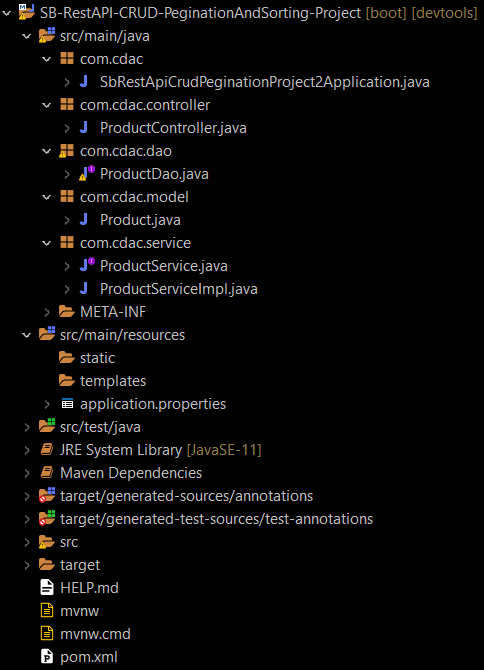
**Project Structure**



**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>2.7.10</version>

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

</parent>

<groupId>com.cdac</groupId>

<artifactId>SB-RestAPI-CRUD-Pegination-Project</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>SB-RestAPI-CRUD-Pegination-Project</name>

<description>Spring Boot Project</description>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

<properties>

<java.version>11</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.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

</dependencies>

</project>

**Product.java**

package com.cdac.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "product\_table3")

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private float price;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public float getPrice() {

return price;

}

public void setPrice(float price) {

this.price = price;

}

}

**ProductDao.java**

package com.cdac.dao;

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

import org.springframework.stereotype.Repository;

import com.cdac.model.Product;

@Repository

public interface ProductDao extends JpaRepository<Product, Integer> {

}

**ProductService.java**

package com.cdac.service;

import java.util.List;

import com.cdac.model.Product;

public interface ProductService {

public void add(Product product);

public void modify(Product product);

public void removeById(int id);

public Product getById(int id);

public List<Product> getAll();

public List<Product> getAlPaginated(int pageNo, int sizeSize, String field);

}

**ProductServiceImpl.java**

package com.cdac.service;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

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

import org.springframework.data.domain.PageRequest;

import org.springframework.data.domain.Pageable;

import org.springframework.data.domain.Sort.Direction;

import org.springframework.stereotype.Service;

import com.cdac.dao.ProductDao;

import com.cdac.model.Product;

@Service

public class ProductServiceImpl implements ProductService {

@Autowired

private ProductDao productDao;

@Override

public void add(Product product) {

productDao.save(product);

}

@Override

public void modify(Product product) {

productDao.save(product);

}

@Override

public void removeById(int id) {

productDao.deleteById(id);

}

@Override

public Product getById(int id) {

Optional<Product> opt = productDao.findById(id);

if (opt.isPresent()) {

return opt.get();

}

return null;

}

@Override

public List<Product> getAll() {

Iterable<Product> itr = productDao.findAll();

List<Product> lst = new ArrayList<Product>();

itr.forEach(ele -> lst.add(ele));

return lst;

}

@Override

public List<Product> getAlPaginated(int pageNo, int sizeSize, String field) {

Pageable p = PageRequest.of(pageNo, sizeSize, Direction.ASC, field);

return productDao.findAll(p).toList();

}

}

**ProductController.java**

package com.cdac.controller;

import java.util.List;

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

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

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

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

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.RestController;

import com.cdac.model.Product;

import com.cdac.service.ProductService;

@RestController

public class ProductController {

@Autowired

private ProductService productService;

@PostMapping(value = { "/product" })

public String productAdd(@RequestBody Product product) {

productService.add(product);

return "Product added successfully";

}

@PutMapping(value = { "/product" })

public String productUpdate(@RequestBody Product product) {

productService.modify(product);

return "Product updated successfully";

}

@DeleteMapping(value = { "/product" })

public String productDelete(@RequestParam int id) {

System.out.println("first delete");

productService.removeById(id);

return "Product deleted successfully";

}

@DeleteMapping(value = { "/product/{id}" })

public String productDelete1(@PathVariable int id) {

System.out.println("second delete");

productService.removeById(id);

return "Product deleted successfully";

}

@GetMapping(value = { "/product/{id}" })

public Product productGet(@PathVariable int id) {

return productService.getById(id);

}

@GetMapping(value = { "/product" })

public List<Product> productList() {

return productService.getAll();

}

@GetMapping("/products")

public List<Product> getAllPaginated (@RequestParam int pageNo, @RequestParam int pageSize, @RequestParam String field) {

return productService.getAlPaginated(pageNo, pageSize, field);

}

}

**SbRestApiCrudPeginationProject2Application**

package com.cdac;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SbRestApiCrudPeginationProject2Application {

public static void main(String[] args) {

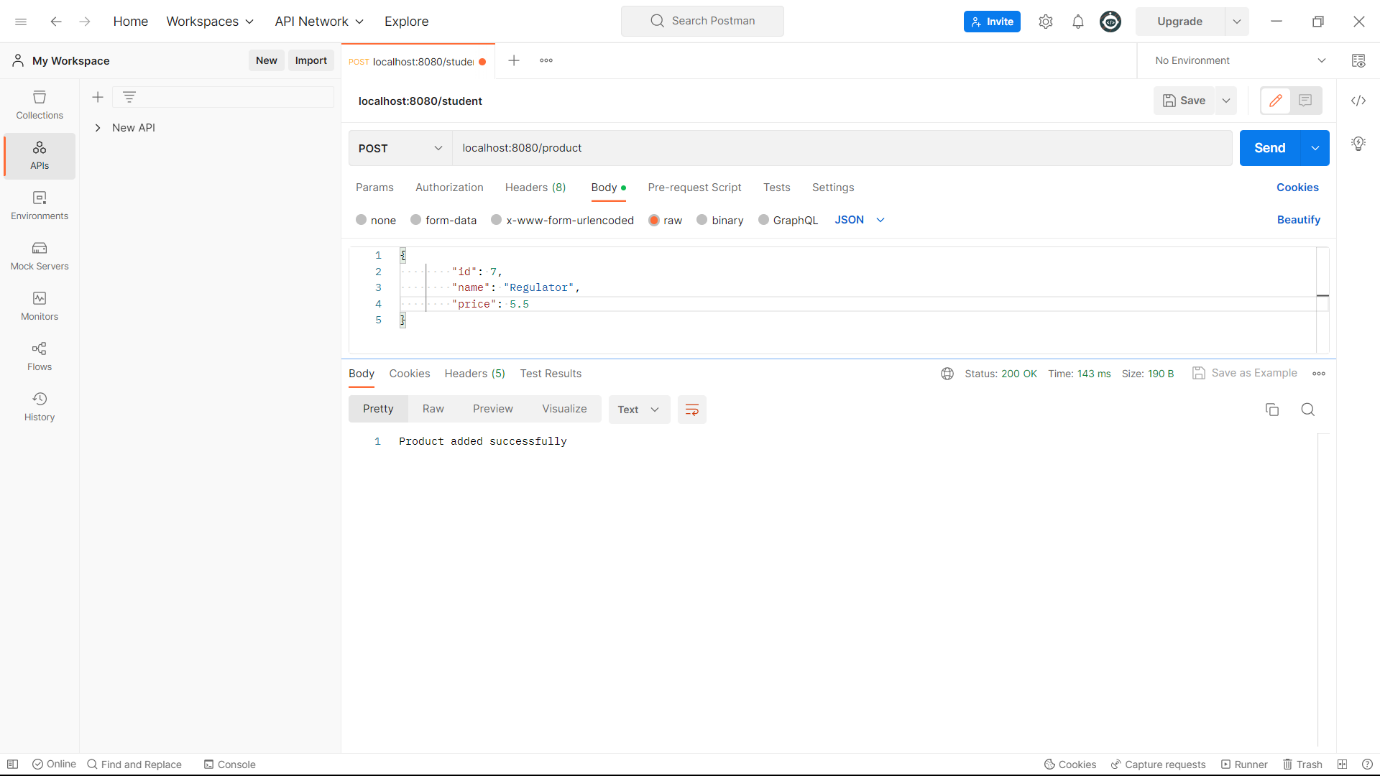
SpringApplication.run(SbRestApiCrudPeginationProject2Application.class, args);

}

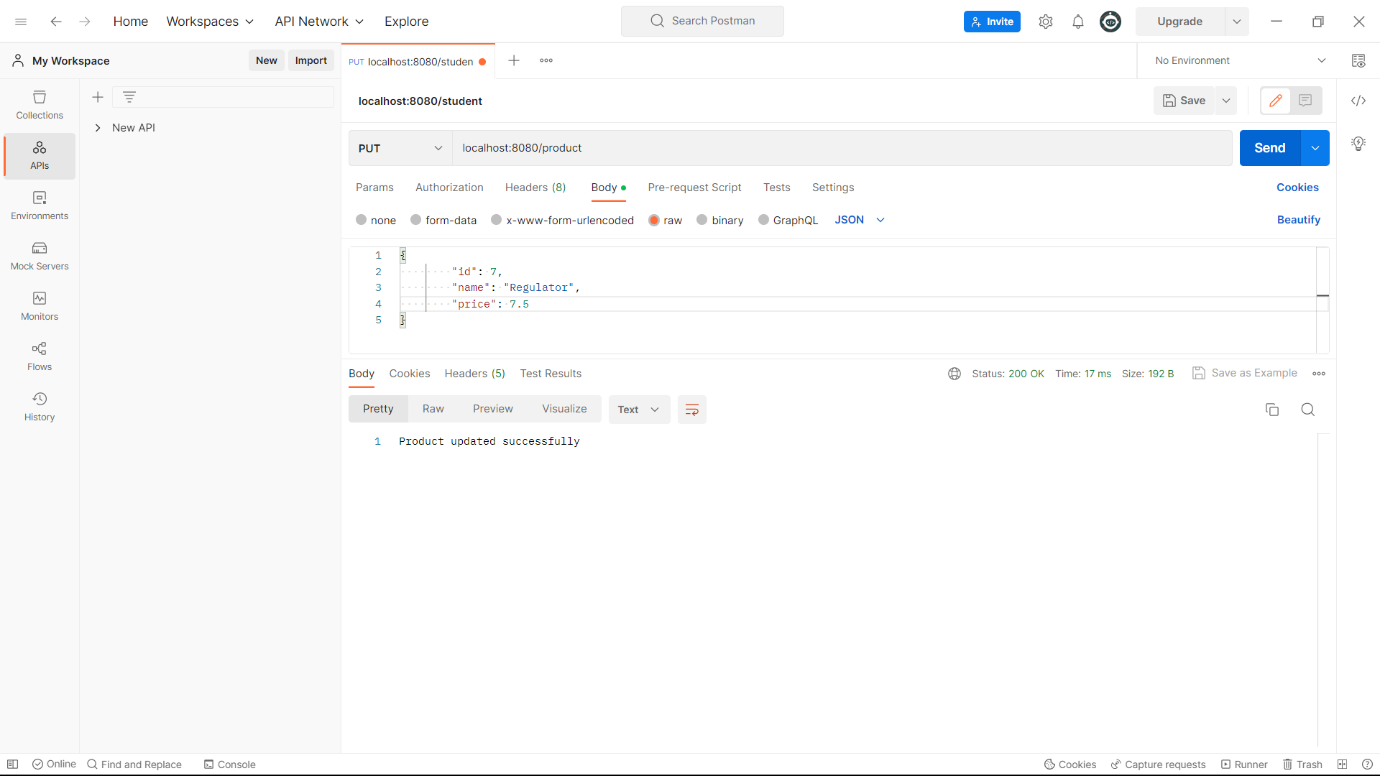
}

**Postman and Database Screenshots**

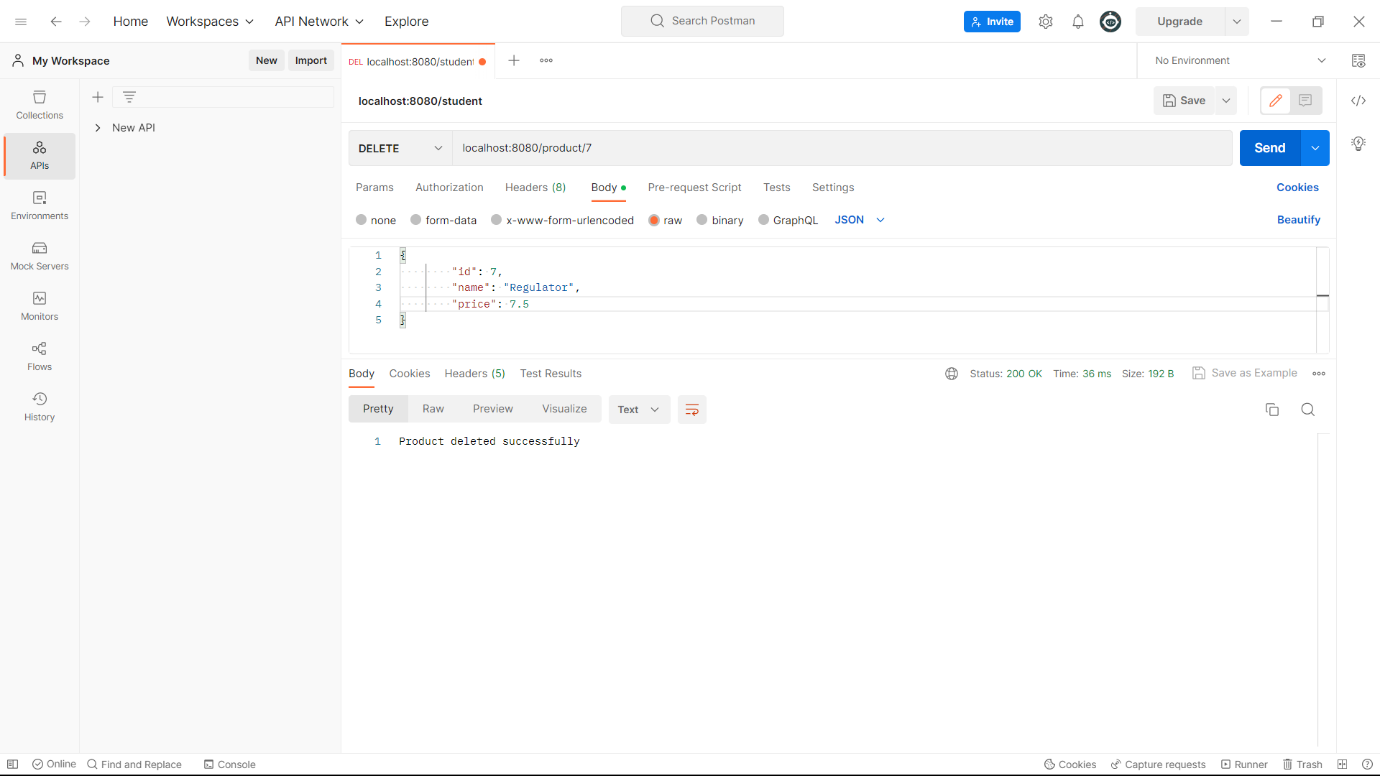
Insert



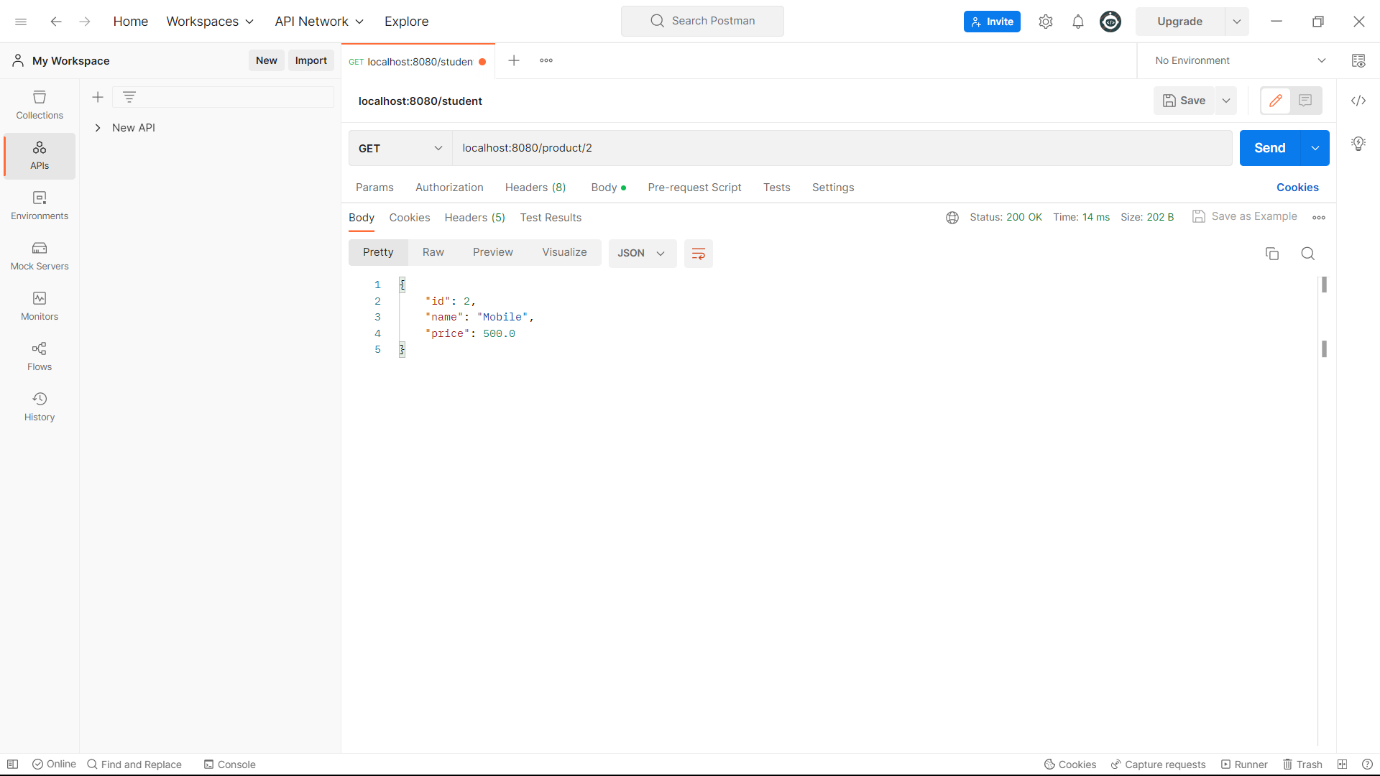
update

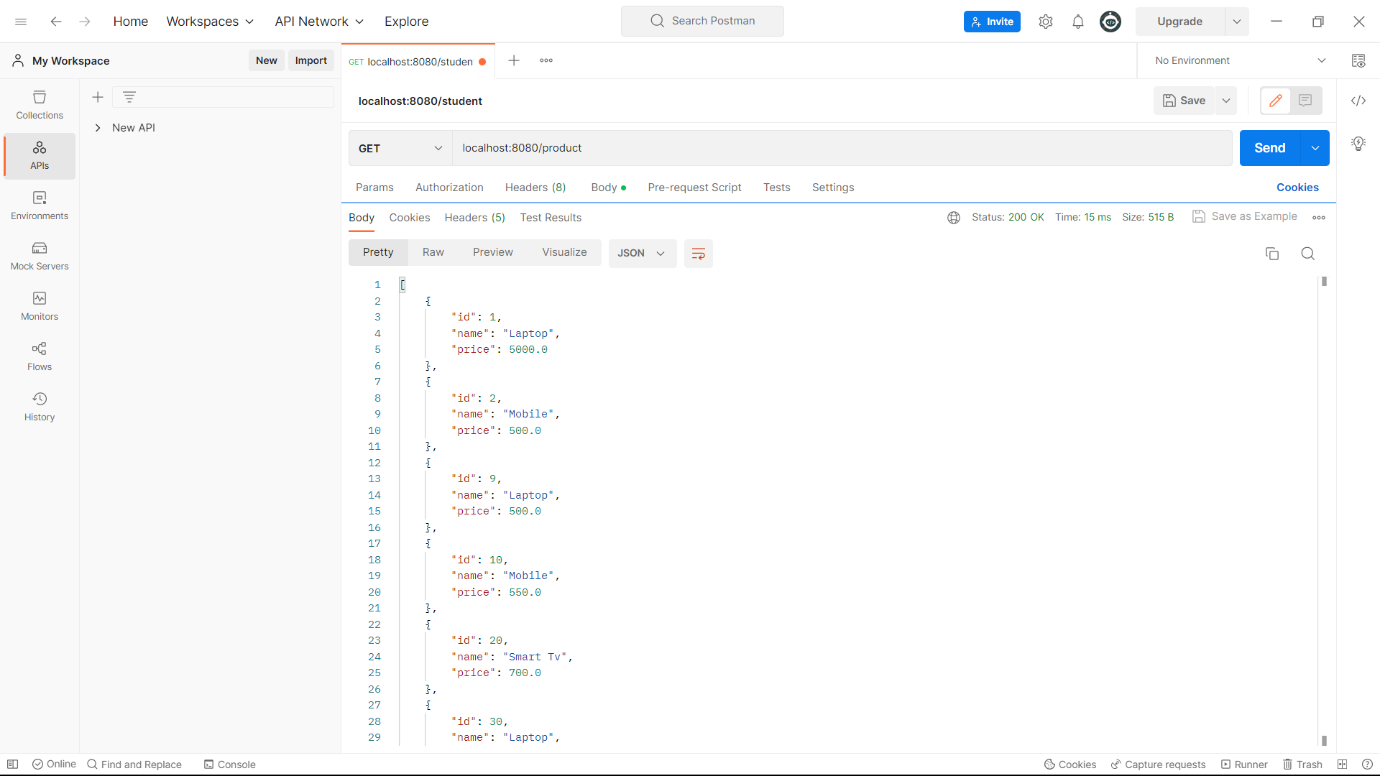


delete

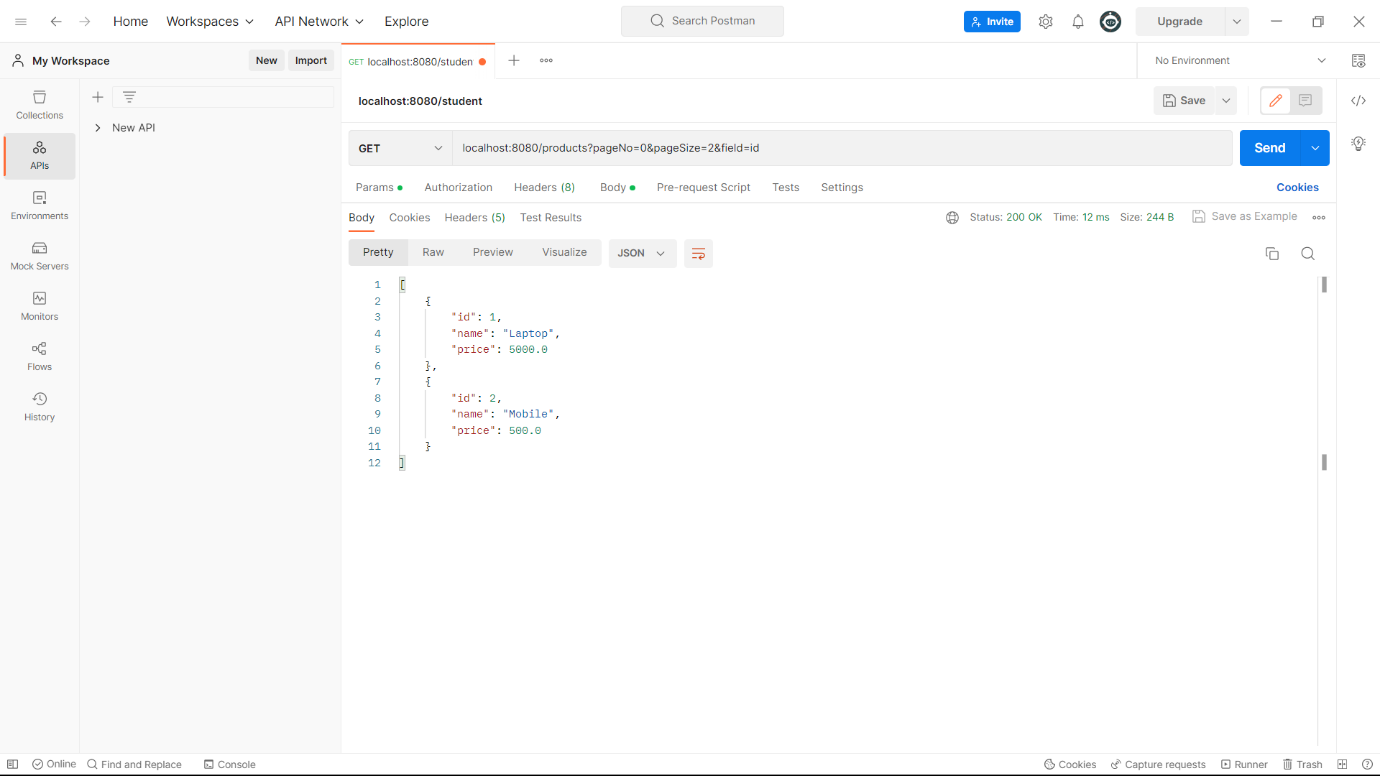


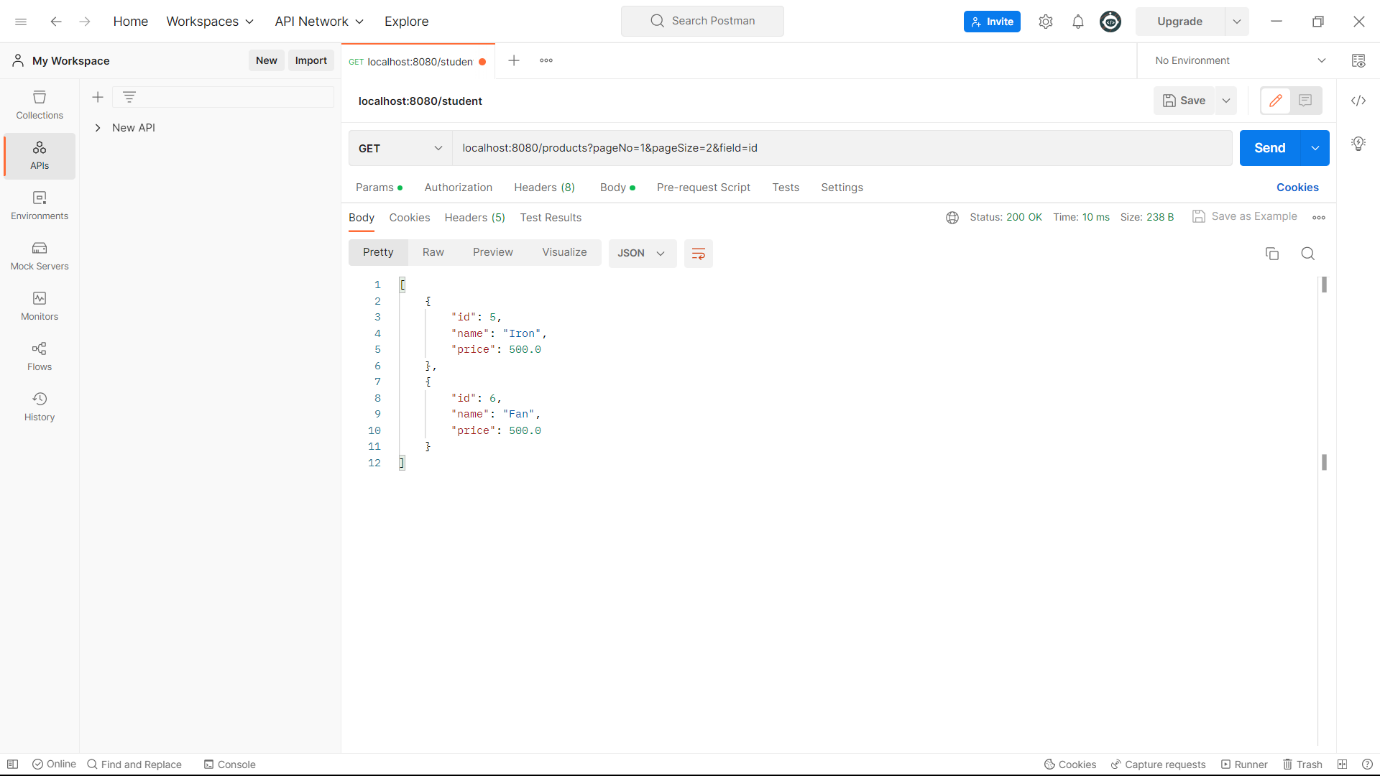
get



getall 

Get all paginated and sorted





Product table Database

