We require two actor

User or Customer etc

Admin

Admin -🡪 SignIn no sign up for admin we have create admin account using database with admin email : [admin@gmail.com](mailto:admin@gmail.com) and password : admin@123

User or Customer -🡪 SignIn and SignUp

Login Table

emailId (PK) password role

[admin@gmail.com](mailto:admin@gmail.com) admin@123 admin

[raj@gmail.com](mailto:raj@gmail.com) 1234 user

[seeta@gmail.com](mailto:seeta@gmail.com) 123123 user

login page

emailId textfield

password passwordfield

submit reset

SignUp

Admin Dashboard

Add , view, update and delete or more , Admin can view all user details.

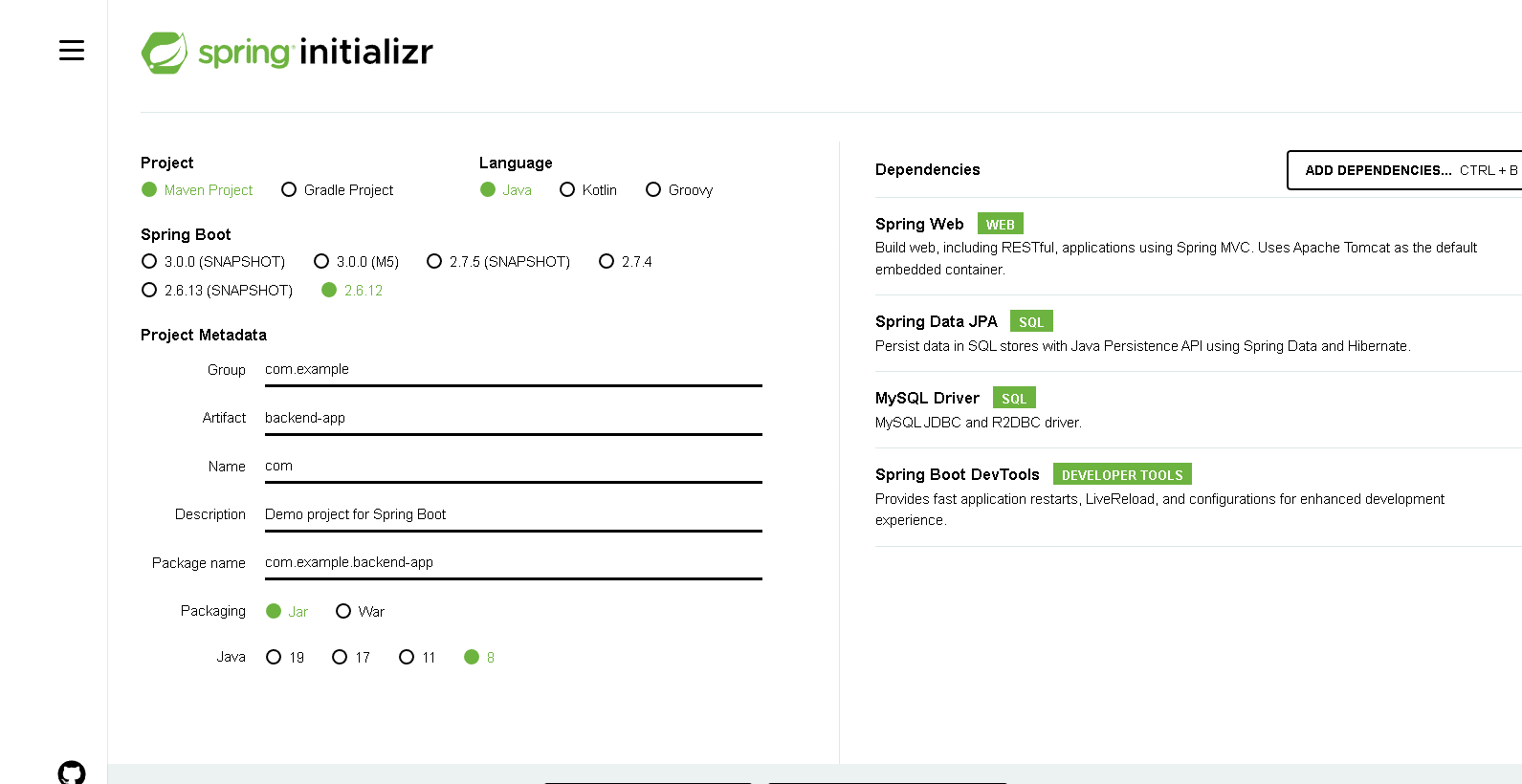
User Dashboard

View all details , Place order, View Order, View its profile details

Capstone Project folder create

Two folder create

Backend : create the spring boot project using spring initialization



Frontend :

ng new frontend-app

with routing Yes

ng g c login

ng g c signup

ng g c admindashboard

ng g c userdashboard

ng g class login

ng g s login

Docker compose for spring boot and Mysql Database

If want to run more than one container and those container interacting with each others.

Like Spring boot application want to connect the mysql database.

Myspring boot image

My sql image

Network environment

Spring-boot-application mysql-image we pull and set username

Password and database name.

docker compose : Docker compose is a tool that was developed to help to defined and share more than one container application like Spring boot and mysql database. With docker compose concept we have to make YML file and inside this file we have to provide both images details and their network environment information. With help of docker compose we can up all container down, build, stop, start etc.

first create the spring boot application with controller, service, dao, bean

then in application.properties file write database details

server.port=9090

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://mysqldb/mydb

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

mysqldb 🡪 host name

mydb 🡪 database name

root 🡪username

root -> password

hibernate ddl property 🡪 auto table creation.

Now using mvn package create the jar file.

When we run mvn package in spring boot initializer contains testing dependencies and that dependencies check database details. So to avoid this error we have to remove testing dependencies remove.

**So please remove testing dependencies.**

Then run the command as mvn package

After jar file created successfully.

Now we have to create the image for spring boot

Create the Dockerfile

**FROM** openjdk:11

**COPY** target/spring-boot-with-docker.jar .

**CMD** ["java","-jar","spring-boot-with-docker.jar"]

Then create the Docker image

docker build -t spring-boot-app . -f Dockerfile

Now create the docker-compose.yml file

version: "3.0"

services:

spring-demo:

image: spring-boot-app

ports:

- "9292:9090"

networks:

- spring-boot-mysql

depends\_on:

- mysqldb

mysqldb:

image: mysql:8

networks:

- spring-boot-mysql

ports:

- "3308:3306"

environment:

- MYSQL\_ROOT\_PASSWORD=root

- MYSQL\_DATABASE=mydb

networks:

spring-boot-mysql:

Then check the docker-compose –-version

docker-compose build : this command use to build the image

docker-compose up this command run both the images

docker-compose up –d This command is use to run both image in background

docker-compose down This command is use to stop all container through file.

To check the docker database

Run the below command with container name or containerId

docker exec -it 1748fb9e6d28 bash

After run the command as

mysql –u root –p

* Root