JUnit Test Cases

ProductServiceTest.java

```
package com.example. filter_api.service;


import com.example. filter_api.model.Product;
import com.example. filter_api.repository.ProductRepository;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.MockitoAnnotations;
import org.springframework.data.domain.PageRequest;
import org.springframework.data.domain.Sort;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.mockito.Mockito.when;


public class ProductServiceTest {


    @Mock
    private ProductRepository productRepository;


    @InjectMocks
    private ProductService productService;


    @BeforeEach
    void setUp() {
        MockitoAnnotations.openMocks(this);
    }

    @Test
    void testGetFilteredAndSortedProducts() {
        Product product1 = new Product("1", "Smartphone", "electronics", 299.99, true, 4.5,
new Date());
        Product product2 = new Product("2", "Laptop", "electronics", 899.99, true, 4.7, new
Date());

        when(productRepository.findAll(PageRequest.of(0, Integer.MAX_VALUE,
Sort.by("price").ascending())))
            .thenReturn(Arrays.asList(product1, product2));

        List<Product> result = productService.getFilteredAndSortedProducts("electronics",
100.0, 1000.0, true, "price", "asc");

        assertEquals(2, result.size());
        assertEquals("Smartphone", result.get(0).getName());
        assertEquals("Laptop", result.get(1).getName());
    }

    @Test
    void testGetFilteredAndSortedProductsWithNoResults() {
        when(productRepository.findAll(PageRequest.of(0, Integer.MAX_VALUE,
Sort.by("price").ascending())))
            .thenReturn(Arrays.asList());

        List<Product> result = productService.getFilteredAndSortedProducts("electronics",
100.0, 1000.0, true, "price", "asc");
```

```
        assertEquals(0, result.size());
    }
}
```

ProductControllerTest.java

```
package com.example. filter_api.controller;

import com.example. filter_api.model.Product;
import com.example. filter_api.service.ProductService;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.MockitoAnnotations;
import org.springframework.http.ResponseEntity;

import java.util.Arrays;
import java.util.Date;
import java.util.List;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.mockito.Mockito.when;

public class ProductControllerTest {

    @Mock
    private ProductService productService;

    @InjectMocks
    private ProductController productController;

    @BeforeEach
    void setUp() {
        MockitoAnnotations.openMocks(this);
    }

    @Test
    void testGetProducts() {
        Product product1 = new Product("1", "Smartphone", "electronics", 299.99, true, 4.5,
new Date());
        Product product2 = new Product("2", "Laptop", "electronics", 899.99, true, 4.7, new
Date());

        when(productService.getFilteredAndSortedProducts("electronics", 100.0, 1000.0, true,
"price", "asc"))
            .thenReturn(Arrays.asList(product1, product2));

        List<Product> result = productController.getProducts("electronics", 100.0, 1000.0, true,
"price", "asc");

        assertEquals(2, result.size());
        assertEquals("Smartphone", result.get(0).getName());
        assertEquals("Laptop", result.get(1).getName());
    }

    @Test
    void testGetProductsWithNoResults() {
        when(productService.getFilteredAndSortedProducts("electronics", 100.0, 1000.0, true,
"price", "asc"))

            .thenReturn(Arrays.asList());


        List<Product> result = productController.getProducts("electronics", 100.0, 1000.0, true,
"price", "asc");


        assertEquals(0, result.size());

    }

}
```