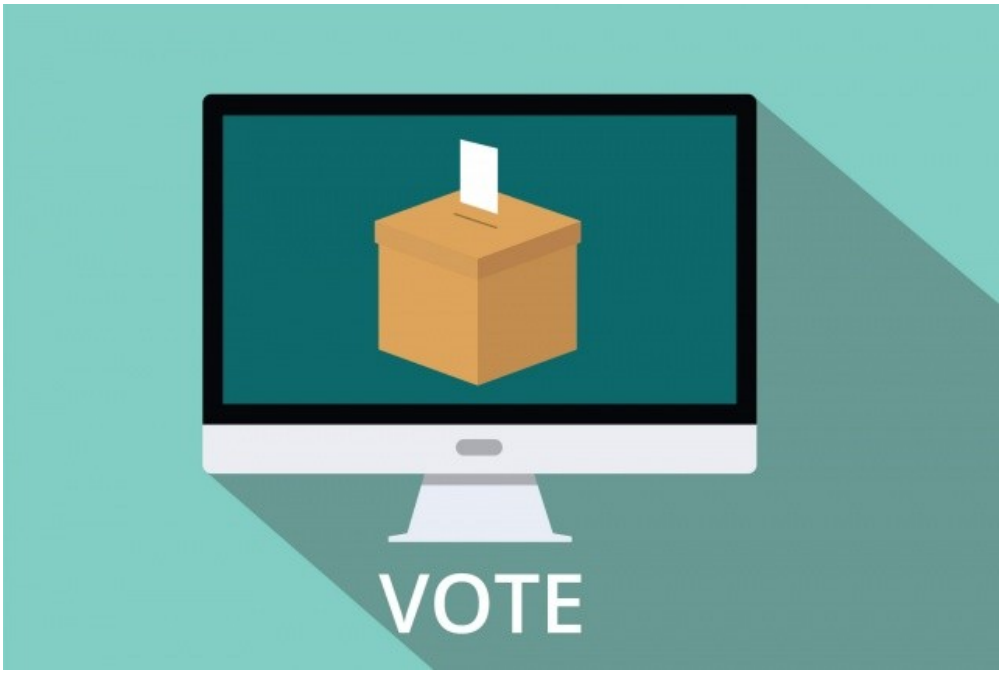
Project Report
on
E-Voting System for SAC Elections



UNDER GUIDANCE OF

Prof. B.Thangaraju
Rajula Vineet Reddy IMT2014045

SUBMITTED BY
Akshada Kamle MT2018006
Amisha Jain MT2018011

INDEX

1. ABSTRACT

The application intends to provide an automated e-voting system for college SAC elections. The present process of SAC election is quite tedious and time consuming. Thus, evoting system makes it easy for the admin as well as the candidates to conduct the election procedure. The students can directly vote for their favorite candidates using the online portal without using the pen and paper.

2. INTRODUCTION

2.1 Why Devops?

DevOps is a combination of software development and operations—and as its name suggests, it's a melding of these two disciplines in order to emphasize communication, collaboration, and cohesion between the traditionally separate developer and IT operations teams. The principles of DevOps are Iterative, Incremental, Collaboration, Quantification, automation, integration and Holistic . Some of its benefits can be viewed in the below figure.

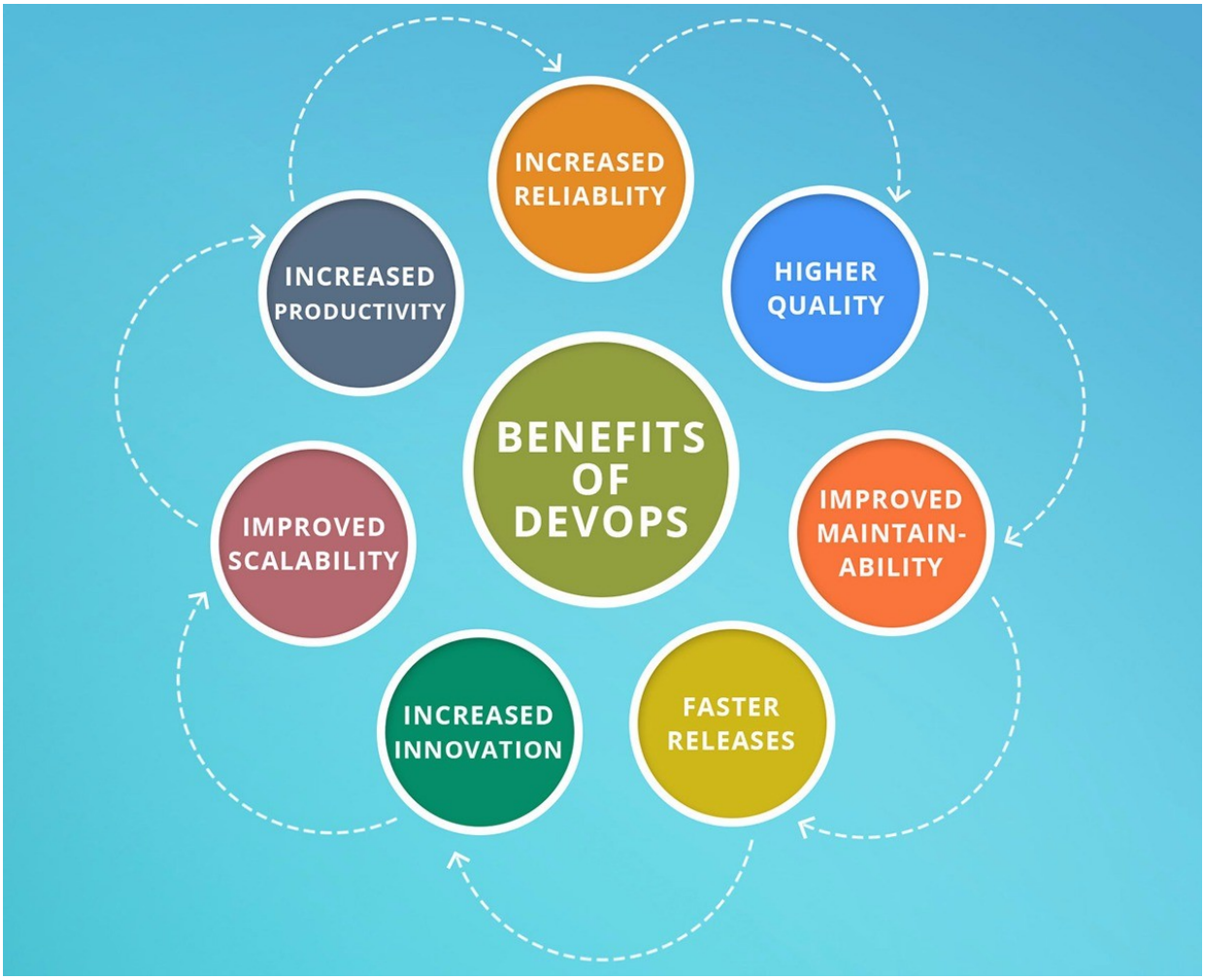


Figure 1. Benefits of DevOps

2.2 About the Application

Our application is an online website for hosting SAC elections in the college. It mainly has three users-

- Admin- the committee that conducts the elections
- Nominees- students who nominate themselves for the post
- Students-who use the application for voting

The application has the following features:

A nominee can-

- register
- add/edit photo and vote appeal
- view details of registration

Admin can-

- approve the nominee registration
- view the details of voters
- update important guidelines on the website

Student can-

- view nominee details
- vote for the nominees

3. Software Development Life Cycle

3.1 Scope of the Project

This project eases the process of SAC election by providing various features. It allows the admin to conduct the elections in less time and lesser efforts. It is developed by using the MVC architecture which makes the code more robust and easy to modify. Test cases are written in such a way that it covers most of the integration testing of the app. Like it checks the authenticity and connection before taking any entries.

3.2 Project Architecture

We've used following Dev-Ops tools for the design-

- Netbeans 8.2 as IDE for development
- Junit for basic testing
- Jenkins in Docker for Continuous Integration
- GitHub for SCM (Source Code Management)
- Ant as Build tool to generate artifact
- Rundeck for Continuous Deployment
- ELK stack for Continuous Monitoring

We only focus to design the code. As we pushed our new code to git, Jenkins automatically captures it with no input or just a single click and rest all process is automatic and proper reports are generated for the same.

3.2.1 Workflow

The application workflow can be seen in the screenshots below-

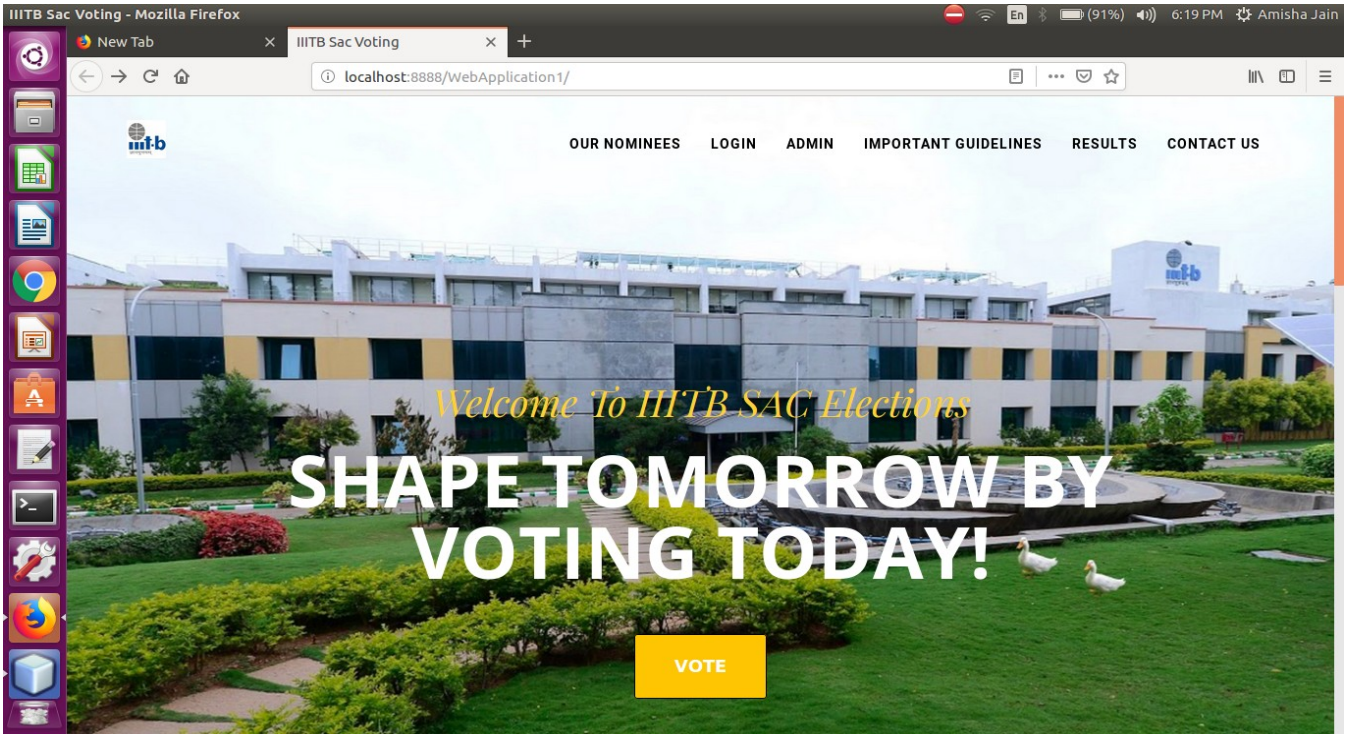


Figure 2. Home Page

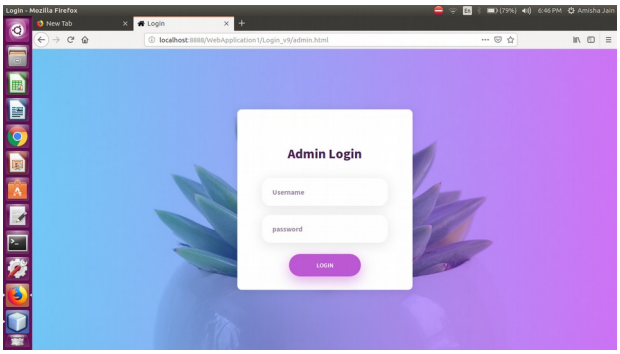


Figure 3. Admin Login Page

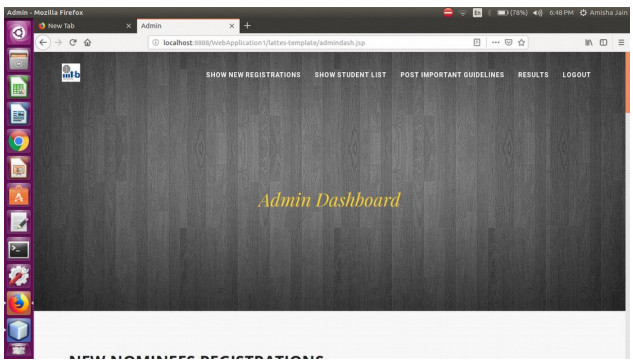


Figure 4. Admin Dashboard

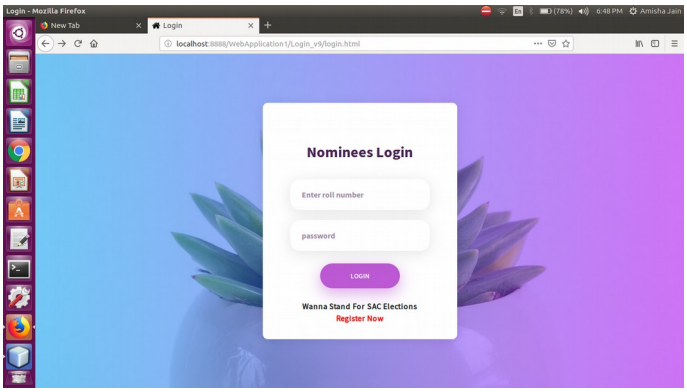


Figure 5. Nominee Login Page

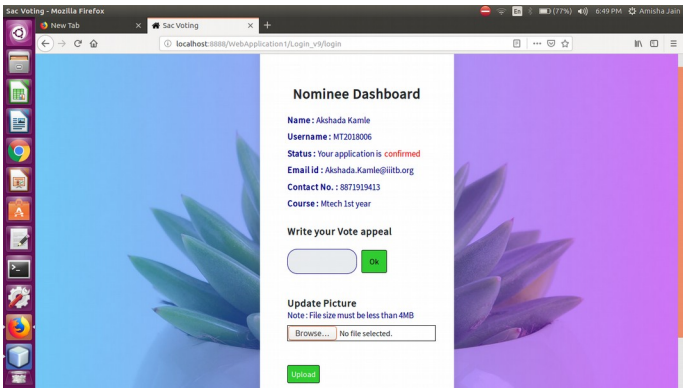


Figure 6. Nominee Dashboard

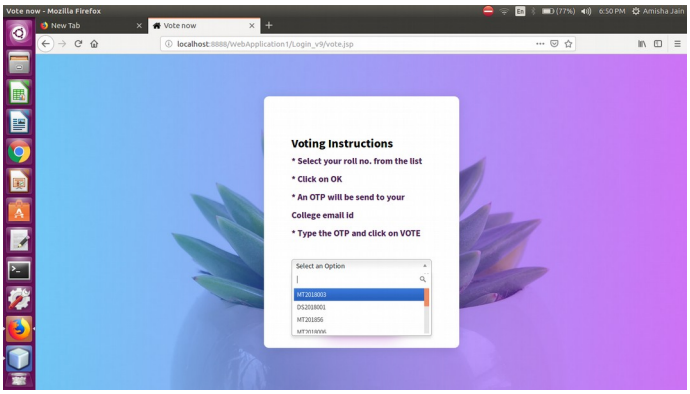


Figure 7. Voting Page

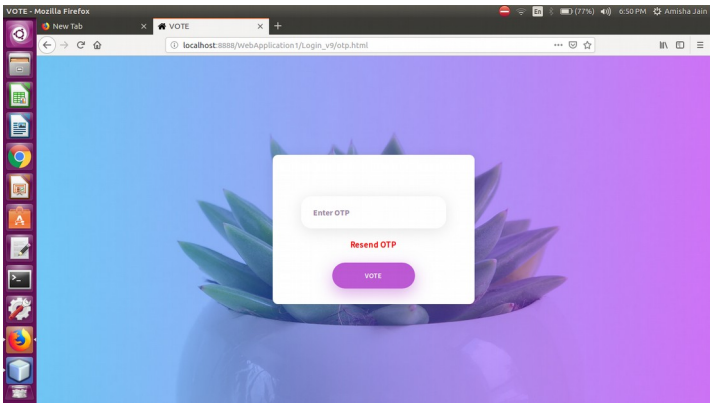


Figure 8. Getting an OTP

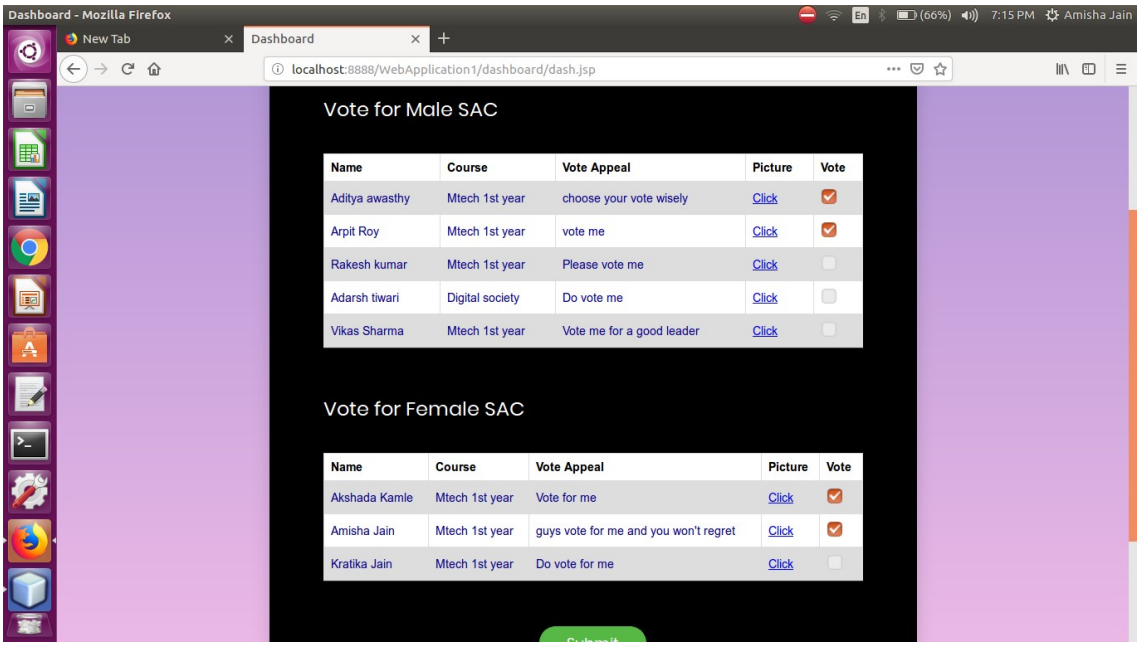


Figure 9. Voting Process

3.3 SCM

For Source Code Management, we are using Git repository. Git provides the following features-

- Feature Branch Workflow
- Distributed Development
- Pull Requests
- Community
- Faster Release Cycle

We have used in built VCS (Version Control System(git)) of Netbeans IDE as it tracks the file changes that is modified or deleted in real time. We can also use its GUI features to add, commit & push operations.

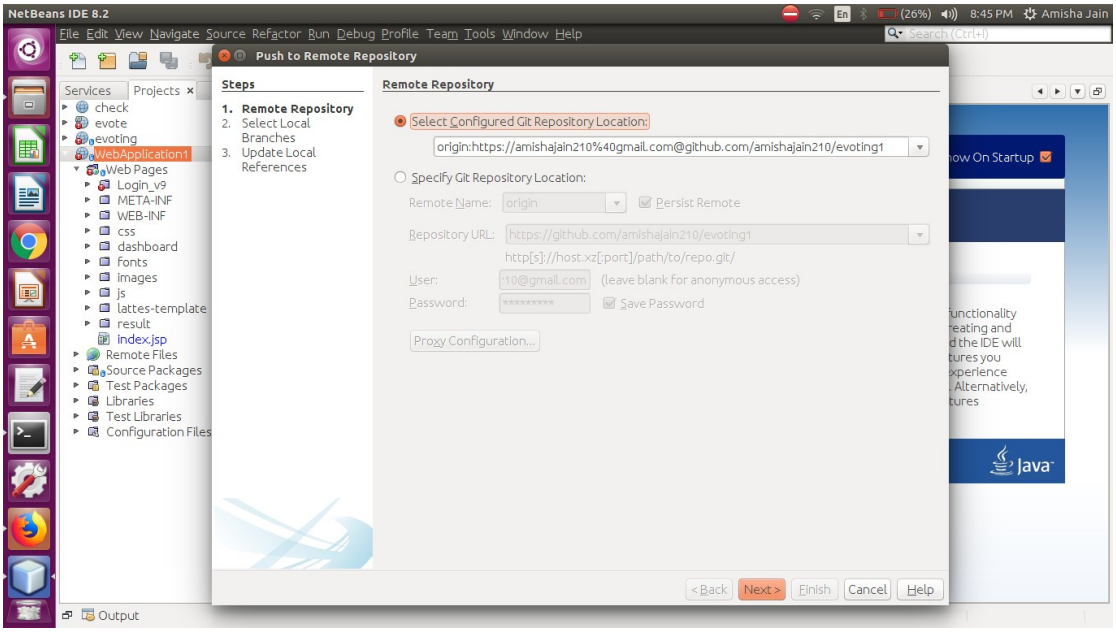


Figure 10. SCM from Netbeans

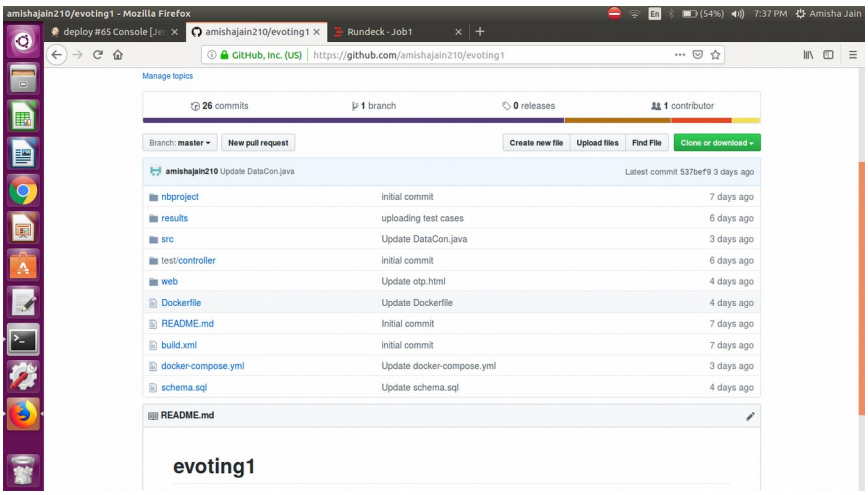


Figure 11. Git Hub Repository

The link of our Git repository is- https://github.com/amishajain210/evoting1

3.4 Build

We have configured Jenkins for the build activity. Ant tool is used in order to build the required artifact. Ant is the most complete Java build and deployment tool available. It is platform neutral and can handle platform specific properties such as file separators.

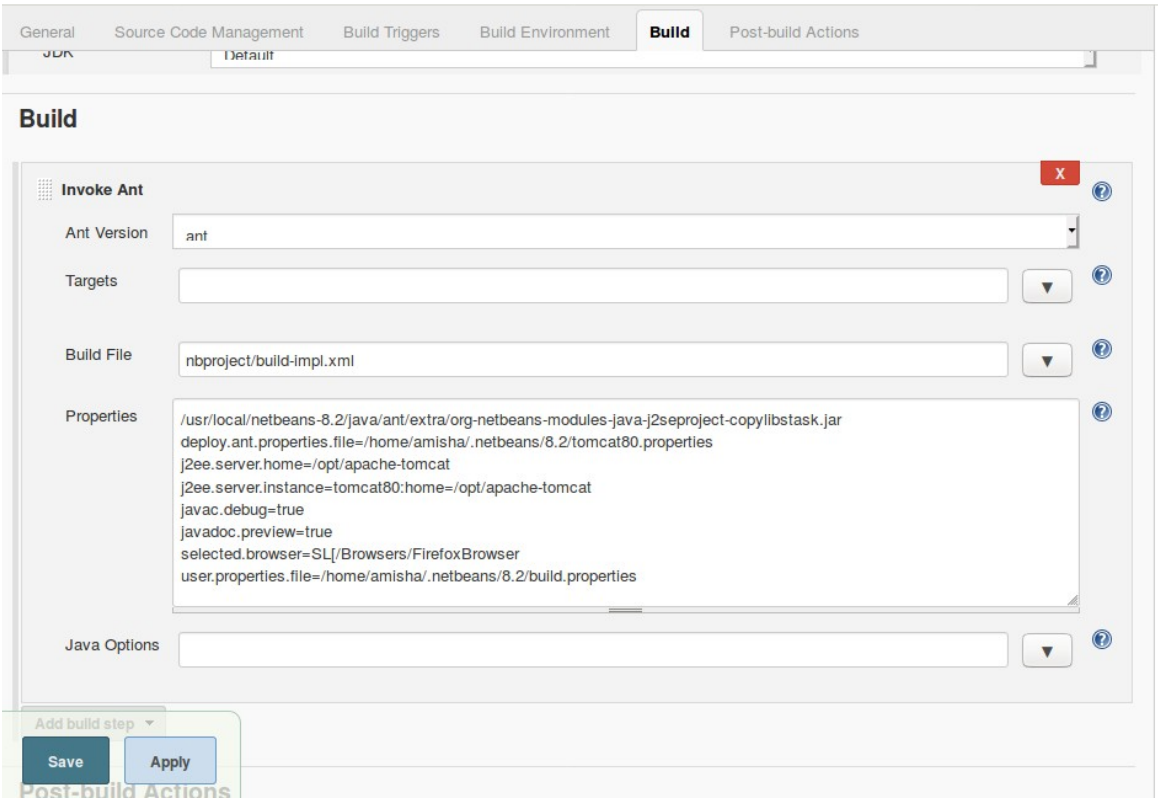


Figure 12. Build Configuration

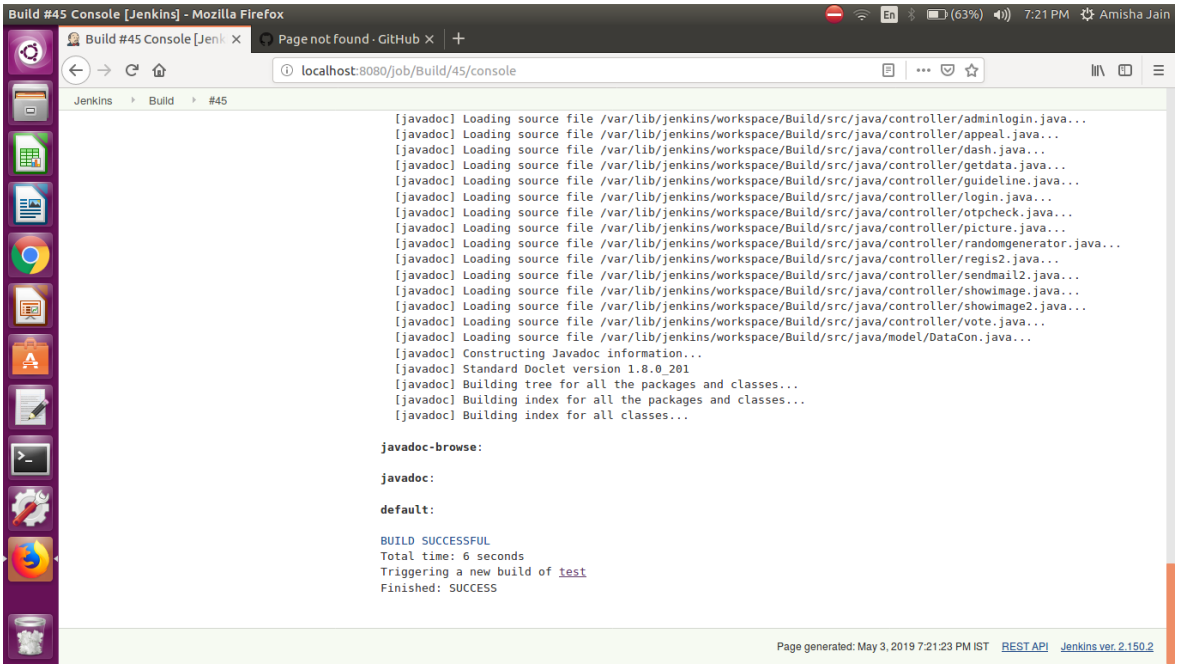


Figure 13. Build Process

3.5 Test

We have applied Junit testing on our application. JUnit is an open source framework, which is used for writing and running tests. Following is the test configuration-

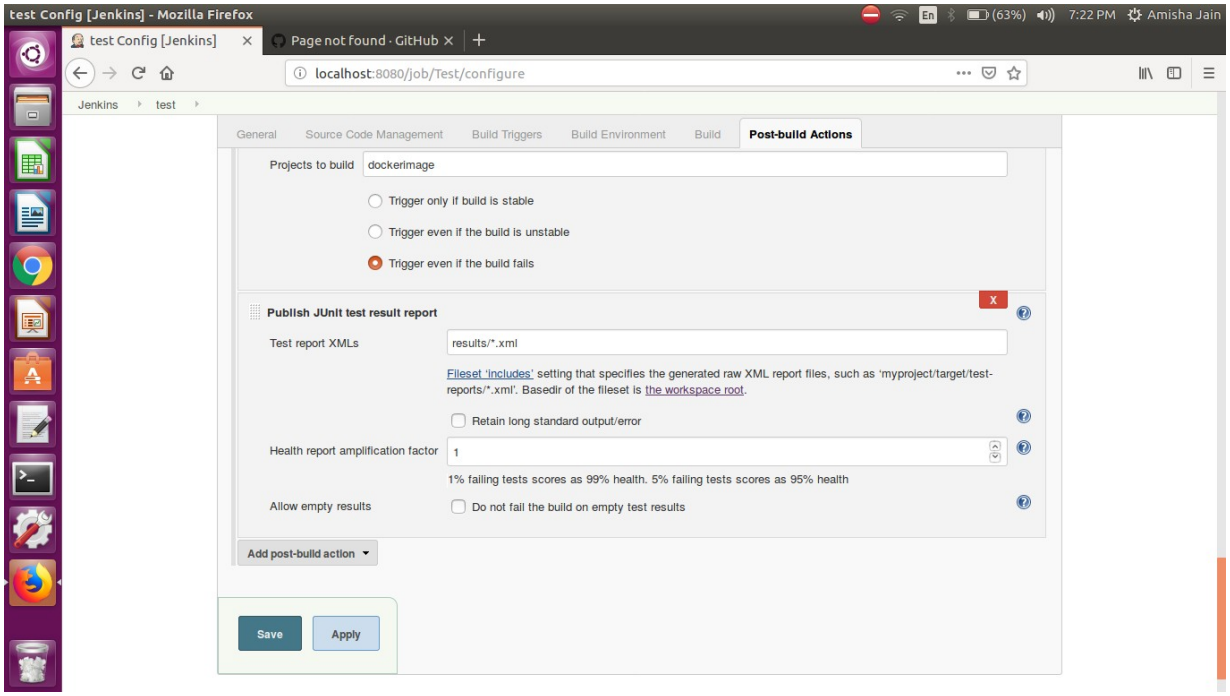


Figure 14. Test Configuration

3.6 Artifact

We are creating the docker image of our project using Jenkins and then pushing it to the Docker hub using the same. Following figures show configuration settings and the Docker image creation -

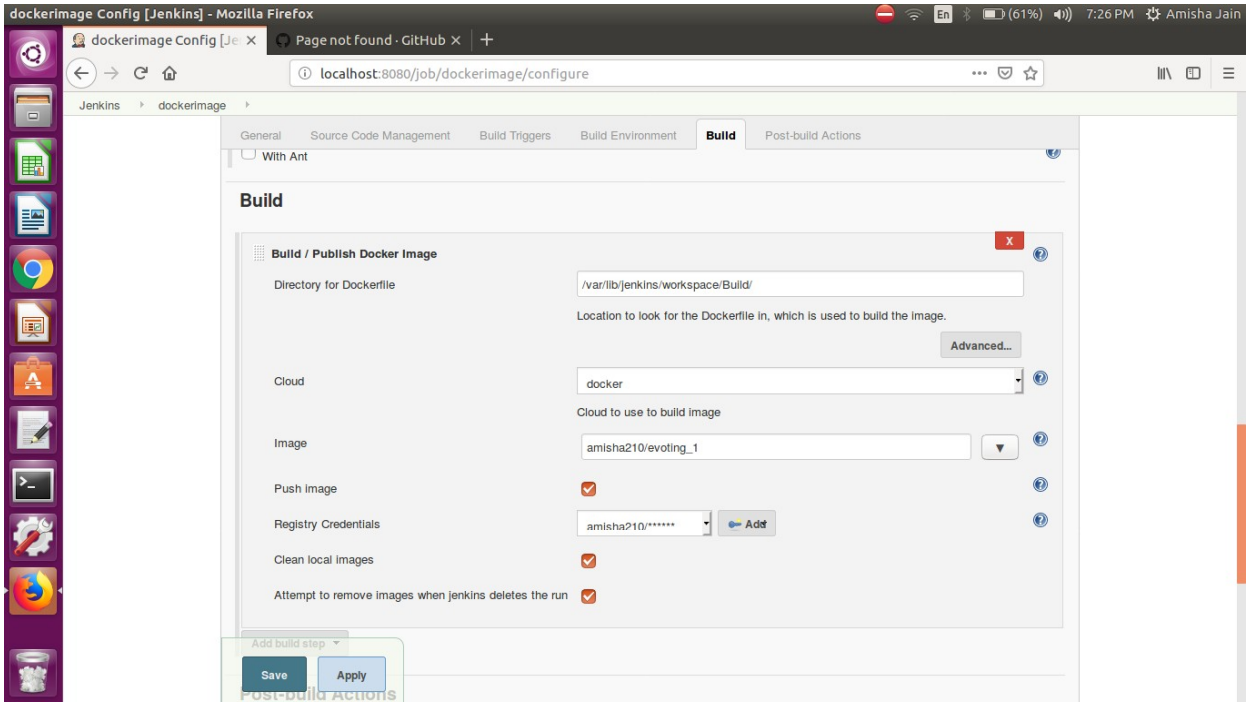


Figure 15. Docker Image build configuration

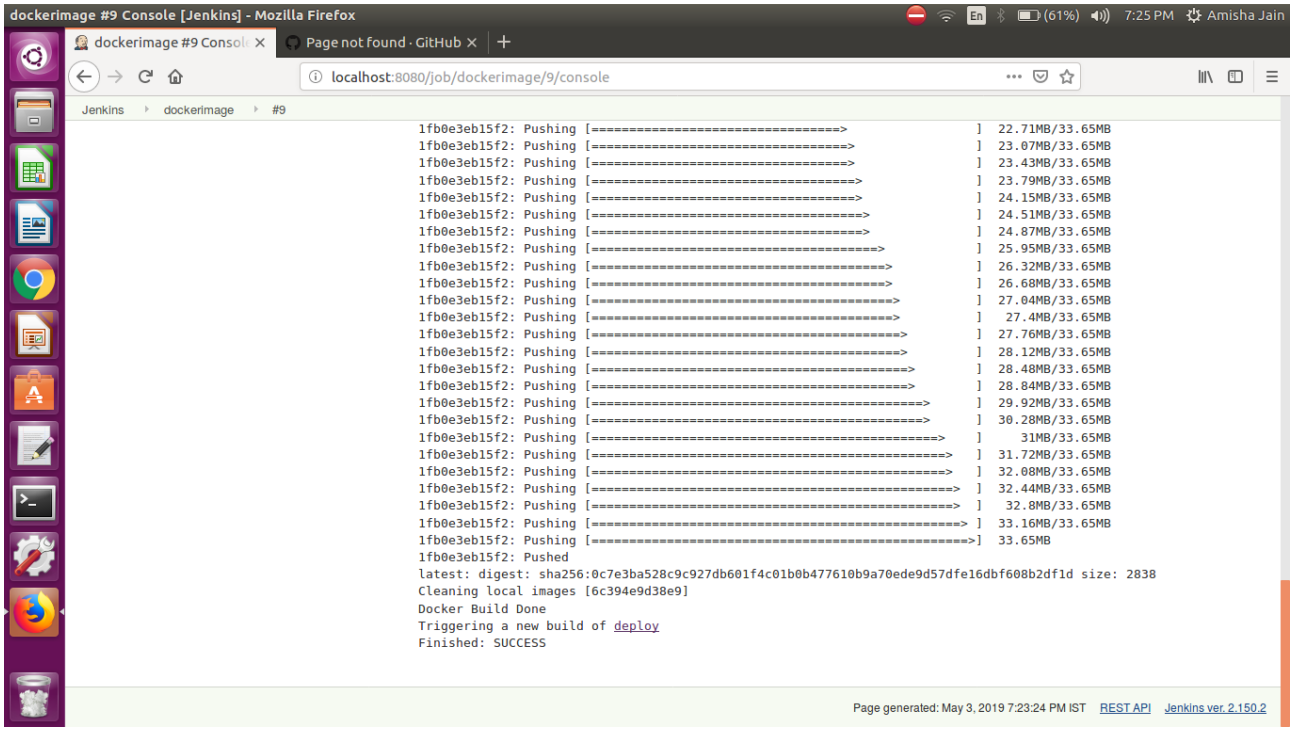


Figure 16. Docker Image creation

3.7 Deploy

We have used Rundeck to deploy our application. Following is the configuration-

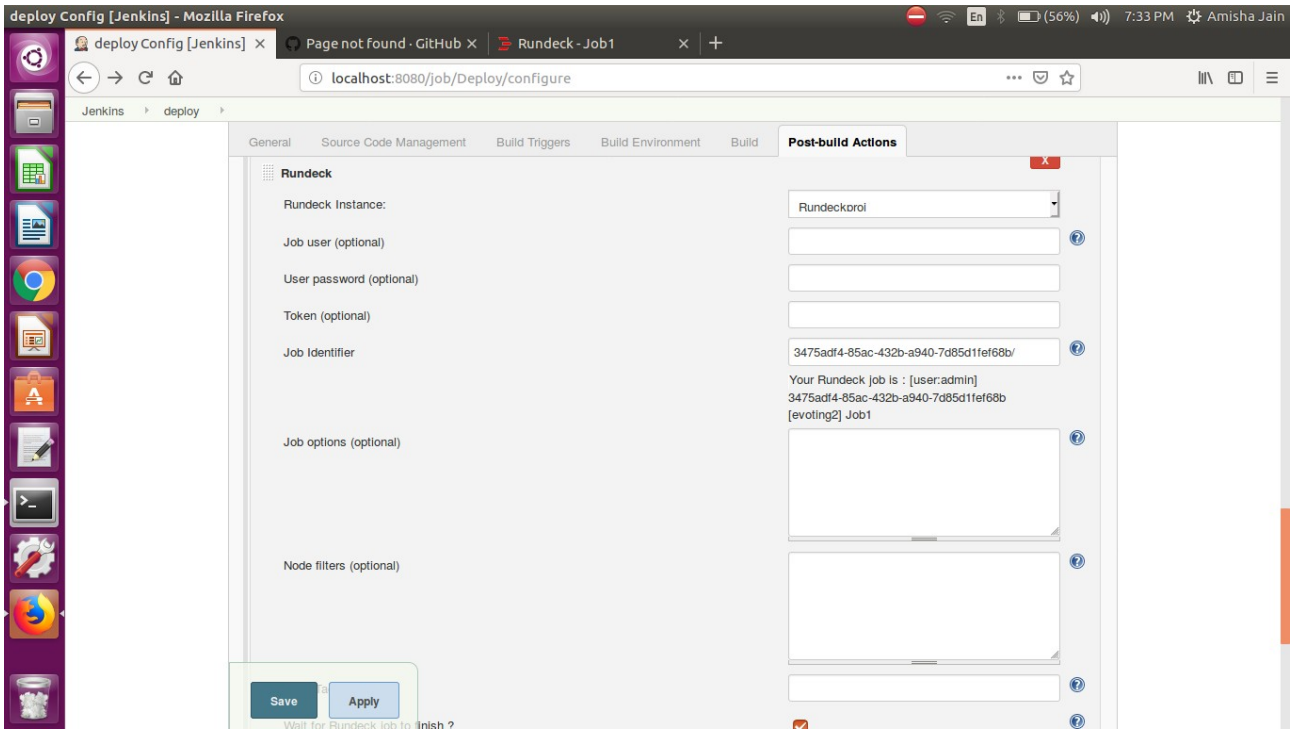


Figure 17. Deploy Configuration

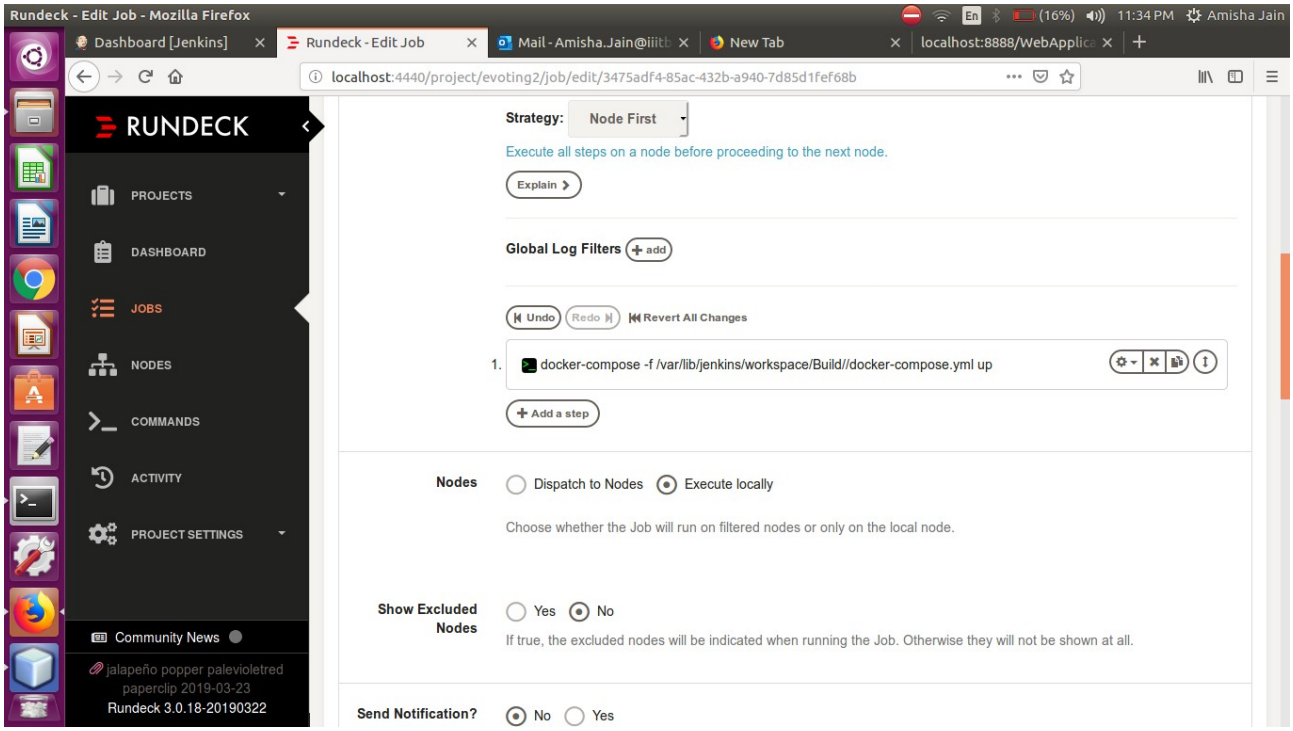


Figure 18. Rundeck Configuration

3.8 Monitor

For continuous monitoring, we are using ELK(Elastic Search, Logstash, Kibana) stack. Elasticsearch is a NoSQL database that is based on the Lucene search engine. Logstash is a log pipeline tool that accepts inputs from various sources, executes different transformations, and exports the data to various targets. Kibana is a visualization layer that works on top of Elasticsearch.

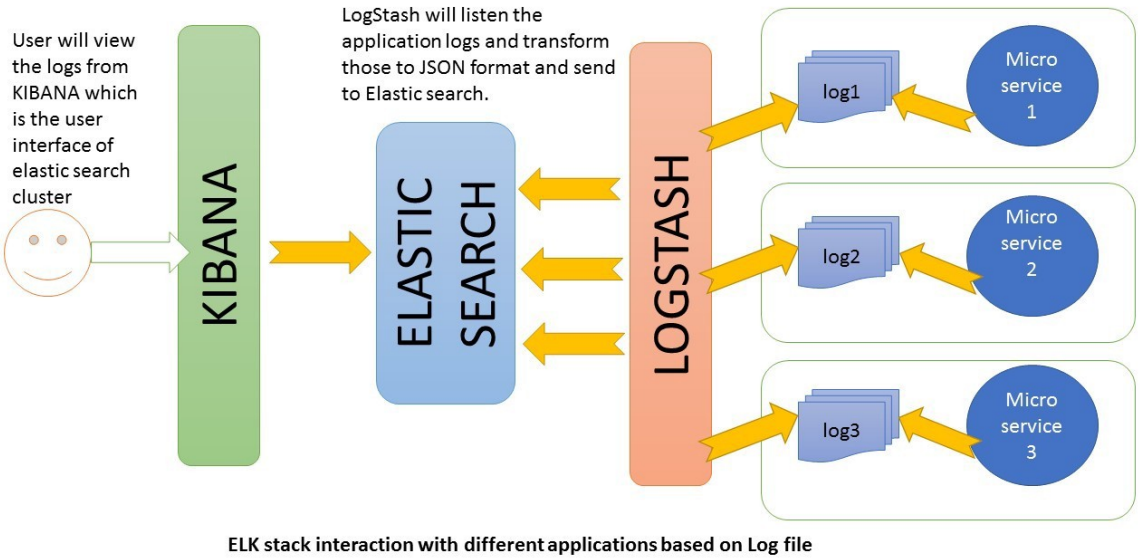


Figure 19. ELK stack

We have visualised the number of votes for each candidate. Below are the various visualisations-

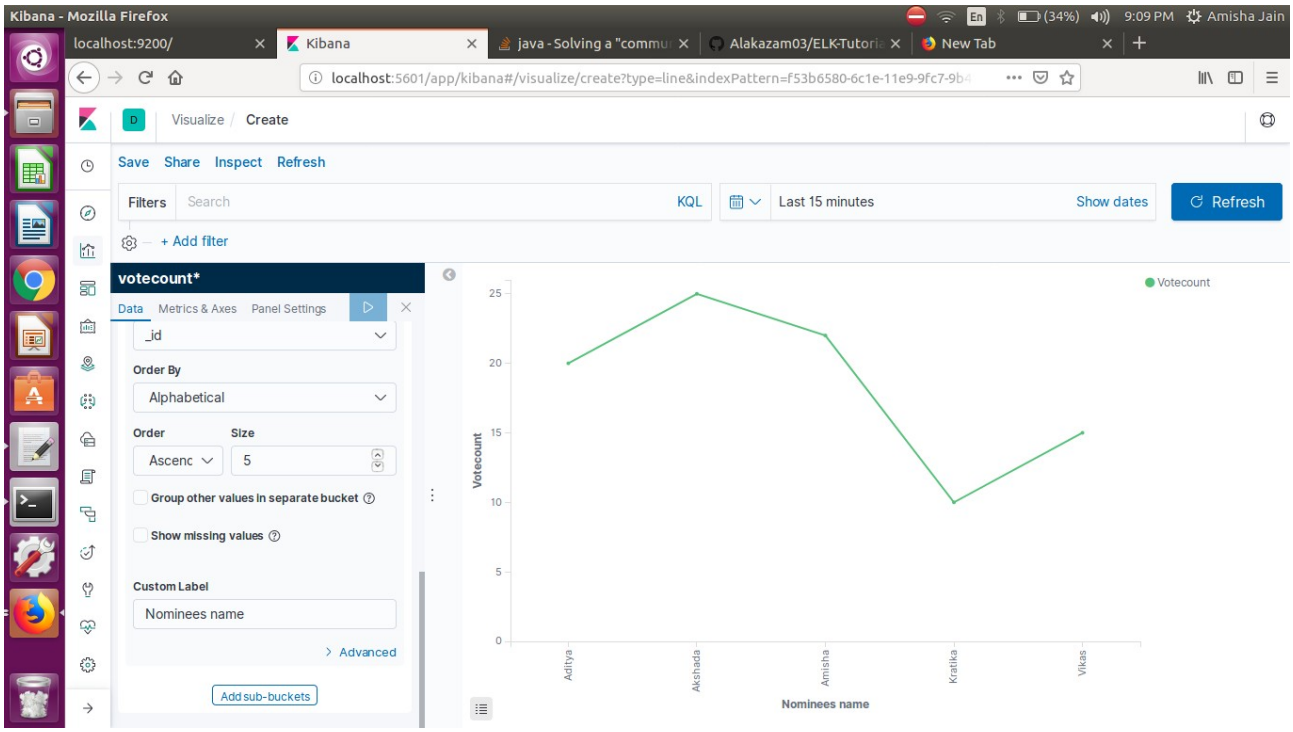


Figure 20. Line graph

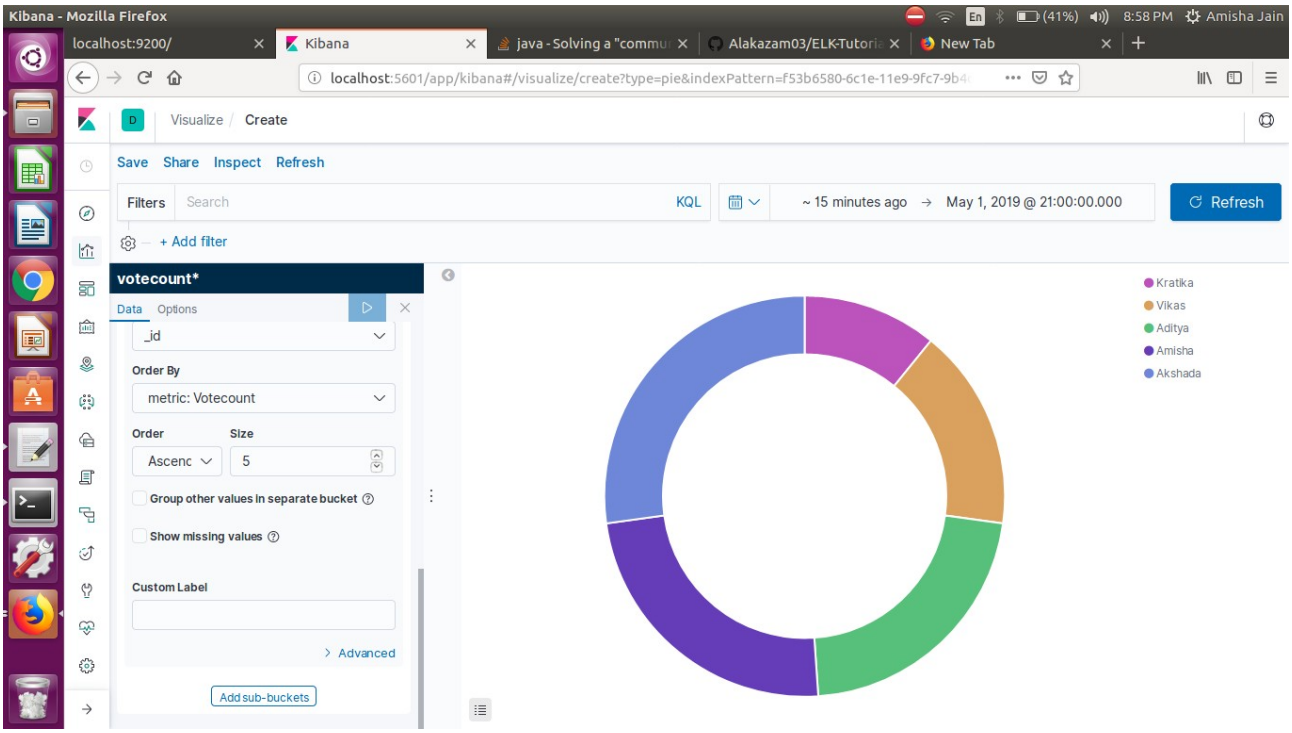


Figure 21. Pie Chart

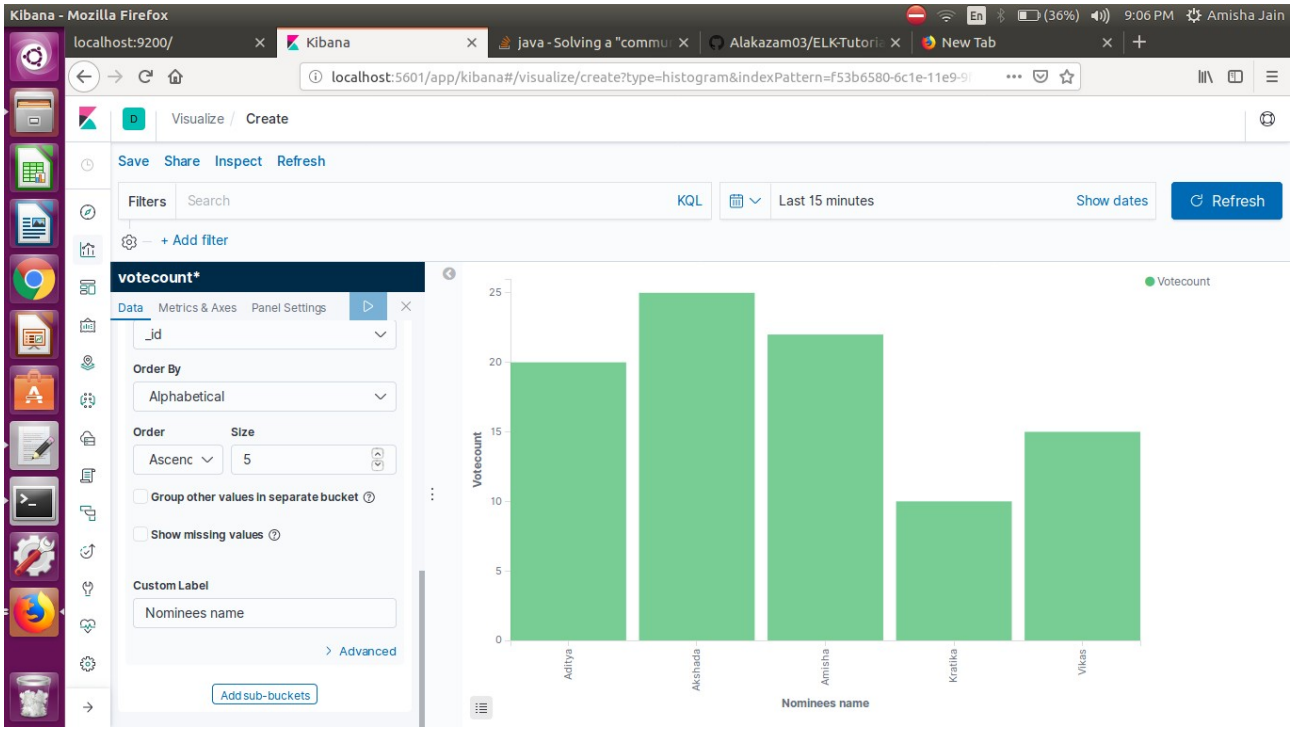


Figure 22. Bar Graph

3.9 Continuous Integration tool- Jenkins

Jenkins is an open source Continuous Integration server capable of orchestrating a chain of actions that help to achieve the Continuous Integration process in an automated fashion. The code is built and tested as soon as Developer commits code. Jenkin will build and test code many times during the day. We have used Jenkins as it provides the following features-

- If the build is successful, then Jenkins will deploy the source into the test server and notifies the deployment team.
- If the build fails, then Jenkins will notify the errors to the developer team.
- The code is built immediately after any of the Developer commits.
- Since the code is built after each commit of a single developer, it's easy to detect whose code caused the built to fail
- Automated build and test process saves timing and reduces defects
- The code is deployed after every successful build and test
- The development cycle is fast
- New features are more readily available to users and increases profits

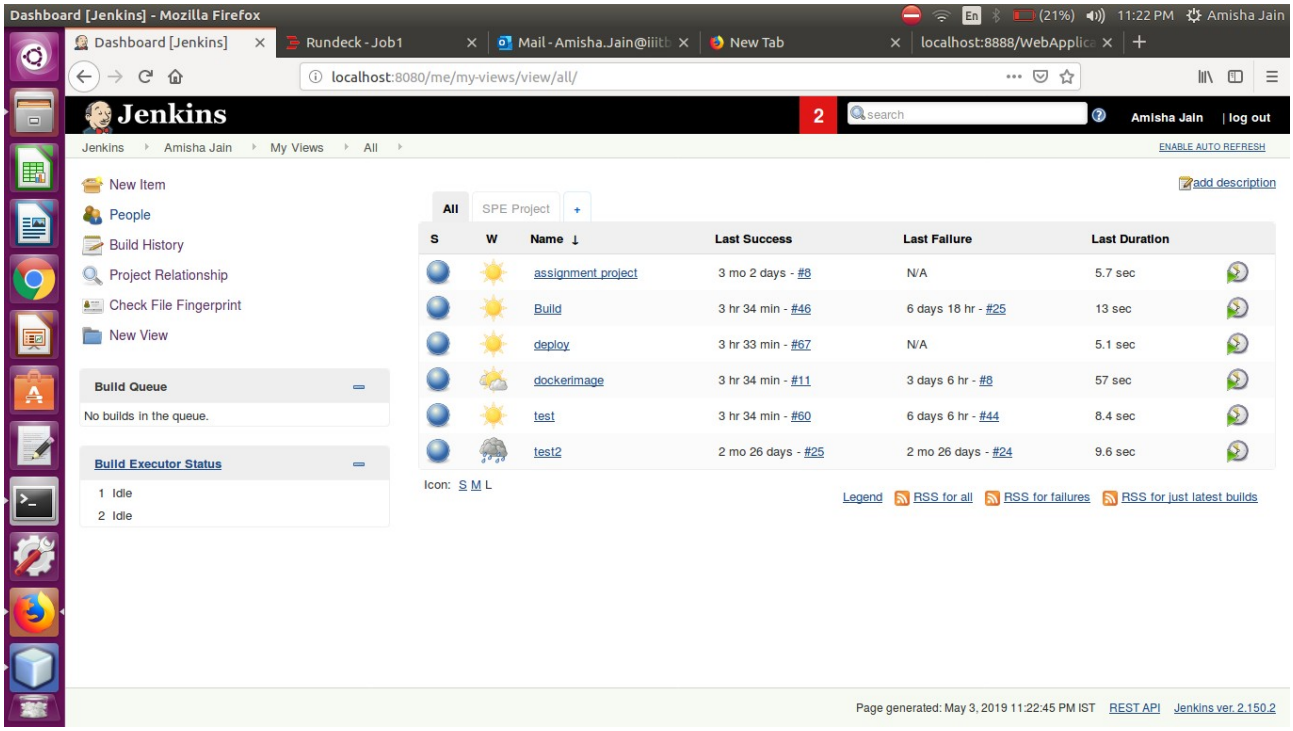


Figure 23. Jenkins Dashboard

4. CI/ CD Pipeline

The pipeline consists of four stages. All the stages are integrated with the same workspace using custom worksapce. Following is the working of the pipeline-

- **Build**- Jenkins pulls the project files from the Git Hub repository in the custom worksapce. It then builds them using Ant tool.
- **Test**- It then runs the specified test cases using JUnit.
- **Docker Image Creation**- Jenkins builds the docker image of the project using docker file and pushes it to Docker Hub.
- **Deploy**- In this stage, RunDeck is triggered from Jenkins which uses the Docker Compose file to deploy the application.

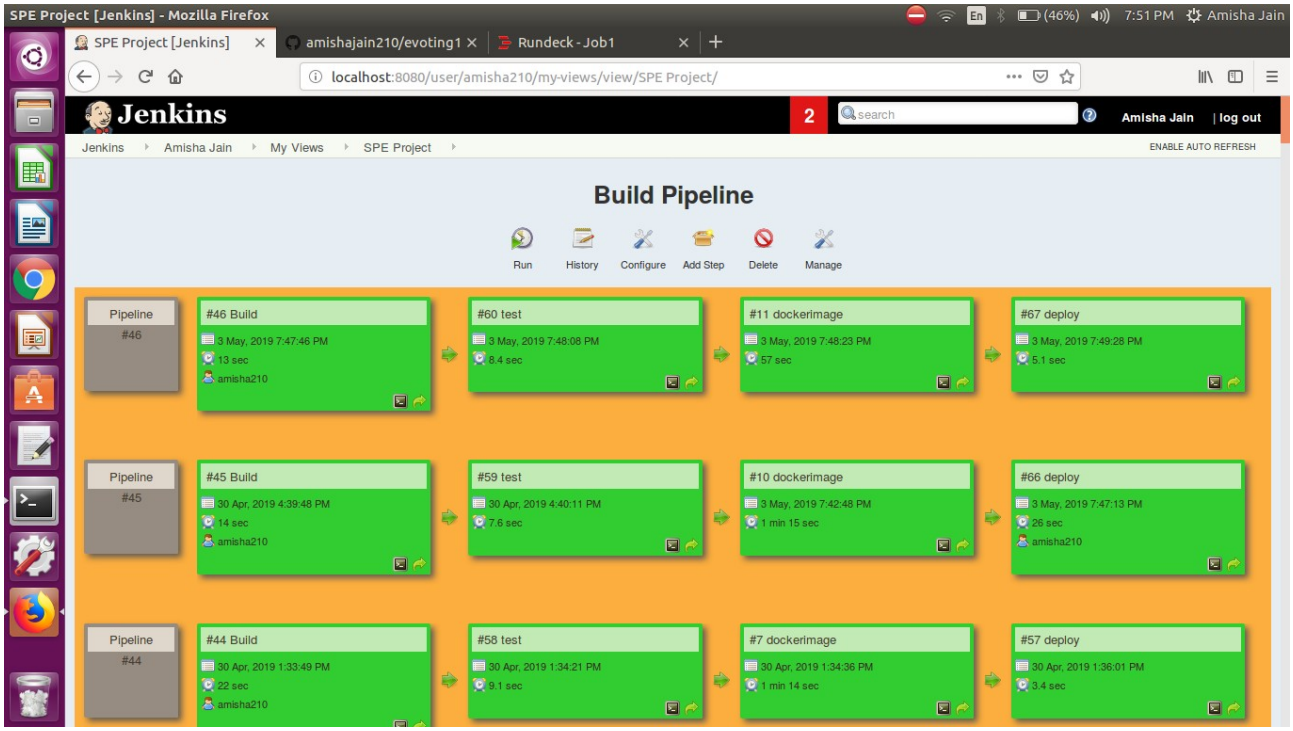


Figure 24. Jenkins Pipeline

5. Result

The application is scalable as it is developed using the MVC architecture. It runs successfully for the desired requirements.

6. Future Work

We can add some more features in our project such as-

- adding more security by using techniques like visual cryptography to generate passwords
- displaying the voting updates in some particular time intervals
- allowing students to give feedback for nominees
- allowing Admin to add a student

7. Conclusion

This project allowed us to learn a lot of new tools and technologies. By integrating DevOps tools we could easily build, test, deploy and monitor the application. This application is very useful for any in-campus elections of this sort. And it can be further scaled for use in various places.