Create a REST API to perform CRUD Operations

Project Name: Pet Store Application

Entity Name	Attributes
Product	private Integer id
	private String name
	private String manufacture
	private Double price
	private String description

Create the appropriate Controller and Service classes for the Product entity.

Test the API using Postman or Hopscotch.

Hint:

Use a Map<Integer, Product> to store and manage Product instances.

API	**Details**
GET /products/{id}	Retrieves the details of the specific Product.
GET /products	Retrieves the details of all Products.
POST /products	Adds a new product.
DELETE /product/{id}	Deletes a specific product.
PUT /product/{id}	Updates a specific product.

Note: Upload the assignment in PDF format containing screenshots of code and output screenshots.

Code

Product.java

```java
package com.restapi.PetStoreApplication.model;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    private String name;
    private String manufacture;
    private Double price;
    private String description;

    public Product() {}
    public Product(String name, String manufacture, Double price, String
description) {
        this.name = name;
        this.manufacture = manufacture;
        this.price = price;
        this.description = description;
    }
    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getManufacture() {
        return manufacture;
    }
    public void setManufacture(String manufacture) {
        this.manufacture = manufacture;
    }
    public Double getPrice() {
        return price;
    }
    public void setPrice(Double price) {
        this.price = price;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

ProductController.java

```java
package com.restapi.PetStoreApplication.controller;

import com.restapi.PetStoreApplication.model.Product;
import com.restapi.PetStoreApplication.service.ProductService;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
@RequestMapping("/products")
public class ProductController {

    private final ProductService productService;

    public ProductController(ProductService productService) {
        this.productService = productService;
    }

    // GET: Retrieve all products
    @GetMapping
    public ResponseEntity<List<Product>> getAllProducts() {
        return ResponseEntity.ok(productService.getAllProducts());
    }

    // GET: Retrieve product by ID
    @GetMapping("/{id}")
    public ResponseEntity<Product> getProductById(@PathVariable Integer id)
{
        Product product = productService.getProductById(id);
        if (product == null) {
            return ResponseEntity.notFound().build();
        }
        return ResponseEntity.ok(product);
    }

    // POST: Add a new product
    @PostMapping("/add")
    public ResponseEntity<Product> addProduct(@RequestBody Product product)
{
        return ResponseEntity.ok(productService.addProduct(product));
    }

    // PUT: Update product by ID
    @PutMapping("/{id}")
    public ResponseEntity<Product> updateProduct(@PathVariable Integer id,
@RequestBody Product product) {
        Product updatedProduct = productService.updateProduct(id, product);
        if (updatedProduct == null) {
            return ResponseEntity.notFound().build();
        }
        return ResponseEntity.ok(updatedProduct);
    }

    // DELETE: Remove product by ID
    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteProduct(@PathVariable Integer id) {
        if (productService.deleteProduct(id)) {
            return ResponseEntity.noContent().build();
        }
        return ResponseEntity.notFound().build();
    }
}
```

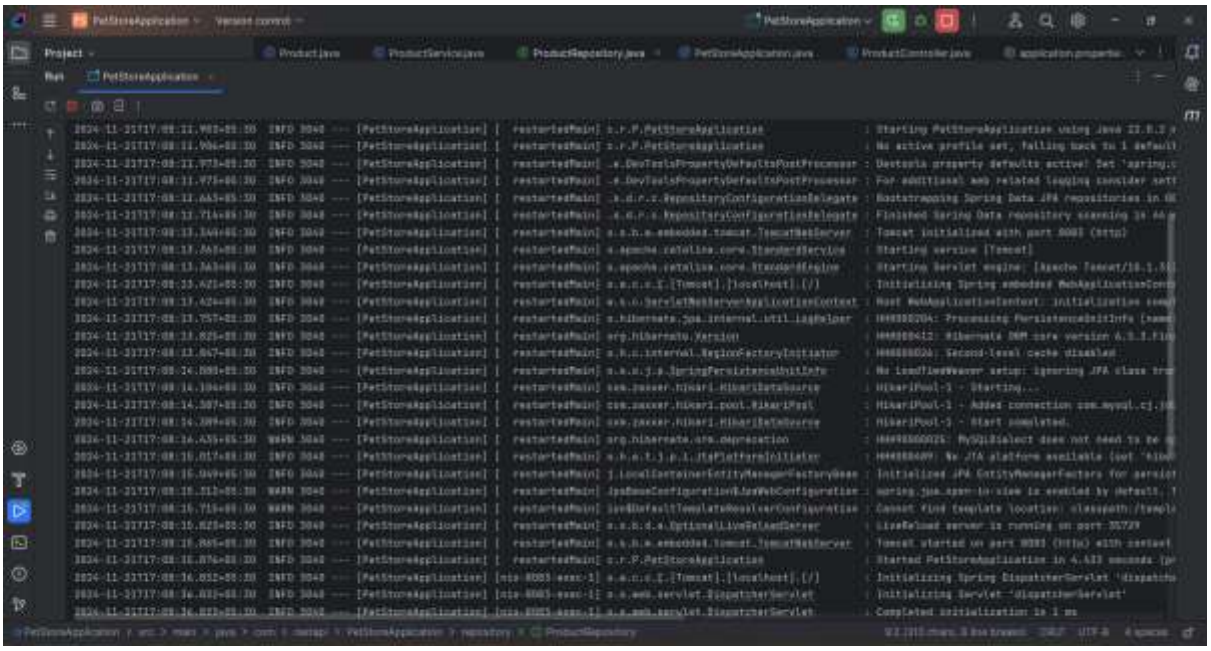
ProductService.java

```java
package com.restapi.PetStoreApplication.service;

import com.restapi.PetStoreApplication.model.Product;
import com.restapi.PetStoreApplication.repository.ProductRepository;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class ProductService {

    private final ProductRepository productRepository;

    public ProductService(ProductRepository productRepository) {
        this.productRepository = productRepository;
    }

    // Retrieve all products
    public List<Product> getAllProducts() {
        return productRepository.findAll();
    }

    // Retrieve product by ID
    public Product getProductById(Integer id) {
        return productRepository.findById(id).orElse(null);
    }

    // Add a new product
    public Product addProduct(Product product) {
        return productRepository.save(product);
    }

    // Update an existing product
    public Product updateProduct(Integer id, Product updatedProduct) {
        if (productRepository.existsById(id)) {
            updatedProduct.setId(id);
            return productRepository.save(updatedProduct);
        }
        return null;
    }

    // Delete a product by ID
    public boolean deleteProduct(Integer id) {
        if (productRepository.existsById(id)) {
            productRepository.deleteById(id);
            return true;
        }
        return false;
    }
}
```

ProductRepository.java

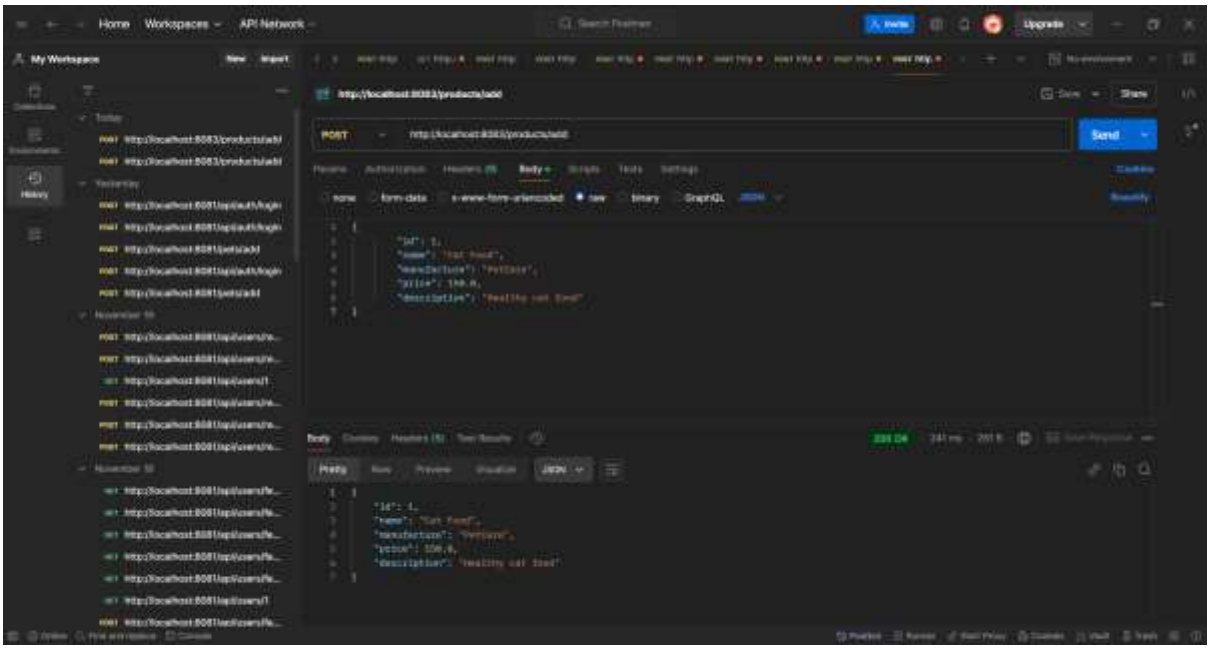
```java
package com.restapi.PetStoreApplication.repository;

import com.restapi.PetStoreApplication.model.Product;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

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

Output:-



```
2024-11-21T17:08:11.903+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] c.r.P.PetStoreApplication                : Starting PetStoreApplication using Java 22.0.2 w
2024-11-21T17:08:11.906+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] c.r.P.PetStoreApplication                : No active profile set, falling back to 1 default
2024-11-21T17:08:11.973+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.c
2024-11-21T17:08:11.975+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider sett
2024-11-21T17:08:12.663+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DE
2024-11-21T17:08:12.714+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 44 m
2024-11-21T17:08:13.344+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8083 (http)
2024-11-21T17:08:13.363+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-11-21T17:08:13.363+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.31]
2024-11-21T17:08:13.421+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationConte
2024-11-21T17:08:13.424+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization compl
2024-11-21T17:08:13.757+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name
2024-11-21T17:08:13.825+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 6.5.3.Fin
2024-11-21T17:08:13.867+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.h.c.internal.RegionFactoryInitiator    : HHH000026: Second-level cache disabled
2024-11-21T17:08:14.089+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver setup: ignoring JPA class tra
2024-11-21T17:08:14.104+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2024-11-21T17:08:14.307+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added connection com.mysql.cj.jd
2024-11-21T17:08:14.309+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2024-11-21T17:08:14.435+05:30  WARN 3048 --- [PetStoreApplication] [  restartedMain] org.hibernate.orm.deprecation            : HHH90000025: MySQLDialect does not need to be s
2024-11-21T17:08:15.017+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000489: No JTA platform available (set 'hibe
2024-11-21T17:08:15.049+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persist
2024-11-21T17:08:15.312+05:30  WARN 3048 --- [PetStoreApplication] [  restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. T
2024-11-21T17:08:15.715+05:30  WARN 3048 --- [PetStoreApplication] [  restartedMain] ion$DefaultTemplateResolverConfiguration : Cannot find template location: classpath:/templa
2024-11-21T17:08:15.825+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2024-11-21T17:08:15.865+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8083 (http) with context
2024-11-21T17:08:15.876+05:30  INFO 3048 --- [PetStoreApplication] [  restartedMain] c.r.P.PetStoreApplication                : Started PetStoreApplication in 4.433 seconds (pr
2024-11-21T17:08:16.832+05:30  INFO 3048 --- [PetStoreApplication] [nio-8083-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatche
2024-11-21T17:08:16.832+05:30  INFO 3048 --- [PetStoreApplication] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2024-11-21T17:08:16.833+05:30  INFO 3048 --- [PetStoreApplication] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 1 ms
```

Post Method(In Postman)



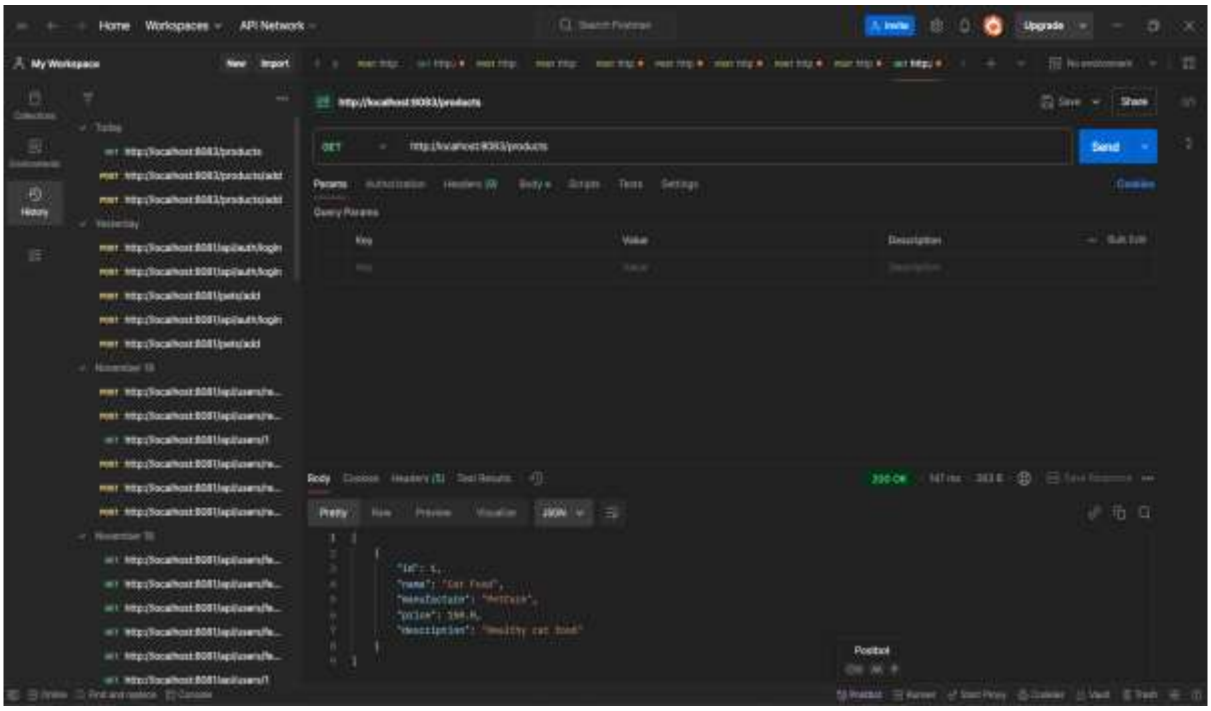
POST http://localhost:8083/products/add

```
{
    "id": 1,
    "name": "Cat Food",
    "manufacture": "Petcare",
    "price": 150.0,
    "description": "Healthy cat food"
}
```

200 OK 241 ms 201 B

```
{
    "id": 1,
    "name": "Cat Food",
    "manufacture": "Petcare",
    "price": 150.0,
    "description": "Healthy cat food"
}
```

Get Method(In Postman)

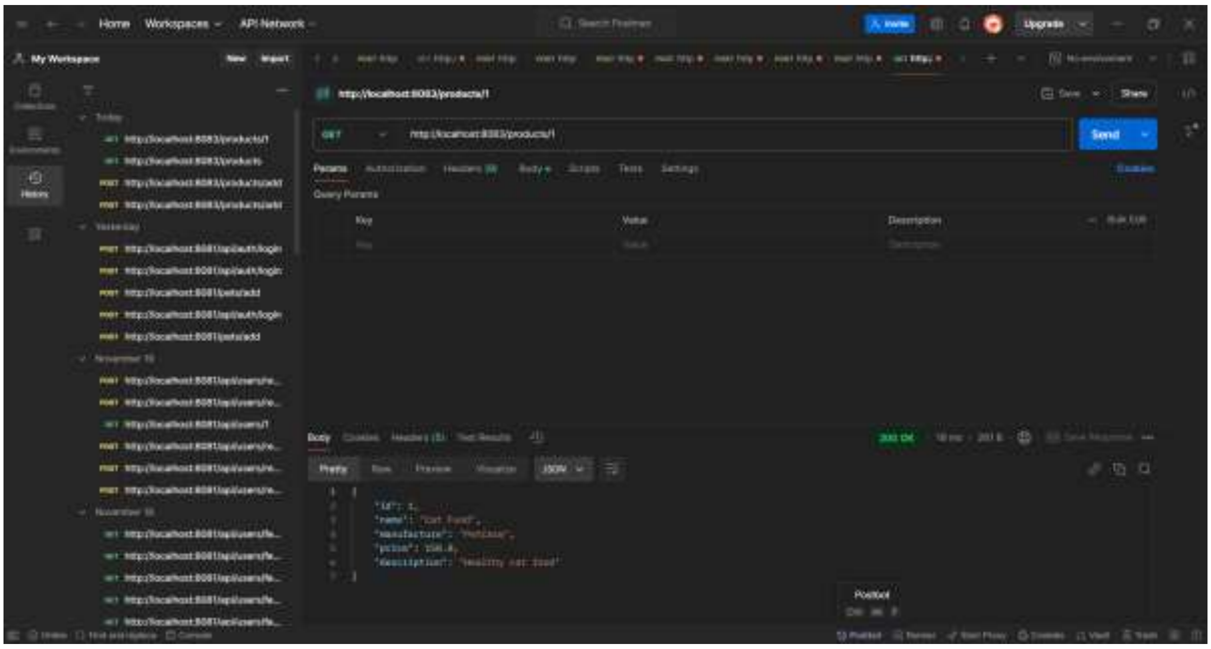


```
GET http://localhost:8083/products
```

200 OK

```
[
    {
        "id": 1,
        "name": "Cat Food",
        "manufacturer": "PetCare",
        "price": 150.0,
        "description": "Healthy cat food"
    }
]
```

Get Method(In Postman)

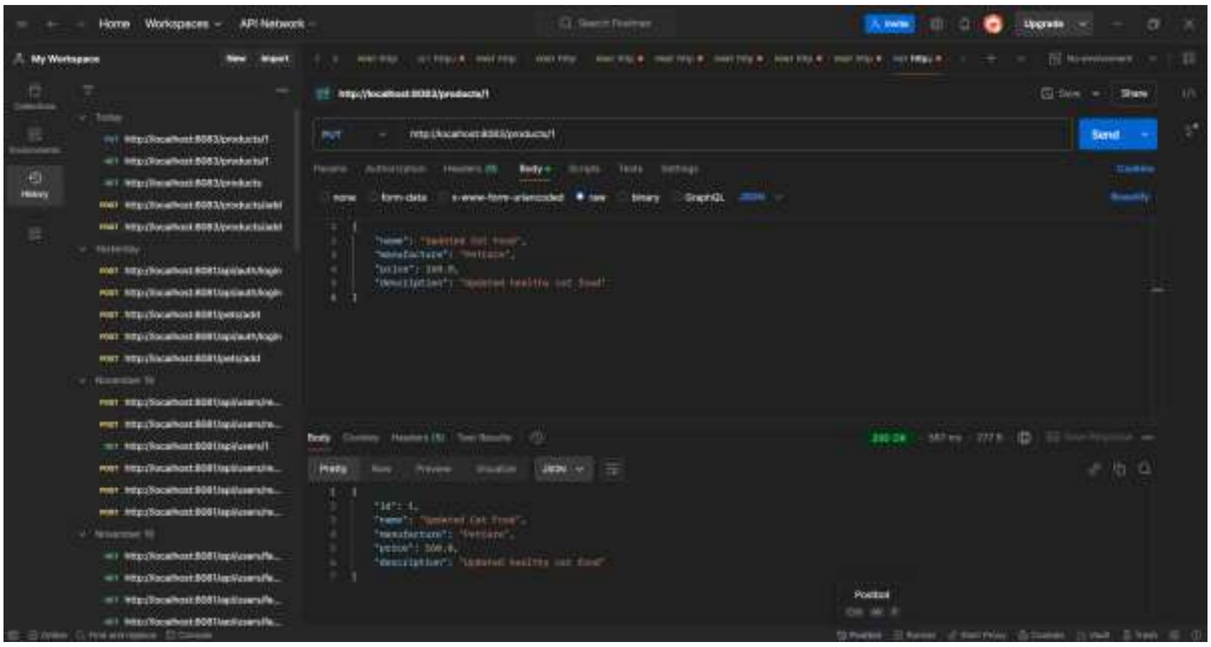


```
GET http://localhost:8083/products/1
```

200 OK

```
{
    "id": 1,
    "name": "Cat Food",
    "manufacturer": "PetCare",
    "price": 150.0,
    "description": "Healthy cat food"
}
```

Put Method(In Postman)



http://localhost:8083/products/1

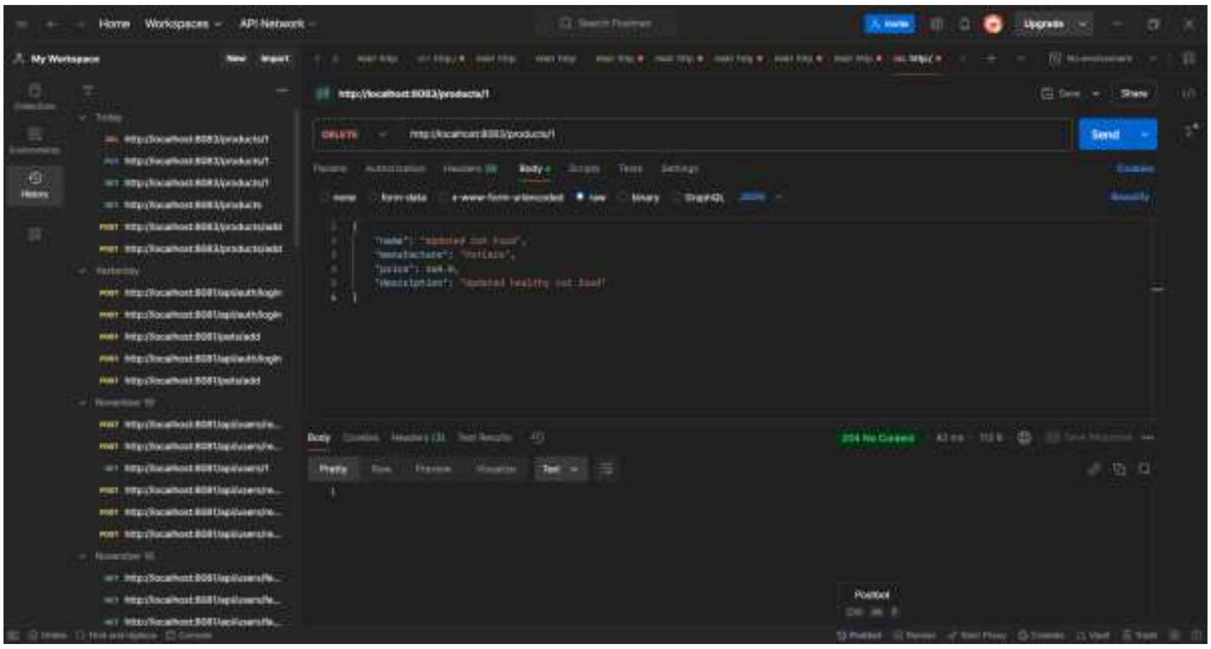
PUT http://localhost:8083/products/1

```
{
    "name": "Updated Cat Food",
    "manufacture": "Petcare",
    "price": 160.0,
    "description": "Updated healthy cat food"
}
```

200 OK · 587 ms · 277 B

```
{
    "id": 1,
    "name": "Updated Cat Food",
    "manufacturer": "Petcare",
    "price": 160.0,
    "description": "Updated healthy cat food"
}
```

Delete Method(In Postman)



http://localhost:8083/products/1

DELETE http://localhost:8083/products/1

```
{
    "name": "Updated Cat Food",
    "manufacture": "Petcare",
    "price": 160.0,
    "description": "Updated healthy cat food"
}
```

204 No Content · 42 ms · 112 B