

International Institute of Information Technology, Bangalore

Software Production Engineering

COMMUNITY SHARING HUB

Under the Guidance of : Prof. B. Thangaraju

Teaching Assistant: Vivek Chaudhary



**VAIBHAV TANDON
MT2020150**

**VIVEK SINGH THAKUR
MT2020074**

Table of Contents

Contents

1. Abstract.....	4
2. Introduction.....	5
2.1 Overview.....	5
2.2. FEATURES	5
2.3. WHY DEVOPS?.....	5
2.3.1. Devops Features.....	6
3. System Configuration	7
3.1. Operating system	7
3.2. CPU and RAM.....	7
3.3. Language.....	7
3.4. Kernel Version	7
3.5. Database	7
3.6. Building tools.....	8
3.7. DevOps Tools	8
4. Software Development life cycle.....	9
4.1. INSTALLATION	9
4.1.1. Angular Web Framework	9
1.Download nodejs.....	9
4.1.1. Spring Framework	12
4.2 Source Control Management	14
4.2.1 CI Pipeline.....	16
4.3 Build.....	19
4.4 Testing.....	20
4.4.1 CI Pipeline.....	22
4.5 DOCKER	23
4.6 DEPLOY	27

4.6.1 Ansible setup.....	28
4.6.2 CI PIPELINE	30
4.7 Monitor.....	32
4.8 Mysql.....	38
5.EXPERIMENTAL SETUP.....	42
5.1 DESIGN DIAGRAM.....	42
6. Results and Discussion	43
6.1 Project snapshots.....	43
6.1.1 Borrower details.....	43
6.1.2 Displaying messages on wall of non registered users	43
6.1.3 Home Page	44
6.1.4 list of users on okta dashboard.....	44
6.1.5 list of products of different category.....	45
6.1.6 Login success	45
6.1.7 Message list for registered user	46
6.1.8 Product description	46
6.1.9 Order Review	47
6.1.10 Product Search results.....	47
6.1.11 Sign-in.....	48
6.1.12 Sign Up	48
6.1.13 Updating Message.....	49
6.1.14 Product Cart	49
6.1.15 Wall for Registered User	50
7.Scope for Future Work.....	51
7.1 End to End Communication.....	51
7.2 User Rewards	51
8. CONCLUSION.....	52
9.References	53

1. Abstract

Community Sharing hub is an online portal for people inside a Community to Share Their Resources and Information inside that Community. It can be used by people of Community to share their Information and Products inside their Community. One Author defined it as **“One of the marvelous things about community is that it enables us to welcome and help people in a way we couldn't as individuals.”**

The architecture of our project demands three layers.

- Front end
- Middle layer
- Back end

The front end of the project is handled by “Angular” Framework. The middle layer is built on “SpringBoot” Framework and communicates with mysql database to show the content on the front end via Rest Api.

2. Introduction

2.1 Overview

Community is for everyone. Community sharing hub is online social Platform for all those who want to help there Community by sharing their worthy Resources and Information. Community Sharing Hub provides you with one interactive user interface where you can do all this activity hassle free. Community Sharing Hub can be used by different communities across the globe to help boost up interaction and growth among the members of their Community.

2.2. FEATURES

- Information Sharing - Users can share any vital information
- Showcasing Resources - Users can showcase what Resources related to any particular domain they possess
- Selling and Sharing - Users can share and sell their resources

2.3. WHY DEVOPS?

Our whole approach of the project was modular, we wanted to make different sets of development modules and wanted to deploy them with every new release without any hindrance. Wanted to test the changed code with continuous integration and then continuously deploying it. So what all fills all these blanks was a culture, a philosophy DevOps. Devops provides all the tools to increase the capability to complete above set goals within minimum time and less trouble for developers. DevOps tools consist of configuration management, test and build systems, application deployment, version control and monitoring tools. Continuous integration, continuous delivery and continuous deployment require different tools.

2.3.1. Devops Features

- Improve deployment frequency
- Achieve faster time to market with lower failure rate
- More stable operating environments
- Improve communication and collaboration among teams

3. System Configuration

3.1. Operating system

Linux Mint 19.3 Tricia and Ubuntu 18.04.04 Bionic Beaver.

3.2. CPU and RAM

4 core processor and RAM 8 GB (preferable 16 GB)

3.3. Language

JavaScript XML JSX, Angular web framework

Java, Spring framework

3.4. Kernel Version

Linux Machine 5.3.0-53-generic

3.5. Database

MySQL 8.0.20 (MySQL Community Server - GPL)

3.6. Building tools

Maven to build java application

3.7. DevOps Tools

- Source Control Management - GitHub
- Continuous Integration - Jenkins
- Containerization - Docker
- Continuous deployment - Ansible
- Monitoring - ELK Stack (Elastic Search, Logstash, Kibana)

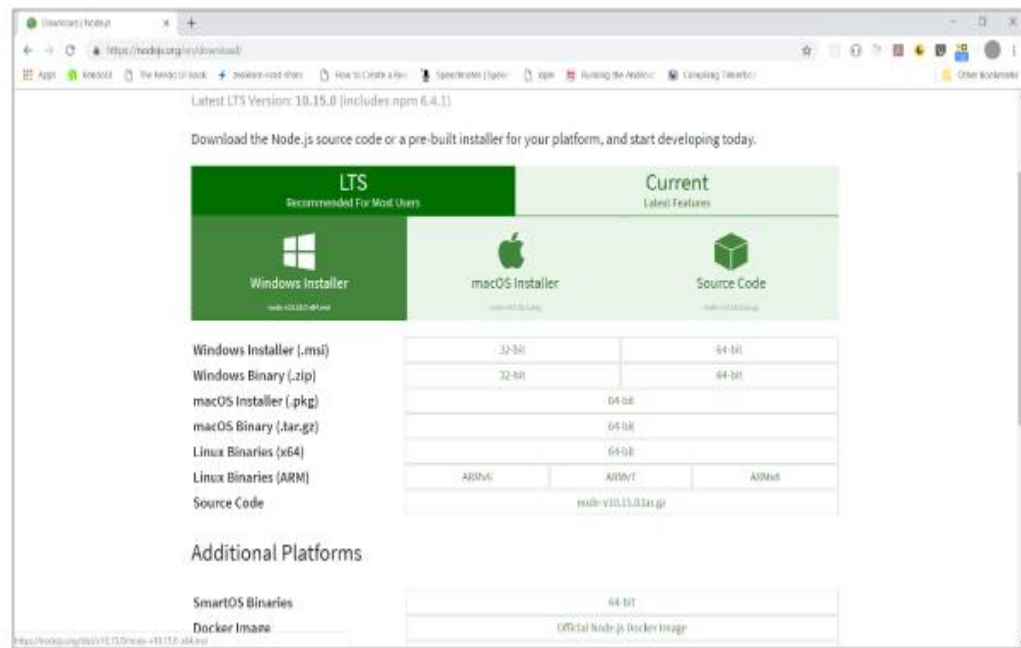
4. Software Development life cycle

4.1. INSTALLATION

4.1.1. Angular Web Framework

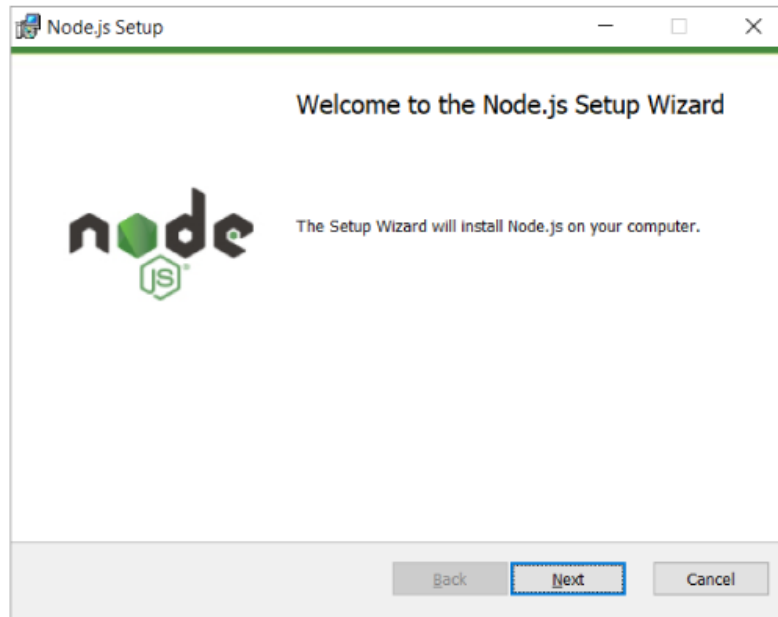
1.Download nodejs

1.Visit nodejs download page [here](https://nodejs.org/en/download/)

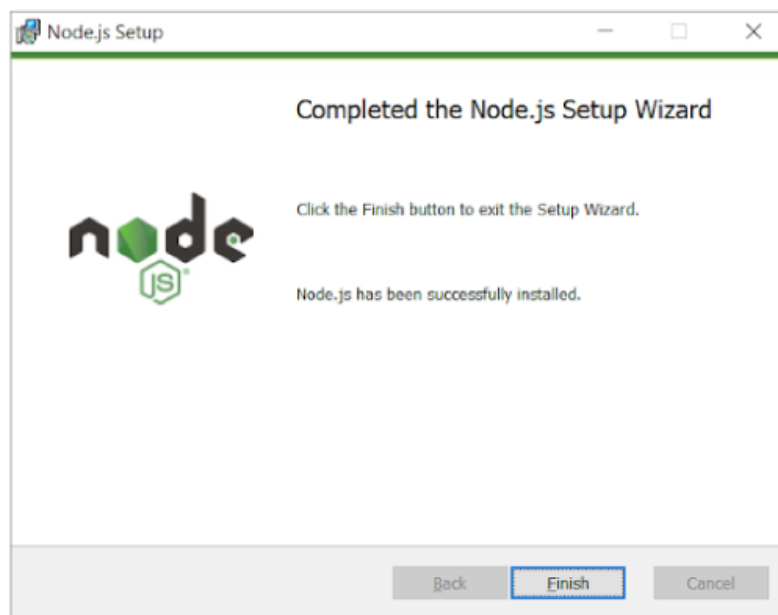


2.Install nodejs

1.Click on the downloaded node-vxx.xx.xx.msi (for example node-v10.15.0.msi) in previous step to start the installation which brings up below screen. Please click **Next**



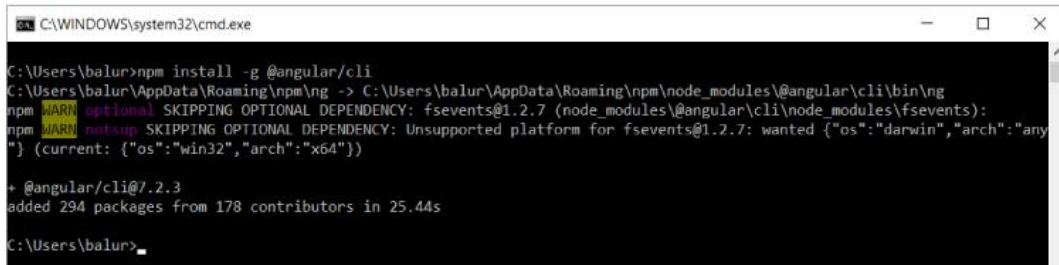
Complete the required steps to finish Installation



Install angular cli

We will install angular cli using npm. On terminal run the install command shown below

```
npm install -g @angular/cli
```



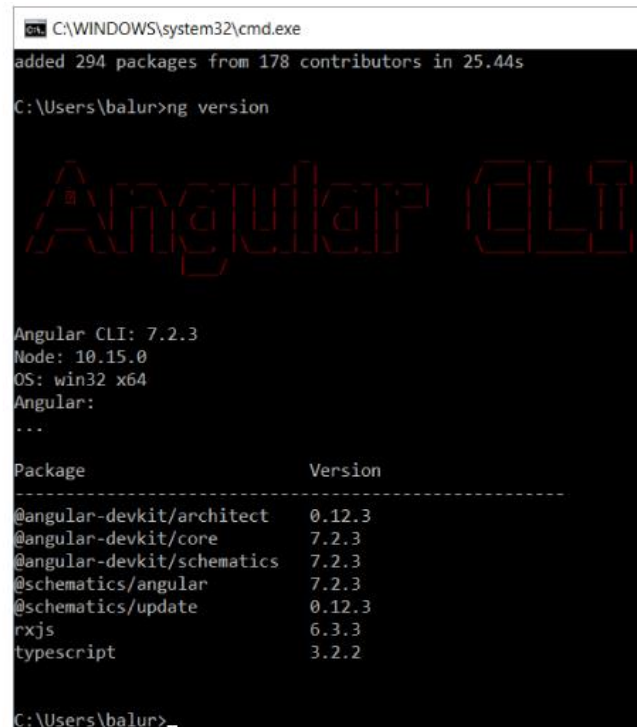
```
C:\WINDOWS\system32\cmd.exe
C:\Users\balur>npm install -g @angular/cli
C:\Users\balur\AppData\Roaming\npm\ng -> C:\Users\balur\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node_modules\@angular\cli\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.7: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ @angular/cli@7.2.3
added 294 packages from 178 contributors in 25.44s
C:\Users\balur>
```

On successful installation you should see the output like above (note your angular/cli version may be different by the time you run this install command)

Test @angular/cli

To test the @angular/cli run ng version command



```
C:\WINDOWS\system32\cmd.exe
added 294 packages from 178 contributors in 25.44s
C:\Users\balur>ng version

Angular CLI
Angular CLI: 7.2.3
Node: 10.15.0
OS: win32 x64
Angular:
...

Package                           Version
-----
@angular-devkit/architect         0.12.3
@angular-devkit/core              7.2.3
@angular-devkit/schematics        7.2.3
@schematics/angular              7.2.3
@schematics/update                0.12.3
rxjs                              6.3.3
typescript                       3.2.2
C:\Users\balur>
```

4.1.1. Spring Framework

Here we have used java to create the Community Sharing Hub code. And maven was used to build the

Spring java project. Reference the figure.

[illegible]

```

sudo dpkg-query -f='${Package} ${Version} ${Architecture}\n' -W -f='${Package} ${Version} ${Architecture}\n' | grep libplexus
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libc-ares2 libhttp-parser2.7.1 libssl1.0.0 libuv1 nodejs-doc
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  libapache-pom-java libapache-pom-jar330-api-java libcdi-api-java libcommons-cli-java libcommons-io-java libcommons-lang3-java
  libcommons-parent-java libgeronimo-annotation-1.3-spec-java libgeronimo-interceptor-3.0-spec-java libguava-java libguice-java libhawtjni-runtime-java libjansi-java
  libjansi-native-java libjstax25-java libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java libmaven3-core-java libplexus-cipher-java
  libplexus-classworlds-java libplexus-component-annotations-java libplexus-interpolation-java libplexus-sec-dispatcher-java libplexus-utilities2-java libsisu-inject-java
  libsisu-plexus-java libslf4j-java libwagon-file-java libwagon-http-shaded-java libwagon-provider-api-java
Suggested packages:
  libapache-pom-java-doc libatinject-jstax25-api-java-doc libhservlet3.1-java libcommons-io-java-doc libcommons-lang3-java-doc libbase-java libcglib-java
  libjstax25-java-doc libmaven-shared-utils-java-doc libhiback-java libplexus-cipher-java-doc libplexus-classworlds-java-doc libplexus-interpolation-java-doc
  libplexus-sec-dispatcher-java-doc libplexus-utilities2-java-doc junit4 testng libcommons-lang3-java liblog4j1.2-java
WARNING: The following new packages will be installed:
  libapache-pom-java libapache-pom-jar330-api-java libcdi-api-java libcommons-cli-java libcommons-io-java libcommons-lang3-java
  libcommons-parent-java libgeronimo-annotation-1.3-spec-java libgeronimo-interceptor-3.0-spec-java libguava-java libguice-java libhawtjni-runtime-java libjansi-java
  libjansi-native-java libjstax25-java libmaven-parent-java libmaven-resolver-java libmaven-shared-utils-java libmaven3-core-java libplexus-cipher-java
  libplexus-classworlds-java libplexus-component-annotations-java libplexus-interpolation-java libplexus-sec-dispatcher-java libplexus-utilities2-java libsisu-inject-java
  libsisu-plexus-java libslf4j-java libwagon-file-java libwagon-http-shaded-java libwagon-provider-api-java maven
0 upgraded, 33 newly installed, 0 to remove and 23 not upgraded.
Need to get 8918 kB of archives.
After this operation, 11.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

```

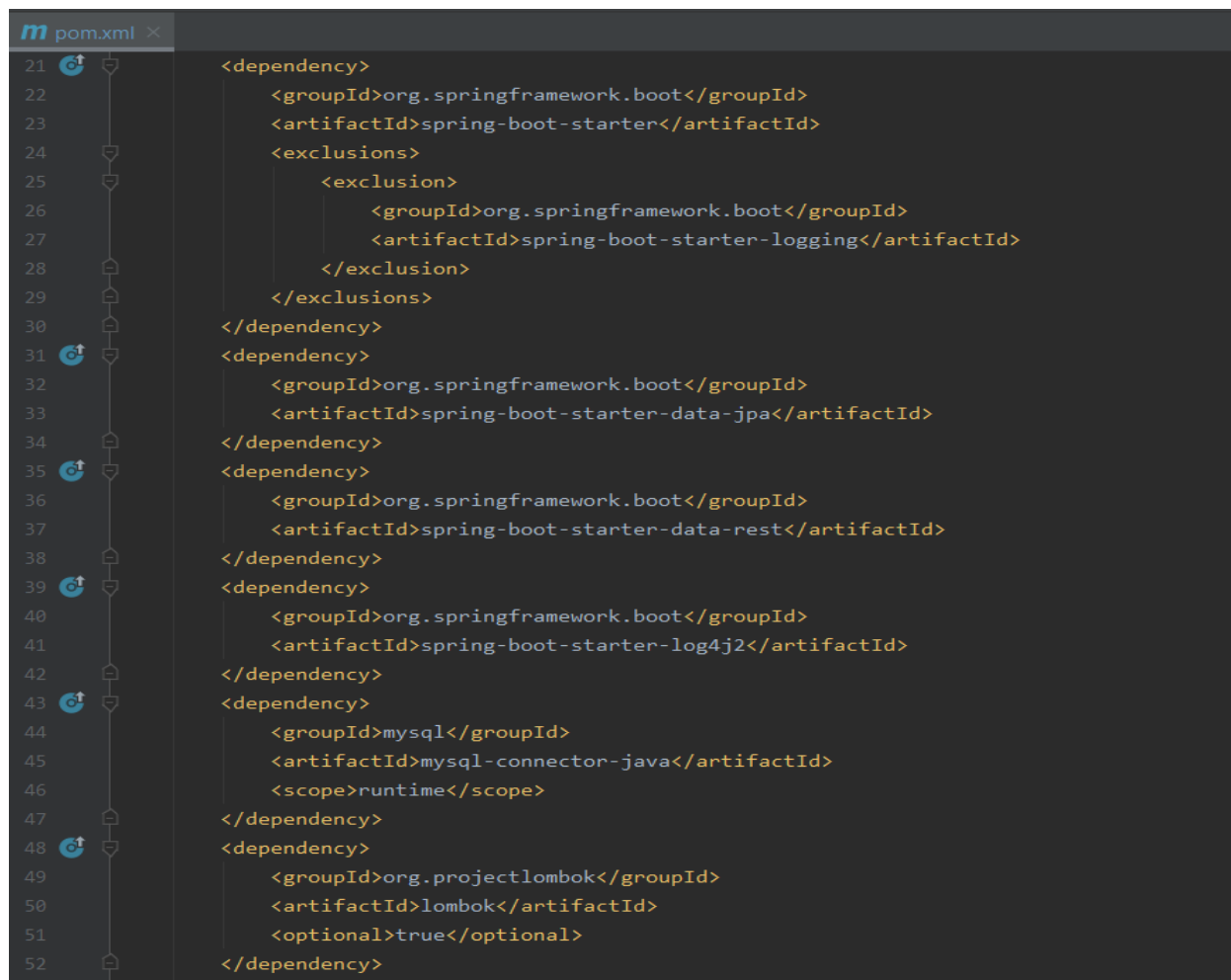
root@56d91278fd4:/# java --version
openjdk 11.0.7 2020-04-14
OpenJDK Runtime Environment (build 11.0.7+10-post-Ubuntu-2ubuntu218.04)
OpenJDK 64-Bit Server VM (build 11.0.7+10-post-Ubuntu-2ubuntu218.04, mixed mode, sharing)
root@56d91278fd4:/# mvn --version
Apache Maven 3.6.0
Maven home: /usr/share/maven
Java version: 11.0.7, vendor: ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en_US, platform encoding: ANSI_X3.4-1968
OS name: "linux", version: "5.3.0-53-generic", arch: "amd64", family: "unix"
root@56d91278fd4:/#

```

Use <https://start.spring.io/> to create a spring project outer layer and choose dependencies accordingly.

Use mvn clean install to build your spring project, this will make an executable jar file in the target folder of your project.

Run the jar file using java -jar jarname IntelliJ was used as an IDE to develop the project



```

m pom.xml x
21 <dependency>
22     <groupId>org.springframework.boot</groupId>
23     <artifactId>spring-boot-starter</artifactId>
24     <exclusions>
25         <exclusion>
26             <groupId>org.springframework.boot</groupId>
27             <artifactId>spring-boot-starter-logging</artifactId>
28         </exclusion>
29     </exclusions>
30 </dependency>
31 <dependency>
32     <groupId>org.springframework.boot</groupId>
33     <artifactId>spring-boot-starter-data-jpa</artifactId>
34 </dependency>
35 <dependency>
36     <groupId>org.springframework.boot</groupId>
37     <artifactId>spring-boot-starter-data-rest</artifactId>
38 </dependency>
39 <dependency>
40     <groupId>org.springframework.boot</groupId>
41     <artifactId>spring-boot-starter-log4j2</artifactId>
42 </dependency>
43 <dependency>
44     <groupId>mysql</groupId>
45     <artifactId>mysql-connector-java</artifactId>
46     <scope>runtime</scope>
47 </dependency>
48 <dependency>
49     <groupId>org.projectlombok</groupId>
50     <artifactId>lombok</artifactId>
51     <optional>true</optional>
52 </dependency>

```

Spring, MySQL and Lombok Dependencies to pom.xml

4.2 Source Control Management

A Source Code Management (SCM) is a software tool used by programmers to manage the source codes. For our project every team member would clone the repository from github. Create a different branch locally on their system and then merge it with master after pulling the latest code from git, resolving any conflicts and then push the changes to git.

- **git clone <repository url>**- This command copies the entire data on the git url
- **git checkout -b <branch_name>**- This command creates a new branch with the name as in 'branch_name'
- **git add <changed files>**- This command adds changes in the working directory to the staging area
- **git commit -m "message while committing"**-This command is used to save your changes to the local repository with -m used to provide a concise description that helps your teammates (and yourself) understand what happened.
- **git checkout master**- This command switches to master branch
- **git pull**-This command is used to update the local version of a repository from a remote.
- **git merge <branch_name>** -This command is used to integrate changes from another branch.
- **git push**-This command will push all the latest code to the repository.

Github repository links for the project

- <https://github.com/vtandonv/CommunityHubBackend.git>
- <https://github.com/vtandonv/CommunityHubFrontend.git>

vtandonv / CommunityHubFrontend

Unwatch

1

Star

0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main

1 branch

0 tags

Go to file

Add file

Code

vaibhtan Added .yml and inventory file			frf5e38 5 hours ago	2 commits
deploy-docker	Added .yml and inventory file		5 hours ago	
e2e	Added docker file		6 hours ago	
src	Added docker file		6 hours ago	
.editorconfig	Added docker file		6 hours ago	
.gitignore	Added docker file		6 hours ago	
Dockerfile	Added docker file		6 hours ago	
README.md	Added docker file		6 hours ago	
angular.json	Added docker file		6 hours ago	
browserslist	Added docker file		6 hours ago	
karma.conf.js	Added docker file		6 hours ago	
package-lock.json	Added docker file		6 hours ago	
package.json	Added docker file		6 hours ago	
tsconfig.app.json	Added docker file		6 hours ago	
tsconfig.json	Added docker file		6 hours ago	
tsconfig.spec.json	Added docker file		6 hours ago	
tslint.json	Added docker file		6 hours ago	

About

No description, website, or topics provided.

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

vaibhtan

vtandonv Vaibhav Tandon

Languages

TypeScript 56.0%

HTML 21.3%

CSS 20.6%

JavaScript 1.9%

Dockertile 0.2%

Frontend Git Repo

vtandonv / CommunityHubBackend

Unwatch

1

Star

0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main

1 branch

0 tags

Go to file

Add file

Code

vtandonv Create README.md			bf5950e now	5 commits
.mvnw/wrapper	new commit		yesterday	
deploy-docker	Added .yml and inventory file		5 hours ago	
src	Added .yml and inventory file		5 hours ago	
.gitignore	new commit		yesterday	
Dockerfile	Added Dockerfile		yesterday	
README.md	Create README.md		now	
mvnw	new commit		yesterday	
mvnw.cmd	new commit		yesterday	
pom.xml	new commit		yesterday	

README.md

CommunityHubBackend

Techstack: Backend: Spring Boot Database: MySQL

This module provides backend functionality of connecting to the database from the front end via REST API and applying our business logic there

About

No description, website, or topics provided.

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

vaibhtan

vtandonv Vaibhav Tandon

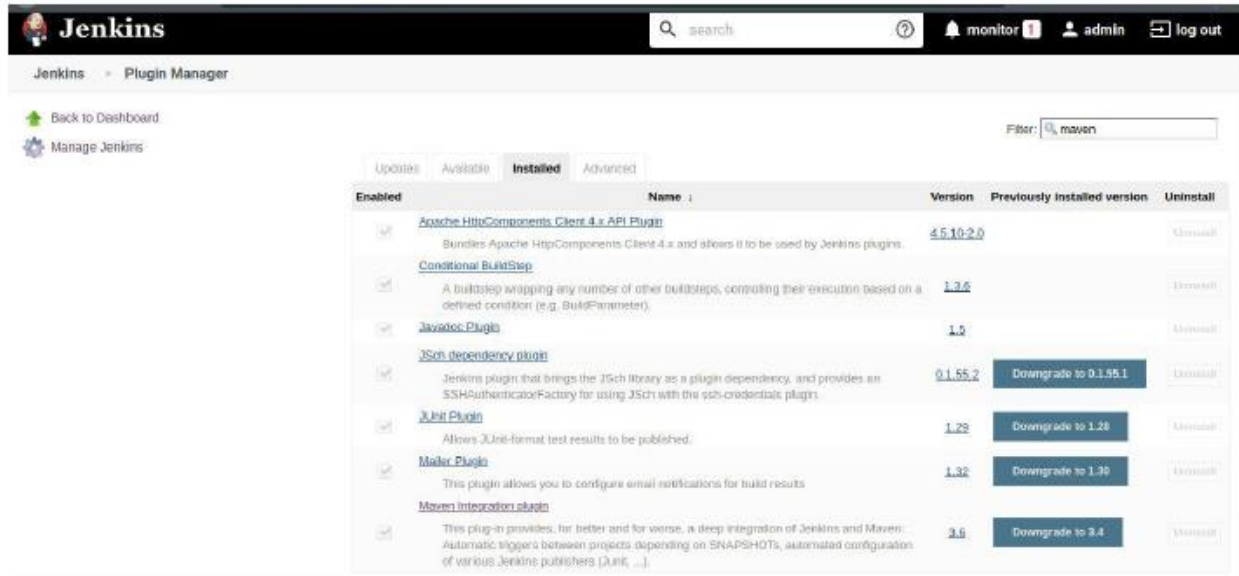
Languages

Java 99.0%

Dockertile 1.0%

Backend Git Repo

4.2.1 CI Pipeline

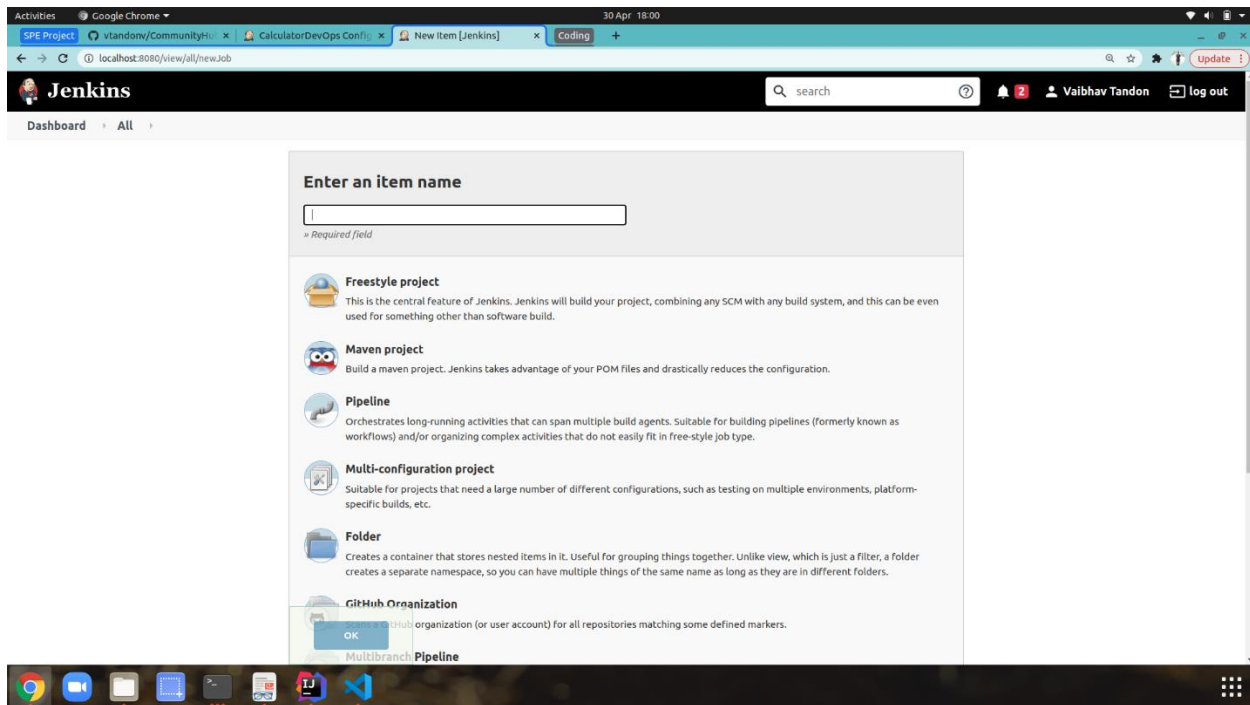


Setting up Jenkins: Adding the required plugins

We used Jenkins pipeline to clone these repositories from git.

But before that,

- First we add credentials (if any) of github repository,
- Now we create a pipeline job in Jenkins. Reference figure



Creating a new Jenkins Pipeline

```
stage('Git pull') {
    steps {
        git branch: 'main', url: 'https://github.com/vtandonv/CommunityHubBackend'
    }
}
```

Pipeline Syntax for Pulling Backend Project from Git

```
stage('Frontend Git pull') {
    steps {
        git branch: 'main', url: 'https://github.com/vtandonv/CommunityHubFrontend'
    }
}
```

Pipeline Syntax for Pulling Frontend Project from Git

4.3 Build

Maven is essentially a project management and comprehension tool and as such provides a way to help with managing builds, dependencies and distribution. To use maven your project must have a pom.xml example

<https://github.com/vtandonv/CommunityHubBackend/blob/main/pom.xml>

A Unit Test Case is a part of code, which ensures that another part of code (method) works as expected. To achieve the desired results quickly, a test framework is required. JUnit is a perfect unit test framework for Java programming language.

A formal written unit test case is characterized by a known input and an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post-condition.

Here in pom.xml we mention all the dependencies of the project and then using maven lifecycles we package the project in an executable jar file. For this to start make sure you have maven installed on your system.

```
sudo apt update && sudo apt install maven
```

This command looks to download the package lists from the repositories and "updates" them to get information on the newest versions of packages and their dependencies. The second command installs Maven.

4.4 Testing

Jenkins provides us with continuous integration which includes integrated testing, so every time we push code to github it integrates all the different modules of the project and tests their proper functioning.

JUnit is a unit testing framework for Java programming language. It plays a crucial role test-driven development, and is a family of unit testing frameworks collectively known as xUnit.

JUnit promotes the idea of "first testing then coding", which emphasizes on setting up the test data for a piece of code that can be tested first and then implemented. This approach is like "test a little, code a little, test a little, code a little." It increases the productivity of the programmer and the stability of program code, which in turn reduces the stress on the programmer and the time spent on debugging.

A Unit Test Case is a part of code, which ensures that another part of code (method) works as expected. To achieve the desired results quickly, a test framework is required. JUnit is a perfect unit test framework for Java programming language.

A formal written unit test case is characterized by a known input and an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post-condition.

For testing within java code we use Junit. Junit test cases are written in covering the scope of the project. Also Junit dependency is added in pom.xml for maven to resolve the dependency

```
SpringBootProjectApplicationTests.java x
1 package com.communityhub.springbootproject;
2 import com.communityhub.springbootproject.controller.CheckoutController;
3 import com.communityhub.springbootproject.controller.MessageController;
4 import org.junit.jupiter.api.Test;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.boot.test.context.SpringBootTest;
7
8 import static org.assertj.core.api.Assertions.assertThat;
9
10 @SpringBootTest
11 class SpringBootProjectApplicationTests {
12
13     @Autowired
14     private MessageController messageController;
15     @Autowired
16     private CheckoutController checkoutController;
17     @Test
18     void contextLoads() throws Exception {
19         assertThat(messageController).isNotNull();
20         assertThat(checkoutController).isNotNull();
21     }
22
23 }
24
```

Testing whether the controller returns NULL or not

```
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ spring-boot-project ---
[INFO] Building jar: F:\SPE Major Project\02-backend\spring-boot-project\target\spring-boot-project-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.4.0:repackage (repackage) @ spring-boot-project ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ spring-boot-project ---
[INFO] Installing F:\SPE Major Project\02-backend\spring-boot-project\target\spring-boot-project-0.0.1-SNAPSHOT.jar to C:\Us
t-project-0.0.1-SNAPSHOT.jar
[INFO] Installing F:\SPE Major Project\02-backend\spring-boot-project\pom.xml to C:\Users\HOME\.m2\repository\com\communityh
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 49.407 s
[INFO] Finished at: 2021-05-03T20:06:17+05:30
[INFO] -----
```

Build Success

4.4.1 CI Pipeline

```
stages {
    stage('Git pull') {
        steps {
            git branch: 'main', url: 'https://github.com/vtandonv/CommunityHubBackend'
        }
    }
    stage('Maven Build') {
        steps {
            script {
                sh 'mvn clean install'
            }
        }
    }
}
```

Pipeline Syntax for above Backend Build Stag

The screenshot displays the Jenkins Pipeline Community Hub interface. On the left is a sidebar with navigation links: Back to Dashboard, Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, Pipeline Syntax, Build History (with a trend icon), a search bar, and Atom feed links for all and failures. The main content area is titled 'Pipeline Community Hub' and includes a 'Recent Changes' section with a notepad icon. Below this is the 'Stage View' section, which shows a table of stage times for two stages: 'Git pull' and 'Maven Build'. The table indicates average stage times of 4s for Git pull and 3min 8s for Maven Build, with an average full run time of ~1min 50s. Two build entries are shown: #4 (Apr 29 16:29, 1 commit) and #2 (Apr 29 12:38, No Changes). The 'Permalinks' section at the bottom lists links for the last build, last stable build, last successful build, and last completed build, all for build #4, 3 min 52 sec ago.

	Git pull	Maven Build
Average stage times: (Average full run time: ~1min 50s)	4s	3min 8s
#4 Apr 29 16:29 1 commit	4s	3min 8s
#2 Apr 29 12:38 No Changes	4s	

Permalinks

- Last build (#4), 3 min 52 sec ago
- Last stable build (#4), 3 min 52 sec ago
- Last successful build (#4), 3 min 52 sec ago
- Last completed build (#4), 3 min 52 sec ago

Built Spring Boot Backend

4.5 DOCKER

Docker is a platform that uses Kernel level virtualization to deliver software in packages called containers. It enables us to separate our applications from infrastructure so that we can deliver software quickly.

Docker is a software platform for building applications based on *containers* — small and lightweight execution environments that make shared use of the operating system kernel but otherwise run in isolation from one another. While containers as a concept have been around for some time, Docker, an open source project launched in 2013, helped popularize the technology, and has helped drive the trend towards *containerization* and *microservices* in software development that has come to be known as [cloud-native development](#).

First, we need to install docker on our system. Once it is installed, we can run it using following command:

```
$ sudo systemctl start docker
```

Here, our aim is to build a docker image out of the jar file that will be created in build phase and push the latest image to Docker Hub, which will then be pulled by Ansible to deploy it to other machines. All this will be automated in Jenkins pipeline. To push the docker image, use the following command:

```
docker push <username>/<repository_name>:tagname
```

To pull the docker image, use the following command:

```
docker pull <docker_image_name:tag>
```

To run the docker image, use the following command:

```
docker run -it <image_name>
```

Also, make sure to add Jenkins to docker group, so that Jenkins can use docker for build docker image. This can be done by following command:

```
$ sudo usermod -aG docker Jenkins
```

Docker hub links related to the project are mentioned below :-

<https://hub.docker.com/repository/docker/vtandonv/communityhubfrontend>

<https://hub.docker.com/repository/docker/vtandonv/communityhubbackend>

```
stage('Docker Build to Image') {
    steps {
        script {
            imageName = docker.build 'vtandonv/communityhubbackend'
        }
    }
}

stage('Push Docker Image') {
    steps {
        script {
            docker.withRegistry('', 'docker-jenkins'){
                imageName.push()
            }
        }
    }
}
```

Pipeline Syntax for Building and pushing the docker image of backend

```
stage('Frontend Docker Build to image') {
    steps {
        script {
            app = docker.build 'vtandonv/communityhubfrontend'
        }
    }
}

stage('Push Frontend Docker Image') {
    steps {
        script {
            docker.withRegistry('', 'docker-jenkins'){
                app.push()
            }
        }
    }
}
```

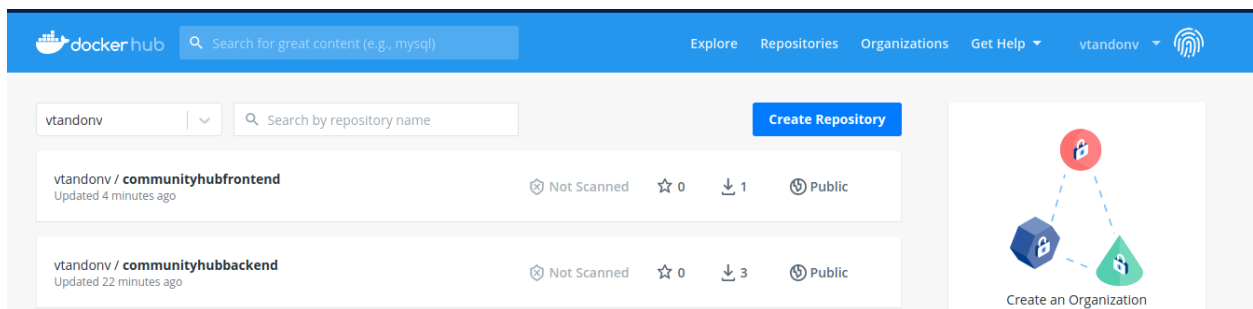
Pipeline Syntax for Building and pushing the docker image of frontend

Frontend Git pull	Frontend Docker Build to Image	Push Frontend Docker Image
24s	4min 39s	12min 17s
24s	4min 39s	12min 17s

Page | 25

```
vaibhav@vaibhav:~/Desktop$ docker images
REPOSITORY              TAG         IMAGE ID      CREATED        SIZE
vtandonv/communityhubfrontend  latest     1a7c16c8e57a  4 hours ago   1.6GB
vtandonv/communityhubbackend  latest     6d624d6cea58  4 hours ago   554MB
node                      latest     6817534de6bd  3 days ago    907MB
openjdk                   8          48ff6191b369  8 days ago    514MB
```

Successfully built the images



The screenshot shows the Docker Hub web interface. The top navigation bar includes the Docker Hub logo, a search bar, and links for Explore, Repositories, Organizations, Get Help, and a user profile for vtandonv. Below the navigation bar, there's a section for the vtandonv user with a search bar and a 'Create Repository' button. Two repositories are listed: 'vtandonv / communityhubfrontend' (updated 4 minutes ago) and 'vtandonv / communityhubbackend' (updated 22 minutes ago). Both repositories show 'Not Scanned' status, 0 stars, and are public. To the right, there's a 'Create an Organization' section with a diagram showing a central red circle connected to two green circles, one of which is a Docker logo.

Successfully pushed the images to docker hub

4.6 DEPLOY

A Deployment pipeline is the process of taking code from version control and making it readily available to users of your application in an automated fashion. When a team of developers are working on projects or features they need a reliable and efficient way to build, test and deploy their work.

We are using Ansible for the deployment. It is a system of configuration management written in Python programming language which uses a declarative markup language to describe configurations. It's used for automation of configuration and OS setup. It is often used to manage Linux-nodes, but Windows is also supported.

Inventory file is used to specify managed hosts.

```
localhost ansible_user=vaibhav
```

Inventory files for frontend and backend

YAML file is used to specify the commands that you want to run on managed hosts.

```
---
- name: Pull docker image of backend
  hosts: all
  tasks:
    - name: Pull image
      docker_image:
        name: vtandonv/communityhubbackend
        source: pull

    - name: Create back-end container
      docker_container:
        name: springboot-con1
        image: vtandonv/communityhubbackend
        ports:
          - "8090:8090"

---
- name: Pull docker image of frontend
  hosts: all
  tasks:
    - name: Pull image
      docker_image:
        name: vtandonv/communityhubfrontend
        source: pull

    - name: Create front-end container
      docker_container:
        name: angular-con1
        image: vtandonv/communityhubfrontend
        ports:
          - "4200:4200"
```

.yaml files for backend and frontend

4.6.1 Ansible setup

Ansible is open-source automation platform and simplest way to deploy applications. It allows us to write 'Playbooks' that are descriptions of the desired state of our systems. A playbook consists of one or more 'plays'. A play performs a series of tasks on the hosts, in the order specified by the play. These plays are expressed in YAML format in a text file. Managed hosts are listed in an 'inventory'. The inventory can be defined in a static text file, or dynamically determined by scripts that get information from external sources.

First, we need to install ansible on our system. Also, Ansible in the backend uses python to run various small codes and uses ssh in linux to connect to the hosts machines. So, we need to make sure that python(generally installed in linux) and ssh server are installed.

We can install them using following commands:

```
$ sudo apt install openssh-server
```

```
$ ssh-keygen -t rsa
```

```
$ sudo apt update
```

```
$ sudo apt install ansible
```

After the ansible is installed, we can verify it by checking its version using

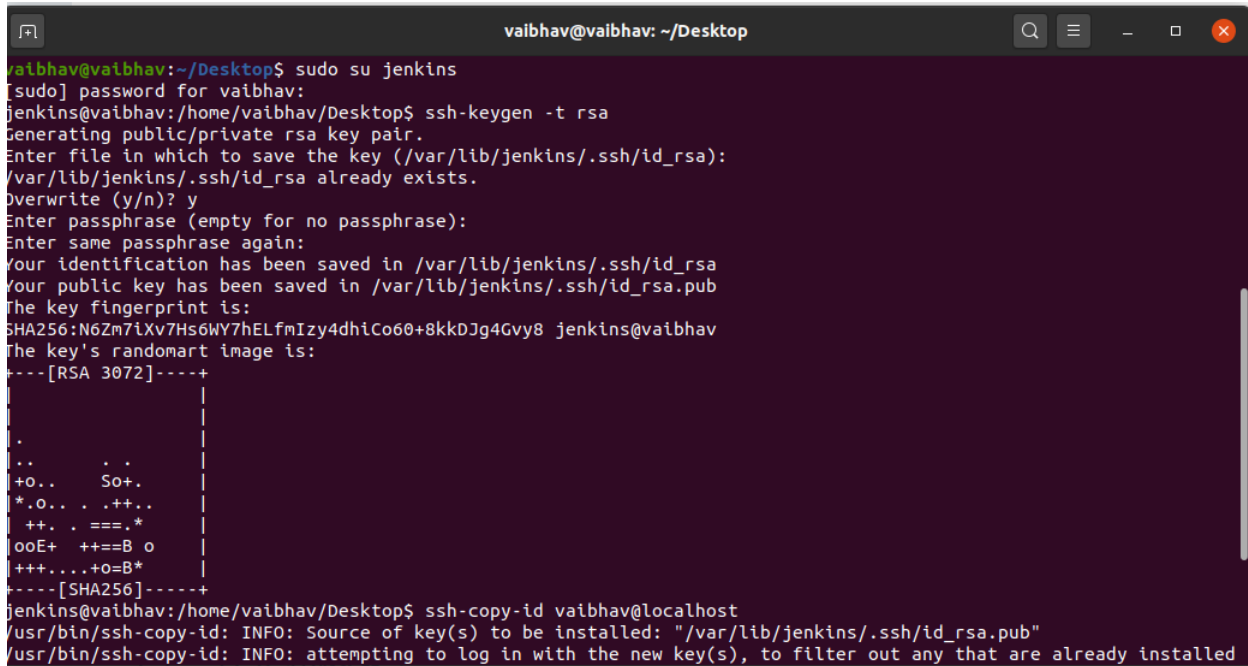
```
$ ansible --version
```

Also, since ansible would be configured within Jenkins pipeline, we need to ensure that Jenkins user can do ssh on localhost. This can be done using following command:

```
$ sudo su Jenkins
```

```
jenkins@vivek: $ ssh-keygen -t rsa
```

jenkins@vivek: \$ ssh-copy-id vivek@localhost



```
vaibhav@vaibhav: ~/Desktop
vaibhav@vaibhav:~/Desktop$ sudo su jenkins
[sudo] password for vaibhav:
jenkins@vaibhav:/home/vaibhav/Desktop$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/jenkins/.ssh/id_rsa):
/var/lib/jenkins/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/id_rsa
Your public key has been saved in /var/lib/jenkins/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:N6Zm7lXv7Hs6WY7hELfmIzy4dhiCo60+8kkDJg4Gvy8 jenkins@vaibhav
The key's randomart image is:
+---[RSA 3072]-----+
|
|..          ..
|+o..      So+
|*.O..    .++..
|++..    .===*
|ooE+  ++==B o
|+++....+O=B*
+---[SHA256]-----+
jenkins@vaibhav:/home/vaibhav/Desktop$ ssh-copy-id vaibhav@localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/jenkins/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
```

First Created new user and added him to sudo and docker group using following commands.

\$sudo adduser vivek2 /*for new user*/

\$sudo usermod -aG sudo vivek2 /* add user to sudo group*/

\$sudo usermod -aG docker vivek2 /* add user to docker group*/

First add jenkins user to a docker group. Then Created ssh keys in both jenkins and new user and copied ssh key of new user into jenkins user.

\$ ssh-copy-id vivek2@localhost or IP

After that check docker existence in python at both control node and host nodes, if docker is not available then install docker where needed using this command.

\$pip3 install docker

Some times ansible gives some error on python version-3 for some machines. There we need to use python 2.7 and install docker in it using following commands.

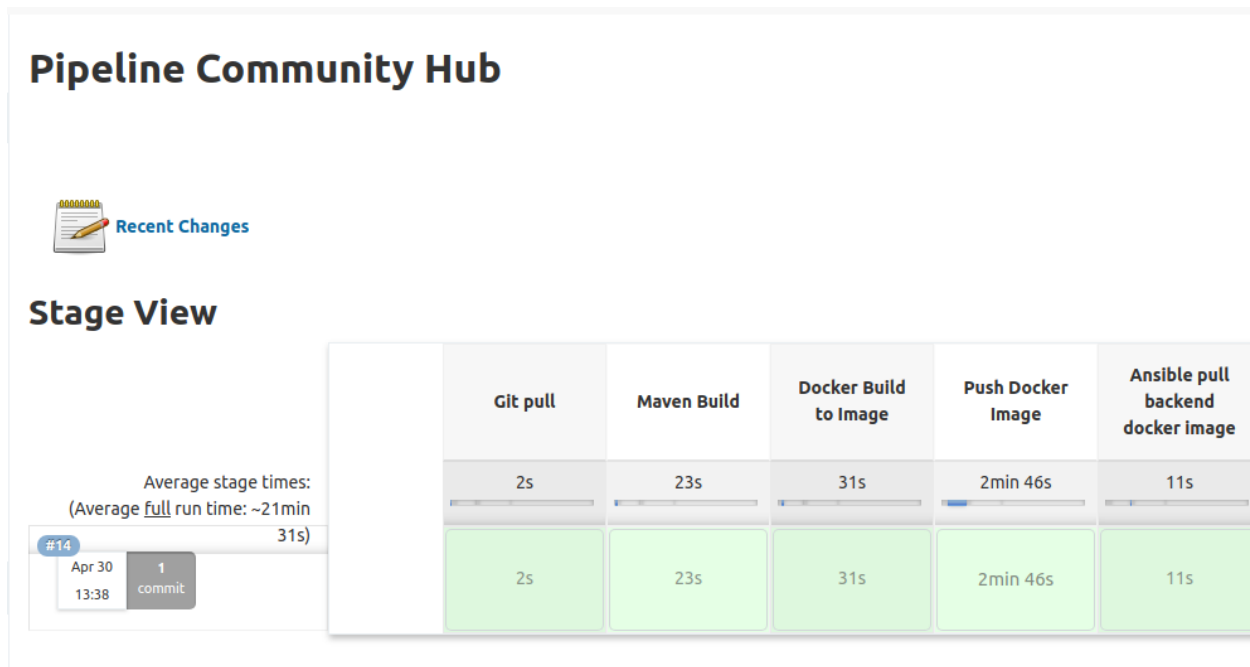
\$sudo -wget <https://bootstrap.pypa.io/pip/2.7/get-pip.py>

After Running entire jenkins pipeline docker image is available to run on managed hosts.

4.6.2 CI PIPELINE

```
stage('Ansible pull backend docker image') {  
  steps {  
    ansiblePlaybook becomeUser: null, colored: true, disableHostKeyChecking: true, installation: 'Ansible', inventory: 'deploy-docker/inventory',  
    playbook: 'deploy-docker/backend-deploy.yml', sudoUser: null  
  }  
}
```

Pipeline Syntax for deploying backend image



Backend image successfully deployed to Ansible host

```

stage('Ansible pull frontend docker image') {
  steps {
    ansiblePlaybook becomeUser: null, colored: true,
    disableHostKeyChecking: true, installation: 'Ansible', inventory
    playbook: 'deploy-docker/frontend-deploy.yml', sudoUser: null
  }
}

```

Pipeline Syntax for deploying frontend image

The screenshot shows the Jenkins Pipeline Community Hub interface. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, Pipeline Syntax, Build History, and Atom feed for all / Atom feed for failures. The main content area is titled 'Pipeline Community Hub' and features a 'Recent Changes' section. Below this is the 'Stage View' section, which displays a table of stage times for a pipeline. The table has 9 columns representing different stages: Git pull, Maven Build, Docker Build to Image, Push Docker Image, Ansible pull backend docker image, Frontend Git pull, Frontend Docker Build to Image, Push Frontend Docker Image, and Ansible pull frontend docker image. Each column shows the average stage time and a bar chart representing the time distribution. The 'Average stage times' section indicates an average full run time of ~21min and 31s. The 'Permalinks' section lists the last build (#14), last stable build (#14), last successful build (#14), and last completed build (#14), all 22 min ago.

Stage	Git pull	Maven Build	Docker Build to Image	Push Docker Image	Ansible pull backend docker image	Frontend Git pull	Frontend Docker Build to Image	Push Frontend Docker Image	Ansible pull frontend docker image
Average stage times	2s	23s	31s	2min 46s	11s	24s	4min 39s	12min 17s	12s
Full run time	2s	23s	31s	2min 46s	11s	24s	4min 39s	12min 17s	12s

Final DevOps Pipeline

4.7 Monitor

ELK stack makes the monitoring tool for any deployed software, it analyzes the logs and the same

The **ELK Stack** is a collection of three open-source products — Elasticsearch, Logstash, and Kibana. ELK stack provides centralized logging in order to identify problems with servers or applications. It allows you to search all the logs in a single place. It also helps to find issues in multiple servers by connecting logs during a specific time frame.

E stands for ElasticSearch: used for storing logs

L stands for LogStash : used for both shipping as well as processing and storing logs

K stands for Kibana: is a [visualization tool](#) (a web interface) which is hosted through Nginx or Apache

Elasticsearch is a NoSQL database. It is based on Lucene search engine, and it is built with RESTful APIS. It offers simple deployment, maximum reliability, and easy management. It also offers advanced queries to perform detail analysis and stores all the data centrally. It is helpful for executing a quick search of the documents.

Logstash is the data collection pipeline tool. It collects data inputs and feeds into the Elasticsearch. It gathers all types of data from the different source and makes it available for further use.

Logstash can unify data from disparate sources and normalize the data into your desired destinations. It allows you to cleanse and democratize all your data for analytics and visualization of use cases.

Kibana is a data visualization which completes the ELK stack. This tool is used for visualizing the Elasticsearch documents and helps developers to have a quick insight into it. Kibana dashboard offers various interactive diagrams, geospatial data, and graphs to visualize complex quires.

analysis can then be viewed on kibana dashboard. First try to generate log file using log4j library. Add necessary dependency in maven. Create Log4j2.xml file which specifies structure of logs.

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
  <Appenders>
    <Console name="ConsoleAppender" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n" />
    </Console>
    <File name="FileAppender" fileName="communityhubbackend.log" immediateFlush="false" append="true">
      <PatternLayout pattern="%d{yyy-MM-dd HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
    </File>
  </Appenders>
  <Loggers>
    <Root level="debug">
      <AppenderRef ref="ConsoleAppender" />
      <AppenderRef ref="FileAppender"/>
    </Root>
  </Loggers>
</Configuration>
```

Adding log4j2.xml file to our Spring Boot Project


```
2021-05-03 15:36:56.646 [main] INFO com.communityhub.springbootproject.SpringBootProjectApplication - Starting SpringBootProjectApplication using Java 11.0.6 on DESKTOP-KJ9TGK with PID 1472
2021-05-03 15:36:56.656 [main] DEBUG com.communityhub.springbootproject.SpringBootProjectApplication - Running with Spring Boot v2.4.0, Spring v5.3.1
2021-05-03 15:36:56.662 [main] INFO com.communityhub.springbootproject.SpringBootProjectApplication - No active profile set, falling back to default profiles: default
2021-05-03 15:36:56.663 [main] DEBUG org.springframework.boot.SpringApplication - Loading source class com.communityhub.springbootproject.SpringBootProjectApplication
2021-05-03 15:36:56.720 [main] DEBUG org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext - Refreshing org.springframework.boot.web.servlet.context.
2021-05-03 15:36:56.742 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'org.springframework.context.annotation.
2021-05-03 15:36:56.756 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'org.springframework.boot.autoconfigure.
2021-05-03 15:36:56.843 [main] DEBUG org.springframework.context.annotation.ClassPathBeanDefinitionScanner - Identified candidate component class: file [F:\SPE Major Project\02-backend\spring
2021-05-03 15:36:56.858 [main] DEBUG org.springframework.context.annotation.ClassPathBeanDefinitionScanner - Identified candidate component class: file [F:\SPE Major Project\02-backend\spring
2021-05-03 15:36:56.865 [main] DEBUG org.springframework.context.annotation.ClassPathBeanDefinitionScanner - Identified candidate component class: file [F:\SPE Major Project\02-backend\spring
2021-05-03 15:36:56.886 [main] DEBUG org.springframework.context.annotation.ClassPathBeanDefinitionScanner - Identified candidate component class: file [F:\SPE Major Project\02-backend\spring
2021-05-03 15:36:57.188 [main] DEBUG org.springframework.core.env.PropertySourcesPropertyResolver - Found key 'spring.jmx.enabled' in PropertySource 'configurationProperties' with value of ty
2021-05-03 15:36:57.189 [main] DEBUG org.springframework.core.env.PropertySourcesPropertyResolver - Found key 'spring.application.admin.enabled' in PropertySource 'configurationProperties' wi
2021-05-03 15:36:57.240 [main] DEBUG org.springframework.core.env.PropertySourcesPropertyResolver - Found key 'spring.jmx.enabled' in PropertySource 'configurationProperties' with value of ty
2021-05-03 15:36:57.245 [main] DEBUG org.springframework.core.env.PropertySourcesPropertyResolver - Found key 'spring.datasource.url' in PropertySource 'configurationProperties' with value of
2021-05-03 15:36:57.477 [main] DEBUG org.springframework.core.env.PropertySourcesPropertyResolver - Found key 'spring.jmx.enabled' in PropertySource 'configurationProperties' with value of ty
2021-05-03 15:36:57.484 [main] DEBUG org.springframework.core.env.PropertySourcesPropertyResolver - Found key 'spring.application.admin.enabled' in PropertySource 'configurationProperties' wi
2021-05-03 15:36:57.496 [main] DEBUG org.springframework.core.env.PropertySourcesPropertyResolver - Found key 'spring.jmx.enabled' in PropertySource 'configurationProperties' with value of ty
2021-05-03 15:36:57.538 [main] INFO org.springframework.data.repository.config.RepositoryConfigurationDelegate - Bootstrapping Spring Data JPA repositories in DEFERRED mode.
2021-05-03 15:36:57.547 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'org.springframework.boot.autoconfigure.
2021-05-03 15:36:57.547 [main] DEBUG org.springframework.boot.autoconfigure.AutoConfigurationPackages - @EnableAutoConfiguration was declared on a class in the package 'com.communityhub.sprin
2021-05-03 15:36:57.550 [main] DEBUG org.springframework.data.repository.config.RepositoryConfigurationDelegate - Scanning for JPA repositories in packages com.communityhub.springbootproject.
2021-05-03 15:36:57.563 [main] DEBUG org.springframework.data.repository.config.RepositoryComponentProvider - Identified candidate component class: file [F:\SPE Major Project\02-backend\spring
2021-05-03 15:36:57.564 [main] DEBUG org.springframework.data.repository.config.RepositoryComponentProvider - Identified candidate component class: file [F:\SPE Major Project\02-backend\spring
2021-05-03 15:36:57.565 [main] DEBUG org.springframework.data.repository.config.RepositoryComponentProvider - Identified candidate component class: file [F:\SPE Major Project\02-backend\spring
2021-05-03 15:36:57.565 [main] DEBUG org.springframework.data.repository.config.RepositoryComponentProvider - Identified candidate component class: file [F:\SPE Major Project\02-backend\spring
2021-05-03 15:36:57.611 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate component class: file [F:\SPE Major Project\02-b
2021-05-03 15:36:57.633 [main] DEBUG org.springframework.data.repository.config.RepositoryConfigurationDelegate - Registering deferred repository initialization listener.
2021-05-03 15:36:57.634 [main] INFO org.springframework.data.repository.config.RepositoryConfigurationDelegate - Finished Spring Data repository scanning in 82 ms. Found 4 JPA repository int
2021-05-03 15:36:57.906 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'propertySourcesPlaceholderConfigurer'
2021-05-03 15:36:57.916 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'entityManagerFactoryPostProcessor'
2021-05-03 15:36:57.921 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'org.springframework.context.event.inten
2021-05-03 15:36:57.930 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'preserveExceptionHandlerTargetClassPostP
```

Generated logs

Create deployment

What do you want to do with the Elastic Stack?


Let us know what your use case is and we'll help you get started with Elasticsearch, Kibana and the full Elastic Stack. [Learn more](#)



General purpose

Not sure what you want? Choose this option, and there'll be help along the way.


Select



Enterprise Search

Add a search experience to your website, applications, or search the apps you use at work.


Select



Observability

Use logs, metrics, and APM data to monitor and react to events in your environment.

✓ Selected



Security

Prevent, detect, and respond to threats with SIEM, endpoint security, and threat hunting.

Select

BUILD MY OWN

START WITH A USE CASE

Settings

Choose the cloud provider, region, and Elastic Stack version.

 Azure us Virginia (eastus2) v7.12.1

[Expand](#)

Name your deployment

You can always change this later.


login-signup-logs

[Customize](#)

[Create deployment](#)

Equivalent API request

Creating a new deployment



Trial - 14 days left

Cloud Deployments login-signup-logs

Deployments

- login-signup-logs
 - Edit
 - Elasticsearch
 - Snapshots
 - API console
- Kibana
- APM
- Enterprise Search
- Logs and metrics
- Activity
- Security
- Performance

Features

Support

login-signup-logs

Virginia (eastus2)

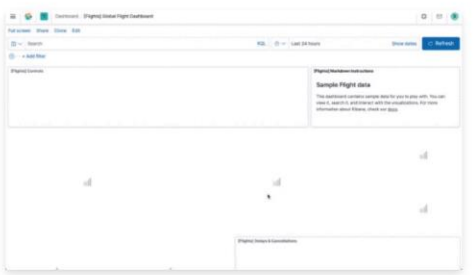
✓ Your deployment has been created.

Get started with your deployment

The next step is to ingest data and create visualizations in Kibana.

[Open Kibana](#)

Forgot to save your credentials? [Reset your deployment password](#)



Deployment name: login-signup-logs [Edit](#)

Deployment ID: 252f8f9

Deployment status: Healthy

[Open Kibana](#) [Manage](#)

Deployment Created

Okta Dashboard showing event trends by category. The interface includes a sidebar with navigation options (Dashboard, Directory, Applications, Security, Workflow, Reports) and a main content area displaying a table of events.

Search: Search...

vaibhtan1997@gmail...
okta-dev-08745849

Show event trends by category

Events: 196 [Download CSV](#)

Time	Actor	Event Info	Targets
May 03 15:57:27	Vaibhav Tandon (User)	User accessing Okta admin app success	Vaibhav Tandon (AppUser)
May 03 15:57:27	Okta Admin Console (PublicClientApp)	OIDC access token is granted success	(User) Access Token (access_token)
May 03 15:57:27	Vaibhav Tandon (User)	User single sign on to app success	Okta Admin Console (AppInstance) Vaibhav Tandon (AppUser)
May 03 15:57:27	Okta Admin Console (PublicClientApp)	OIDC id token is granted success	(User) ID Token (id_token)
May 03 15:57:26	Okta Admin Console (PublicClientApp)	OIDC authorization code request success	(User) Authorization Code (code)
May 03 15:57:19	Vaibhav Tandon (User)	Verify user identity success	
May 03 15:57:19	Vaibhav Tandon (User)	Evaluation of sign-on policy allow	Default Policy (PolicyEntity) Default Rule (PolicyRule)
May 03 15:57:19	Vaibhav Tandon (User)	User login to Okta	

Thanks for trying the Okta Starter plan. Upgrade to the Advanced plan to create more apps and get more Monthly Active Users.

[Upgrade](#)

Downloading Login/Signup logs from Okta Dashboard

Elasticsearch Home page. The interface includes a sidebar with navigation options (Home, Add data, Manage, Dev tools) and a main content area displaying various search and observability tools.

elastic Search Elastic

Press F11 to exit full screen

Home

[Add data](#) [Manage](#) [Dev tools](#)

Enterprise Search
Search everything →

Observability
Centralize & monitor →

Security
SIEM & Endpoint Security →

Kibana
Visualize & analyze →

Analyze data in dashboards.
Search and find insights.
Design pixel-perfect presentations.
Plot geographic data.
Model, predict, and detect.
Reveal patterns and relationships.

Ingest your data [Try our sample data](#)

Add data
Ingest data from popular apps and services.

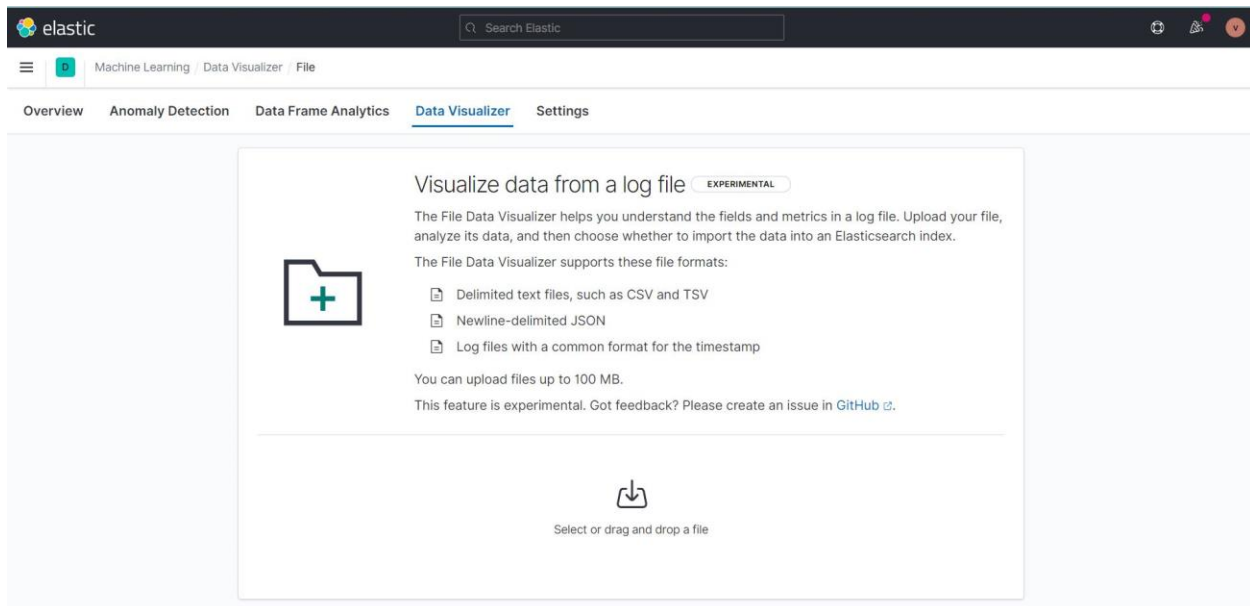
Add Elastic Agent
Add and manage your fleet of Elastic Agents and integrations.

Upload a file
Import your own CSV, NDJSON, or log file.

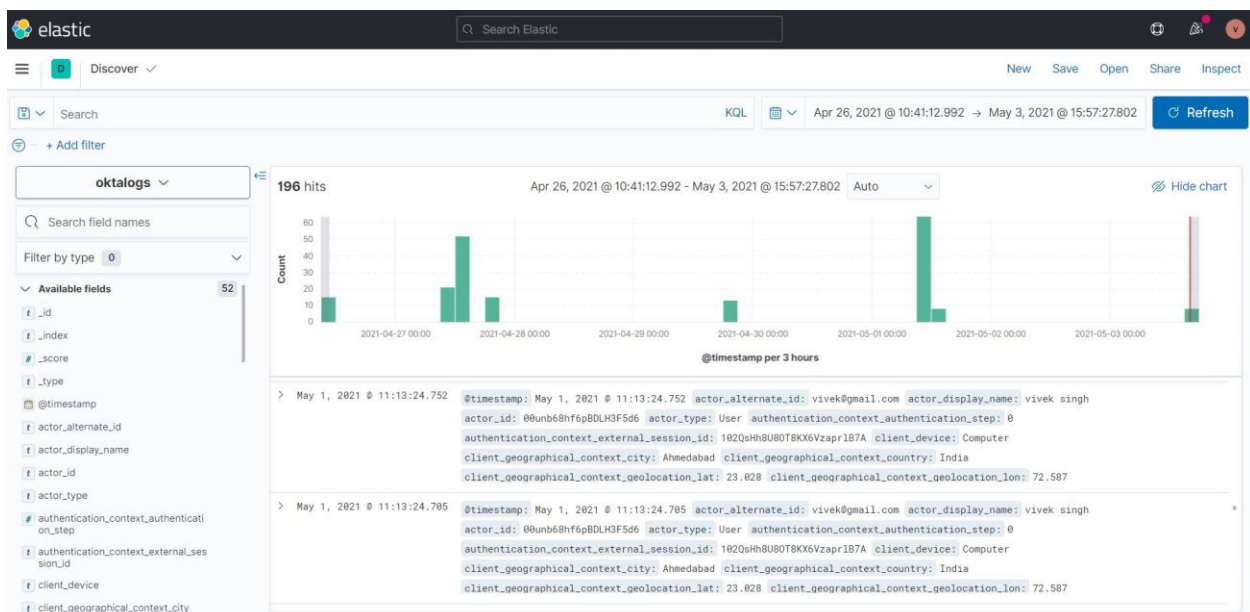
Manage your data

[Manage permissions](#) [Monitor the stack](#) [Back up and restore](#) [Manage index lifecycles](#)

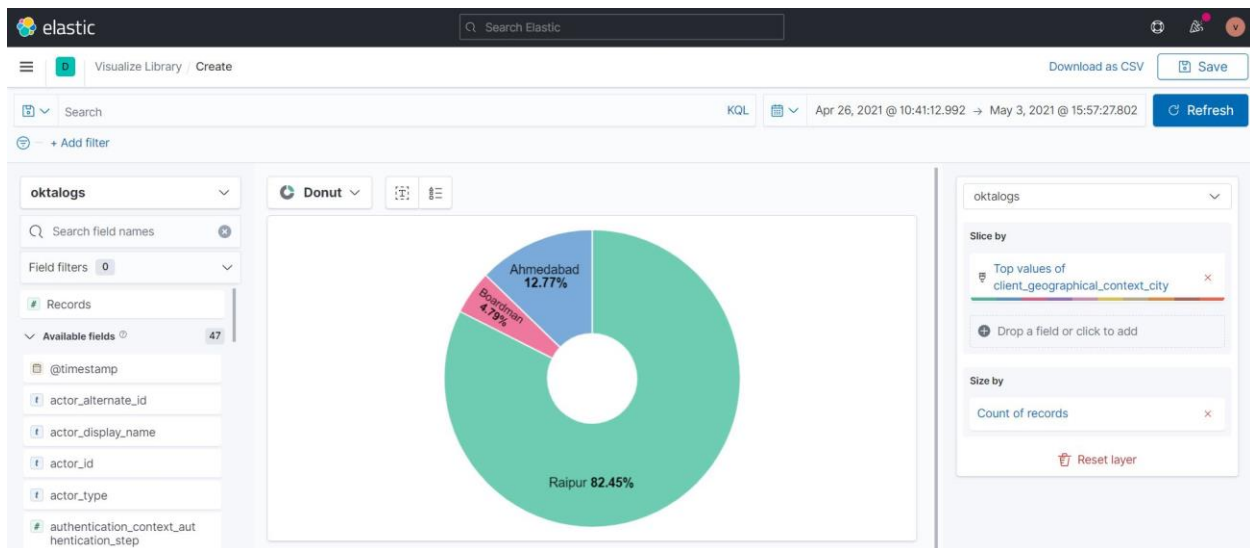
Kibana Opened



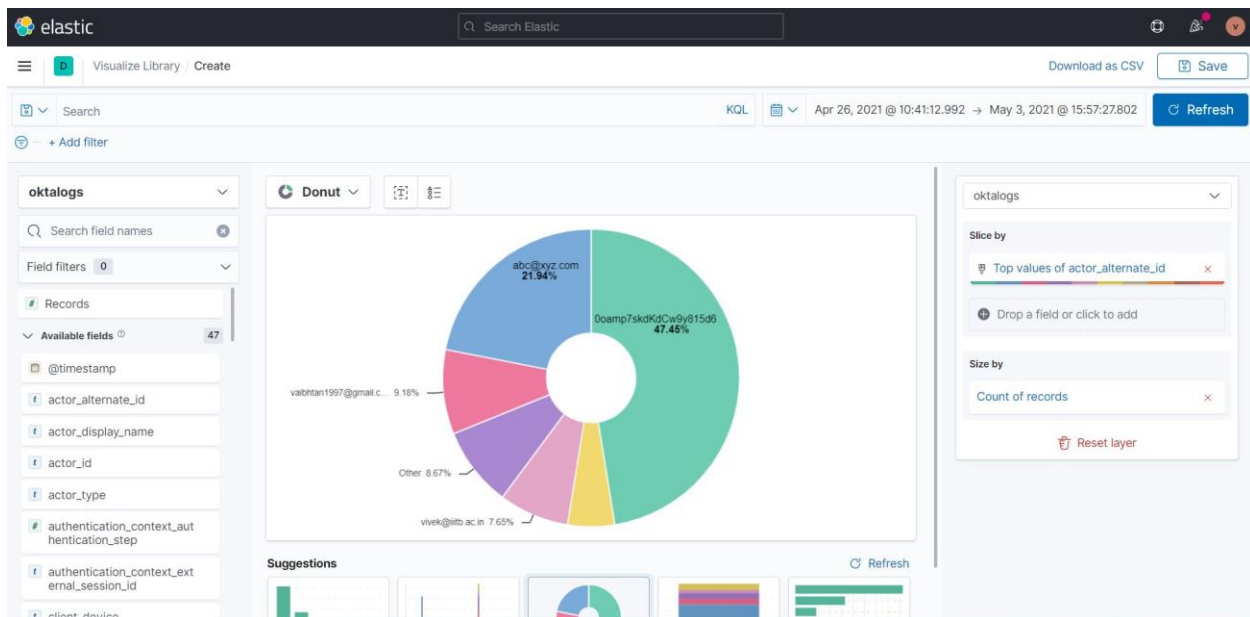
Upload the log file here



Viewing Index Pattern



Viewing Visualization of the location of login/signup users



Viewing Visualization of the number of logins/signups

4.8 Mysql

(A) MYSQL:

```
$ sudo apt install mysql-server
$ sudo mysql_secure_installation
```

```
Securing the MySQL server deployment. https://www.google.com/document/d/1tancKuenHnRkumCwWY1tuoP4C5WtVnQqE3...
Connecting to MySQL using a blank password.
VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?
Press y|Y for Yes, any other key for No: y
There are three levels of password validation policy:
LOW Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Please set the password for root here.
New password:
Re-enter new password:
Estimated strength of the password: 50
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.
```


Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n

... skipping.

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y

- Dropping test database...

Success.

- Removing privileges on test database...

Success.

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y

Success.

All done!

```
Activities Terminal 30 Apr 18:03
valbhav@valbhav: ~/Desktop

mysql> CREATE USER 'root'@'localhost' IDENTIFIED BY 'Superhuman1@';
Query OK, 0 rows affected (0.66 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.11 sec)

mysql> mysql -u root -p
?
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'mysql -u root -p' at line 1
mysql> exit
Bye
valbhav@valbhav:~/Desktop$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.23-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> create database hub;
Query OK, 1 row affected (0.19 sec)

mysql> use hub;
Database changed
mysql> source /media/valbhav/KINGSTON/Database/community_hub.sql
Query OK, 0 rows affected (0.00 sec)

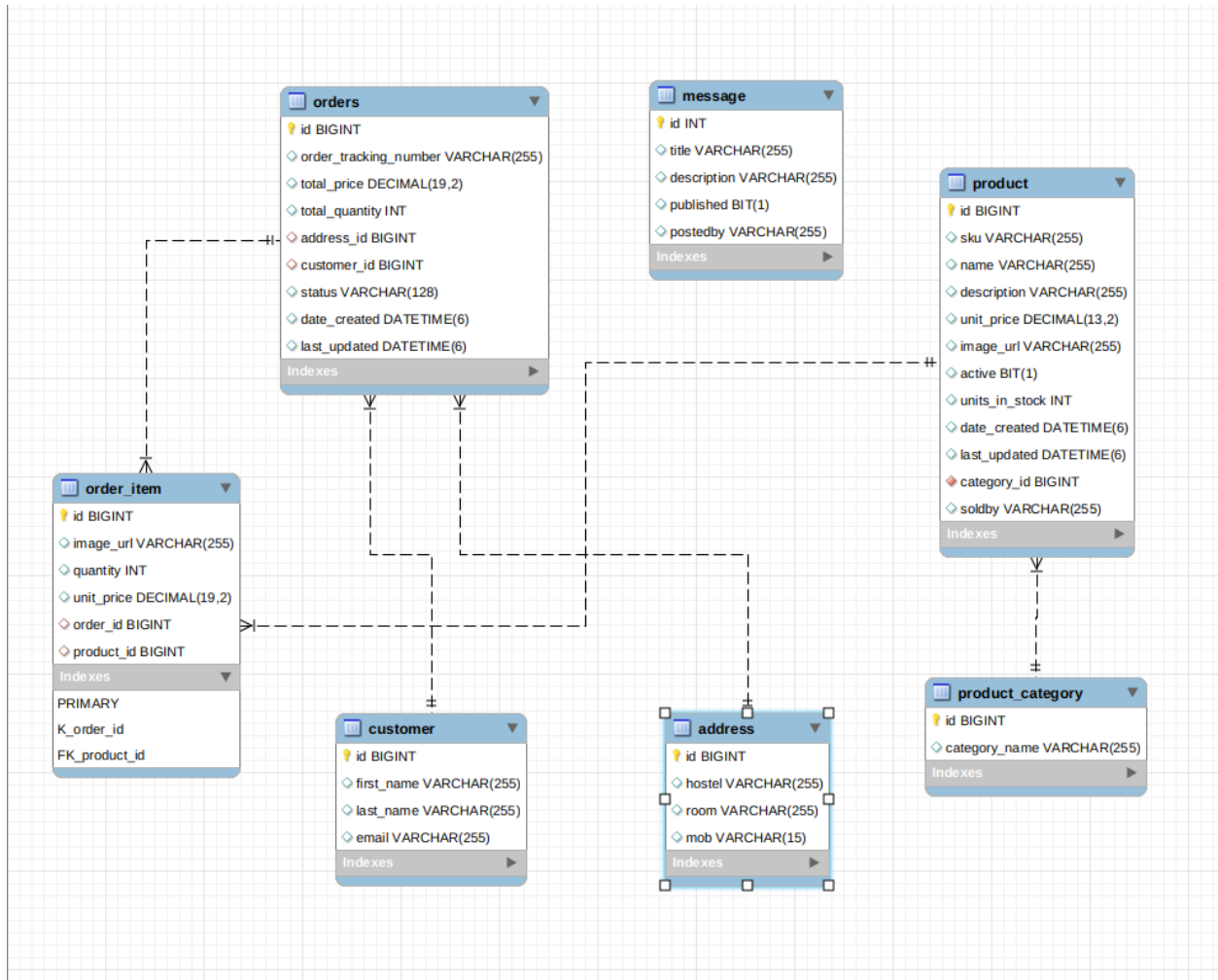
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
Activities Terminal 30 Apr 18:01
valbhav@valbhav: ~/Desktop

valbhav@valbhav:~/Desktop$ sudo apt update
[sudo] password for valbhav:
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:3 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:5 http://ln.archive.ubuntu.com/ubuntu focal InRelease
Hit:6 http://dl.google.com/linux/chrome/deb stable InRelease
Hit:7 https://download.docker.com/linux/ubuntu focal InRelease
Hit:8 https://packages.microsoft.com/repos/ms-teams stable InRelease
Hit:9 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
198 packages can be upgraded. Run 'apt list --upgradable' to see them.
valbhav@valbhav:~/Desktop$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libatoi libbgl-fast-perl libbgl-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinycsa
The following NEW packages will be installed:
  libatoi libbgl-fast-perl libbgl-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-core-8.0
0 to upgrade, 16 to newly install, 0 to remove and 198 not to upgrade.
Need to get 30.6 MB of archives.
After this operation, 249 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://security.ubuntu.com/ubuntu focal-security/main amd64 mysql-client-core-8.0 amd64 8.0.23-0ubuntu0.20.04.1 [4,215 kB]
Get:2 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 libatoi amd64 0.3.112-5 [7,184 B]
Get:3 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 libevent-core-2.1-7 amd64 2.1.11-stable-1 [89.1 kB]
Get:4 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 libevent-pthreads-2.1-7 amd64 2.1.11-stable-1 [7,372 B]
Get:5 https://ln.archive.ubuntu.com/ubuntu focal/main amd64 libmecab2 amd64 0.990-10build1 [223 kB]
Get:6 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 libfcgi-perl all 4.46-1 [186 kB]
Get:7 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 libfcgi-perl amd64 0.79-1 [33.1 kB]
Get:8 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 libbgl-fast-perl all 1:2.15-1 [16.5 kB]
Get:9 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 libhtml-template-perl all 2.97-1 [59.0 kB]
Get:10 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 mecab-utf8 amd64 0.990-10build1 [4,912 B]
Get:11 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 mecab-ipadic all 2.7.0-20070801+main-2.1 [6,714 kB]
Get:12 http://ln.archive.ubuntu.com/ubuntu focal/main amd64 mecab-ipadic-utf8 all 2.7.0-20070801+main-2.1 [4,389 B]
Get:13 https://security.ubuntu.com/ubuntu focal-security/main amd64 mysql-client-8.0 amd64 8.0.23-0ubuntu0.20.04.1 [22.0 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/main amd64 mysql-server-core-8.0 amd64 8.0.23-0ubuntu0.20.04.1 [17.7 MB]
Get:15 http://security.ubuntu.com/ubuntu focal-security/main amd64 mysql-server-8.0 amd64 8.0.23-0ubuntu0.20.04.1 [1,275 kB]
Get:16 https://security.ubuntu.com/ubuntu focal-security/main amd64 mysql-server all 8.0.23-0ubuntu0.20.04.1 [9,540 B]
Fetched 30.6 MB in 7s (4,644 kB/s)
Preconfiguring packages ...
```

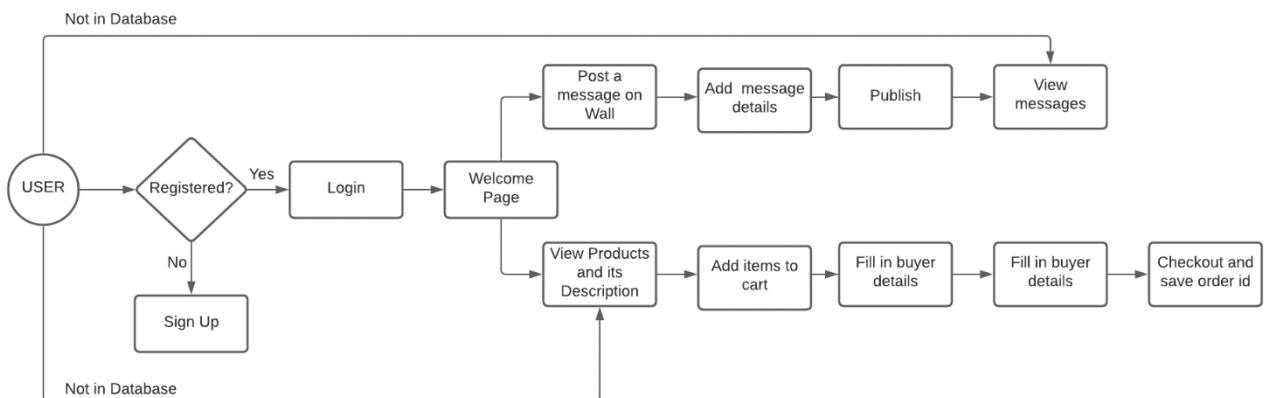



Database Schema of the project

5.EXPERIMENTAL SETUP

5.1 DESIGN DIAGRAM

Here the user if not present in the database will only be able to buy/borrow the items and will be able to view the messages on the Wall. But on the other hand if registered not only he/she will have the previous functionalities but also will be able to post messages on the Wall. The below DFD represents the same.



6. Results and Discussion

6.1 Project snapshots

6.1.1 Borrower details

community HUB

Search for items ... Search Login Wall USD 58.97 3

Books
Coffee Mugs
Mouse Pads
Luggage Tags

Student Details

First Name Vivek

Last Name Singh

Email vivek@iiitb.org

On Campus Location

Hostel Name Bhaskara

Room Number 766

Mobile Number 7894561230

6.1.2 Displaying messages on wall of non registered users

community HUB

Search for items ... Search Login Wall USD 58.97 3

Books
Coffee Mugs
Mouse Pads
Luggage Tags

Student Details

First Name Vivek

Last Name Singh

Email vivek@iiitb.org

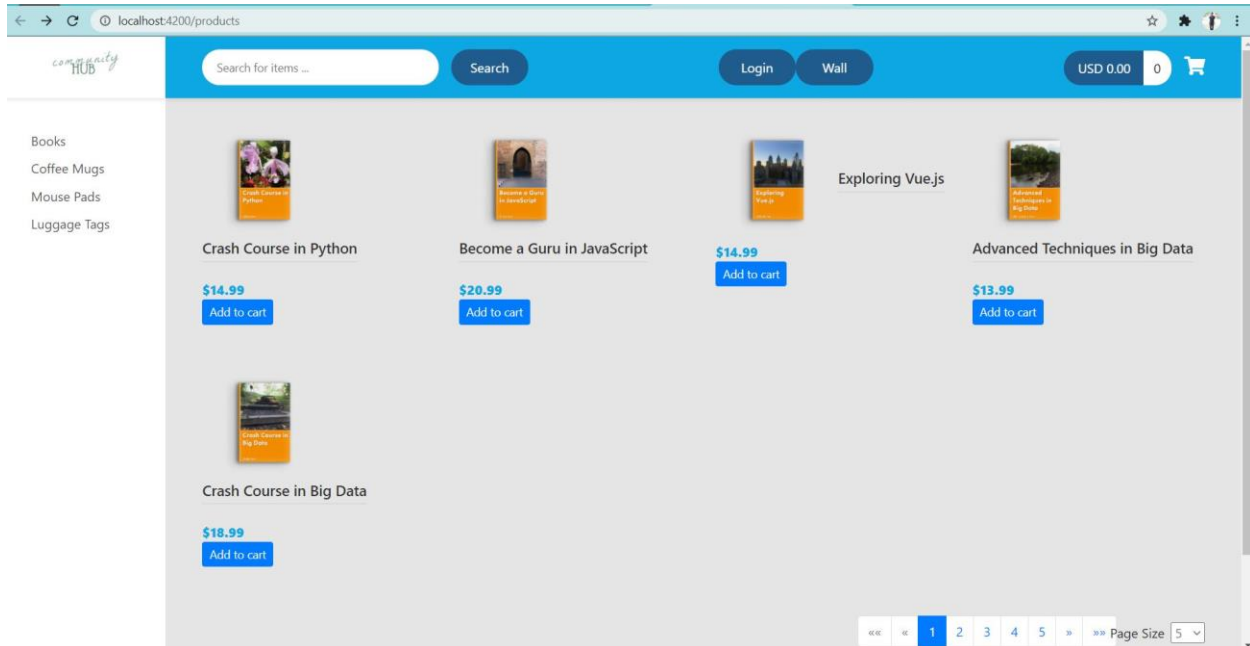
On Campus Location

Hostel Name Bhaskara

Room Number 766

Mobile Number 7894561230

6.1.3 Home Page



6.1.4 list of users on okta dashboard

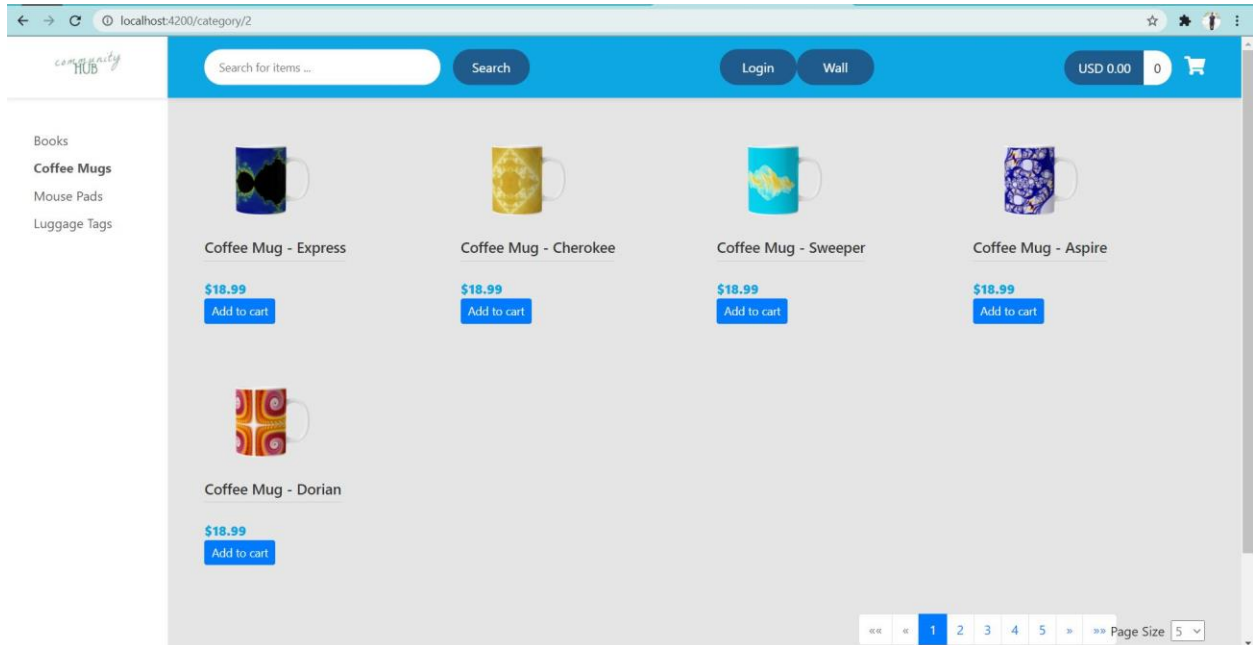
The screenshot shows the 'Okta' dashboard. The left sidebar contains navigation links: Dashboard, Directory, People (selected), Groups, Profile Editor, Directory Integrations, Self-Service, and Registration. The main content area is titled 'People' and includes buttons for 'Add person', 'Reset passwords', 'Reset multifactor', and 'More actions'. Below these is a search bar and a table listing users.

	Person & username	Primary email	Status
Everyone	6		
Onboarding			
Staged	0		
Pending user action	0		
Active			
Active	6		
Password reset	0		
Locked out	0		
Inactive			
Suspended	0		
Deactivated	0		

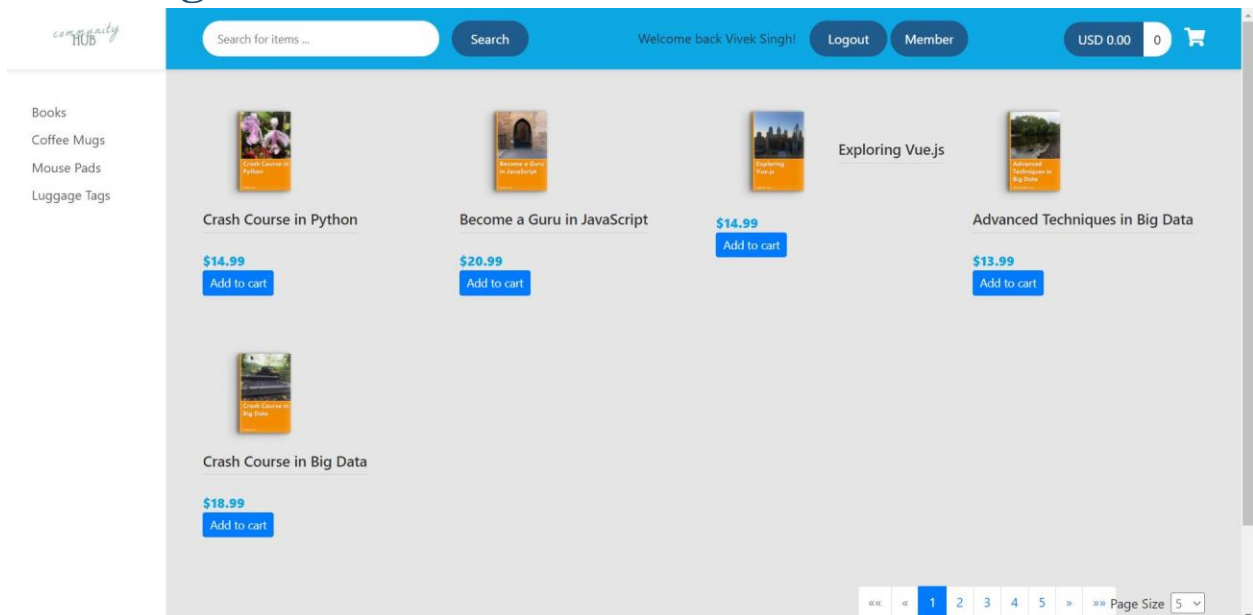
The table contains the following user entries:

- Onboarding**: Vivek Singh, vivek@gmail.com, Primary email: vivek@gmail.com, Status: Active.
- Staged**: Vivek Singh, vivek@iitb.ac.in, Primary email: vivek@iitb.ac.in, Status: Active.
- Pending user action**: Vaibhav Tandon, vaibhtan1997@gmail.com, Primary email: vaibhtan1997@gmail.com, Status: Active.
- Active**: Vaibhav Tandon, vtandonv@gmail.com, Primary email: vtandonv@gmail.com, Status: Active.
- Password reset**: Vaibhav Tandon, abc@xyz.com, Primary email: abc@xyz.com, Status: Active.
- Locked out**: Vivek Thakur, def@xyz.com, Primary email: def@xyz.com, Status: Active.

6.1.5 list of products of different category



6.1.6 Login success



6.1.7 Message list for registered user

community
HUB

Search for items ...

Search

Welcome back Vivek Singh!

Logout

Member

Books
Coffee Mugs
Mouse Pads
Luggage Tags

Search by title

Search

Messages List

New Test

Test 2

Test 3

New Test 4

Remove All

Message

Title: New Test

Description: Hello

Posted By: VT

Edit

6.1.8 Product description

community
HUB

Search for items ...

Search

Login

Wall

USD 0.00

0

Books
Coffee Mugs
Mouse Pads
Luggage Tags



Become a Guru in JavaScript

\$20.99

Add to cart

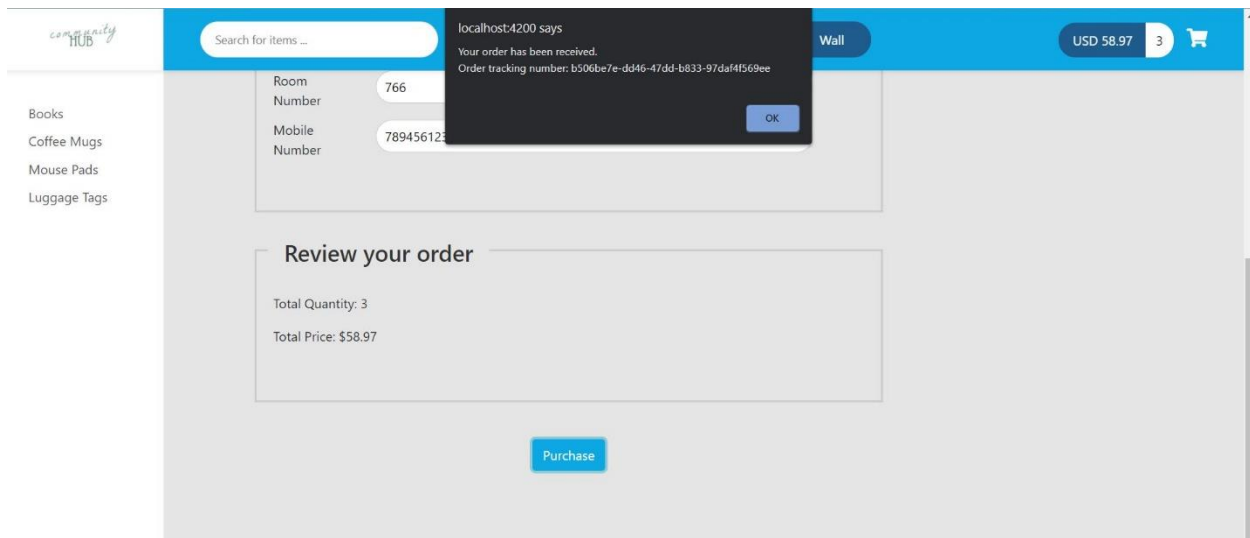
Description

Learn JavaScript at your own pace. The author explains how the technology works in easy-to-understand language. This book includes working examples that you can apply to your own projects. Purchase the book and get started today!

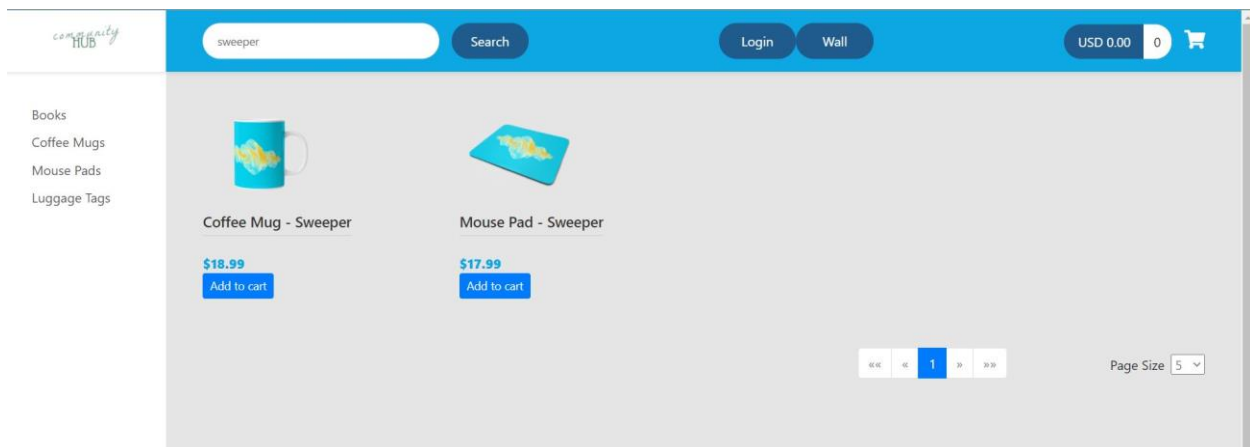
Sold By

MT2020150 Vaibhav Tandon

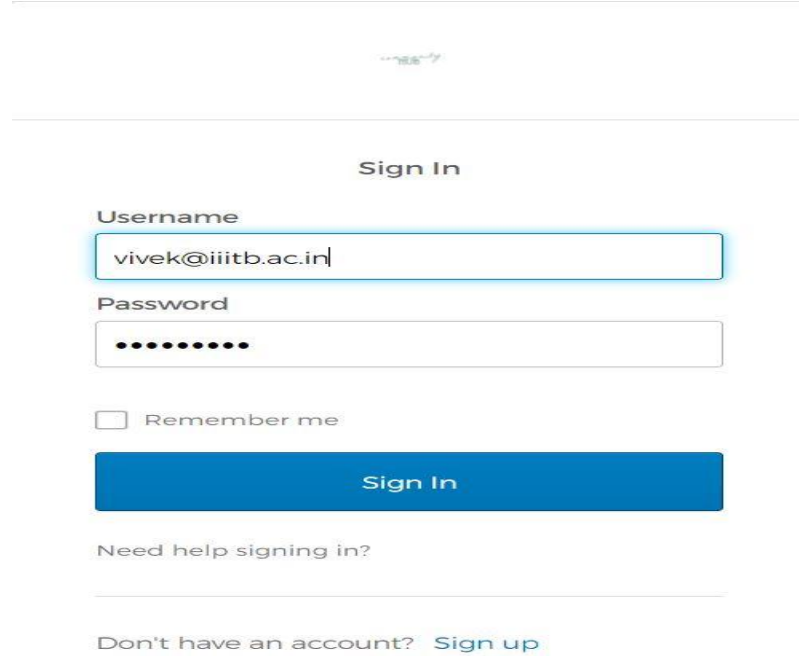
6.1.9 Order Review



6.1.10 Product Search results



6.1.11 Sign-in



A screenshot of a web application's sign-in page. At the top, there is a small, faint logo. Below it, the heading "Sign In" is centered. The form contains two input fields: "Username" with the text "vivek@iiitb.ac.in" and "Password" with masked characters. Below the password field is a checkbox labeled "Remember me". A prominent blue button with the text "Sign In" is positioned below the checkbox. Underneath the button is the text "Need help signing in?". At the bottom of the form, there is a link that says "Don't have an account? Sign up".

Sign In

Username
vivek@iiitb.ac.in

Password
.....

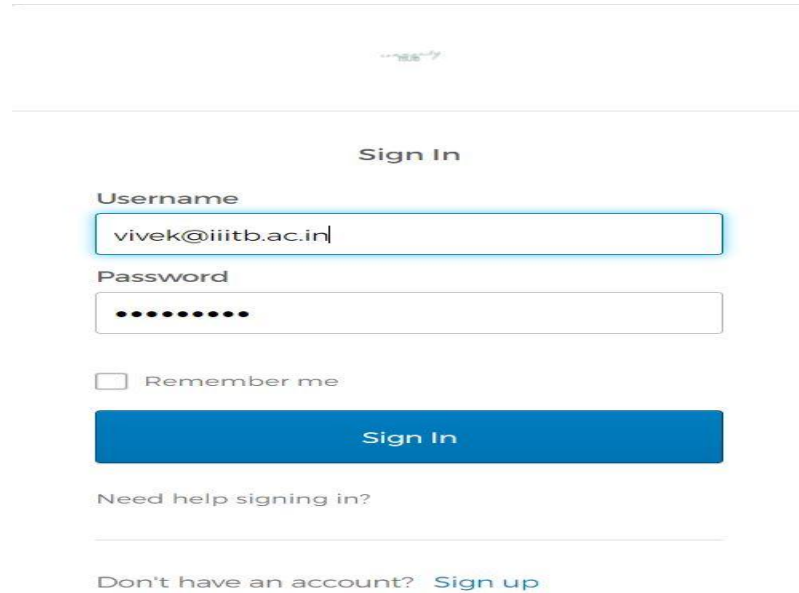
☐ Remember me

Sign In

Need help signing in?

Don't have an account? [Sign up](#)

6.1.12 Sign Up



A screenshot of a web application's sign-up page. It features the same layout as the sign-in page, including a small logo at the top and the heading "Sign In". The form fields are identical: "Username" (vivek@iiitb.ac.in) and "Password" (masked). It also includes a "Remember me" checkbox, a blue "Sign In" button, and the text "Need help signing in?". At the bottom, the link "Don't have an account? Sign up" is present.

Sign In

Username
vivek@iiitb.ac.in

Password
.....

☐ Remember me

Sign In

Need help signing in?

Don't have an account? [Sign up](#)

6.1.13 Updating Message

community
HUB

Search

Welcome back Vivek Singh!

Logout

Member

Books

Coffee Mugs

Mouse Pads

Luggage Tags

Message

Title

New Test

Description

Hello

Posted By

Vaibhav

Delete

Update

This message was updated successfully!

6.1.14 Product Cart

community
HUB

Search

Login

Wall

USD 58.97



3

Books

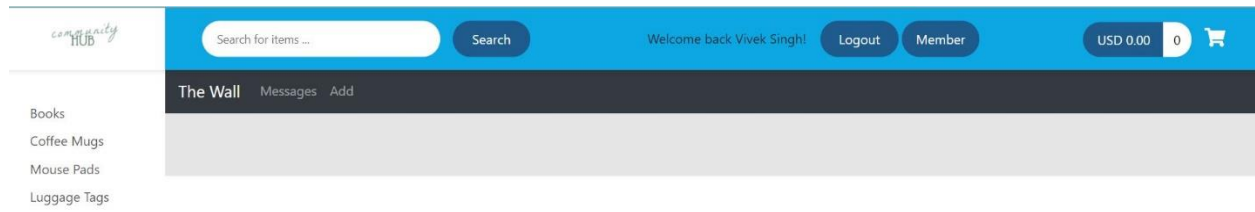
Coffee Mugs

Mouse Pads

Luggage Tags

Product Image	Product Detail	
	Become a Guru in JavaScript \$20.99	Quantity: <div><div>+</div><div>1</div><div>-</div></div> <div>Remove</div> <div>Subtotal: \$20.99</div>
	Coffee Mug - Express \$18.99	Quantity: <div><div>+</div><div>2</div><div>-</div></div> <div>Remove</div> <div>Subtotal: \$37.98</div>
		<div>Total Quantity: 3</div> <div>Total Price: \$58.97</div> <div>Checkout</div>

6.1.15 Wall for Registered User



7.Scope for Future Work

7.1 End to End Communication

In future Modification We would like to Add End to End Communication between the Borrower and the Lender.

7.2 User Rewards

In future Modification We would like to Add the feature of awarding rewards to users who are willing to help the community by providing useful information and by sharing their vital resources.

8. CONCLUSION

We have Successfully build a social media platform for social media users who want to share their resources and vital information along with their community to help the Community Grow through connectivity with each other.

The Devops tools that we have used are : GitHub,Jenkins,Docker,Ansible. These Tools are integrated using Jenkins.The enitre pipeline has been automated. I have automated the entire Software Development Life Cycle with the help of DevOps Tools. This makes the life of Developers and Operations team easy as Developers can now focus on adding new features without worrying about the failures in deployment environment. This can be highly beneficial to large companies where commits are made every single minute or seconds. It would have been herculean task to manually build, test, validate and deploy commits which occurs so frequently. DevOps automates the complete process of Integration, Delivery, Deployment which allows a software Company to focus more on the business logic, as the developed code of the newer version is sent to end user as soon as it passes the automated build and test stages.

9. References

1. <https://git-scm.com/doc>
2. <https://maven.apache.org/guides/index.html>
3. <https://junit.org/junit4/javadoc/latest/index.html>
4. <https://www.jenkins.io/doc/>
5. <https://docs.docker.com/>
6. <https://www.jenkins.io/doc/developer/guides/>
7. <https://docs.ansible.com/>
8. https://docs.ansible.com/ansible/2.8/modules/list_of_all_modules.html
9. <https://logz.io/learn/complete-guide-elk-stack/#elasticsearch>
10. <https://logz.io/learn/complete-guide-elk-stack/#logstash>
11. <https://logz.io/learn/complete-guide-elk-stack/#kibana>

