**EXERCISE 4**

**Implement a POST Endpoint to Create a New Customer:**

**1.Define the Customer Entity:**

First, create a Customer entity class in the model package:

package com.example.bookstoreapi.model;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Customer {

private Long id;

private String name;

private String email;

private String address;

}

**2.Implement the POST Endpoint in CustomerController:**

* Create a new controller class named CustomerController in the controller package.
* Implement the POST method to accept a JSON request body and create a new customer.

package com.example.bookstoreapi.controller;

import com.example.bookstoreapi.model.Customer;

import org.springframework.web.bind.annotation.\*;

import java.util.ArrayList;

import java.util.List;

@RestController

@RequestMapping("/customers")

public class CustomerController {

private List<Customer> customers = new ArrayList<>();

@PostMapping

public Customer createCustomer(@RequestBody Customer customer) {

customers.add(customer);

return customer;

}

}

* @RequestBody Customer customer: This annotation binds the JSON request body to the Customer object.

**Form Data: Process Customer Registrations**

**Implement an Endpoint to Process Form Data:**

**1.Define the Endpoint in CustomerController:**

You can use @RequestParam to capture form data fields in the controller.

@PostMapping("/register")

public Customer registerCustomer(@RequestParam String name,

@RequestParam String email,

@RequestParam String address) {

Customer customer = new Customer();

customer.setId((long) (customers.size() + 1)); // Simple ID generation

customer.setName(name);

customer.setEmail(email);

customer.setAddress(address);

customers.add(customer);

return customer;

}

* @RequestParam String name, . These annotations bind form fields to method parameters.