

Project Report  
on  
**IIITB Complaint Portal**  
(CS816 Software Production Engineering)



**Team Members:**

**Aman Gupta (MT2018008)**

**Namanraj Varshney (MT2018067)**

**Rishabh Singh (MT2018095)**

# 1. INTRODUCTION

## 1.1 Why DevOps ?

DevOps is a set of processes that coordinate to unify development teams and processes to complement software development. The main reason behind DevOps' is that it allows enterprises to create and improve products at a faster pace than traditional software development methods. Some of the advantage of devops is:

### **i). Shorter Development Cycles, Faster Innovation:**

When we have a biased response from the development and operations teams, it is often difficult to tell if the application is operational. With joint development and operations efforts, the team's applications are ready to use more quickly.

### **ii). Reduce Implementation Failure, Reflections and Recovery Time:**

The main reason for the failure of the teams in the implementation failure is due to programming defects. With shorter development cycles, DevOps promotes frequent code versions. This, in turn, makes it easy to detect code defects.

### **iii). Better Communication and Cooperation:**

Improved DevOps software development culture. The common teams are happier and more productive. Culture focuses on performance rather than individual goals. When teams trust each other, they can experiment and innovate more effectively.

### **iv). Greater Competencies:**

High efficiency helps accelerate development and makes it less prone to errors. There are ways to automate DevOps tasks. This means that software engineers can focus on completing tasks that can not be automated.

### **v). Agility:**

The key advantage for adopting DevOps in the current business environment is business agility.

### **vi). Innovation:**

DevOps provides the ability for teams to deliver innovation rapidly.

## **1.2 About The Application**

The aim of the project is to automate the complaint system for hostel. This is done to provide a easy platform for the students to register complaints about almost anything starting from Cleaning, Food, Electricity, Sports, Internet, etc.

We have automated every step from development to deployment i.e. to incorporate the complete devops framework to facilitate one click deployment to production.

## **2. Software Development Life Cycle**

### **2.1 Scope of the Project**

The project aims to remove all the pen paper work used for registering student complaints. This process is hectic as well as time consuming.

Using our application, student can register their complaints online easily and can track the status of their complaints whether the complaint is Resolved , Accepted, Rejected or still Pending. The application was designed while keeping in mind to reduce the workload for different committees as well.

**Following are the features of the application :-**

1. Students can Sign Up with Roll no. and College Email Id.
2. Students can recover his Password.
3. Students can submit the complaint in various domains(Internet, Food, Cleaning etc.) with easy interface.
4. Students can track his complaint status.(Resolved , Accepted, Rejected, Pending).
5. Admin from different committees can send message regarding submitted complaints.
6. Students can give feedback.
7. Security is taken care of.

## 2.2 Project Architecture:

Project Architecture is summarized as follows:

### Overview :-

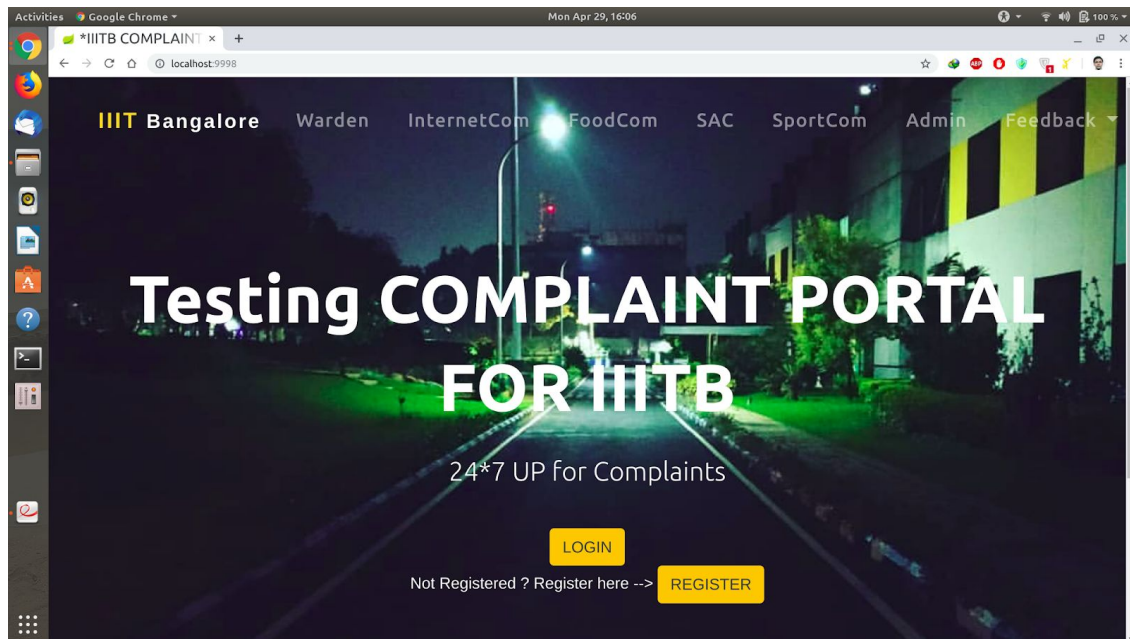
Framework	-	Java Spring Boot Framework
Database	-	MySQL (db4free.net)
Deployment Server	-	Tomcat
Repository Hosting	-	Github
Artifact	-	WAR file
Server	-	Docker Container
Github Repo	-	<a href="https://github.com/raj-naman/CompPortal">https://github.com/raj-naman/CompPortal</a>

### Automation Tools -

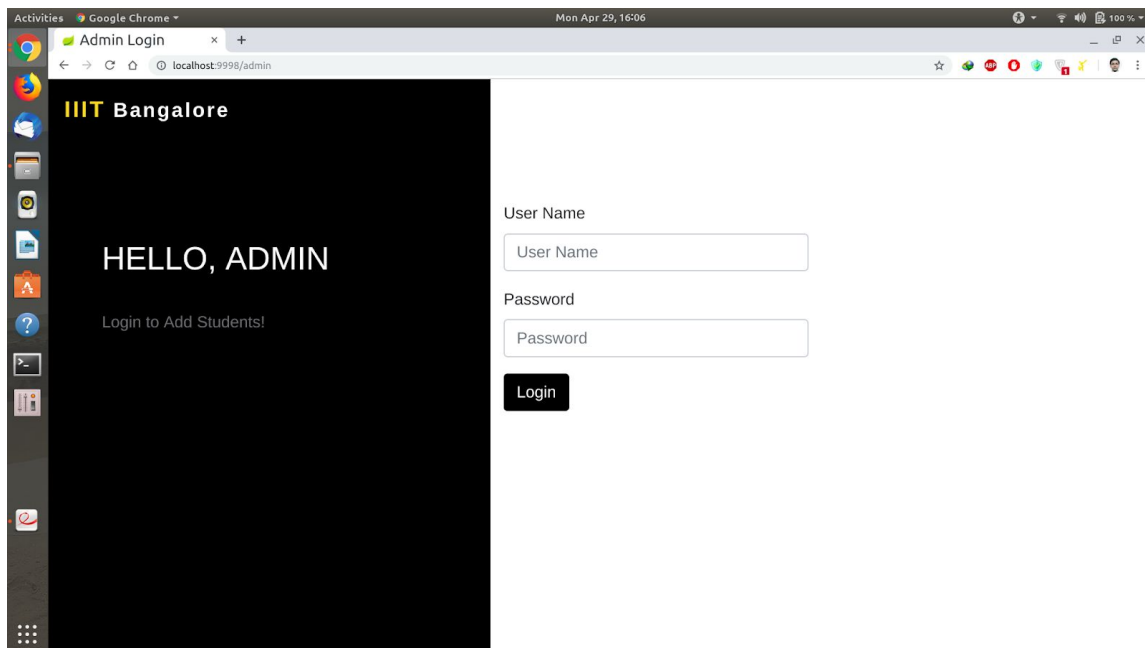
SCM	-	Git
Build	-	Maven integrated with Jenkins
Testing	-	Junit, Mockito
Deployment	-	Jenkins, Rundeck, Docker
Monitoring	-	ELK Stack (Elastic Search, Logstash, Kibana)

## 2.3 Project Workflow:

### Screenshots :



#### 1. Home Page



#### 2. Admin Login Page

Activities Google Chrome Mon Apr 29, 16:07

Admin:Add Student x +

localhost:9998/addStudent

Logout

## Student Entry Form

Roll no.:

Room No.:

Hostel :

Submit

Student with Roll No. MT2018025 and Room No. 436 added Successfully

### 3. Student Entry Form for Admin

Activities Google Chrome Mon Apr 29, 16:07

Register-Yourself x +

localhost:9998/register

IIIT Bangalore

HELLO,  
UNREGISTERED  
STUDENT

Enroll yourself to register Complaints!

Roll No.

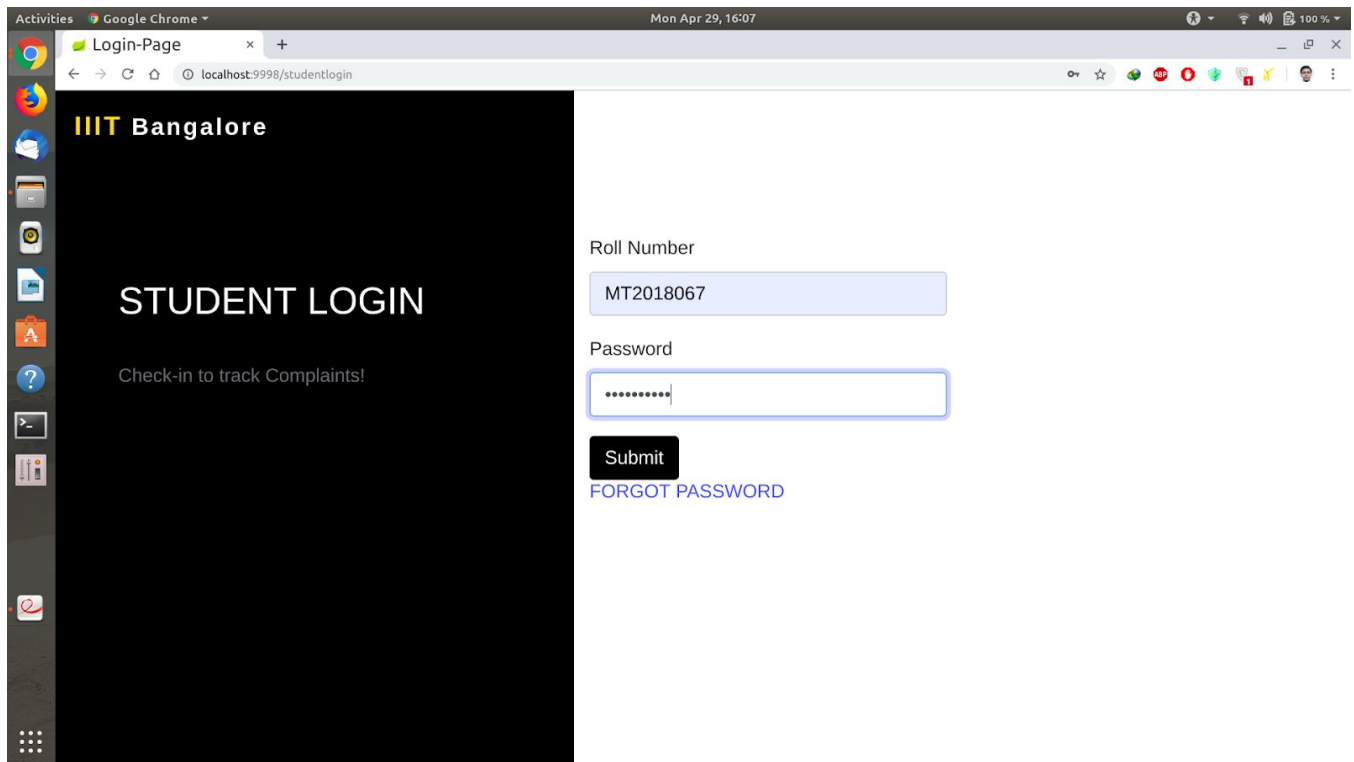
Email(College Id)

Name

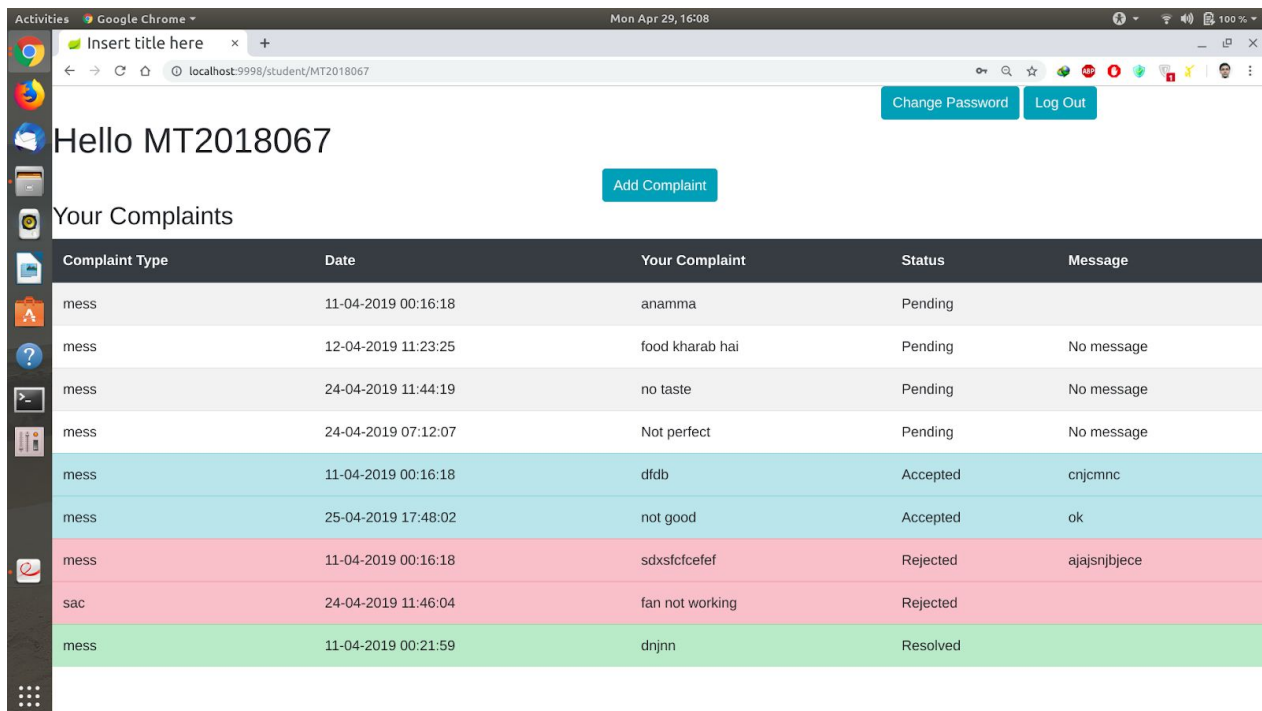
Room No.

Register

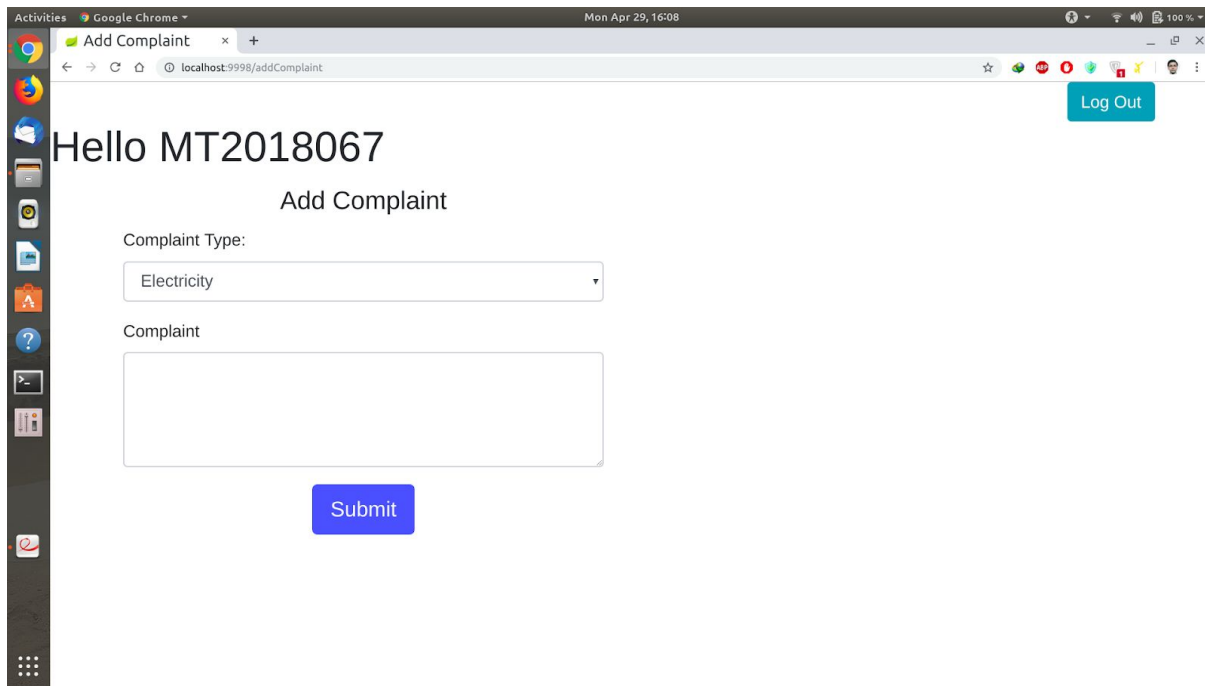
### 4. Student Registration



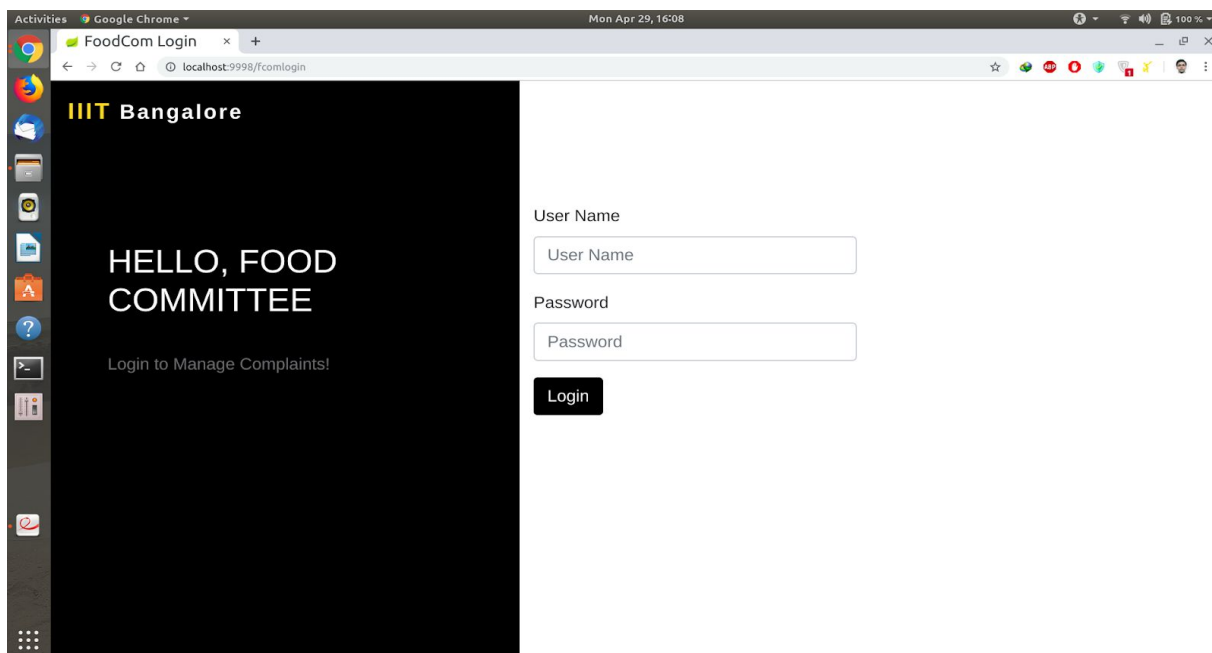
## 5. Student Login



## 6. Student Dashboard



## 7. Add Complaint



## 8. FoodCom Login



Activities Google Chrome Mon Apr 29, 16:08

Welcome Food Committee [Log Out](#)

Roll No	Name	Room No	Date	Status
MT2018067	Namanraj Varshney	505	11-04-2019 00:16:18	Pending
MT2018067	Namanraj Varshney	505	12-04-2019 11:23:25	Pending
MT2018067	Namanraj Varshney	505	24-04-2019 11:44:19	Pending
MT2018067	Namanraj Varshney	505	24-04-2019 07:12:07	Pending
MT2018095	Rishabh Singh	425	26-04-2019 01:53:53	Pending
MT2018067	namanraj11396	145	11-04-2019 00:16:18	Accepted
MT2018067	Namanraj Varshney	505	25-04-2019 17:48:02	Accepted
IMT2014044	Harika	256	26-04-2019 05:19:28	Accepted
MT2018067	Namanraj Varshney	201	11-04-2019 00:16:18	Rejected
MT2018007	Akshat Agrawal	330	24-04-2019 18:57:11	Rejected
MT2018067	Namanraj Varshney	505	11-04-2019 00:21:59	Resolved

## 9. Food Committee Dashboard

Activities Google Chrome Mon Apr 29, 16:09

Insert title here

localhost:9998/foodcomplaint/50

MT2018067  
Namanraj Varshney

Complaint Message :

anamma

Response

Status

Accepted

Message

Submit

## 10. Add Status and Message for User Complaint

## 2.4 Source Code Management (SCM):

Git has been used as the Version Control System integrated with Github for online repository management.

The repository for the project is available at the following link -

<https://github.com/raj-naman/CompPortal>

- GIT has various benefits as VCS listed as follows :-

- Revert the code files back to their previous state.
- Recall and revert the entire project back to its previous state.
- Compare code changes over specific durations of time.
- Find who last modified a piece of code that might be causing an issue or a problem.
- Who introduced a particular issue and when.

and some advantages of Git are:

### **i) Performance:**

Git performs very strongly and reliably when compared to other version control systems. New code changes can be easily committed, version branches can be effortlessly compared and merged, and code can also be optimized to perform better.

### **ii) Security:**

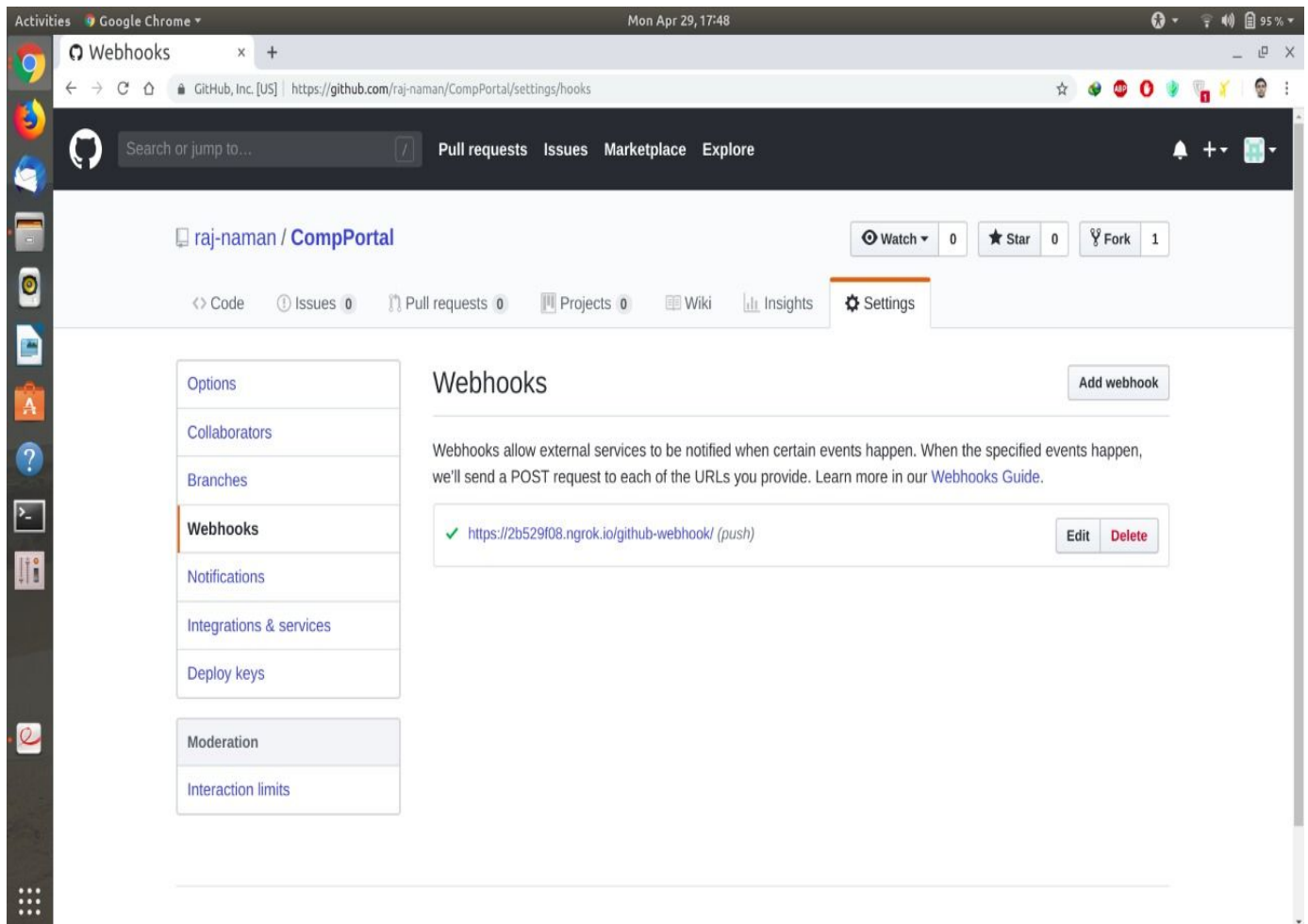
Git is designed specially to maintain the integrity of source code. It uses SHA1 algorithm which protects the code and change history against accidental as well as malicious damage.

### **iii) Flexibility:**

A key design objective of Git is the kind of flexibility it offers to support several kinds of nonlinear development workflows and its efficiency in handling both small scale and large scale projects as well as protocols.

- We have also added ***Git Webhook*** in our application. So, whenever there is a push or change in repository, Jenkins Pipeline will start automatically.

## Webhook Screenshot:



## 2.5 Build :-

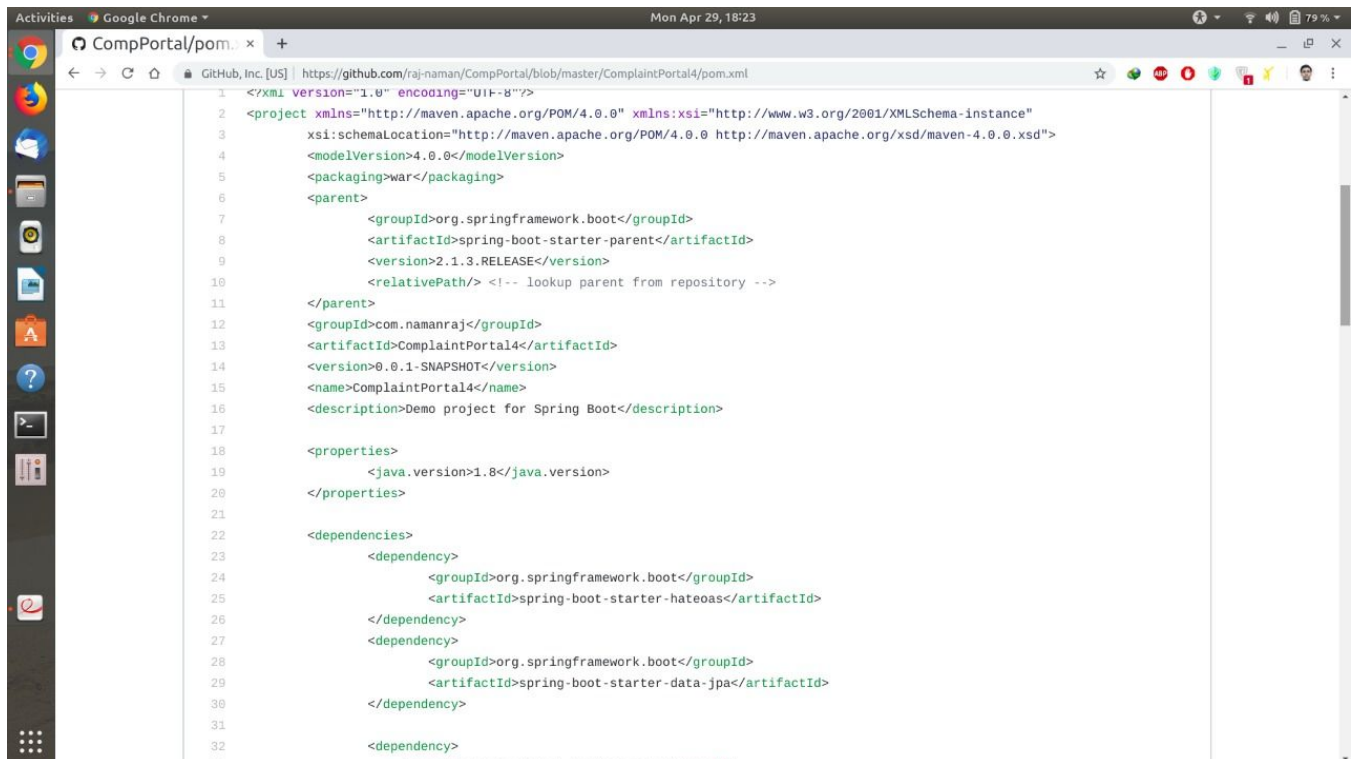
Maven is used with Jenkins for building the application as well as for dependency management. Maven made the dependency management task quite easy.

For the build step we integrated Maven with Jenkins so that whenever the build step is triggered Jenkins first pulls the updated code from the Github repo and then invokes maven to build the application. Following is the pom.xml that we defined for our project.

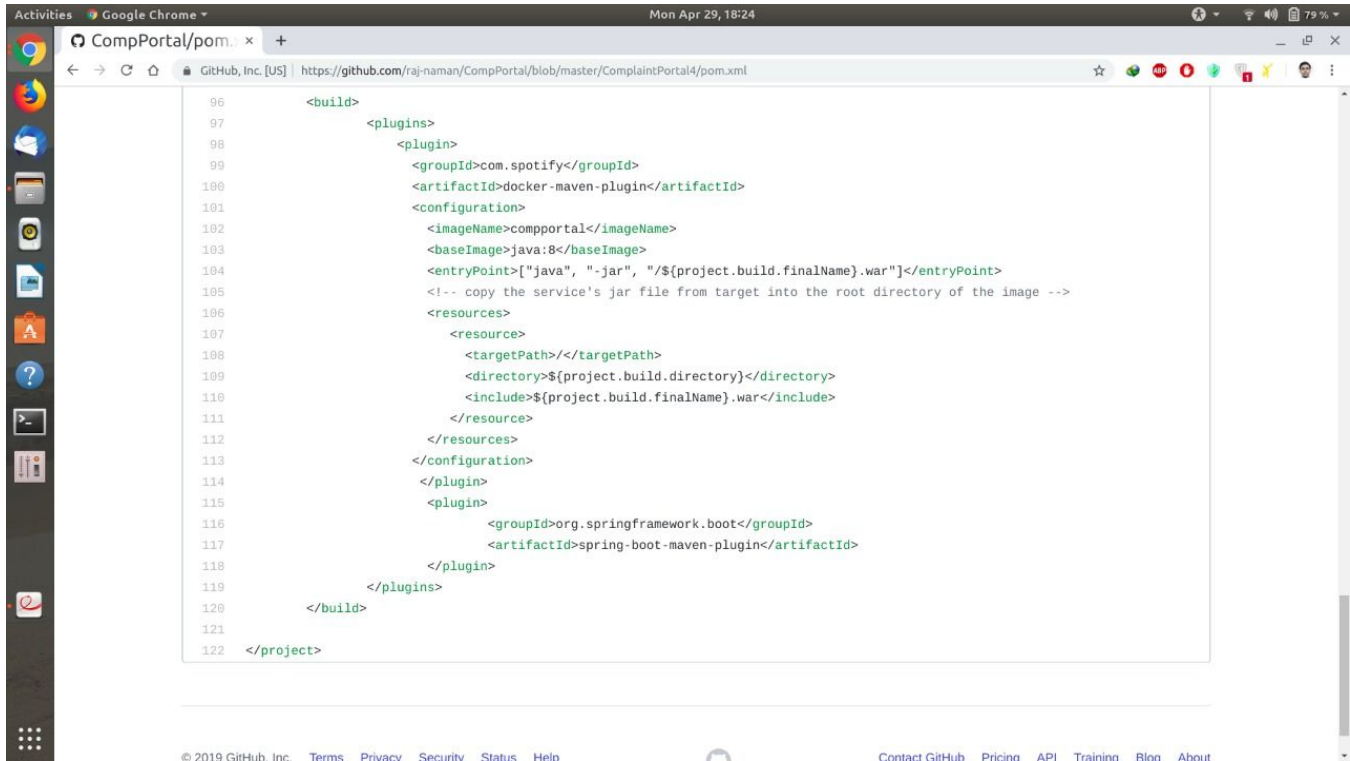
Maven has the following advantages :-

1. It makes a project easy to build.
2. It provides uniform build process (maven project can be shared by all the maven projects)
3. It provides project information (log document, cross referenced sources, mailing list, dependency list, unit test reports etc.)
4. It is easy to migrate for new features of Maven

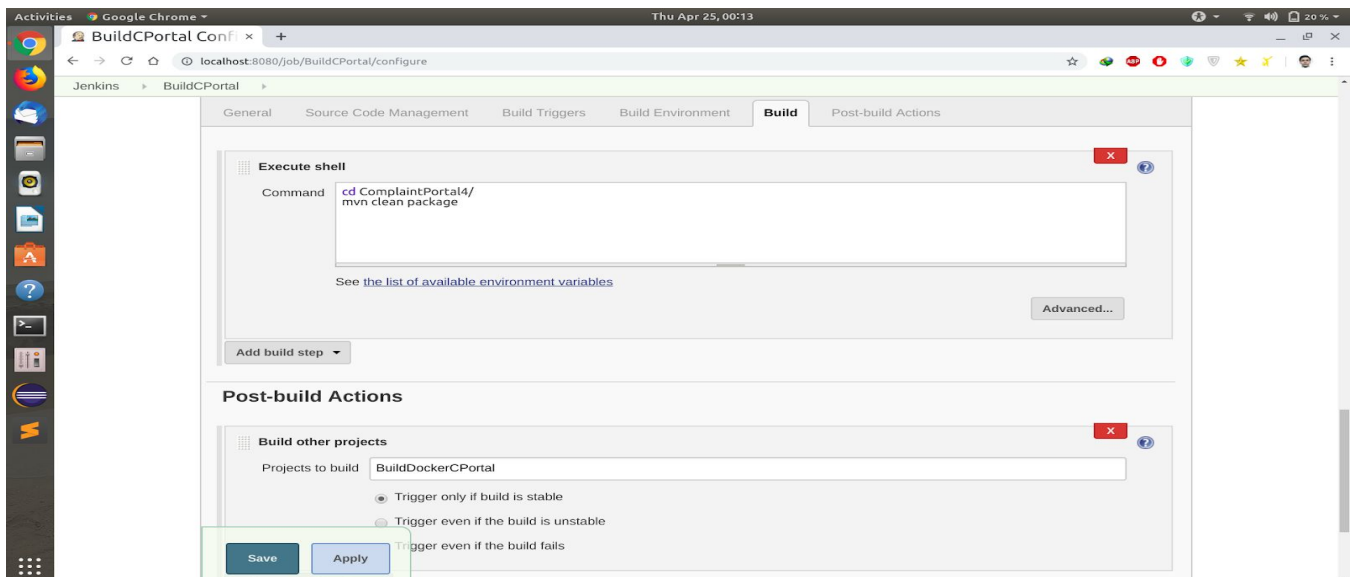
- Pom.Xml file that we used in our project is attached below -



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <packaging>war</packaging>
6     <parent>
7         <groupId>org.springframework.boot</groupId>
8         <artifactId>spring-boot-starter-parent</artifactId>
9         <version>2.1.3.RELEASE</version>
10        <relativePath/> <!-- lookup parent from repository -->
11    </parent>
12    <groupId>com.namanraj</groupId>
13    <artifactId>ComplaintPortal4</artifactId>
14    <version>0.0.1-SNAPSHOT</version>
15    <name>ComplaintPortal4</name>
16    <description>Demo project for Spring Boot</description>
17
18    <properties>
19        <java.version>1.8</java.version>
20    </properties>
21
22    <dependencies>
23        <dependency>
24            <groupId>org.springframework.boot</groupId>
25            <artifactId>spring-boot-starter-hateoas</artifactId>
26        </dependency>
27        <dependency>
28            <groupId>org.springframework.boot</groupId>
29            <artifactId>spring-boot-starter-data-jpa</artifactId>
30        </dependency>
31
32        <dependency>
```



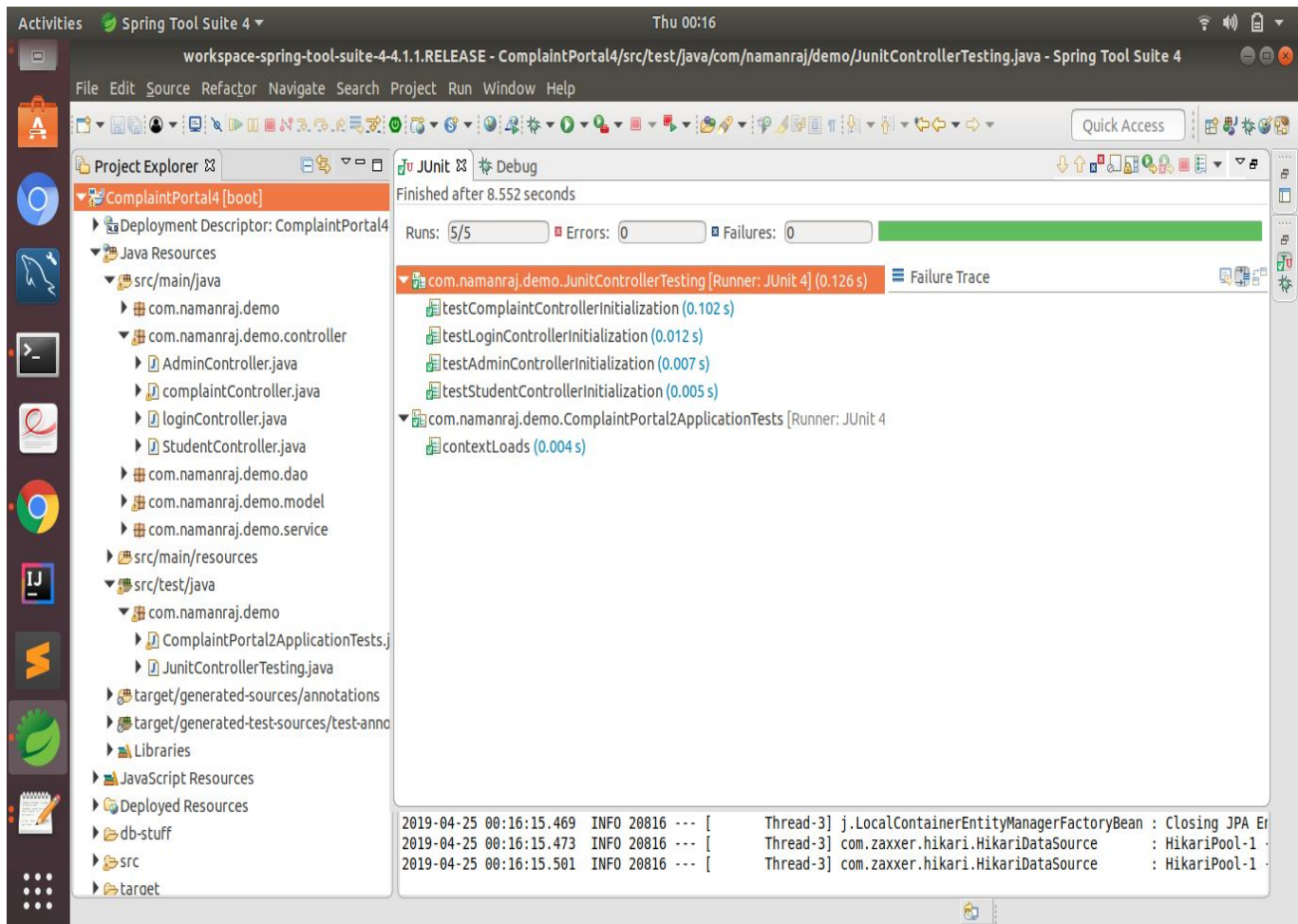
Following is the Jenkins screenshot which is used in building the application -



## 2.6 Testing :-

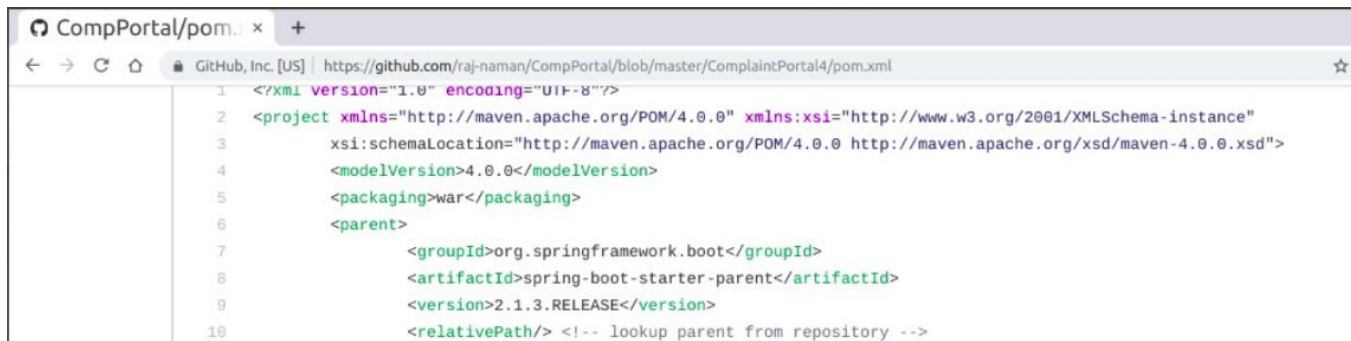
Testing is performed using JUnit and Mojito. Application is developed using Spring Boot and JUnit is used for testing the Java application.

Following code snippet demonstrates how the application is tested using JUnit:



## 2.7 Artifact :-

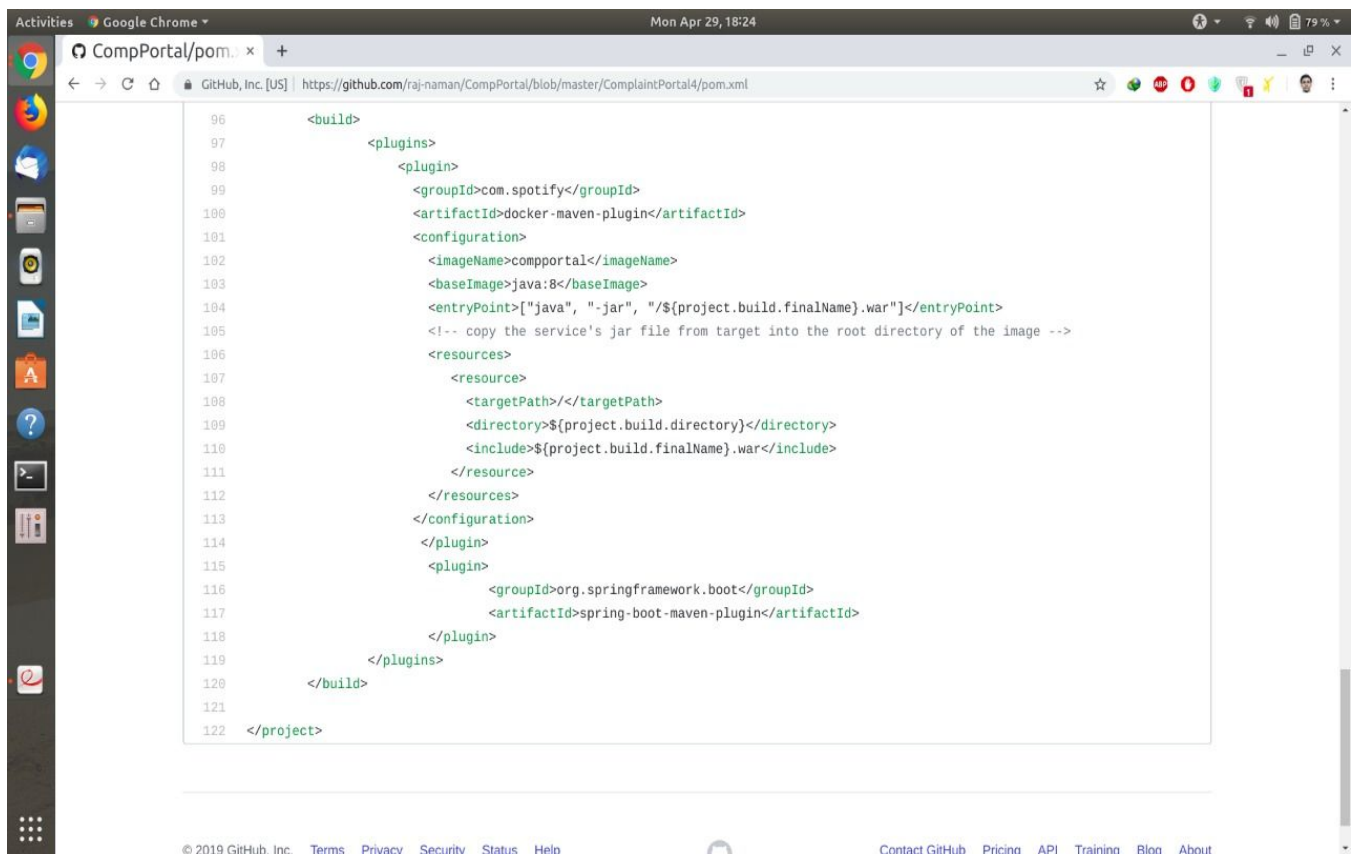
- We are packaging the project as WAR file, the screenshot of which is attached below :-



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <packaging>war</packaging>
6     <parent>
7         <groupId>org.springframework.boot</groupId>
8         <artifactId>spring-boot-starter-parent</artifactId>
9         <version>2.1.3.RELEASE</version>
10        <relativePath/> <!-- lookup parent from repository -->
```

### 1. Application Packaging as WAR

- We are creating Docker Image as follows :-

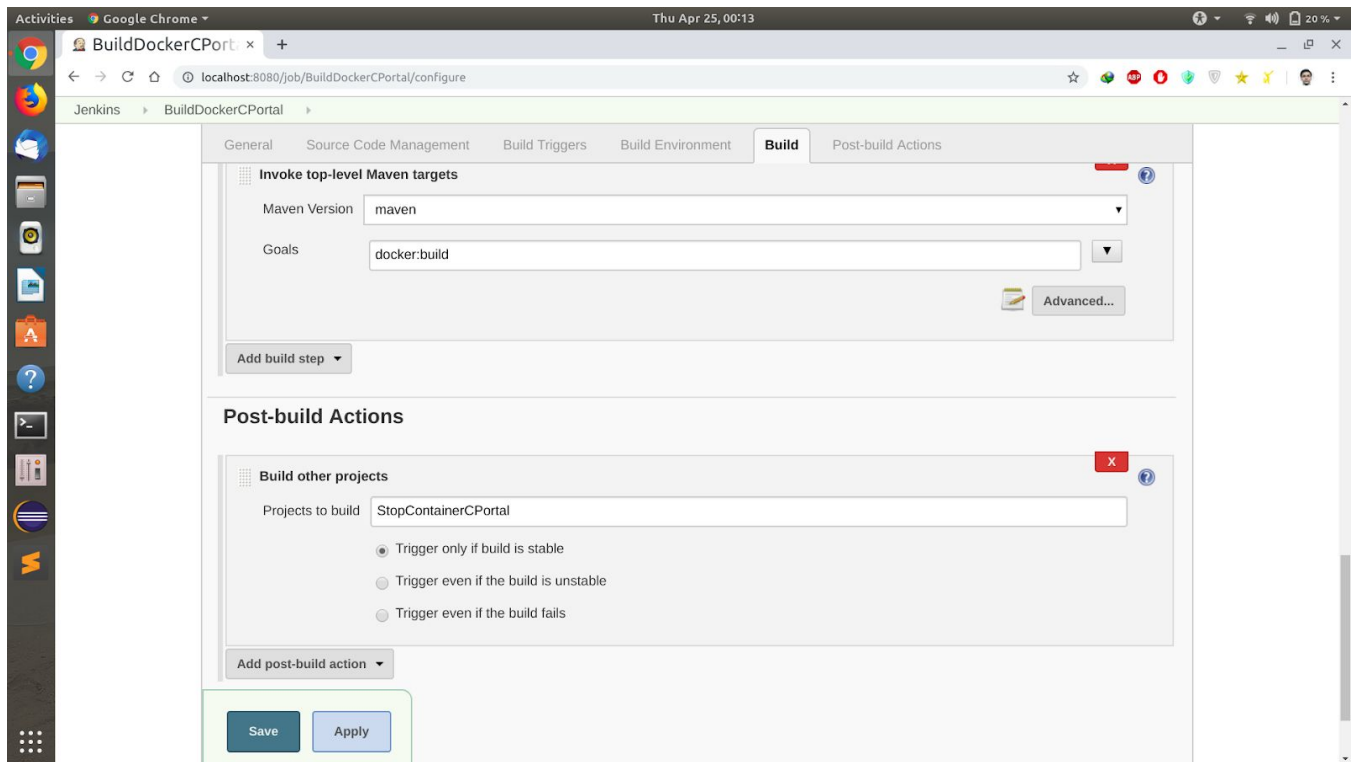


```
96 <build>
97     <plugins>
98         <plugin>
99             <groupId>com.spotify</groupId>
100            <artifactId>docker-maven-plugin</artifactId>
101            <configuration>
102                <imageName>compportal</imageName>
103                <baseImage>java:8</baseImage>
104                <entryPoint>["java", "-jar", "${project.build.finalName}.war"]</entryPoint>
105                <!-- copy the service's jar file from target into the root directory of the image -->
106                <resources>
107                    <resource>
108                        <targetPath/></targetPath>
109                        <directory>${project.build.directory}</directory>
110                        <include>${project.build.finalName}.war</include>
111                    </resource>
112                </resources>
113            </configuration>
114        </plugin>
115        <plugin>
116            <groupId>org.springframework.boot</groupId>
117            <artifactId>spring-boot-maven-plugin</artifactId>
118        </plugin>
119    </plugins>
120 </build>
121
122 </project>
```

### 2. Docker Image Code



- Docker image is created using **docker:build** command as follows :-



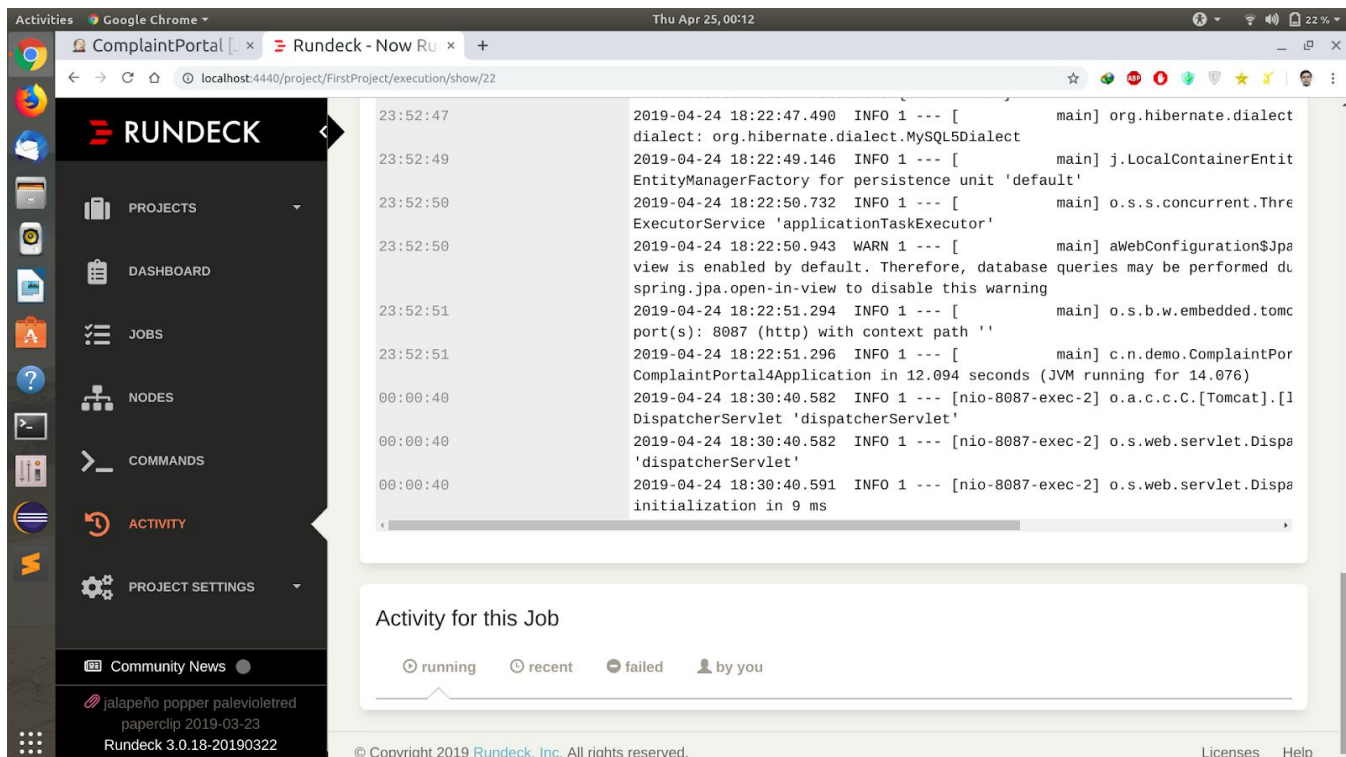
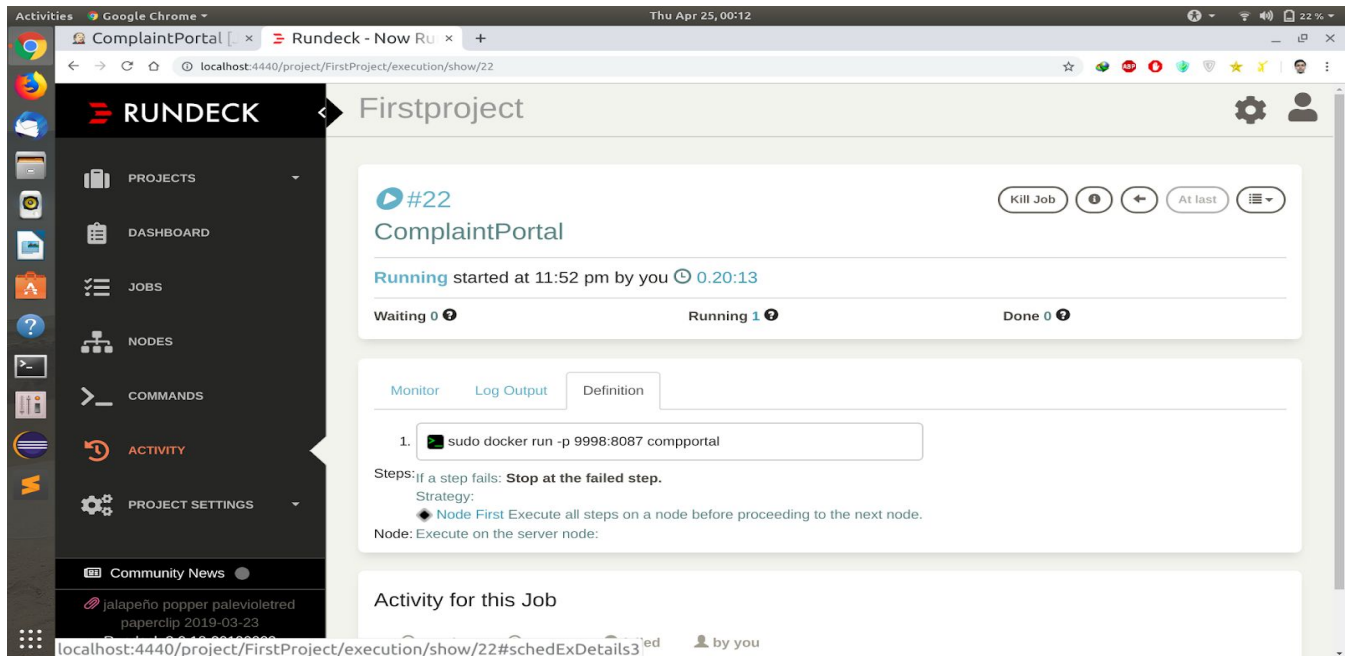
### 3. Building Docker Image



## 2.7 Deployment :-

Rundeck is used to deploy the application.

Container is running using “sudo docker run -p 9998:8087 compportal” command. We are exposing the application on port 8087 and deploying on port 9998 .

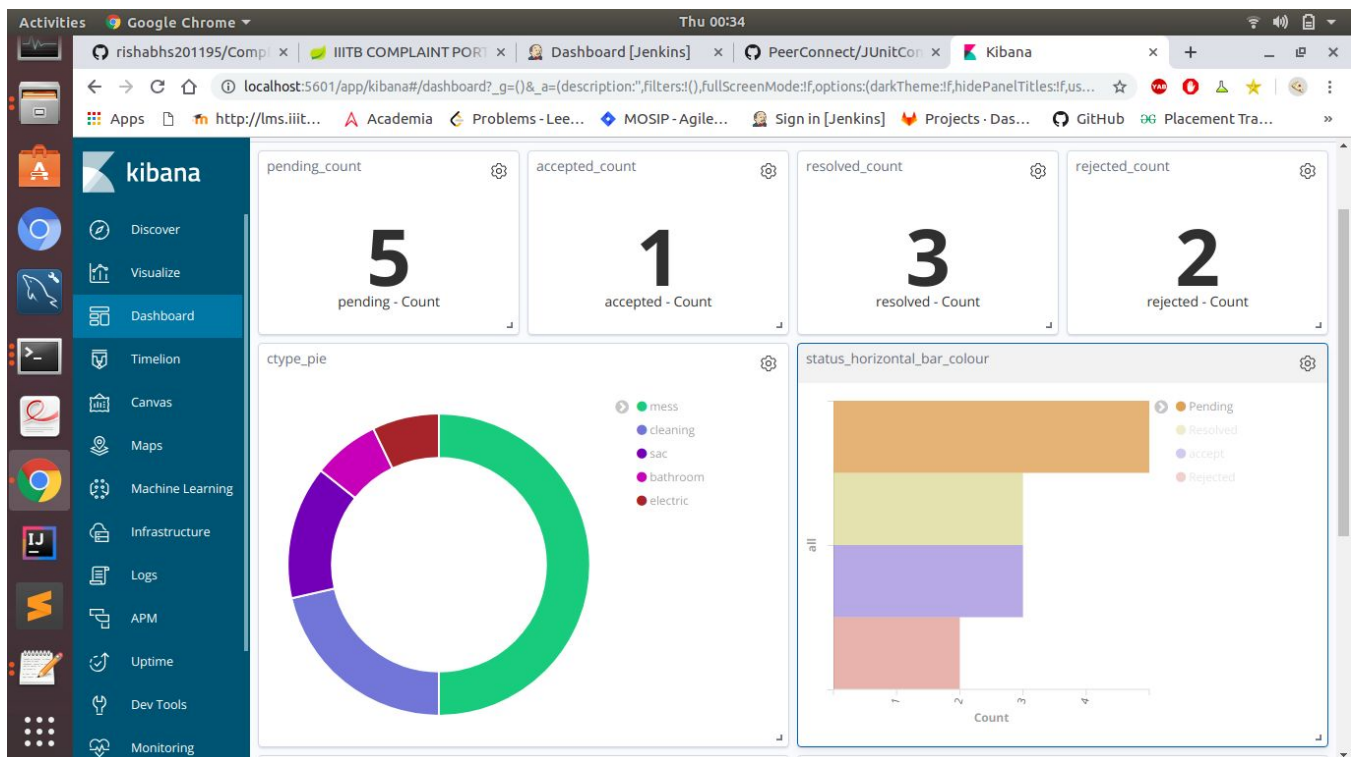


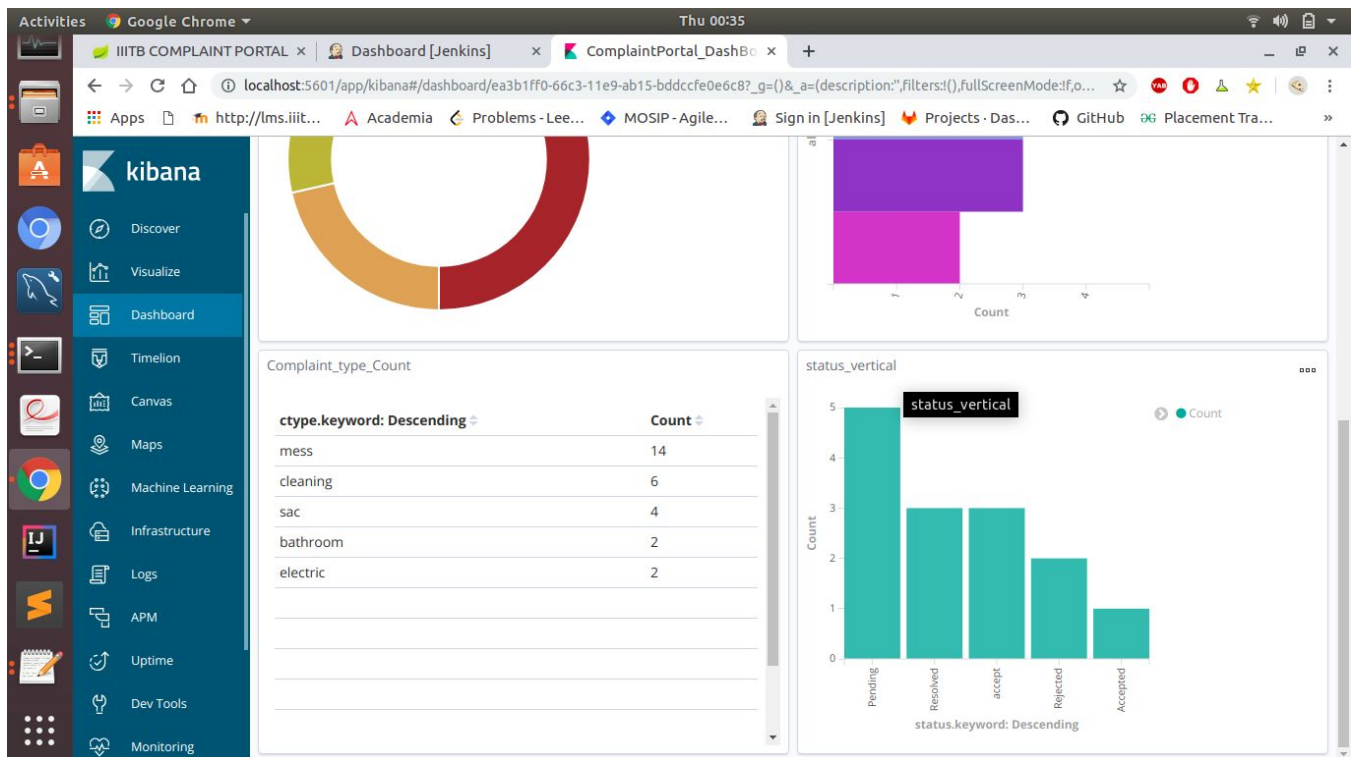
## 2.8 Monitoring :-

We used Elasticsearch with Logstash and Kibana for monitoring mysql database. MySQL database is where the data is safely persisted to disk. So Monitoring, in general, is intended to gives more clear and simpler understanding of complex query to make predictions.

We created few metrics, charts & graphs which shows

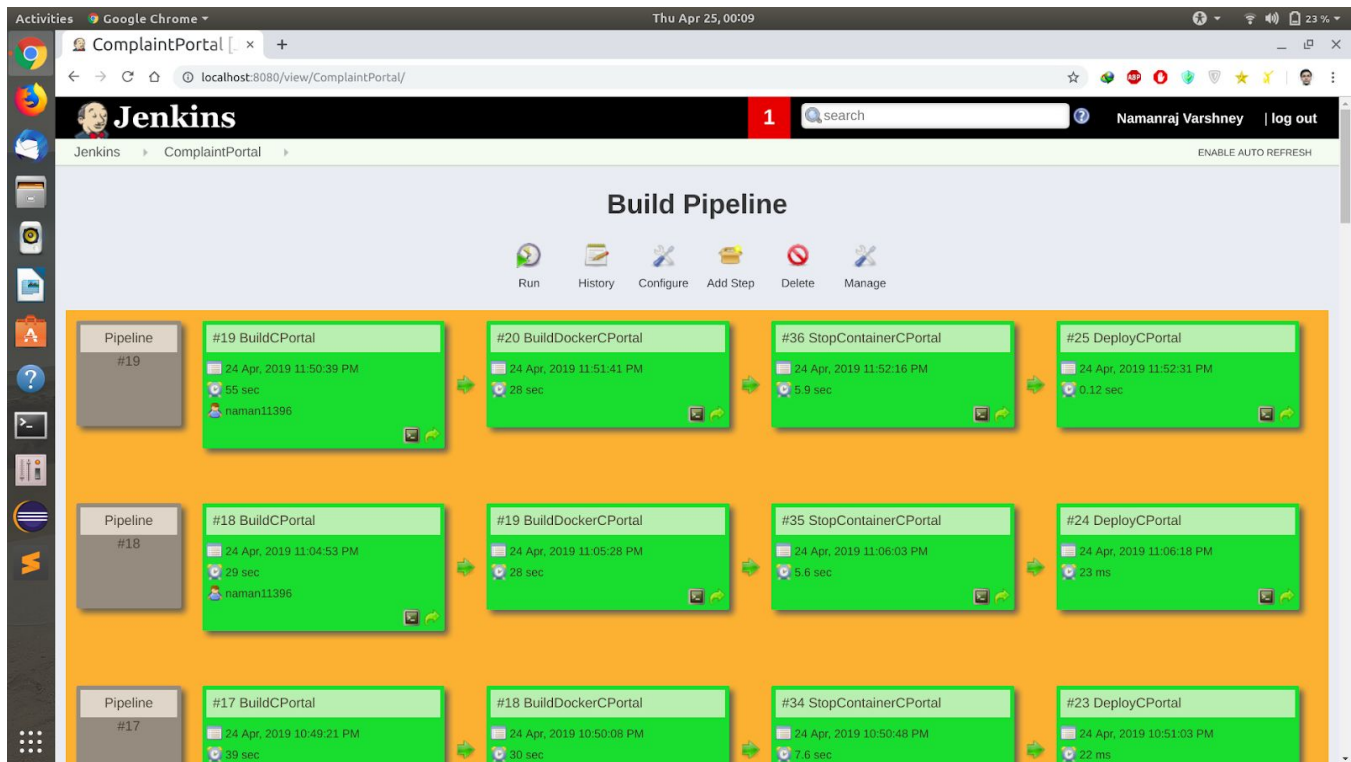
1. Total no of pending complaints
2. Total no. of accepted complaints
3. Total no of rejected complaints
4. Total no of resolved complaints
5. Total no. of complaints for each commitee





### 3. Pipeline :-

We have created the pipeline to process each job one after the other. In post build actions, the next job can be given to create pipeline. Whenever there is a commit in Github, with the help of webhook the pipeline will be started. This will happen in real time and in continuous manner.



### 4. Future Work :-

Many new features can be incorporated into the application, listed as follows :-

1. Display Notifications - Whenever student submit a request, the respective department must get the notification instead of checking it again and again. Same goes with the student. Student must also get the notification if the request has been accepted, rejected or pending or any messages from the department.
2. Provide users with credibility as to how much genuine request they are submitting.
3. Rating system can be developed so that each user profile is given a rating like in ola/uber app.
4. Scalability and security can be enhanced as per the needs.

## 5. Conclusion :-

Overall, it was great experience to learn and implement the DevOps tools as it will give us a headstart in the corporate world where these tools are used on a daily basis.

DevOps tools are making the task quite easier manual interaction is getting reduced. Once all the things are set up, these tools will run automatically and building, integration, deployment, monitoring, testing will take place in a continuous manner.

## 6. References :-

- [1] <https://spring.io/guides/gs/spring-boot/>
- [2] <https://dev.mysql.com/doc/refman/8.0/en/tutorial.html/>
- [3].<https://tomcat.apache.org/tomcat-8.0-doc/config/valve.html>
- [4].<https://guides.github.com/>
- [5].<https://stackoverflow.com/>
- [6].<https://docs.docker.com/get-started/>
- [7].<https://jenkins.io/doc/tutorials/>
- [8].<https://maven.apache.org/guides/getting-started>
- [9].<https://docs.spring.io/spring-boot/docs/current/reference/html/bootfeatures-testing.html>
- [10].[https://docs.chef.io/chef\\_overview.html](https://docs.chef.io/chef_overview.html)
- [11].<https://docs.rundeck.com/docs/tutorials/>
- [12].<https://www.elastic.co/guide/en/elasticsearch/reference/current/gettingstarted.html>