Product.java

```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

```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
```

```java
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);
    }
}
```

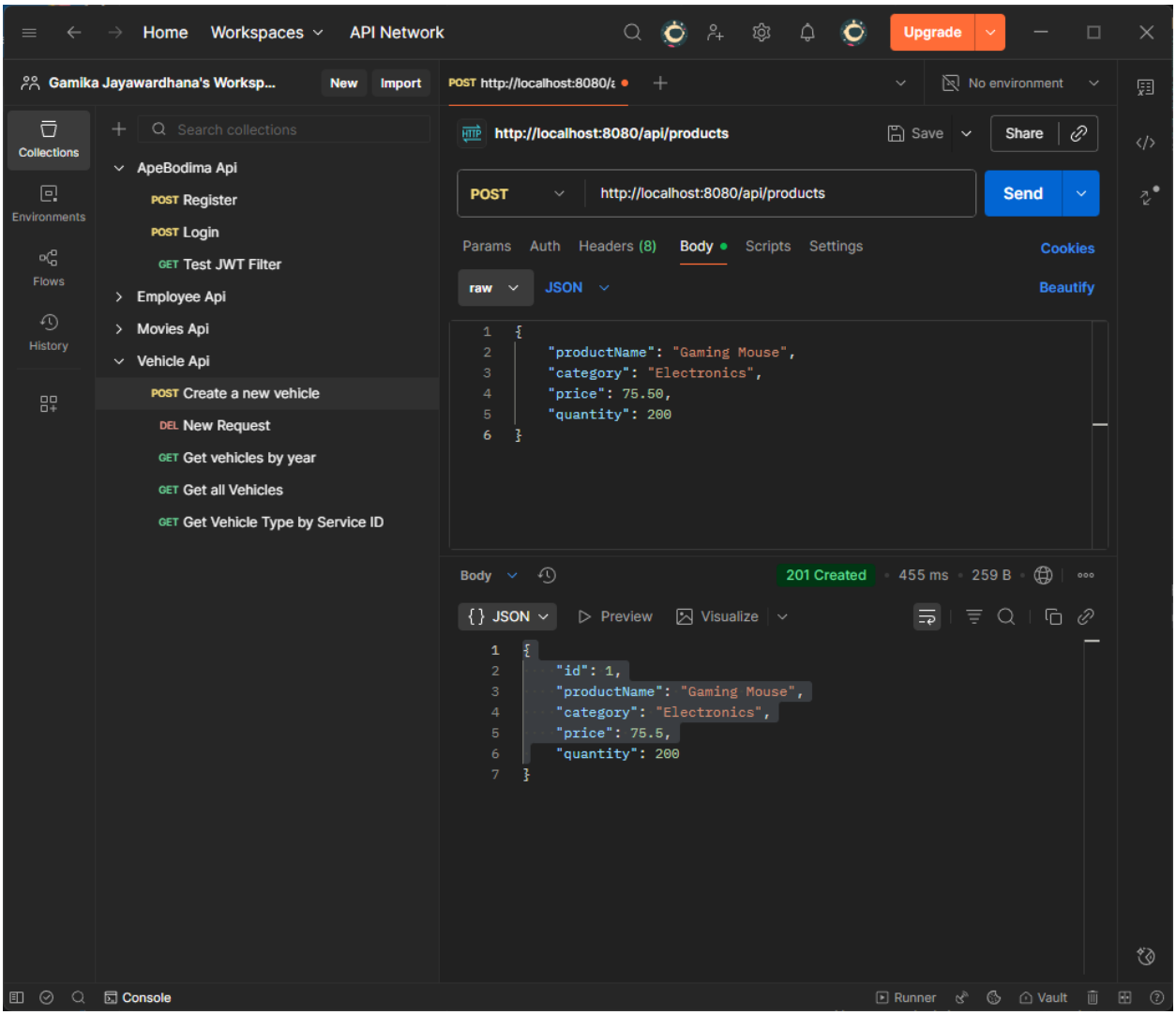
ProductController.java

```
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);
        }
    }
}
```

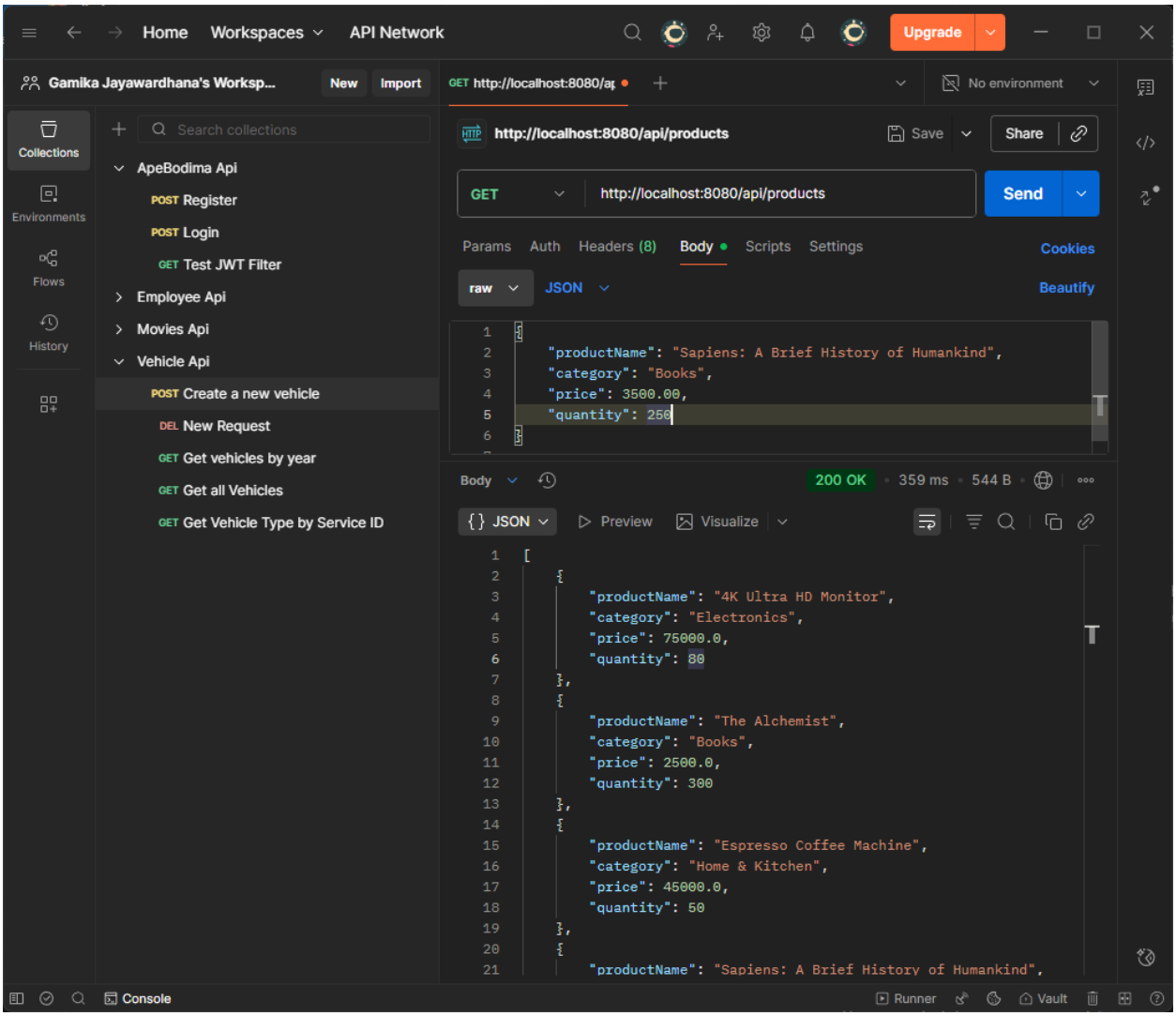
LAB 04

EC/2021/006 W.K.G.K JAYAWARDANA



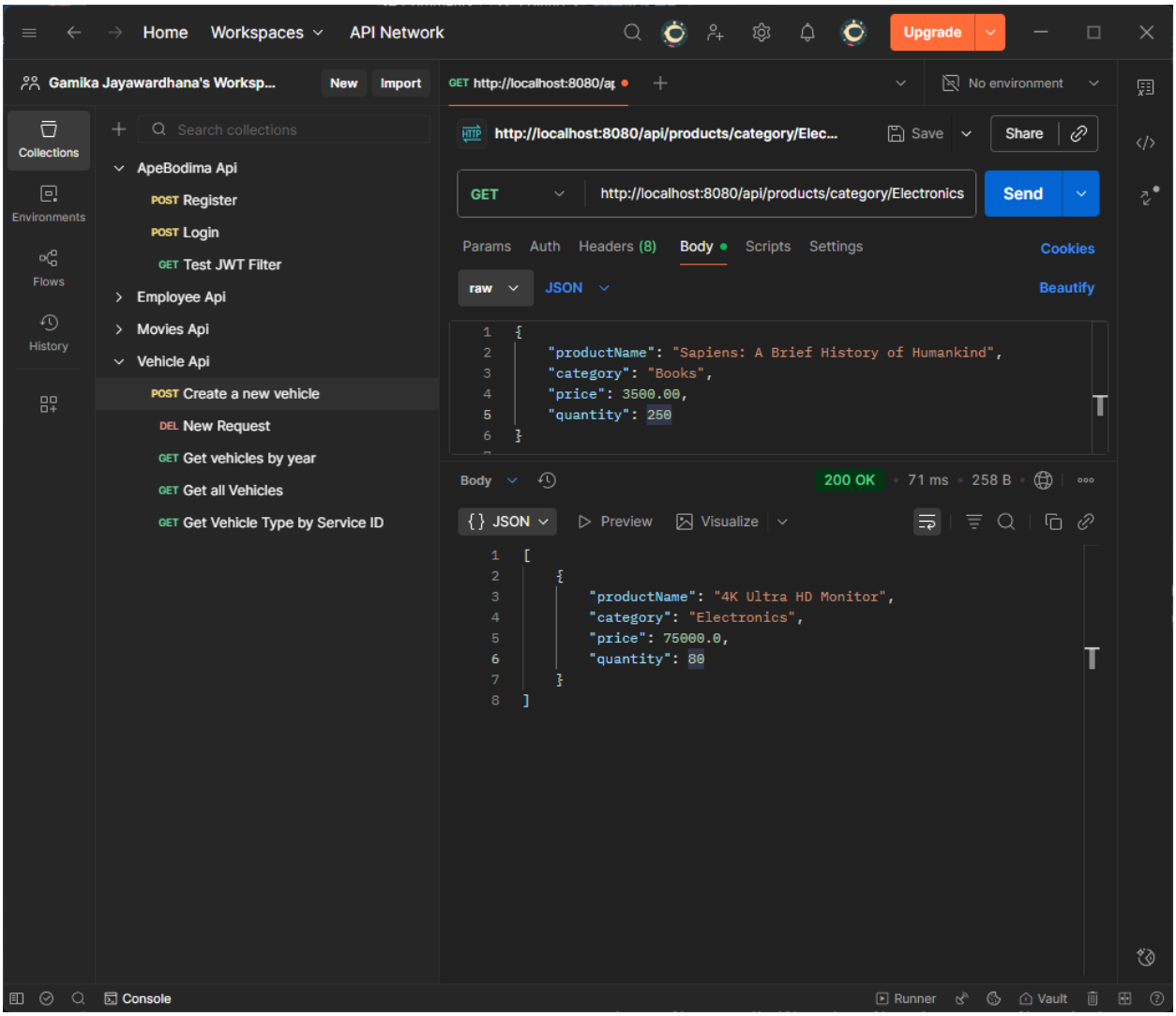
LAB 04

EC/2021/006 W.K.G.K JAYAWARDANA



LAB 04

EC/2021/006 W.K.G.K JAYAWARDANA



LAB 04

EC/2021/006 W.K.G.K JAYAWARDANA