International Institute of Information Technology, Bangalore

Software Production Engineering Final Project

Bidding Platform

In the Guidance of: Prof. B. Thangaraju

Teaching Assistant: Rohan Mehta

Abhishek Garg	K.V.Lasya	Nitesh Jain
MT2020021	IMT2017023	MT2020118

1. Table of Contents

2. Abstract

A marketplace auction platform is exactly what it sounds like: auctioneers list their catalogues in an online auction marketplace right alongside their competitors. The competitors interested in the product, bid for the product with the amount they are willing to pay for that product. The owner gets to know the bids placed on the product and chooses the one who bids with the maximum amount. The online bidding platform enables the product to be sold worldwide without being physically present at the local.

The current existing structure doesn't provide an efficient platform for this bidding platform .So we are building a website which will provide a platform to share complete details which will be useful to others to bid for products in live scenario.We are also using DevOps tools to build a high quality website in less time.Our online auction platform i.e. Bidding Application lets your customers sell their product by auction, an interested person will bid on the available products, and the winner will get the product. It has been deployed as iaas or developed further to meet your specific business requirements.

The architecture of our project demands three layers.

- Front end
- Middle layer
- Back end

The front end of the project is handled by "React" Framework.The middle layer is built on "SpringBoot" Framework and communicates with mysql database.The back end is swiftly handled by "MySQL", using "Azure database" for interaction between back end and middle layer.

3. Introduction

3.1. Importance of the problem :

This project sets out to solve the problem of taking the real-time auction to an online platform to enable easy functioning of auction without being actually present at the local Place. Everyone nowadays prefers to enjoy the pleasure of staying at home. The technology ensures that the pleasure is not disturbed. The online bidding platform makes the user bid for any product worldwide and get it at his doorsteps without stepping out of his home. It's also beneficial for the owner of the product as his product gets a worldwide exposure rather than only local people.

3.2. Work plan :

The UI of the application enables the user to register itself on the platform to either sell or buy products of its wish. Once registered, the user is capable of putting ads for the product that he wishes to sell. The product can be anything such as antiques, used electronics, paintings etc. For putting an ad, the parameters required are product name, description of the product and minimum amount for bid.

Once the ad has been posted it displays on the products list page with all other products. The user can select the product of his wish from the list of products in front of him. By clicking on the product, the user is able to get all the details about the product as well as the owner of the product. By clicking on the bid button, the user enters the page where he can fill in the amount he is willing to pay. The amount should be obviously greater than the minimum bid amount fixed by the owner else the bid won't be set showing an alert regarding the same.

Once the bid has been placed, the bid is now visible in the product details dashboard. The user has the option to select for the bid he wishes, it's obvious that the user opts for the bid with a maximum bid amount. Once selected, the product becomes inactive and thus no further bids can be placed on the product.

3.3. What is DevOps :

1. DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.
2. DevOps is also characterized by operations staff making use of many of the same techniques/tools as developers for their systems work.
3. DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

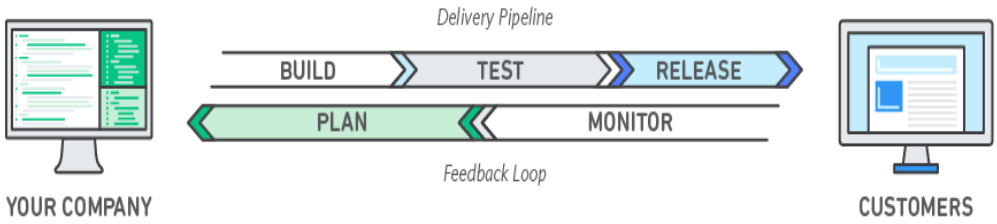

Fig-1 : Devops model

3.4. Why DevOps :

We plan to build this project in a growing way. The design has ideas and services that work independently. Given the complexity of the project, it is impossible for any of us to create and test the entire code manually every time we make a small change.And since, the three of us work from different locations, the automatic pipeline will not only make our job easier, and make it more efficient. The amount of communication that must take place between us will decrease. DevOps helps us focus on key aspects of the project, improving efficiency, stability and security. There is a small range of manual errors as well. And since we plan to build this product at some point, continuous delivery makes it easier. Also, monitoring allows us to better understand usage and help us improve the application.

4. System Configuration

4.1 Operating System :

Linux Mint 19.3 Tricia and Ubuntu 18.04.04 Bionic Beaver.

4.2 CPU and RAM :

Quad-core processor and RAM 8 GB (preferable 16 GB)

4.3 Language :

Java (SpringBoot framework),React JS,MySQL

4.4 Database :

Azure Database for MySQL server

4.5 Building Tools :

Maven to build java application,yarn to build react application

4.6 Devops Tools :

1. Github: version control system
2. Jenkins: CI/CD pipeline
3. Maven: Build tool
4. JUnit: Testing
5. Ansible: Configuration management and infrastructure as code
6. ELK: Continuous Monitoring
7. Docker, Multipass: Deployment/ containerisation

5. Software Development Life Cycle (SDLC)

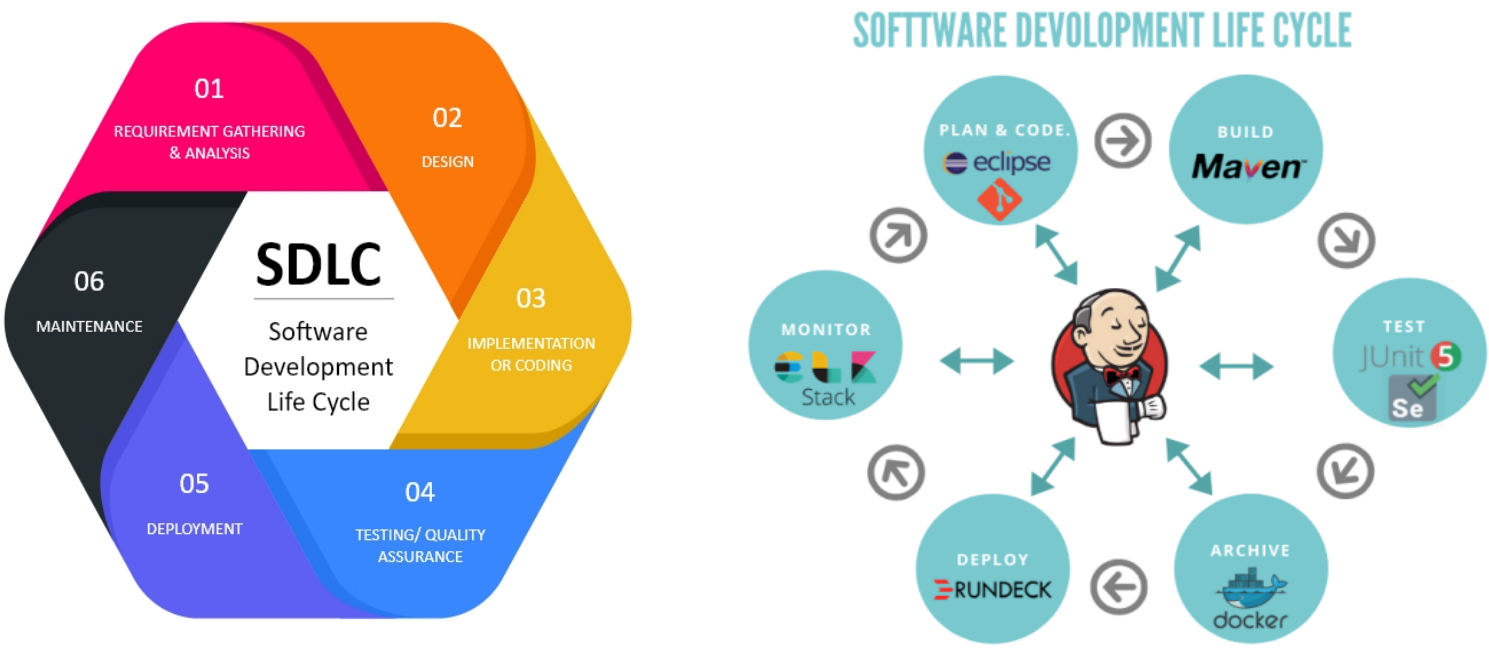

Fig-2 : Bidding Platform SDLC

5.1. Source Code Management (SCM):

(Link to repository : https://github.com/niteshjan/Bidding-Platform.git)

SCMs are used to give versions/revisions to the program. Each type is given a time stamp and includes the person responsible for the change. Even different versions can be compared and integrated with other types. That is why SCM is also called Version Control, Revision Control or Source Control.

In order to achieve SCM, we need to create a github repository on github.com by specifying the name and description about the project. This creates an empty repository on the github. We can also add a readme file in the repository that contains some information about the project.

After creating an empty repo on github we need to clone an empty project to the local system. This would create a directory in the name of the project in which we can add the files of our project. In this directory, add all the files of the project.

Now add these files to the staging area.

$ git add .

Then commit those changes so that the files would get added to the local repo.

$ git commit -m “message”

Now in order to add these files to the github repo, we need to push the files to the repo.

$ git push origin master

We have now successfully added our project to the github that enable the other users to use the same project and made required modifications to the project by git pull.

After pushing the files to github the github would look as below.

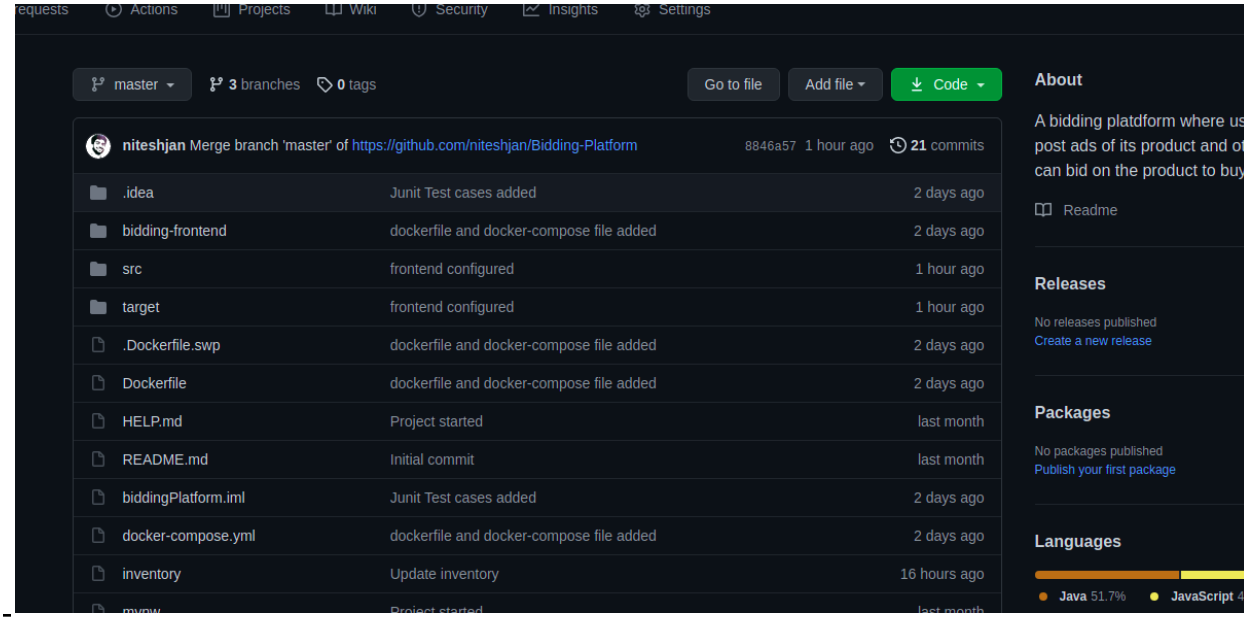

Fig-3 : GitHub repository

5.2. Build :

Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

The primary goal of Maven is to provide developer with the following −

- A comprehensive model for projects, which is reusable, maintainable, and easier to comprehend.
- Plugins or tools that interact with this declarative model.

Maven project structure and contents are declared in an xml file, pom.xml, referred to as Project Object Model (POM), which is the fundamental unit of the entire Maven system.

The maven enables the dependency management and helps the user to build the project to create the JAR file that can be transported to other systems to enable it to run to other systems also.The maven maintains a pom.xml file that contains all the dependencies that are required for the project to run.

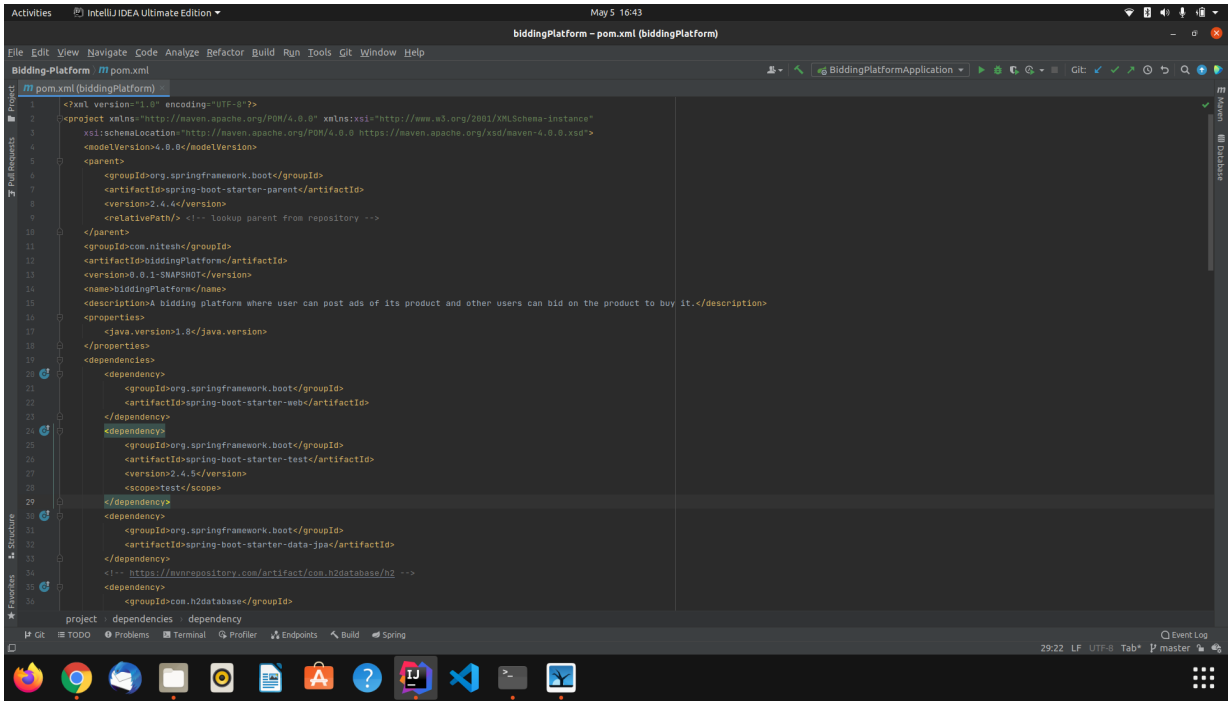

Fig-4 : Pom.xml file

The above is the pom.xml file that contains all the dependencies that are required for the project. It also maintains the proper version of the dependency to ensure proper functioning of the applications.

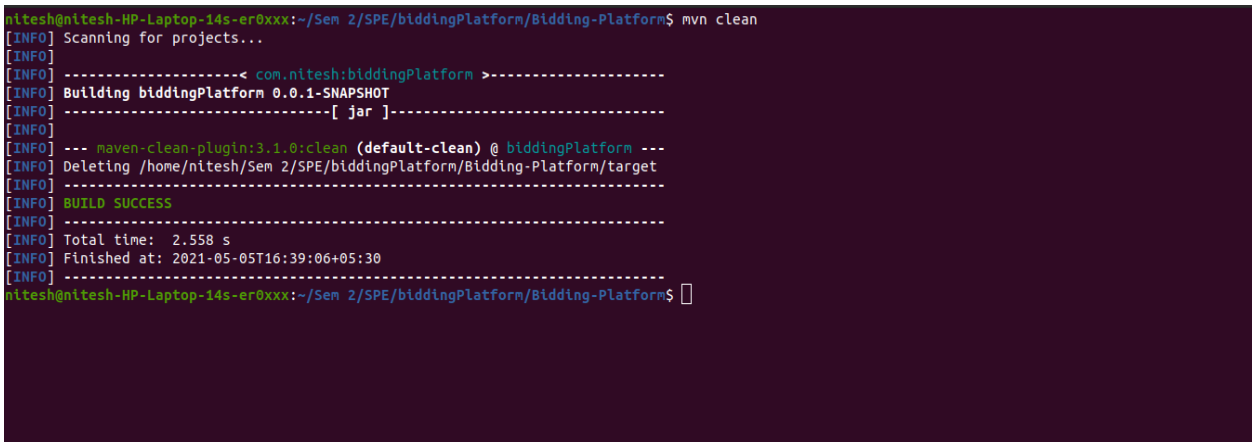

Fig-5 : Building the project

The clean life cycle has only one phase named clean that is automatically bound to the only goal of the plugin with the same name. This goal can, therefore, be executed with the command mvn clean. This command cleans the maven project by deleting the target directory.

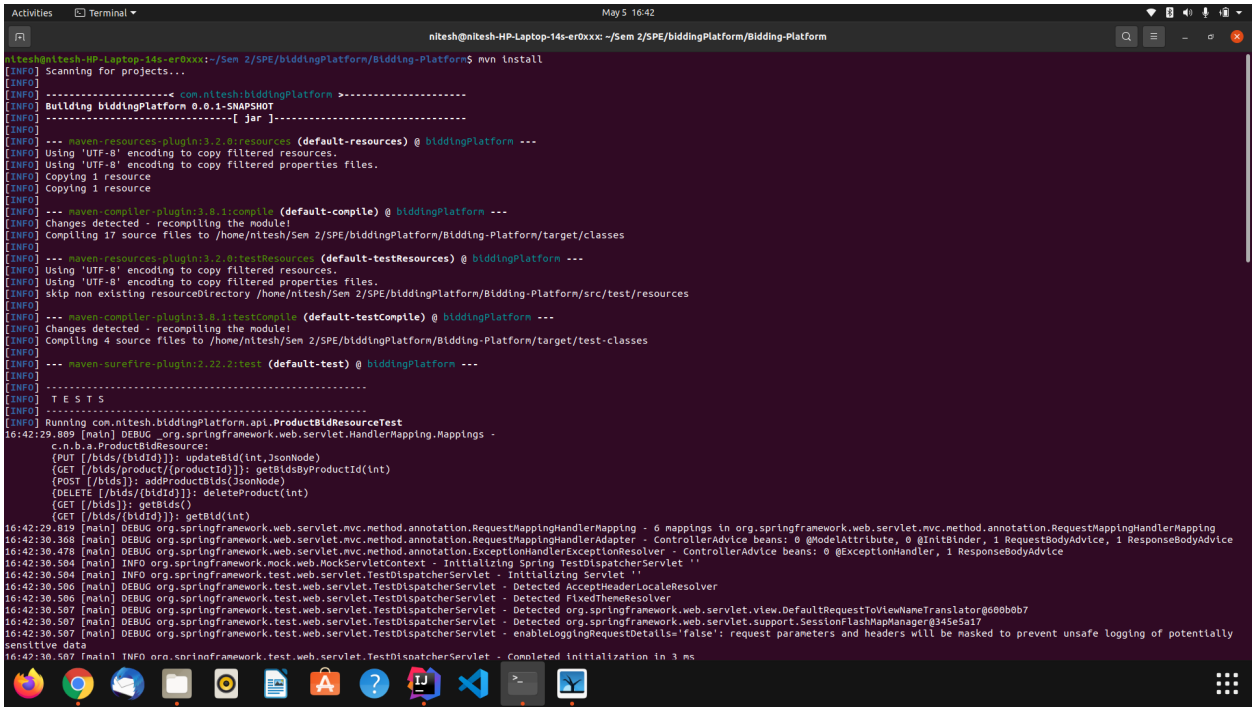

Fig-6 : Testing our test cases (mvn install)

This command builds the maven project and installs the project files (JAR, WAR, pom.xml, etc) to the local repository. On a mvn install , it frames a dependency tree based on the project configuration pom. xml on all the sub projects under the super pom. xml (the root POM) and downloads/compiles all the needed components in a directory called.

5.3. Testing :

A unit test is a piece of code written by a developer that executes a specific functionality in the code to be tested and asserts a certain behavior or state. The percentage of code which is tested by unit tests is typically called test coverage. A unit test targets a small unit of code, e.g., a method or a class. External dependencies should be removed from unit tests, e.g., by replacing the dependency with a test implementation or a (mock) object created by a test framework. Unit tests are not suitable for testing complex user interfaces or component interaction. For this, you should develop integration tests.

5.3.1. Junit :

It is an open-source framework, which is used for writing test cases and running test cases. Provides annotations to identify test methods, assertions for testing expected results, etc. It is useful for java developers to write and run repeatable tests. Erich Gamma and Kent Beck initially developed it. It is an instance of xUnit architecture. As the name implies, it is used for unit testing of a small chunk of code.Developers who are following test-driven methodology must write and execute unit tests first before any code.

A JUnit test is a method contained in a class which is only used for testing. This is called a Test class. To define that a certain method is a test method, annotate it with the @Test annotation.This method executes the code under test. You use an assert method, provided by JUnit or another assert framework, to check an expected result versus the actual result. These calls are typically called asserts or assert statements.Assert statements typically allow to define messages which are shown if the test fails. You should provide meaningful messages here to make it easier for the user to identify and fix the problem. This is especially true if someone looks at the problem, who did not write the code under test or the test code.

Once you are done with code, you should execute all tests, and it should pass. Every time any code is added, you need to re-execute all test cases and make sure nothing is broken.

Test and Build Result:

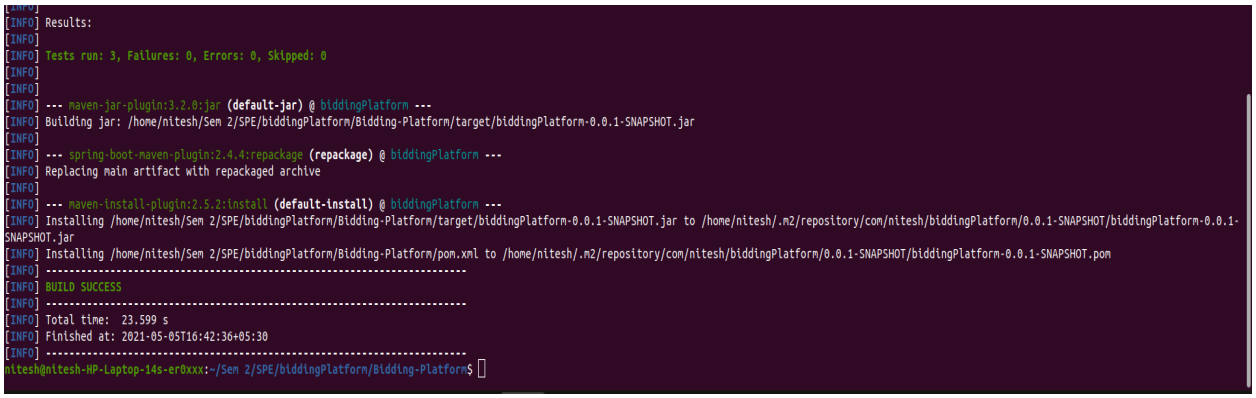

Fig-7 : Successful testing of the project

5.3.2. Mockito :

It is a mocking framework that tastes really good. It lets you write beautiful tests with a clean & simple API. Mockito doesn't give you a hangover because the tests are very readable and they produce clean verification errors.The framework allows the creation of test double objects in automated unit tests for the purpose of test-driven development or behavior-driven development. The framework's name and logo are a play on mojitos, a type of drink

5.3.3. Test Example :

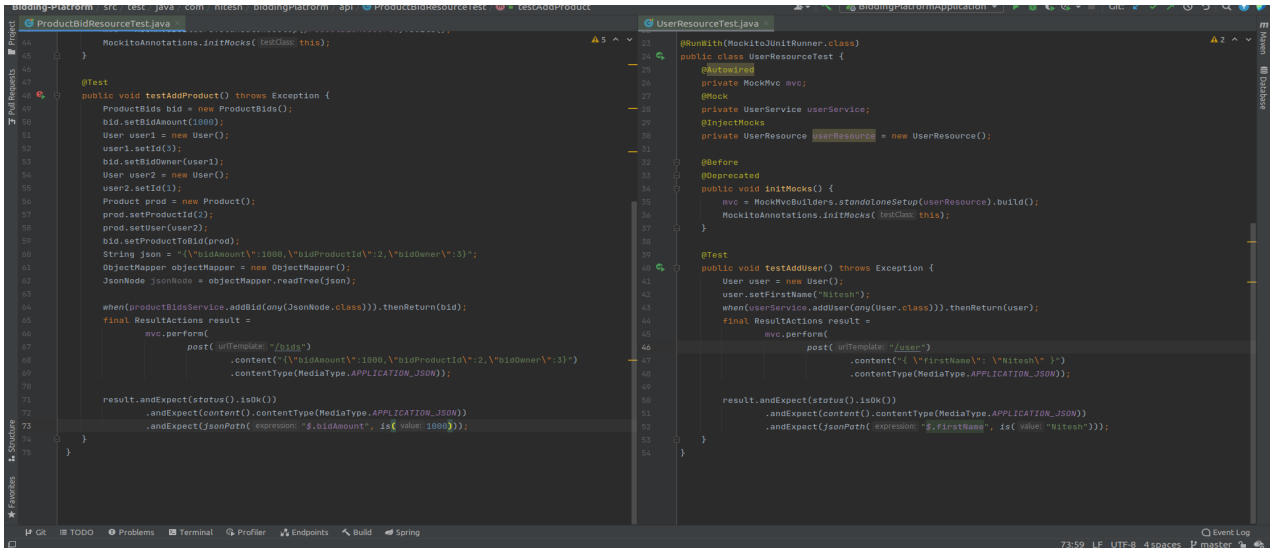

Fig-8 : Code for test cases

5.4. Archive - Docker :

Archiving is the process by which inactive information, in any format, is securely stored for long periods of time. A war file is generated after the project code successfully builds and passes all

the test cases.Then a docker image is built which contains a JAR file(for the spring boot backend) generated from building the project and React code structure with node_modudles(for react js frontend), will be copied to the same image. A dockerfile is written to perform this task, It contains the following commands.

We create docker images for both react and spring boot and push this image to the docker hub repository. The image created contains the latest version of the source code which can be built successfully to see all the features that have been implemented so far.

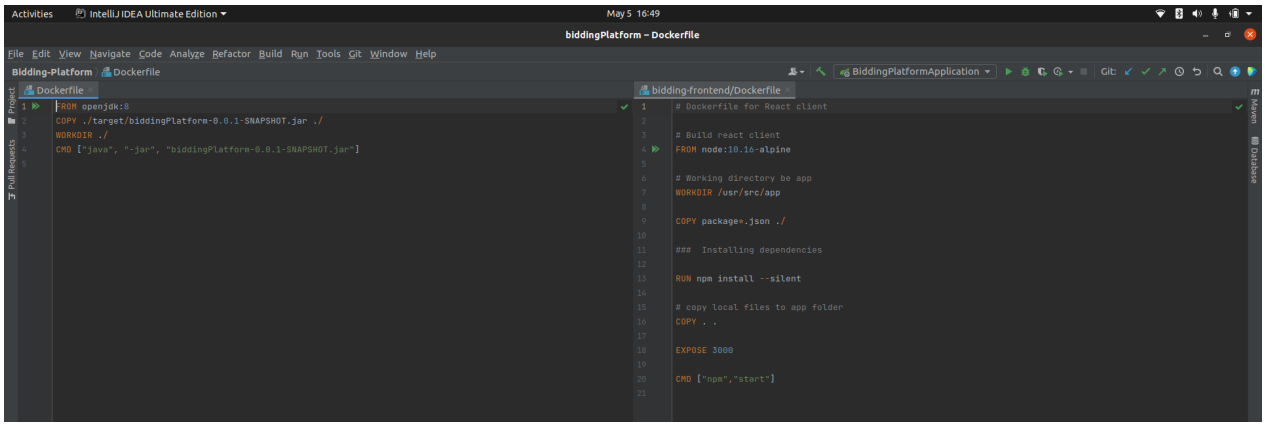

Fig-9 : Docker file

5.5. Deployment - Ansible :

- Ansible is a radically simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.
- Designed for multi-tier deployments since day one, Ansible models your IT infrastructure by describing how all of your systems interrelate, rather than just managing one system at a time.
- It uses no agents and no additional custom security infrastructure, so it's easy to deploy - and most importantly, it uses a very simple language (YAML, in the form of Ansible Playbooks) that allow you to describe your automation jobs in a way that approaches plain English.
- Ansible works by connecting to your nodes and pushing out small programs, called "Ansible modules" to them. These programs are written to be resource models of the desired state of the system. Ansible then executes these modules (over SSH by default),

and removes them when finished.Your library of modules can reside on any machine, and there are no servers, daemons, or databases required.

- Typically you'll work with your favorite terminal program, a text editor, and probably a version control system to keep track of changes to your content.
- Passwords are supported, but SSH keys with ssh-agent are one of the best ways to use Ansible. Though if you want to use Kerberos, that's good too.
- Root logins are not required, you can login as any user, and then su or sudo to any user. Ansible's "authorized_key" module is a great way to use ansible to control what machines can access what hosts.

5.5.1. Ansible Configuration :

In order to run ansible, we need to create the connection of the host machine to all the node machines. This is done by making a SSH Connection between the machines.

The steps involved in creating SSH connection are as follows:

1. $ssh-keygen - This command is used for the generation of public and private keys in the host machine. These keys are required to be sent to the node machines to establish a connection between each other.

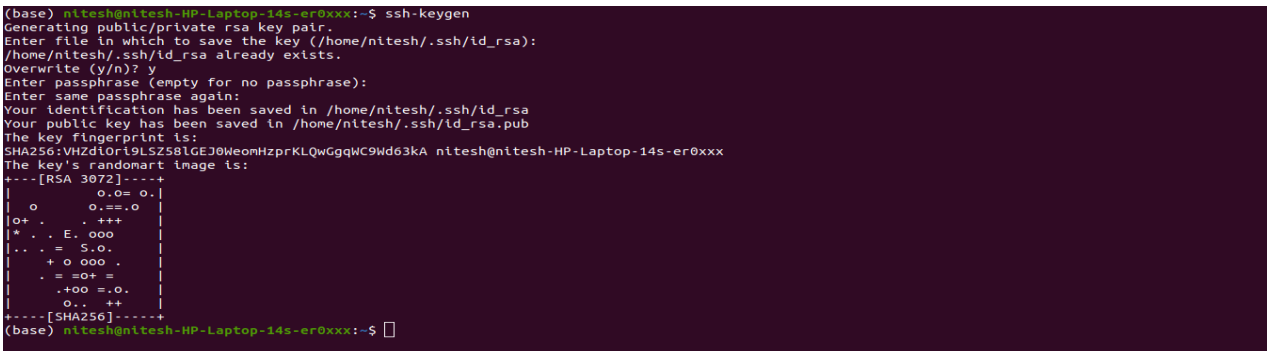

Fig-10 : Generating and sharing ssh key

2. $ssh-copy-id <username>@<IP address> : This command is run in the host machine to send the public key of the host machine to the node machine. This command copies the public key of the host machine to the .ssh/authorized_keys file of node machine whose IP address is provided in the command.

$ ssh-copy-id nitesh@168.62.161.35

The above command ensures the connection of both machines that won't require any password.

3. The successful execution of this command ensures that we got connected to the node machine as it did not ask for any password to get connected using ssh.
4. As the execution of ansible playbooks is automated using jenkins thus we need to perform all the above tasks from the jenkins user of the host machine because the jenkins server uses the jenkins user to perform all its tasks.

5.5.2. Inventory :

Ansible works against multiple systems in your infrastructure at the same time. It does this by selecting portions of systems listed in Ansible's inventory file, which defaults to being saved in the location /etc/ansible/hosts. You can specify a different inventory file using the -i <path> option on the command line.

We can also create our own inventory that contains the details specific to that project.

Fig-11 : Inventory file

5.5.3. Ansible-Playbook :

Ansible Playbooks offer a repeatable, re-usable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications. If you need to execute a task with Ansible more than once, write a playbook and put it under source control. Then you can use the playbook to push out new configurations or confirm the configuration of remote systems.

In this project, the task performed by ansible is to pull the image from dockerhub into the node machines so that the project would be able to run in all the node machines.

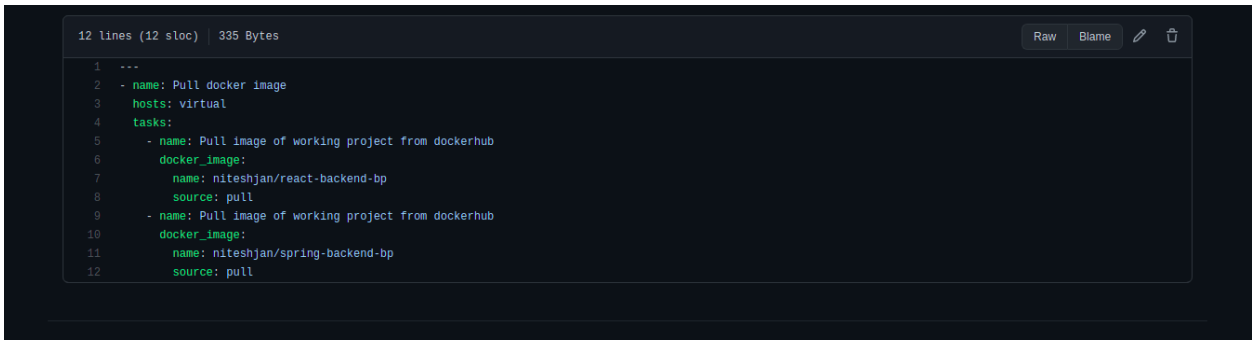

Fig-12 : Ansible playbook

In the above playbook, we have used the docker_image module of ansible that enables us to pull the image existing on docker hub into the node machines.

5.6. Continuous Integration - Jenkins :

Jenkins is a powerful application that allows continuous integration and continuous delivery of projects, regardless of the platform you are working on. It is a free source that can handle any kind of build or continuous integration. You can integrate Jenkins with a number of testing and deployment technologies.

Install jenkins and then run the command $systemctl start jenkins in order to start the jenkins service. The login page would appear where you can give your credentials to login into the jenkins server.

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.In this project, jenkins is used to automate the tasks as follows:

1. git clone
2. maven build
3. create docker image
4. push image to dockerhub
5. pull images to node machines

5.6.1. Building jenkins Pipeline :

For the above automation, we need to create a jenkins pipeline that contains the script for the complete task.

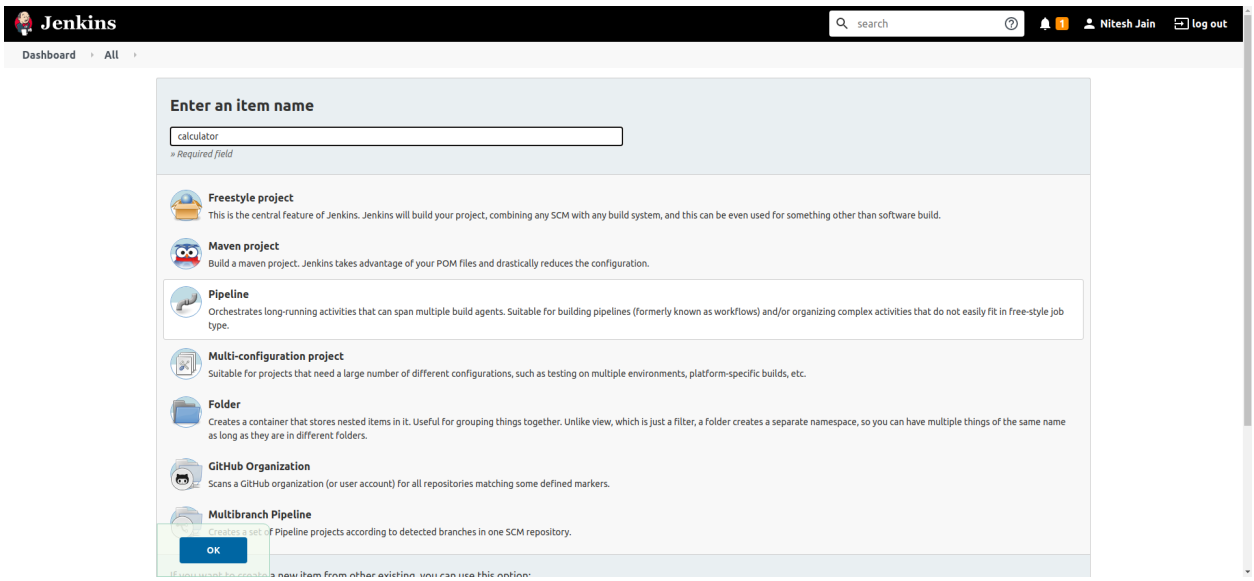

Fig-13 : Creating new jenkins pipeline

From the above window, we can create a jenkins pipeline and give a suitable name to the pipeline.This pipeline gives us the option to write the tasks that we need to perform using jenkins. All the tasks get performed one after another on just a single click.

The script for all the above tasks is as follows:

```
pipeline {
    agent any

    stages {
        stage('Github Clone') {
            steps {
                git 'https://github.com/niteshjan/Bidding-Platform.git'
            }
        }
        stage('Build the artifact') {
            steps {
                sh 'mvn clean install'
            }
        }
        stage('Create docker images') {
            steps {
                sh 'docker-compose build'
            }
        }
        stage('Push docker image to dockerhub'){
            steps{
                script{
                    docker.withRegistry('','miniproject'){
                        sh 'docker push niteshjan/spring-backend-bp:latest'
                        sh 'docker push niteshjan/react-backend-bp:latest'
                    }
                }
            }
        }
        stage("Pull images from dockerhub to the host machine"){
            steps{
                ansiblePlaybook installation: 'Ansible', inventory: 'inventory', playbook: 'playbook.yml'
            }
        }
    }
}
```

Fig-14 : Jenkins script

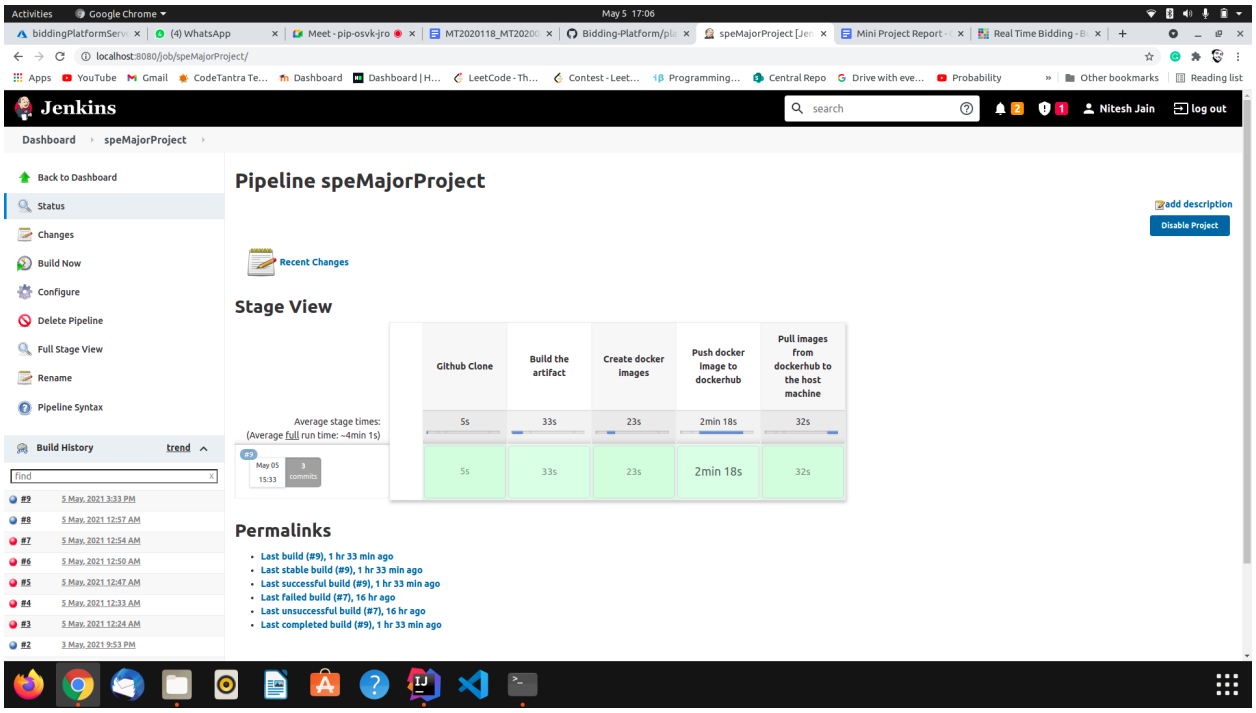

Fig-15 : Working of jenkins pipeline

6. Installation Procedure

6.1. SCM - git :

We should start out by running general OS and package updates. On Ubuntu we'll do this by running:

```
$ sudo apt-get update
```

After you have run the general updates on the server you can get started with installing Git.

1. Install Git using the following command :

```
$ sudo apt-get install git-core
```

2. You may be asked to confirm the download and installation; simply enter **y** to confirm. It's that simple, Git should be installed and ready to use!

3. Confirm the Git installation
   With the main installation done, first check to ensure that the executable file is set up and accessible. This can be done simply by running the Git version command.

```
$ git --version
```

6.2. Build - maven :

Prerequisites:

- For using Maven 3.3+ ,we need JDK 1.7 or above to be installed.

To install the JDK1.7 version in your system. Follow the below steps.

Updating the package index of the system and install the OpenJDK package:

```
$ sudo apt update
$ sudo apt install default-jdk
```

Verify the JDK1.7 java version installation by running the below command:

```
$ java -version
```

Maven can be installed by running the following commands:

```
$ sudo apt install maven
```

Maven installation can be verified by running the maven version command:

```
$ mvn -version
```

6.3. Testing -junit :

No separate installation needed. Here are the maven dependencies:

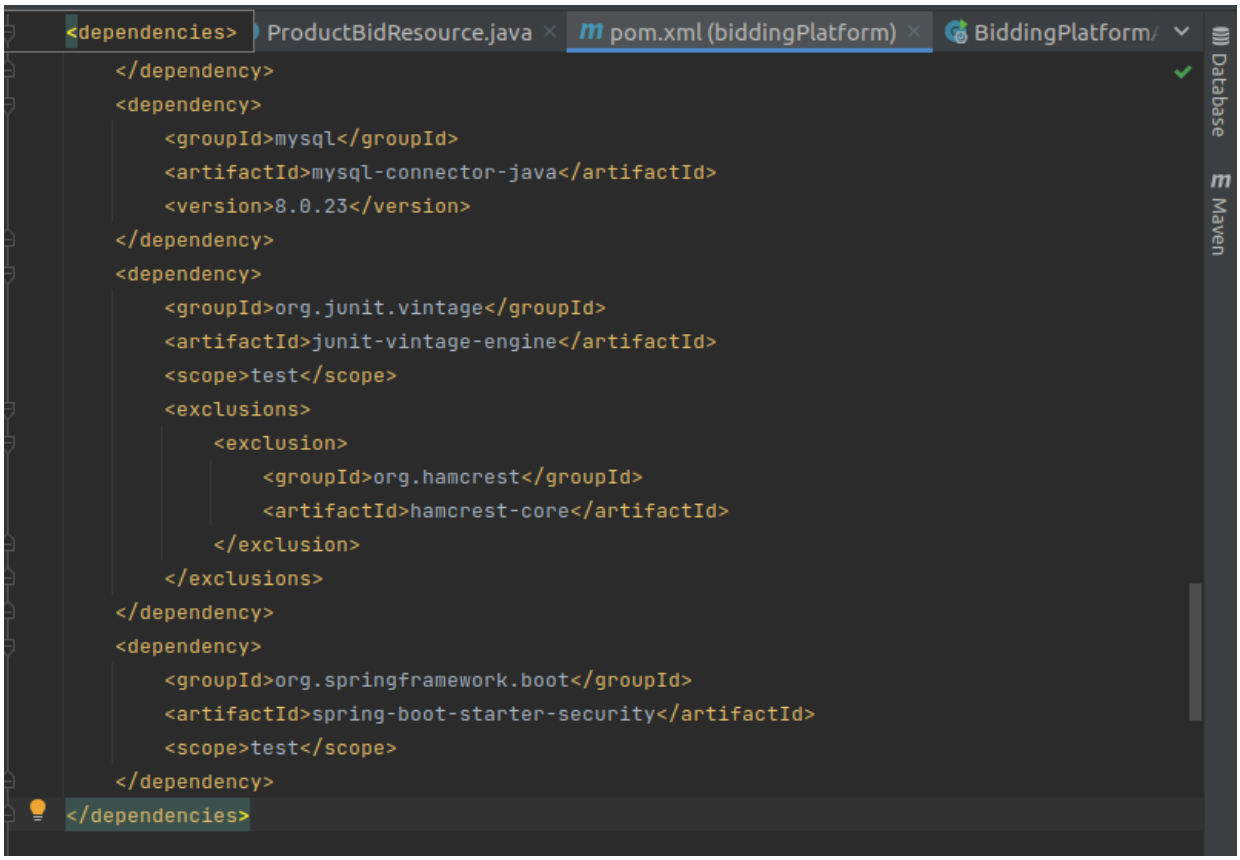

Fig-16 : Junit dependencies

6.4. Artifact - Docker :

Docker is an application that makes packaging and running an application process easy. It provides OS-level virtualization to deliver the software in a package called the container.

There are two methods to install Docker.

1. Installing it on an existing operating system (Ubuntu 16.04).
2. Using a tool called Docker Machine.

We are going to follow the first method. You can access the second method here.

6.4.1. Installing Docker :

- Laptop with Ubuntu 16.04 installed in it
- Docker Hub account to create and push your own images to the docker hub.

Ubuntu 16.04 repo may not contain the latest docker installation package. So It is better to download and install the latest version from the official docker repository.

Add The GPG key to the official Docker repo of your system to ensue the downloads are valid

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Then add Docker repository to APT sources of your Ubuntu system by running following command

```
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

Update the Ubuntu package Database

```
$ sudo apt-get update
```

To make sure we are installing from official docker repository run the following command :

```
$ apt-cache policy docker-ce
```

The output would look like this :

```
Installed: (none)
  Candidate: 5:19.03.9-3-0-ubuntu-xenial
  Version table:
    5:19.03.9-3-0-ubuntu-xenial 500
        500 https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
```

Finally Install docker

```
$ sudo apt-get install -y docker-ce
```

Docker should be installed now. Checking status of the docker by running the below command

```
$ sudo systemctl status docker
```

The output should look like :

```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-05-25 20:36:12 IST; 1h 42min ago
     Docs: https://docs.docker.com
 Main PID: 2435 (dockerd)
    Tasks: 30
   Memory: 40.7M
    CPU: 11.075s
   CGroup: /system.slice/docker.service
           ├─ 2435 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
           └─15162 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8081 -container-ip 172.17.0.2
container-port 8080
```

If the status of the docker is off. You can start the docker service using the following command

```
$ sudo systemctl start docker
```

Executing the docker without sudo command

By default running the docker commands required root privileges (prefix **sudo** to commands). In general a **docker group** is created by default. You need to add your Ubuntu User to docker group to run the docker without sudo.If the user is not added and tries to run a docker command without sudo then the output would look like this

```
Output
docker: Cannot connect to the Docker daemon. Is the docker daemon running on this host?.
See 'docker run --help'.
```

To run docker commands without sudo, add your Ubuntu username to the docker group

```
$ sudo usermod -aG docker ${USER}
```

For this newly created group membership to apply you need to logout and login or restart the device.If you want to add another user who is not a an administrator you can use the following command.

```
$ sudo usermod -aG docker username
```

You can get the documentation on all the docker commands here [49]. or run **docker** in command prompt. Tutorials on how to work with docker images and containers are available here [50].

To Deploy our Web application we are using the Tomcat web server and ExistDB as the database.

6.5. Deploy - ansible :

Ansible is used to automate the ad-hoc and routine tasks. It is generally used in data centers and cloud environments to avoid doing repetitive tasks.

Installation procedure for ansible

Minimum Requirements

- Java version 1.8

Installing Ansible

To install the ansible check the following link and configure ansible using this link.

https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-18-04

6.6. Monitoring - ELK :

ELK stands for three tools namely elastic search, logstash, and kibana. We have installed these three tools in Ubuntu 16.04 using exe available in elastic.co website.

First update your system using

```
$  sudo apt-get update
```

6.6.1. Installing Elasticsearch :

Elasticsearch is an open-source software used to store and analyze large volumes of data in real-time. It is a text search and analytics engine.

For Ubuntu 16.04 bit install using :

```
$  echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-7.x.list
$  sudo apt-get install elasticsearch
$  sudo systemctl start elasticsearch
$  sudo systemctl enable elasticsearch
```

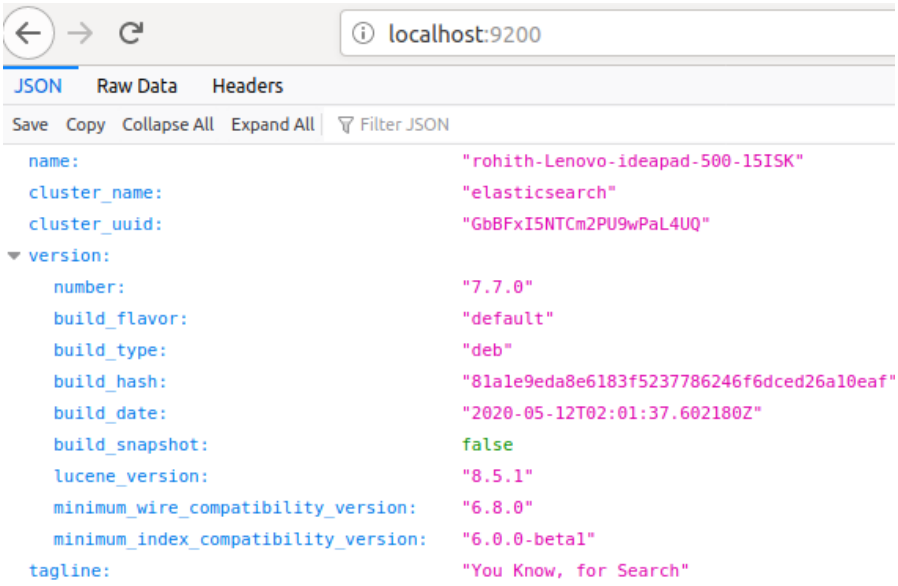

Fig-17 : elasticsearch running on localhost:9200

6.6.2. Installing Logstash :

Logstash is essentially a log aggregator that collects data from various input sources, like servers, applications, docker containers etc and performs various transformations and pushes it into elasticsearch.

Again for Ubuntu 16.04 bit install using :

```
$  sudo apt-get install logstash
$  sudo systemctl start logstash
```

6.6.3. Installing Kibana :

Kibana is used for visualization and it works on top of Elasticsearch. It also provides us with the ability to analyze or query the data. These visualizations could be in the form of tables, plots, charts etc.

Again for Ubuntu 16.04 bit install using :

```
$  sudo apt-get install kibana
$  sudo systemctl start kibana
```

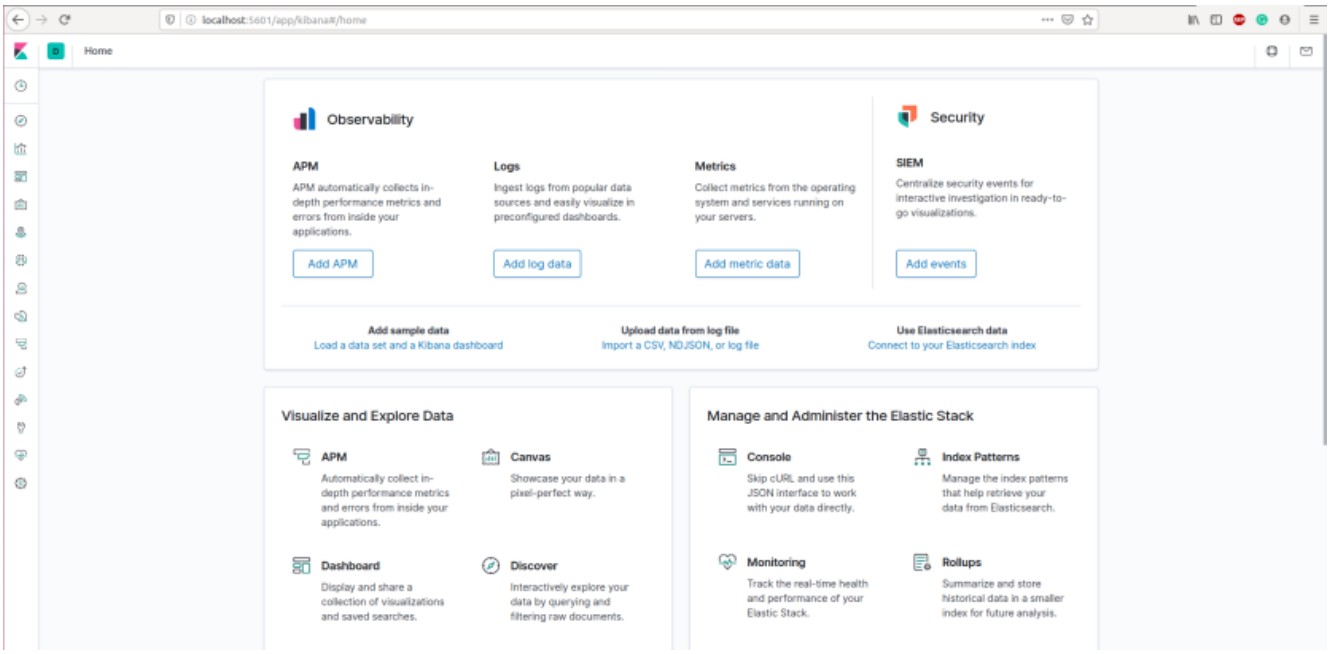

Fig-18 : Kibana Home page running in localhost 5601

6.6.3.1. Kibana Dashboard :

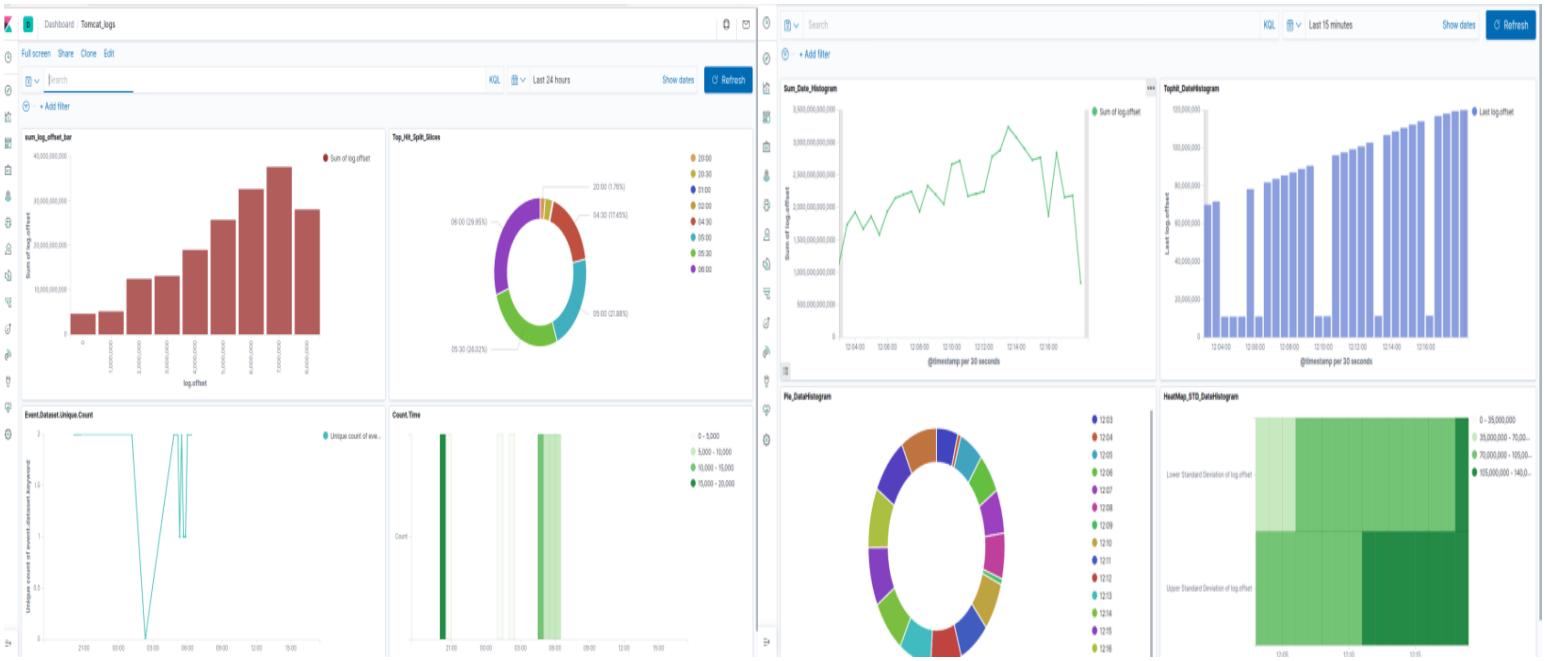

Fig-19 : Kibana Dashboard

6.7. Continuous Integration - Jenkins :

Prerequisites :

Jenkins is a Java based application, for using jenkins first step is to install java. Use the following commands to Update the package index of Ubuntu 16.04 and then to install the Java 8 OpenJDK package :

```
$ sudo apt update
$ sudo apt install openjdk-8-jdk
```

Next we need to add the Jenkins Debian repository. Use the following command to import the GPG keys of the Jenkins repository :

```
$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
```

After running the above command you should get a OK message , which means that the key has been successfully imported and packages from this repository will be considered trusted.

Next step is to add the Jenkins repository to the system using following commands:

```
$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/
> /etc/apt/sources.list.d/jenkins.list'
```

Then update the Ubuntu apt package list and install jenkins using following commands :

```
$ sudo apt update
$ sudo apt install jenkins
```

By default Jenkins service will start immediately once the installation process is completed. You can verify the status of the Jenkins using the following command

```
$ systemctl status jenkins
```

Git in jenkins :

We need to add the git plugin to jenkins.Click on the Manage Jenkins tab on your Jenkins dashboard and open manage plugins, search for GIT Plugin and choose install without restart . Installation might take some time to download all the dependencies,restart the jenkins after installation successfully finished.

Fig-20 : jenkins git plugin

Maven in Jenkins :

Maven Integration plugin helps us to build projects that use Apache Maven in Jenkins.Go to Manage Jenkins, select Manage Plugins, select the Available tab, then find the Maven Integration plugin,install it and restart the jenkins once installation is completed.

Docker in Jenkins :

A docker plugin is needed to add docker to the jenkins pipeline, To install Docker plugin in jenkins. Add jenkins in the docker group to execute docker commands from jenkins without root privileges (sudo).

```
$ sudo usermod -aG docker jenkins
```

Then add docker hub credentials in the credentials section in jenkins. This helps to login into your docker hub and push the docker image directly via jenkins pipeline so that continuous deployment can be done.Restart Jenkins after doing this. Jenkinsfile can now be edited to add the docker build and push stages.

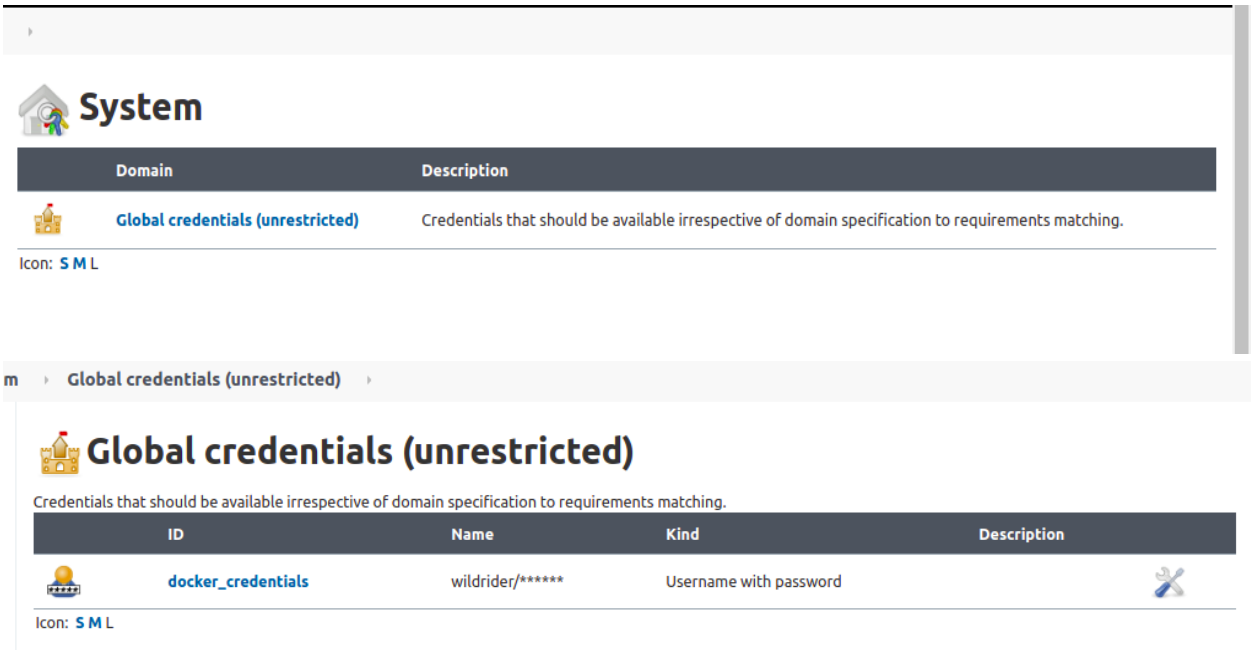

Fig-21 : Adding docker credentials in jenkins

Ansible in jenkins :

Install ansible plugin in Jenkins.To install follow the given procedure:

In Manage Jenkins > Configure System > Ansible , fill the required details and do Test Connection . If it is working, it shows a successful message.

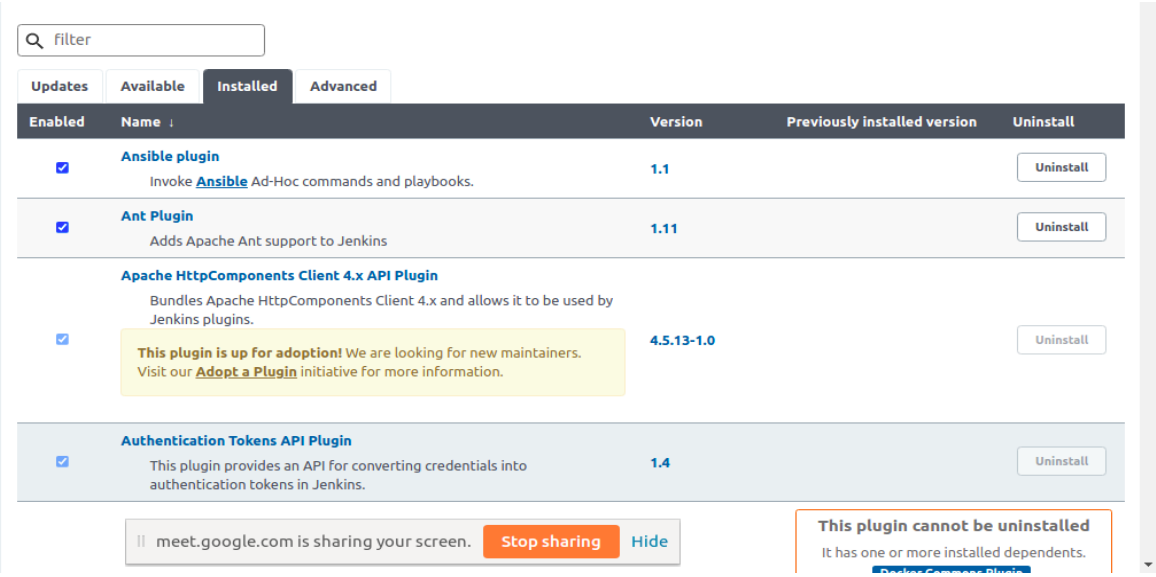

Fig-22 : ansible plugin in jenkins

7. Experimental Setup

7.1. Functionalities :

- A user can be both bidder and owner at the same time.

- A user can register with its details such as name, contact number, gender, email etc.
- The user can see all the products posted by him as well as other users in the platform
- The user can post an ad of its product by specifying few details about the product such as product name, description, minimum bid etc.
- Out of the list of products, the user can select the product that he is interested in and get the details of the product as well as the user.
- The user can bid on the product on which he is interested.
- The bid amount should be more than the minimum bid else the bid would not be able to be set.
- Once the bid is successfully placed then it would be visible in the product details section.
- The owner of the product can then decide which bid he wants to select out the bids placed.
- Once the bid is selected then it will not be able to be bid by other users that is, it would become inactive.

7.2. Controllers :

7.2.1. User Controller :

The user controller in the java class that has the following functionalities:

```
1. @PostMapping
   public User addUser(@RequestBody User user)
```

The add user function enables the user to add users to the database. When the form is filled in the front end, its data is being filled in the json and then sent to the backend with the help of axios libraries.

The data from the frontend is converted to user object and passed as a parameter. This adds the user to the database.

The user model has two oneToMany entity relationships with product model and productBid model that corresponds to adding foreign key constraints in the database.

2.
```
@GetMapping
public List<User> getUsers()
```

This function gets all the rows from the user table in the database.

3.
```
@GetMapping(value = "{userId}")
public User getUser(@PathVariable int userId)
```

This function gets a particular row from the user table in the database. In this the userId is passed from the frontend and corresponding to this userId, the row is fetched from the database. If the user of this particular userId does not exists then it would through ResourceNotFoundException.

4.
```
@PutMapping(value = "{userId}")
public User updateUser(@PathVariable int userId, @RequestBody User
user)
```

This function is used to update the already existing user in the database. It takes a json object of updated user and the userId from the frontend and then updates the existing user with the new details provided by the user. It does not create a new entry rather update the same one.

7.2.2. Product Controller :

1.
```
@PostMapping
public Product addProduct(@RequestBody Product product)
```

The add product function enables the user to add products to the database. When the form is filled in the front end, its data is being filled in the json and the sent to the backend with the help of axios libraries.

The data from the frontend is converted to product object and passed as a parameter. This adds the product to the database.

The product model has a manyToOne entity relationship with the User table and oneToMany entity relationship with the productBid table to enable the foriegn key constraints in the database.

2.
```
@GetMapping
public List<Product> getProducts()
```

This function gets all the rows from the product table in the database.

3.
```
@GetMapping(value = "{productId}")
public Product getProduct(@PathVariable int productId)
```

This function gets a particular row from the product table in the database. In this the userId is passed from the frontend and corresponding to this productId, the row is fetched from the database. If the user of this particular productId does not exist then it would through ResourceNotFoundException.

4.
```
@PutMapping(value = "{userId}")
public User updateUser(@PathVariable int userId, @RequestBody User
user)
```

This function is used to update the already existing product in the database. It takes a json object of updated product and the productId from the frontend and then updates the existing product with the new details provided by the product. It does not create a new entry rather update the same one.

7.2.3. Product Bid Controller :

1.
```
@PostMapping
public ProductBids addProductBid(@RequestBody ProductBids
productBids)
```

The add product bids function enables the user to add products to the database. When the form is filled in the front end, its data is being filled in the json and the sent to the backend with the help of axios libraries.

The data from the frontend is converted to product bid object and passed as a parameter. This adds the product bid to the database.

The product bid model has a ManyToOne entity relationship with the User table and ManyToOne entity relationship with the Product table to enable the foriegn key constraints in the database.

2.
```
@GetMapping
public List<ProductBids> getBids()
```

This function gets all the rows from the productBid table in the database.

3.
```
@GetMapping(value = "{bidId}")
public ProductBids getBid(@PathVariable int bidId)
```

This function gets a particular row from the product bid table in the database. In this the userId is passed from the frontend and corresponding to this bidId, the row is fetched from the database. If the user of this particular bidId does not exist then it would through ResourceNotFoundException.

4.
```
@PutMapping(value = "{bidId}")
public ProductBids updateBid(@PathVariable int bidId, @RequestBody
JsonNode jsonNode)
```

This function is used to update the already existing bid in the database. It takes a json object of updated product and the bidId from the frontend and then updates the existing bid with the new details provided by the user. It does not create a new entry rather update the same one.

7.3. Models :

There are three models in the projects which are:

1. User
2. Product
3. ProductBids

These are java classes with @Entity notation. The @Entity notation makes the class entity class which directly corresponds to the tables in the database. In these classes we have also defined the entity relationship such as OneToMany and ManyToOne

OneToMany entity Relationship : We can specify a *many-to-one* relationship by using the *@OneToMany* annotation. *One-to-many* mapping means that one row in a table is mapped to multiple rows in another table.

ManyToOne entity Relationship : We can specify a *many-to-one* relationship by using the *@ManyToOne* annotation. A *many-to-one* mapping means that many instances of this entity are mapped to one instance of another entity – **many items in one cart**.

7.3.1 User Model :

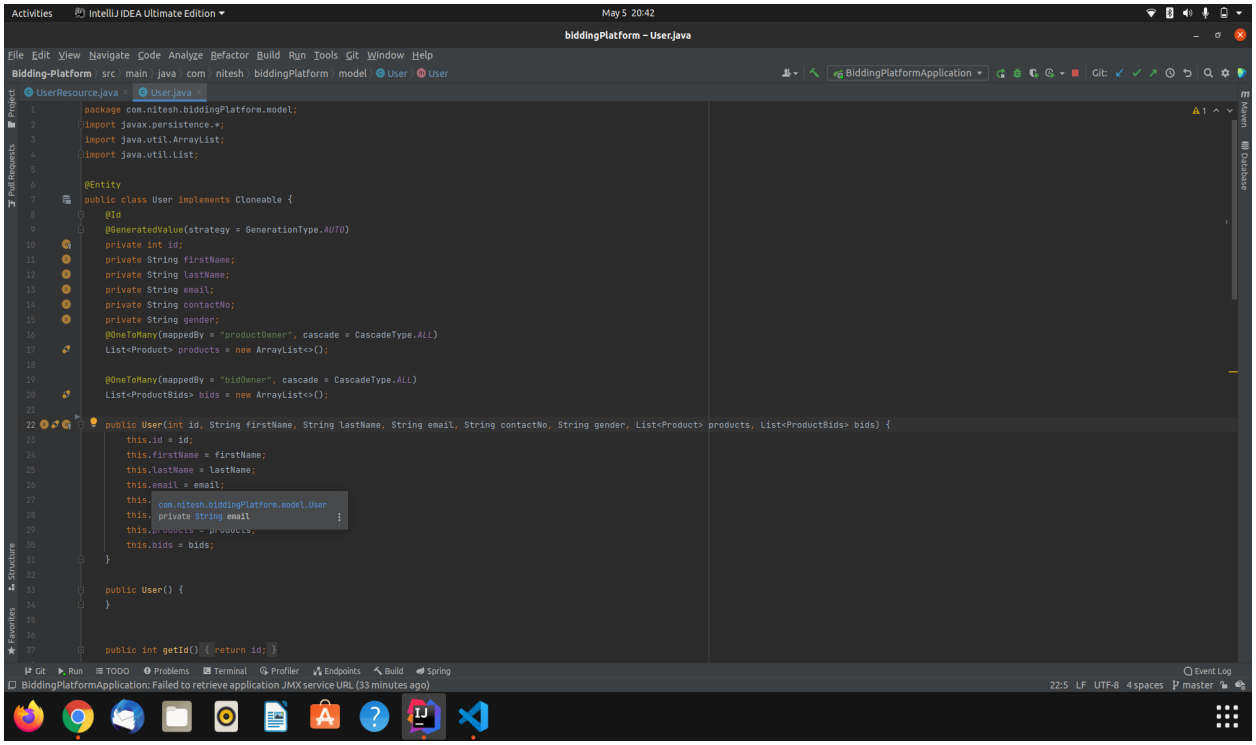

Fig-23 : user model code

7.3.2 Product Model :

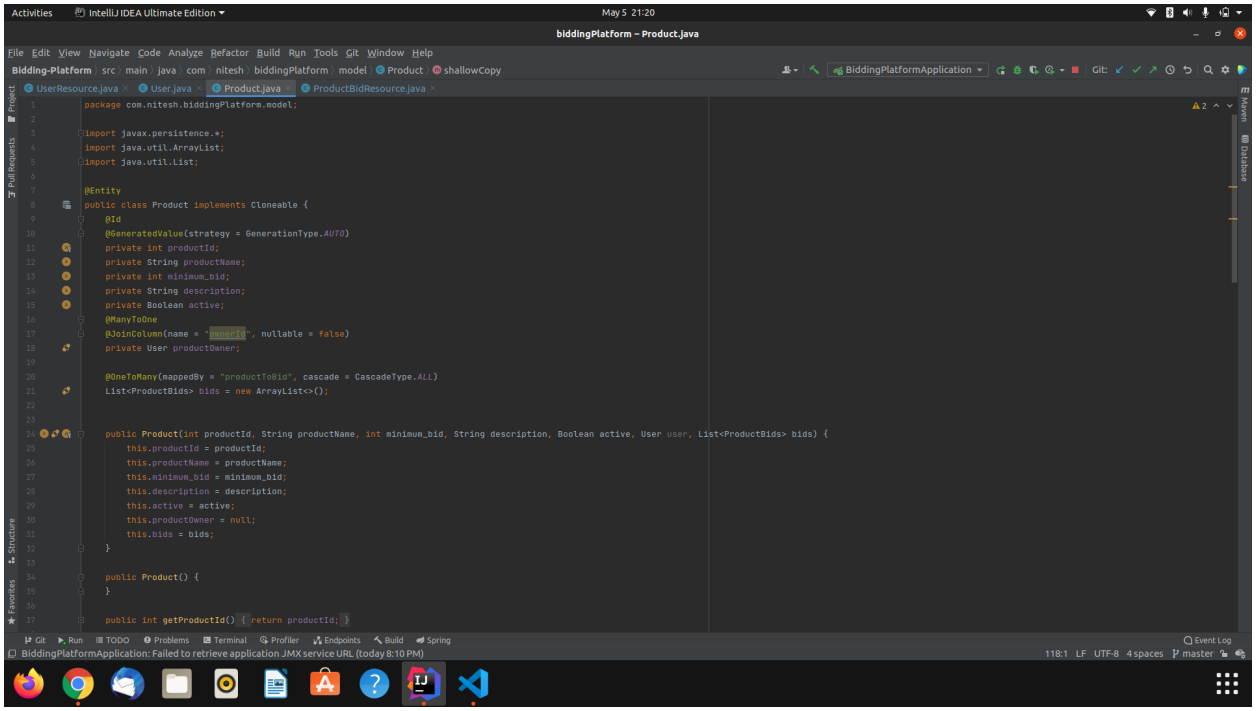

Fig-24 : product model code

7.3.3 ProductBids Model :

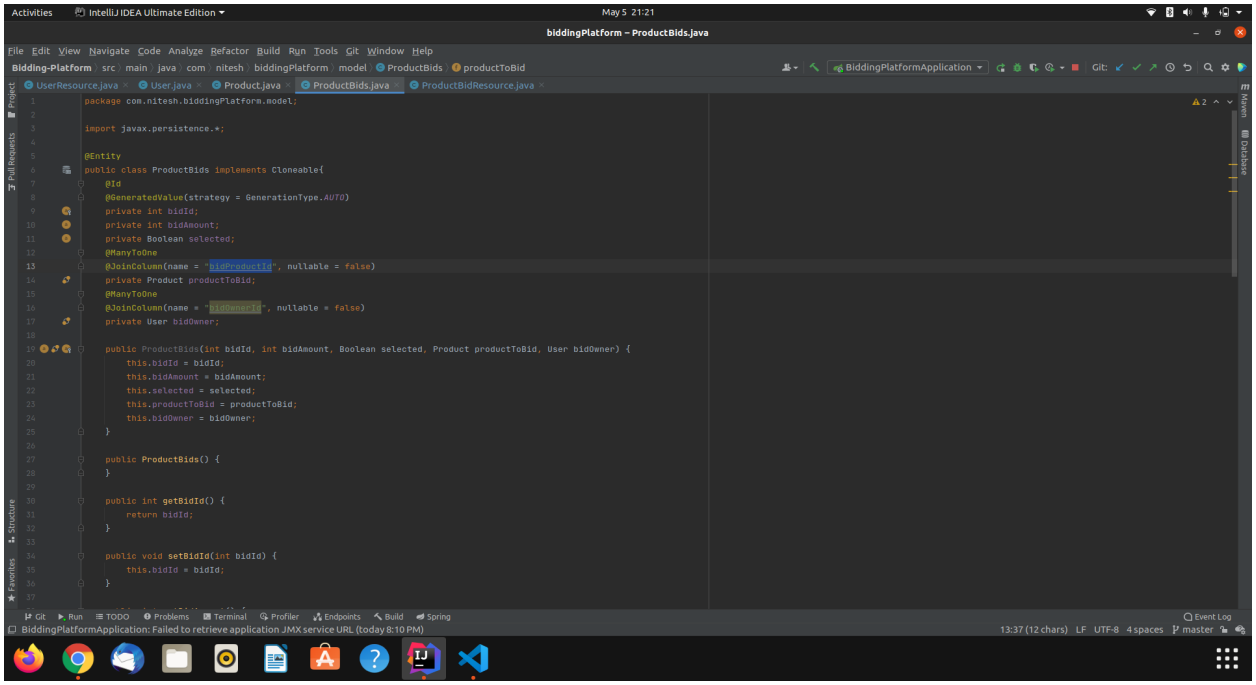

Fig-25 : product bids model code

7.4. Database :

We have used Azure Database for MySQL server. Azure Database for MySQL is a relational database service powered by the MySQL community edition.You can use either Single Server or Flexible Server (Preview) to host a MySQL database in Azure. It's a fully managed database as a service offering that can handle mission-critical workloads with predictable performance and dynamic scalability.Azure Database for MySQL is easy to set up, manage and scale. It automates the management and maintenance of your infrastructure and database server, including routine updates, backups and security. Enjoy maximum control of database management with custom maintenance windows and multiple configuration parameters for fine grained tuning with Flexible Server.

Steps to create a MYSQL server in azure:-

1. Go to the home page of Microsoft Azure, where we can see all the resources that have already been created by the user. Along with that, it also shows all the resources out of which we can select the resource that we want to deploy.

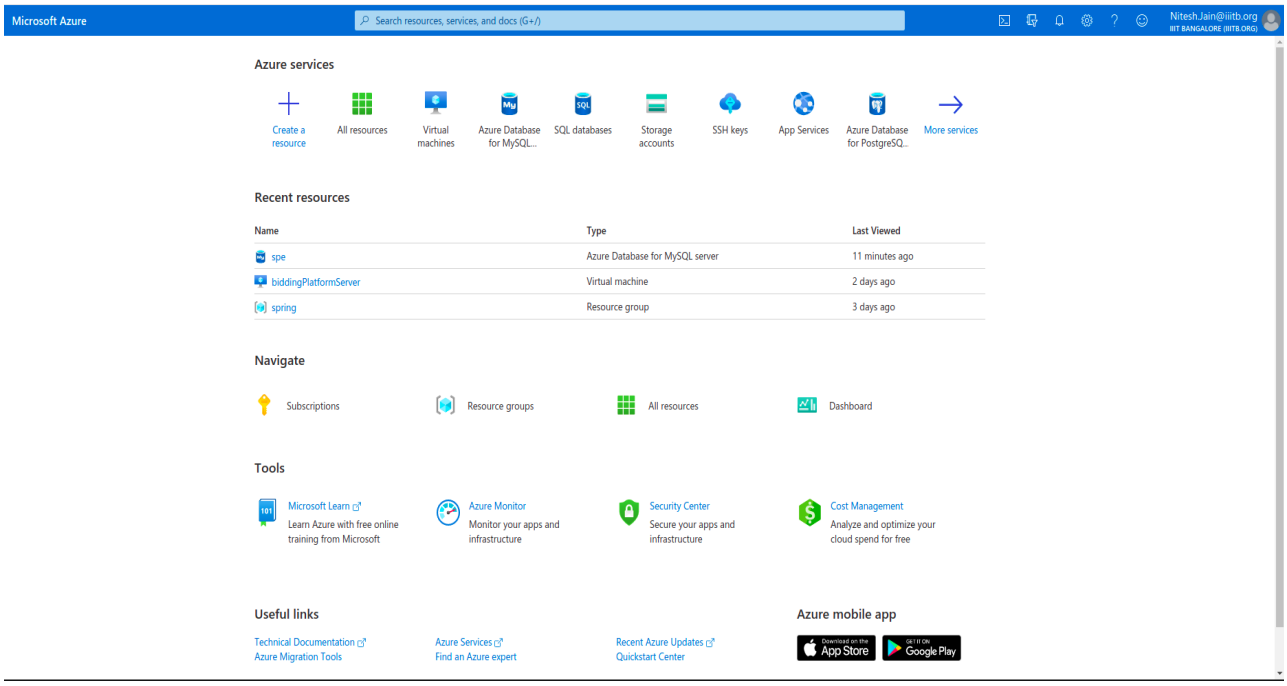

Fig-26 : azure dashboard

2. From the dashboard, select the Azure database for MYSQL server resource. From the page that opens up, click on the add button to create a new server.

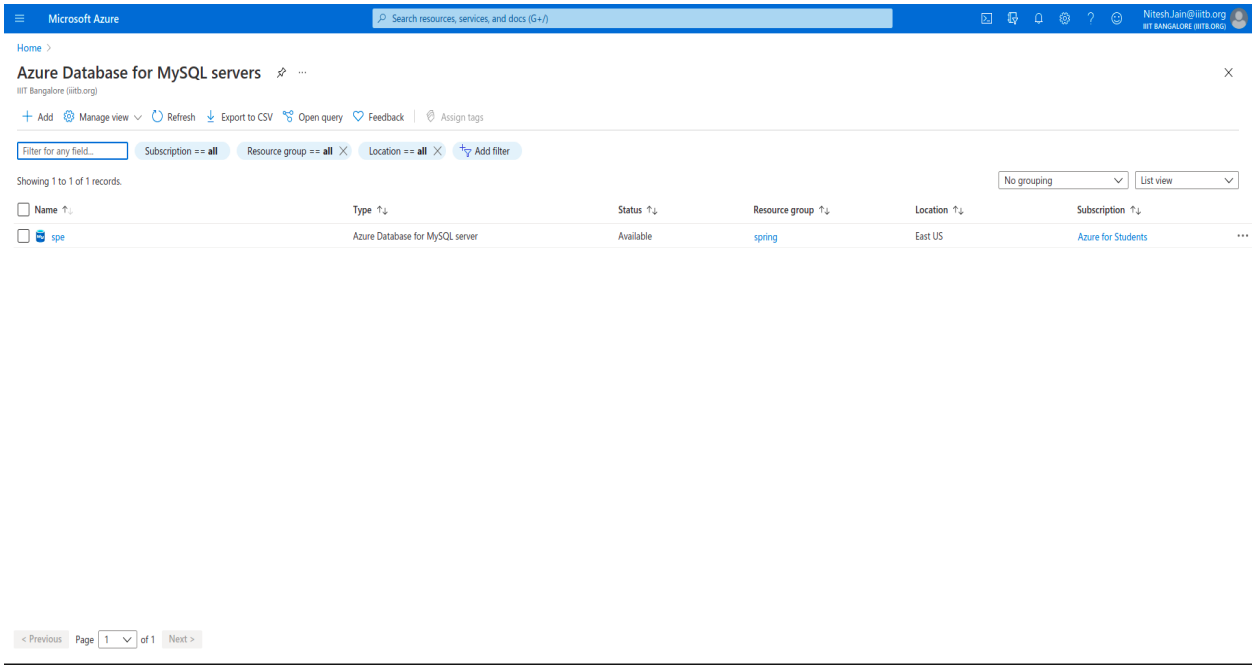

Fig-27 : creating new server(part - 1)

3. Then a form appears where we need to fill in the details required to deploy the server. The details include the name of the resource group, server name, location ,version, storage and the administrator account details.

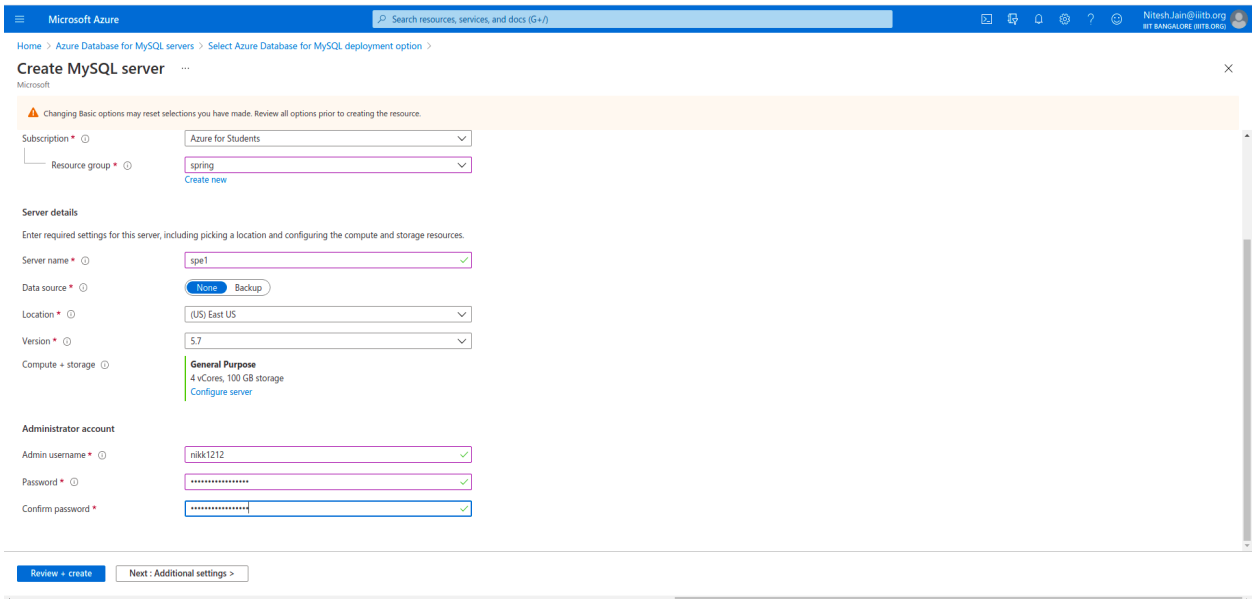

Fig-28 : creating new server(part - 2)

4. After filling in the details, click on review and create a button. On this page, the complete details that are filled by us appear. After reviewing the details, click on the create button to create and deploy the resource.

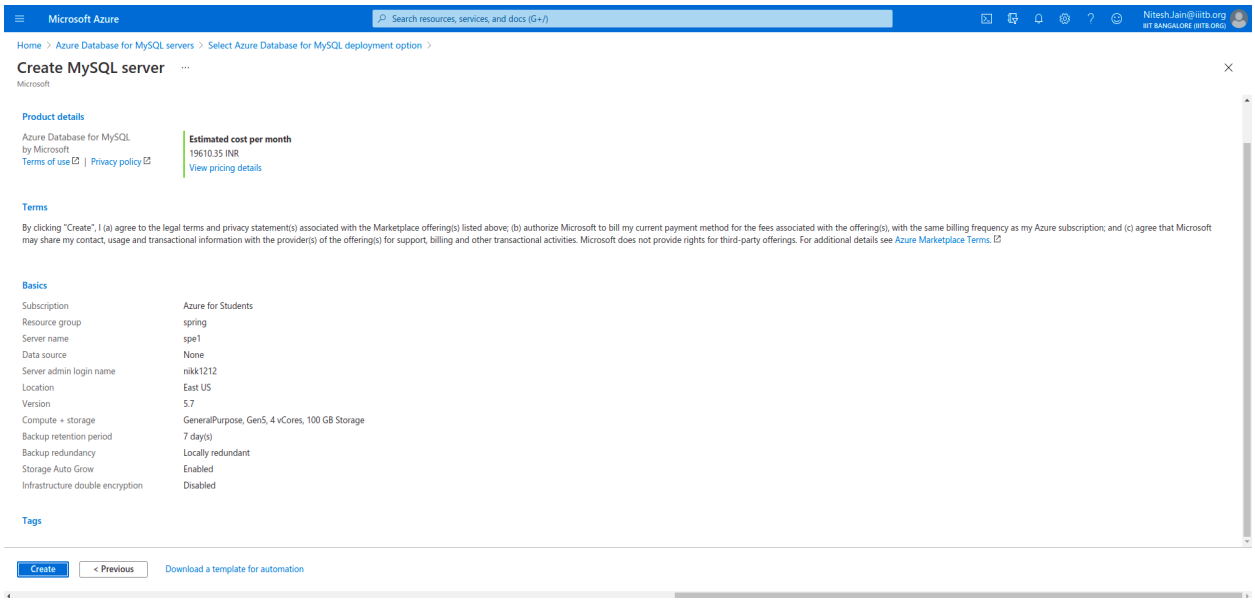

Fig-29 : creating new server(part - 3)

5. Below is the dashboard of Azure Database for MYSQL server from where we can start and stop the database.

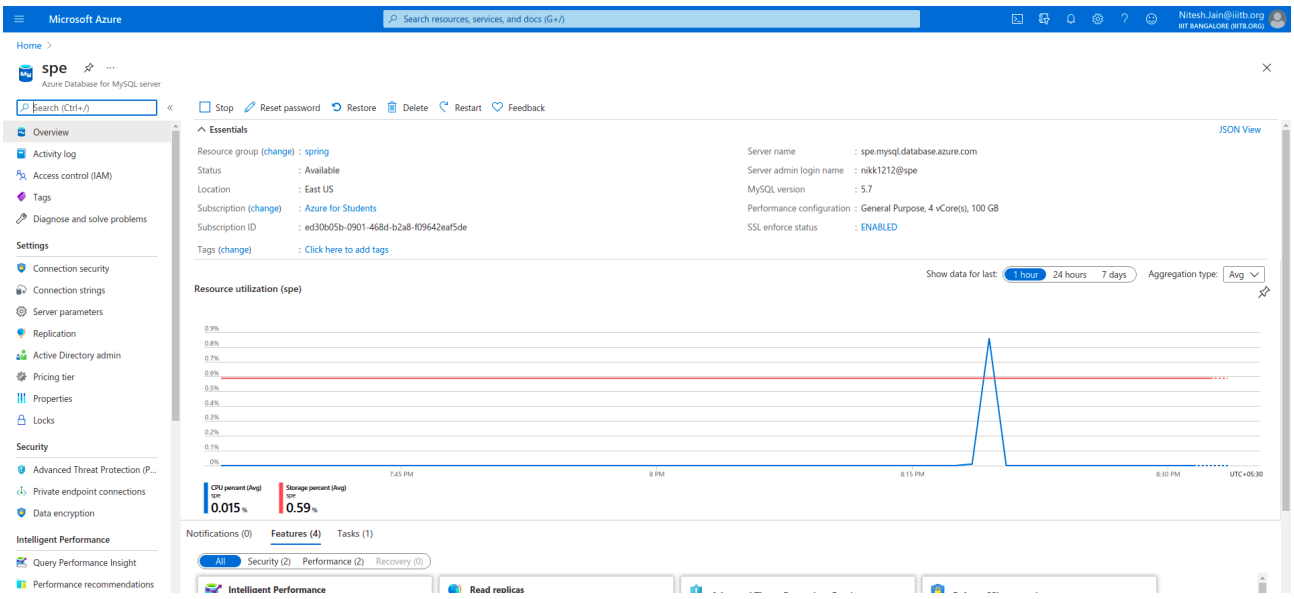

Fig-30 : MySQL dashboard for azure

This database can be viewed using the MySQL workbench. It requires the server name along with the administrator's credentials to access the database.When we open the mysql workbench it requires the administrator password, it's the same password that we gave during the resource creation.

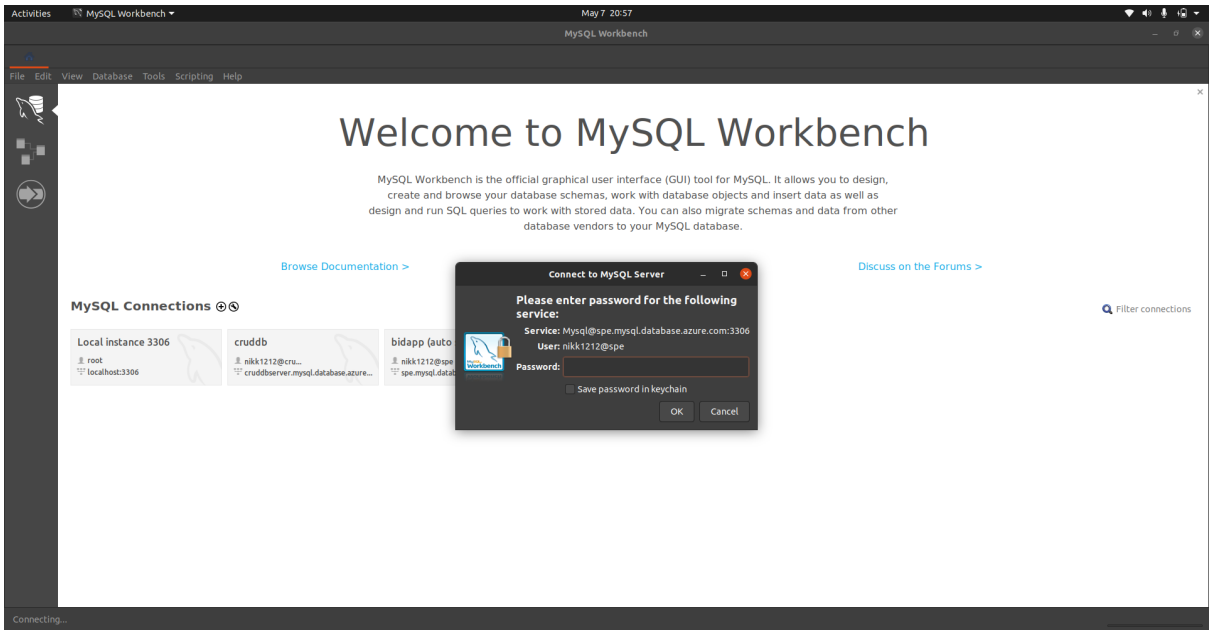

Fig-31 : MySQL workbench

After the successful authentication, the database opens up where all the tables are visible. We can perform several SQL Commands that we want in the same.

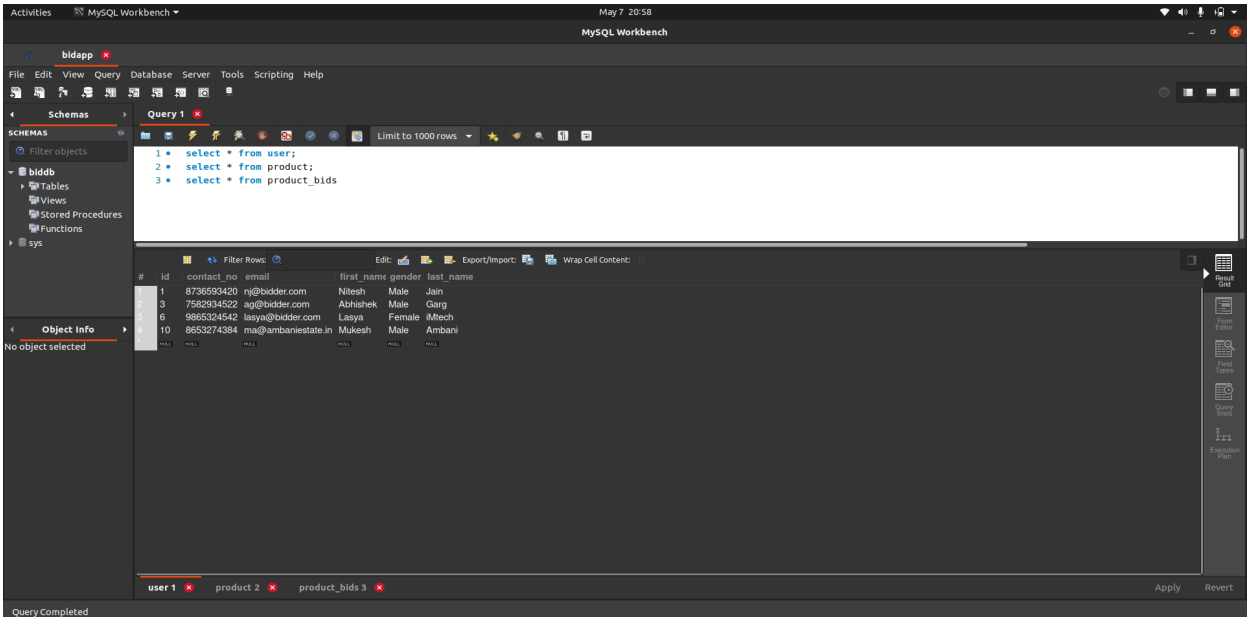

Fig-32 : MySQL database tables

8. Results

8.1. Website - Bidding Platform :

8.1.1. Home Page : The home page consists of the ads of all the products that have been placed by all the users. It consists of details of the product.

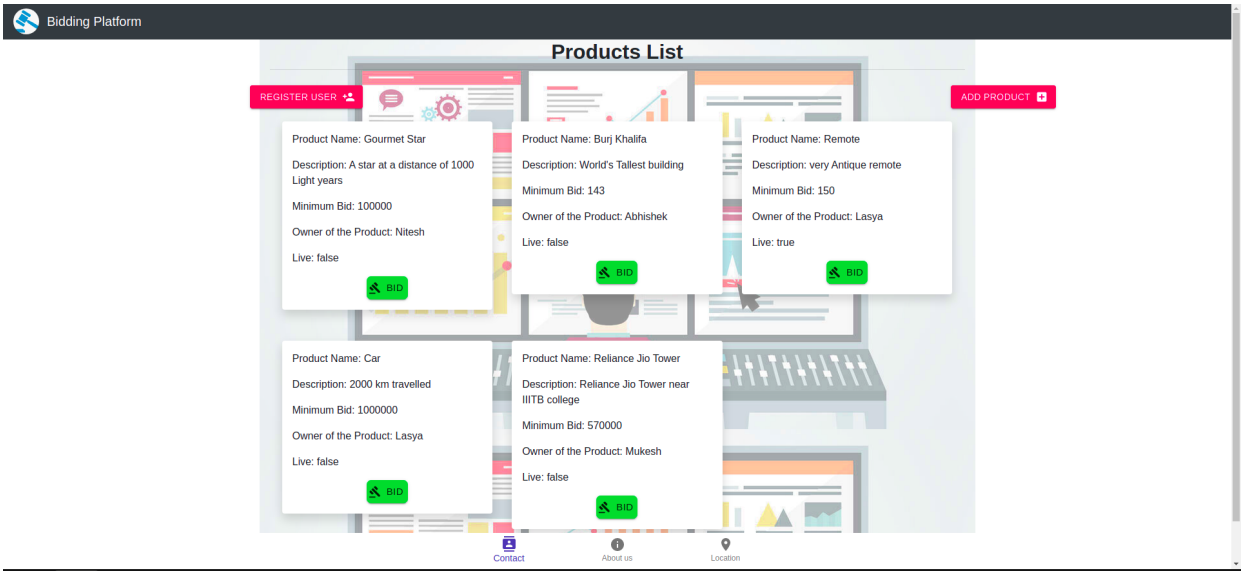

Fig-33 : Home Page of our application

8.1.2. Register User Page : The register user page enables the user to register itself. The user fills in the required details and clicks on the register button to make himself able to post ads as well as buy products.

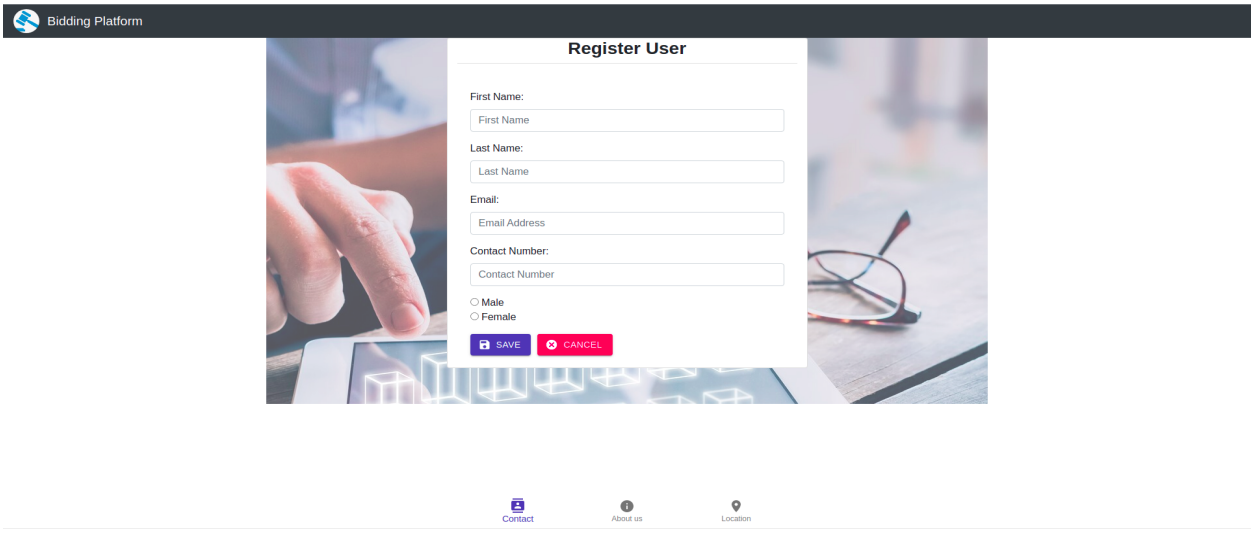

Fig-34 : Registering new user

8.1.3. Add Product Page : The add product page enables the user to post the ad for his product on which other people can bid. The page requires few details about the product to fill up and clicking on the add product button, posts the ad of the product and the product would now be visible in the products list.

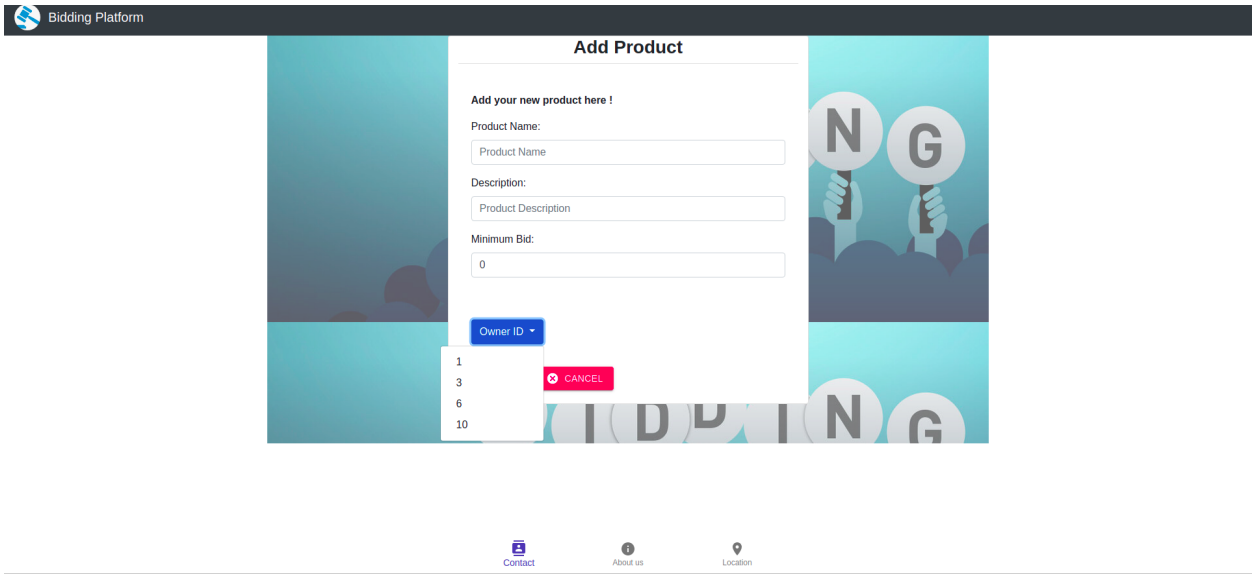

Fig-35 : Adding new product

8.1.4. Product Details Page : The product details page consists of details about the product as well as the owner of the product. This page contains a Update Product button that enables the user to update the product details if required. The page also shows all the bids that have been placed on the product

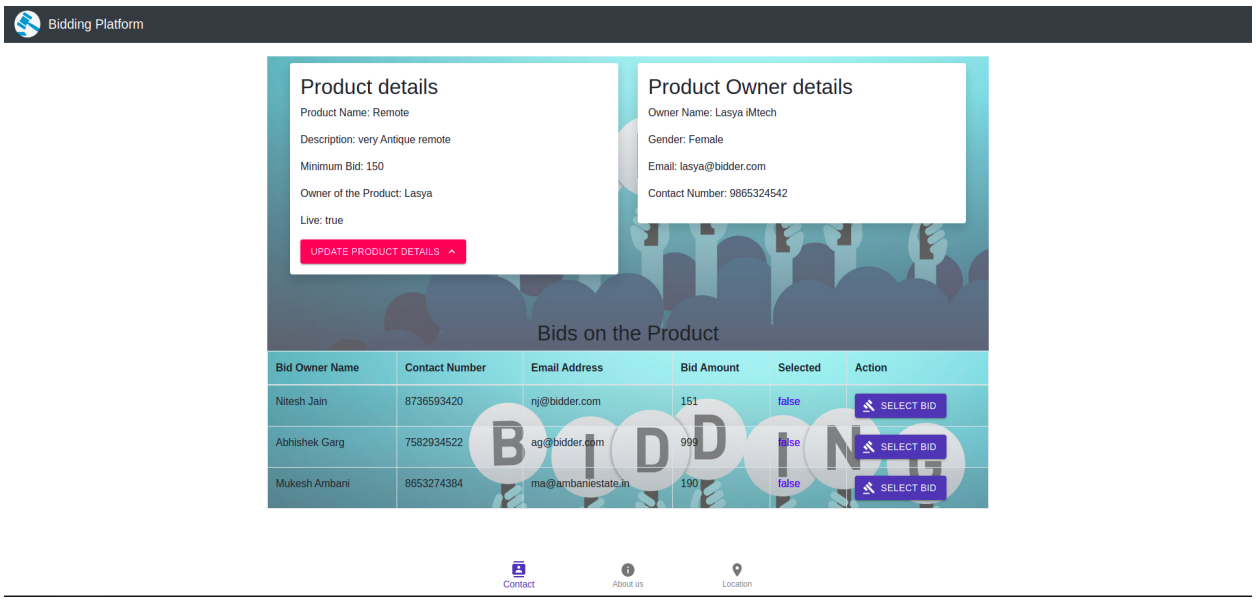

Fig-36 : Viewing product details

8.1.5. Update Product Details Page : The update product page enables the user to update the product details that have already been put up by him. This also enables the user to deactivate the product which makes the product to be unable to bid by users.

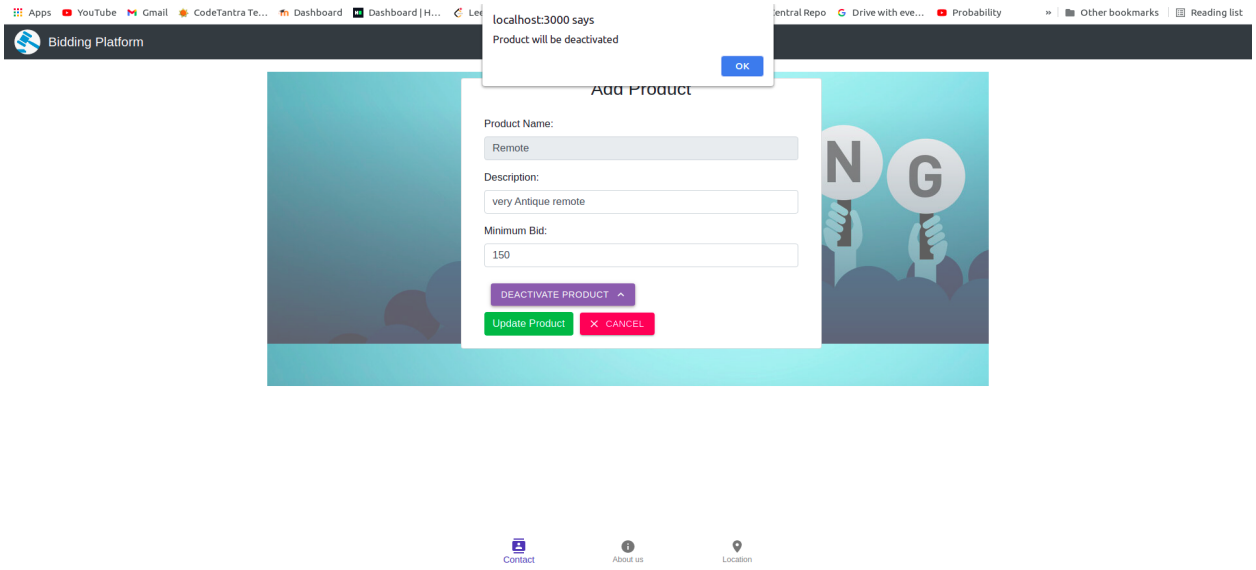

Fig-37 : Updating product details

8.1.6. Bid Product Page : The Bid product page allows the user to bid for the product that he wishes. When the user clicks on the bid button in the product list page, he gets to the bid product page. Now there is a minimum bid set below which the user can not bid. If the user bids less than or equal to the minimum bid set by the user then the frontend would give an alert and the bid would not be able to set.

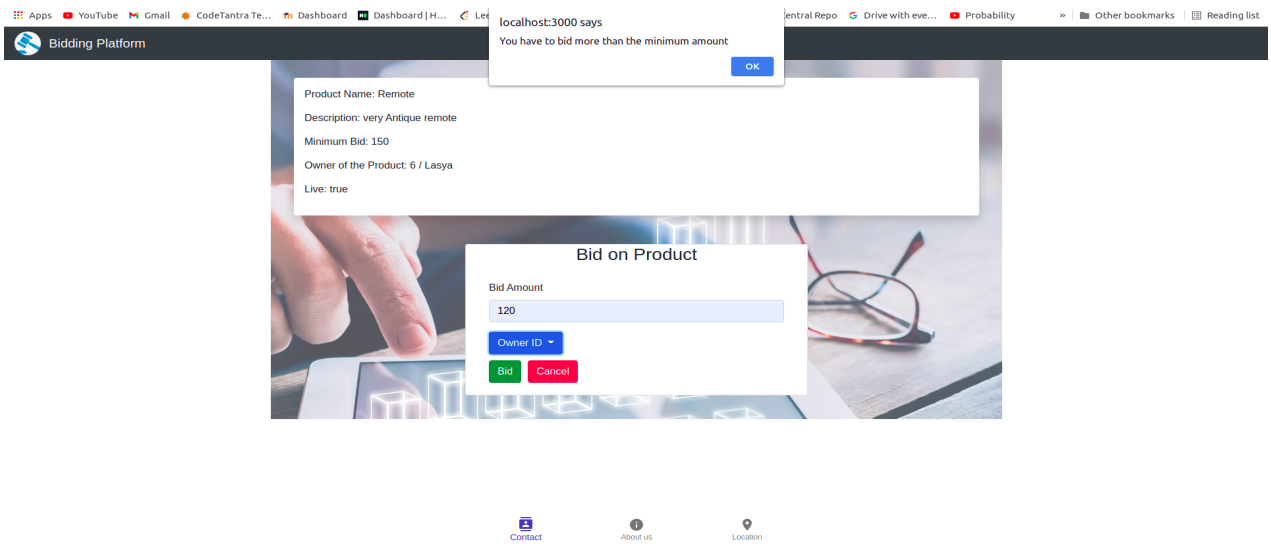

Fig-38 : Bidding for a product

8.2. Additional functionalities :

- *Can not bid on inactive products:*

On the product details section, there are all the bids visible to the user out of which the user has to select the bid that he wishes. Now once the bid has been selected that means the product has been sold out. Now no more bids can be placed on the product and the product becomes inactive.

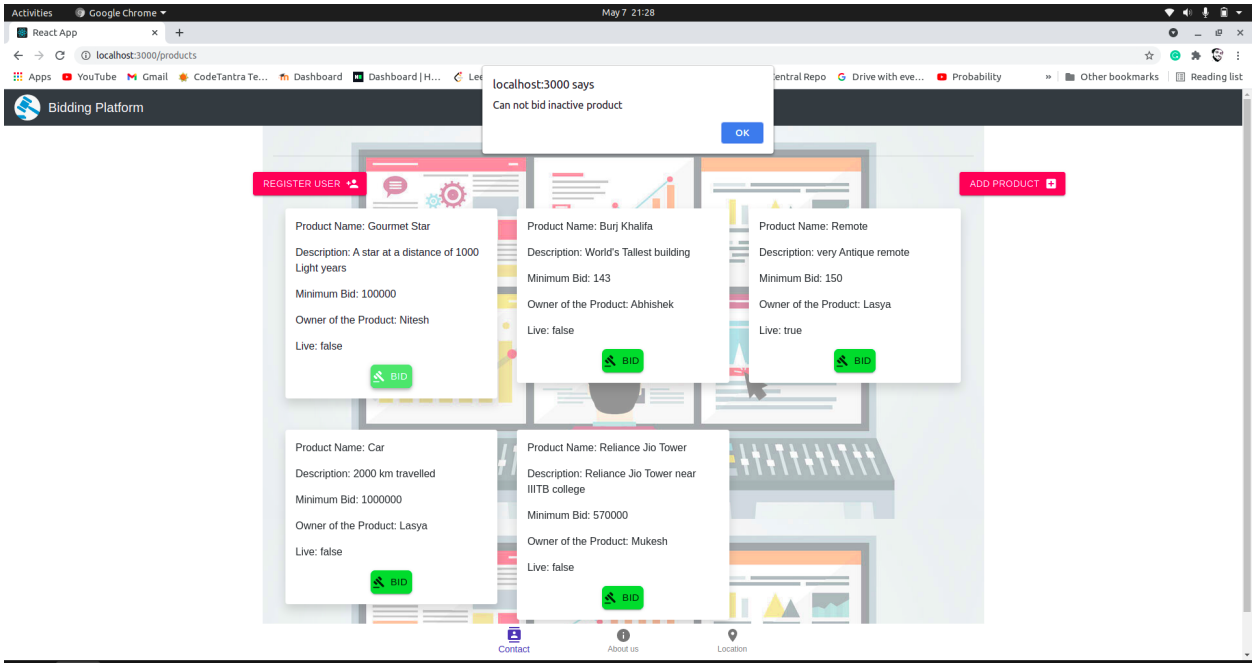

Fig-39 : No bidding on inactive products

- *Contact details at the bottom of the page:*

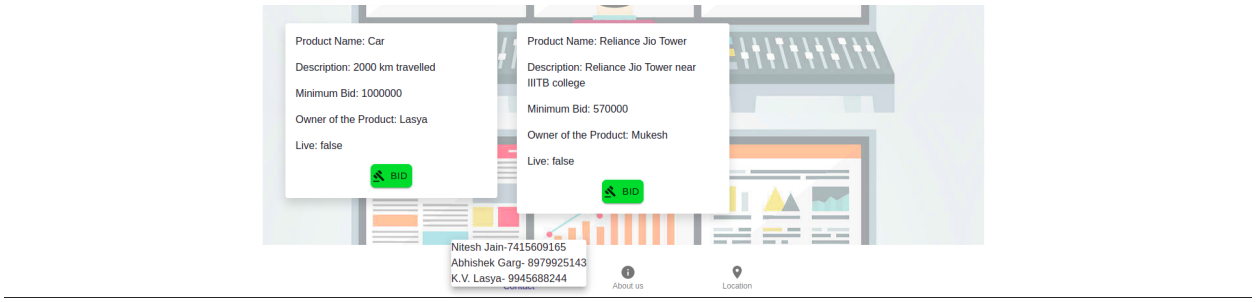

Fig-40 : Contact details and About us (Footer bar)

8.3. GitHub and Docker Link :

GitHub Account :

https://github.com/niteshjan/Bidding-Platform

Front End :

https://hub.docker.com/repository/docker/niteshjan/react-backend-bp

Back End :

https://hub.docker.com/repository/docker/niteshjan/spring-backend-bp

8.4. Limitations of the project :

- Only registered users can do an auction on the website and registered users can place bids on the website.
- Users can only do auctions on given categories.
- Users can not create their own series of categories.

8.5. Future Scope :

In this digital era, companies prefer to fetch business through online platforms. Therefore, they tend to approach their existing and new clients via the internet source, so it is the responsibility of an online bidder to search the clients who are interested in availing the services offered by you.Moreover, while working as an online bidder, you will get an opportunity to upgrade your knowledge about recent technologies.

This portal gives the option to users to sell and purchase a product online and can have so many options to sell and purchase for example: users can use this platform to sell their properties. At this time, every person has a shortage of time so users can sell and purchase the product online.In particular, some research areas need more investigation, especially bidder behavior in B2B reverse auctions, and online auction design that shows how technology can be incorporated into auction design to affect the price paid by bidders. Also, more seller-level analysis, especially in the context of marketing research, could be performed to show how seller-level effects, such as promotion, market presence, etc., can affect the price paid in online auctions.

9. References

- https://www.wikipedia.org/
- https://spring.io/projects/spring-boot
- https://junit.org/junit5/
- https://maven.apache.org/
- https://reactjs.org
- https://material-ui.com/
- https://www.jenkins.io/doc/
- https://docs.docker.com/
- https://www.ansible.com/
- https://www.elastic.co/what-is/elk-stack
- https://docs.microsoft.com/en-us/azure/mysql/connect-workbench
- https://docs.microsoft.com/en-us/azure/architecture/aws-professional/compute
- https://core.ac.uk/download/pdf/6784807.pdf
- https://www.researchgate.net/publication/310615150_An_effective_online_auction_system