

**Name:SRIHARIVISHNU JANAKIRAMAN**

**ASSIGNMENT NO:5**

**DATE:14/12/2022**

- 1. Implement CRUD Operations with REST API bean validations for the customers' information**

**Custid→ auto generated**

**Cust\_name → No null/empty/blank and no special chars ,numerics**

**Location → No null/empty/blank and no special chars ,numerics**

**Email -----→ valid email id**

**Phone number --- in the format of +91-9999999999**

**Salesamount--→ \$10000**

**Output:**

**1) Cust\_Id→ auto generated**

```
@Id  
@GeneratedValue(strategy = GenerationType.IDENTITY)  
int Cust_Id;
```

## 2)Cust\_name → No null/empty/blank and no special chars ,numeric

The screenshot shows the Postman interface with a REST client request configured as follows:

- Method:** POST
- URL:** http://localhost:8082/addList
- Body (JSON):**

```
{  "location": "",  "cust_Id": 2,  "cust_name": "",  "email": "abc@gmail.com",  "salesAmount": 100000.0,  "phone_Number": "3216549879"}
```

The response status is **400 Bad Request** with the following JSON body:

```
{  "time": "2022-12-14T11:59:33.323+00:00",  "message": "Name Shouldn't Empty",  "errorDetails": "Validation Error"}
```

The error message indicates that the 'cust\_name' field is invalid because it is empty.

## 3)Location → No null/empty/blank and no special chars ,numeric

The screenshot shows the Postman interface with a REST client request configured as follows:

- Method:** POST
- URL:** http://localhost:8082/addList
- Body (JSON):**

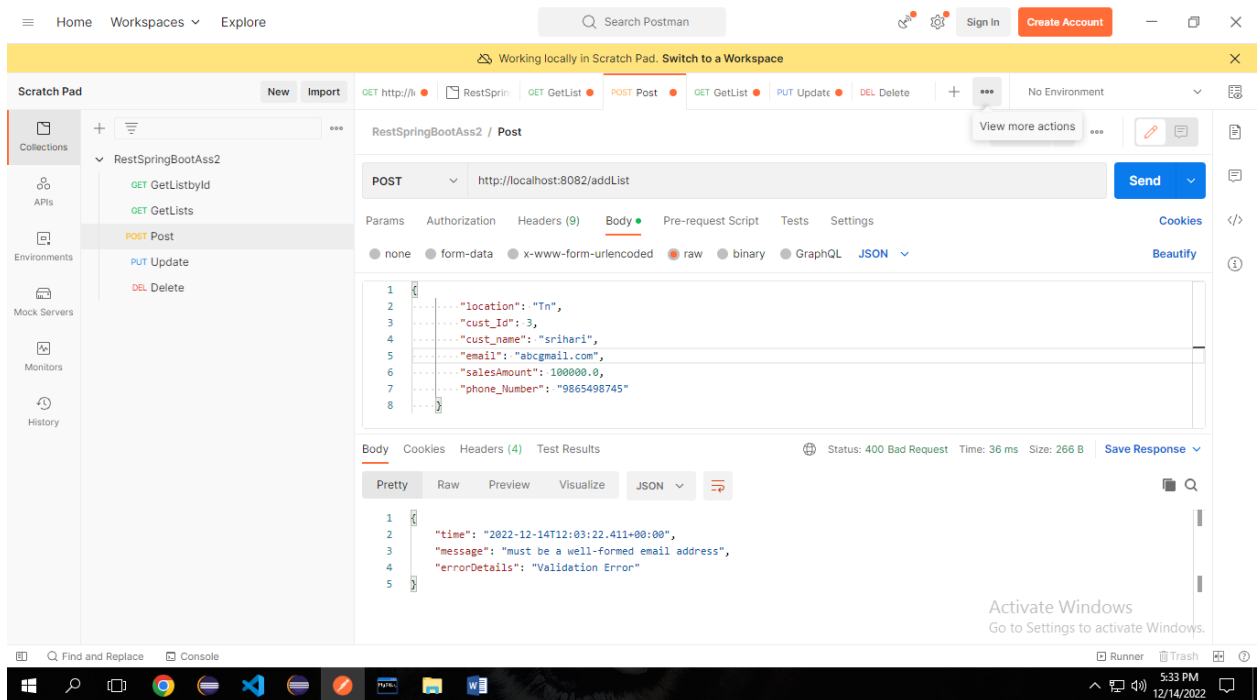
```
{  "location": "",  "cust_Id": 2,  "cust_name": "srihari",  "email": "abc@gmail.com",  "salesAmount": 100000.0,  "phone_Number": "3216549879"}
```

The response status is **400 Bad Request** with the following JSON body:

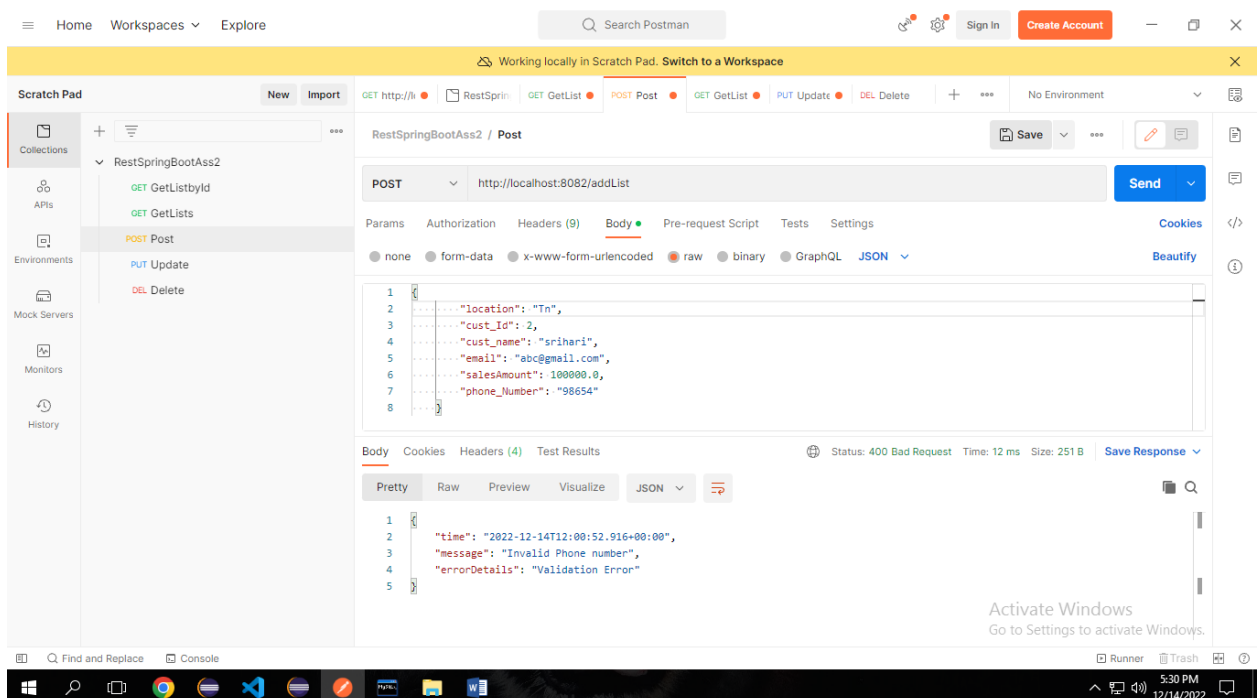
```
{  "time": "2022-12-14T12:00:08.189+00:00",  "message": "Location Shouldn't Empty",  "errorDetails": "Validation Error"}
```

The error message indicates that the 'location' field is invalid because it is empty.

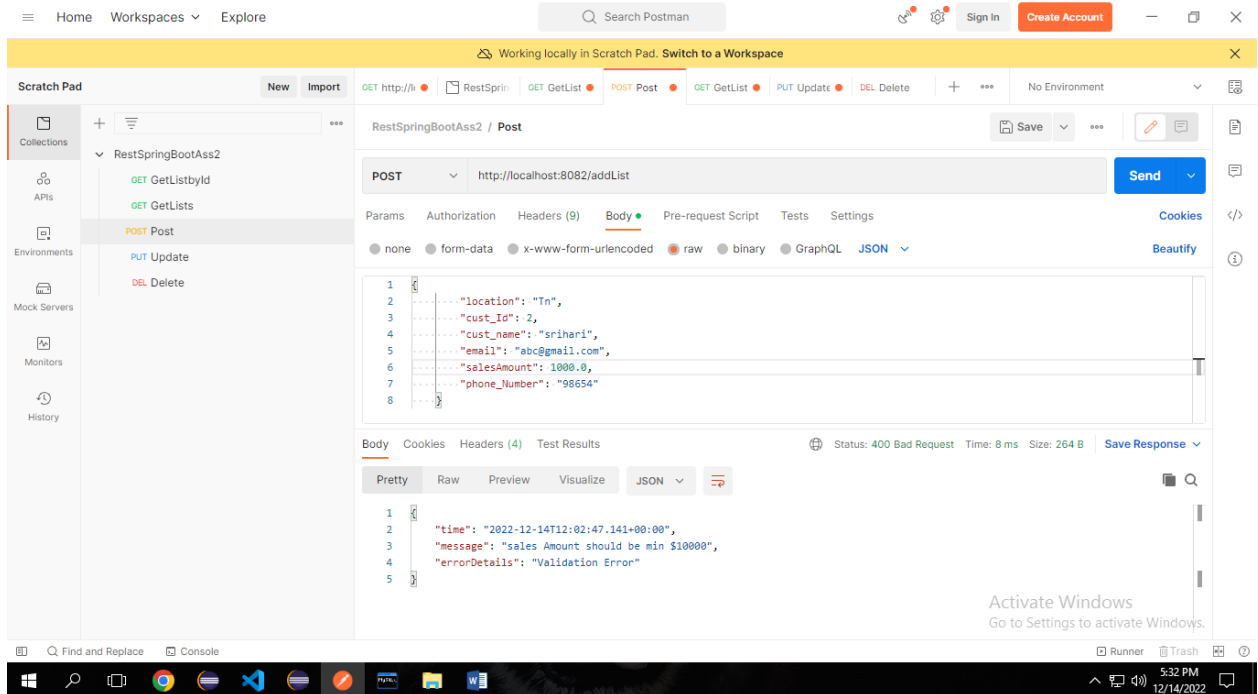
## 4) Email -----> valid email id



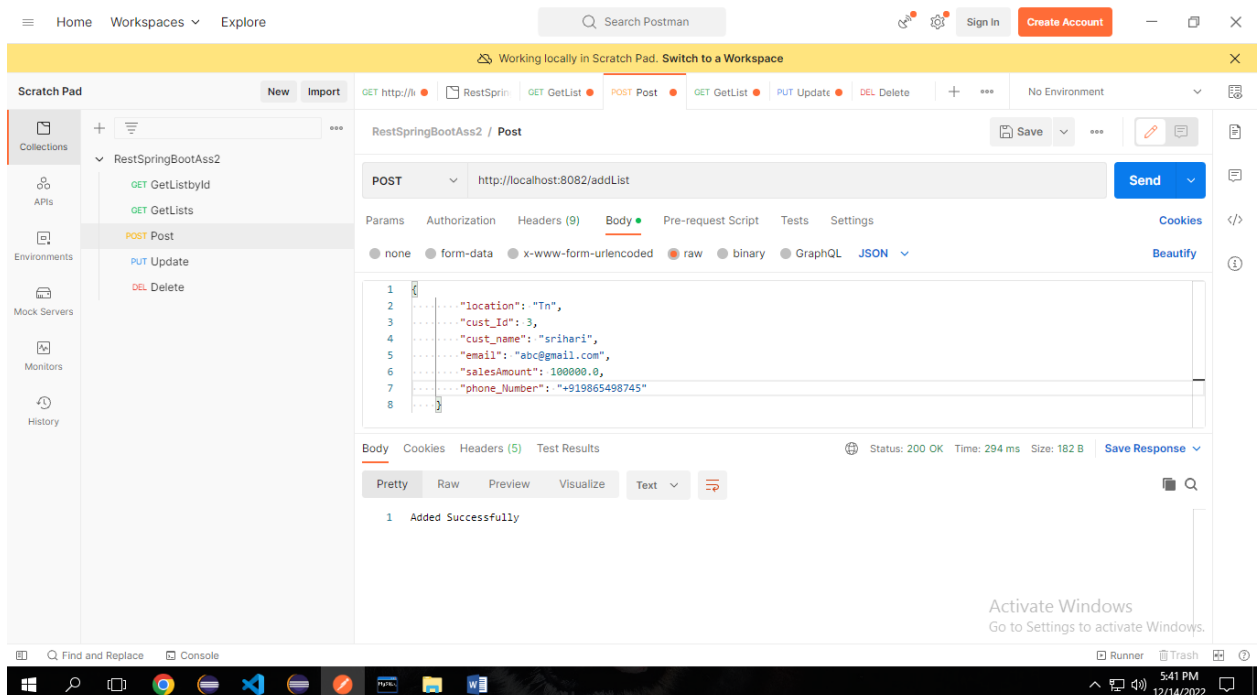
## 5) Phone number --- in the format of +91-9999999999



## 6) Salesamount--→ \$10000

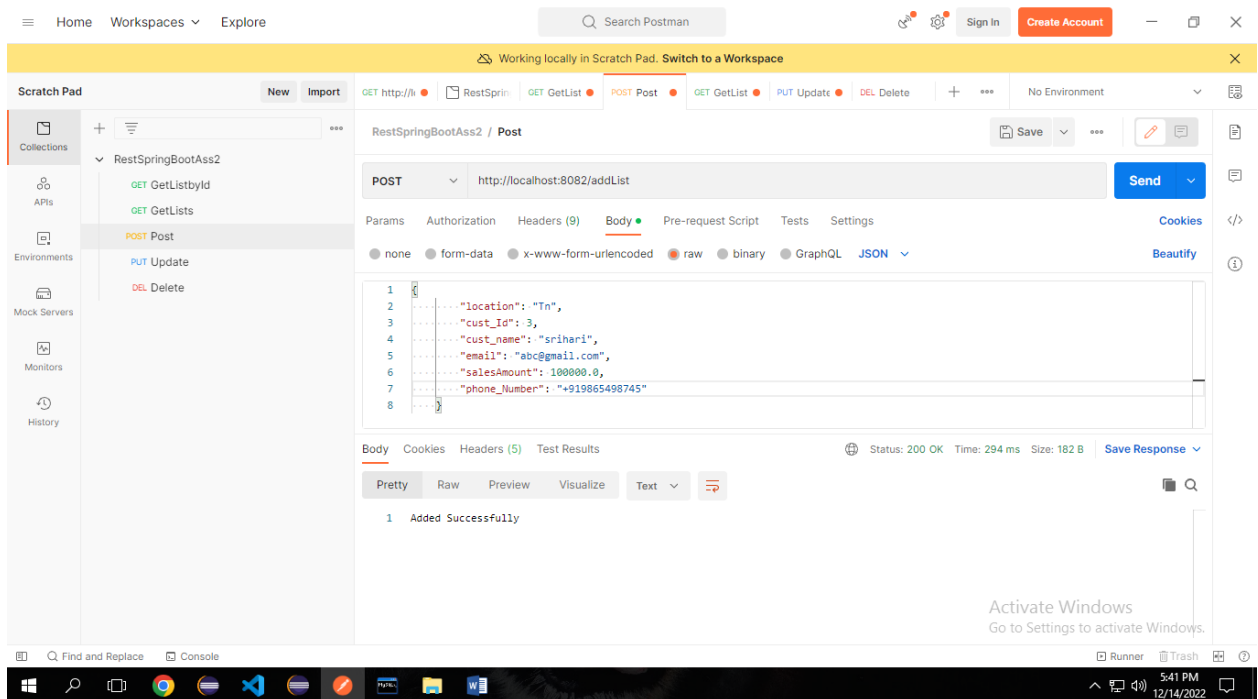


## 7)After giving all valid Credentials

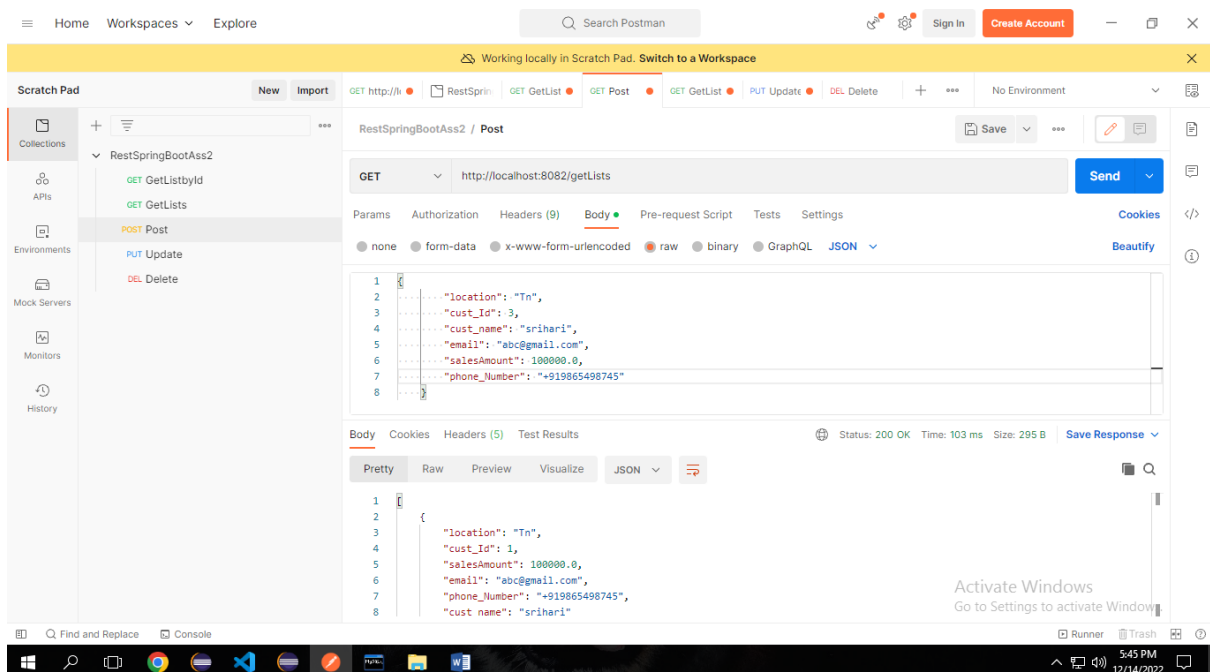


## Crud Operations:

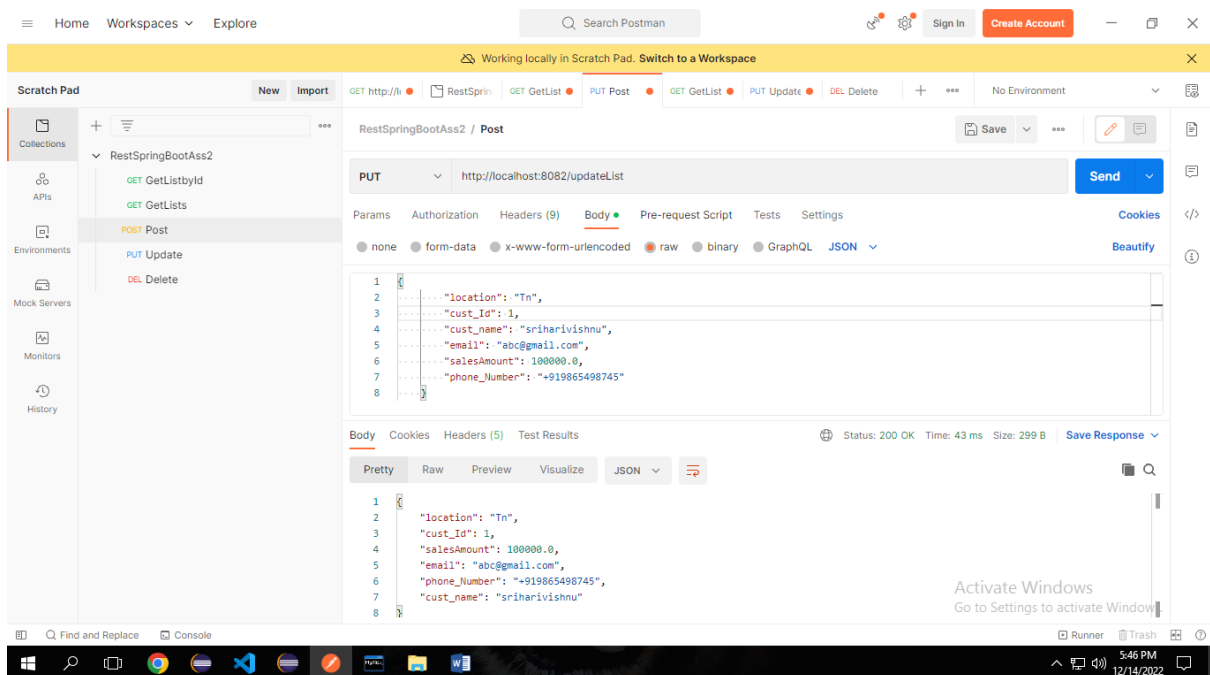
### Adding data into the database:



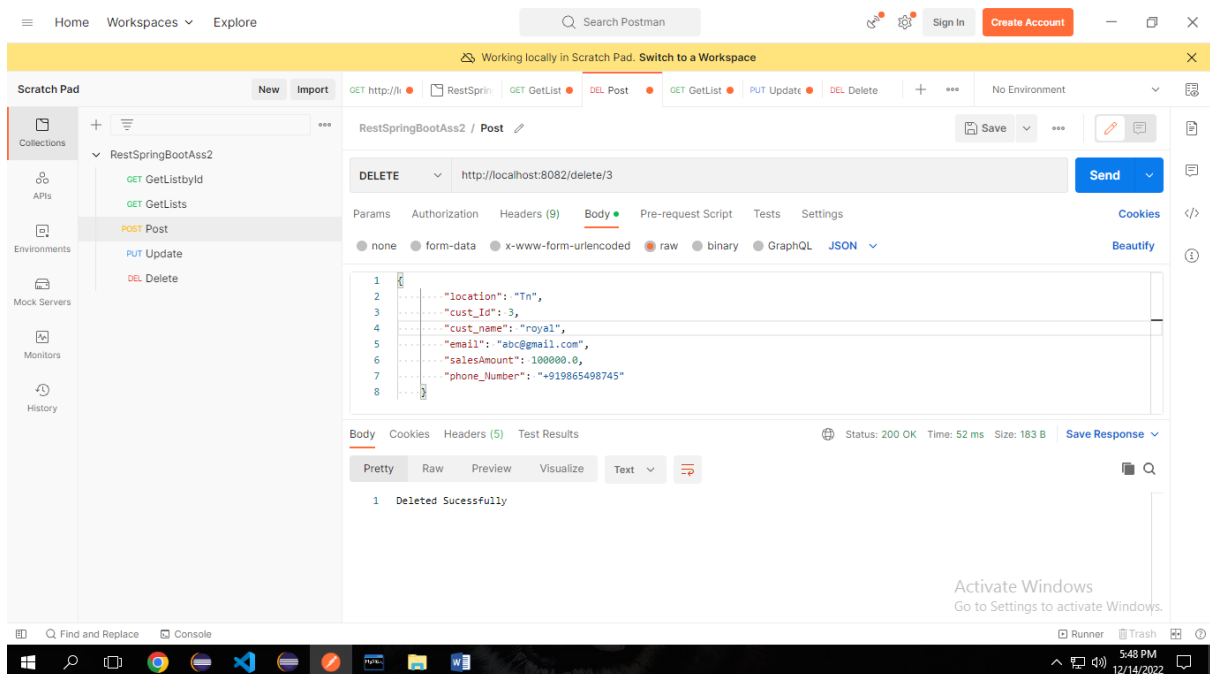
### 2) Getting data from database:



### 3)updating Data :



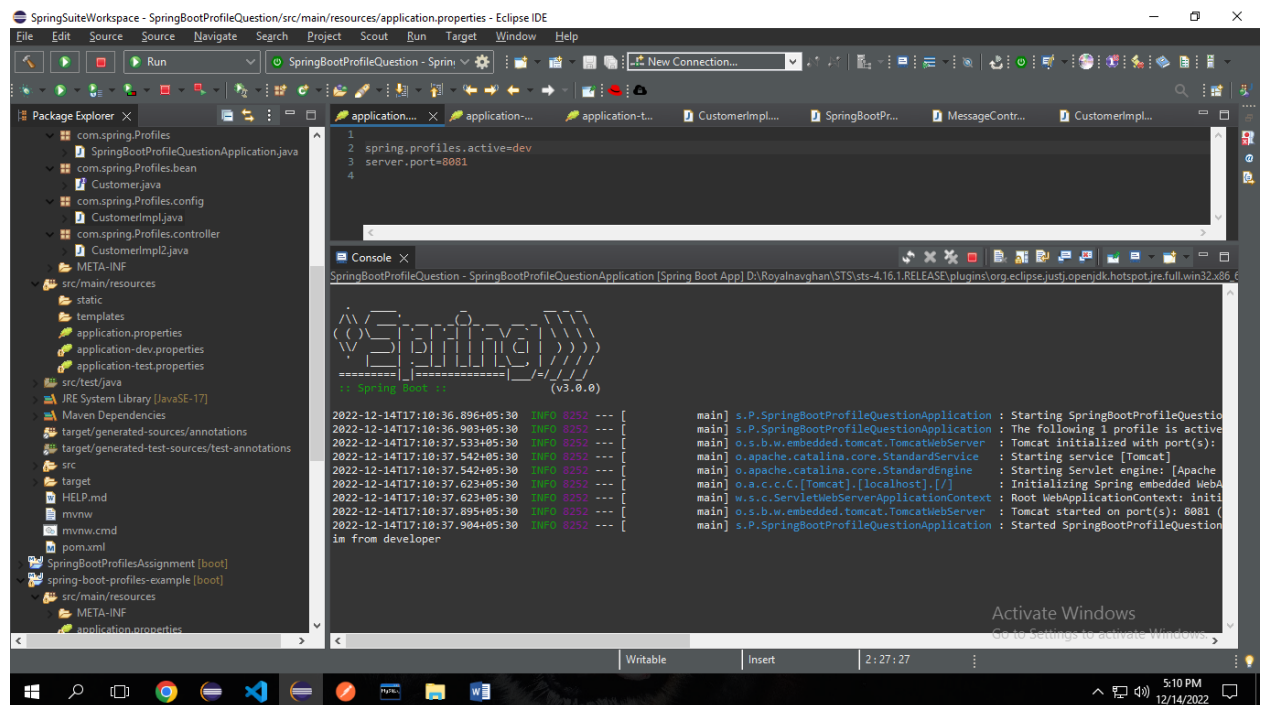
### 4)deleting data from database:



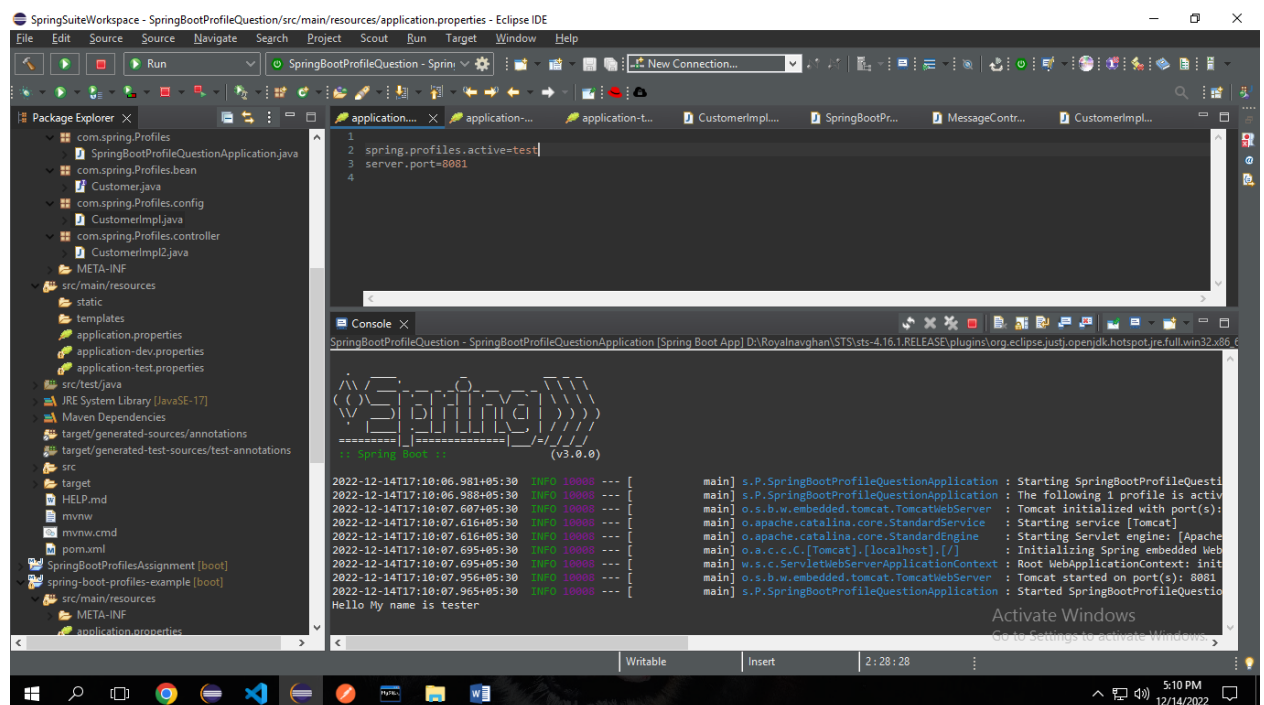
## 2. Implement spring profile for various environments DEV.QA and PROD using Property file inheritance.

### Outputs Screenshots:

#### 1)From developer side:



#### 2)From Tester side:



### 3. Implement JPA query annotation to build below user defined methods for the Supplier Table.

Suppld

SuppName

SuppLocation

SuppDate

SuppQty.

suppcost

1.findbysuppidandname()

2.findbysuppidorname()

3.findsuppllocation()

4. Find aggregated info over the suppcost field.

Output Screenshot: