Student.java

```
package com.TestJPA.TestJPA;

//imports

@Entity
@Table(name="student")
public class Student {

	@Id
	@GeneratedValue(strategy = GenerationType.SEQUENCE)
	int id;

	int totalmarks;

	String usn,name, address;

				/// setter and getter methods
	@Override
	public String toString() {
		return "Student [id=" + id + ", totalmarks=" + totalmarks + ", usn=" + usn + ", 
name=" + name + ", address="
				+ address + "]";
	}
}
```

StudentMain.java

```
package com.TestJPA.TestJPA;

//imports

@SpringBootApplication
public class StudentMain {
	public static void main(String[] args) {
		SpringApplication.run(StudentMain.class, args);
	}
}
```

StudentRepo.java //Internship

```
package com.TestJPA.TestJPA;

import org.springframework.data.jpa.repository.JpaRepository;

public interface StudentRepo extends JpaRepository<Student, Integer> {

}
```

StudentService.java

```
package com.TestJPA.TestJPA;

import java.util.List;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class StudentService {

	@Autowired
	StudentRepo sr;

	public void insertStudent(String name,String address,String usn,int totalmarks) {
		Student s = new Student();
		s.setAddress(address);
		s.setTotalmarks(totalmarks);
		s.setName(name);
		s.setUsn(usn);
		sr.save(s);
	}

	public List<Student> display(){
		return sr.findAll();
	}

	public void remove(int id) {
		// TODO Auto-generated method stub
		Optional<Student> s =sr.findById(id);
		if(s.isPresent()){
			sr.deleteById(id);
		}else {
			System.out.println("not found by id"+id);
		}
	}
}
```

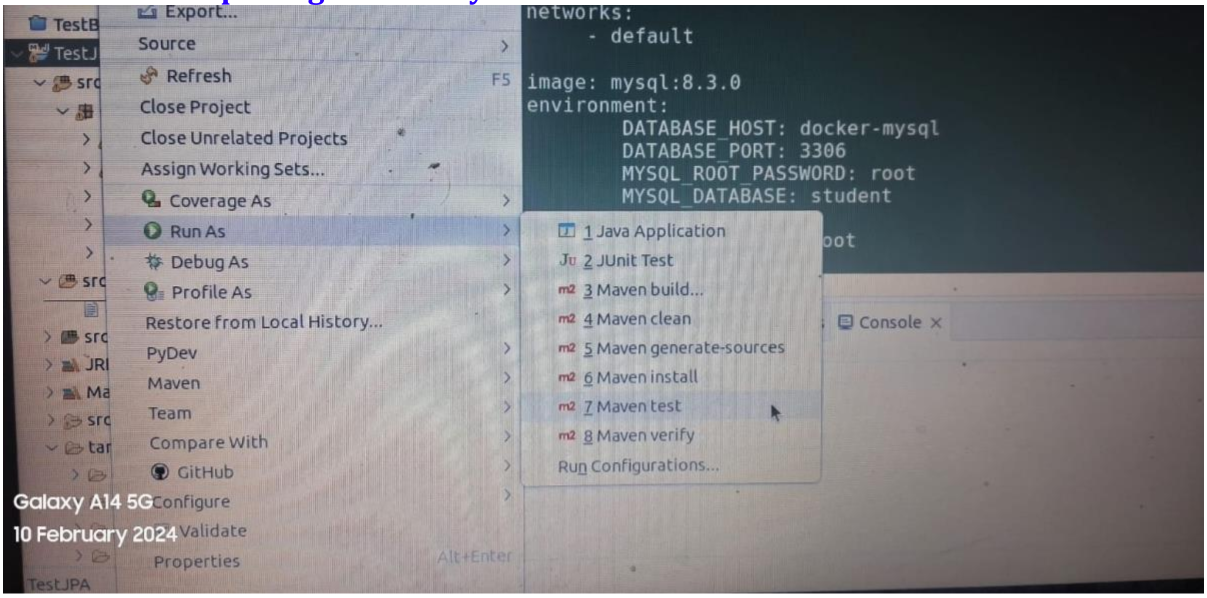
StudentWeb.java

```
package com.TestJPA.TestJPA;

import java.util.List;

//imports

@RestController
@RequestMapping("/student")
public class StudentWeb {
    @Autowired
    StudentService ss;

    @GetMapping("/get")
    public List<Student> display(){
        return ss.display();
    }

    @PostMapping("/post")
    public void insertStudent(@RequestBody Student s) {
        ss.insertStudent(s.getName(),s.getAddress(),s.getUsn(),s.getTotalmarks());
        System.out.println(" inserted successfully");
    }

    @DeleteMapping("/{id}")
    public void deleteStudent(@PathVariable int id ) {
        ss.remove(id);
    }
}
```

application.properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/student
spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver
spring.datasource.username=root
spring.datasource.password=rvce
spring.jpa.show-sql: true
spring.jpa.hibernate.ddl-auto=update
```

Install the maven packages in the system:



```
networks:
    - default

image: mysql:8.3.0
environment:
        DATABASE_HOST: docker-mysql
        DATABASE_PORT: 3306
        MYSQL_ROOT_PASSWORD: root
        MYSQL_DATABASE: student
```

Dockerfile

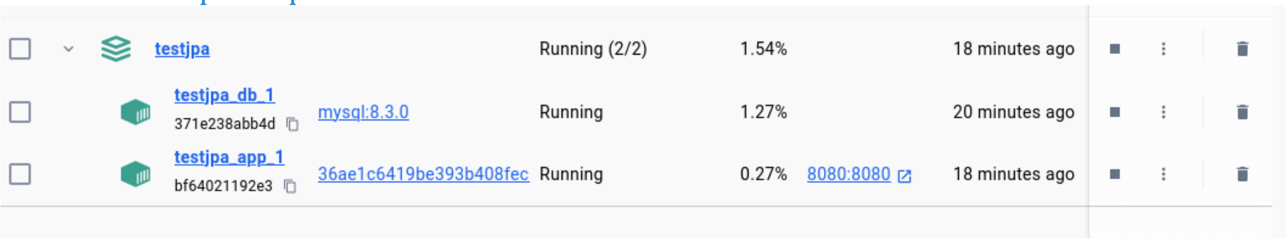
```
FROM openjdk:17-alpine
WORKDIR /opt
ENV PORT 8080
EXPOSE 8080
COPY /target/*.jar /opt/app.jar
ENTRYPOINT exec java $JAVA_OPTS -jar app.jar

#Build a docker image
$docker build --build-arg JAR_FILE=target/*.jar -t myorg/myapp .
```

Docker-compose.yml

```
version: "3.8"

services:
```

```
app:
  image: 36ae1c6419be393b408fec0ee86e5b2539e7ba018b1d9c581bdfa7f63c381c3f
  ports:
  - 8080:8080
  environment:
      spring.datasource.url: jdbc:mysql://db:3306/student
      spring.datasource.driverClassName: com.mysql.cj.jdbc.Driver
      spring.datasource.username: admin
      spring.datasource.password: root
  depends_on:
     - db
  networks:
     - default
db:
  image: mysql
  environment:
       DATABASE_HOST: docker-mysql
       DATABASE_PORT: 3306
       MYSQL_ROOT_PASSWORD: root
       MYSQL_DATABASE: student
       MYSQL_USER: admin
       MYSQL_PASSWORD: root
  networks:
   - default
```

```
>>docker-compose up
```



☐	testjpa		Running (2/2)	1.54%		18 minutes ago
☐	testjpa_db_1 371e238abb4d	mysql:8.3.0	Running	1.27%		20 minutes ago
☐	testjpa_app_1 bf64021192e3	36ae1c6419be393b408fec	Running	0.27%	8080:8080	18 minutes ago