

International Institute of Information Technology, Bangalore

Software Production Engineering

Project Elective Report

JobSeek

Under the Guidance of

Prof. B. Thangaraju

Teaching Assistant: Yashasvi Khandelwal



HIMANK JAIN

MT2021054

MUGDHA AGARWAL

MT2021079

Table of Contents

1. ABSTRACT.....	3
2. INTRODUCTION.....	4
2.1. Overview.....	4
2.2. Features.....	4
2.3. Why DevOps?.....	5
2.4. DevOps Features.....	5
3. SYSTEM CONFIGURATION.....	6
3.1. Operating System.....	6
3.2. CPU and RAM.....	6
3.3. Languages/Technologies.....	6
3.4. Database.....	6
3.5. Build Tools.....	6
3.6. DevOps Tools.....	6
4. Software Development Life Cycle.....	7
4.1. Installation.....	7
4.1.1 IDE.....	7
4.1.2 Java Setup.....	7
4.1.3 Jenkins.....	8
4.1.4 Git.....	8
4.1.5 Docker.....	8
4.1.6 Ansible.....	9
5. Source Control Management.....	10
6. Maven Build.....	11
7. Testing.....	11
8. Continuous Integration – Jenkins.....	13
9. Docker Containerisation.....	17
10. Continuous Deployment.....	18
11. Running Application on Remote Machine.....	21
12. Monitoring.....	22
13. ARCHITECTURE DIAGRAMS.....	27
13.1. Database.....	27
13.2. Seeker Workflow.....	28
13.3. Recruiter Workflow.....	29
14. API docs.....	30
15. Results and Discussion.....	31
15.1. Homepage.....	31
15.2. Register Page.....	32
15.3. Login.....	34
15.4. Recruiter's View.....	35
Dashboard.....	35
15.5. Seeker's View.....	43
Dashboard.....	43
16. Conclusion.....	47
17. Scope for Future Work.....	47
18. References.....	48

1. ABSTRACT

In today's busy, competitive and fast moving world, getting a right job and getting best employees for that job is nearly impossible. As we know we run from one platform to other platform in order to find the right job. **JobSeek** is an online job finding and posting platform from where a seeker can easily find a job and with just a click of button can easily apply for it whereas a recruiter can easily post a job and can hire the employees.

The objective of this application is to bring the seeker and recruiter to same platform so as to ease the whole process right from posting a job to finding the jobs and thereby hiring best candidates.

The architecture of our project demands three layers:

- Front end
- Middle Layer
- Back end

The front end of the application is handled by HTML, CSS, BOOTSTRAP.

The Back end of the application is handled by Spring Boot for the interaction of database and front end.

The Middle Layer is our SQL Database.

GitHub Repository: <https://github.com/himankjn/MajorProj.git>

DockerHub Repository: <https://hub.docker.com/u/jobseek>

2. INTRODUCTION

2.1. Overview

JobSeek is an online web application which allows seekers and recruiters to share a common platform. The seeker can register themselves and see all the jobs with the facility of filtering jobs as well, can apply for jobs and even list it as favorites. The recruiter can register themselves and can post as many jobs as they want. The candidates who would have applied for their job will be shown to the recruiter so that he can proceed with the further procedure. The recruiter can even update the posted jobs and can even delete it.

The application also provides them the facility to update their profile.

2.2. Features

- *Seeker*
 - *User Profile*
 - *Dashboard*
 - *Search All Jobs (either based on filter or as a whole)*
 - *See All Interested Jobs*
 - *See the Job Profile*
 - *Apply for the Job*
 - *Mark it as favorite / remove it from favorites*
- *Recruiter*
 - *User Profile*
 - *Dashboard*
 - *Job Posting*
 - *Update/Delete a posted Job*
 - *See all the Posted Jobs*
 - *See all the Applicants for that job*
 - *See the Application Status of the Applicant*

2.3. Why DevOps?

The classic problem “finger point” between Dev and Ops. The problem stated that the developers and operators usually blame that “It’s not my code, it’s your machines!” or “it is not my machine, it is your code created the problem!”

The only possible way to build ,test and deploy a new workable software is to make development and operations transparent and integrated.

DevOps is a software development approach through which superior quality software can be developed quickly and with more reliability. DevOps tools help to automate the process of continuous integration and continuous deployment i.e. CI/CD pipeline.

It gives us the opportunity to add new features as and when required and helps to easily deploy it into production environment.

It consists of various stages such as continuous development, continuous integration, continuous testing, continuous deployment, and continuous monitoring.

2.4. DevOps Features

DevOps spans the entire delivery pipeline thus helps in:

- Improved deployment frequency
- Shortened leadtime between fixes
- Faster Time to Market
- Lower Failure rate of new release
- Faster mean time to recovery in the event of new release crashing
- Greater scalability and availability

3. SYSTEM CONFIGURATION

3.1. Operating System

Ubuntu 20.04.3

3.2. CPU and RAM

4 core processor i5-1135G7 and RAM 8 GB

3.3. Languages/Techologies

HTML, CSS, BOOTSTRAP, SpringBoot

3.4. Database

MySQL 8.0.29 for Linux

3.5. Build Tools

Maven for spring application

3.6. DevOps Tools

- Source Control Management - GitHub
 - Continuous Integration - Jenkins
 - Containerization - Docker
 - Testing- Junit 5 Unit testing
 - Continuous deployment - Ansible
 - Log creation - log4j
 - Monitoring and Visualization: ELK stack (elastic search, logstash,kibana)
-

4. Software Development Life Cycle

4.1. Installation

4.1.1 IDE

The project was developed on IntelliJ IDEA : an *Integrated Development Environment (IDE)* for JVM languages designed to maximize developer productivity. It does the routine and repetitive tasks for you by providing clever code completion, static code analysis, and refactorings, and lets you focus on the bright side of software development, making it not only productive but also an enjoyable experience.

We have used java for developing spring application and maven to build it.

Maven version: Apache Maven 3.6.3

We have used IntelliJ IDEA for development.

Spring Initializer was used to initialize the spring project.

`$mvn clean install` : was used to build the code as it creates a war file in target directory and using this war file we run the spring project.

4.1.2 Java Setup

We have used JDK 11 for this project. The following steps were followed to install JDK 11 in the system:

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

```
mugdha@mugdha:~$ java --version  
openjdk 11.0.15 2022-04-19  
OpenJDK Runtime Environment (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1)  
OpenJDK 64-Bit Server VM (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1, mixed mode, sharing)  
mugdha@mugdha:~$ █
```

Figure 1: java version

For switching into different java versions we can write the following command:

```
$ sudo update-alternatives --config java
```

```
mugdha@mugdha:~$ sudo update-alternatives --config java  
[sudo] password for mugdha:  
There are 2 choices for the alternative java (providing /usr/bin/java).  
  
Selection Path Priority Status  
-----  
* 0 /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111 auto mode  
1 /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111 manual mode  
2 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java 1081 manual mode  
  
Press <enter> to keep the current choice[*], or type selection number: 0  
mugdha@mugdha:~$ █
```

Figure 2: Configuration of java version

4.1.3 Jenkins

Jenkins is a free and open-source automation server which is used to automate all types of software development tasks such as building, testing, and delivering or deploying software.

We have used Jenkins as an integration tool. It is a server-based system that runs in servlet containers such as Apache Tomcat.

The following steps were followed to install Jenkins in our localhost:

```
$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'
$ sudo apt-get update
$ sudo apt-get install jenkins
```

In order to run the jenkins, open web browser: localhost:8080 and login as admin and then copy the passwd from */var/lib/jenkins/secrets/initialAdminPassword* and paste it in the jenkins login page and create user account.

4.1.4 Git

It is a free and open source distributed version control system designed to handle everything from small to large objects with speed and efficiency. It is used to maintain Linux Kernel and makes collaborations as quick and painless as possible. As developers working on our own project, we can keep track of changes made in our source code using git terminal tool.

The following steps were followed to download command line tool for ubuntu:

```
$ sudo apt update
$ sudo apt install git
```

4.1.5 Docker

Docker is an open source tool that helps in management of containers. Docker Automates the development of applications inside software containers. It implements high level API to provide lightweight containers that run processes in isolation. By using containers, resources can be isolated, services restricted and process to have private view of OS. It is used for application development, testing, packaging and deployment.

The following steps were followed to install docker:

```
$ sudo usermod -aG docker Jenkins
$ sudo less /etc/gshadow | grep jenkins
```

```
mugdha@mugdha:~$ sudo less /etc/gshadow | grep jenkins
[sudo] password for mugdha:
adm:*::syslog,mugdha,jenkins
sudo:*::mugdha,jenkins
jenkins:!:!
mugdha@mugdha:~$
```

The following commands were used to install docker-compose

```
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
$ docker-compose version
```

```
mugdha@mugdha:~$ docker-compose version
docker-compose version 1.29.0, build 07737305
docker-py version: 5.0.0
CPython version: 3.7.10
OpenSSL version: OpenSSL 1.1.0l 10 Sep 2019
mugdha@mugdha:~$
```

Figure 3: docker-compose version

4.1.6 Ansible

Ansible is an IT automation tool that can configure systems, deploy software, and manage IT tasks like continuous deployments. It works by connecting to hosts on a network and sending Ansible modules to these hosts. Ansible is simple and is easy to learn as well as it is agentless that is for communication it uses standard SSH protocol and also provides cross platform support.

The following steps were followed to install ansible:

```
$ sudo apt install openssh-server
$ sudo apt update
$ sudo apt install ansible
$ ansible --version
```

```
mugdha@mugdha:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['~/home/mugdha/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Mar 15 2022, 12:22:08) [GCC 9.4.0]
mugdha@mugdha:~$
```

Figure 4: Ansible Version

In order to communicate with the managed hosts, we need to install OpenSSH

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

5. Source Control Management

We have started working with git by creating a repository on GitHub and cloning it, and then we can push our source code into the remote repository or can pull the changed code into our local repository. All the collaborators can push the changes in the github.

Git commands:

- git clone - This command copies the entire data on the git url
- git checkout -b - This command creates a new branch with the ‘branch_name’
- git add - 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 -This command is used to integrate changes from another branch.
- git push-This command will push all the latest code to the repository.

The screenshot shows a GitHub repository page for 'himankjn/MajorProj'. The top navigation bar includes links for Search or jump to..., Pull requests, Issues, Marketplace, and Explore. The repository name 'himankjn/MajorProj' is shown as private. Below the header, there are tabs for Code, Issues, Pull requests, Actions, Projects, Security, and Insights, with 'Code' being the active tab. A message indicates the branch is up to date with master. The main content area displays a list of recent commits from 'himankjn' with details like commit message, author, date, and file changes. To the right, sections show repository statistics (40 commits, 1 star, 1 watching, 0 forks), releases (none), packages (none), contributors ('himankjn' and 'mugdha98'), and languages (Java 99.1%, Other 0.9%). A detailed description of the project is provided in the README.md file, mentioning a Spring Job Portal developed with Spring and Hibernate, and instructions for installing and running the application using Docker and Jenkins.

6. Maven Build

Maven is based around the central concept of a build lifecycle, that is, it is the process for building and distributing a particular artifact (project) is clearly defined. For the person building a project, this means that it is only necessary to learn a small set of commands to build any Maven project, and the POM will ensure they get the results they desired.

We used maven to build our project as it can resolve dependencies to the code. We started with spring initializer and imported the project in local environment. We added necessary Dockerfile which would be used to create docker image of the build. The docker file tells the build should be built on what image (here it is openjdk 11, after that we copy the created jar file and copy it to the working directory) and what command should run when container is running. For unit testing we have included junit dependency and written test cases to test while we build our project. We added manifest within pom.xml which tells java to where to look for main class in project for an executable war.

The following steps were followed to install Maven:

```
$ sudo apt-get update
$ sudo apt-get install maven
$ mvn -version
mugdha@mugdha:~/Documents/Sem_II/MajorProj$ mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.15, vendor: Private Build, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "5.13.0-40-generic", arch: "amd64", family: "unix"
mugdha@mugdha:~/Documents/Sem_II/MajorProj$
```

Figure 5: Maven version

The command used for building the project is **\$mvn clean install**

7. Testing

JUnit is an open source Unit Testing Framework for JAVA. 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 test first before any 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 makes sure nothing is broken. Add junit dependency in pom.xml in below way to use Junit.

```
<dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
    <version>5.8.2</version>
    <scope>test</scope>
</dependency>
```

Figure 6: Junit dependency

Command-

```
$ mvn clean test
```

For our project we have created 4 test files for the DAO classes where we are checking whether each and every functionality given in the class is working perfectly or not.

Explaining one testing function out of all the impleteded test functions:

```
24 ► public class CompanyDaoTest {  
25     @Autowired  
26     private CompanyDao companyDao;  
27     @Autowired JobPostingDao jp;  
28     @Test  
29     @Transactional  
30     @Rollback(true)  
31 ►     public void passwordLookUp() throws Exception {  
32         Company c=new Company();  
33         c.setVerified(true);  
34         c.setVerificationCode(1234);  
35         c.setCompanyName("testcomp1");  
36         c.setCompanyUser("testcomp1@gmail.com");  
37         c.setPassword("testpswd");  
38         c.setHeadquarters("testloc1");  
39         companyDao.createCompany(c);  
40  
41         String actual=companyDao.PasswordLookUp( emailid: "testcomp1@gmail.com").get(0);  
42         String expected="testpswd";  
43         assertEquals(expected,actual, message: "checking password fucntionality");  
44     }  
45 }
```

Figure 7: checking password lookup test in CompanyDaoTest

In the given test we have checking that by giving the email the function PasswordLookUp implemented in the DAO class is returning the correct password or not. And after completeing this test we are rolling back this transaction as we have just used it for testing and its not a real data.

If the function has not worked properly that is the expected and actual result do not match then it will throw an error.

8. Continuous Integration – Jenkins

For our project we have used Jenkins tools to build our pipeline.

Jenkins starts at port number 8080, so login on to <http://localhost:8080> onto the browser and login with your credentials.

- Go to manage plugins and download Build pipeline plugin, Docker plugin, GitHub, Maven integration plugin, Ansible Pulgin.

The screenshot shows the Jenkins Plugin Manager interface. At the top, there is a search bar containing the text "docker". Below the search bar, there are tabs for "Updates", "Available", "Installed" (which is selected), and "Advanced". A table lists several Jenkins plugins:

Name	Enabled
Docker 1.2.6 This plugin integrates Jenkins with Docker Report an issue with this plugin	<input checked="" type="checkbox"/> X
Docker API 3.1.5.2 This plugin provides docker-java API for other plugins. Report an issue with this plugin	<input checked="" type="checkbox"/> X
Docker Commons Plugin 1.19 Provides the common shared functionality for various Docker-related plugins. Report an issue with this plugin	<input checked="" type="checkbox"/> X
Docker Pipeline 1.28 Build and use Docker containers from pipelines. Report an issue with this plugin	<input checked="" type="checkbox"/> X

Figure 8: Docker Plugin

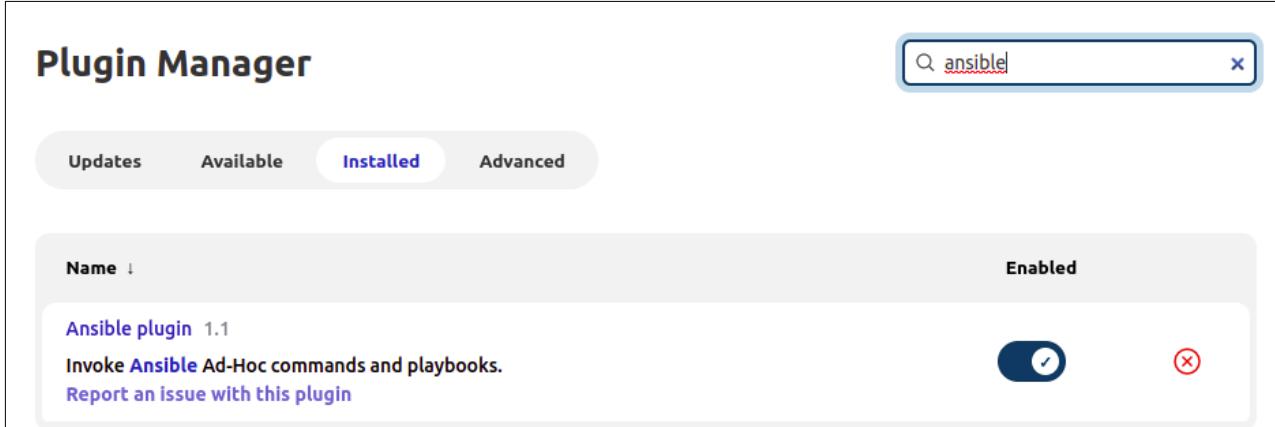


Figure 9:Ansible Plugin

- Add the docker credentials in the jenkins by going to manage plugin and then to credentials . Click on ADD Credentials.

Dashboard > Credentials > System > Global credentials (unrestricted) > jobseek/******** (dockerhub credential for jobseek account)

Back to Global credentials (unrestricted) Scope ?
Update Global (Jenkins, nodes, items, all child items, etc)
Delete Username ? jobseek
Move Treat username as secret ?
Password ? Concealed Change Password
ID ? dockerhubjobseek
Description ? dockerhub credential for jobseek account
Save

Figure 10:dockerhub credentials

- Add Ansible configuration in Global Configuration Management tool.

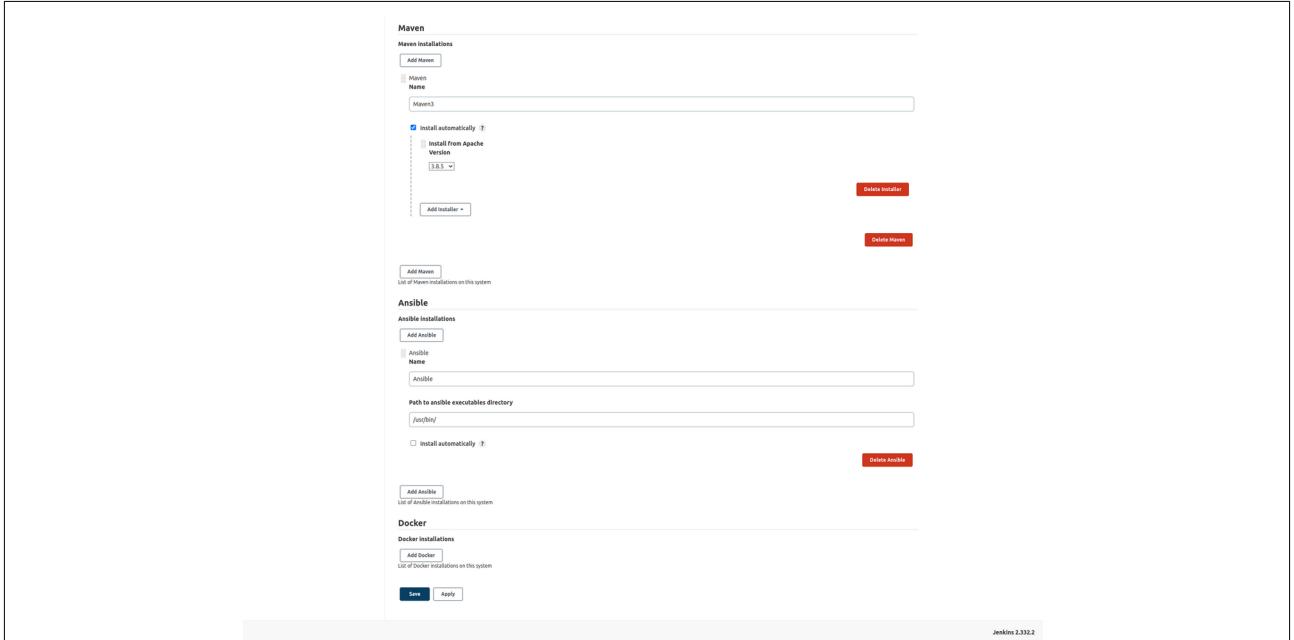
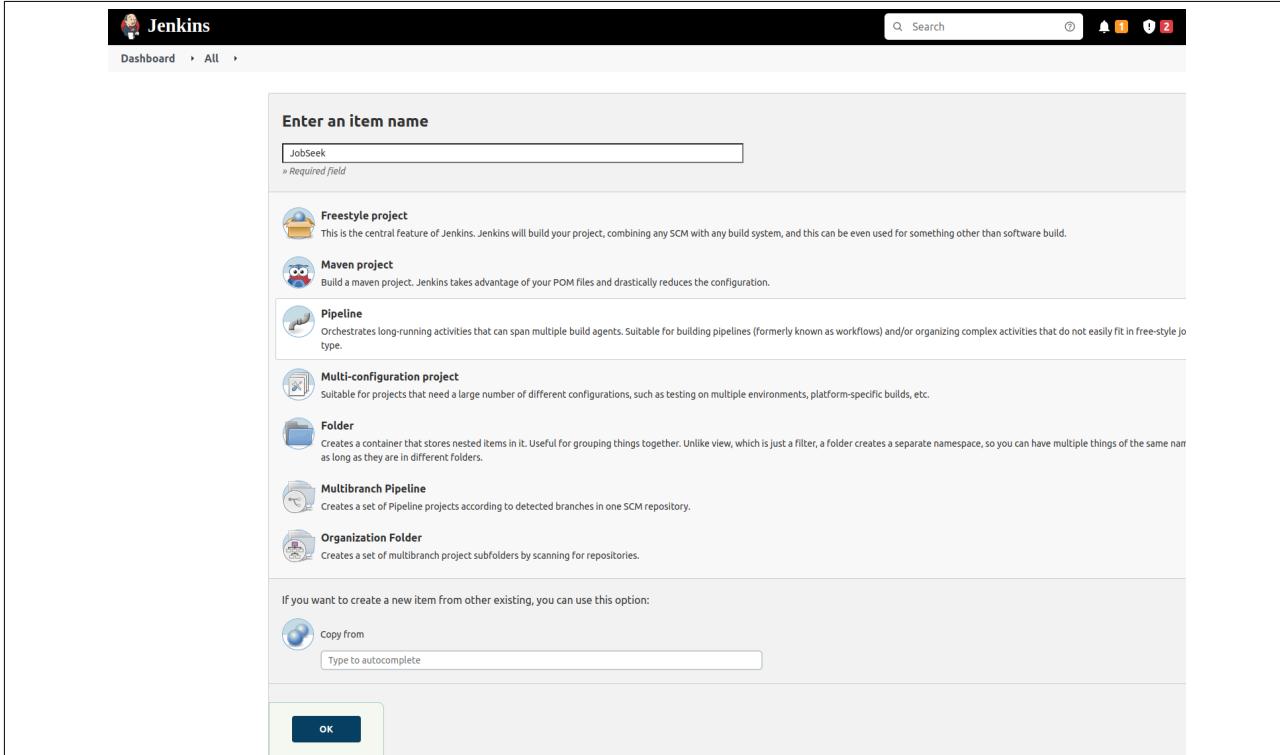


Figure 11:Global Tool Configuration

- Click on new Item and enter the name and click on pipeline and OK

S	W	Name	Last Success	Last Failure	Last Duration
		Ansible_Jenkins	1 mo 13 days #3	1 mo 13 days #2	20 sec
		Calculator	2 mo 28 days #5	2 mo 28 days #1	67 ms
		Calculator_Mini_Project_Pipeline	10 days #65	10 days #66	1 min 1
		CPU_Info_Job_Chaining	2 mo 7 days #1	N/A	67 ms
		Day1lab	3 mo 6 days #1	N/A	0.14 sec
		Day2lab_EmailNotification	3 mo 6 days #3	3 mo 6 days #2	3 sec



- Write down the pipeline stages to pull the code from github, build and test using maven, converting the war file into docker image, pushing the same onto docker hub and pulling the image on remote server using ansible and running it. Click on Apply and Save after writing the pipeline.

Dashboard > JobSeek >

General	Build Triggers	Advanced Project Options	Pipeline
<pre> 1 * pipeline { 2 // The "agent" section configures on which nodes the pipeline can be run. 3 // Specifying "agent any" means that Jenkins will run the job on any of the 4 // available nodes. 5 agent any 6 7 environment { 8 registry = "jobseek/jobportal" 9 registryCredential = 'dockerhubjobseek' 10 dockerImage = '' 11 } 12 13 stages { 14 stage('Git Pull') { 15 steps { 16 // Get code from a GitHub repository 17 // Make sure to add your oage022222222211111wn git url and credentialsId 18 git url: 'https://github.com/hlmanjkjn/MajorProj.git', branch: 'master',credentialsId:"newgitcred" 19 } 20 } 21 stage('Maven Build') { 22 steps { 23 // Maven build 24 sh 'mvn clean install' 25 } 26 } 27 stage('Building our Image') { 28 steps { 29 script { 30 dockerImage = docker.build registry + ":latest" 31 } 32 } 33 } 34 stage('Deploy our image') { 35 steps { 36 script { 37 docker.withRegistry('', registryCredential) { 38 dockerImage.push() 39 } 40 } 41 } 42 } 43 } 44 stage('Ansible Deploy') { 45 steps { 46 //Ansible Deploy to remote server (managed host) 47 ansiblePlaybook colorized: true, disableHostKeyChecking: true, installation: 'Ansible', inventory: 'inventory', playbook: 'playbook' 48 } 49 } 50 } 51 } 52 }</pre>			
<input type="button" value="Save"/> <input type="button" value="Apply"/>			

Figure 12:Pipeline Script

- Now Build the stages in pipeline by clicking on Build Now.

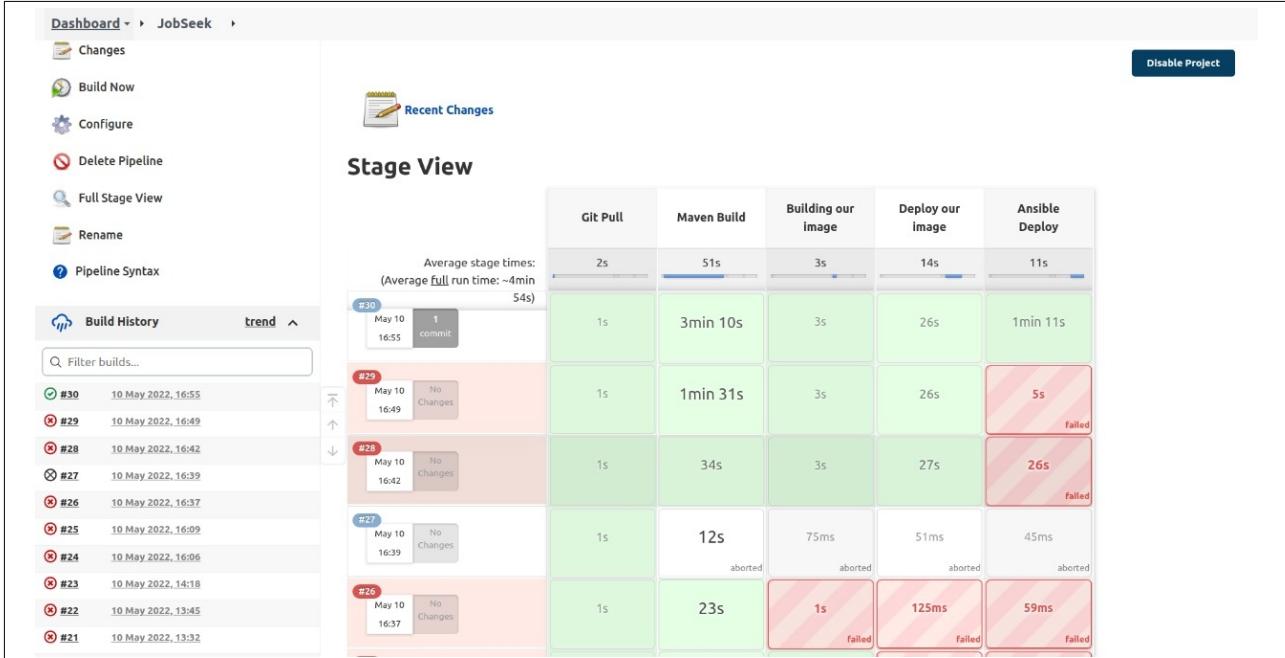


Figure 13: Stage View

9. Docker Containerisation

Containerization is a very important step in DevOps as it lets us deploy our application in a portable manner by packaging the application along with all its dependencies, in an isolated, lightweight environment, called as Container. A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI. You can connect a container to one or more networks, attach storage to it, or even create a new image based on its current state. A container is defined by its image as well as any configuration options you provide to it when you create or start it. When a container is removed, any changes to its state that are not stored in persistent storage disappear.

For this project, we have used Docker as the containerization tool as it packages the applications in a virtual container runnable on Linux server. Docker also provides an online repository called Docker Hub for storing the docker images.

Docker builds images automatically by reading the instructions from a Dockerfile which is a text file that contains all commands, in order, needed to build a given image. A Dockerfile adheres to a specific format and set of instructions which you can find at Dockerfile reference.

For our project we have used only one Dockerfile,

```

1  FROM openjdk:8
2  ADD ./target/jobboard-0.0.1-SNAPSHOT.war ./jobportal.war
3  ENTRYPOINT ["java", "-jar", "jobportal.war"]
4  EXPOSE 8095
5

```

Figure 14: Dockerfile

Following steps are performed in this Dockerfile to create the docker image:

- The base image which we have used for this project is openjdk:8
 - We then added the war file from the current directory of localhost to the docker hub repository with name jobportal.war
 - An ENTRYPOINT allows you to configure a container that will run as an executable.
 - The EXPOSE instruction indicates the ports on which a container listens for connections.
-

10. Continuous Deployment

Continuous Deployment is a software engineering approach in which software functionalities are delivered frequently through automated deployment. The goal of Continuous Deployment is building, testing and releasing software with greater speed and frequency.

For this project, we have used Ansible as Continuous Deployment tool. Ansible is open-source and agentless. Apart from being CD tool, it is also a Configuration Management tool, that aims at making sure the system remains in desired and consistent state with each change made in the system. Ansible contains an automation engine that runs something called as Ansible Playbooks. The Ansible Playbook contain set of defined tasks as well as environments and workflows definition. Other than the playbook, we also write a file called as Inventory. Ansible contain two set of nodes – control node (a machine where ansible is installed and run) and managed nodes (set of machines where ansible deploys the software). So, the Inventory file contain the list of these managed nodes.

The ansible playbook.yml file for our project looks like this:

```

1      ---  

2      |   - name: dockercompose for mysql and backend frontend  

3      |   hosts: all  

4      |   tasks:  

5      |       - name: Start docker service  

6      |           service:  

7      |               name: docker  

8      |               state: started  

9      |  

10     |       - copy:  

11     |           src: docker-compose.yml  

12     |           dest: ./docker-compose.yml  

13     |           owner: mugdha  

14     |           group: mugdha  

15     |           mode: 0644  

16     |  

17     |       - name: deploy Docker Compose  

18     |           docker_compose:  

19     |               project_src: ./  

20     |               files:  

21     |                   - docker-compose.yml

```

Figure 15: playbook.yml file

According to playbook.yml file:

- **host: all** our application can be deployed on any machine.
- **Start docker service** it is used so that if by any means the docker service has stopped it will start it again and if it's already running then it will ignore this task
- **copy** this task is defining that we have to copy the docker-compose.yml file from the host machine to the remote machine whose owner and group is mugdha and the permissions for this compose file will be 0644 and it will be copied in the current folder of the remote machine
- **deploy Docker Compose** it is telling to run the docker compose file from the current directory.

Docker Compose is a tool that was developed to help define and share multi-container applications. With Compose, we can create a YAML file to define the services and with a single command, can spin everything up or tear it all down. The big advantage of using Compose is you can define your application stack in a file, keep it at the root of your project repo (it's now version controlled), and easily enable someone else to contribute to your project. Someone would only need to clone your repo and start the compose app.

```

1  version: "2.1"
2
3  services:
4    mysql-jobportal:
5      container_name: mysql-jobportal
6      image: mysql
7      environment:
8        - MYSQL_ROOT_PASSWORD=password
9        - MYSQL_DATABASE=jobportal
10     volumes:
11       - mysql_data_volume:/var/lib/mysql
12     networks:
13       - jobnet
14     ports:
15       - "3307:3306"
16     healthcheck:
17       test: [ "CMD", "mysqladmin" , "ping", "-h", "localhost" ]
18       timeout: 10s
19       retries: 10
20    jobportal:
21      container_name: jobportal
22      image: himankjn/jobportal
23      ports:
24       - "8095:8095"
25     networks:
26       - jobnet
27     environment:
28       SPRING_DATASOURCE_URL: jdbc:mysql://mysql-jobportal:3306/jobportal?autoReconnect=true&useSSL=false&allowPublicKeyRetrieval=true&createDatabaseIfNotExist=true
29
30   depends_on:
31     mysql-jobportal:
32       condition: service_healthy
33
34   networks:
35     jobnet:
36
37   volumes:
38     mysql_data_volume:

```

Figure 16: docker-compose.yml file

Explaining docker-compose file:

- The version we are using for our project of the docker-compose is 2.1
- Services tells us which kind of docker service we want to create/execute.
 - We are creating the **service mysql-jobportal** for our project.
 - The container which we will be using for this service is mysql-jobportal
 - The image is mysql if it is not present in our local system then it will pull the image from dockerhub
 - The environment is same which we have used in our project to connect to database
 - the root password and the database name should match with the name given in the project
 - To have our data persistent we have used volumes
 - To run both the containers in the same network we have used networks: jobnet
 - We have mapped the port from 3306 of container to 3307 of localhost because 3306 of localhost is already running its own local mysql server.
 - Healthcheck is ensure that our container is healthy.
 - We have also created another **service jobportal** which contains our backend and frontend
 - The container which we will be using for this service is jobportal

- The image is himankjn/jobportal if it is not present in our local system then it will pull the image from dockerhub
- We have mapped the port from 8095 of container to 8095 of localhost.
- To run both the containers in the same network we have used networks: jobnet
- SPRING_DATASOURCE_URL is an environment URL which is used to specify where the mysql server is running so that the container can connect to the database.
- This service is dependent on mysql-jobportal service once the mysql-jobportal service is executed then only this service will start executing.

Also, because, we are deploying our application on the remote machine. So, our inventory file contains just one entry that is of remote server:

```

1 [ubuntu18]
2 172.16.144.179 ansible_user=mugdha
3

```

Figure 17: inventory file

11. Running Application on Remote Machine

```
mugdha@mugdha:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7c578ab55c86 himankjn/jobportal "java -jar jobportal_"
cd4418cf8eb2 mysql "docker-entrypoint.s..." 5 hours ago Up 3 hours 0.0.0.0:8095->8095/tcp, :::8095->8095/tcp 0.0.0.0:3306/tcp, 0.0.0.0:3307->3306/tcp, :::3307->3306/tcp mysql-jobportal
mugdha@mugdha:~$
```

Figure 18:Remote server machine

```

jenkins@himank:~$ 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:ZwBcEyEzB1La6jojt8o5g54VR2a0IluHBB5r/xRuVE jenkins@himank
The key's randomart image is:
+---[RSA 3072]---+
|+++.+++
|++ . ++
|+ . + +
|+ . + oE
|+= o . ooo
|=.. . +$ o
|=+ .. oo
|oo+ . .
|+ . +
|+ . .
+---[SHA256]---+
jenkins@himank:~$ ssh-copy-id mugdha@172.16.144.179
/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
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
mugdha@172.16.144.179's password:
Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'mugdha@172.16.144.179'"
and check to make sure that only the key(s) you wanted were added.
jenkins@himank:~$
```

Figure 19:Control Server Machine

12. Monitoring

Continuous monitoring is a technology and process that IT organizations use to get near-immediate feedback and insight into performance and interactions across the network, which helps drive operational, security and business performance. Continuous monitoring, sometimes referred to as ConMon or Continuous Control Monitoring (CCM) provides security and operations analysts with real-time feedback on the overall health of IT infrastructure, including networks and applications deployed in the cloud.

Continuous monitoring tool is ELK (Elastic search logstash kibana)

Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch. Elasticsearch is an open source, full-text search and analysis engine, based on the Apache Lucene search engine. Logstash is a log aggregator that collects data from various input sources, executes different transformations and enhancements and then ships the data to various supported output destinations. Kibana is a visualization layer that works on top of Elasticsearch, providing users with the ability to analyze and visualize the data. And last but not least — Beats are lightweight agents that are installed on edge hosts to collect different types of data for forwarding into the stack.

To Generate Log File

- create a file name log4j2.xml under src/main/resources.
- Enter the format in which log should be generated.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Configuration status="INFO">
3      <Appenders>
4          <Console name="ConsoleAppender" target="SYSTEM_OUT">
5              <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n" />
6          </Console>
7          <File name="FileAppender" fileName="jobseeklogs.log" immediateFlush="false" append="true">
8              <PatternLayout pattern="%d{yyy-MM-dd HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
9          </File>
10     </Appenders>
11     <Loggers>
12         <Root level="debug">
13             <AppenderRef ref="ConsoleAppender" />
14             <AppenderRef ref="FileAppender" />
15         </Root>
16     </Loggers>
17 </Configuration>
```

Figure 20:log4j2

- Add log4j2 dependency in pom.xml
- Upon executing command mvn clean install , check for logs in calculator.log file.
- For Better Visualization of Logs ,Use ELK Stack.
- Drop the generated calculator.log file in <https://www.elastic.co>

The screenshot shows the Elastic Cloud interface with the following sections:

- Elasticsearch Service**: Shows a deployment named "JobSeek" in a "Healthy" state, version 8.2.0, in the GCP - Iowa (us-central1) cloud region.
- Documentation**: Includes links to Elastic documentation, indexing data into Elasticsearch, and the Elasticsearch REST API.
- Community**: Features a "Community" section with links to ElasticON events, AWS Summit Stockholm, and Daily Elastic Byte - Elastic Cloud Edition.
- Support**: Provides a search bar, links to Elastic documentation, indexing data into Elasticsearch, and the Elasticsearch REST API, and a "Contact support" button.
- Cloud status**: Shows "All systems operational".
- News**: Lists recent news items: "What's new in Elastic Security 8.2: Streamline analyst workflows with context and expertise" (MAY 3, 2022), "Elastic Enterprise Search 8.2: Relevance controls for Elasticsearch" (MAY 3, 2022), and "Secure your cloud with Cloud Workload Protection in Elastic Security" (MAY 3, 2022).
- Training**: Encourages users to start free training, mentioning the Elastic Learning Portal.

Welcome home

The screenshot shows the Elastic Home page with a dark header bar. The main title "Welcome home" is centered above four colored cards. Each card has an icon and a brief description:

- Enterprise Search**: Create search experiences with a refined set of APIs and tools.
- Observability**: Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.
- Security**: Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.
- Analytics**: Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Below these cards is a section titled "Get started by adding integrations". It includes a callout for "Add Integrations" and links to "Try sample data" and "Upload a file". To the right is a decorative graphic of a house with various data visualization elements (charts, bars) floating around it.

Management

The "Management" section contains four sub-links:

- Manage permissions**: Control who has access and what tasks they can perform.
- Monitor the stack**: Track the real-time health and performance of your stack.
- Back up and restore**: Save snapshots to a backup repository and restore to.
- Manage index lifecycles**: Define lifecycle policies to automatically perform.

Dev Tools and **Stack Management** links are also visible at the top of this section.

More ways to add data

In addition to adding integrations, you can try our sample data or upload your own data.

Sample data **Upload file** (selected)

Visualize data from a log file

Upload your file, analyze its data, and optionally import the data into an Elasticsearch index.

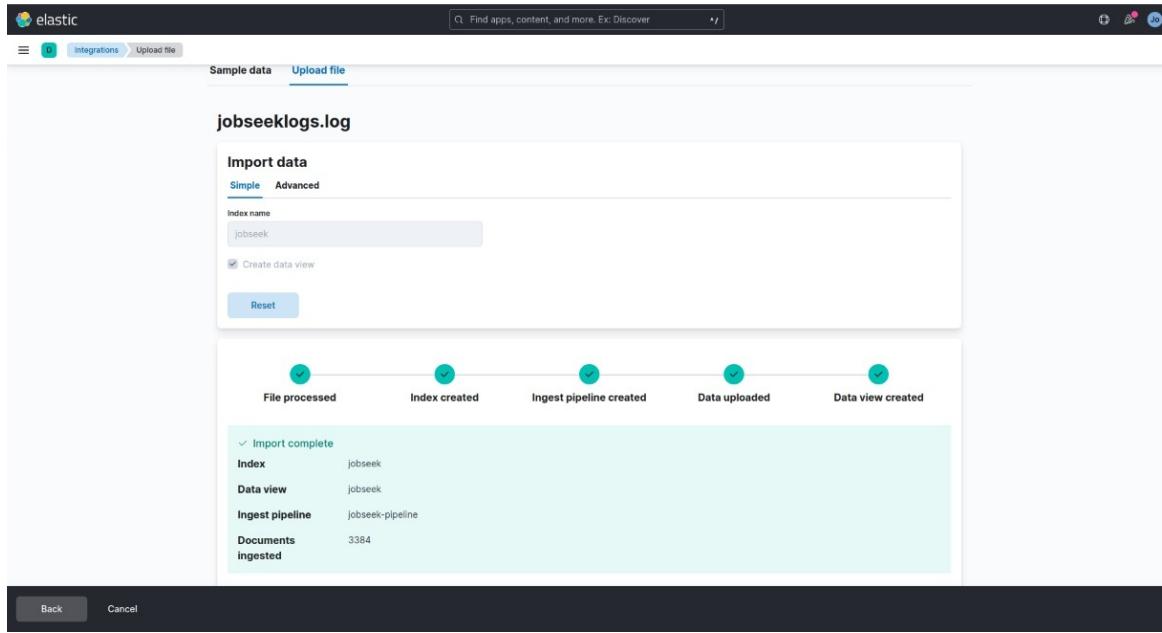
The following file formats are supported:

- Delimited text files, such as CSV and TSV
- Newline-delimited JSON
- Log files with a common format for the timestamp

You can upload files up to 100 MB.

Select or drag and drop a file

- Drop jobseeklogs.log



- Click on View Index to see the logs

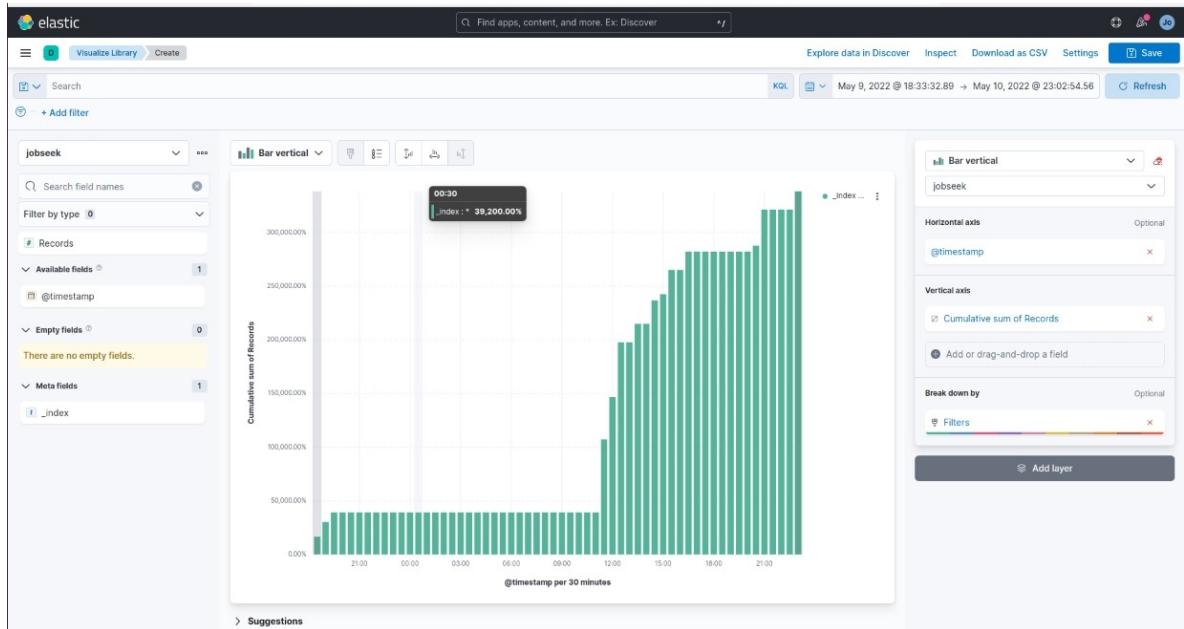


Figure 21: Cumulative sum of records vs timestamp

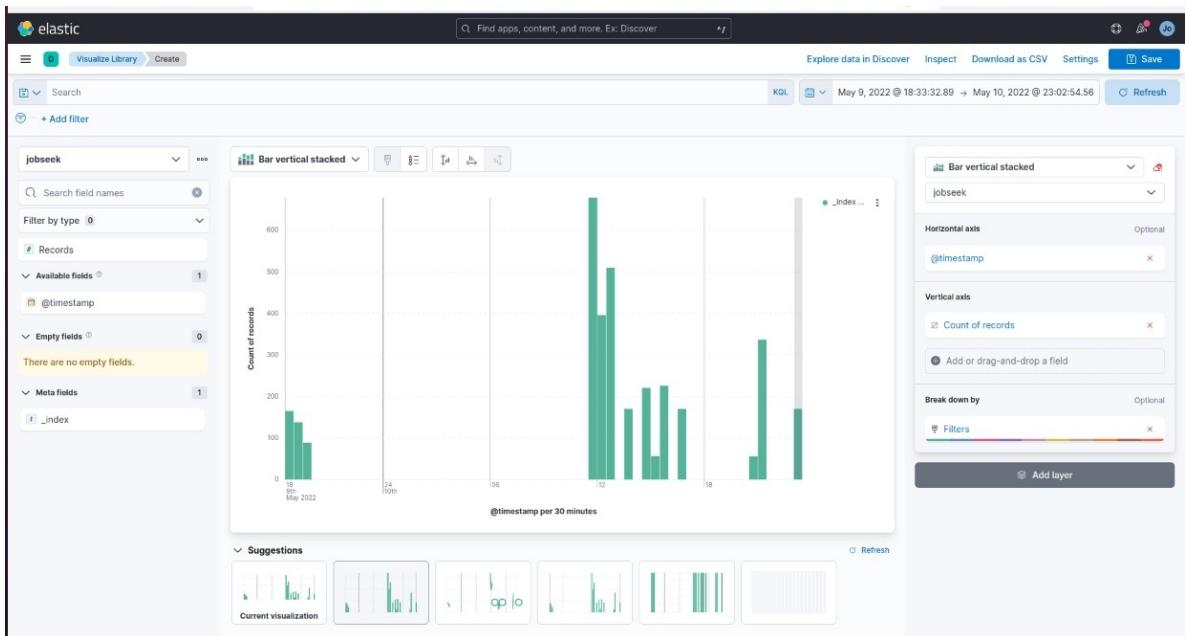
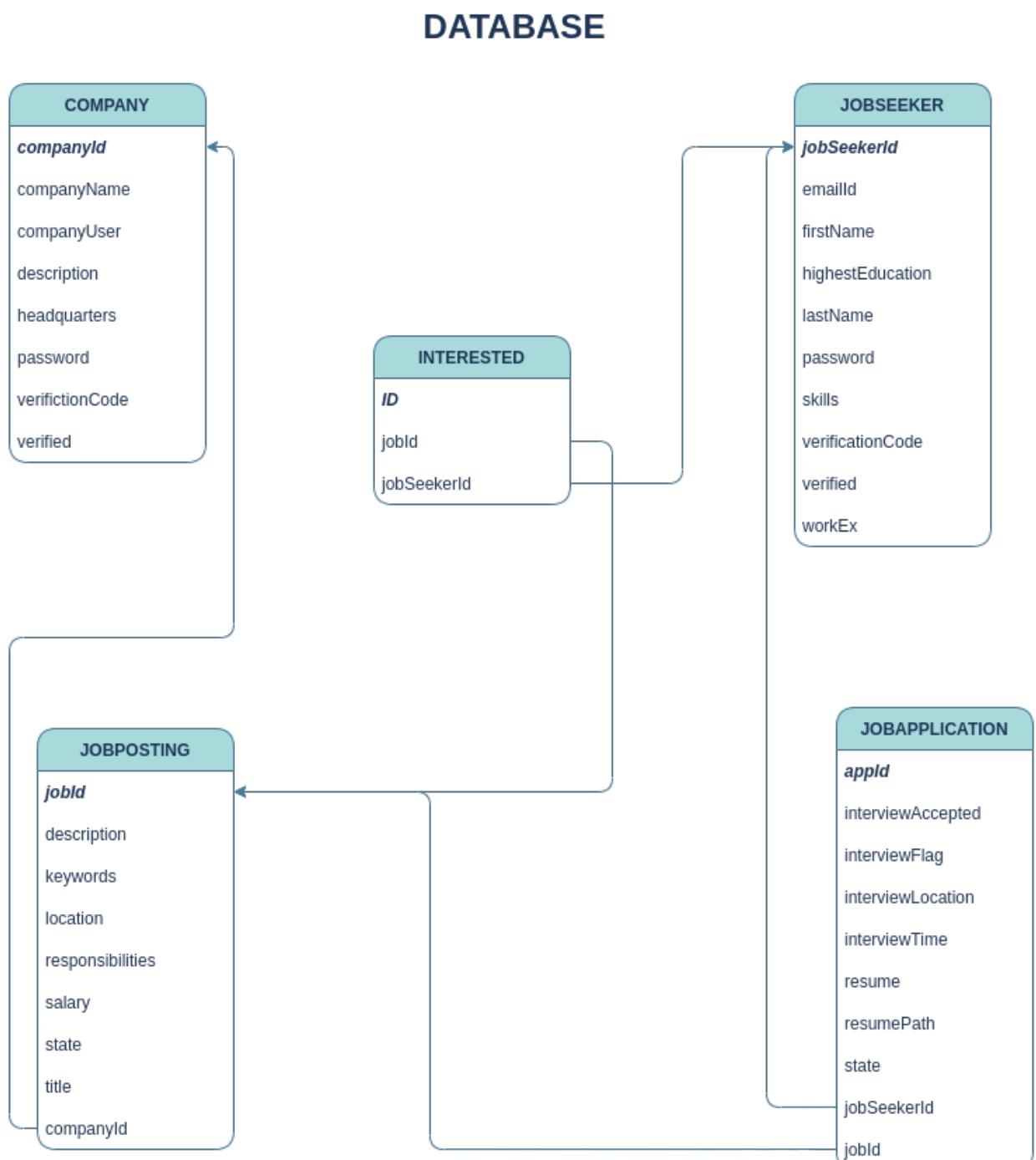


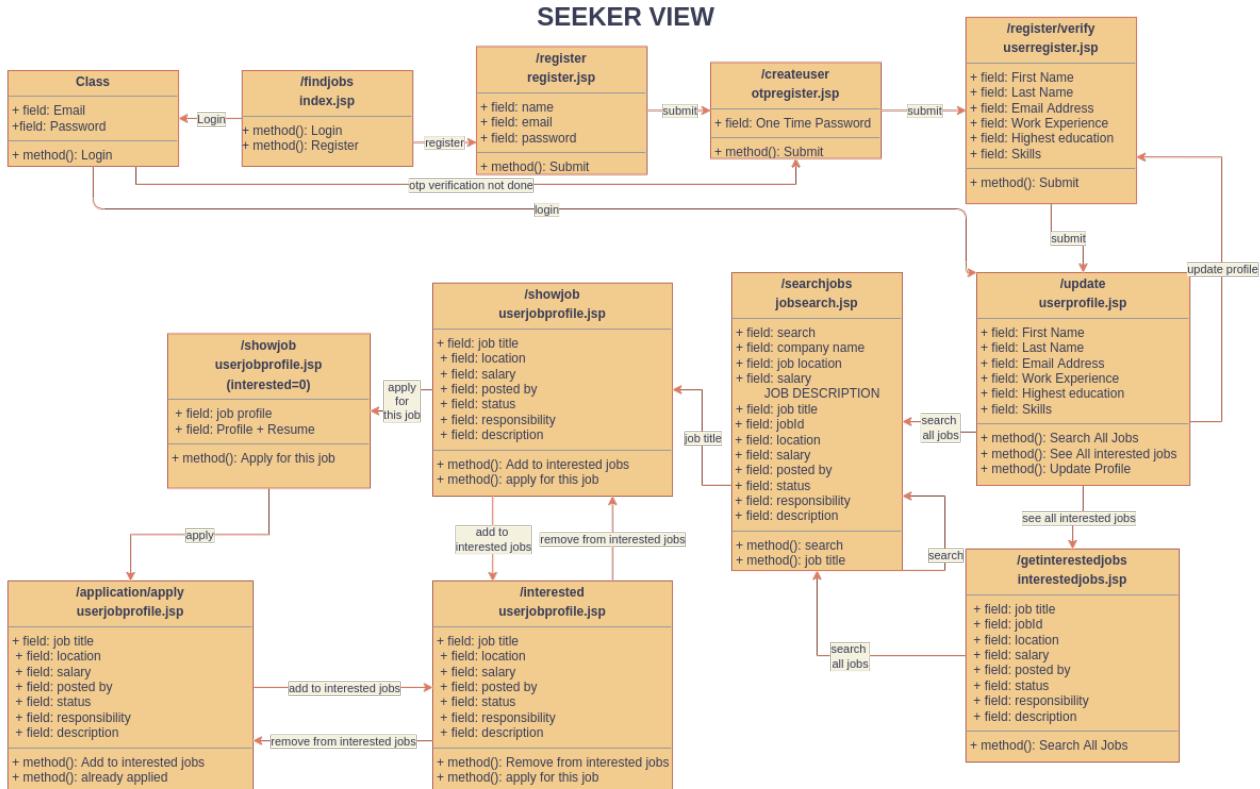
Figure 22:count of records vs timestamp

13. ARCHITECTURE DIAGRAMS

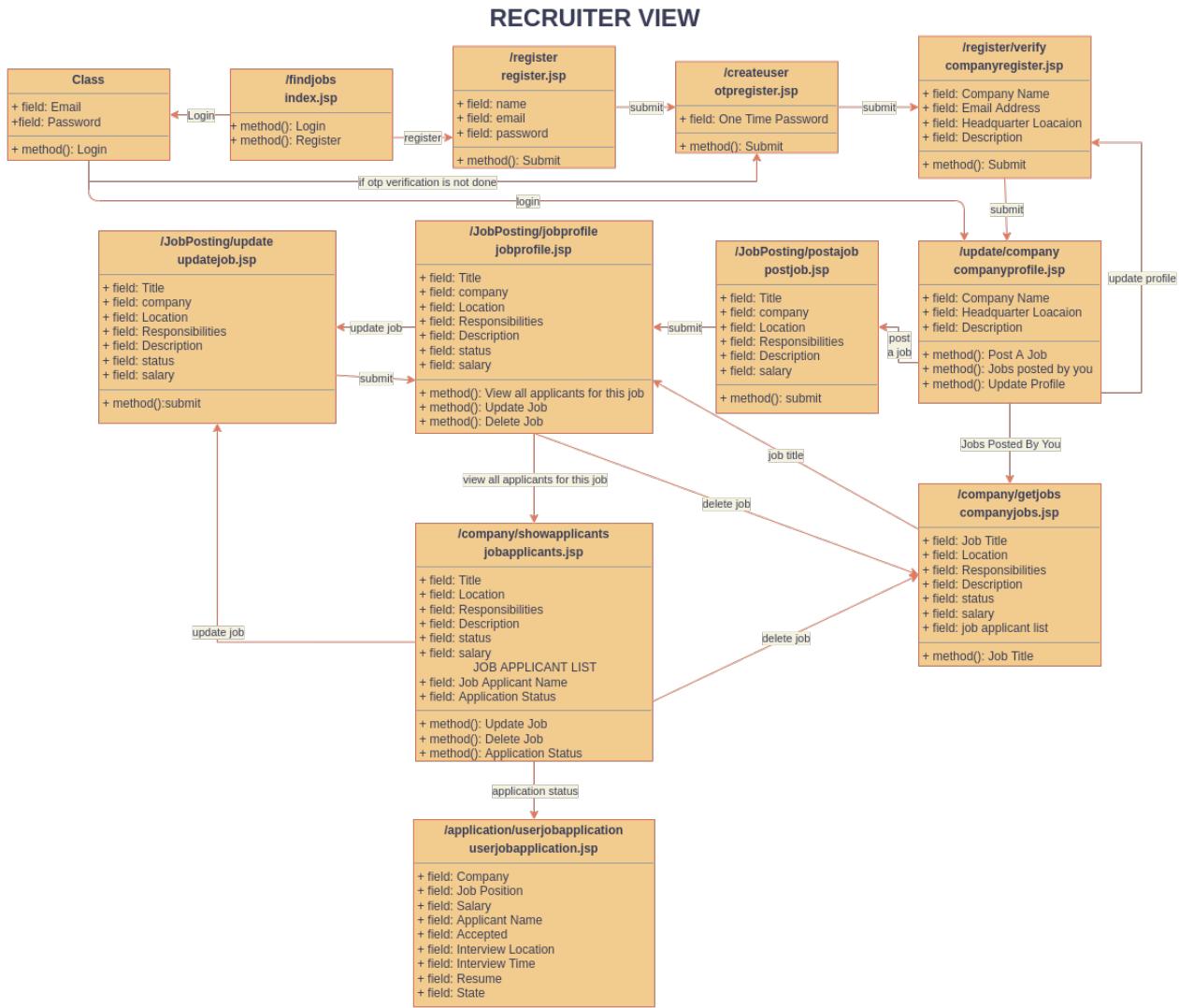
13.1. Database



13.2. Seeker Workflow



13.3. Recruiter Workflow



14. API docs

EndPoint	HTTP METHOD	INPUT	Functionality	Controller
/findjobs	GET	-	index page	Main
/register	GET	-	returns register.jsp webpage	Main
/login	POST	emailId, Password,Type	Depending on the type of user and his verification, redirects to the profile or verification page.	Main
/register/verify	POST	Type,Pin,UserId	Verifies user account and redirects to profile page	Main
/profile	POST	companyId	redirects to company profile page	Company
/showjob	POST	companyID,jobID	Returns Job Profile page	Company
/getjobs	POST	companyID	Return page with all jobs posted by this company	Company
/showapplicants	POST	jobID	returns page with all applicants for this job	Company
/postajob	POST	CompanyID	page for posting a newJobPosting job	JobPosting
/jobprofile	POST	job details like title,salary,etc	creates a new job posting or updates an existing one	JobPosting
/delete	POST	jobID	delete an existing job	JobPosting
/update	POST	jobID,companyID	returns webpage for updating a job posting	JobPosting
/update/{id}	POST	jobposting details	updates already existing job	JobPosting
/apply	POST	userId,JobID,application details	helps applying for a job	JobApplication
/cancel	POST	applicationID	cancel Job Application	JobApplication
/userjobapplication	POST	userID,jobApplicationID,jobID	return jobapplication page for a user	JobApplication
/searchjobs	POST	search params	returns jobs as per the search filters	JobSeeker
/showjob	POST	userID,jobID	shows job page to user	JobSeeker
/createuser	POST	user details	creates new jobseeker	JobSeeker
/userprofile	POST	userID	redirects to user profile	JobSeeker
/update	POST	user details	Updates user details	JobSeeker
/update/company	POST	company details	Updates company details	JobSeeker

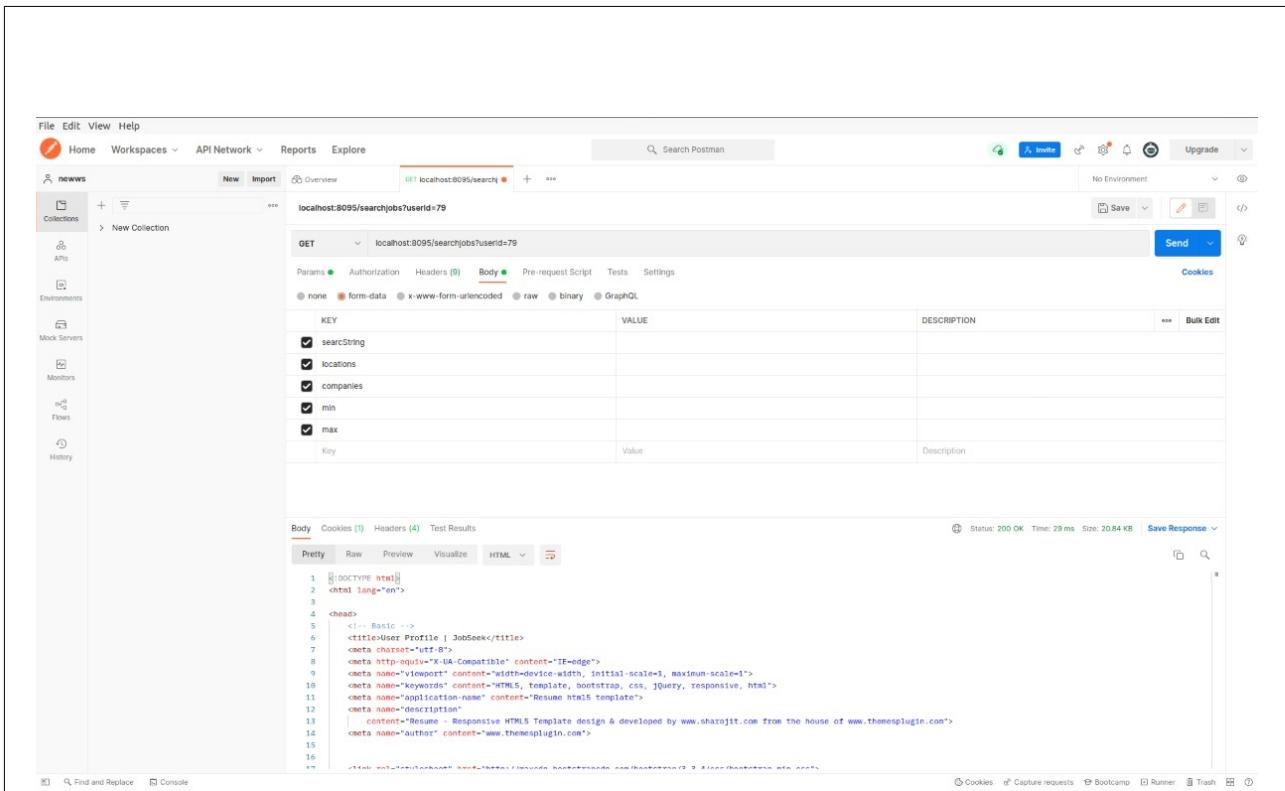


Figure 23: SearchJobs API via PostMan

15. Results and Discussion

15.1. Homepage

On starting the application, the user will be redirected to the homepage of the website from where the seeker and recruiter can able to login and register themselves.

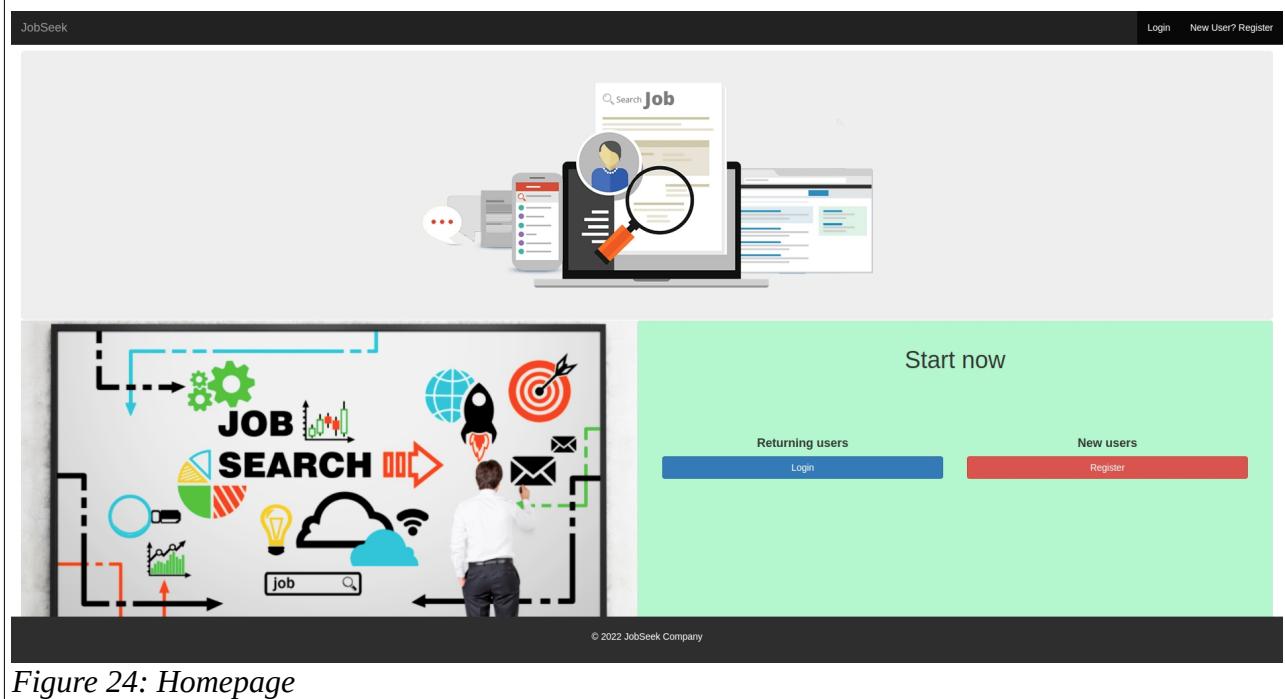
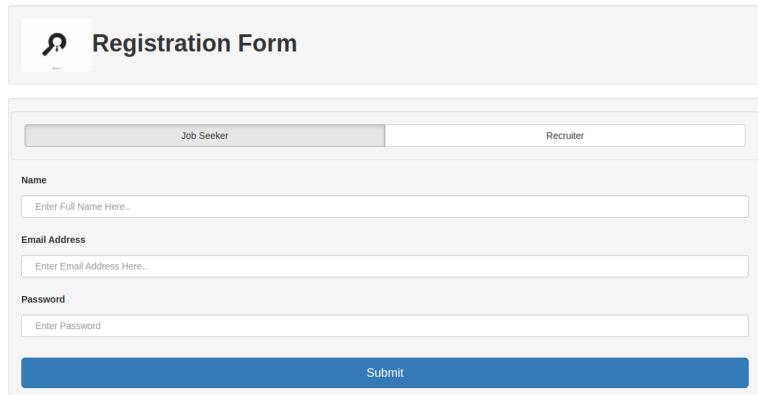


Figure 24: Homepage

15.2. Register Page

The new users has to register themselves as seeker or recruiter in order to access the website.



A screenshot of a registration form titled "Registration Form". The form is divided into two sections: "Job Seeker" and "Recruiter". The "Job Seeker" section contains fields for Name, Email Address, and Password, each with a placeholder text box. The "Recruiter" section is currently empty. At the bottom is a blue "Submit" button.

After clicking on the submit button an OTP will be sent on the user's mail id.

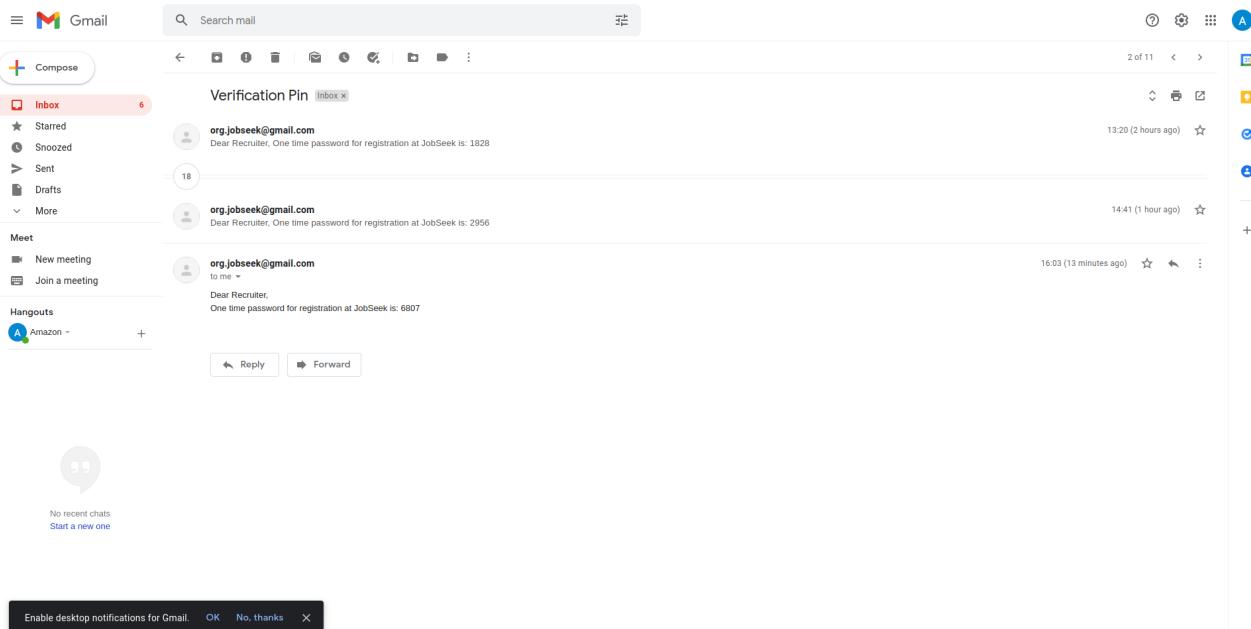


Figure 25: otp mail

The user will be redirected to Account Verification Page where the user has to write the OTP given in their mail.

The screenshot shows a web page titled "Account Verification". At the top, a green header bar displays the message "A verification OTP has been sent to your email address. Please verify!". Below this, the main content area has a light gray background. It features a logo icon followed by the text "Account Verification". A form section is present with the label "One Time Password" and a text input field containing placeholder text "Enter OTP sent to your mail". A blue "Submit" button is located at the bottom of the form. The footer of the page includes a dark bar with the copyright notice "© 2022 JobSeek Company".

Figure 26: account verification

After writing the correct OTP the user will be redirected to Complete Registration page where they have to fill the details for thier profile.

Recruiter's View

The screenshot shows a web page titled "Recruiter Profile". At the top, a dark header bar displays the application name "JobSeek" and a "Dashboard" link. To the right of the dashboard link, it says "logged in as Amazon". Below this, the main content area has a light gray background. It features a logo icon followed by the text "Recruiter Profile". A form section is present with three fields: "Company Name" (containing "Amazon"), "Email Address" (containing "amazon369123@gmail.com"), and "Headquarter location" (containing "Bangalore"). There is also a "Description" field with the text "E-commerce Website". A blue "Submit" button is located at the bottom of the form. The footer of the page includes a dark bar with the copyright notice "© 2022 JobSeek Company".

Figure 27: Recruiter Profile Page

Seeker's View

The screenshot shows the 'Seeker Profile' page. At the top, there is a header bar with 'JobSeek' on the left and 'Dashboard' and 'logged in as Mugdha' on the right. Below the header is a title 'Seeker Profile' with a user icon. The main content area contains a form with fields for 'First Name' (Mugdha), 'Last Name' (Agarwal), 'Email Address' (agarwalmugdha1998@gmail.com), 'Work experience (in years)' (0), 'Highest education' (radio buttons for Master's degree, Bachelor's degree, and Ph.D., with Master's degree selected), and 'Skills (comma separated)' (empty input field). A blue 'Submit' button is at the bottom.

© 2022 JobSeek Company

Figure 28: Seeker Profile Page

15.3. Login

The user can login the system via this page,

The screenshot shows the login page. At the top, there is a header bar with 'JobSeek' on the left and 'Login' and 'New User? Register' on the right. Below the header is a search bar with 'Search Job'. A central modal window titled 'User' contains fields for 'Job Seeker' and 'Recruiter', both labeled 'Email'. Below these are 'Enter Email' and 'Enter Password' fields. A green 'Login' button is at the bottom of the modal. In the background, there is a decorative graphic with the words 'JOB SEARCH' and various icons like gears, charts, and a rocket. To the right, there are buttons for 'Returning users' (blue) and 'New users' (red), with 'Start now' text above them. The footer of the page includes the copyright notice '© 2022 JobSeek Company'.

Figure 29: Login page

Case 1: If the user has completed the whole registration process

The user will be redirected to the respective dashboard page

Case 2: If user has not completed the OTP verification process while doing the registration

While logging in the system a new OTP will be sent to their mail and only after completing the OTP verification process and completing the whole registration process, the user will be able to enter the dashboard page.

The screenshot shows a web page titled "Account Verification". At the top, there is a dark header bar with the text "JobSeeker". Below it, a pinkish-red banner displays the message: "Your email hasn't been verified! Please Verify the email before trying to login. We have sent an email with new OTP.". The main content area has a light gray background. It features a logo of a keyhole and the text "Account Verification". Below this, a section labeled "One Time Password" contains a text input field with the placeholder "Enter OTP sent to your mail". At the bottom of the form is a blue "Submit" button. In the bottom right corner of the page, there is a small copyright notice: "© 2022 JobSeek Company".

Figure 30: Case 2: If user has not completed the OTP verification process while doing the registration

15.4. Recruiter's View

Dashboard

After completing the registration the user will be redirected to this page where the user can see all the details filled during the registration process that is,

- Company Name
- Location
- Company Description

From Dashboard the user can access the following functionality:

- Update Profile
- Post a Job
- Jobs Posted By You

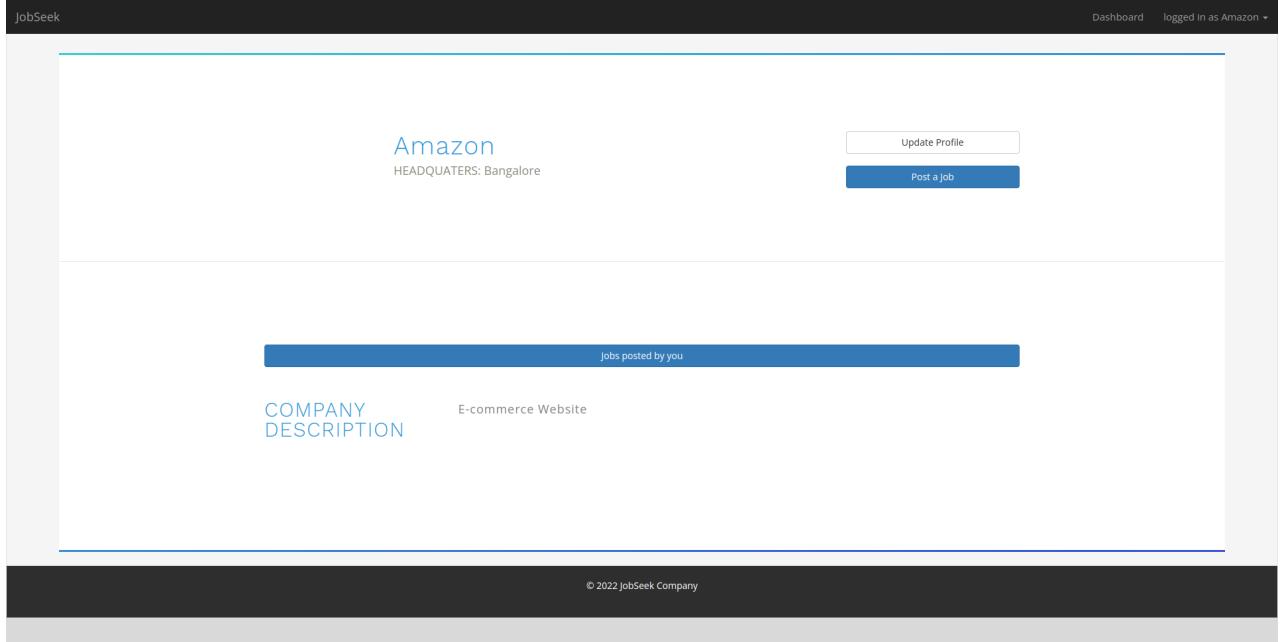


Figure 31: Recruiter's Dashboard

Update Profile

After clicking on Update profile button in the **dashboard**, the user will be redirected to Profile page where the user can update their profile as per their wish.

Post a Job

The user can post an opening for a new job by clicking on this button present in the **dashboard**. The user has to fill the details about this job opening and has to click on the submit button so that the seekers can view this new job opening and can apply for it.

The screenshot shows the JobSeek application's 'Post a Job' page. At the top, there is a navigation bar with 'JobSeek' on the left and 'Dashboard' and 'logged in as Amazon' on the right. Below the navigation is a decorative graphic of a blue sticky note pinned to a board with the words 'JOB POST'. The main form area has several input fields: 'Title' (placeholder 'Enter title Here...'), 'Location' (placeholder 'Enter Location Here...'), 'Company' (set to 'Amazon'), 'Responsibilities' (placeholder 'Enter Responsibilities Here...'), 'Description' (placeholder 'Enter Description Here...'), and 'Salary(\$ p.a.)' (placeholder 'Enter Salary in \$'). A large blue 'Submit' button is at the bottom of the form. The footer contains the copyright notice '© 2022 JobSeek Company'.

Figure 32: Post a Job

Job Profile

After posting a new job the user will be redirected to the Job Profile page where he can see the details of this job.

The user also can avail the following functionality:

- View job applicants for this job
- Update Job
- Delete Job

JobSeek

Dashboard logged in as Amazon ▾



Job Profile

[View all job applicants for this job](#)

SDE

Bangalore

Salary: 12

Posted by: Amazon

Job responsibilities: Backend Developer

Status: Open

[Update job](#)

JOB DESCRIPTION

Java, Spring Boot

[Delete job](#)

© 2022 JobSeek Company

Figure 33: Job Profile

Update Job

After clicking on this button from the **Job Profile Page** the user will be redirected to update job page where the user can update the job and after updating the user will again be redirected to Job Profile page.

JobSeek

Cancel | logged in as Amazon ▾

Update the job

Title SDE	Location Bangalore	Company Amazon
Responsibilities Backend Developer		
Description Java, Spring Boot		
Salary(\$ p.a.) 12		
Open	Filled	Cancelled
Submit		

© 2022 JobSeek Company

Figure 34: Update Job

Delete Job

After clicking on this button from the **Job profile Page** the job entry for this job will get deleted from the database and the user will be redirected to Company job posting page.

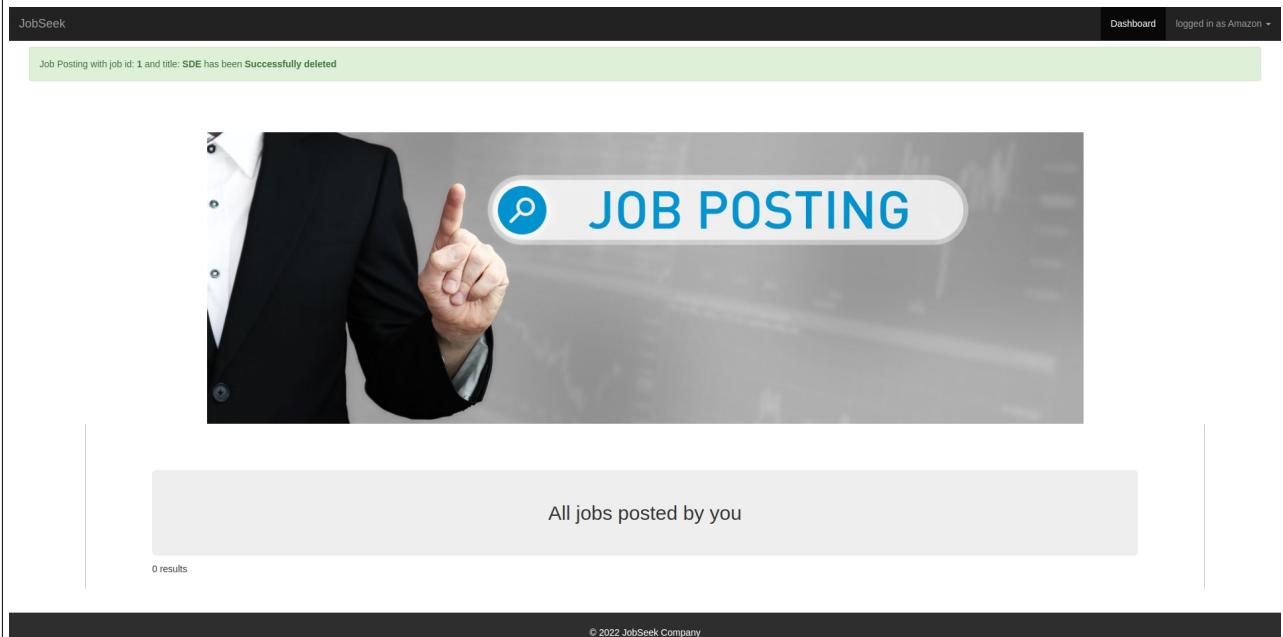


Figure 35: Company Job Posting Page

View Job Applicants for this Job

After clicking on this button from the **Job profile Page** the user will be redirected to job applicants page, where he can see the respective job details along with the applicants list.

The user can still update the job or even delete the job.

The Update job button will redirect the user to Update Page.

The Delete Job button will remove this job entry from the database and will redirect the user to the Company job posting page.

Case 1: When no Applicants has applied for this job

The screenshot shows a job listing for an SDE position. The job title is "SDE". The location is "Bangalore" and the salary is "12". The status is "Open" and the job responsibilities are "Backend Developer". The job description is "JOB DESCRIPTION: Java, Spring Boot". Below the job details are two buttons: "Update job" (white) and "Delete job" (red). At the bottom of the page, there is a "Job Applicants List" section with a message "Search Results: 0". The footer of the application includes the text "© 2022 JobSeek Company".

Figure 36: Case 1: Job Applicant Page

Case 2: When some applicants has applied for this job

If some applicants has applied for this job then it will get reflected to the job applicants list, which will show the applicant name and their application status.

JobSeek

Dashboard logged in as Amazon



Job Applicants

SDE

Location: Bangalore

Salary: 12

Status: Open

Job responsibilities: Backend Developer

JOB DESCRIPTION: Java, Spring Boot

Update job

Delete Job

Job Applicants List

Search Results: 1

Mugdha Agarwal

Application Status

© 2022 JobSeek Company

Figure 37: Case 2: Job Applicant Page

The user can click on the application status to see the status of that particular applicant for this job.

JobSeek

Dashboard logged in as Amazon

Job Application Status

Company: Amazon

Job Position: SDE

Salary: 12

Applicant Name: Mugdha Agarwal

Accepted: false

Interview Location:

Interview Time:

Resume: false

State: 0

Figure 38: Job Application Status

Jobs posted by you

After clicking on this button in **dashboard** the user will be redirected to Company Job Posting page where he can see all the jobs which has been posted by him so far.

Case 1: If no jobs has been posted yet

If user has not posted any jobs then the Company Job Posting page will show no results as depicted by the given picture.

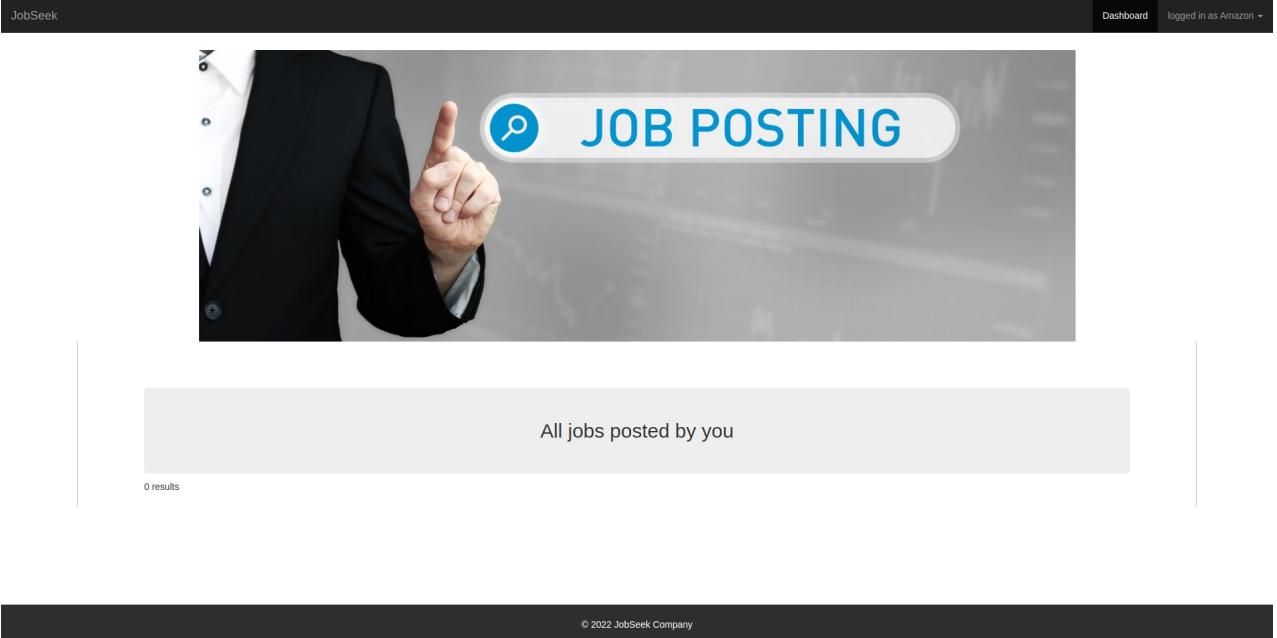


Figure 39: Case 1: Company Job Posting Page

Case 2: If user has posted some job opening

If user has posted some jobs then the Company Job Posting page will reflect all the jobs that has been posted by the user.

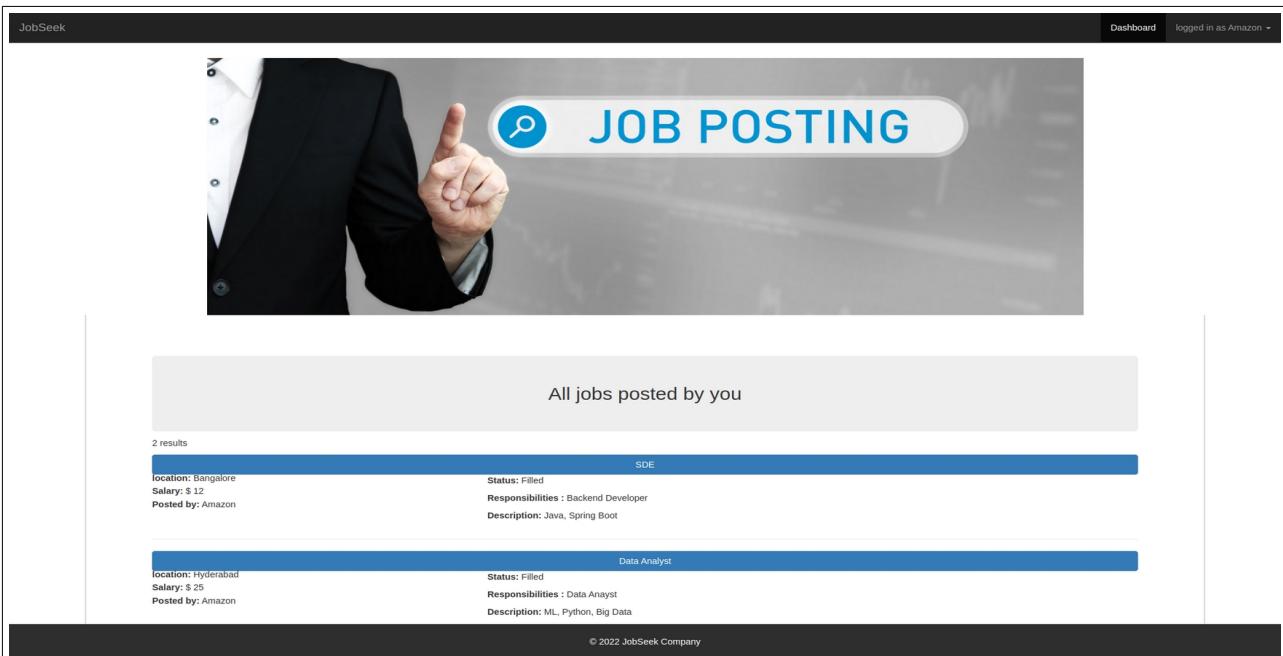


Figure 40: Case 2: Company Job Posting Page

The user can click on the job title to view the job profile of this Job opening from where he can see all the applicants who have applied for this job or can update or delete the job.

15.5. Seeker's View

Dashboard

After completing the registration the user will be redirected to this page where the user can see all the details filled during the registration process that is,

- First Name
- Last Name
- Email Address
- Work Experience (in years)
- Highest Education
- Skills

Also the user can use the following functionality:

- Search all jobs
- See all interested jobs
- Update Profile

Update Profile

After clicking on update profile button from the **dashboard** the user will be redirected to the profile page where he can update his profile.

Search All Jobs

After clicking on this button from the **dashboard** the user will be redirected to search job page where he can search the jobs based on the following filters:

- Search
- Company Name
- Job Locations
- Salary

Also the search results of all the job openings will be shown in the page by default so the user can even click on the job title to go on the job profile page.

The screenshot shows the JobSeek search interface. At the top, there is a navigation bar with 'JobSeek' on the left, 'Dashboard' in the center, and 'logged in as Mugdha' on the right. Below the navigation bar is a large search bar with the placeholder 'Enter Search Keywords'. To the right of the search bar is a magnifying glass icon containing the word 'Jobs'. Below the search bar are three filter sections: 'Company name' (with a dropdown menu 'comma separated'), 'Job locations' (with a dropdown menu 'comma separated'), and 'Salary \$ p.a.' (with input fields for 'Min' (0) and 'Max' (100000)). A blue 'Search' button is located below these filters. The main content area is titled 'Search Results:' and displays two job listings. The first listing is for a 'SDE' position with job ID 2, located in Bangalore, salary \$12, posted by Amazon. The second listing is for a 'Data Analyst' position with job ID 23, located in Hyderabad, salary \$25, posted by Amazon. Both listings show 'Status: Open', 'Responsibilities : Backend Developer' (for SDE) or 'Data Analyst' (for Data Analyst), and 'Description: Java, Spring Boot' (for SDE) or 'ML, Python, Big Data' (for Data Analyst). At the bottom of the page is a dark footer bar with the text '© 2022 JobSeek Company'.

Figure 41: Search Jobs

Job Profile

By clicking on the job title from the **search job page** the user will be redirected to job profile page where he will be able to see the job details along with the following functionality:

- Add to interested jobs

This job has been Successfully added to your interests

SDE

Bangalore

Posted by: Amazon

Status: Open

Salary: \$ 12

Job responsibilities:
Backend Developer

JOB DESCRIPTION Java, Spring Boot

© 2022 jobGeek Company

➤ Remove from interested jobs

This job has been Successfully removed from your interests

SDE

Bangalore

Salary: \$ 12

Job responsibilities:

➤ Apply for this job

User can apply via their profile or profile + resume where they can upload the resume

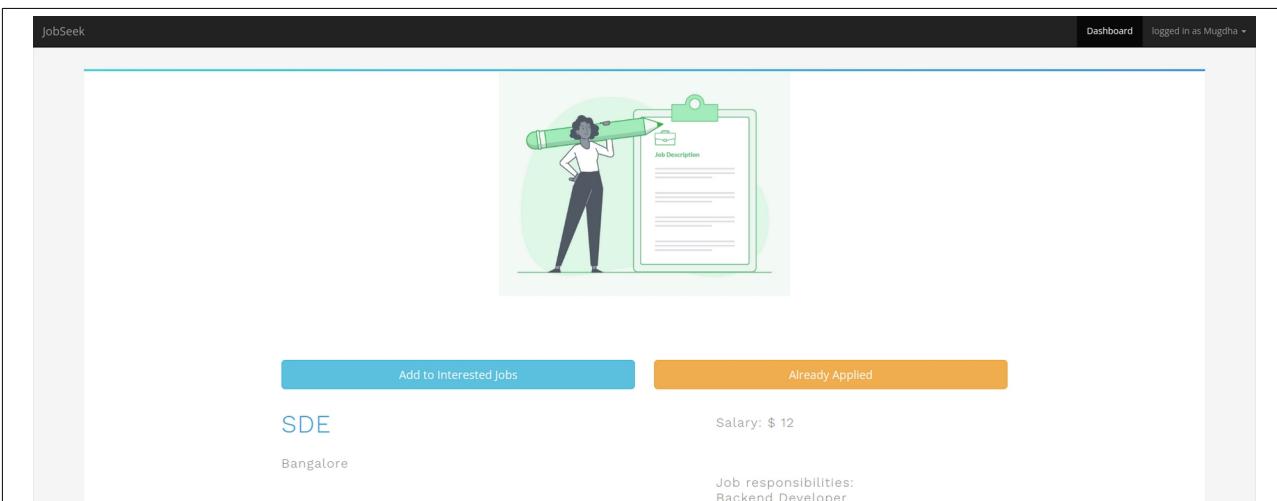
Add to Interested Jobs

Apply For This Job

SDE

Salary: \$ 12

After applying for the job the Job profile page will look like this



See all interested jobs

After clicking on this button the user will be redirected to Interested Jobs page where he can see all the jobs he has marked as interested ones'. Also the user can click on search all jobs, which will redirect the user to search job page.

Case 1: No job is marked as interested one

If user hasn't marked any jobs as the interested one then the Interested job page will look like this:

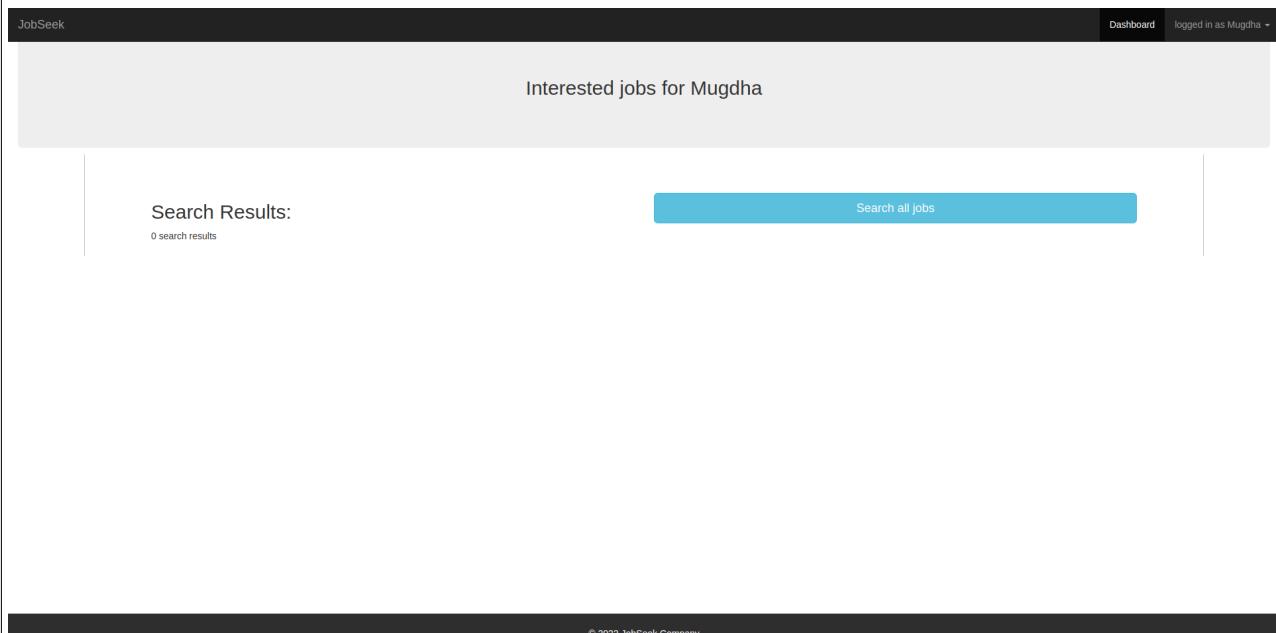
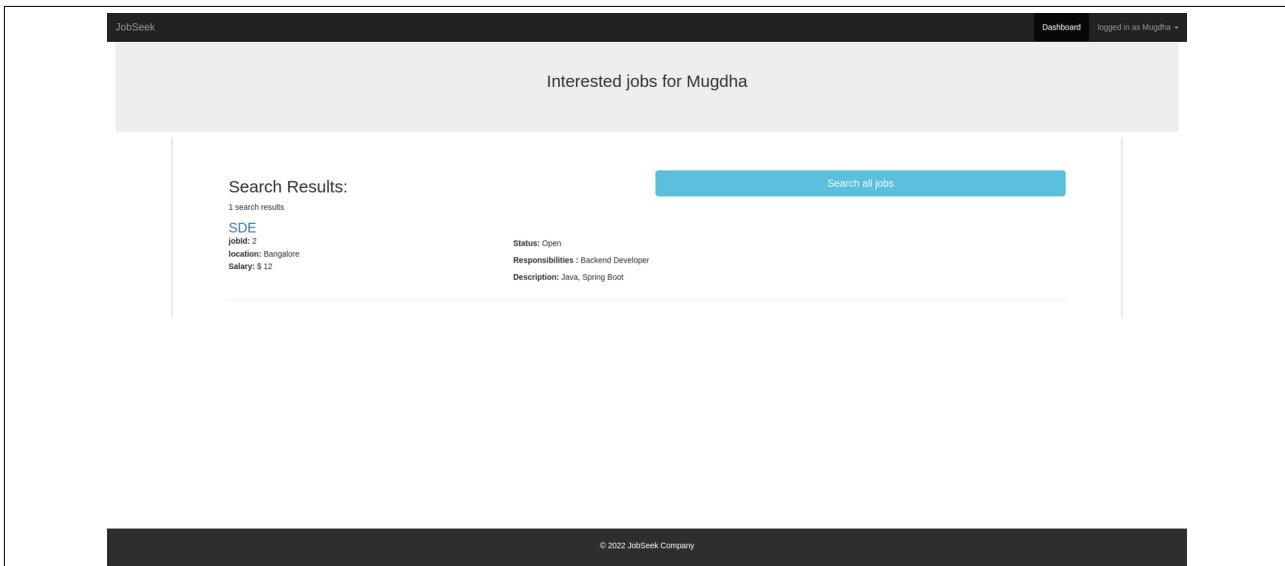


Figure 42: Case 1: No interested job present

Case 2: User has marked jobs in interested ones'

All the interested jobs marked by the user will reflect here and by clicking on the job title the user will get redirected to job profile page. If user want to search for more jobs then he can click on search all jobs and it will redirect him to the search jobs page.



16. Conclusion

We hope that by using this application the struggle of finding jobs and getting appropriate employee will be hassle free and both the seeker as well as the recruiter will be able to get want they want in an easy manner. There exists scope of improvement of adding more features which will enhance our application.

Since the entire project is built using DevOps methodology and DevOps lifecycle which consists of Continuous Integration, Continuous Deployment, Testing and Continuous Monitoring. So adding new features and fixing bugs is now easy and fast.

17. Scope for Future Work

- Right now the seeker does not know whether it's a full time job or part time job so in the future development the recruiter will be able to put these options while posting a job so that only the desired applicants can apply for that job.
 - The seeker as well as the recruiter does not know the rating in order to give preference while choosing an applicant or choosing a job so in future we can give the facility of showing the rating of each user.
 - The seeker as well as the recruiter right now can not give the feedback for the job so we can add this feature also.
-

18. References

- <https://www.baeldung.com/spring-boot-start>
- <https://spring.io/guides/gs/spring-boot-docker/>
- <https://www.bezkoder.com/docker-compose-spring-boot-mysql/>
- <https://www.baeldung.com/spring-boot-testing>
- https://www.tutorialspoint.com/spring_boot/spring_boot_sending_email.htm#:~:text=By%20using%20Spring%20Boot%20RESTful,in%20your%20build%20configuration%20file
- <https://www.baeldung.com/hibernate-one-to-many>
- <https://www.baeldung.com/jpa-get-auto-generated-id>
- <https://www.baeldung.com/spring-data-entitymanager>
- <https://www.baeldung.com/spring-requestmapping>
- <https://www.baeldung.com/transaction-configuration-with-jpa-and-spring>
- <https://www.baeldung.com/spring-boot-logging>
- <https://stackoverflow.com/questions/50671681/cannot-load-driver-class-com-mysql-jdbc-driver-spring-boot>
- <https://stackoverflow.com/questions/11746499/how-to-solve-the-failed-to-lazily-initialize-a-collection-of-role-hibernate-ex>
- <https://www.baeldung.com/spring-boot-failed-to-configure-data-source>
- <https://stackoverflow.com/questions/43574426/how-to-resolve-java-lang-noclassdeffounderror-javax-xml-bind-jaxbexception>
- <https://stackoverflow.com/questions/45414391/i-cant-connect-to-the-mysql-database-with-spring-boot-jpa>
- <https://www.baeldung.com/spring-request-method-not-supported-405>
- <https://stackoverflow.com/questions/34189756/warning-about-ssl-connection-when-connecting-to-mysql-database>
- <https://www.baeldung.com/spring-data-jpa-generate-db-schema>
- <https://stackoverflow.com/questions/26881739/unable-to-get-spring-boot-to-automatically-create-database-schema>
- <https://www.baeldung.com/junit-5-runwith>