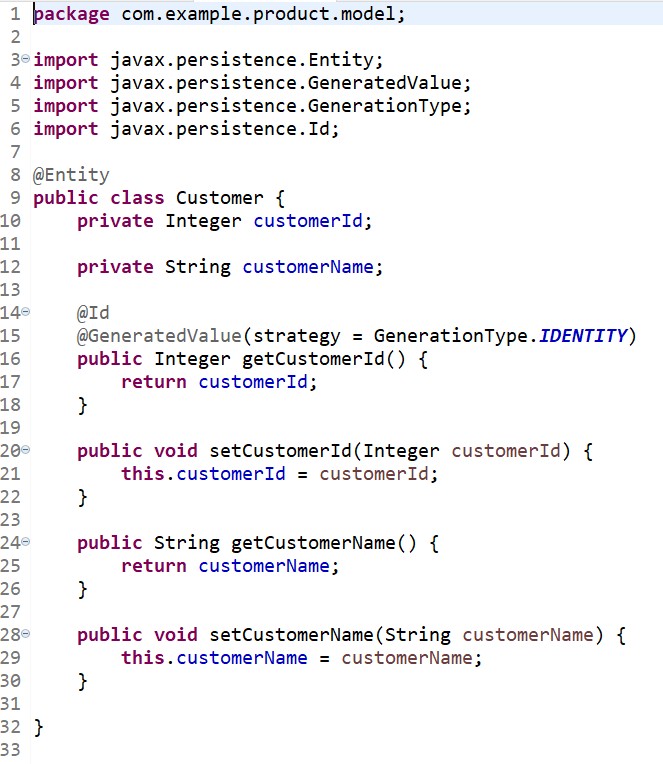
**Anisha priyadarshanee sahoo**

**1941012085**

## Model: Customer.java



Manufacturer.java



Product.java

package com.example.product.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue; import javax.persistence.GenerationType; import javax.persistence.Id;

@Entity

public class Product {

private Integer productId;

private String productName; private float productPrice;

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY) public Integer getProductId() {

return productId;

}

public String getProductName() { return productName;

}

public float getProductPrice() { return productPrice;

}

public void setProductId(Integer productId) { this.productId = productId;

}

public void setProductName(String productName) { this.productName = productName;

}

public void setProductPrice(float productPrice) { this.productPrice = productPrice;

}

}

# Service:

## CustomerRepository.java

package com.example.product.service;

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

import com.example.product.model.Customer;

@Repository

public interface CustomerRepository extends JpaRepository<Customer, Integer>{}

ProductRepository.java

package com.example.product.service;

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

import com.example.product.model.Product;

@Repository

public interface ProductRepository extends JpaRepository<Product, Integer>{}

# Dao:

## ProductController.java

package com.example.product.dao;

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

import org.springframework.beans.factory.annotation.Autowired; import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping; import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.PostMapping; import org.springframework.web.bind.annotation.PutMapping; import org.springframework.web.bind.annotation.RequestBody; import org.springframework.web.bind.annotation.RequestParam; import org.springframework.web.bind.annotation.ResponseBody; import org.springframework.web.bind.annotation.RestController;

import com.example.product.exception.ResourceNotFoundException; import com.example.product.model.Product;

import com.example.product.service.ProductRepository;

@RestController

public class ProductController { @Autowired

private ProductRepository productRepository;

// RESTful API methods for Retrieval operations @GetMapping("/products")

public List<Product> list() {

return productRepository.findAll();

}

// RESTful API methods for Retrieval of a specific product @GetMapping("/product")

public Product getProduct(@RequestParam int productId) {

Optional<Product> productOptional = productRepository.findById(productId);

return productOptional.get();

}

// RESTful API method for Create operation @PostMapping("/saveproduct")

public @ResponseBody String saveProduct(@RequestBody Product product)

{

productRepository.save(product); return "New product saved";

}

// RESTful API method for Update operation @PutMapping("/updateproduct")

public ResponseEntity<Product> updateProduct(@RequestBody Product product) {

Product updateProduct = productRepository.findById(product.getProductId()).orElseThrow(

()-> new ResourceNotFoundException("Product not exist

with id: "));

updateProduct.setProductName(product.getProductName()); updateProduct.setProductPrice(product.getProductPrice()); productRepository.save(updateProduct);

return ResponseEntity.ok(updateProduct);

}

// RESTful API method for Delete operation @DeleteMapping("/removeproduct")

public String deleteUser(@RequestBody Product product)

{

Product deleteProduct = productRepository.findById(product.getProductId()).orElseThrow(

()-> new ResourceNotFoundException("Product not exist

with id: "));

}

productRepository.deleteById(deleteProduct.getProductId()); return "Product deleted";

}

Exception: ResourceNotFoundException.java *package com.example.product.exception;*

import org.springframework.http.HttpStatus;

import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(value= HttpStatus.NOT\_FOUND)

public class ResourceNotFoundException extends RuntimeException{ private static final long serialVersionUID = 1L;

public ResourceNotFoundException(String message) { super(message);

}

}

# Pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="<http://maven.apache.org/POM/4.0.0>" xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xsi:schemaLocation="<http://maven.apache.org/POM/4.0.0> https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.6</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>product</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>product</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>