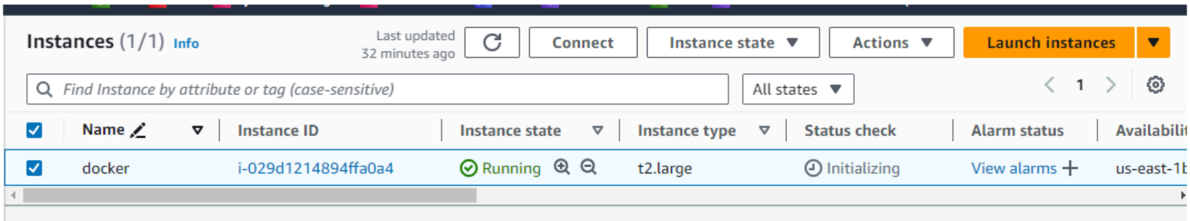
ANGULER JAVA PROJECT

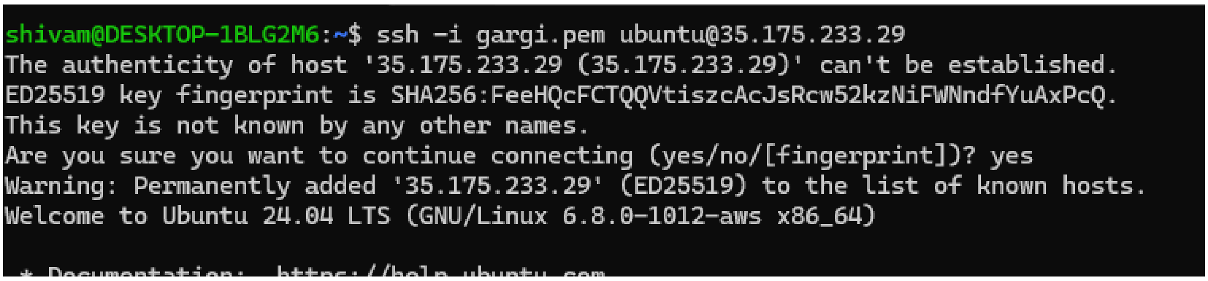
creating the instance



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabili
☑	docker	i-029d1214894ffa0a4	Running	t2.large	Initializing	View alarms +	us-east-1b

Login with ssh using public ip of created instance using command

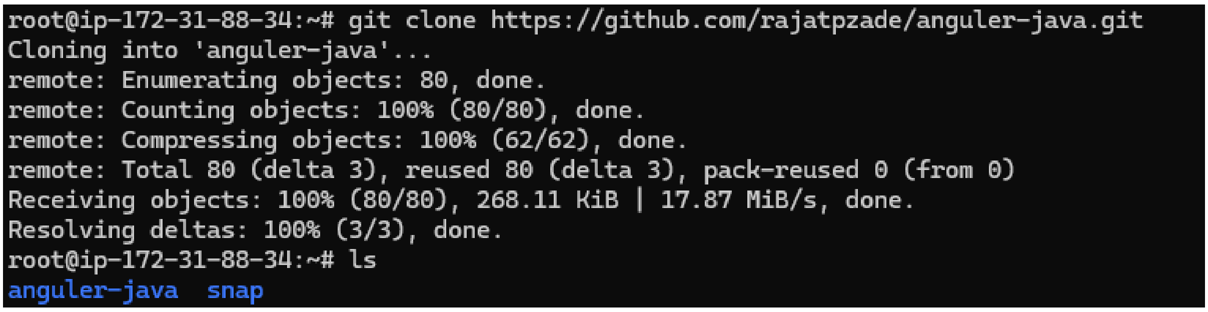
Ssh -I key username@public ip



```
shivam@DESKTOP-1BLG2M6:~$ ssh -i gargi.pem ubuntu@35.175.233.29
The authenticity of host '35.175.233.29 (35.175.233.29)' can't be established.
ED25519 key fingerprint is SHA256:FeeHQcFCTQQVtiszcAcJsRcw52kzNiFWNndfYuAxPcQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.175.233.29' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
```

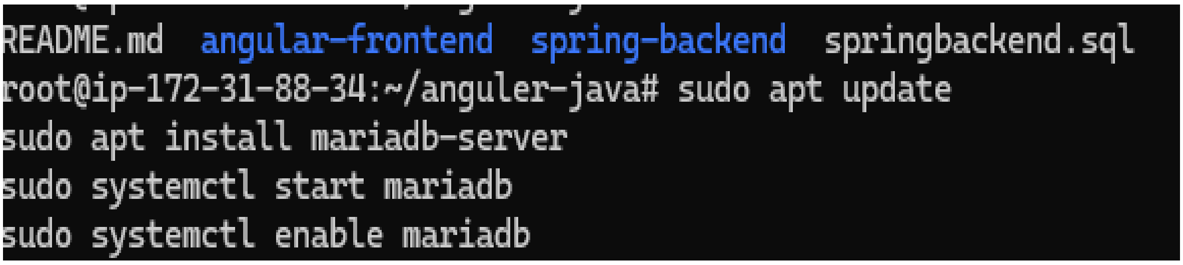
Cloning the repository from the provided documentation using command

Git clone https://github.com/rajatpzade/anguler-java.git



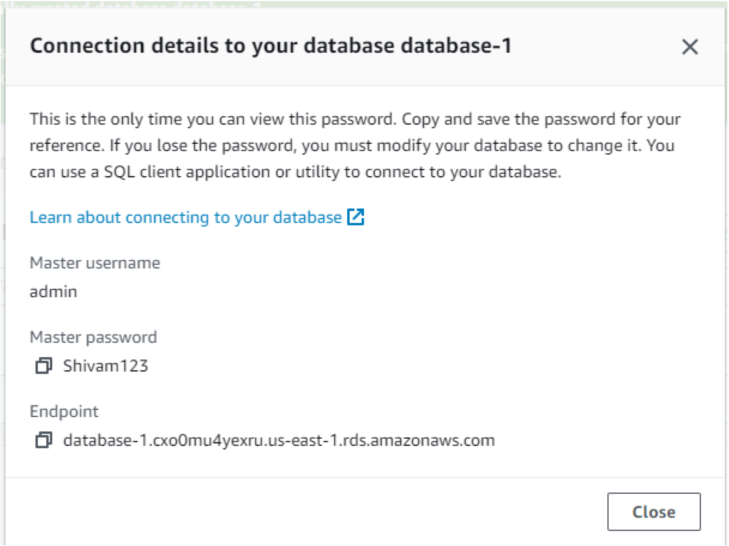
```
root@ip-172-31-88-34:~# git clone https://github.com/rajatpzade/anguler-java.git
Cloning into 'anguler-java'...
remote: Enumerating objects: 80, done.
remote: Counting objects: 100% (80/80), done.
remote: Compressing objects: 100% (62/62), done.
remote: Total 80 (delta 3), reused 80 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (80/80), 268.11 KiB | 17.87 MiB/s, done.
Resolving deltas: 100% (3/3), done.
root@ip-172-31-88-34:~# ls
anguler-java  snap
```

Installing mariadb-server



```
README.md  angular-frontend  spring-backend  springbackend.sql
root@ip-172-31-88-34:~/anguler-java# sudo apt update
sudo apt install mariadb-server
sudo systemctl start mariadb
sudo systemctl enable mariadb
```

Using RDS server we made a database which provided us a end point username and password



Connection details to your database database-1

This is the only time you can view this password. Copy and save the password for your reference. If you lose the password, you must modify your database to change it. You can use a SQL client application or utility to connect to your database.

Learn about connecting to your database

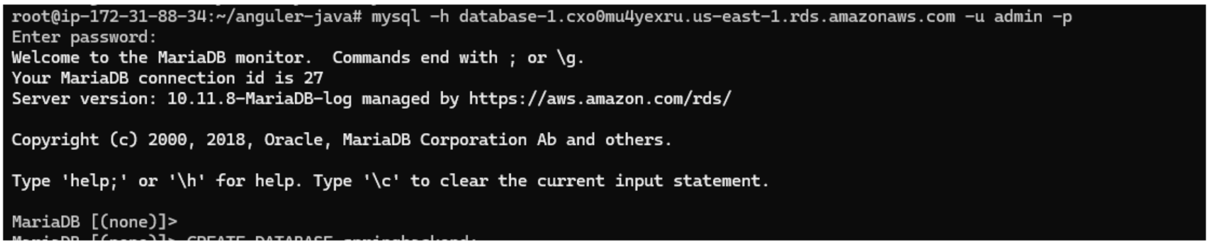
Master username
admin

Master password
Shivam123

Endpoint
database-1.cxo0mu4yexru.us-east-1.rds.amazonaws.com

Login to mariadb database using endpoint and credential using command

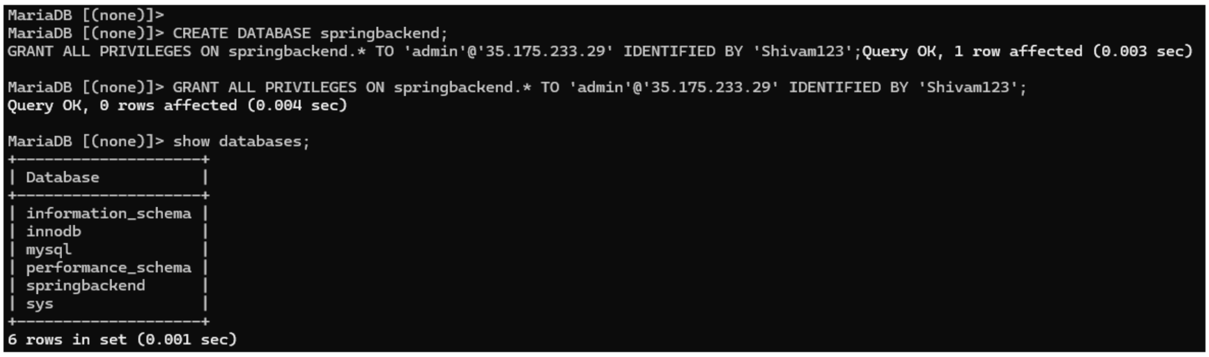
Mysql -h database endpoint -u username -p password



```
root@ip-172-31-88-34:~/anguler-java# mysql -h database-1.cxo0mu4yexru.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 27
Server version: 10.11.8-MariaDB-log managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
MariaDB [(none)]> CREATE DATABASE springbackend;
```

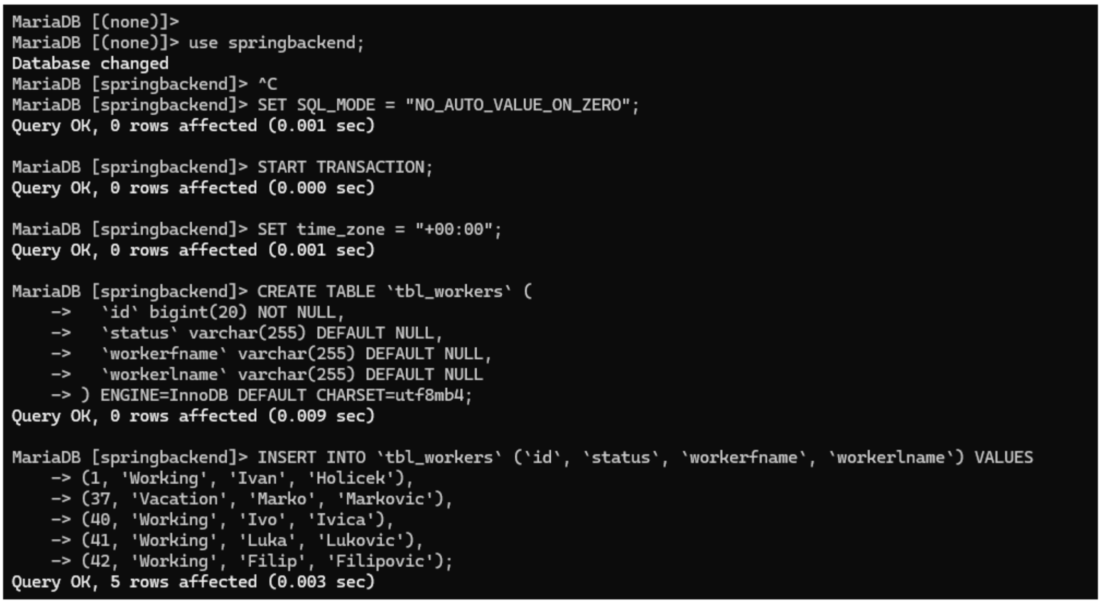
Creating a database called springbackend;

Granting all privileges editing username localhost and pass by giving the public ip at localhost admin at username and password which u give to rds database



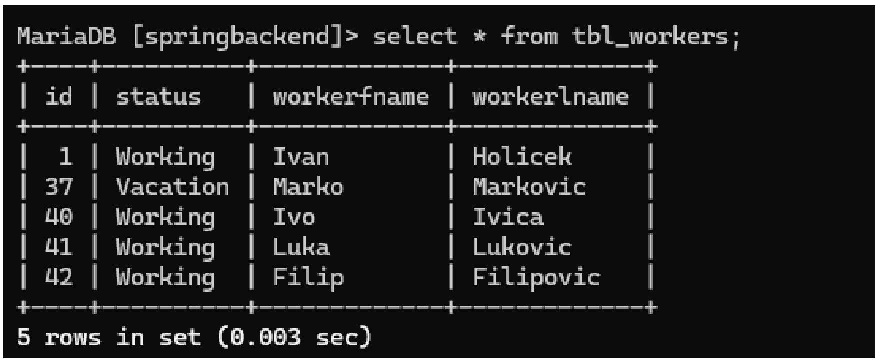
```
MariaDB [(none)]>
MariaDB [(none)]> CREATE DATABASE springbackend;
GRANT ALL PRIVILEGES ON springbackend.* TO 'admin'@'35.175.233.29' IDENTIFIED BY 'Shivam123';Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON springbackend.* TO 'admin'@'35.175.233.29' IDENTIFIED BY 'Shivam123';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| springbackend      |
| sys                |
+--------------------+
6 rows in set (0.001 sec)
```

After creating database we use the database and add the all data given in the documentation



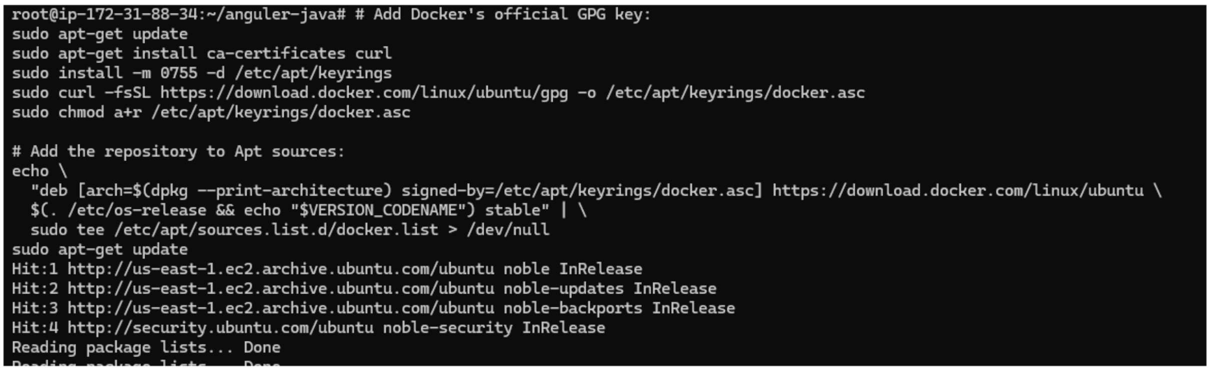
```
MariaDB [(none)]>
MariaDB [(none)]> use springbackend;
Database changed
MariaDB [springbackend]> ^C
MariaDB [springbackend]> SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
Query OK, 0 rows affected (0.001 sec)

MariaDB [springbackend]> START TRANSACTION;
Query OK, 0 rows affected (0.000 sec)

MariaDB [springbackend]> SET time_zone = "+00:00";
Query OK, 0 rows affected (0.001 sec)

MariaDB [springbackend]> CREATE TABLE `tbl_workers` (
    ->   `id` bigint(20) NOT NULL,
    ->   `status` varchar(255) DEFAULT NULL,
    ->   `workerfname` varchar(255) DEFAULT NULL,
    ->   `workerlname` varchar(255) DEFAULT NULL
    -> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.009 sec)

MariaDB [springbackend]> INSERT INTO `tbl_workers` (`id`, `status`, `workerfname`, `workerlname`) VALUES
    -> (1, 'Working', 'Ivan', 'Holicek'),
    -> (37, 'Vacation', 'Marko', 'Markovic'),
    -> (40, 'Working', 'Ivo', 'Ivica'),
    -> (41, 'Working', 'Luka', 'Lukovic'),
    -> (42, 'Working', 'Filip', 'Filipovic');
Query OK, 5 rows affected (0.003 sec)
```

Checking the data using the given command



```
MariaDB [springbackend]> select * from tbl_workers;
+----+----------+-------------+-------------+
| id | status   | workerfname | workerlname |
+----+----------+-------------+-------------+
|  1 | Working  | Ivan        | Holicek     |
| 37 | Vacation | Marko       | Markovic    |
| 40 | Working  | Ivo         | Ivica       |
| 41 | Working  | Luka        | Lukovic     |
| 42 | Working  | Filip       | Filipovic   |
+----+----------+-------------+-------------+
5 rows in set (0.003 sec)
```

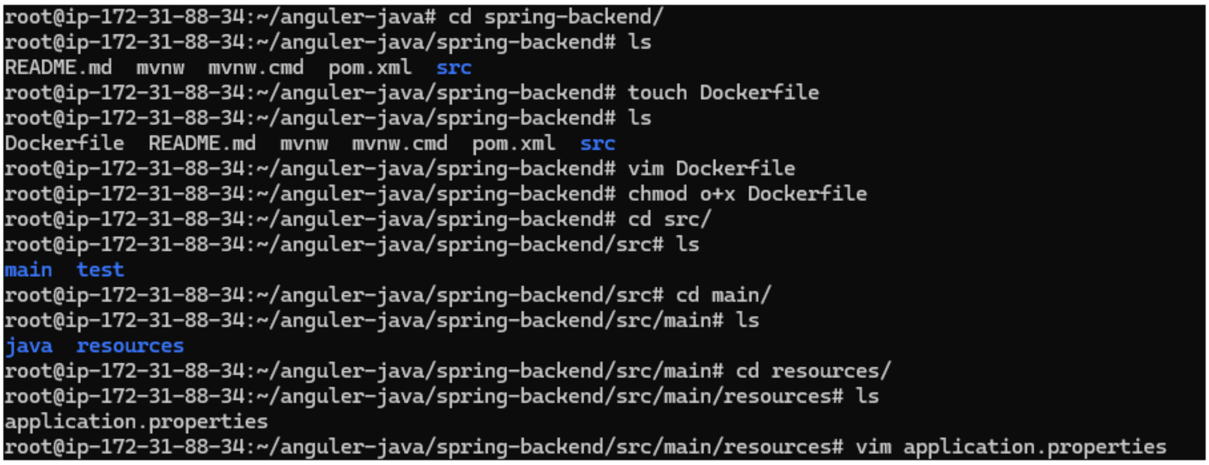
Install docker



```
root@ip-172-31-88-34:~/anguler-java# # Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Reading package lists... Done
```

Firstly make change in the spring backend;

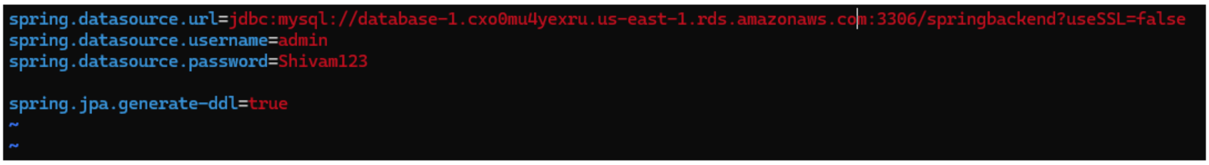
Create a dockerfile and copying script in the file giving execution permission



```
root@ip-172-31-88-34:~/anguler-java# cd spring-backend/
root@ip-172-31-88-34:~/anguler-java/spring-backend# ls
README.md  mvnw  mvnw.cmd  pom.xml  src
root@ip-172-31-88-34:~/anguler-java/spring-backend# touch Dockerfile
root@ip-172-31-88-34:~/anguler-java/spring-backend# ls
Dockerfile  README.md  mvnw  mvnw.cmd  pom.xml  src
root@ip-172-31-88-34:~/anguler-java/spring-backend# vim Dockerfile
root@ip-172-31-88-34:~/anguler-java/spring-backend# chmod o+x Dockerfile
root@ip-172-31-88-34:~/anguler-java/spring-backend# cd src/
root@ip-172-31-88-34:~/anguler-java/spring-backend/src# ls
main  test
root@ip-172-31-88-34:~/anguler-java/spring-backend/src# cd main/
root@ip-172-31-88-34:~/anguler-java/spring-backend/src/main# ls
java  resources
root@ip-172-31-88-34:~/anguler-java/spring-backend/src/main# cd resources/
root@ip-172-31-88-34:~/anguler-java/spring-backend/src/main/resources# ls
application.properties
root@ip-172-31-88-34:~/anguler-java/spring-backend/src/main/resources# vim application.properties
```

Giving RDS endpoint and credentials in the application.properties;

Path cd anguler-java/spring-backend/src/main/resources /application.properties

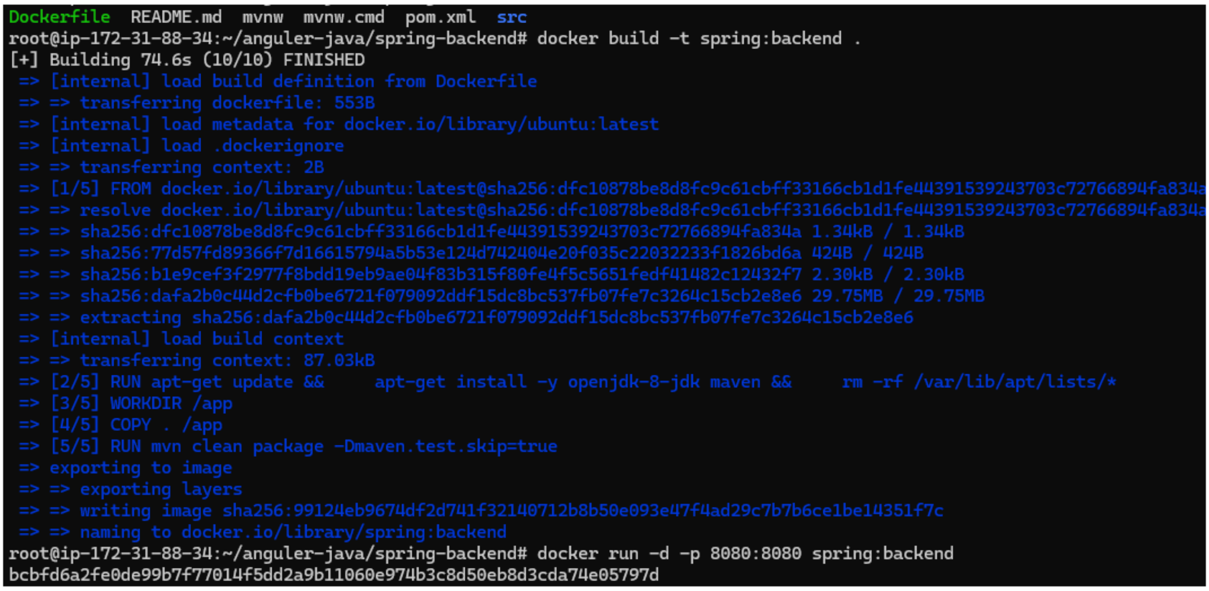


```
spring.datasource.url=jdbc:mysql://database-1.cxo0mu4yexru.us-east-1.rds.amazonaws.com:3306/springbackend?useSSL=false
spring.datasource.username=admin
spring.datasource.password=Shivam123

spring.jpa.generate-ddl=true
~
~
```

Coming back to spring backend build and run the spring:backend by 8080 port using the commands

Docker build -t spring:backend .

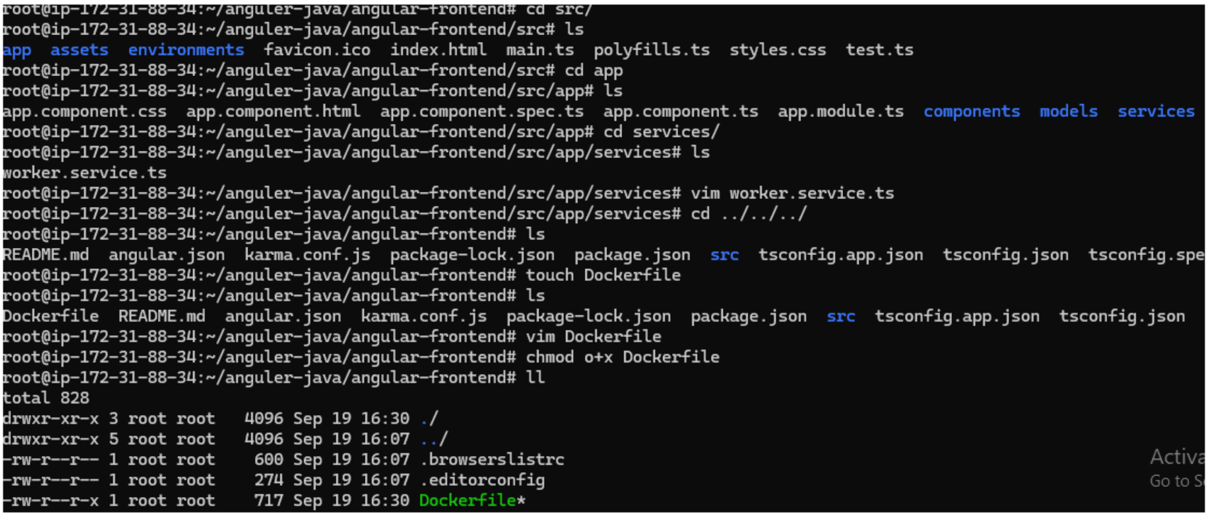
Docker run -d -p 8080:8080 spring:backend



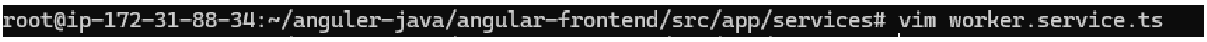
```
Dockerfile  README.md  mvnw  mvnw.cmd  pom.xml  src
root@ip-172-31-88-34:~/anguler-java/spring-backend# docker build -t spring:backend .
[+] Building 74.6s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 553B
 => [internal] load metadata for docker.io/library/ubuntu:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/ubuntu:latest@sha256:dfc10878be8d8fc9c61cbff33166cb1d1fe44391539243703c72766894fa834a
 => => resolve docker.io/library/ubuntu:latest@sha256:dfc10878be8d8fc9c61cbff33166cb1d1fe44391539243703c72766894fa834a
 => => sha256:dfc10878be8d8fc9c61cbff33166cb1d1fe44391539243703c72766894fa834a 1.34kB / 1.34kB
 => => sha256:77d57fd89366f7d1661579 4a5b53e124d742404e20f035c22032233f1826bd6a 424B / 424B
 => => sha256:b1e9cef3f2977f8bdd19eb9ae04f83b315f80fe4f5c5651fedf41482c12432f7 2.30kB / 2.30kB
 => => sha256:dafa2b0c44d2cfb0be6721f079092ddf15dc8bc537fb07fe7c3264c15cb2e8e6 29.75MB / 29.75MB
 => => extracting sha256:dafa2b0c44d2cfb0be6721f079092ddf15dc8bc537fb07fe7c3264c15cb2e8e6
 => [internal] load build context
 => => transferring context: 87.03kB
 => [2/5] RUN apt-get update &&     apt-get install -y openjdk-8-jdk maven &&     rm -rf /var/lib/apt/lists/*
 => [3/5] WORKDIR /app
 => [4/5] COPY . /app
 => [5/5] RUN mvn clean package -Dmaven.test.skip=true
 => exporting to image
 => => exporting layers
 => => writing image sha256:99124eb9674df2d741f32140712b8b50e093e47f4ad29c7b7b6ce1be14351f7c
 => => naming to docker.io/library/spring:backend
root@ip-172-31-88-34:~/anguler-java/spring-backend# docker run -d -p 8080:8080 spring:backend
bcbfd6a2fe0de99b7f77014f5dd2a9b11060e974b3c8d50eb8d3cda74e05797d
```

Same process for anguler-frontend

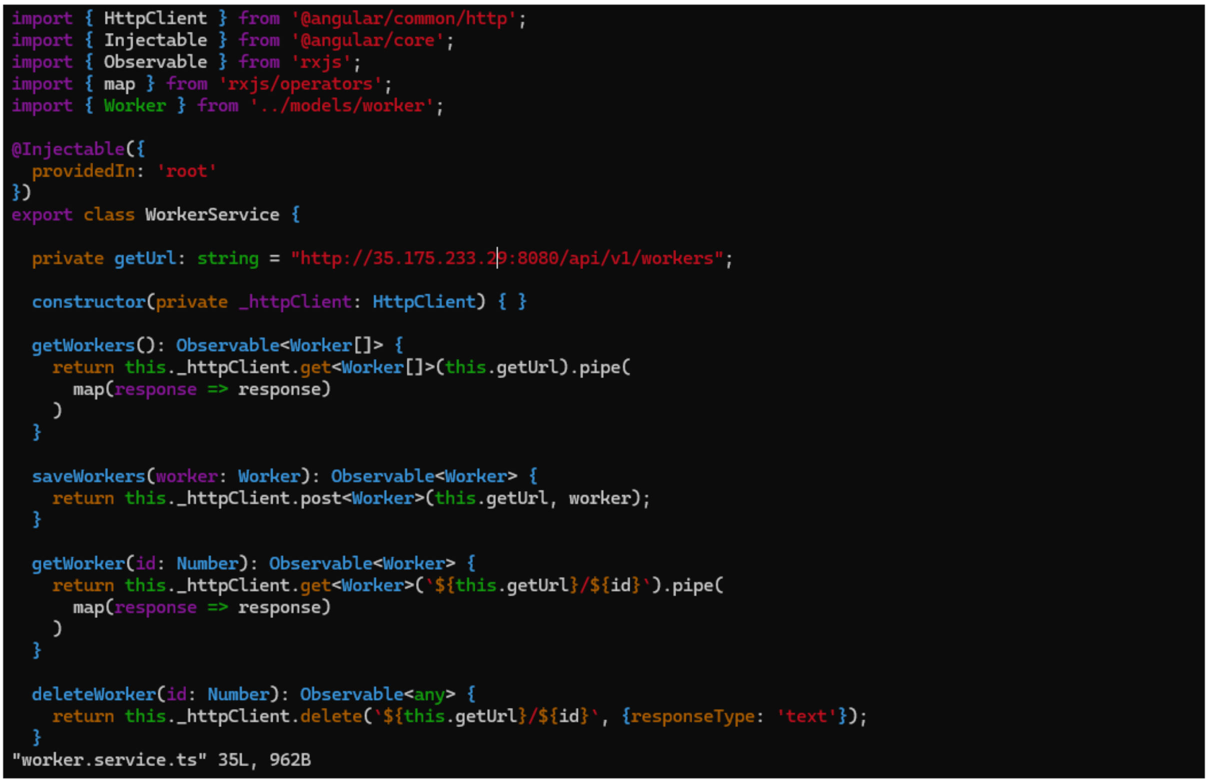
Edit the worker.service.ts file in anguler-frontend/src/app/services/worker.service.ts Give the public ip at the localhost



```
root@ip-172-31-88-34:~/anguler-java/angular-frontend# cd src/
root@ip-172-31-88-34:~/anguler-java/angular-frontend/src# ls
app  assets  environments  favicon.ico  index.html  main.ts  polyfills.ts  styles.css  test.ts
root@ip-172-31-88-34:~/anguler-java/angular-frontend/src# cd app
root@ip-172-31-88-34:~/anguler-java/angular-frontend/src/app# ls
app.component.css  app.component.html  app.component.spec.ts  app.component.ts  app.module.ts  components  models  services
root@ip-172-31-88-34:~/anguler-java/angular-frontend/src/app# cd services/
root@ip-172-31-88-34:~/anguler-java/angular-frontend/src/app/services# ls
worker.service.ts
root@ip-172-31-88-34:~/anguler-java/angular-frontend/src/app/services# vim worker.service.ts
root@ip-172-31-88-34:~/anguler-java/angular-frontend/src/app/services# cd ../../../
root@ip-172-31-88-34:~/anguler-java/angular-frontend# ls
README.md  angular.json  karma.conf.js  package-lock.json  package.json  src  tsconfig.app.json  tsconfig.json  tsconfig.spe
root@ip-172-31-88-34:~/anguler-java/angular-frontend# touch Dockerfile
root@ip-172-31-88-34:~/anguler-java/angular-frontend# ls
Dockerfile  README.md  angular.json  karma.conf.js  package-lock.json  package.json  src  tsconfig.app.json  tsconfig.json
root@ip-172-31-88-34:~/anguler-java/angular-frontend# vim Dockerfile
root@ip-172-31-88-34:~/anguler-java/angular-frontend# chmod o+x Dockerfile
root@ip-172-31-88-34:~/anguler-java/angular-frontend# ll
total 828
drwxr-xr-x 3 root root   4096 Sep 19 16:30 ./
drwxr-xr-x 5 root root   4096 Sep 19 16:07 ../
-rw-r--r-- 1 root root    600 Sep 19 16:07 .browserslistrc
-rw-r--r-- 1 root root    274 Sep 19 16:07 .editorconfig
-rw-r--r-x 1 root root    717 Sep 19 16:30 Dockerfile*
```



```
root@ip-172-31-88-34:~/anguler-java/angular-frontend/src/app/services# vim worker.service.ts
```

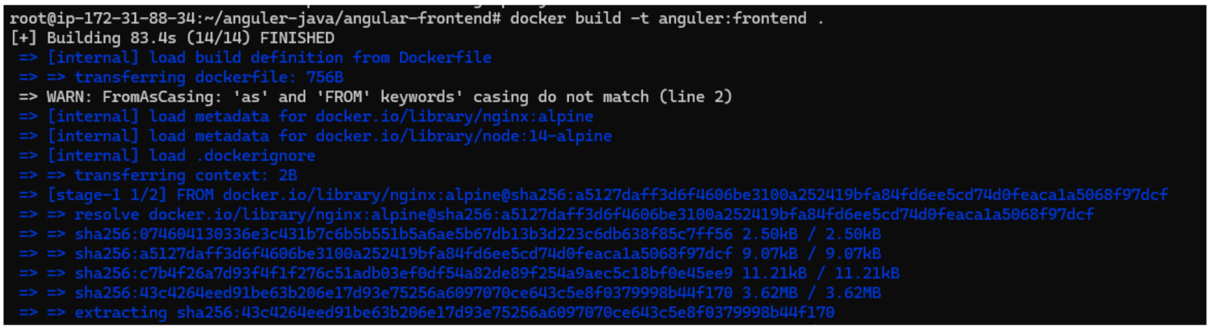


```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { map } from 'rxjs/operators';
import { Worker } from '../models/worker';

@Injectable({
  providedIn: 'root'
})
export class WorkerService {

  private getUrl: string = "http://35.175.233.29:8080/api/v1/workers";

  constructor(private _httpClient: HttpClient) { }

  getWorkers(): Observable<Worker[]> {
    return this._httpClient.get<Worker[]>(this.getUrl).pipe(
      map(response => response)
    )
  }

  saveWorkers(worker: Worker): Observable<Worker> {
    return this._httpClient.post<Worker>(this.getUrl, worker);
  }

  getWorker(id: Number): Observable<Worker> {
    return this._httpClient.get<Worker>(`${this.getUrl}/${id}`).pipe(
      map(response => response)
    )
  }

  deleteWorker(id: Number): Observable<any> {
    return this._httpClient.delete(`${this.getUrl}/${id}`, {responseType: 'text'});
  }
"worker.service.ts" 35L, 962B
```

Build and run the anguler-frontend using the command

Docker build -t anguler:frontend .

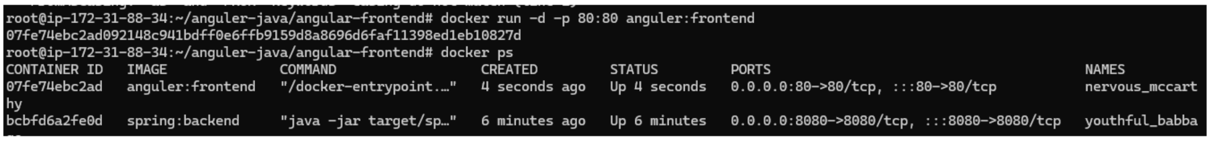
Docker run -d -p 80:80 anguler:frontend



```
root@ip-172-31-88-34:~/anguler-java/angular-frontend# docker build -t anguler:frontend .
[+] Building 83.4s (14/14) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 756B
 => WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 2)
 => [internal] load metadata for docker.io/library/nginx:alpine
 => [internal] load metadata for docker.io/library/node:14-alpine
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [stage-1 1/2] FROM docker.io/library/nginx:alpine@sha256:a5127daff3d6f4606be3100a252419bfa84fd6ee5cd74d0feaca1a5068f97dcf
 => => resolve docker.io/library/nginx:alpine@sha256:a5127daff3d6f4606be3100a252419bfa84fd6ee5cd74d0feaca1a5068f97dcf
 => => sha256:074604130336e3c431b7c6b5b551b5a6ae5b67db13b3d223c6db638f85c7ff56 2.50kB / 2.50kB
 => => sha256:a5127daff3d6f4606be3100a252419bfa84fd6ee5cd74d0feaca1a5068f97dcf 9.07kB / 9.07kB
 => => sha256:c7b4f26a7d93f4f1f276c51adb03ef0df54a82de89f254a9aec5c18bf0e45ee9 11.21kB / 11.21kB
 => => sha256:43c4264eed91be63b206e17d93e75256a6097070ce643c5e8f0379998b44f170 3.62MB / 3.62MB
 => => extracting sha256:43c4264eed91be63b206e17d93e75256a6097070ce643c5e8f0379998b44f170
```

Check the images using the command docker ps

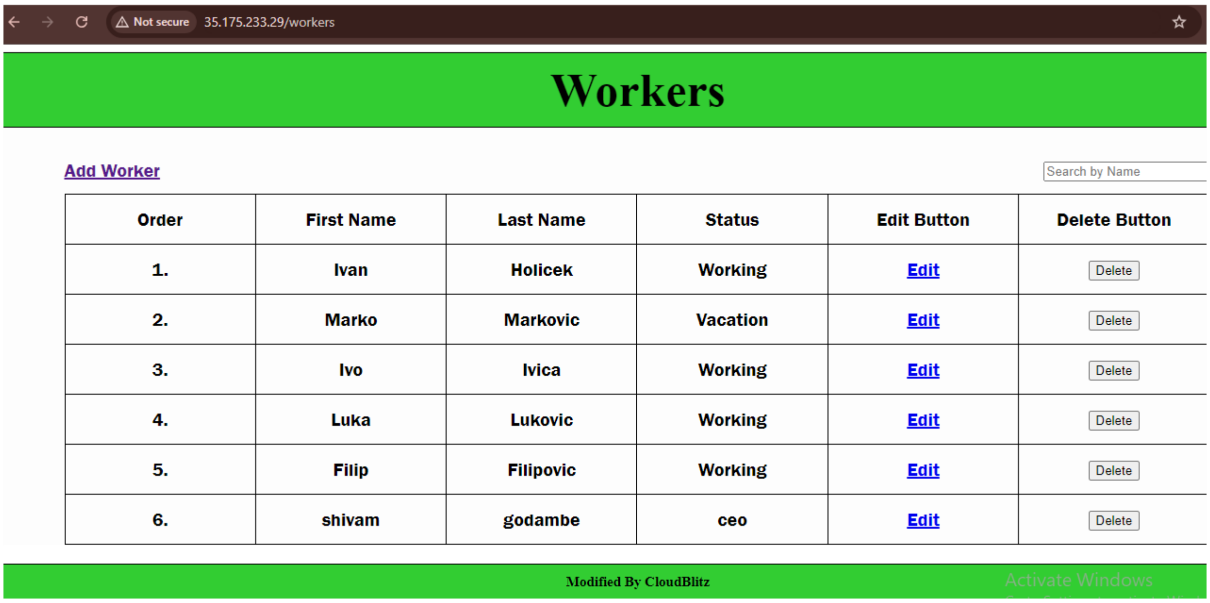
It show the container



```
root@ip-172-31-88-34:~/anguler-java/angular-frontend# docker run -d -p 80:80 anguler:frontend
07fe74ebc2ad092148c941bdff0e6ffb9159d8a8696d6faf11398ed1eb10827d
root@ip-172-31-88-34:~/anguler-java/angular-frontend# docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                       NAMES
07fe74ebc2ad   anguler:frontend   "/docker-entrypoint.…"   4 seconds ago    Up 4 seconds    0.0.0.0:80->80/tcp, :::80->80/tcp           nervous_mccart
hy
bcbfd6a2fe0d   spring:backend     "java -jar target/sp…"   6 minutes ago    Up 6 minutes    0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   youthful_babba
```

Hit the public ip address provided by the instance

Successfully hosted



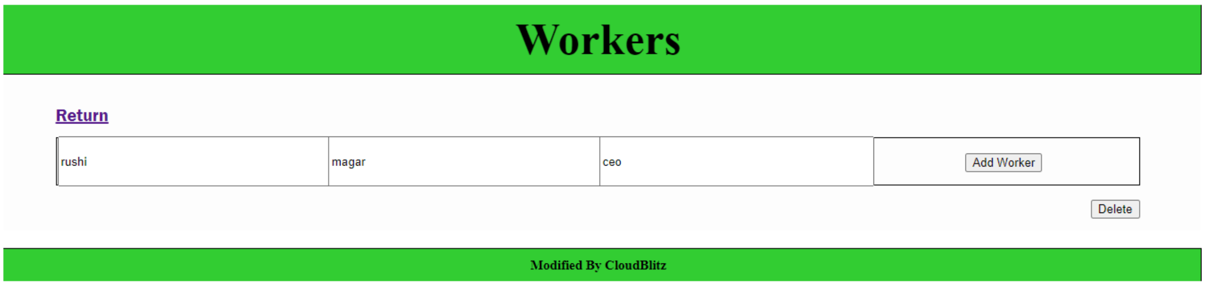
Workers

Add Worker

Order	First Name	Last Name	Status	Edit Button	Delete Button
1.	Ivan	Holicek	Working	Edit	Delete
2.	Marko	Markovic	Vacation	Edit	Delete
3.	Ivo	Ivica	Working	Edit	Delete
4.	Luka	Lukovic	Working	Edit	Delete
5.	Filip	Filipovic	Working	Edit	Delete
6.	shivam	godambe	ceo	Edit	Delete

Modified By CloudBlitz

Adding the data of user



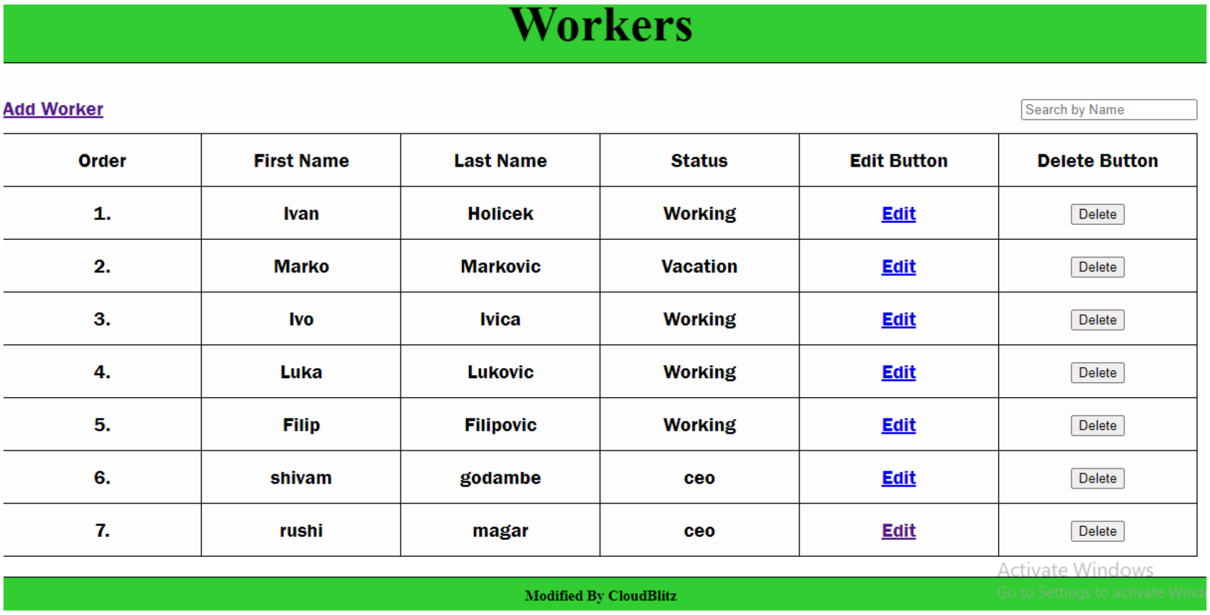
Workers

Return

rushi	magar	ceo	Add Worker

Modified By CloudBlitz

Successfully adding the data to the worker table



Workers

Order	First Name	Last Name	Status	Edit Button	Delete Button
1.	Ivan	Holicek	Working	Edit	Delete
2.	Marko	Markovic	Vacation	Edit	Delete
3.	Ivo	Ivica	Working	Edit	Delete
4.	Luka	Lukovic	Working	Edit	Delete
5.	Filip	Filipovic	Working	Edit	Delete
6.	shivam	godambe	ceo	Edit	Delete
7.	rushi	magar	ceo	Edit	Delete

Modified By CloudBlitz