Chat Application Using DevOps Tools

DevOps Project Report

Submitted by:

Neha Agarwal (9917103254)

Abhinav Verma (9917103251)

Anuranjan Dubey (9917103247)

Pranjal Das (9917103244)

Under the supervision of:

Shariq Murtuza

Department of Computer Science and Engineering

Jaypee Institute of Information Technology University, Noida

November 2020

PROJECT DESCRIPTION

CHAT APPLICATION

Main Components

1. Create React App.
2. Profile => Component depicts the randomly generated main user avatar, name.
3. Contact => Component depicts the randomly generated contact list (almost 15) and their recent messages.
4. MessageBox => Component shows the structure of the whole message box and messages (almost 50 per contact) are sorted according to date and time.
5. Message => Components show the structure of each message.
6. Input => Component for sending an input message, Things to note here is until we don't type, the microphone icon is shown on right, as soon we start typing it changes to the send button. Also, we can send messages by the stroke of entering Key, but we can't send an empty message.

Main Features

1. We can send and receive messages.
2. We can send messages with the stroke of Enter Key.
3. We can't send an empty message.
4. Messages are sorted on the basis of date and time.
5. Always most recent message will appear in the contact list, with atmost 30 character.
6. Until we don't start typing microphone icon is shown on the right, as soon we start typing it changes to the send button.
7. Messages are also sorted on the basis of date and time in the contact list as well.

Languages Used

1. HTML
2. CSS
3. Javascript

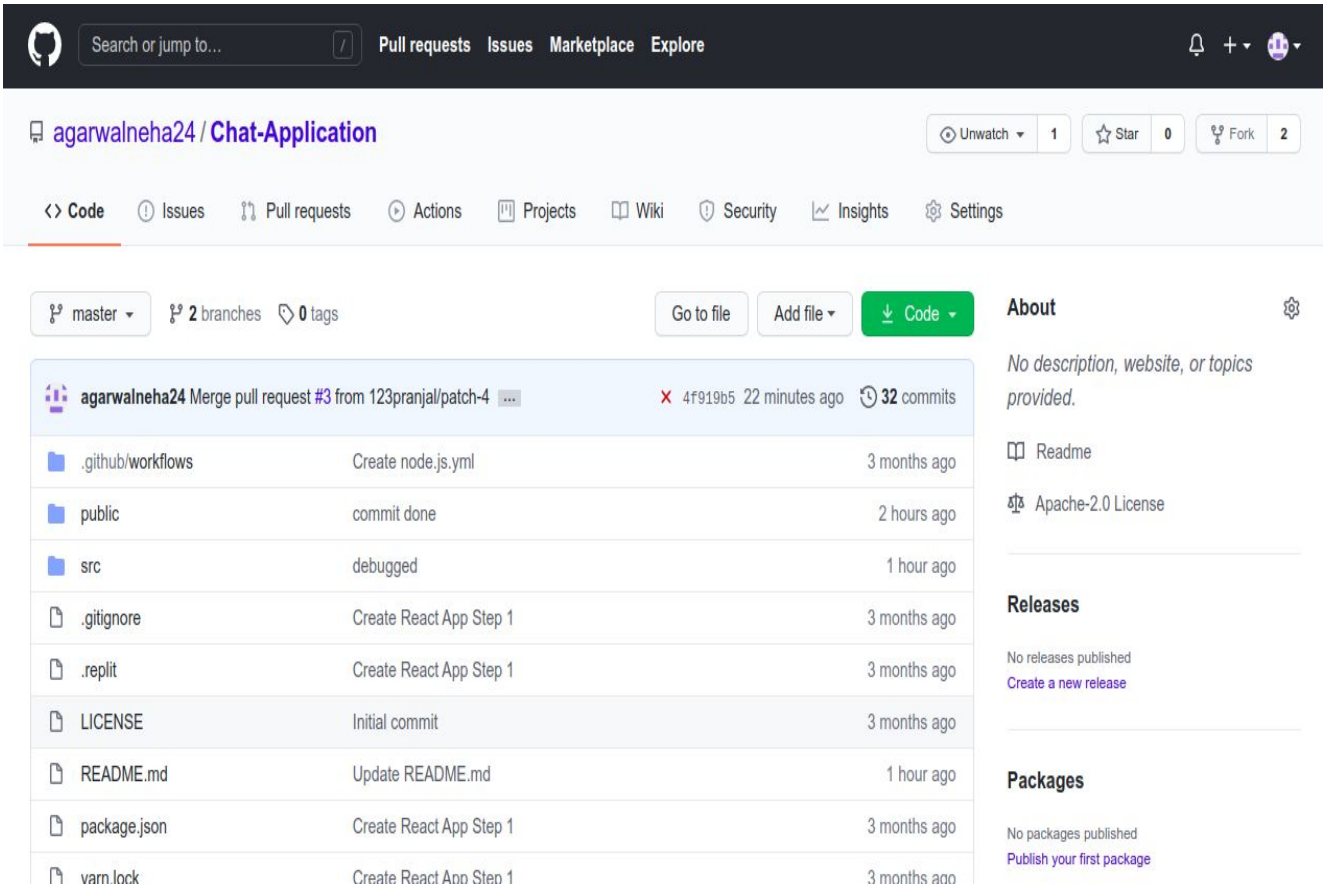
VERSION CONTROL SYSTEM: GIT

The term Version control refers to a system that records changes to a file or set of files over time called the ‘versions’. In other words, these versions help us in tracking the changes in your codes/project and if necessary, undo those changes as well. This feature of being able to compare, differentiate and revert changes between two versions of a particular project becomes really helpful when working on a larger project. Larger projects mean more people working on the same code which increases the chances of conflicts. Using version control we can easily prevent these conflicts.

Git is a version control system which tracks the changes when working with computer codes while GitHub is a Web-based Git version control repository hosting service. It provides all of the distributed version control and source code management (SCM) functionalities of Git while topping it with a few of its own features.

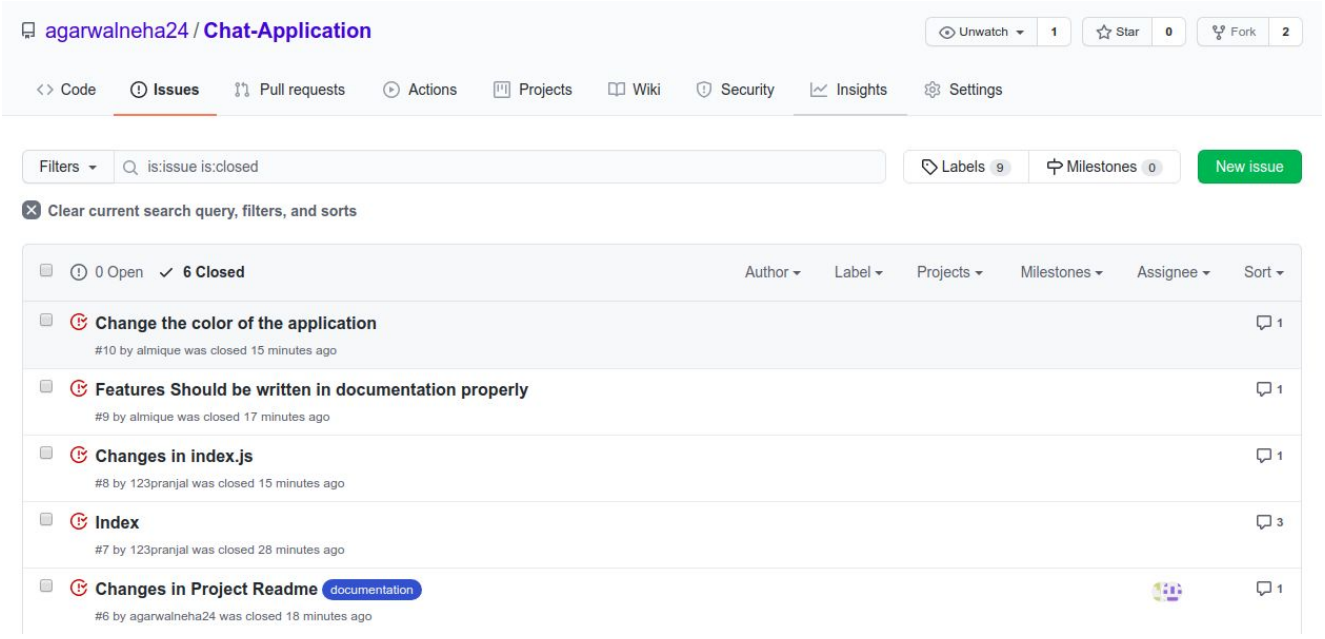
Github Repository link:

[Click here]

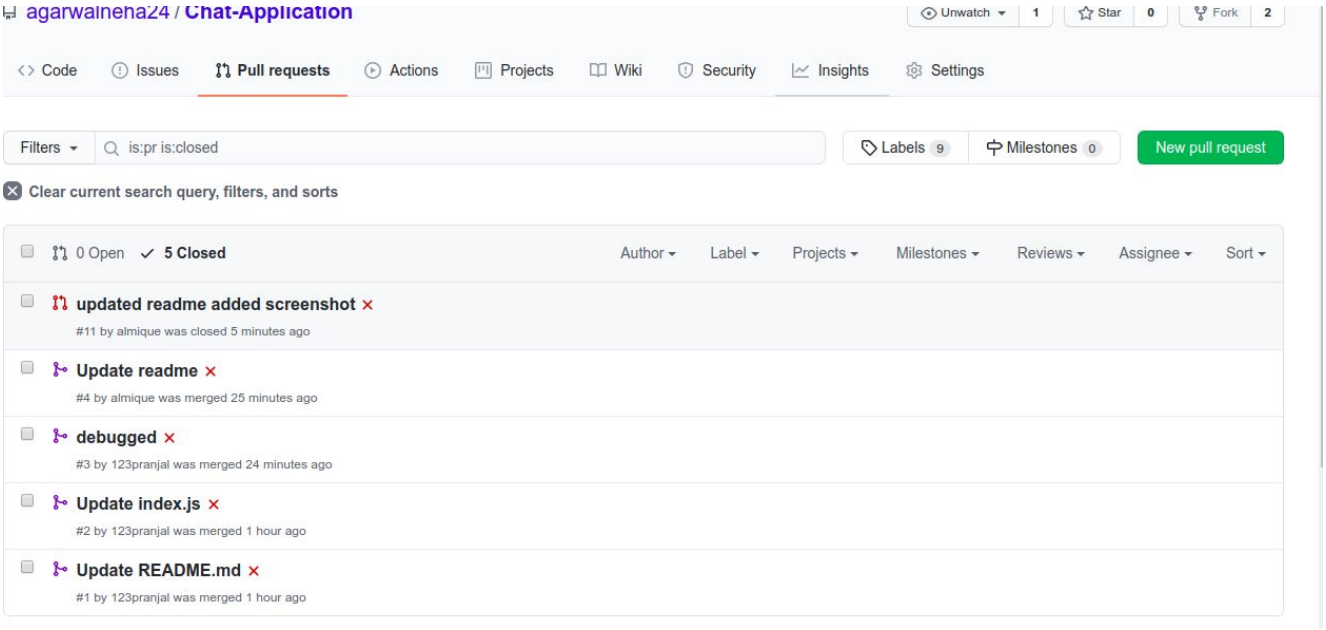


FEATURES OF VERSION CONTROL SYSTEM USED:

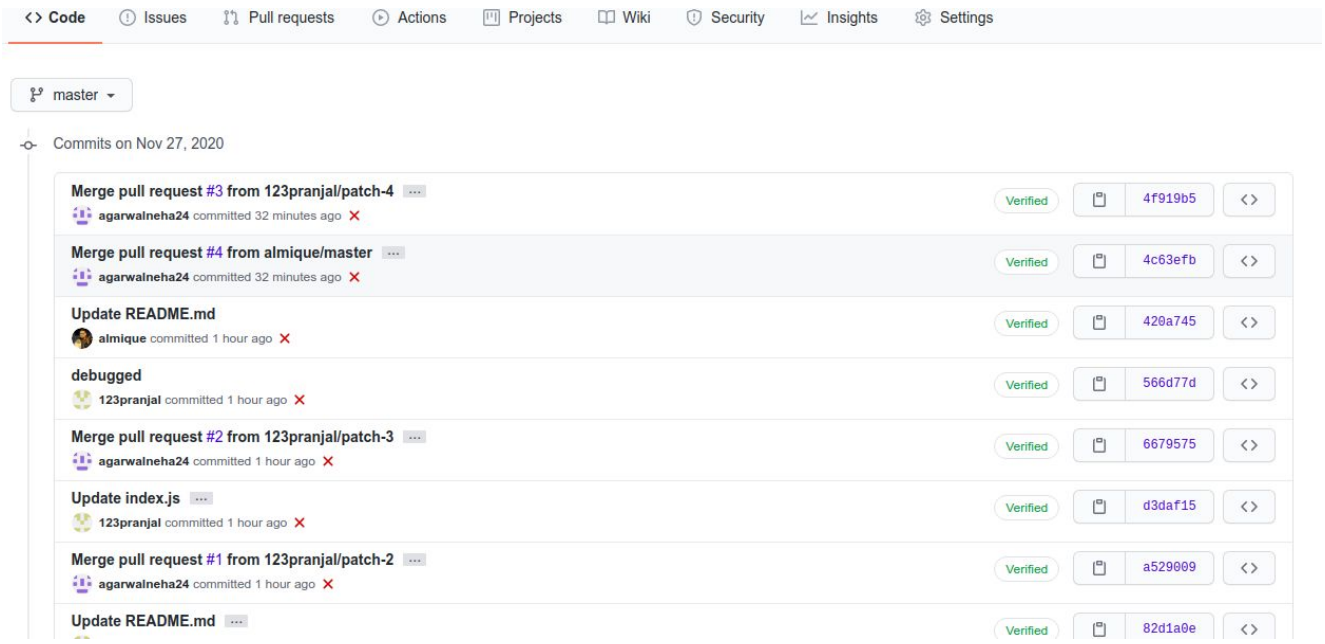
Issues are a great way to keep track of tasks, enhancements, and bugs for your projects. They're kind of like email—except they can be shared and discussed with the rest of your team. Most software projects have a bug tracker of some kind. **GitHub's** tracker is called **Issues** and has its own section in every repository.



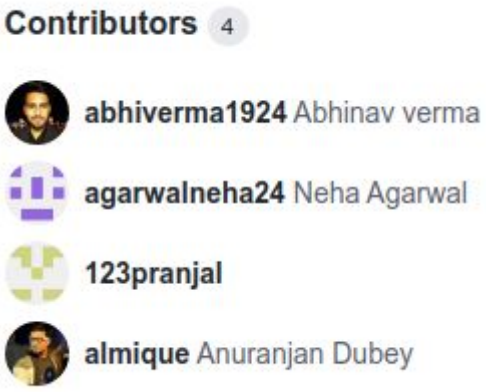
Pull requests are **important** because they help ensure that quality reviewed code is merged into **GitHub** repositories. Without PRs, messy and confusing code can easily run rampant in a codebase.



COMMIT HISTORY



CONTRIBUTORS



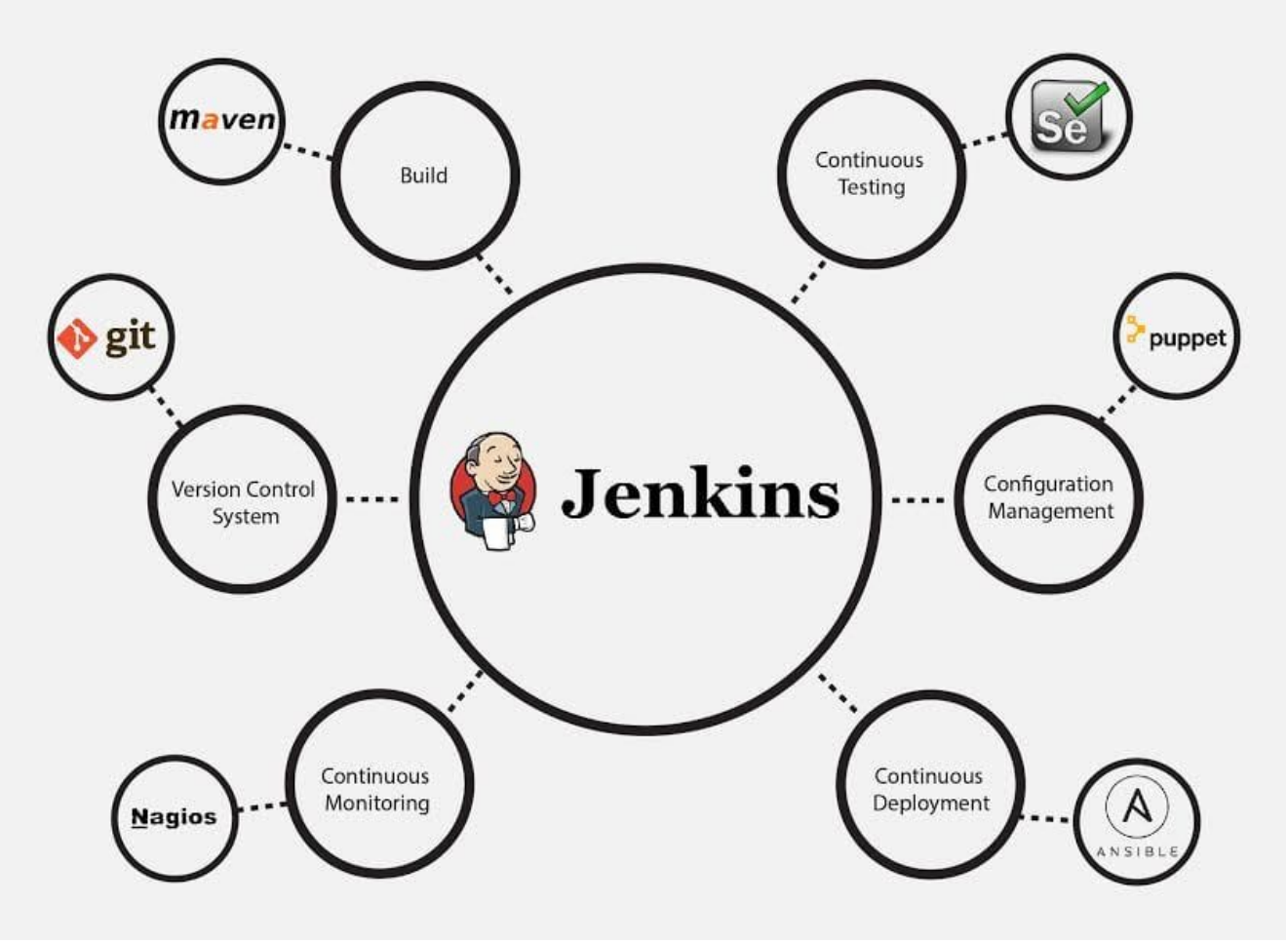
CI/CD PIPELINE- JENKINS

In software engineering, CI/CD or CICD generally refers to the combined practices of continuous integration and either continuous delivery or continuous deployment. CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications.

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes. Jenkins is used to build and test our software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows us to continuously deliver our software by integrating with a large number of testing and deployment technologies.

With Jenkins, organizations can accelerate the software development process through automation. Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis, and much more.

Jenkins achieves Continuous Integration with the help of plugins. Plugins allow the integration of Various DevOps stages. If you want to integrate a particular tool, you need to install the plugins for that tool. For example Git, Maven 2 project, Amazon EC2, HTML publisher etc.

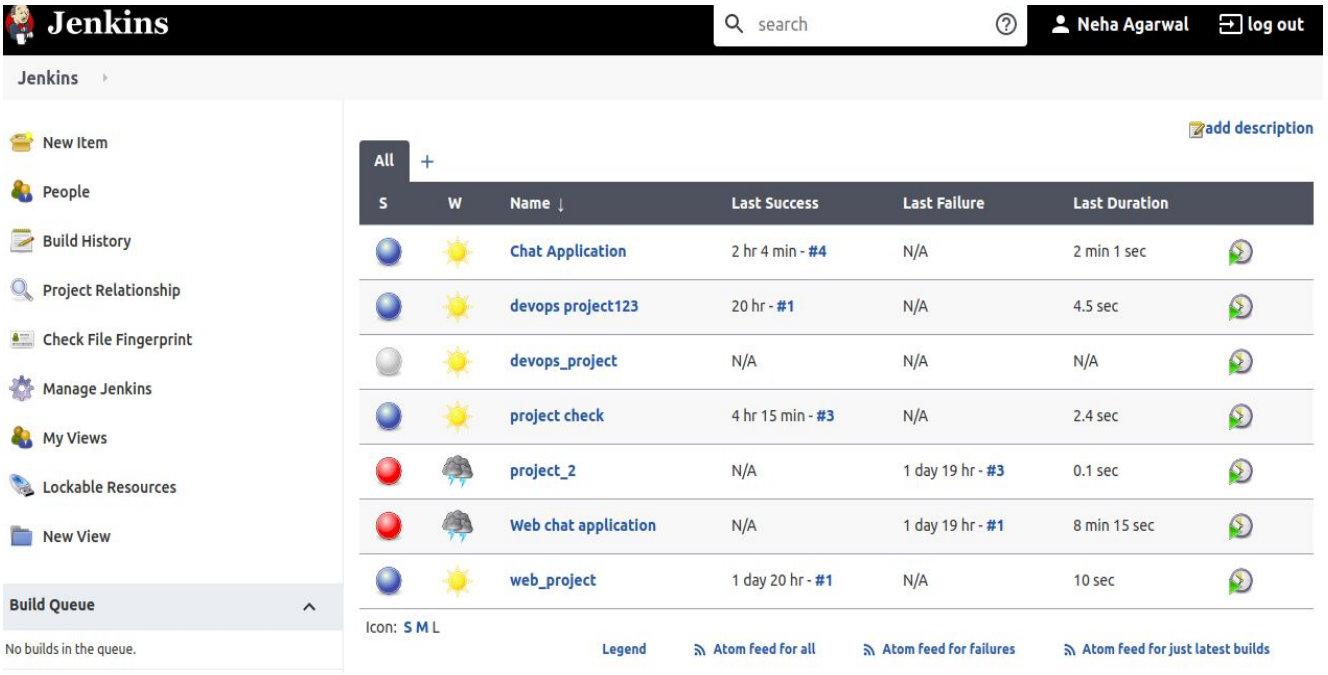


CHAT APPLICATION JENKINS PIPELINE DESCRIPTION

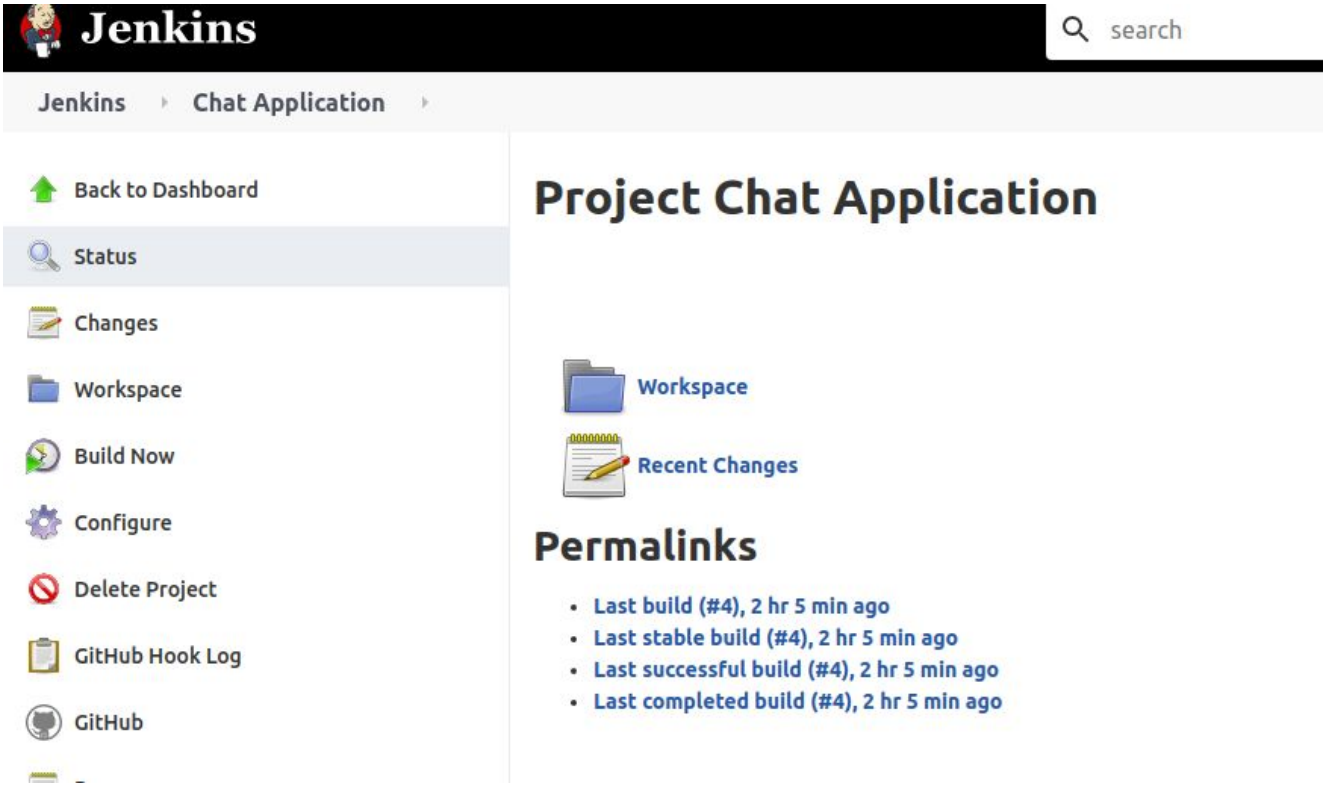
Each time we will commit a change to our Github repository, automated deployment on Heroku will be done through the Jenkins CI/CD pipeline thus making it easier to integrate and tackle errors. On building the project, Jenkins will automatically clone the recent code, test, and deploy it on Heroku.

Clone the code → Test the project → Deploy on Heroku

JENKINS DASHBOARD

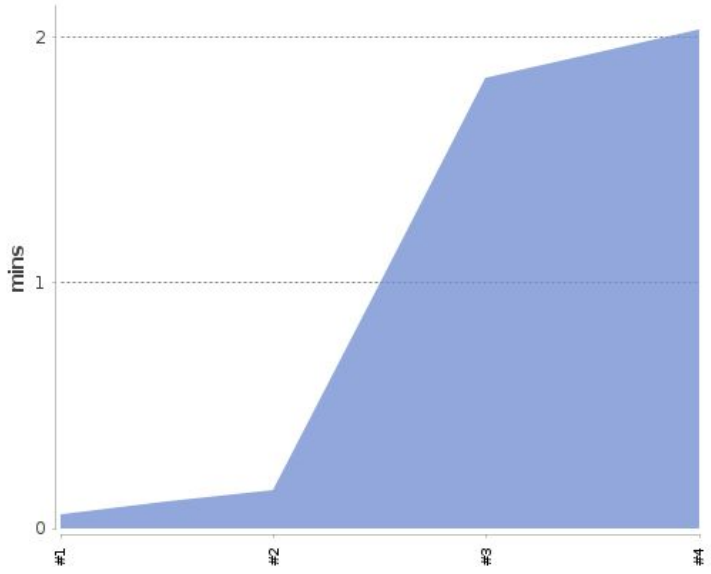


PROJECT CHAT APPLICATION BUILD HISTORY



Build Time Trend

Build	Duration
#4	2 min 1 sec
#3	1 min 49 sec
#2	9.3 sec
#1	3.2 sec



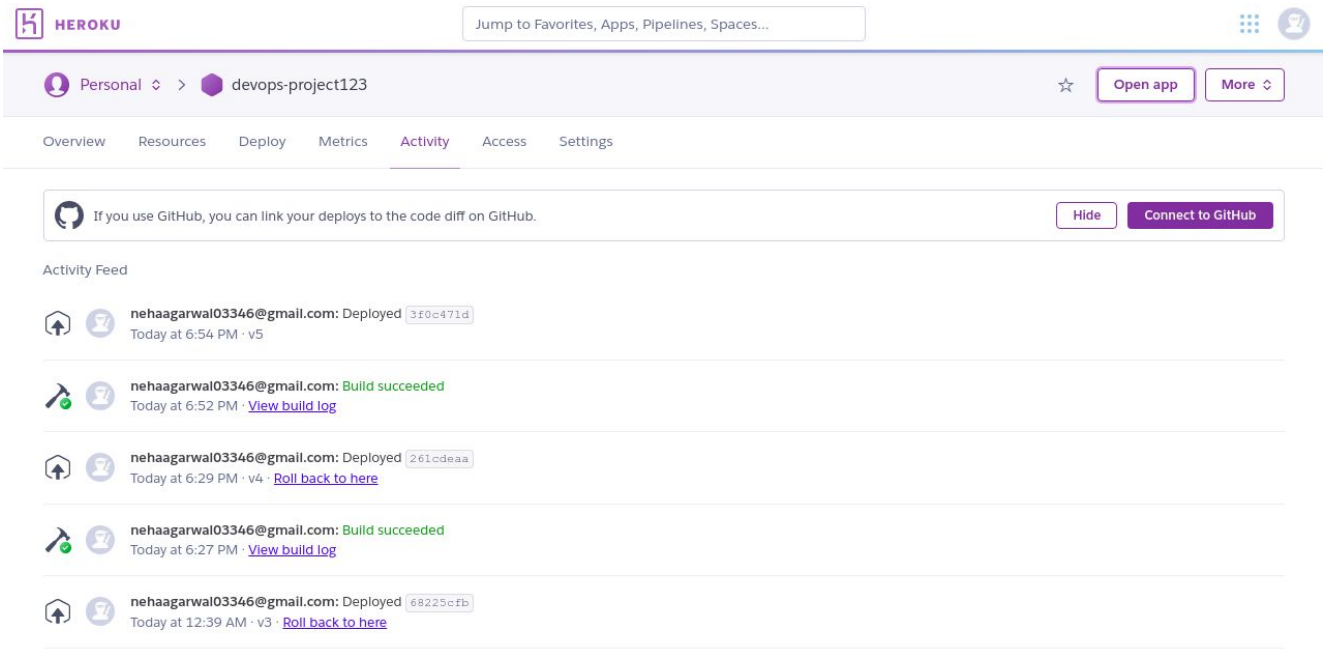
CONSOLE OUTPUT

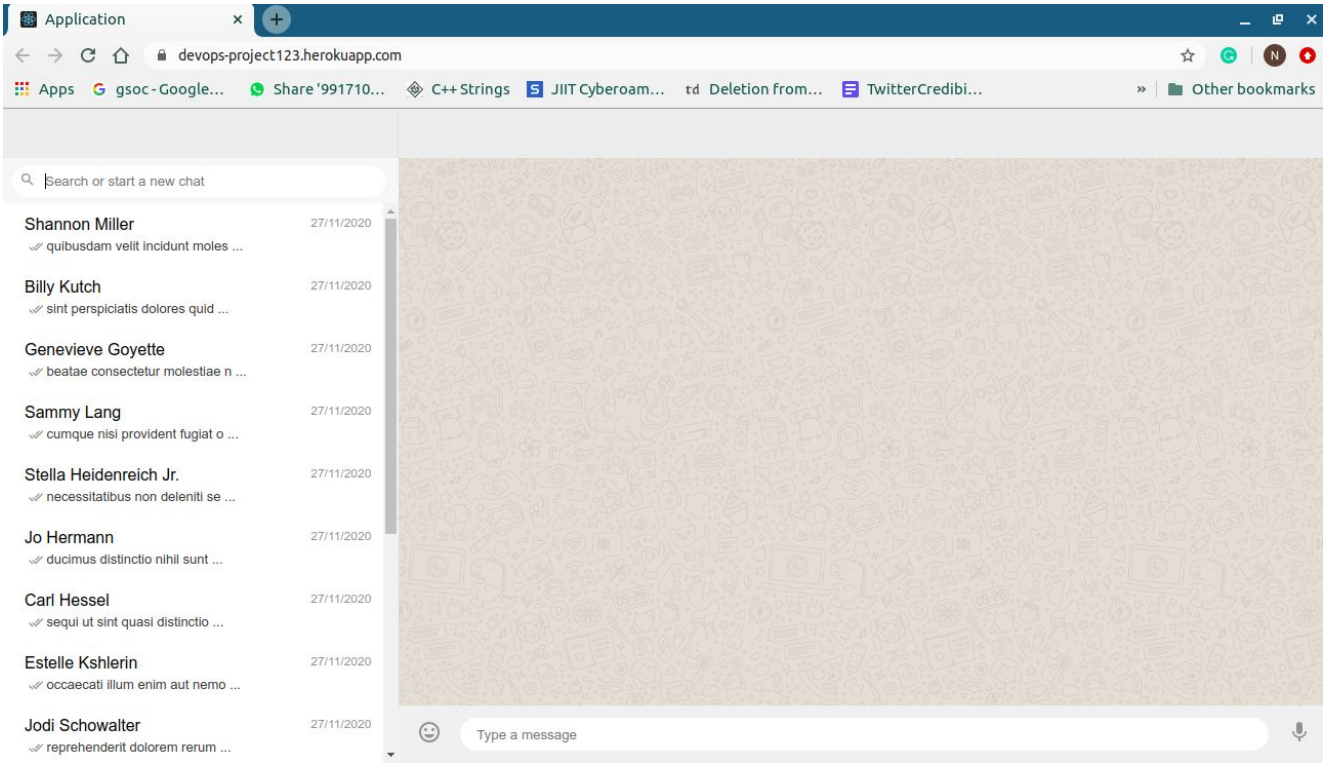
Console Output

```
Started by user Neha Agarwal
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Chat Application
The recommended git tool is: NONE
No credentials specified
using credential 127bc898-9ef7-437c-a9e0-99407b11178e
Cloning the remote Git repository
Cloning repository git@heroku.com:devops-project123.git
 > git init /var/lib/jenkins/workspace/Chat Application # timeout=10
Fetching upstream changes from git@heroku.com:devops-project123.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
using GIT_SSH to set credentials
 > git fetch --tags --force --progress -- git@heroku.com:devops-project123.git +refs/heads/*:refs/remotes/abcd/* #
timeout=10
 > git config remote.abcd.url git@heroku.com:devops-project123.git # timeout=10
 > git config --add remote.abcd.fetch +refs/heads/*:refs/remotes/abcd/* # timeout=10
Avoid second fetch
 > git config remote.origin.url https://github.com/agarwalneha24/Chat-Application # timeout=10
Fetching upstream changes from https://github.com/agarwalneha24/Chat-Application
 > git fetch --tags --force --progress -- https://github.com/agarwalneha24/Chat-Application
+refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository abcd/master
Seen branch in repository origin/main
Seen branch in repository origin/master
Seen 3 remote branches
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 3f0c471dbe67b1efd62da292f1fbdb4b4b25b978 # timeout=10
Commit message: "commit done"
 > git rev-list --no-walk 261cdeaac12282a04af14f97d0f024a3c3067db4 # timeout=10
The recommended git tool is: NONE
No credentials specified
using credential 127bc898-9ef7-437c-a9e0-99407b11178e
Pushing HEAD to branch master at repo abcd
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
using GIT_SSH to set credentials
 > git push git@heroku.com:devops-project123.git HEAD:master -f # timeout=10
Finished: SUCCESS
```

HEROKU-DEPLOYMENT

Heroku is a cloud platform as a service supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go.



CHAT APPLICATION VIEW

Link- [Click here]