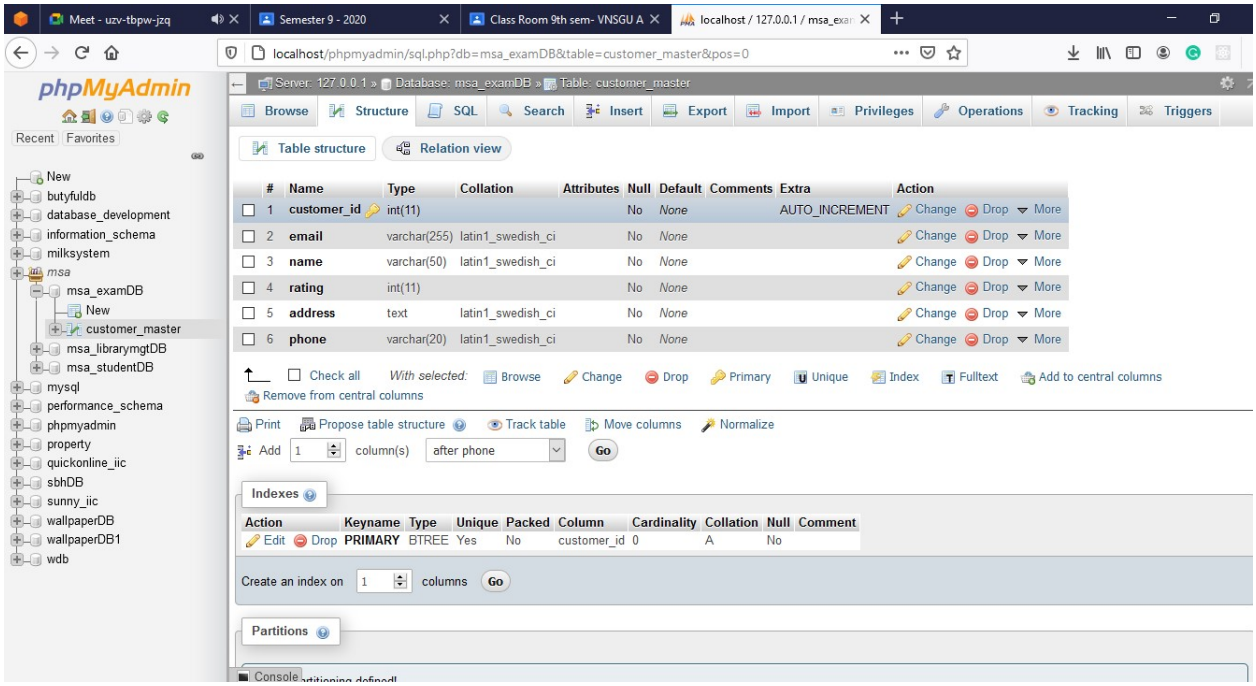
Name: Sanket H. Vanani

No: 30

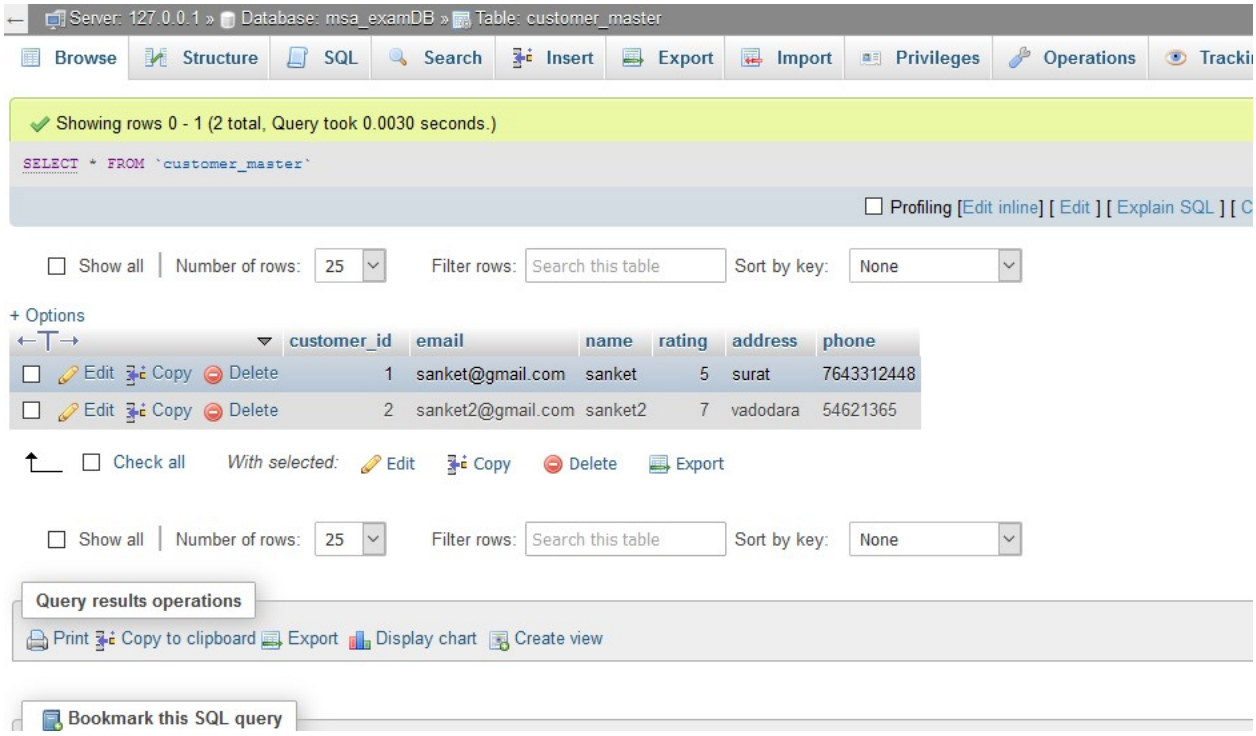
Sem: IT 9th

Subject : Cloud Computing

1. Create Database in mysql through PHPMyadmin.

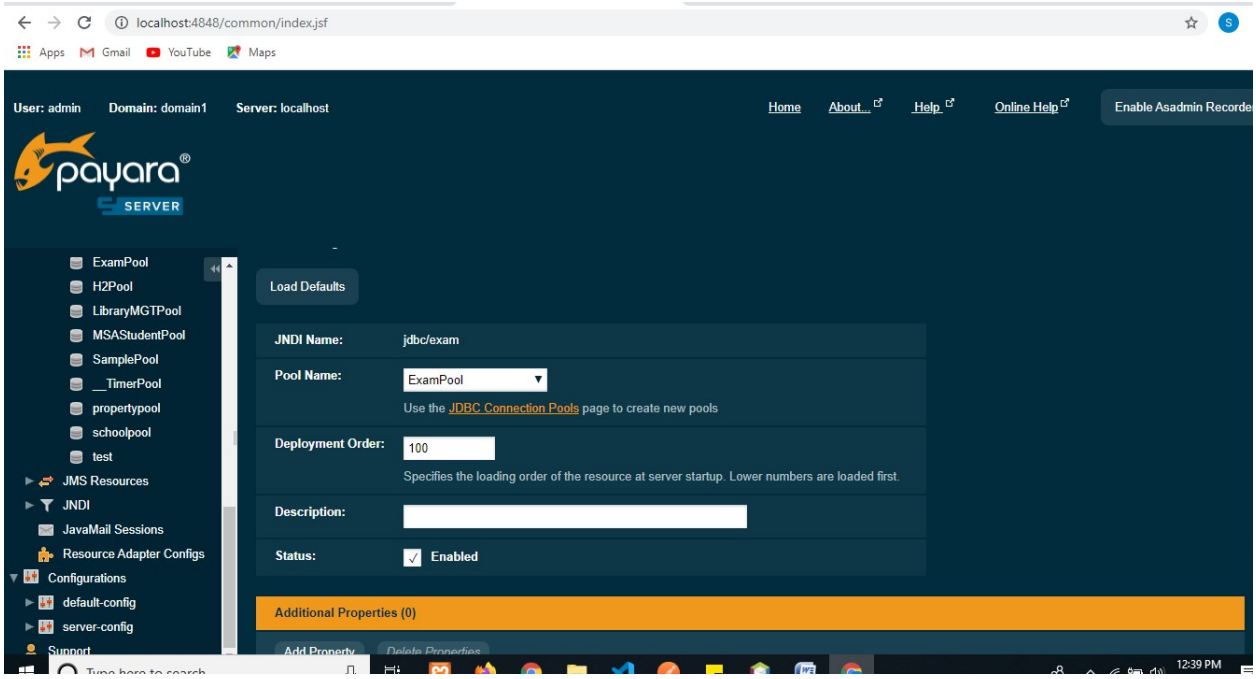


2. Add 2-3 data for testing.

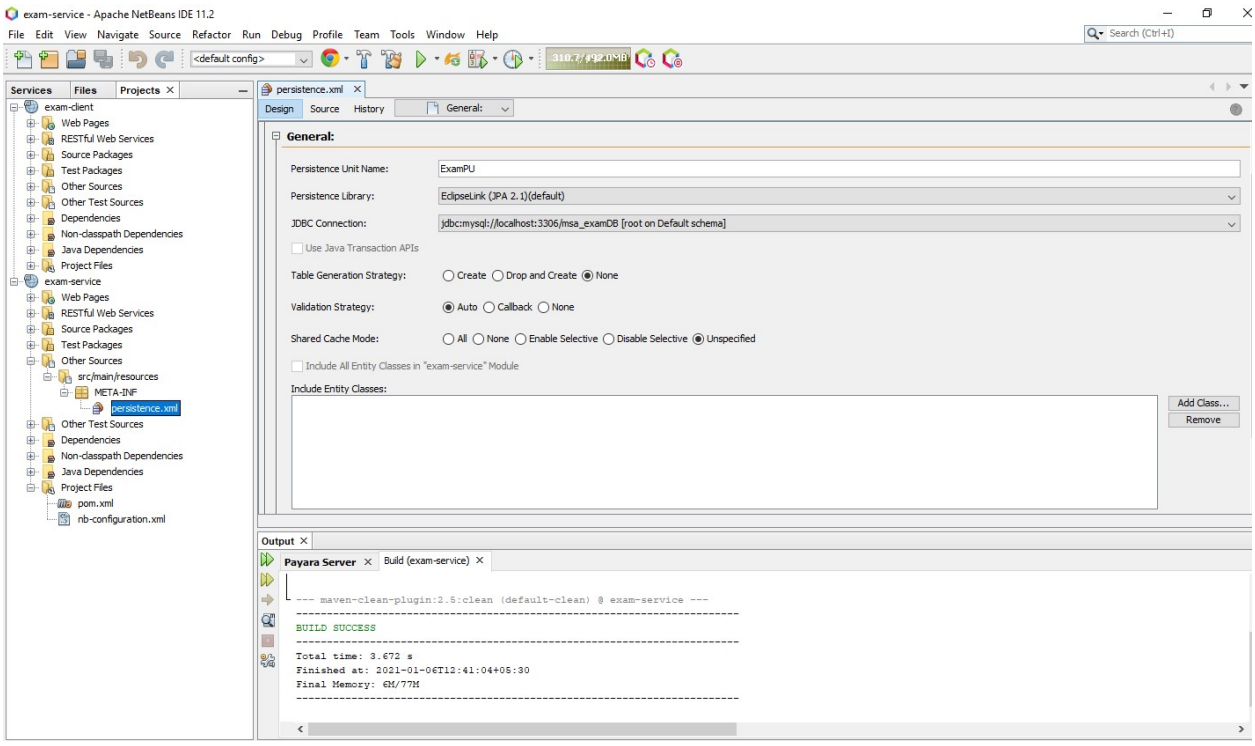


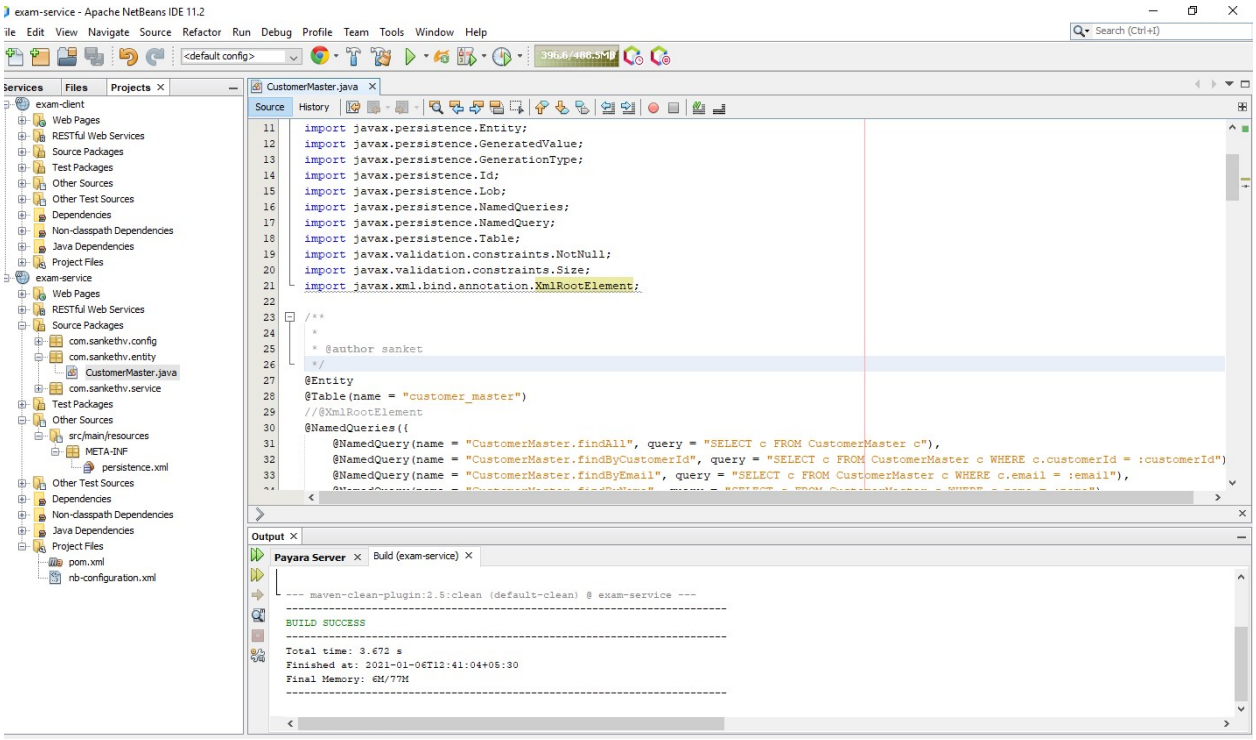
3. **For creating MSA Application we need to make Project with Archtype and I have created 2 project one called exam-client and another one is exam-service.**

4. **Then we need to create the JDBC pool and JDBC resource in Payara Server.**

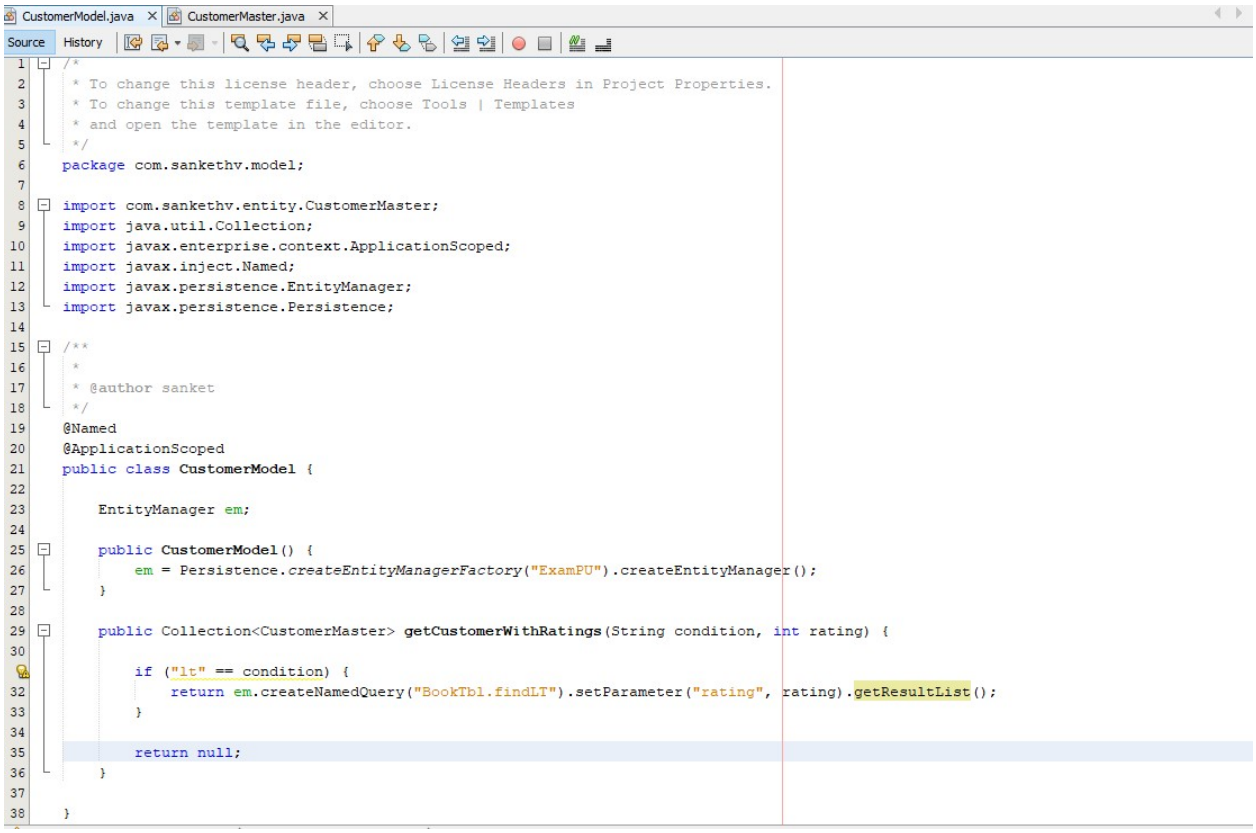


5. Now we will create Persistence unit and Entity from the database.





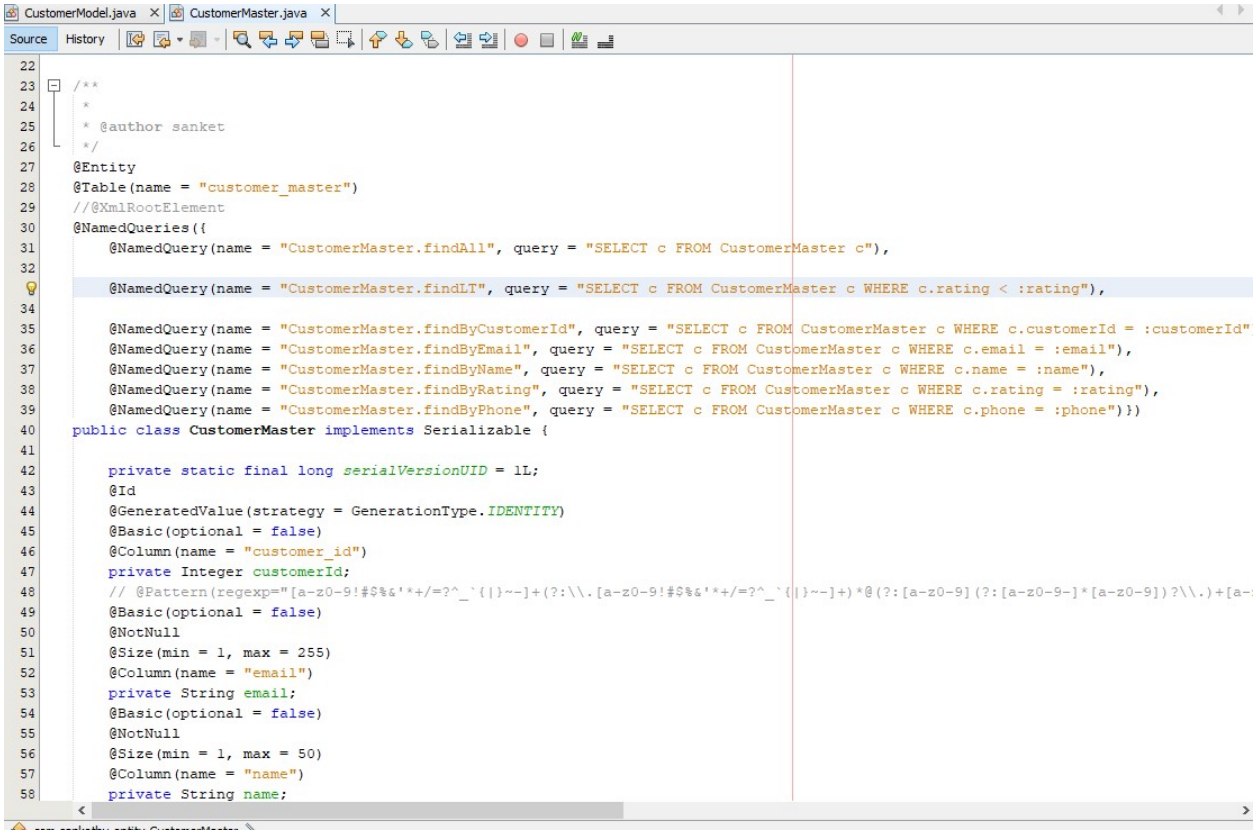
6. Now we will create the model Customer and add query as per our requirements in Entity.




```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sankethv.model;

import com.sankethv.entity.CustomerMaster;
import java.util.Collection;
import javax.enterprise.context.ApplicationScoped;
import javax.inject.Named;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;

/**
 *
 * @author sanket
 */
@Named
@ApplicationScoped
public class CustomerModel {

    EntityManager em;

    public CustomerModel() {
        em = Persistence.createEntityManagerFactory("ExamPU").createEntityManager();
    }

    public Collection<CustomerMaster> getCustomerWithRatings(String condition, int rating) {

        if ("lt" == condition) {
            return em.createNamedQuery("BookTbl.findLT").setParameter("rating", rating).getResultList();
        }

        return null;
    }

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sankethv.model;

import com.sankethv.entity.CustomerMaster;
import java.util.Collection;
import javax.enterprise.context.ApplicationScoped;
import javax.inject.Named;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;

/**
 *
 * @author sanket
 */
@Named
@ApplicationScoped
public class CustomerModel {

    EntityManager em;

    public CustomerModel() {
```

```
        em = Persistence.createEntityManagerFactory("ExamPU").createEntityManager();
    }

    public Collection<CustomerMaster> getCustomerWithRatings(String condition, int rating) {

        if ("lt" == condition) {
            return em.createNamedQuery("BookTbl.findLT").setParameter("rating", rating).getResultList();
        }

        return null;
    }

}
```



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sankethv.entity;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Lob;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author sanket
```

```
 */
@Entity
@Table(name = "customer_master")
//@XmlRootElement
@NamedQueries({
   @NamedQuery(name = "CustomerMaster.findAll", query = "SELECT c FROM CustomerMaster c"),

   @NamedQuery(name = "CustomerMaster.findLT", query = "SELECT c FROM CustomerMaster c
WHERE c.rating < :rating"),

   @NamedQuery(name = "CustomerMaster.findByCustomerId", query = "SELECT c FROM
CustomerMaster c WHERE c.customerId = :customerId"),
   @NamedQuery(name = "CustomerMaster.findByEmail", query = "SELECT c FROM CustomerMaster
c WHERE c.email = :email"),
   @NamedQuery(name = "CustomerMaster.findByName", query = "SELECT c FROM CustomerMaster
c WHERE c.name = :name"),
   @NamedQuery(name = "CustomerMaster.findByRating", query = "SELECT c FROM CustomerMaster
c WHERE c.rating = :rating"),
   @NamedQuery(name = "CustomerMaster.findByPhone", query = "SELECT c FROM CustomerMaster
c WHERE c.phone = :phone")})
public class CustomerMaster implements Serializable {

   private static final long serialVersionUID = 1L;
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   @Basic(optional = false)
   @Column(name = "customer_id")
   private Integer customerId;
```

// @Pattern(regexp="[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?", message="Invalid email")//if the field contains email address consider using this annotation to enforce field validation

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 255)

@Column(name = "email")

private String email;

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 50)

@Column(name = "name")

private String name;

@Basic(optional = false)

@NotNull

@Column(name = "rating")

private int rating;

@Basic(optional = false)

@NotNull

@Lob

@Size(min = 1, max = 65535)

@Column(name = "address")

private String address;

// @Pattern(regexp="^\\(?(\\d{3})\\)?[-]?(\\d{3})[-]?(\\d{4})$", message="Invalid phone/fax format, should be as xxx-xxx-xxxx")//if the field contains phone or fax number consider using this annotation to enforce field validation

@Basic(optional = false)

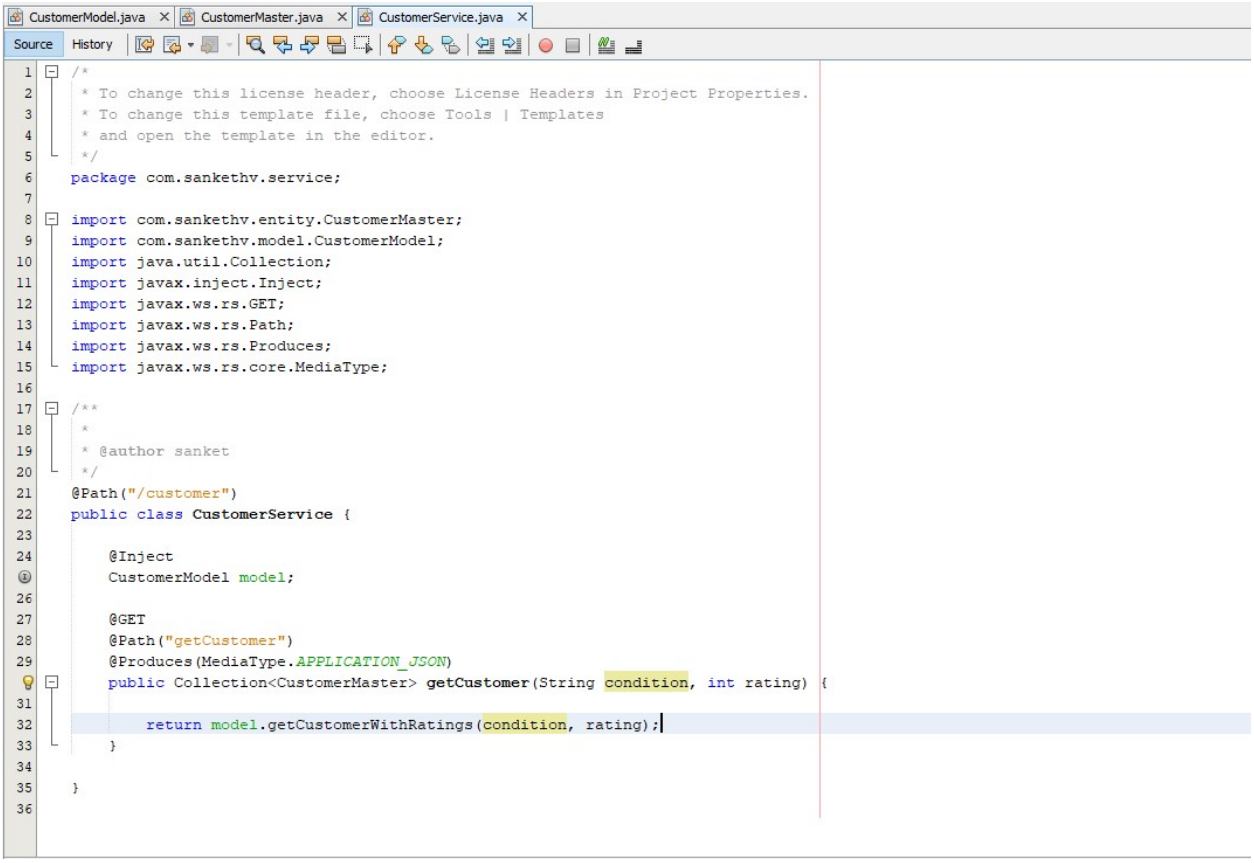
```
    @NotNull
    @Size(min = 1, max = 20)
    @Column(name = "phone")
    private String phone;

    public CustomerMaster() {
    }

    public CustomerMaster(Integer customerId) {
        this.customerId = customerId;
    }

    public CustomerMaster(Integer customerId, String email, String name, int rating, String address,
String phone) {
        this.customerId = customerId;
        this.email = email;
        this.name = name;
        this.rating = rating;
        this.address = address;
        this.phone = phone;
    }

    public Integer getCustomerId() {
        return customerId;
    }
```

```
public void setCustomerId(Integer customerId) {
   this.customerId = customerId;
}

public String getEmail() {
   return email;
}

public void setEmail(String email) {
   this.email = email;
}

public String getName() {
   return name;
}

public void setName(String name) {
   this.name = name;
}

public int getRating() {
   return rating;
}

public void setRating(int rating) {
```

```
    this.rating = rating;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getPhone() {
    return phone;
}

public void setPhone(String phone) {
    this.phone = phone;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (customerId != null ? customerId.hashCode() : 0);
    return hash;
}
```

```
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof CustomerMaster)) {
            return false;
        }
        CustomerMaster other = (CustomerMaster) object;
        if ((this.customerId == null && other.customerId != null) || (this.customerId != null &&
!this.customerId.equals(other.customerId))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "com.sankethv.entity.CustomerMaster[ customerId=" + customerId + " ]";
    }

}
```

7. Now create the service for testing.



package com.sankethv.service;

import com.sankethv.entity.CustomerMaster;

import com.sankethv.model.CustomerModel;

import java.util.Collection;

import javax.inject.Inject;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

```
/**
 *
 * @author sanket
 */
@Path("/customer")
public class CustomerService {

    @Inject
    CustomerModel model;

    @POST
    @Path("getCustomer")
    @Produces(MediaType.APPLICATION_JSON)
    @Consumes(MediaType.APPLICATION_JSON)
    public Collection<CustomerMaster> getCustomer(String condition, int rating) {

        return model.getCustomerWithRatings(condition, rating);
    }

}
```