

Product.java

```
package com.example.product_api.entity;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "products")
public class Product {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String productName;
    private String category;
    private Double price;
    private int quantity;
}
```

ProductDTO.java

```
package com.example.product_api.dto;

import jakarta.validation.constraints.Min;
import jakarta.validation.constraints.NotBlank;
```

```
import lombok.Data;

@Data
public class ProductDTO {

    @NotBlank(message = "Product name cannot be empty")
    private String productName;

    @NotBlank(message = "Category cannot be empty")
    private String category;

    @Min(value = 1, message = "Price must be positive")
    private Double price;

    @Min(value = 1, message = "Quantity must be positive")
    private int quantity;
}
```

ProductRepository.java

```
package com.example.product_api.repository;

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

import com.example.product_api.entity.Product;
import java.util.List;
import java.util.Optional;

@Repository
public interface ProductRepository extends JpaRepository<Product,
Long> {

    List<Product> findByCategoryIgnoreCase(String category);
    Optional<Product> findByProductNameIgnoreCase(String productName);
}
```

```
}
```

ProductService.java

```
package com.example.product_api.service;

import java.util.List;

import com.example.product_api.dto.ProductDTO;

public interface ProductService {

    List<ProductDTO> getAllProducts();
    List<ProductDTO> searchProductsByCategory(String category);
    ProductDTO addProduct(ProductDTO productDTO);
    void deleteProductByName(String name);

}
```

ProductServiceImpl.java

```
package com.example.product_api.service.impl;

import java.util.List;
import java.util.stream.Collectors;

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

import com.example.product_api.dto.ProductDTO;
import com.example.product_api.entity.Product;
import com.example.product_api.repository.ProductRepository;
import com.example.product_api.service.ProductService;

@Service
```

```
public class ProductServiceImpl implements ProductService {

    @Autowired
    private ProductRepository productRepository;

    private ProductDTO convertToDTO(Product product) {
        ProductDTO dto = new ProductDTO();
        dto.setProductName(product.getProductName());
        dto.setCategory(product.getCategory());
        dto.setPrice(product.getPrice());
        dto.setQuantity(product.getQuantity());
        return dto;
    }

    @Override
    public ProductDTO addProduct(ProductDTO productDTO) {

        Product product = new Product();
        product.setProductName(productDTO.getProductName());
        product.setCategory(productDTO.getCategory());
        product.setPrice(productDTO.getPrice());
        product.setQuantity(productDTO.getQuantity());

        Product savedProduct = productRepository.save(product);

        return convertToDTO(savedProduct);
    }

    @Override
    public List<ProductDTO> getAllProducts() {
        return productRepository.findAll()
            .stream()
            .map(this::convertToDTO)
            .collect(Collectors.toList());
    }
}
```

```
@Override
public List<ProductDTO> searchProductsByCategory(String category)
{
    return productRepository.findByCategoryIgnoreCase(category)
        .stream()
        .map(this::convertToDTO)
        .collect(Collectors.toList());
}

@Override
public void deleteProductByName(String name) {
    Product product =
productRepository.findByProductNameIgnoreCase(name)
        .orElseThrow(() -> new RuntimeException("Product not
found with name: " + name));
    productRepository.delete(product);
}
}
```

ProductController.java

```
package com.example.product_api.controller;

import com.example.product_api.dto.ProductDTO;
import com.example.product_api.service.ProductService;
import jakarta.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

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

    @Autowired
```

```
private ProductService productService;

@PostMapping
public ResponseEntity<ProductDTO> addProduct(@Valid @RequestBody
ProductDTO productDTO) {
    ProductDTO newProductDTO =
productService.addProduct(productDTO);
    return new ResponseEntity<>(newProductDTO,
HttpStatus.CREATED);
}

@GetMapping
public ResponseEntity<List<ProductDTO>> getAllProducts() {
    List<ProductDTO> products = productService.getAllProducts();
    return ResponseEntity.ok(products);
}

@GetMapping("/category/{category}")
public ResponseEntity<List<ProductDTO>>
getProductsByCategory(@PathVariable String category) {
    List<ProductDTO> products =
productService.searchProductsByCategory(category);
    return ResponseEntity.ok(products);
}

@DeleteMapping("/{name}")
public ResponseEntity<String> deleteProductByName(@PathVariable
String name) {
    try {
        productService.deleteProductByName(name);
        return ResponseEntity.ok("Product '" + name + "' deleted
successfully.");
    } catch (RuntimeException e) {
        return new ResponseEntity<>(e.getMessage(),
HttpStatus.NOT_FOUND);
    }
}
}
```

LAB 04

EC/2021/006

W.K.G.K JAYAWARDANA

The screenshot displays the API Network application interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The user is logged in as 'Gamika Jayawardhana's Worksp...'. The left sidebar shows a collection of APIs under 'ApeBodima Api', including 'POST Register', 'POST Login', 'GET Test JWT Filter', 'Employee Api', 'Movies Api', and 'Vehicle Api'. The 'Vehicle Api' is expanded, showing endpoints like 'POST Create a new vehicle', 'DEL New Request', 'GET Get vehicles by year', 'GET Get all Vehicles', and 'GET Get Vehicle Type by Service ID'. The main panel shows a POST request to 'http://localhost:8080/api/products'. The request body is a JSON object:

```
{  "productName": "Gaming Mouse",  "category": "Electronics",  "price": 75.50,  "quantity": 200}
```

. The response is a 201 Created status with a response time of 455 ms and a body size of 259 B. The response body is a JSON object:

```
{  "id": 1,  "productName": "Gaming Mouse",  "category": "Electronics",  "price": 75.5,  "quantity": 200}
```

. The bottom status bar shows 'Console', 'Runner', 'Vault', and other utility icons.

The screenshot displays the API Network application interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The main workspace is titled 'Gamika Jayawardhana's Worksp...'. The left sidebar shows a collection of API endpoints under 'ApeBodima Api', including 'POST Register', 'POST Login', 'GET Test JWT Filter', 'Employee Api', 'Movies Api', and 'Vehicle Api'. The 'Vehicle Api' section is expanded, showing endpoints like 'POST Create a new vehicle', 'DEL New Request', 'GET Get vehicles by year', 'GET Get all Vehicles', and 'GET Get Vehicle Type by Service ID'.

The main area shows a GET request to `http://localhost:8080/api/products`. The request body is a JSON array of product objects:

```
1 [
2   {
3     "productName": "Sapiens: A Brief History of Humankind",
4     "category": "Books",
5     "price": 3500.00,
6     "quantity": 250
7   },
8   {
9     "productName": "4K Ultra HD Monitor",
10    "category": "Electronics",
11    "price": 75000.0,
12    "quantity": 80
13  },
14  {
15    "productName": "The Alchemist",
16    "category": "Books",
17    "price": 2500.0,
18    "quantity": 300
19  },
20  {
21    "productName": "Espresso Coffee Machine",
22    "category": "Home & Kitchen",
23    "price": 45000.0,
24    "quantity": 50
25  },
26  {
27    "productName": "Sapiens: A Brief History of Humankind",
28    "category": "Books",
29    "price": 3500.00,
30    "quantity": 250
31  }
32 ]
```

The response is a 200 OK status with a response time of 359 ms and a body size of 544 B. The response body is a JSON array of product objects:

```
1 [
2   {
3     "productName": "4K Ultra HD Monitor",
4     "category": "Electronics",
5     "price": 75000.0,
6     "quantity": 80
7   },
8   {
9     "productName": "The Alchemist",
10    "category": "Books",
11    "price": 2500.0,
12    "quantity": 300
13  },
14  {
15    "productName": "Espresso Coffee Machine",
16    "category": "Home & Kitchen",
17    "price": 45000.0,
18    "quantity": 50
19  },
20  {
21    "productName": "Sapiens: A Brief History of Humankind",
22    "category": "Books",
23    "price": 3500.00,
24    "quantity": 250
25  }
26 ]
```

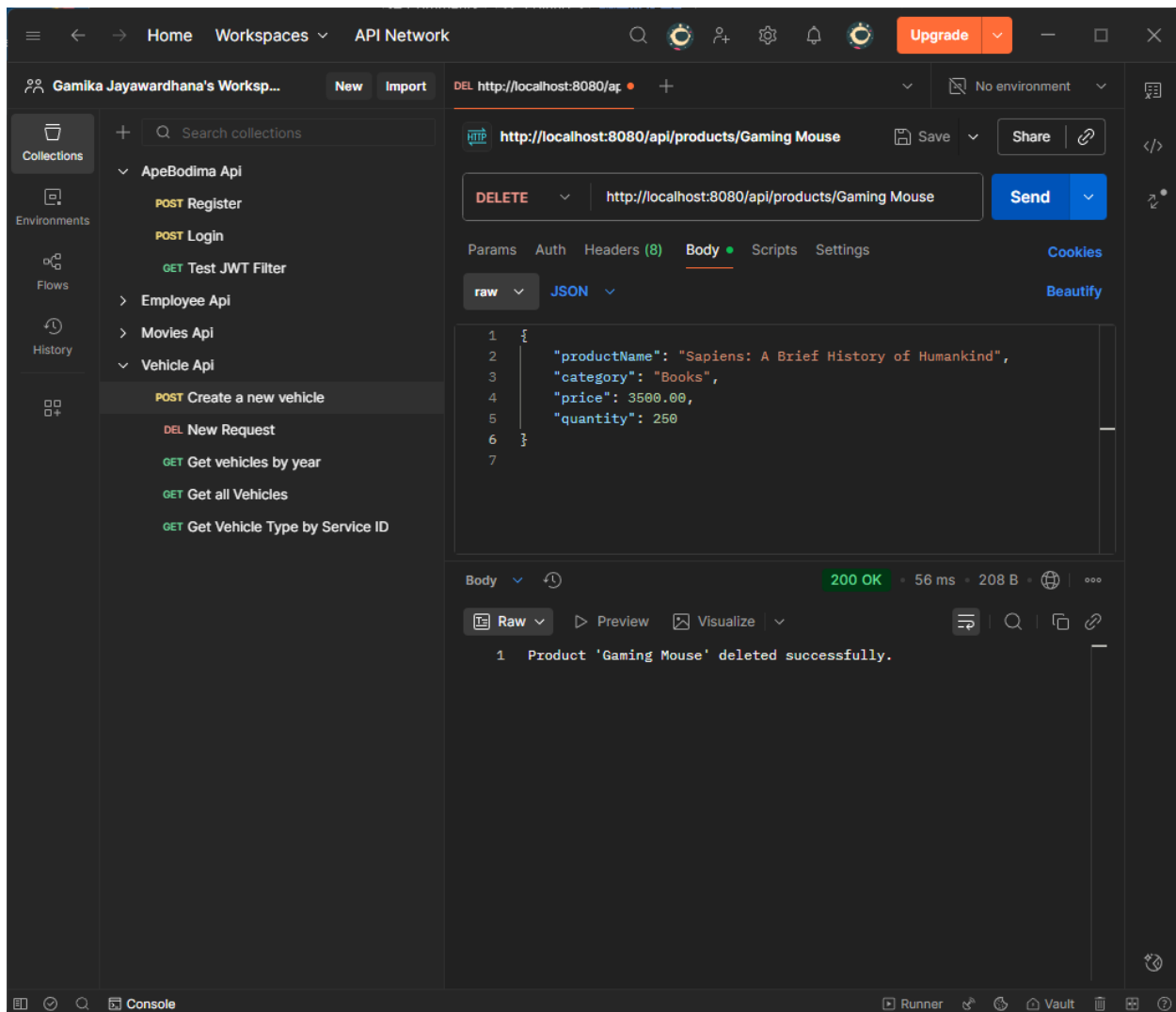

The screenshot displays the API Network application interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The user is logged in as 'Gamika Jayawardhana's Worksp...'. The main panel shows a GET request to 'http://localhost:8080/api/products/category/Elec...'. The request body is a JSON object:

```
1 {
2   "productName": "Sapiens: A Brief History of Humankind",
3   "category": "Books",
4   "price": 3500.00,
5   "quantity": 250
6 }
```

The response is a 200 OK status with a response time of 71 ms and a body size of 258 B. The response body is a JSON array:

```
1 [
2   {
3     "productName": "4K Ultra HD Monitor",
4     "category": "Electronics",
5     "price": 75000.0,
6     "quantity": 80
7   }
8 ]
```

The sidebar on the left shows a collection of API endpoints under 'ApeBodima Api', including 'POST Register', 'POST Login', 'GET Test JWT Filter', 'Employee Api', 'Movies Api', and 'Vehicle Api'. The bottom console shows the 'Console' tab.



The screenshot displays the API Network application interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The workspace is named 'Gamika Jayawardhana's Worksp...'. The left sidebar shows a tree view of collections, including 'ApeBodima Api' with endpoints like 'POST Register', 'POST Login', 'GET Test JWT Filter', 'Employee Api', 'Movies Api', and 'Vehicle Api'. The 'Vehicle Api' collection is expanded, showing 'POST Create a new vehicle', 'DEL New Request', 'GET Get vehicles by year', 'GET Get all Vehicles', and 'GET Get Vehicle Type by Service ID'.

The main panel shows a DELETE request to `http://localhost:8080/api/products/Gaming Mouse`. The request body is a JSON object:

```
1 {
2   "productName": "Sapiens: A Brief History of Humankind",
3   "category": "Books",
4   "price": 3500.00,
5   "quantity": 250
6 }
```

The response is a 200 OK status with a message: "Product 'Gaming Mouse' deleted successfully." The response body is shown in the 'Raw' tab.