false, keepAll: false, reportDir: '', reportFiles: 'index.html', reportName: 'HTML Report', reportTitles: '', useWrapperFileDirectly: true])  }  }  }  post{  success{  echo 'this will be printed if the build success'  }  failure{  echo 'this will be printed failure'  }  always{  echo 'this will be printed even build failures'  }  }  } |

After writing the entire code, save and click on Build Now.

➢ It takes time to build as it consists of various stages

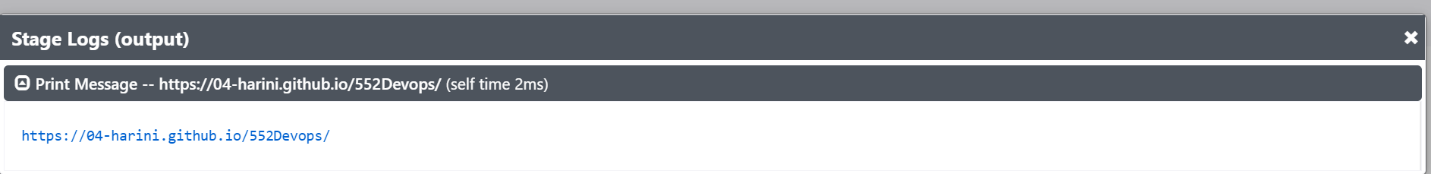
➢ After build is completed, if all stages are green it means that the entire building of pipeline is success.

➢ Otherwise, you must go to configure again, cross check the errors and build again until your pipeline is successfully build.

****

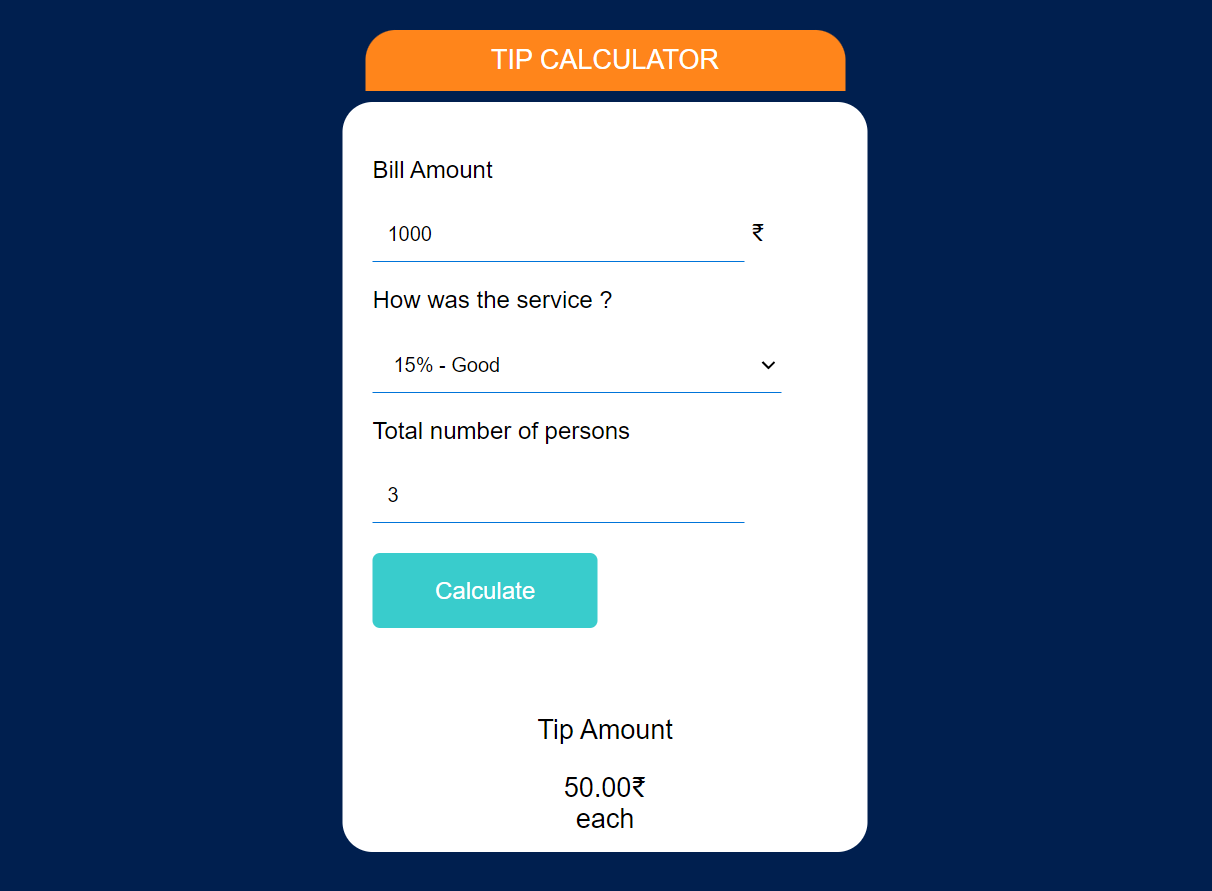
**Results:**

Click on the logs in the post-build task stage.  
Open the url provided.



After clicking the url Result of the tip calculator app displays:

****

****

4.conclusion

A tip calculator is a useful tool for quickly and accurately determining the amount of gratuity to leave when dining at a restaurant or receiving a service. It helps individuals avoid the hassle of manual calculations and ensures that the tip is fair and in line with their satisfaction and the level of service provided. The use of tip calculators can also help promote transparency and consistency in tipping practices, making it easier for both customers and service providers to understand and agree on the gratuity amount. Overall, tip calculators are a practical and convenient tool that can enhance the dining and service experience for both parties involved.

Tip calculators have become an essential tool for diners and customers in various service industries, offering numerous benefits. First and foremost, they simplify the often complex and sometimes anxiety-inducing task of determining an appropriate tip. With the help of a tip calculator, patrons can quickly and accurately calculate the gratuity based on their bill and the level of service they received. This eliminates the need for manual calculations, reduces the risk of mathematical errors, and ensures that the tip amount is in line with one's budget and satisfaction.

In summary, tip calculators are not only convenient but also contribute to a more equitable and informed tipping culture. They simplify the process of calculating tips, ensure that tips are in line with the quality of service received, and promote transparency in tipping practices. By using tip calculators, customers can navigate the sometimes awkward task of tipping with confidence, and service providers can receive fair compensation for their hard work.